

Navicat™

Version 11

ユーザーガイド



目次

Navicat を始めよう	8
システム要件	9
登録	10
インストール	10
メンテナンス/アップグレード	11
エンドユーザーライセンス使用許諾契約書	12
接続	23
Navicat Cloud	24
基本設定	27
詳細設定	30
SSL 設定	33
SSH 設定	34
HTTP 設定	35
サーバーオブジェクト	37
MySQL/MariaDB オブジェクト	37
MySQL /MariaDB テーブル	37
MySQL/MariaDB テーブルのフィールド	38
MySQL/MariaDB テーブルのインデックス	40
MySQL/MariaDB テーブルの外部キー	41
MySQL/MariaDB テーブルのトリガー	42
MySQL/MariaDB テーブルのオプション	43
MySQL/MariaDB ビュー	46
MySQL/MariaDB ファンクション/プロシージャ	47
MySQL/MariaDB イベント	49
Oracle オブジェクト	50
Oracle データポンプ (フルバージョンで使用可能)	51
Oracle データポンプエクスポート	51
Oracle データポンプインポート	55
Oracle デバッグ (フルバージョンで使用可能)	59
Oracle 物理属性/デフォルトストレージ特性の編集	60
Oracle テーブル	62
Oracle 標準テーブル	62
Oracle テーブルのフィールド	63
Oracle テーブルのインデックス	64
Oracle テーブルの外部キー	65
Oracle テーブルの UNIQUE 制約	66
Oracle テーブルのチェック制約	67
Oracle テーブルのトリガー	67
Oracle テーブルのオプション	69
Oracle 外部テーブル	70

Oracle 外部テーブルのフィールド	70
Oracle 外部テーブルの外部プロパティ	71
Oracle 外部テーブルへのアクセスパラメータ	71
Oracle 索引構成テーブル	72
Oracle 索引構成テーブルのオプション	72
Oracle ビュー	73
Oracle ファンクション/プロシージャ	74
Oracle データベースリンク	75
Oracle インデックス	76
Oracle Java	78
Oracle マテリアライズド・ビュー	79
Oracle マテリアライズド・ビュー・ログ	82
Oracle パッケージ	83
Oracle シーケンス	85
Oracle シノニム	85
Oracle トリガー	86
Oracle タイプ	89
Oracle XML スキーマ	90
Oracle リサイクルビン	91
Oracle ディレクトリ	92
Oracle テーブルスペース	92
Oracle パブリックデータベースリンク	95
Oracle パブリックシノニム	95
PostgreSQL オブジェクト	95
PostgreSQL デバッガ (フルバージョンで使用可能)	96
PostgreSQL スキーマ	98
PostgreSQL テーブル	98
PostgreSQL 標準テーブル	99
PostgreSQL テーブルのフィールド	99
PostgreSQL テーブルのインデックス	101
PostgreSQL テーブルの外部キー	102
PostgreSQL テーブルの UNIQUE 制約	103
PostgreSQL テーブルのチェック制約	104
PostgreSQL テーブルの除外制約	105
PostgreSQL テーブルのルール	106
PostgreSQL テーブルのトリガー	107
PostgreSQL テーブルのオプション	108
PostgreSQL 外部テーブル	109
PostgreSQL 外部テーブルのフィールド	110
PostgreSQL 外部テーブルのチェック制約	111
PostgreSQL 外部テーブルのトリガー	112
PostgreSQL 外部テーブルのテーブルオプション	113

PostgreSQL ビュー	114
PostgreSQL 関数(ファンクション)	115
PostgreSQL 集約関数	118
PostgreSQL 符号化方式変換	119
PostgreSQL ドメイン	119
PostgreSQL インデックス	121
PostgreSQL マテリアライズド・ビュー	122
PostgreSQL 演算子	123
PostgreSQL 演算子クラス	125
PostgreSQL シーケンス	126
PostgreSQL トリガー	127
PostgreSQL トリガー関数	129
PostgreSQL タイプ	131
PostgreSQL テーブルスペース	134
PostgreSQL キャスト	135
PostgreSQL 外部サーバー	135
PostgreSQL 言語	136
SQLite オブジェクト	137
SQLite テーブル	138
SQLite テーブルのフィールド	138
SQLite テーブルのインデックス	140
SQLite テーブルの外部キー	141
SQLite テーブルの UNIQUE 制約	142
SQLite テーブルのチェック制約	143
SQLite テーブルのトリガー	143
SQLite テーブルのオプション	144
SQLite ビュー	145
SQLite インデックス	146
SQLite トリガー	147
SQL プレビュー	148
保守	148
MySQL/MariaDB の保守	148
Oracle の保守	149
PostgreSQL の保守	153
SQLite の保守	154
テーブルビューワー	156
グリッドビュー	156
ナビゲーションバーの使用	156
レコードの編集	158
レコードのソート/検索/置換	163
レコードにフィルターをかける	165
Raw データの操作	165

テーブルグリッドの書式設定	166
フォームビュー(フルバージョンで使用可能)	168
補助エディタ	168
フィルターウィザード	168
クエリ	171
クエリビルダ(フルバージョンで使用可能)	171
クエリエディタ	173
エディタの高度機能	174
クエリの結果	177
クエリパラメータ	178
Oracle クエリのデバッグ(フルバージョンで使用可能)	178
モデル(Navicat Premium と Enterprise バージョンで使用可能)	179
モデルサイドバー	179
モデルエクスプローラーペイン	179
モデル履歴ペイン	180
モデルプロパティペイン	180
モデル概要ペイン	183
ダイアグラムキャンバス	183
テーブルの作成	183
ビューの作成	184
外部キーの作成	185
ラベルの作成	186
ノートの作成	187
イメージの作成	188
形状の作成	189
レイヤーの作成	190
ダイアグラムの書式設定	191
モデルのプレビューと印刷	192
リバースエンジニアリング	192
スクリプトの生成	193
SQL のエクスポートの基本設定	193
SQL のエクスポートの詳細設定	193
フォワードエンジニアリング	194
同期の種類を選択	194
スキーマ/オブジェクトと接続を選択	195
比較/実行オプションを選択	195
比較結果の表示	196
モデルの変換	197
モデル ヒントと Tips	197
高度なツール	200
インポートウィザード	200
インポートファイル形式の設定 (ステップ 1)	200

ソースファイルの名前の設定 (ステップ 2)	200
区切り文字の設定 (ステップ 3) - TXT、XML	200
追加オプションの設定 (ステップ 4) - TXT、XML、HTML	201
ターゲットテーブルの設定 (ステップ 5)	202
フィールド構造の適用とフィールドのマッピング (ステップ 6)	202
インポートモードの選択 (ステップ 7)	203
インポートの保存と確認 (ステップ 8)	204
エクスポートウィザード	204
エクスポートファイル形式の設定 (ステップ 1)	205
エクスポート先のファイル名の設定 (ステップ 2)	205
エクスポート対象フィールドの選択 (ステップ 3)	206
追加オプションの設定 (ステップ 4)	206
エクスポートの保存と確認 (ステップ 5)	207
データ転送(フルバージョンで使用可能)	207
データ転送の基本設定	208
データ転送の詳細設定 (同種サーバー間)	208
データ転送の詳細設定 (クロスサーバー)(Navicat Premium のみ使用可能)	211
データの同期 (フルバージョンで使用可能)	212
データの同期の基本設定	213
データの同期の詳細設定	213
構造の同期 (フルバージョンで使用可能)	213
構造の同期の基本設定	214
構造の同期の結果	216
バックアップ/復元 (フルバージョンで使用可能)	217
バックアップ	217
復元	218
SQL の抽出	219
バッチジョブ/スケジュール (フルバージョンで使用可能)	220
バッチジョブ/スケジュールの基本設定	221
バッチジョブ/スケジュールの詳細設定	222
バッチジョブコンバーター (Navicat Premium で使用可能)	223
SQL ファイルにダンプ/SQL ファイルの実行	223
データベース/スキーマ/テーブルの構造の印刷 (フルバージョンで使用可能)	223
サーバーセキュリティ	224
MySQL/MariaDB セキュリティ	224
MySQL/MariaDB ユーザーデザイナー	224
Oracle セキュリティ	226
Oracle ユーザーデザイナー	226
Oracle ロールデザイナー	228
PostgreSQL セキュリティ	229
PostgreSQL Server 7.3 ~ 8.0	229
PostgreSQL ユーザーデザイナー	229

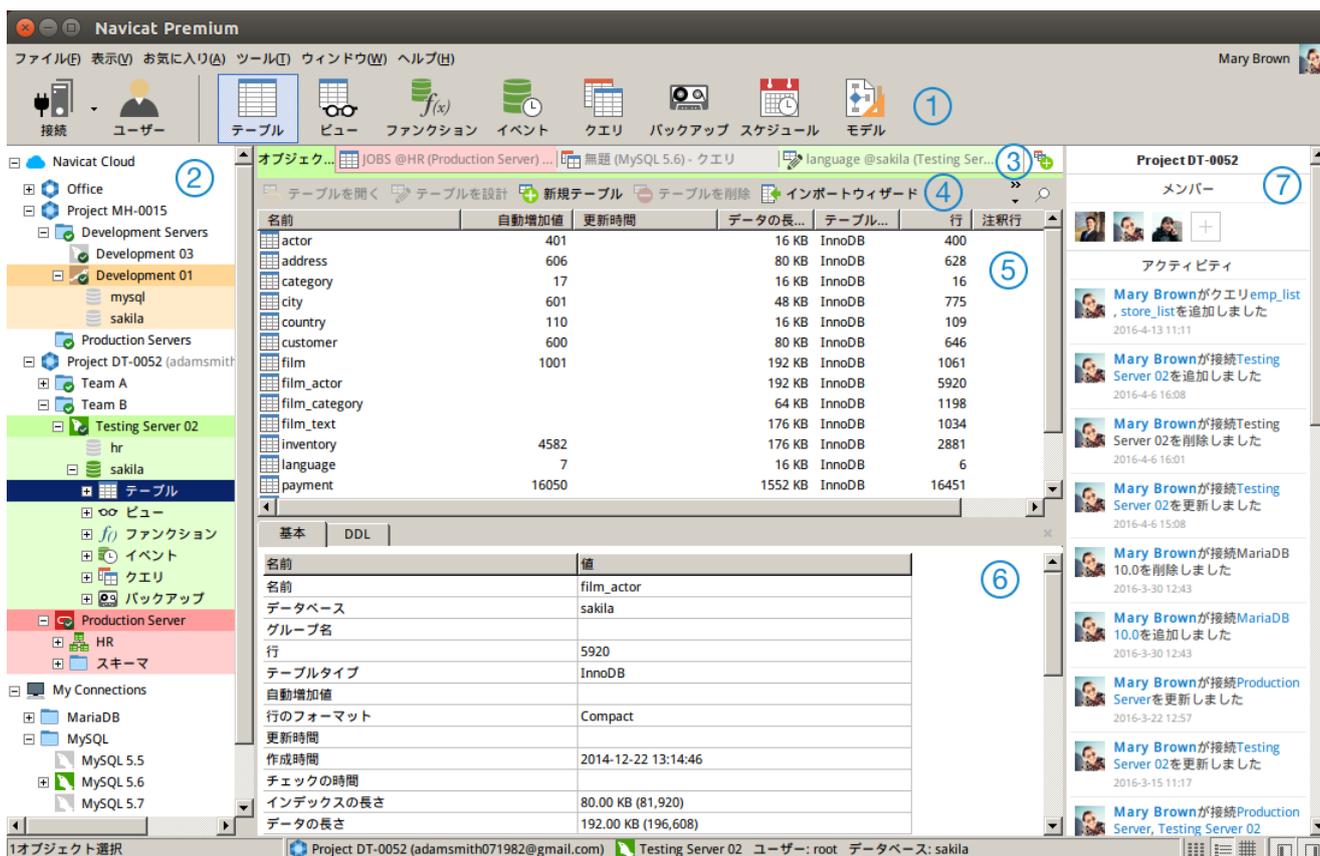
PostgreSQL グループデザイナー	230
PostgreSQL Server 8.1 以降	231
PostgreSQL ロールデザイナー	231
SQLite セキュリティ	233
SQLite ユーザーデザイナー	233
権限マネージャー	233
便利なツール	234
リスト/詳細/ER ダイアグラム ビュー	234
オブジェクト情報	236
サーバー監視 (フルバージョンで使用可能)	237
バーチャルグループ ping (フルバージョンで使用可能)	238
接続の色付け	238
お気に入り (フルバージョンで使用可能)	238
データベース/スキーマ内検索 (フルバージョンで使用可能)	239
検索フィルター	239
オプション	241
基本オプション	241
表示オプション	242
モデルオプション(フルバージョンで使用可能)	248
その他のオプション	248
コマンド(フルバージョンで使用可能)	250
ホットキー	251
ログファイル	254

Navicat を始めよう

Navicat は、マルチ接続データベース管理ツールです。MySQL/Oracle/PostgreSQL/SQLite/MariaDB データベースに同時に接続することができ、異なる種類の複数のデータベースに対するデータベース管理を非常に簡単にします。Amazon RDS と Amazon Redshift を管理することも可能です。Navicat の機能は、プロの開発者のどのような要望も満たすほど非常に高性能です。また、データベースサーバーの初心者ユーザーが簡単に理解できるようにも作られています。使いやすくデザインされた Navicat のグラフィカルユーザーインターフェイス(GUI)を使用すると、安全かつ簡単な方法で、高速かつ簡単に情報を作成、整理、アクセス、共有することができます。

Navicat は、3つのプラットフォーム - Microsoft Windows、Mac OS X、Linux で使用可能です。Navicat は、ユーザーをローカル/リモートサーバーに接続し、さらに、データの保守に関するプロセスを簡素化するためにデータモデリングやデータ転送、データ/構造の同期、インポート/エクスポート、バックアップ/復元、スケジュールといった様々な実用ツールを提供します。詳細については、弊社のウェブサイトをご覧ください: <https://www.navicat.com>

※ Navicat 日本語版に関しましては、株式会社スマートスタイル運営の Navicat 公式日本語ウェブサイトをご覧ください: <http://jp.navicat.com/>



① Navicat メインツールバー

Navicat メインツールバーは、基本的なオブジェクトや機能にアクセスするために使用されます。例えば、接続やユーザー、テーブル、バックアップ、スケジュールなど。小さいアイコンを使用する または キャプションを非表示にするためには、ツールバーを右クリックし、**大きいアイコンを使用** または **キャプションを表示**を無効にします。

② 接続

接続ペインは、接続やデータベース、データベースオブジェクトを操作するために使用されます。ツリー構造を採用し、ユーザーはポップアップメニューを使用してすばやく簡単にデータベースやそのオブジェクトに対して操作を行うことができます。[Navicat Cloud](#) 機能にログインすると、接続ペインは **Navicat Cloud** と **My Connections** の部分に分かれます。開かれたオブジェクトのみを表示するためには、メインメニューから**表示** -> **アクティブなオブジェクトのみを表示**を選択します。接続ペインを表示/非表示にするためには、メインメニューから**表示** -> **接続を表示**を選択します。

③ タブバー

タブバーは、オブジェクトリストとタブウィンドウを切り替えるために使用されます。また、ポップアップ画面を常に新しいタブに表示するか、常に新しいウィンドウに表示するかを選択することができます。複数のタブを開いている場合、CTRL+TAB を使って簡単に他のタブに切り替えられます。[オプション](#)もご参照ください。

④ オブジェクトリストツールバー

オブジェクトリストツールバーは、オブジェクトの操作に使用されます。

⑤ オブジェクトリスト

オブジェクトリストペインは、テーブルやビュー、クエリなどのオブジェクトのリストを表示します。

⑥ オブジェクト情報

オブジェクト情報ペインは、サーバーオブジェクトや Navicat オブジェクトの詳細な情報を表示します。オブジェクト情報ペインを表示/非表示にするためには、メインメニューから**表示** -> **オブジェクト情報を表示**を選択します。

⑦ Navicat Cloud アクティビティ

Navicat Cloud アクティビティペインは、プロジェクトのメンバーとアクティビティを表示します。接続ペインのプロジェクトまたはオブジェクトリストペインの Navicat Cloud オブジェクトを選択します。Navicat Cloud アクティビティペインを表示/非表示にするためには、メインメニューから**表示** -> **Navicat Cloud アクティビティを表示**を選択します。

システム要件

Windows 用システム要件

- Microsoft Windows XP SP3, Vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Server 2003, Server 2008, Server 2012

Mac OS X 用システム要件

- Mac OS X 10.7 Lion, 10.8 Mountain Lion, 10.9 Mavericks, 10.10 Yosemite, 10.11 EI Capitan
- Intel CPU

Linux 用システム要件

- Ubuntu 12.04, Ubuntu 14.04, CentOS 6.7, CentOS 7, Fedora 22, Fedora 23, Linux Mint 13, Linux Mint 17.3, openSUSE 13.2, openSUSE 42.1

登録

Navicat 日本語版をご利用のお客様用

- ・ライセンスキーやダウンロード情報、カスタマーセンターなどに関するお問い合わせや、ご意見、ご要望は、[Navicat お問い合わせ](#) からご連絡ください。
- ・ Navicat 日本語版のご注文は、[Navicat 日本語公式ウェブサイト](#) をご利用ください。
- ・ 日本語版のカスタマーセンターは、[こちら](#) をご利用ください。

Navicat 英語版をご利用のお客様用

PremiumSoft 社の製品を経済的かつ効率的に購入するために、95%以上のお客様は、PremiumSoft 社の[オンラインショップ](#)から、主要なクレジットカード - MasterCard、Visa、Euro card、JCB や American Express を使って Navicat を注文されます。オンラインの注文はすべて、**PayPal** と **WorldPay** によって処理されます。提供される SSL 処理用ペリサイン証明書によって、安全なオンライン取引が保証されています。

Navicat ソフトウェアを注文され、その注文情報をもう一度見たい場合や、注文、支払い、配送方法について質問がある場合、PremiumSoft 社の [Navicat 販売部](#)までご連絡ください。

ご購入後、PremiumSoft 社が注文を受領してから 24 時間以内に、ライセンスを受けた Navicat を有効にするための**ライセンスキー**を、e メールで送付いたします。注文時には、必ず有効な e メールアドレスを入力してください。24 時間以内にキーが送付されない場合、PremiumSoft 社が送付した e メールがお客様の e メールスパムフィルターによってブロックされた可能性があります。ダウンロード情報とキーを再送するために、お客様が登録された e メールアドレスを PremiumSoft 社の[カスタマーセンター](#)にお送りください。再送フォームから返信が全くない場合、PremiumSoft 社の [Navicat 販売部](#)までご連絡ください。

また、インターネット上での個人情報の提供が不安なお客様は、注文書と銀行振込/電子送金によるご注文を受け付けております。PremiumSoft 社の[オフライン注文](#)をご覧ください。

インストール

現在開いているアプリケーションがある場合は、シャットダウンすることを強くお勧めします。これによって、円滑にインストールが行えます。

注意: Navicat のインストールには、サーバーのインストールは含まれませんので、ご自身でダウンロードし、サーバーをインストールしてください。

未登録バージョンをお試しのユーザーの場合、ポップアップ登録画面に**ライセンスキー** (16 桁)を入力してください。

ダウンロードバージョンのインストール方法

1. **gzip** ファイルを開き、保存します。
2. 任意の場所にファイルを解凍します。
3. 解凍されたフォルダを開きます。
4. **start_navicat** をクリックして Navicat を起動します。起動にはしばらく時間がかかります。

- その後、登録画面がポップアップで表示されます。**キー**にキーを入力し、**アクティベーション実行**をクリックして、キーをオンラインアクティベーションします。

CD バージョンのインストール方法

- CD-ROM ドライブから Navicat CD インストールディスクを読み込みます。
- 任意の場所に **gzip** ファイルを解凍します。
- 解凍されたフォルダを開きます。
- start_navicat** をクリックして Navicat を起動します。起動にはしばらく時間がかかります。
- その後、登録画面がポップアップで表示されます。**キー**にキーを入力し、**アクティベーション実行**をクリックして、キーをオンラインアクティベーションします。

Navicat を新しいコンピュータに移動する

- Navicat で、**ファイル** -> **接続をエクスポート**を選択します。エクスポートファイル(.ncx)に全ての接続設定が含まれます。
- エクスポートファイル(.ncx)をバックアップします。
- Navicat で、**ヘルプ** -> **登録**を選択し、**アクティベーションを解除する**をクリックして、オンラインでキーのアクティベーションを解除します。
- 現行のコンピュータから、Navicat をアンインストールします。
- 新しいコンピュータに Navicat をインストールしなおします。
- 新しいコンピュータで Navicat を開き、**ファイル** -> **接続をインポート**を選択します。

新しい接続が確立されると、Navicat は[設定の保存場所](#)の下にサブフォルダを作成します。ほとんどのファイルはこのサブフォルダに保存されます。パスを探すためには、接続を右クリックし、**接続を編集** -> **詳細** -> **設定の保存場所**を選択します。

また、保存される全てのプロファイルは [profiles](#) の下に保存されます。パスを探すためには、**ツール** -> **オプション** -> **その他** -> **プロファイルの保存場所**を選択します。

メンテナンス/アップグレード

ソフトウェアメンテナンスの購入方法

Navicat ユーザーは、Navicat ソフトウェアメンテナンスを購入すると、その契約期間中は追加料金なしで、ソフトウェアアップグレードやバグフィックスリリースを受け取ることができ、Eメールによるサポートを優先的に受けることができます。

ソフトウェアメンテナンスは、ソフトウェアライセンスの購入日または、その購入日から 90 日以内に申し込むことができます - それ以降に、以前購入された製品に対してソフトウェアメンテナンスを付けることはできません。詳細については、[ソフトウェアメンテナンス](#)をご覧ください。

Navicat のアップグレード方法

インストール済みの Navicat を最新バージョンにアップグレードしたい場合、**ヘルプ** -> **更新の確認**を選択し、アップデーターを起動してください。アップデーターは、インストール済みのバージョンを自動的に確認します。そして、新しいバージョンがリリ

ースされている場合、アップデーターの指示に従って、お使いの Navicat をアップグレードしてください。最新バージョンが前のバージョンに置き換わりませんが、設定は変更されずに残ります。

または、登録 e メールアドレスで [カスタマーセンター](#) にアクセスし、最新バージョンのインストーラをダウンロードすることも可能です。

注意: Ver.7.x 以前の Navicat から最新版へアップグレードされる場合、元の設定を自動的に引き継ぐことができません。お問い合わせください。

エンドユーザーライセンス使用許諾契約書

注意: Navicat Cloud サービスのライセンス使用許諾契約につきましては、[こちら](#)をクリックしてください。

重要: 本ソフトウェア エンドユーザーライセンス使用許諾契約は、お客様(個人、あるいは購入または取得の目的となる組織)とプレミアムソフト社との間の契約です。本ソフトウェアをインストールし使用する前に以下の条項をよくお読みください。本契約は本ソフトウェアを使用するライセンスを提供し、保証情報と責任制限を含んでいます。本ソフトウェアをインストールし使用することにより、お客様は本契約に同意したものとみなされます。本契約に同意されない場合、本ソフトウェアをインストールせず購入先に返品してください。本ソフトウェアの中で他のソフトウェアが参照、あるいは言及されていたとしても、本契約はプレミアムソフト社から提供されたソフトウェアにのみ適用されます。

1. 定義

- a. 「教育版」とは、(i)企業、会社、提携会社や関連会社、またはその他の事業体や団体ではない個人 (ii)学生、教員、または教育機関の関係者 (iii)非営利団体または慈善団体のスタッフに限り使用を許諾した、本ソフトウェアを意味します。この定義条項の目的として、「教育機関」とは、公立または私立学校、専門学校、大学そして他の中等教育機関を意味します。非営利団体とは、問題や事態をサポートする目的とした、個人会社や非商用的な公共事業を意味します。
- b. 「再販禁止版」とは、レビューと評価用にのみ使用される、再販禁止版と名づけられた本ソフトウェアのことです。
- c. 「PremiumSoft」とは、PREMIUMSOFT CYBERTECH 社とそのライセンス付与者のことを意味します。
- d. 「本ソフトウェア」とは、PremiumSoft 社が提供する、PremiumSoft 社のソフトウェアプログラムとサードパーティ製のソフトウェアプログラム、対応するドキュメント、関連するメディア、印刷物、そしてオンライン、あるいは電子媒体のドキュメントを意味します。
- e. 「未登録版」、「試用版」または「デモ版」とは、未登録ソフトウェアを初めてインストールしてから 14 日間、評価目的のために使用される、登録されていないソフトウェア ("未登録ソフトウェア") を意味します。試用期間が終了すると、そのユーザーは、本ソフトウェアを登録するか、インストールしたシステムからそれを削除しなければならない。未登録ソフトウェアは、評価目的のために他のユーザーへ自由にコピーして配布できます。
- f. 「Navicat Essentials」は、商用目的で使用することが認められているソフトウェアのバージョンを意味します。

2. ライセンスの許諾

この第 2 章で許諾されたライセンスは、本契約に記述されている諸条件が適用されます。

- a. 第 2 章(b)により、お客様は本ソフトウェアを一台のコンピュータにインストールして使用できます。また、お客様が本ソフトウェアをインストールし使用する、個々のコンピュータ毎にライセンスをお持ちの場合は、ネットワークサーバのようなストレージ機器に本ソフトウェアをインストールして保存し、内部ネットワークを通じて他のコンピュータに本ソフ

トウェアをインストールすることができます。第2章(b)の範囲以外では、本ソフトウェアの一ライセンスは、異なるコンピュータ上で同時に共有、インストール、あるいは使用できません。

- b. 第2章(a)で許可された本ソフトウェアの複製に加えて、本ソフトウェアがインストールされたコンピュータのプライマリユーザは、以下の条件下でのみ、本ソフトウェアの複製物を作成し、その人個人が使用する場合に限り、それを携帯用コンピュータ、あるいは自宅のコンピュータにインストールできます。
 - A. 携帯用コンピュータ、あるいは自宅のコンピュータ上の本ソフトウェアの複製物は、(i) プライマリコンピュータ上の本ソフトウェアの複製と同時に使用できません。そして(ii) 当該のバージョンもしくはエディション(例えば教育利用のみ)が許可されたプライマリユーザのみ使用できます。
 - B. 当該ユーザが本ソフトウェアがインストールされたプライマリコンピュータのプライマリユーザではなくなった後は、本ソフトウェアの複製物をインストールも使用もできません。
- c. 本ソフトウェアがパッケージソフト製品(総称して「Studio」)の一部として、プレミアムソフト社の他のソフトウェア製品といっしょに配布された場合は、「Studio」は一つの製品としてライセンス供与され、本ソフトウェアを含む全てのStudio内の製品は、複数台のコンピュータでのインストール、または使用はできません。
- d. お客様はバックアップ目的で本ソフトウェアの複製を、コンピュータが読取可能な形式で一つだけ作成できます。お客様は本ソフトウェアの元の複製に付随する全ての複製権表示とその他の所有権の注意事項をどの複製にも再作成しなければなりません。お客様はバックアップ目的で作成した本ソフトウェアの複製を販売、譲渡できません。
- e. お客様は、妥当な通知を行うことにより、PremiumSoft社がお客様がこれらの条項を遵守して本ソフトウェアを使用しているかどうか、いかなる時にも検査できることに同意します。当該の検査によりお客様が本契約の条項に違反して本ソフトウェアを使用していることが判明した場合、お客様は、この検査と条項違反の結果として負担する可能性のあるその他の義務に加えて、関連する全ての妥当な費用を、PremiumSoft社に払わなければなりません。
- f. 本契約下のお客様のライセンス権は、独占的ではありません。

3. ライセンス制限

- a. 第2章に記載された範囲を超えて、お客様は、本ソフトウェアの複製の作成あるいは配布をしたり、あるコンピュータから他のコンピュータへ、ネットワークを通じて電子媒体として本ソフトウェアを転送できません。
- b. お客様は本ソフトウェアを変更、併合、修正、改変、あるいは翻訳することができません。あるいは、逆コンパイル、リバースエンジニアリング、逆アセンブル、または本ソフトウェアを人間が読める形式に変更することはできません。
- c. ここで説明されている範囲外で、お客様は本ソフトウェアを貸したり、譲渡したり、あるいはライセンス権を付与したりできません。
- d. 本ソフトウェアの評価/デモ版そして Lite 版、あるいは再販禁止版を除いて、複製を一切持たず本ソフトウェアを全て譲渡するのであれば、お客様は本契約に基づき販売あるいは譲渡の一部として全ての権利を永久に譲渡することができます(全てのコンポーネント、媒体と印刷物、いかなるアップグレード、当該契約、シリアル番号、そして可能であれば、本ソフトウェアと共に提供されたその他の全てのソフトウェア製品を含む)。そして、受領者はこの契約の条項に同意します。本ソフトウェアがアップグレードであれば、いかなる譲渡も本ソフトウェアの以前の全てのバージョンを含まなければなりません。本ソフトウェアの複製が(上記に記載された)「Studio」の一部としてライセンス供与されている場合、本ソフトウェアは、個別にではなく全 Studio 製品の販売あるいは譲渡としてのみ、譲渡されなければなりません。お客様は本ソフトウェアの複製を保持することはできません。お客様は、本ソフトウェアの評価/デモ版そして Lite 版も再販禁止版も販売、あるいは譲渡できません。
- e. ここに規定された範囲外では、お客様は本ソフトウェアを変更、あるいは本ソフトウェアに基づいた派生物を作成することはできません。
- f. 教育版は、商用目的では、いかなる相手にも使用または配布できません。
- g. ここに規定された範囲外では、お客様は、

- A. 本ソフトウェアの評価/デモ版そして Lite 版の複製を合計して二つ以上インストール、あるいは使用してはなりません。
- B. ユーザ名を二つ以上使用して本ソフトウェアの評価/デモ版そして Lite 版をダウンロードしてはなりません。
- C. 本ソフトウェアの評価/デモ版の使用期間を、一ライセンスに与えられた評価期間を超えて使用できるよう ハードドライブの内容、あるいはコンピュータシステムを変更してはなりません。
- D. PremiumSoft 社の事前の書面による許可なく、評価/デモ版そして Lite 版を用いた性能検証の結果を第 3 者に公開してはなりません。
- E. 本ソフトウェアの商用版、あるいはに対するライセンスを購入するかどうかを決定する目的以外の目的で、本ソフトウェアの評価/デモ版を使用してはなりません。しかしながら、上記にかかわらず、お客様が、いかなる商用トレーニング目的で本ソフトウェアの評価/デモ版または Lite 版をインストール、あるいは使用することは厳しく禁止されています。
- h. お客様は本ソフトウェアをレビュー、評価するためにのみ、本ソフトウェアの再販禁止版を使用することができます。
- i. お客様は本ソフトウェアを二つ以上の媒体で受け取ることがありますが、一つの媒体しかインストール、あるいは使用できません。お客様が受け取った媒体の数にかかわらず、本ソフトウェアがインストールされる予定のサーバ、あるいはコンピュータに適した媒体のみ使用できます。
- j. お客様は複数のプラットフォーム用の本ソフトウェアを受け取ることがありますが、一つのプラットフォームにのみインストール、あるいは使用するものとします。
- k. お客様は本ソフトウェアを使用して、本ソフトウェアと同じ主要機能を持ついかなるアプリケーションも開発してはなりません。
- l. お客様が本契約に従わない場合、PremiumSoft 社はお客様のライセンスを終了させることができます。その場合、お客様は本ソフトウェアと、その全ての複製を破棄しなければなりません。(このように、ライセンスが終了された後も、本契約の両者の権利と条項は全て有効です。)
- m. This program may include Oracle Instant Client (OCI). You agree that you shall
 - 1. not use of the Oracle Instant Client to the business operations;
 - 2. not assign, give, or transfer the Oracle Instant Client or an interest in them to another individual or entity;
 - a. make the Programs available in any manner to any third party for use in the third party's business operations; and
 - b. title to the Programs from passing to the end user or any other party;
 - 3. not reverse engineer, disassemble or decompilation the Oracle Instant Client and duplicate the Programs except for a sufficient number of copies of each Program for your licensed use and one copy of each Program media;
 - 4. discontinue use and destroy or return to all copies of the Oracle Instant Client and documentation after termination of the Agreement;
 - 5. not publish any results of benchmark tests run on the Programs;
 - 6. comply fully with all relevant export laws and regulations of the United States and other applicable export and import laws to assure that neither the Oracle Instant Client, nor any direct product thereof, are exported, directly or indirectly, in violation of applicable laws;
 - 7. allow PremiumSoft to audit your use of the Oracle Instant Client;

4. アップグレード

本ソフトウェアの複製が、本ソフトウェアの以前のバージョンからのアップグレード版の場合、それは、ライセンス交換に基づいてお客様に提供されています。お客様は本ソフトウェアの当該の複製をインストール、あるいは使用することにより、任意に以前の契約を終了させ、本ソフトウェアの以前のバージョンを続けて使用しないこと、あるいは第3章に準拠する譲渡を除いては、それを他の個人または組織に譲渡しないことに同意します。

5. 所有権

先述のライセンスは、本ソフトウェアを使用するに当たり制限されたライセンスをお客様に供与します。PremiumSoft 社とその供給者は、(お客様が開発したアプリケーションの基礎となる、独立した著作物として、基本となる著作物としての)本ソフトウェアとその全ての複製に対して、全ての著作権と知的財産権を含み、全ての権利、名称、そして利権を保持します。連邦著作権と国際著作権を含み、本契約に明示的に供与されていない全ての権利は、PremiumSoft 社とその供給者に保有されます。

6. 限定保証と免責条項

- a. 本ソフトウェアの評価/デモ版そして Lite 版と再販禁止版を除いて、PremiumSoft 社は、配布日(お客様のレシートのコピーで確認)から 30 日間、本ソフトウェアが提供された物理媒体が通常の使用において材料および仕上がりに欠陥がないことを保証します。本ソフトウェアは「保証なし」で提供されます。明示的でも黙示的でも、取引の過程や商慣習から発生しても、法的にも、いかなることにしても、PremiumSoft 社は一切の保証を負いません。
- b. PREMIUMSOFT 社は本ソフトウェアの評価/デモ版そして Lite 版と再販禁止版については、明示的であっても黙示的であっても、PremiumSoft 社は何の救済策も何の保証も提供しません。本ソフトウェアの評価/デモ版そして Lite 版と再販禁止版は「保証なし」で提供されます。
- c. 本ソフトウェアの評価/デモ版そして Lite 版と再販禁止版以外の本ソフトウェアに関する前述の限定保証での定義を除いて、市販性、特定目的との適合性を含む明示的、暗示的、もしくはそれ以外の保証をすることなく、PremiumSoft 社とその供給者はすべての他の保証と言明事項を破棄できます。。また、権利と所有権を侵害していないこと、または安居権を保証致しません。PremiumSoft 社は本ソフトウェアにバグがないこと、あるいは障害なく動作することを保証致しません。本ソフトウェアは、デザイン、工事、核施設の維持運用、飛行機の運航、通信システム、航空管制、生命維持、武器システムを含み、フェイルセーフな制御を要求する危険な環境での使用を目的に設計、および、それを目的としていません。また、ライセンス供与されてもいません。PremiumSoft 社は明確に、そのような目的への適性を、明示的にも黙示的にも保証致しません。
- d. 準拠法が本ソフトウェアに関するなんらかの保証を要求する場合、全ての当該の保証は配布日から 30 日間に限定されます。
- e. PremiumSoft 社、ディーラー、販売業者、代理人、あるいは従業員から与えられた口頭の、あるいは書面の情報、あるいはアドバイスは、保証を提供されるものではなく、また、ここで提供される保証範囲を広げるものでもありません。

7. 賠償責任の制限

(a) 例え Premiumsoft もしくはその販売代理人がそのような損失の可能性を助言したとしても、PremiumSoft 社もその供給者も、本ソフトウェアの使用または使用できないことから生じ、契約違反、保証違反、(過失を含む)不法行為、その他の製品責任を含むいかなる見解に基づく、いかなる間接的、特別の、付随的、懲罰的、あるいは結果的に生じる損害(これだけに限られないが、機器使用不能、あるいはデータアクセス不能に対する損害、ビジネス損失、利益の損失、事業中断、あるいは同種のものを含む)に対して、お客様に対してもサードパーティに対しても責任を負わないものとします。

8. サードパーティソフトウェア

本ソフトウェアは注意事項、追加条項や条件を要求するサードパーティ製のソフトウェアを含んでいることがあります。本契約を受け入れることによって、お客様はそのサードパーティ製ソフトウェアの追加条項や条件を受け入れることとなります。

9. 一般条項

PremiumSoft 社のディーラー、代理人、あるいは従業員は、本契約に修正を要求する権利を持ちません。

本契約は、本件に関する当事者間の完全な同意を含んでおり、口頭であれ書面であれ、本契約締結以前のすべての協定や合意より優先します。お客様は、購入注文、あるいは書面での通知、あるいはライセンスされた本ソフトウェアに関連してお客様が発行したドキュメントに含まれる様々な条項、あるいは追加条項には効力がありません。PremiumSoft 社が、本契約に基づく権利の実行に失敗や遅延、あるいは本契約を違反しても、それらの権利の放棄や、違反とはみなされません。

本契約のいずれかの条項が管轄裁判所によって法に反すると判定された場合、この条項は許容の最大範囲において施行され、本契約のその他の条項は引き続き効力を持ちます。

10. 売買の基礎

上記で説明された限定保証と免責条項、そして有限責任は、PremiumSoft 社とお客様との間の同意の基礎となる、基本要素です。PremiumSoft 社はこの上記の制限無しで経済活動として本ソフトウェアを提供することはできません。上記の限定保証と免責条項、有限責任は PremiumSoft 社のライセンス保持者の利益に役立ちます。

11. 期限

本ソフトウェアをダウンロード、インストールすることにより、ライセンス付与者は本契約の規約に合意したものとします。

このライセンスは、終了するまでは有効です。ライセンス付与者は、このライセンスのいずれかの条項に同意できなければ、即座にお客様のライセンスを終了させる権利を持っています。

「保証なし」において。ライセンス付与者は、明示的でも黙示的でも、取引の過程や商慣習から発生しても、法的にも、いかなることにしても、PremiumSoft 社は一切の保証を負いません。

12. 準拠法

このライセンスは香港で施行されている法律に準拠しています。お客様は本契約から発生する論争を解決するために、香港の、非独占的司法権と裁判地に同意します。

お客様がこのライセンスの有効性について質問がございましたら、licensing@navicat.com にご連絡ください。お客様がその他の理由のためライセンス付与者に連絡したい場合は、support@navicat.com に連絡してください。

本ソフトウェアに含まれる PremiumSoft とその他の商標は、合衆国とその他の国で登録済みの PremiumSoft CyberTech 社の商標あるいは商標登録です。サードパーティの商標、商標名、製品名やロゴは商標、あるいは登録商標です。お客様は本ソフトウェアのいかなる商標、商標名、製品名、ロゴ、著作権、あるいはその他の著作権の注意事項、注釈、記号、ラベルを削除、あるいは変更してはいけません。本契約は、お客様が PremiumSoft 社、あるいはそのライセンス付与者の名称、あるいはそれらのトレードマークを使用する権利を認可していません。

(注)End-User License Agreement for PremiumSoft Navicat は英文が正文です。日本語で書かれたこのエンドユーザーライセンス使用許諾契約は参考のため英文を翻訳したものであり、不明な点は以下の英文をご参照ください。

End-User License Agreement

Note: For the License Agreement of Navicat Cloud service, please click [here](#).

IMPORTANT: THIS SOFTWARE END USER LICENSE AGREEMENT ("EULA") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR, IF PURCHASED OR OTHERWISE ACQUIRED BY OR FOR AN ENTITY, AN ENTITY) AND PREMIUMSOFT CYBERTECH LTD..READ IT CAREFULLY BEFORE COMPLETING THE INSTALLATION PROCESS AND USING THE SOFTWARE. IT PROVIDES A LICENSE TO USE THE SOFTWARE AND CONTAINS WARRANTY INFORMATION AND LIABILITY DISCLAIMERS. BY INSTALLING AND USING THE SOFTWARE, YOU ARE CONFIRMING YOUR ACCEPTANCE OF THE SOFTWARE AND AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO BE BOUND BY THESE TERMS, THEN DO NOT INSTALL THE SOFTWARE AND RETURN THE SOFTWARE TO YOUR PLACE OF PURCHASE. THIS EULA SHALL APPLY ONLY TO THE SOFTWARE SUPPLIED BY PREMIUMSOFT CYBERTECH LTD. HEREWITH REGARDLESS OF WHETHER OTHER SOFTWARE IS REFERRED TO OR DESCRIBED HEREIN.

1. Definitions

- a. "Non-commercial Version" means a version of the Software, so identified, for use by i) the individual who is a natural person and not a corporation, company, partnership or association or other entity or organization (ii) the individual who is a student, faculty or staff member at an educational institution, and (iii) staff of a non-profit organization or charity organization only. For purposes of this definition, "educational institution" means a public or private school, college, university and other post secondary educational establishment. A non-profit organization is an organization whose primary objective is to support an issue or matter of private interest or public concern for non-commercial purposes.
- b. "Not For Resale (NFR) Version" means a version, so identified, of the Software to be used to review and evaluate the Software, only.
- c. "PremiumSoft" means PREMIUMSOFT CYBERTECH LTD. and its licensors, if any.
- d. "Software" means only the PremiumSoft software program(s) and third party software programs, in each case, supplied by PremiumSoft herewith, and corresponding documentation, associated media, printed materials, and online or electronic documentation.
- e. "Unregistered version", "Trial version" or "Demo version" means an unregistered copy of the SOFTWARE ("UNREGISTERED SOFTWARE") which may be used by the USER for evaluation purposes for a period of fourteen (14) days following the initial installation of the UNREGISTERED SOFTWARE. At the end of the trial period ("TRIAL PERIOD"), the USER must either register the SOFTWARE or remove it from his system. The UNREGISTERED SOFTWARE may be freely copied and distributed to other users for their evaluation.
- f. "Navicat Essentials" means a version of the Software, so identified, to be used for commercial purpose.

2. License Grants

The licenses granted in this Section 2 are subject to the terms and conditions set forth in this EULA:

- a. Subject to Section 2(b), you may install and use the Software on a single computer; OR install and store the Software on a storage device, such as a network server, used only to install the Software on your other computers over an internal network, provided you have a license for each separate computer on which the Software is installed and run. Except as otherwise provided in Section 2(b), a license for the Software may not be shared, installed or used concurrently on different computers.
- b. In addition to the single copy of the Software permitted in Section 2(a), the primary user of the computer on which the Software is installed may make a second copy of the Software and install it on either a portable computer or a computer located at his or her home for his or her exclusive use, provided that:
 - A. the second copy of the Software on the portable or home computer (i) is not used at the same time as the copy of the Software on the primary computer and (ii) is used by the primary user solely as allowed for such version or edition (such as for educational use only),
 - B. the second copy of the Software is not installed or used after the time such user is no longer the primary user of the primary computer on which the Software is installed.
- c. In the event the Software is distributed along with other PremiumSoft software products as part of a suite of products (collectively, the "Studio"), the license of the Studio is licensed as a single product and none of the products in the Studio, including the Software, may be separated for installation or use on more than one computer.
- d. You may make one copy of the Software in machine-readable form solely for backup purposes. You must reproduce on any such copy all copyright notices and any other proprietary legends on the original copy of the Software. You may not sell or transfer any copy of the Software made for backup purposes.
- e. You agree that PremiumSoft may audit your use of the Software for compliance with these terms at any time, upon reasonable notice. In the event that such audit reveals any use of the Software by you other than in full compliance with the terms of this Agreement, you shall reimburse PremiumSoft for all reasonable expenses related to such audit in addition to any other liabilities you may incur as a result of such non-compliance.
- f. Your license rights under this EULA are non-exclusive.

3. License Restrictions

- a. Other than as set forth in Section 2, you may not make or distribute copies of the Software, or electronically transfer the Software from one computer to another or over a network.
- b. You may not alter, merge, modify, adapt or translate the Software, or decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form.
- c. Unless otherwise provided herein, you may not rent, lease, or sublicense the Software.
- d. Other than with respect to a Trial / Demo Version, Non-commercial Lite Version or a Not For Resale Version of the Software, you may permanently transfer all of your rights under this EULA only as part of a sale or transfer, provided you retain no copies, you transfer all of the Software (including all component parts, the media and printed materials, any upgrades, this EULA, the serial numbers, and, if applicable, all other software products provided together with the Software), and the recipient agrees to the terms of this EULA. If the Software is an upgrade, any transfer must include all prior versions of the Software from which you are upgrading. If the copy of the Software is licensed as part of the whole Studio (as defined above), the Software shall be transferred only with and as part of the sale or transfer of the whole Studio, and not separately. You may retain no copies of the Software. You may not sell or transfer any Trial / Demo Version, Non-commercial Lite Version or Not For Resale Version of the Software.

- e. Unless otherwise provided herein, you may not modify the Software or create derivative works based upon the Software.
- f. Non-commercial Versions of the Software may not be used for, or distributed to any party for, any commercial purpose.
- g. Unless otherwise provided herein, you shall not
 - A. in the aggregate, install or use more than one copy of the Trial / Demo Version and Non-commercial Lite Version of the Software,
 - B. download the Trial / Demo Version and Non-commercial Lite Version of the Software under more than one username,
 - C. alter the contents of a hard drive or computer system to enable the use of the Trial / Demo Version of the Software for an aggregate period in excess of the trial period for one license to such Trial / Demo Version,
 - D. disclose the results of software performance benchmarks obtained using the Trial / Demo Version or Non-commercial Lite Version to any third party without PremiumSoft prior written consent, or
 - E. use the Trial / Demo Version of the Software for a purpose other than the sole purpose of determining whether to purchase a license to a commercial or education version of the software; provided, however, notwithstanding the foregoing, you are strictly prohibited from installing or using the Trial / Demo Version or Non-commercial Lite Version of the Software for any commercial training purpose.
- h. You may only use the Not for Resale Version of the Software to review and evaluate the Software.
- i. You may receive the Software in more than one medium but you shall only install or use one medium. Regardless of the number of media you receive, you may use only the medium that is appropriate for the server or computer on which the Software is to be installed.
- j. You may receive the Software in more than one platform but you shall only install or use one platform.
- k. You shall not use the Software to develop any application having the same primary function as the Software.
- l. In the event that you fail to comply with this EULA, PremiumSoft may terminate the license and you must destroy all copies of the Software (with all other rights of both parties and all other provisions of this EULA surviving any such termination).
- m. This program may include Oracle Instant Client (OCI). You agree that you shall
 - 1. not use of the Oracle Instant Client to the business operations;
 - 2. not assign, give, or transfer the Oracle Instant Client or an interest in them to another individual or entity;
 - a. make the Programs available in any manner to any third party for use in the third party's business operations; and
 - b. title to the Programs from passing to the end user or any other party;
 - 3. not reverse engineer, disassemble or decompilation the Oracle Instant Client and duplicate the Programs except for a sufficient number of copies of each Program for your licensed use and one copy of each Program media;
 - 4. discontinue use and destroy or return to all copies of the Oracle Instant Client and documentation after termination of the Agreement;
 - 5. not publish any results of benchmark tests run on the Programs;

6. comply fully with all relevant export laws and regulations of the United States and other applicable export and import laws to assure that neither the Oracle Instant Client, nor any direct product thereof, are exported, directly or indirectly, in violation of applicable laws;
7. allow PremiumSoft to audit your use of the Oracle Instant Client;

4. Upgrades

If this copy of the Software is an upgrade from an earlier version of the Software, it is provided to you on a license exchange basis. You agree by your installation and use of such copy of the Software to voluntarily terminate your earlier EULA and that you will not continue to use the earlier version of the Software or transfer it to another person or entity unless such transfer is pursuant to Section 3.

5. Ownership

The foregoing license gives you limited license to use the Software. PremiumSoft and its suppliers retain all rights, title and interest, including all copyright and intellectual property rights, in and to, the Software (as an independent work and as an underlying work serving as a basis for any application you may develop), and all copies thereof. All rights not specifically granted in this EULA, including Federal and International Copyrights, are reserved by PremiumSoft and its suppliers.

6. LIMITED WARRANTY AND DISCLAIMER

- a. Except with respect to Trial / Demo Version, Non-commercial Lite Version and Not For Resale Version of the Software, PremiumSoft warrants that, for a period of thirty (30) days from the date of delivery (as evidenced by a copy of your receipt): the physical media on which the Software is furnished will be free from defects in materials and workmanship under normal use. The Software is provided "as is". PremiumSoft makes no warranties, express or implied, arising from course of dealing or usage of trade, or statutory, as to any matter whatsoever.
- b. PremiumSoft provides no remedies or warranties, whether express or implied, for Trial / Demo version, Non-commercial Lite version and the Not for Resale version of the Software. Trial / Demo version, Non-commercial Lite version and the Not for Resale version of the Software are provided "as is".
- c. Except as set Forth in the foregoing limited warranty with respect to software other than Trial/ Demo version, Non-commercial Lite version and Not for Resale version, PremiumSoft and its suppliers disclaim all other warranties and representations, whether express, implied, or otherwise, including the warranties of merchantability or fitness for a particular purpose. Also, there is no warranty of non-infringement and title or quiet enjoyment. PremiumSoft does not warrant that the Software is error-free or will operate without interruption. The Software is not designed, intended or licensed for use in hazardous environments requiring fail-safe controls, including without limitation, the design, construction, maintenance or operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, and life support or weapons systems. PremiumSoft specifically disclaims any express or implied warranty of fitness for such purposes.
- d. If applicable law requires any warranties with respect to the Software, all such warranties are limited in duration to thirty (30) days from the date of delivery.
- e. No oral or written information or advice given by PremiumSoft, its dealers, distributors, agents or employees shall create a warranty or in any way increase the scope of ANY warranty PROVIDED HEREIN.

7. LIMITATION OF LIABILITY

(a) Neither PremiumSoft nor its suppliers shall be liable to you or any third party for any indirect, special, incidental, punitive or consequential damages (including, but not limited to, damages for the inability to use equipment or access data, loss of business, loss of profits, business interruption or the like), arising out of the use of, or inability to use, the Software and based on any theory of liability including breach of contract, breach of warranty, tort (including negligence), product liability or otherwise, even if PremiumSoft or its representatives have been advised of the possibility of such damages.

8. Third Party Software

The Software may contain third party software which requires notices and/or additional terms and conditions. By accepting this EULA, you are also accepting the additional terms and conditions of the third party software.

9. General

No PremiumSoft dealer, agent or employee is authorized to make any amendment to this EULA.

This EULA contains the complete agreement between the parties with respect to the subject matter hereof, and supersedes all prior or contemporaneous agreements or understandings, whether oral or written. You agree that any varying or additional terms contained in any purchase order or other written notification or document issued by you in relation to the Software licensed hereunder shall be of no effect. The failure or delay of PremiumSoft to exercise any of its rights under this EULA or upon any breach of this EULA shall not be deemed a waiver of those rights or of the breach.

If any provision of this EULA shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this EULA will remain in full force and effect.

10. Basis of Bargain

The Limited Warranty and Disclaimer and Limited Liability set forth above are fundamental elements of the basis of the agreement between PremiumSoft and you. PremiumSoft would not be able to provide the Software on an economic basis without such limitations. Such Limited Warranty and Disclaimer and Limited Liability inure to the benefit of PremiumSoft's licensors.

11. Term

By downloading and/or installing this SOFTWARE, the Licensor agrees to the terms of this EULA.

This license is effective until terminated. Licensor has the right to terminate your License immediately if you fail to comply with any term of this License.

"as is". Licensor makes no warranties, express or implied, arising from course of dealing or usage of trade, or statutory, as to any matter whatsoever. In particular, any and all warranties or merchantability, fitness for a particular purpose or non-infringement of third party rights are expressly excluded.

12. Governing Law

This License will be governed by the laws in force in Hong Kong. You hereby consent to the non-exclusive jurisdiction and venue sitting in Hong Kong to resolve any disputes arising under this EULA.

Should you have any questions concerning the validity of this License, please contact: licensing@navicat.com. If you desire to contact the Licensor for any other reason, please contact support@navicat.com.

PremiumSoft and other trademarks contained in the Software are trademarks or registered trademarks of PremiumSoft CyberTech Ltd. in the United States and/or other countries. Third party trademarks, trade names, product names and logos may be the trademarks or registered trademarks of their respective owners. You may not remove or alter any trademark, trade names, product names, logo, copyright or other proprietary notices, legends, symbols or labels in the Software. This EULA does not authorize you to use PremiumSoft or its licensors names or any of their respective trademarks.

接続

Navicat でサーバー管理を始めるためには、まず接続ウィンドウを使用して 1 つ以上の接続を作成する必要があります。サーバーまたは'Net in general'の初心者の場合や、物事がどのように動作するのかよくわからない場合には、以下を参考にしてください:

- [MySQL ユーザーマニュアル](#)
- [Oracle データベースドキュメント](#)
- [PostgreSQL ユーザーマニュアル](#)
- [SQLite ユーザーマニュアル](#)
- [MariaDB ドキュメント](#)

新しい接続を作成するためには、 をクリックするか、**ファイル** -> **新しい接続**を選択します。それから、接続設定ウィンドウに必要な情報を入力します。

接続を作成したら、お使いのデータベース/スキーマが接続ペインに表示されます。[オプション](#)ウィンドウで**接続ペインにオブジェクトを表示**オプションにチェックが付けられている場合、すべてのデータベース/スキーマオブジェクトもペインに表示されます。データベース/スキーマに接続するためには、ペインのデータベース/スキーマをダブルクリックします。

注意: Navicat は、様々なプラットフォーム(Windows、Mac、Linux、UNIX)で稼働しているリモートサーバーに接続することを許可しています。

接続を右クリックし、**接続を編集**を選択すると、接続プロパティを編集することができます。

Navicat Cloud

My Connections と [Navicat Cloud](#) の間で接続をコピーまたは移動するためには、接続を右クリックし、**接続をコピー**または**接続を移動**を選択します。

MySQL/MariaDB 接続のフラッシュ

フラッシュには、様々な内部キャッシュのクリアやリロード、テーブルのフラッシュ、ロックの取得といったいろいろな異なる型があります。フラッシュを実行するためには、*Reload* 権限がなければなりません。[MySQL/MariaDB セキュリティ](#)を参照してください。

接続を右クリックし、ポップアップメニューから**フラッシュ**を選択してください。

権限	<i>mysql</i> データベースの権限テーブルから権限をリロードします。
ホスト	ホストキャッシュテーブルを空にします。お使いのホストが IP 番号を変更した場合や <i>Host 'host_name' is blocked</i> エラーメッセージを受け取った場合、host テーブルをフラッシュする方がいいです。MySQL サーバーへの接続中、任意のホストに関して連続して <i>max_connect_errors</i> を超えるエラーが発生したら、MySQL は何か不正が発生したと推測し、それ以上の接続要求からホストをブロックします。host テーブルをフラッシュすることによって、再びそのホストへの接続を試せるようになります。

ログ	すべてのログファイルを閉じてから再び開きます。拡張子なしにアップデートログファイルまたはバイナリログファイルを指定した場合、ログファイルの延長番号は、前のファイルに対して 1 大きくなります。ファイル名に拡張子を使用した場合、MySQL はアップデートログファイルを閉じてから再び開きます。
状態	ほとんどの状態変数をゼロにリセットします。これは、クエリのデバッグ時にのみ使用されるべきものです。
テーブル	すべての開いているテーブルを閉じ、使用中のすべてのテーブルを強制的に閉じます。

テストアカウント

Navicat は、テスト目的のために使えるテストアカウントを提供しています。

リモート MySQL サーバーの接続設定:

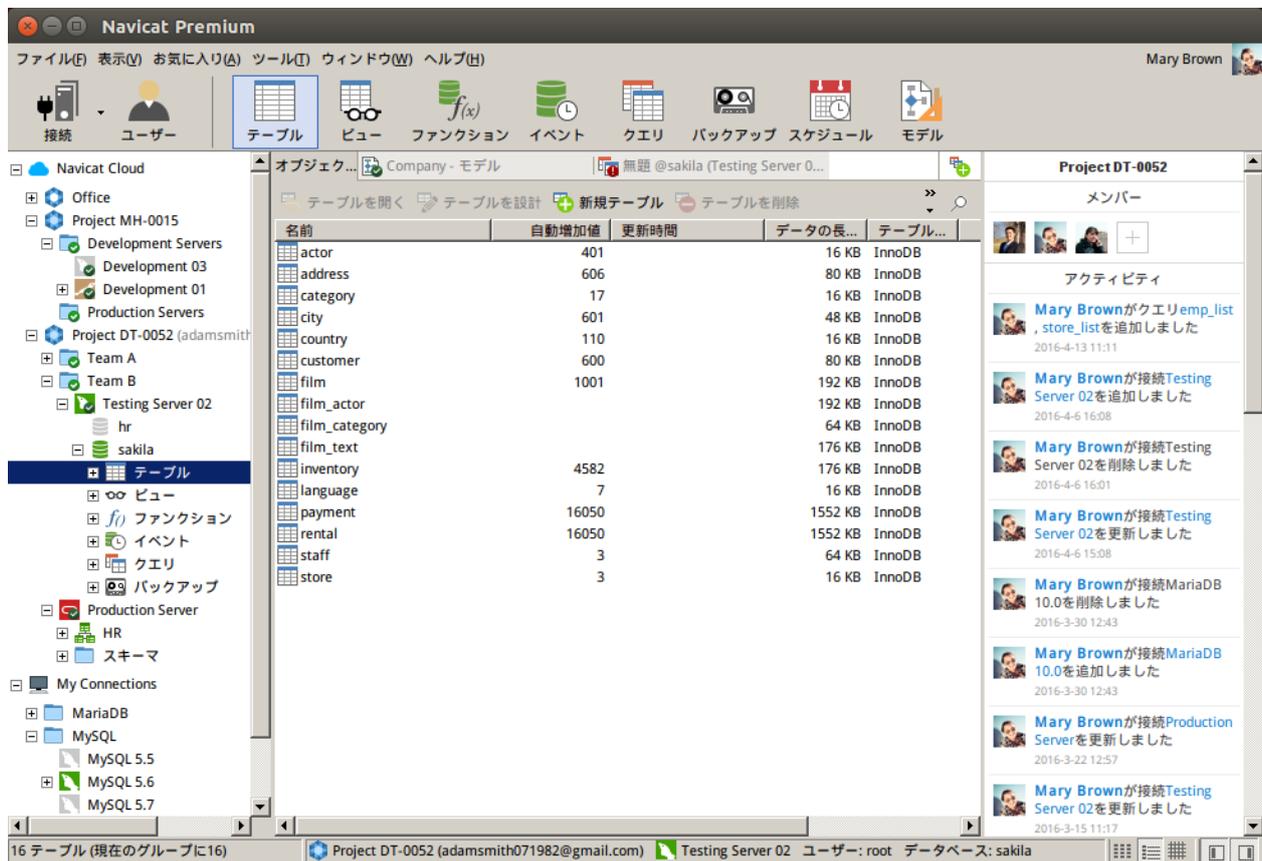
- ホスト名/IP アドレス: server1.navicat.com
- ポート: 4406
- ユーザー名: navicat
- パスワード: testnavicat

リモート PostgreSQL サーバーの接続設定:

- ホスト名/IP アドレス: server1.navicat.com
- ポート: 5432
- 初期データベース: HR
- ユーザー名: navicat
- パスワード: testnavicat

Navicat Cloud

Navicat Cloud は、別のマシンやプラットフォームから Navicat の接続やクエリ、モデル、バーチャルグループを同期するクラウドサービスを提供します。Navicat Cloud への接続を追加すると、その接続設定とクエリは Navicat Cloud に保存されます。モデルファイルを Navicat Cloud に同期することやバーチャルグループを Navicat Cloud に作成することが可能です。全ての Navicat Cloud オブジェクトは、それぞれのプロジェクトの下に置かれます。コラボレーションのために、プロジェクトを他の Navicat Cloud アカウントと共有することができます。



新しいアカウントを作成する

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
2. **Navicat ID の作成**をクリックします。
3. 必要な情報を入力し、**サインアップ**ボタンをクリックします。確認用 E メールがお客様の E メールアドレスに送信されます。
4. その E メール内のリンクをクリックして、新しいアカウントの確認を行います。

ヒント: Navicat カスタマーセンターでお使いの Navicat ID を使用して、サインインすることができます。

Navicat Cloud にサインインする

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
2. **Navicat ID** と**パスワード**を入力します。
3. **サインイン**ボタンをクリックします。
4. [Navicat Cloud Portal](#) サイトでの 2 段階認証を有効にしている場合、コードがお客様の電話にモバイルアプリを通じて送られます。サインインするために、受け取ったコードを入力します。

プロジェクトを作成する

1. **Navicat Cloud** を選択します。
2. それを右クリックし、**新規プロジェクト**を選択します。

プロジェクトにメンバーを追加する

1. プロジェクトを右クリックし、**コラボレーションする**を選択します。
2. **メンバーを追加**をクリックします。
3. メンバーの Navicat ID を入力し、メンバーの役割を選択します。
4. **追加**をクリックします。

メンバーの役割	権限
所有者	オブジェクトの読み込み、オブジェクトの書き込み、メンバーの管理、プロジェクトの削除
Admin	オブジェクトの読み込み、オブジェクトの書き込み、メンバーの管理
メンバー	オブジェクトの読み込み、オブジェクトの書き込み
ゲスト	オブジェクトの読み込み

注意: 一度に 10 人までメンバーを追加することが可能です。カンマを使用するか、編集ボックスに別々にメンバーを入力します。

プロジェクトのメンバーを管理する

1. プロジェクトを右クリックし、**コラボレーションする**を選択します。
2. 変更後、**適用**をクリックします。

注意: 所有者または Admin の場合、**x** ボタンをクリックして、メンバーを削除することができます。

プロジェクトを終了する

1. プロジェクトを右クリックし、**プロジェクトを終了**を選択します。

Navicat Cloud に接続を移動/コピーする

1. **My Connections** の接続を右クリックし、**接続を移動**または**接続をコピー**を選択します。
2. 既存のプロジェクトを選択するか、新しいプロジェクトを作成します。
3. 接続は Navicat Cloud に移動またはコピーされます。そして、そのすべてのクエリファイルは Navicat Cloud に保存されます。

My Connections に接続を移動/コピーする

1. **Navicat Cloud** の接続を右クリックし、**接続を移動**または**接続をコピー** -> **My Connections** を選択します。
2. 接続は My Connections に移動またはコピーされます。

Navicat Cloud にモデルを移動する

1. **My Connections** のモデルファイルを選択します。
2. モデルファイルを **Navicat Cloud** のプロジェクトにドラッグアンドドロップします。

My Connections にモデルを移動する

1. **Navicat Cloud** のプロジェクトのモデルファイルを選択します。
2. モデルファイルを **My Connections** にドラッグアンドドロップします。

Cloud の使用状況を表示

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。

注意: 1つの接続、クエリ、モデル、バーチャルグループは、1ユニットに値します。

アバターを変更する

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
2. 画像をクリックします。
3. 画像ファイルを選択します。

Navicat Cloud アカウントを管理する

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
2. お使いのEメールをクリックし、**アカウント管理**を選択します。
3. ブラウザに [Navicat Cloud Portal](#) サイトが開きます。

Navicat Cloud プランをアップグレード/管理する

1. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
2. **アップグレード**または**プランを管理**をクリックします。
3. ブラウザに [Navicat Cloud Portal](#) サイトが開きます。

Navicat Cloud をサインアウトする

1. **Navicat Cloud** を右クリックし、**すべての接続を終了**を選択して、Navicat Cloud のすべての接続を閉じます。
2. メインメニューから**ファイル** -> **Navicat Cloud** を選択します。
3. **サインアウト**をクリックします。

基本設定

ローカル/リモートサーバーへの新しい接続を確立するためには、SSL、SSH、HTTP を介した接続かどうかにかかわらず、基本タブに接続プロパティを設定してください。お使いのインターネットサービスプロバイダ(ISP)がそのサーバーへの直接接続を提供していない場合は、セキュアトンネリングプロトコル(SSH)/HTTPを使用した接続をお試しください。**接続名**テキストボックスには、その接続をもっともよく表した親しみやすい名前を入力してください。

[Navicat Cloud](#) 機能にログインしたら、**追加先**ドロップダウンリストから、**My Connections** に接続を保存するか、**Navicat Cloud** にプロジェクトを保存するかを選択することができます。Navicat Cloud の接続を編集する時、**ユーザー名**を **Navicat Cloud と同期**オプションを有効にすることによって、ユーザー名を Cloud に同期することを選択できます。

MySQL/MariaDB 接続

お使いの MySQL サーバーにリモートで接続できます。しかしながら、安全面の理由から、MySQL サーバーへの元来のリモートダイレクト接続は不可能です。したがって、[ユーザー権限](#)を設定していない場合、お使いのコンピューターで動いている Navicat Premium または他の同じような MySQL 管理アプリケーションを使用して、リモートサーバーに直接接続することはできません。

デフォルトで、MySQL は、ユーザー名として "root" を提供しており、そのパスワードフィールドは空です。

ホスト名/IP アドレス

データベースが置かれているホスト名、または、サーバーの IP アドレス。

ポート

データベースサーバーに接続するための TCP/IP ポート。

ユーザー名

データベースサーバーに接続するためのユーザー名。

パスワード

サーバーに接続するためのパスワード。

Oracle 接続

Navicat は、Oracle サーバーについて、**Basic** 接続に対応しています。**Basic** モードでは、Navicat は、Oracle Call Interface(OCI) を通して Oracle サーバーに接続します。OCI は、アプリケーションプログラミングインターフェースであり、Oracle データベースサーバーにアクセスし、SQL ステートメント実行の全段階をコントロールするための、第三世代言語の固有のプロシージャもしくはファンクションコールをアプリケーション開発者が利用できるようにします。OCI は、標準的なデータベースのアクセスと検索機能を持つダイナミックリンクライブラリ形式のライブラリです。

デフォルトで、Oracle はインストール時にいくつかのユーザーアカウントを作成します。管理アカウント: SYS、SYSTEM、SYSMAN、DBSNMP。サンプルスキーマアカウント: SCOTT、HR、OE、OC、PM、IX、SH。

接続種類

サーバーへの接続に関する接続の種類: **Basic**。

ホスト名/IP アドレス

データベースが置かれているホスト名、または、サーバーの IP アドレス。

ポート

データベースサーバーに接続するための TCP/IP ポート。

サービス名/SID

接続を行う時にユーザーが接続するサービス名/SID を設定します。該当するラジオボタンを選択してください。

ユーザー名

データベースサーバーに接続するためのユーザー名。

パスワード

サーバーに接続するためのパスワード。

参照:

[OCI オプション](#)

PostgreSQL 接続

安全面の理由から、PostgreSQL サーバーへの元来のリモートダイレクト接続は不可能です。したがって、お使いのコンピュータで動いている Navicat Premium または他の同じような PostgreSQL 管理アプリケーションを使用して、リモートサーバーに接続できない可能性があります。デフォルトで、PostgreSQL は、TCP/IP 接続を使用したローカルマシンからの接続のみを許可しています。他のマシンから接続するためには、`postgresql.conf` ファイルの `listen_addresses` を修正し、`$PGDATA/pg_hba.conf` ファイルを修正することによってホストベースの権限を有効にし、サーバーを再起動しなければなりません。詳しい情報: [Client Authentication](#)

デフォルトで、PostgreSQL はユーザー名として "postgres" を提供しており、そのパスワードフィールドは空です。

ホスト名/IP アドレス

データベースが置かれているホスト名、または、サーバーの IP アドレス。

ポート

データベースサーバーに接続するための TCP/IP ポート。

初期データベース

接続を行う時にユーザーが接続する初期データベース。

ユーザー名

データベースサーバーに接続するためのユーザー名。

パスワード

サーバーに接続するためのパスワード。

SQLite 接続

SQLite データベースの **タイプ** を選択し、ファイルパスを指定することができます。

既存データベースファイル

データベースファイル にある既存データベースに接続します。

新規 SQLite 3

データベースファイル に新しい SQLite 3 データベースを作成します。

新規 SQLite 2

データベースファイル に新しい SQLite 2 データベースを作成します。

データベースファイル

初期データベースファイルを指定します。HTTP トンネルを使用する場合は、ウェブサーバーにあるデータベースファイルの絶対ファイルパスを入力する必要があります。

ユーザー名

既存のデータベースに接続するためのユーザー名。

パスワード

既存のデータベースに接続するためのパスワード。

詳細設定

設定の保存場所

新しい接続が確立されると、Navicat は設定の保存場所の下にサブフォルダを作成します。ほとんどのファイルはこのサブフォルダに保存されます:

Navicat オブジェクト	サーバーの種類	ファイルの拡張子
クエリ	全て	.sql
クエリ結果のエクスポートプロファイル	MySQL	.npeq
	Oracle	.nopeq
	PostgreSQL	.nppeq
	SQLite	.nlpeq
	MariaDB	.nmpeq
ビュー結果のエクスポートプロファイル	MySQL	.npev
	Oracle	.nopev
	PostgreSQL	.nppev
	SQLite	.nlpev
	MariaDB	.nmpev
バックアップ	MySQL, PostgreSQL, SQLite, MariaDB	圧縮 (.psc), 解凍 (.psb)
バックアッププロファイル	MySQL	.npb
	PostgreSQL	.np_pb
	SQLite	.nlpb
	MariaDB	.nmpb
インポートウィザードプロファイル	MySQL	.npi
	Oracle	.nopi
	PostgreSQL	.np_pi
	SQLite	.nlpi
	MariaDB	.nmpi
エクスポートウィザードプロファイル	MySQL	.npe
	Oracle	.nope
	PostgreSQL	.nppe

	SQLite	.nlpe
	MariaDB	.nmpe
マテリアライズド・ビューのエクスポートプロファイル	Oracle	.nopem
	PostgreSQL	.nppem
ER ダイアグラムファイル	全て	.ned
データポンプのエクスポートプロファイル	Oracle	.exp

その他のファイルは、**profiles** ディレクトリに保存されます。そのパスを探すためには、**ツール -> オプション -> その他 -> プロファイルの保存場所** を選択します。

その他のファイル	サーバーの種類	ファイルの拡張子
データ転送	MySQL	.npt
	Oracle	.nopt
	PostgreSQL	.nppt
	SQLite	.nlpt
	MariaDB	.nmppt
	Premium (クロスサーバー)	.napt
データの同期	MySQL	.npd
	Oracle	.nopd
	PostgreSQL	.nppd
	SQLite	.nlpd
	MariaDB	.nmpd
構造の同期	MySQL	.nps
	Oracle	.nops
	PostgreSQL	.npps
	MariaDB	.nmps
バッチジョブ	MySQL	.npj
	Oracle	.nopj
	PostgreSQL	.nppj
	SQLite	.nlpj
	MariaDB	.nmpj
	Premium (クロスサーバー)	.napj
モデルファイル	全て	.ndm
バーチャルグルーピング	全て	vgroup.json - オブジェクトがどのように分類されているかを保存。

参照:

[ログファイル](#)

自動接続する

このオプションを有効にすると、Navicat は、アプリケーション起動時に、接続と登録済みデータベースを自動的に開きます。

MySQL/MariaDB 接続

エンコード

MySQL のキャラクターセットが使用されない場合の MySQL サーバーとの通信用コードページを選択します。

Keepalive 間隔 (秒)

このオプションを使用すると、ping を飛ばし接続確認することによって、サーバーと接続し続けることができます。編集フィールドに、ping を飛ばす間隔を設定することができます。

圧縮を使用

このオプションによって、圧縮プロトコルを使用することができます。クライアントとサーバーの両方が zlib 圧縮をサポートし、クライアントが圧縮を要求する場合に、圧縮プロトコルが使用されます。

Named Pipe、ソケットを使用

このオプションを有効にすると、Navicat は localhost 接続にソケットファイルを使用します。

Oracle 接続

ロール

データベースユーザーが、**Default**、**SYSOPER**、**SYSDBA** のいずれのシステム権限を用いて接続するかを示します。

Keepalive 間隔 (秒)

このオプションを使用すると、ping を飛ばし接続確認することによって、サーバーと接続し続けることができます。編集フィールドに、ping を飛ばす間隔を設定することができます。

OS 認証

このオプションを有効にすると、Oracle データベースはデータベースユーザー認証のために Windows のユーザーログイン認証を利用します。

PostgreSQL 接続

Keepalive 間隔 (秒)

このオプションを使用すると、ping を飛ばし接続確認することによって、サーバーと接続し続けることができます。編集フィールドに、ping を飛ばす間隔を設定することができます。

SQLite 接続

暗号化

暗号化された SQLite データベースに接続する場合、このオプションを有効にし、**パスワード**を提供してください。

追加データベース

接続にデータベースを追加/切り離します。

詳細データベースプロパティ (MySQL/PostgreSQL/MariaDB)

詳細データベースプロパティを設定します。これは必須ではありません。詳細データベースプロパティを使用するためには、**詳細設定を使用する**にチェックをつけます。詳しい説明は以下のとおりです：

ここで選択したデータベースを接続ペインにおいて **close** 状態で表示するためには、データベースリストボックスにある希望のデータベースをクリックしてください。そのチェックボックスは、 のように表示されます。

ここで選択したデータベースを接続ペインにおいて **open** 状態で表示するためには、データベースリストボックスにある希望のデータベースをダブルクリックしてください。そのチェックボックスは、 のように表示されます。

隠れたデータベースを追加するためには

1. **DB をリストに追加**ボタンをクリックしてください。
2. データベース名を入力してください。
3. データベースリストボックスに新しく追加されたデータベースを選択してください。

データベースを削除するためには、データベースリストボックスにあるデータベースを選択し、**DB をリストから削除**ボタンをクリックしてください。

注意： データベースはデータベースリストボックスから削除されるだけです。サーバーには存在したままです。

詳細データベースプロパティ (SQLite)

データベースファイルを追加するためには、**データベースを追加**をクリックしてください。

オプション	説明
データベースファイル	データベース用のファイルパスを設定してください。
データベース名	Navicat に表示するデータベース名を入力してください。
暗号化	暗号化された SQLite データベースに接続する場合、このオプションを有効にし、 パスワード を提供してください。

データベースを切り離すためには、リストからそのデータベースを選択し、**データベースを切り離す**ボタンをクリックしてください。

SSL 設定

セキュア・ソケット・レイヤー (SSL) は、インターネットを介して個人的な文書を送信するためのプロトコルです。安全な接続を確保するために、まず最初に、OpenSSL ライブラリーをインストールし、データベースソースをダウンロードする必要があります。

注意： MySQL/PostgreSQL/MariaDB で使用可能です。
PostgreSQL 8.4 以降でサポートされています。

MySQL/MariaDB 接続

認証詳細情報を提供するために、**認証を使用する**を有効にし、必要な情報を入力してください：

クライアントキー

安全な接続を確立するために使用される PEM 形式の SSL キーファイル。

クライアント証明書

安全な接続を確立するために使用される PEM 形式の SSL 証明書ファイル。

CA 証明書

信頼される SSL 証明書発行元リストを含む PEM 形式のファイルへのパス。

CA 証明書の名前を確認

サーバーがクライアントに送る証明書に含まれるサーバーの Common Name の値を確認します。

暗号鍵

SSL 暗号化に使用するための許容暗号鍵のリスト。

PostgreSQL 接続

SSL モードを選択してください:

require	SSL 接続のみを試みます。
verify-ca	SSL 接続のみを試み、サーバー証明書が信頼される CA から発行されていることを検証します。
verify-full	SSL 接続のみを試み、サーバー証明書が信頼される CA から発行されていることと、そのサーバーのホスト名が証明書内のものと一致することを検証します。

認証詳細情報を提供するために、**認証を使用する**を有効にし、必要な情報を入力してください:

クライアントキー

クライアントプライベートキーのパス。

クライアント証明書

クライアント証明書のパス。

CA 証明書

信頼される証明書発行元のパス。

証明書失効リスト

SSL の証明書失効リスト(CRL:Certificate Revocation List)のファイルパス。

SSH 設定

セキュア・シェル (SSH) は、ネットワーク上での他のコンピュータへのログインや、リモートサーバー上でのコマンドの実行、1つのマシンから別のマシンへのファイルの移動を行うためのプログラムです。SSH は、安全ではないネットワーク上において、厳密な認証を提供し、**SSH ポート転送 (トンネリング)** として知られる、2つのホスト間の安全な暗号化通信を提供しています。一般的に、Telnet の暗号化バージョンとして使用されます。

Telnet のセッション内では、ユーザー名やパスワードを含む全ての通信はテキスト形式で送られるため、誰もがセッションを盗聴したり、パスワードや他の情報を盗むことができます。また、そのようなセッションはセッションハイジャックされやすく、一度認証したセッションが悪意のあるユーザーに支配されてしまいます。SSH はそのような脆弱性を防ぐ働きをします。そして、SSH によって、セキュリティを侵害されることなく、リモートサーバーのシェルにアクセスすることができます。

注意: MySQL/Oracle/PostgreSQL/MariaDB で使用可能です。

Linux サーバーのパラメータ - "AllowTcpForwarding"の値が"yes"に設定されていることを確認してください。そうではない場合、SSH ポート転送が無効になります。そのパスは、`/etc/ssh/sshd_config` です。デフォルトで、SSH ポート転送は有効になっているはずです。設定値を再確認してください。

****** サーバーが SSH トンネルをサポートしている場合でも、ポート転送が無効になっている場合、Navicat は、SSH ポート 22 を介して接続することはできません。

ホスト名/IP アドレス

SSH サーバーが起動してるホスト。

ポート

SSH サーバーが起動しているポート。デフォルトは 22 です。

ユーザー名

SSH サーバーマシンのユーザー。(データベースサーバーのユーザーではありません。)

認証方式

パスワード	SSH サーバーユーザーのパスワードを提供してください。
公開鍵	プライベートキー 公開鍵とともに使用されます。プライベートキーはユーザーのみが読めるはずです。 パスフレーズ パスフレーズは、ユーザーが作成しているキーに適用されることと、アカウントではないこと以外は、パスワードと同じようなものです。

注意: 基本タブの Navicat ホスト名は、お使いのデータベースホスティング会社によって提供される SSH サーバーに対して相対的に設定されるべきです。

HTTP 設定

HTTP トンネリングは、ウェブサーバーが使用しているものと同じプロトコル (`http://`) と同じポート (ポート 80) を使用して、サーバーに接続する方法です。お使いの ISP が直接接続を許可せず、HTTP 接続の確立を許可している場合に、この方法が使用されます。

注意: MySQL/PostgreSQL/SQLite/MariaDB で使用可能です。

トンネリングスクリプトのアップロード

この接続方法を使用するために、まず最初に、お使いのサーバーが配置されているウェブサーバーにトンネリングスクリプトをアップロードする必要があります。

注意: `ntunnel_mysql.php` (MySQL と MariaDB 用)、`ntunnel_pgsql.php`、`ntunnel_sqlite.php` は、Navicat のインストールフォルダにあります。

HTTP トンネルの設定

HTTP 接続を設定する方法について説明します。

1. HTTP タブを選択し、**HTTP トンネルを使用する**を有効にします。
2. トンネリングスクリプトの URL を入力します。例えば、`http://www.navicat.com/ntunnel_mysql.php`。
3. お使いのサーバーにウェブアプリケーションファイアウォールがインストールされている場合、**送信クエリを base64 エンコード**オプションにチェックをつけることができます。
4. トンネルスクリプトがパスワードで保護されたサーバーに置かれている場合、または、プロキシサーバーを介してインターネットにアクセスしなければならない場合は、**認証**または**プロキシ**タブに必要な認証詳細情報を入力します。
5. 基本設定ページにある Navicat ホスト名は、データベースホスティング会社から提供された HTTP サーバーに対して相対的に設定されなければなりません。

注意: HTTP トンネルと SSH トンネルは、同時に使用できません。HTTP トンネルを選択すると、SSH トンネルは無効となります。逆も同じです。

サーバーオブジェクト

Navicat は、データベースやテーブル、ビュー、ファンクションなどのサーバーオブジェクトを管理するための高性能なツールを提供しています。

注意: Navicat でサーバーオブジェクトの操作を開始する前に、まず接続を確立しなければなりません。

MySQL/MariaDB オブジェクト

サーバーオブジェクトの操作を開始する前に、接続を作成し、接続を開かなければなりません。サーバーが空の場合、接続ペインでその接続を右クリックし、**新規データベース**を選択して、新しいデータベースを作成する必要があります。

既存データベースのプロパティを編集するためには、接続ペインのデータベースを右クリックし、**データベースを編集**を選択してください。現在のところ、MySQL はそのインターフェースからのデータベースのリネームに対応していませんので、ご注意ください。データベースが保存されているディレクトリにアクセスしてください。デフォルトで、すべてのデータベースは MySQL インストールフォルダの下にある **data** と呼ばれるディレクトリに保存されます。例: `/var/lib/mysql/`。

注意: データベースをリネームする前に MySQL を停止しなければなりません。

データベース名

新しいデータベースの名前を設定します。

キャラクターセット

データベースのデフォルトキャラクターセットを指定します。

照合

データベースのデフォルト照合を指定します。

MySQL /MariaDB テーブル

リレーショナルデータベースは、データを保存するためにテーブルを使用します。全てのデータ操作は、テーブル自身に対して行われるか、結果として別のテーブルを生成します。テーブルは行と列のセットで、その交わる部分がフィールドです。一般的な観点から、テーブル内の列は、その列のフィールドに関して行によって検出されるデータの名前やタイプを表します。テーブル内の行は、対応する列の名前とタイプによって、左から右へ記述されるフィールドから成るレコードを表します。行の各フィールドは、その行の他の各フィールドと暗黙的に関係しています。  をクリックして、**テーブル用オブジェクトリスト**を開きます。

グラフィカルフィールドを含むテーブルを開く場合、テーブルを右クリックし、ポップアップメニューから  **テーブルを開く (高速)** を選択します。セルがクリックされるまで BLOB フィールド(イメージ)はロードされませんので、より高速なパフォーマンスでグラフィカルテーブルを開くことができます。Navicat でテーブルを開く時にすべてのイメージをロードしたい場合、  **テーブルを開く** を使用してください。

テーブルを空にするためには、選択したテーブルを右クリックし、ポップアップメニューから**テーブルをクリア**を選択します。このオプションは、自動増加値はリセットせずに既存のすべてのレコードをクリアしたい場合にのみ適用されます。テーブルを空にする時に自動増加値をリセットするためには、**テーブルをトランケート**を使用してください。

MySQL/MariaDB テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成、挿入や、選択したフィールドの移動、削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+Fを押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。 注意: MySQL 3.22 以降でサポートされています。
 フィールドを削除	選択したフィールドを削除します。
 プライマリーキー	フィールドにプライマリーキーを設定します。
 上に移動/ 下に移動	選択したフィールドを上/下に移動します。 注意: MySQL 4.0.1 以降でサポートされています。

既存のフィールドの1つに変更を加えた新規フィールドを追加するためには、右クリックしてポップアップメニューから**フィールドを複製**を選択します。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならぬことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプを定義します。詳細については、[MySQL データタイプ](#) / [MariaDB データタイプ](#)を参照してください。

長さ編集ボックスを使用してフィールドの長さを定義します。**小数**編集ボックスを使用して浮動小数点データタイプの小数点後の桁数(スケール)を定義します。

注意: フィールドの長さを短くする時には、データが失われる可能性があることに注意してください。

ヌルでない

このボックスにチェックをつけると、フィールドに null 値が許可されません。

バーチャル

このボックスにチェックをつけると、フィールドに対してバーチャルカラム設定を有効にします。

注意: MariaDB 5.2 以降でサポートされています。

プライマリーキー

プライマリーキーは、レコードを一意に定義するための単一フィールドまたは複数のフィールドの組み合わせです。プライマリーキーの一部であるフィールドはいずれも、null 値を含むことはできません。

フィールドのプロパティ

注意: 以下のオプションは選択されたフィールドによって異なります。

デフォルト

フィールドのデフォルト値を設定します。

注釈行

現在のフィールドを説明する任意の文字列を設定します。

キャラクターセット

キャラクターセットは、符号とエンコーディングのセットです。

照合

照合は、キャラクターセットの文字を比較するためのルール式です。

注意: MySQL は以下の方法で列のキャラクターセットと照合を選択します:

- CHARACTER SET X と COLLATE Y の両方が指定された場合、その後キャラクターセット X と照合 Y が使用されます。
- COLLATE は指定されず、CHARACTER SET X が指定された場合、その後キャラクターセット X とそのデフォルトの照合が使用されます。
- それ以外の場合、テーブルのキャラクターセットと照合が使用されます。

キー長

プライマリーキーが設定されると、編集ボックスが有効になります。

バイナリ

MySQL 4.1 以降では、CHAR と VARCHAR フィールドの値は、そのフィールドに割り当てられたキャラクターセットの照合に従って、ソート/比較されます。

MySQL 4.1 より前では、ソートと比較はサーバーのキャラクターセットの照合が基本となります; バイナリ属性を持つフィールドを宣言すると、フィールド値の中のバイト数値を基本とするソートや比較を行うことができます。バイナリは、フィールド値の保存や検索方法に影響しません。

自動増加

自動増加属性は、新しい行に一意的な同一性を生成するために使用されます。自動増加値を 1 以外から始めるためには、オプションタブでその値を設定することができます。

符号無し

符号無しの値は、フィールドに負ではない数だけを許可したい場合や、フィールドの数値幅の上限をより大きくする必要がある時に使用されます。

MySQL 4.0.2 以降では、浮動小数点や固定小数点のタイプでも符号無しにすることが可能です。整数タイプとは違い、列の値の上限範囲に変更はありません。

ゼロ詰め

スペースのデフォルトパディングはゼロで置き換えられます。例えば、フィールドが INT(5) ZEROFILL として宣言された場合、4 の値は 00004 として取り出されます; フィールドが FLOAT(20,10) ZEROFILL として宣言された場合、0.1 の値は、000000000.1000000015 として取り出されます。

注意: 数値タイプにゼロ詰めを指定する場合、MySQL は自動的にフィールドに符号無し属性を追加します。

On Update Current_Timestamp

4.1.2 以降では、自動的に初期化され、現在の timestamp に更新される TIMESTAMP フィールドをより柔軟に決定できます。

可能な値

この編集ボックスを使用して、SET/ENUM の要素を定義します。

バーチャルタイプ

バーチャルカラムのタイプを選択します。**VIRTUAL** は、カラムはコマンドがそのカラムを指定する時に計算されることを意味します。**PERSISTENT** は、その値がテーブルに物理的に保存されることを意味します。

注意: MariaDB 5.2 以降でサポートされています。

式

書き込み時に評価する式またはファンクションを指定します。

注意: MariaDB 5.2 以降でサポートされています。

MySQL/MariaDB テーブルのインデックス

インデックスは、テーブルにある特定の列の組織化されたバージョンです。MySQL は、レコードの高速検索を容易にするためにインデックスを使用します。インデックスを使用すると、MySQL は直接希望のレコードにジャンプすることができます。インデックスを使用しない場合、MySQL は正確なレコードを見つけるためにデータファイル全体を読まなければなりません。

インデックスタブにおいて、編集を行うためにはインデックスフィールドをクリックします。インデックスツールバーを使用して、インデックスフィールドの新規作成や選択したインデックスフィールドの編集や削除を行うことができます。

ボタン	説明
 インデックスを追加	テーブルにインデックスを追加します。
 インデックスを削除	選択したインデックスを削除します。

名前編集ボックスを使用してインデックス名を設定します。

インデックスにフィールドを追加するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。リストからフィールドを選択します。インデックスからフィールドを除くためには、同様の方法でチェックを外します。また、インデックスのフィールド順を変更するためには、矢印ボタンを使用します。**サブパート**編集ボックスは、インデックスのキー長を設定するために使用されます。

注意: いくつかのデータタイプは、複数のフィールドによるインデックス化を許可しません。例: BLOB

インデックスタイプ

テーブルインデックスのタイプを定義します。

Normal	NORMAL インデックスは、最も基本的なインデックスです。一意性のような制約は全くありません。
Unique	UNIQUE インデックスは、1 つの違い以外は NORMAL インデックスと同じです。その違いは、インデックス化された列の全ての値は一度だけ発生できるということです。
Full Text	FULL TEXT インデックスは、MySQL が全文検索に使用します。

インデックスメソッド

インデックスの作成時にインデックスタイプを指定します: BTREE または HASH。

注釈行

現在のインデックスを説明する任意の文字列を設定します。

注意: MySQL 5.5.3 以降でサポートされています。

MySQL/MariaDB テーブルの外部キー

外部キーは、別のテーブルのプライマリーキー列と一致する、リレーショナルテーブルにあるフィールドです。外部キーは、テーブルの相互参照に使用することができます。

外部キータブにおいて、編集を行うためには外部キーフィールドをクリックします。外部キーツールバーを使用して、外部キーフィールドの新規作成や選択した外部キーフィールドの編集や削除を行うことができます。

注意: 外部キーは、MySQL 3.23.44 以降でサポートされています。

外部キーの編集は、MySQL 4.0.13 以降でサポートされています。

ボタン	説明
 外部キーを追加	テーブルに外部キーを追加します。 注意: 両方のテーブルが <i>InnoDB</i> タイプでなければなりません (solidDB for MySQL をお持ちの場合、 <i>solidDB</i> タイプ)。参照テーブルには、外部キー列が同じ順番で最初の列として並べられているインデックスがなければなりません。MySQL 4.1.2 以降では、そのようなインデックスが参照テーブルに存在しない場合、自動的に作成されます。
 外部キーを削除	選択した外部キーを削除します。 注意: MySQL 4.0.13 以降でサポートされています。

名前編集ボックスに新規キーの名前を入力します。

被参照データベースと被参照テーブルドロップダウンリストを使用して、外部のデータベースとテーブルをそれぞれ選択します。

キーにフィールド/被参照フィールドを追加するためには、**フィールド/被参照フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

削除時と**更新時**ドロップダウンリストは、行われるアクションの種類を定義します。

CASCADE	対応する外部キーを削除するか、対応する外部キーをプライマリーキーの新しい値に更新します。
SET NULL	対応する外部キーの列全てを NULL に設定します。
NO ACTION	外部キーを変更しません。
RESTRICT	削除や更新が外部キーの制約違反を引き起こすことを知らせるエラーを出します。

関連トピック:

[外部キーのデータ選択](#)

MySQL/MariaDB テーブルのトリガー

トリガーは、テーブルに関連付けられた名前の付いたデータベースオブジェクトで、テーブルに対して特定のイベントが起きた時に起動されます。

トリガータブにおいて、編集を行うためにはトリガーフィールドをクリックします。トリガーツールバーを使用して、トリガーフィールドの新規作成や選択したトリガーフィールドの編集や削除を行うことができます。

注意: トリガーは、MySQL 5.0.2 以降でサポートされています。

ボタン	説明
 トリガーの追加	テーブルにトリガーを追加します。
 トリガーの削除	選択したトリガーを削除します。

名前編集ボックスを使用してトリガー名を設定します。

起動ドロップダウンリストを使用してトリガーのアクションのタイミングを定義します。**Before** または **After** のいずれかとなり、トリガーを起動するステートメントの前または後のどちらにトリガーを起動するかを示します。

挿入

テーブルに新しい行が挿入される時はいつも、トリガーが起動されます。例えば、**INSERT**、**LOAD DATA** や **REPLACE** 文です。

更新

行が修正される時はいつも、トリガーが起動されます。例えば、**UPDATE** 文です。

削除

テーブルから行が削除される時はいつも、トリガーが起動されます。例えば、**DELETE** や **REPLACE** 文です。しかし、テーブルに対する **DROP TABLE** や **TRUNCATE** 文はトリガーを起動しません。

定義タブには、トリガーが起動する時に実行されるステートメントを定義します。ステートメントを追加するためには、編集ボックスをクリックして入力します。複数のステートメントを実行したい場合、**BEGIN ... END** 複合文定義を使用します。例:

```
BEGIN
  set new.capacity = new.capacity + 100;
  set new.amount = new.amount + 100;
END
```

MySQL/MariaDB テーブルのオプション

エンジン

テーブルのエンジンを定義します。

キャラクターセット

テーブルのキャラクターセットの種類を定義します。

照合

テーブルの照合順序を選択します。

自動増加

編集フィールドで**自動増加**の値を設定/リセットします。自動増加値は次のレコードに対する値を表します。

チェックサム/テーブルのチェックサム (MyISAM と Aria テーブルのみ)

MySQL が全ての行に対してライブチェックサムを維持するようにしたい場合、このオプションにチェックをつけます。

注意: Aria テーブルは、MariaDB 5.1 以降でサポートされています。

ページのチェックサム (Aria テーブルのみ)

更なる安全性のために、インデックスとデータにページのチェックサムを使用したい場合、このオプションにチェックをつけます。

注意: MariaDB 5.1 以降でサポートされています。

トランザクション (Aria テーブルのみ)

クラッシュセーフにしたい場合、このオプションにチェックをつけます。

注意: MariaDB 5.1 以降でサポートされています。

行のフォーマット

行がどのように保存されるべきかを定義します。

平均行長

テーブルの平均の行の長さの近似値。様々なサイズの行を持つ大容量のテーブルに対してのみ、この設定が必要です。

最大行数

テーブルに保存予定の最大行数。これは、絶対的な制限ではなく、テーブルは少なくともこれくらい多くの行を保存可能でなければならないということを、ストレージエンジンに対して示唆するためのものです。

最小行数

テーブルに保存予定の最小行数。

キーブロックサイズ

このオプションは、インデックスキーブロックに使用するバイトサイズについて、ストレージエンジンに対して示唆します。エンジンは、必要に応じて、値を変更することができます。0 は、デフォルト値を使用することを表します。

キーを圧縮 (MyISAM テーブルでのみ有効)

より小さいインデックスにしたい場合、このオプションを 1 に設定します。これで、通常、更新処理は遅く、読み込み処理は速くなります。このオプションを 0 に設定すると、キーの全ての圧縮が無効になります。**DEFAULT** に設定すると、長い *CHAR*、*VARCHAR*、*BINARY*、*VARBINARY* 列のみを圧縮するように、ストレージエンジンに命令します。

遅延キー書込 (MyISAM テーブルのみ)

テーブルが閉じられるまでテーブルへのキーの更新を遅延させたい場合、このオプションにチェックをつけます。

データディレクトリ

MyISAM ストレージエンジンがテーブルのデータファイルを置くべき場所を指定します。

インデックスディレクトリ

MyISAM ストレージエンジンがテーブルのインデックスファイルを置くべき場所を指定します。

Union (MRG_MYISAM テーブルのみ)

UNION は、まるで 1 つのテーブルにアクセスするように同一の *MyISAM* テーブルの集合にアクセスしたい場合に使用されます。これは、*MERGE* テーブルでのみ機能します。*MERGE* テーブルにマップするテーブルに対する *SELECT*、*UPDATE*、*DELETE* 権限を持っていない限りなりません。

挿入メソッド (MRG_MYISAM テーブルのみ)

MERGE テーブルにデータを挿入したい場合、行が挿入されるべきテーブルを *INSERT_METHOD* を使用して指定する必要があります。*INSERT_METHOD* は、*MERGE* テーブルに対してのみ有用なオプションです。最初または最後のテーブルに挿入処理を行う場合は、**FIRST** または **LAST** の値を使用します。もしくは、**NO** の値を使って、挿入できないようにします。

接続 (FEDERATED テーブルのみ)

リモートテーブルに連合されるローカルテーブルを作成します。ローカルテーブルを作成し、リモートテーブルへの接続に使用する接続文字列(サーバー名、ログイン、パスワードを含む)を接続編集ボックスに指定することができます。

接続文字列には、データを物理的に保存するために使用されるテーブルを含む、リモートサーバーへの接続に必要な情報が含まれます。接続文字列には、サーバー名やログイン証明、ポート番号、データベース/テーブル情報を指定します。

接続文字列の形式は以下のとおりです:

```
scheme://user_name[:password]@host_name[:port_num]/db_name/tbl_name
```

接続文字列のサンプル:

```
CONNECTION='mysql://username:password@hostname:port/database/tablename'
```

```
CONNECTION='mysql://username@hostname/database/tablename'
```

```
CONNECTION='mysql://username:password@hostname/database/tablename'
```

テーブルスペース (*ndbcluster* テーブルのみ)

ストレージ用のテーブルスペースを指定します。

注意: MySQL 5.1.6 以降でサポートされています。

ストレージ (*ndbcluster* テーブルのみ)

使用されるストレージのタイプ (ディスクまたはメモリ) を指定します。 **DISK**、**MEMORY**、**DEFAULT** のいずれかとなります。

注意: MySQL 5.1.6 以降でサポートされています。

Stats Auto Recalc

InnoDB テーブルに関して、持続的な統計情報を自動的に再計算するかどうかを指定します。

注意: MariaDB 10.0 以降でサポートされています。

Stats Persistent

ANALYZE TABLE によって作成された InnoDB の統計情報がディスクに残るかどうかを指定します。

注意: MariaDB 10.0 以降でサポートされています。

パーティション

パーティションオプションを設定します。

注意: MySQL 5.1 以降でサポートされています。

オプション	説明
パーティション決定関数	パーティションの決定に使用される関数を選択します: HASH 、 KEY 、 LINEAR HASH 、 LINEAR KEY 、 RANGE 、 LIST 。
パーティション	パーティション番号を設定します。
サブパーティション決定関数	サブパーティションの決定に使用される関数を選択します: Hash 、 Key 。
サブパーティション	サブパーティション番号を設定します。
パーティション定義	+ または - パーティション を使用して、パーティションを追加または削除することができます。 + または - サブパーティション を使用して、サブパーティションを追加または削除することができます。
可能な値	レンジ分割の場合は、各パーティションに VALUES LESS THAN 句を追加しなければなりません; リスト分割の場合は、各パーティションに対して VALUES IN 句を指定しなければなりません。これはどの行がこのパーティションに保存されるのかを決定するために使用されます。

エンジン	パーティションとサブパーティションの両方のためのストレージエンジンを選択します。
データディレクトリ	このパーティションに関するデータが保存されているディレクトリ。
インデックスディレクトリ	このパーティションに関するインデックスが保存されているディレクトリ。
最大行数	パーティションに保存される最大行数。
最小行数	パーティションに保存される最小行数。
テーブルスペース	パーティションのテーブルスペースを指定します。Falcon に対してのみ使用されます。
ノードグループ	ノードグループを設定します。
注釈行	パーティションに関する注釈を入力します。

MySQL/MariaDB ビュー

ビュー(更新可能なビューを含む)は、MySQL Server 5.0 で実装され、5.0.1 以降はバイナリリリースで使用可能です。ユーザーが単一テーブルにアクセスしているように複数の関連テーブルにアクセスできるようにしたり、そのアクセスを制限したりするために、ビューが役に立ちます。ビューは、行(特定テーブルの一部)へのアクセスを制限するために使用されることもあります。列へのアクセス制限のために、MySQL サーバーで最新の権限システムを使用することもできます。 をクリックして、**ビュー用のオブジェクトリスト**を開きます。

ボタン	説明
 プレビュー	ビューの結果をプレビューします。
 説明	ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: **表示** -> **結果を表示** -> **クエリエディタの下** または **新規タブ** を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQLの知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

詳細プロパティ

アルゴリズム

Undefined	どのアルゴリズムを使用するか、MySQL が選択します。通常 Merge の方が効率的であるため、そして、一時テーブルの使用時にビューは更新不可であるため、MySQL はできるだけ Temptable よりも Merge の方を選びます。
Merge	ビューを参照する文のテキストとビューの定義がマージされ、ビューの定義の一部は文の対応部分に置き換わります。
Temptable	ビューからの結果は一時テーブルに取り込まれ、その後、文の実行に使用されます。

定義者

デフォルトの定義者の値は、`CREATE VIEW` 文を実行するユーザーです。(これは、`DEFINER = CURRENT_USER` と同じです。) ユーザーの値が与えられている場合、`'user_name'@'host_name'` 書式 (`GRANT` 文で使用されるのと同じ書式)の MySQL アカウントであるべきです。user_name と host_name の両方の値が必要とされます。

セキュリティ

SQL セキュリティの特性は、ビューの実行時にビューに関するアクセス権を確認する場合に使用するための MySQL アカウントを決定します。正当な特性の値は **Definer** と **Invoker** です。これらの値はそれぞれ、ビューを定義したユーザーまたは呼び出したユーザーによってビューが実行可能でなければならないことを表します。デフォルトのセキュリティの値は Definer です。

チェックオプション

Local	定義されているビューだけにチェックオプションを制限します。
Cascaded	評価される実際のビューに対しても同様にチェックを行います。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

注意: トランザクションは、更新可能なビューに対して使用可能です。

MySQL/MariaDB ファンクション/プロシージャ

ストアド・ルーチン(プロシージャとファンクション)は、MySQL 5.0 でサポートされています。ストアド・ルーチンは、サーバーに保存されることが可能な SQL 文の集まりです。いったんこれが実行されると、クライアントはそれぞれの文を再発行し続ける必要はなく、代わりにストアド・ルーチンに問い合わせることができます。 をクリックして、**ファンクション**用のオブジェクトリストを開きます。

ファンクションウィザード

オブジェクトリストツールバーから、 **新規ファンクション**をクリックします。**ファンクションウィザード**がポップアップ画面で開きます。このウィザードを使用して、簡単にプロシージャ/ファンクションを作成することができます。新規プロシージャ/ファンクションを作成時に**ファンクションウィザード**を表示しないようにすることができます。

1. ルーチンの種類を選択します：**プロシージャ** または **ファンクション**。
2. パラメータを定義します。該当する列に、**モード**、**名前**、**タイプ**パラメータを設定します。

3. ファンクションを作成する場合、リストから**リターンタイプ**を選択し、該当する情報を入力します：**長さ、小数、キャラクターセット、Enum**。

ヒント: **次回ウィザードを表示**のチェックを外した場合、[オプション](#)で再度有効にすることができます。

定義

定義は、有効な SQL プロシージャ文から成ります。これは、*SELECT* または *INSERT* のような単純な文、または、*BEGIN* と *END* を使用して書かれた複合文の場合があります。複合文は、宣言、ループ、その他の制御定義文を含むことができます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

パラメータ

ファンクション/プロシージャのパラメータを定義します。

リターンタイプ

このテキストボックスは、ファンクションの作成時のみ有効となります。ファンクションのリターンタイプを示します。

タイプ

ドロップダウンリストから、作成したいストアドルーチン、つまり **PROCEDURE** と **FUNCTION** のいずれかを選択します。

詳細プロパティ

セキュリティ

ルーチンがそのルーチンを作成するユーザーの権限またはルーチンを呼び出すユーザーの権限のどちらを使用して実行されるべきかを指定します。

定義者

定義者のデフォルト値は、*CREATE PROCEDURE* または *CREATE FUNCTION* 文を実行するユーザーです。(これは、*DEFINER = CURRENT_USER* と同じです。) ユーザーの値が与えられる場合、'user_name'@'host_name'形式の MySQL アカウントでなければなりません (*GRANT* 文で使用されるのと同じ形式)。user_name と host_name の値は両方とも必須です。

データアクセス

様々な特性がルーチンによるデータ使用の本質に関する情報を提供します。

Contains SQL	ルーチンが、データの読み込みや書き込みを行う文を含まないことを表します。これらの特性のどれも明確に設定されていない場合、これがデフォルトとなります。
No SQL	ルーチンが、SQL 文を全く含まないことを表します。
Reads SQL Data	ルーチンが、データを読み込む文を含むが、データを書き込む文を含まないことを表します。
Modifies SQL Data	ルーチンが、データを書き込む可能性のある文を含むことを表します。

決定的

プロシージャ及びファンクションは、同じ入力パラメータに対して常に同じ結果をもたらす場合に、決定的と見なされます。そうでない場合は、決定的ではありません。

結果

プロシージャ/ファンクションを実行するために、ツールバーの  **実行** をクリックします。SQL 文が正しい場合、その文が実行されます。そして、その文がデータを返すことになっている場合、プロシージャ/ファンクションによって返されたデータを表示した **結果** タブが開きます。プロシージャ/ファンクションの実行時にエラーが発生した場合、実行は停止し、適切なエラーメッセージが表示されます。ファンクション/プロシージャがパラメータの入力を必要とする場合、**パラメータ入力** ボックスがポップアップ表示されます。パラメータを区切る場合は、',' を使用します。

ヒント: Navicat は、10 個の結果セットの返信に対応しています。

MySQL/MariaDB イベント

MySQL イベントスケジューラーは、MySQL 5.1.6 で追加されました。MySQL のイベントは、スケジュールに従って実行される処理です。そのため、私達は時々、それらをスケジュールされたイベントとして参照します。イベントを作成する時、指定した日や時間に始まって終わり 1 回以上定期的に実行される SQL 文を 1 つ以上含んだ、名前の付いたデータベースオブジェクトを作成します。概念的に、これは Windows タスクスケジューラーの考え方と似ています。  をクリックして、**イベント** 用のオブジェクトリストを開きます。

定義

定義は、有効な SQL 文から成ります。これは、*SELECT* または *INSERT* のような単純な文、または、*BEGIN* と *END* を使用して書かれた複合文の場合があります。複合文は、宣言、ループ、その他の制御定義文を含むことができます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#) をご覧ください。

定義者

イベント実行時、アクセス権限を確認する時に使用されるユーザーアカウントを指定します。定義者のデフォルト値は、*CREATE EVENT* 文を実行するユーザーです。（これは、*DEFINER = CURRENT_USER* と同じです。）ユーザーの値が与えられている場合、'user_name'@'host_name'形式の MySQL アカウントでなければなりません（*GRANT* 文で使用されるのと同じ形式）。user_name と host_name の値は両方とも必須です。

状態

イベントを作成し、そのイベントを *DISABLE* キーワードを使用して有効でないようにすることができます。また、*ENABLE* を使用して、デフォルトの状態、つまり有効な状態であることを明示することができます。

ON COMPLETION

通常、いったんイベントが終了すると、そのイベントはすぐに削除されます。*ON COMPLETION PRESERVE* を指定することによって、この動作を無視することができます。*ON COMPLETION NOT PRESERVE* を使用して、デフォルトの非永続的動作を単に明確化します。

スケジュール

AT

AT タイムスタンプは、1 回限りのイベントに対して使用されます。イベントがタイムスタンプとして与えられた日付と時間に 1 度だけ実行することを指定します。タイムスタンプは、日付と時間の両方を含んでいるか、datetime 値に解決できる表現でなければなりません。現在の日付と時間に対して未来のある時点で起きるイベントを作成するためには、**+INTERVAL** を使用します。

EVERY

定期的に繰り返される動作のために、EVERY 句を使用することができます。EVERY 句には、後に間隔が続きます。(**+INTERVAL** は、EVERY とともに使用されません。)

STARTS

EVERY 句は、任意の STARTS 句を含むこともできます。STARTS の後には、その動作が繰り返し開始するべき時を表すタイムスタンプ値が続きます。そして、**+INTERVAL** 間隔を使用して、"今から" の時間の長さを指定することもできます。

例: **EVERY 3 MONTH STARTS CURRENT_TIMESTAMP + 1 WEEK** は、"今から 1 週間後に開始して、3 ヶ月ごとに"を意味します。

ENDS

EVERY 句は、任意の ENDS 句を含むこともできます。ENDS キーワードの後には、イベントが繰り返しを止めるべき時を MySQL に伝えるタイムスタンプ値が続きます。ENDS と共に、**+INTERVAL** 間隔を使用することもできます。

例: **EVERY 12 HOUR STARTS CURRENT_TIMESTAMP + INTERVAL 30 MINUTE ENDS CURRENT_TIMESTAMP + INTERVAL 4 WEEK** は、"今から 30 分後に開始して、12 時間ごとに、そして今から 4 週間で終了"を意味します。

P.S. タイムスタンプは未来でなければなりません - 過去に起きるイベントをスケジュールすることはできません。

間隔は、量と*時間の単位の 2 つの部分から成ります。

*YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE |

WEEK | SECOND | YEAR_MONTH | DAY_HOUR | DAY_MINUTE |

DAY_SECOND | HOUR_MINUTE | HOUR_SECOND | MINUTE_SECOND

Oracle オブジェクト

サーバーオブジェクトの操作を開始する前に、接続を作成し、接続を開かなければなりません。ユーザーアカウントを作成する時、そのユーザーに対してスキーマも暗黙的に作成しています。スキーマは、そのユーザーが作成するデータベースオブジェクト(テーブル、ビュー、トリガーなど)のための論理的な入れ物です。スキーマ名はユーザー名と同じで、そのユーザーによって所有されるオブジェクトを明確に参照するために使用されます。他のユーザースキーマは  **スキーマ** に表示されます。

ヒント: Oracle は、引用符で囲まれていないオブジェクト識別子を大文字として解釈します。Navicat では、全てのオブジェクト識別子が引用符で囲まれます。つまり、Navicat は、入力したものを正確に保存します。

Oracle データポンプ (フルバージョンで使用可能)

Oracle データポンプ技術は、ひとつのデータベースから別のデータベースにデータやメタデータを非常に高速で移動させることを可能にします。2つのユーティリティを含みます：データポンプエクスポートとデータポンプインポート。

データポンプエクスポートは、ダンプファイルセットと呼ばれるオペレーティングシステムファイルにデータやメタデータをアンロードするためのユーティリティです。ダンプファイルセットは、データポンプインポートユーティリティによってのみインポートされることができます。ダンプファイルセットは、同じシステムにインポートされることが可能です。または、別のシステムに移動され、そこでロードされることが可能です。

データポンプインポートは、エクスポートダンプファイルセットをターゲットシステムにロードするためのユーティリティです。ダンプファイルセットは、テーブルデータやデータベースオブジェクトメタデータ、コントロール情報を含む1つ以上のディスクファイルで構成されています。ファイルは、所有者のバイナリ形式で書かれています。データポンプインポートユーティリティは、インポート操作中、ダンプファイルセットにある各データベースオブジェクトを配置するためにこれらのファイルを使用します。

 をクリックして、**データポンプ**用のオブジェクトリストを開きます。

オブジェクトリストペインのどこかで右クリックしポップアップメニューから**ディレクトリを変更**を選択することによって、ダンプファイルセットの**ディレクトリ**を変更することができます。

注意: Oracle 10g 以降でサポートされています。

データポンプは SYSDBA ロールを必要とします。ダンプファイルセットはサーバーに保存されます。

Oracle データポンプエクスポート

データポンプエクスポートを実行する前に、**SQL を生成**ボタンをクリックして SQL を見直します。それから、**実行**ボタンをクリックして実行することができます。

データポンプエクスポートの設定をプロファイルに保存することができます。**保存**ボタンをクリックしてください。

ヒント: プロファイル(.exp)は**設定の保存場所**に保存されます。

隠れているタブ(詳細オプション)を表示するためには、**詳細オプションを表示**ボックスにチェックをつけます。

基本プロパティ

ジョブ名

ジョブの名前。

モード

FULL	フルデータベースエクスポートでは、データベース全体がアンロードされます。このモードを使うためには、EXP_FULL_DATABASE ロールを持っていないければなりません。
------	----------------------------------------------------------------------------------------

TABLESPACE	テーブルスペースモードでは、指定されたテーブルスペースに含まれるテーブルがアンロードされます。テーブルがアンロードされる場合、その依存オブジェクトもアンロードされます。オブジェクトのメタデータとデータの両方がアンロードされます。
SCHEMAS	EXP_FULL_DATABASE ロールを持っている場合、スキーマのリストを指定することができます。スキーマ定義そのものを任意で追加することができます。同様に、それらのスキーマにシステム権限を与えることができます。EXP_FULL_DATABASE ロールを持っていない場合、ユーザー本人のスキーマのみをエクスポートできます。
TABLE	テーブルモードでは、指定されたテーブルやパーティション、それらの依存オブジェクトがアンロードされます。ユーザー本人のスキーマにないテーブルを指定するためには、EXP_FULL_DATABASE ロールを持っていない限りなりません。すべての指定されたテーブルは 1 つのスキーマに属さなければなりません。

内容

ALL	データとメタデータの両方をアンロードします。
DATA_ONLY	テーブルの行データをアンロードします; データベースオブジェクト定義は全くアンロードされません。
METADATA_ONLY	データベースオブジェクトの定義をアンロードします; テーブルの行データは全くアンロードされません。

データをエクスポート

エクスポートするオブジェクトを選択します。TABLE モードを選択する場合、**スキーマをエクスポート(テーブルモード)**ドロップダウンリストからスキーマを選択してください。

ダンプファイル

エクスポートに関するダンプファイルセットにダンプファイルを追加します。

メタデータフィルター

エクスポート操作からオブジェクト一式を **INCLUDE** または **EXCLUDE** します。**オブジェクトタイプ**を選択し、**名前句**を指定します。

データフィルター

クエリ

テーブルに対する SELECT 文の最後に追加されるサブクエリを指定します。

サンプル

移動されるデータブロックをサンプリングする割合を指定します。

データを再マップ

テーブルスキーマ

再マップされる列を含むスキーマ。

テーブル名

再マップされる列を含むテーブル。

列名

再マップされる列の名前。

パッケージスキーマ

パッケージのスキーマ。

パッケージ名

パッケージ名。

パッケージファンクション

指定された列に関するデータを修正するために呼び出される PL/SQL パッケージファンクション。

暗号化

暗号化内容

ダンプファイルセットに暗号化するものを指定します。

ALL	エクスポート操作においてすべてのデータとメタデータに対して暗号化を有効にします。
DATA_ONLY	データのみが、暗号化形式でダンプファイルセットに書き込まれます。
ENCRYPTED_COLUMNS_ONLY	暗号化された列のみが、暗号化形式でダンプファイルセットに書き込まれます。
METADATA_ONLY	メタデータのみが、暗号化形式でダンプファイルセットに書き込まれます。
NONE	どのデータも暗号化形式でダンプファイルセットに書き込まれません。

暗号化アルゴリズム

暗号化を実行するために使用されるべき暗号化アルゴリズムを特定します。

暗号化モード

オプション	説明
Transparent	必要な Oracle 暗号化ウォレットを使用できる場合、暗号化されるダンプファイルセットをデータベース管理者(DBA)の介入なしで作成することができます。
Encryption Password	暗号化されたダンプファイルセットの作成時のパスワードを提供します。
Dual	Oracle 暗号化ウォレットまたは ENCRYPTION_PASSWORD パラメータで指定されたパスワードのいずれかを使用して後でインポートされることが可能なダンプファイルセットを作成します。

暗号化パスワード

暗号化されたテーブル列やメタデータ、テーブルデータはダンプファイルセット内でクリアテキストとして書き込まれていないため、それらを再暗号化するためのキーを指定します。

パスワード確認

パスワードを再入力してください。

詳細プロパティ

スレッド数

ジョブのために使用され得るワーカプロセスの最大数。

ファイルを再利用

前から存在するファイルは上書きされます。

XMLCLOBS を有効にする

XMLCLOBS に関するデータオプションを有効にするためには、このボックスにチェックをつけます。

バージョン

抽出されるデータベースオブジェクトのバージョン。

COMPATIBLE	メタデータのバージョンは、データベース互換性レベルと機能の互換性リリースレベルに対応しています。
LATEST	メタデータのバージョンは、データベースのバージョンに対応しています。

圧縮タイプ

ALL	ユーザーデータとメタデータの両方を圧縮します。
DATA_ONLY	ダンプファイルセットのユーザーデータのみを圧縮します。
METADATA_ONLY	ダンプファイルセットのメタデータのみを圧縮します。
NONE	圧縮されていないフォーマットでデータファイルセットを保存します。

トランスポータブル

選択されたテーブルスペース一式の中のテーブル(とそれらに依存するオブジェクト)に関するメタデータに対して作用し、トランスポータブルテーブルスペースエクスポートを実行します。

データベースリンク

現在のジョブのデータやメタデータのソースとなるリモートデータベースへのデータベースリンク名。

見積もり

テーブルサイズの見積もりメソッドがジョブ開始前に実行されるべきであることを指定します。

ログファイルディレクトリ

ログファイルディレクトリを選択します。

ログファイル名

ログファイルの名前を入力します。

フラッシュバック SCN

ユーザーデータを読みこむためのトランザクション一貫性ポイントとなるシステム変更番号(SCN)。

フラッシュバック時刻

ユーザーデータを読みこむための一貫性ポイントの決定に使用される日付と時間、または、TO_TIMESTAMP(...)形式の文字列のいずれか。

Oracle データポンプインポート

データポンプインポートを実行する前に、**SQL を生成** ボタンをクリックして SQL を見直します。それから、**実行** ボタンをクリックして実行することができます。

隠れているタブ(詳細オプション)を表示するためには、**詳細オプションを表示** ボックスにチェックをつけます。

基本プロパティ

ジョブ名

ジョブの名前。

モード

FULL	フルデータベースエクスポートでは、データベース全体がアンロードされます。このモードを使うためには、EXP_FULL_DATABASE ロールを持っていないとなりません。
TABLESPACE	テーブルスペースモードでは、指定されたテーブルスペースに含まれるテーブルがアンロードされます。テーブルがアンロードされる場合、その依存オブジェクトもアンロードされます。オブジェクトのメタデータとデータの両方がアンロードされます。
SCHEMAS	EXP_FULL_DATABASE ロールを持っている場合、スキーマのリストを指定することができます。スキーマ定義そのものを任意で追加することができます。同様に、それらのスキーマにシステム権限を与えることができます。EXP_FULL_DATABASE ロールを持っていない場合、ユーザー本人のスキーマのみをエクスポートできます。
TABLE	テーブルモードでは、指定されたテーブルやパーティション、それらの依存オブジェクトがアンロードされます。ユーザー本人のスキーマにないテーブルを指定するためには、EXP_FULL_DATABASE ロールを持っていないとなりません。すべての指定されたテーブルは 1 つのスキーマに属さなければなりません。

内容

ALL	データとメタデータの両方をアンロードします。
DATA_ONLY	テーブルの行データをアンロードします; データベースオブジェクト定義は全くアンロードされません。

METADATA_ONLY	データベースオブジェクトの定義をアンロードします; テーブルの行データは全くアンロードされません。
---------------	---------------------------------------------------

テーブル存在時のアクション

データが既存のテーブルにロードされる場合に実行されるアクションを指定します。

SKIP	既存のテーブルは変更せずそのままにします。
APPEND	新しい行がテーブルの既存の行に追加されます。
TRUNCATE	既存のテーブルから行を削除した後、インポートから行を挿入します。
REPLACE	既存の表が新しい定義に置き換えられます。新しいテーブルを作成する前に、古いテーブルが削除されます。

インポートデータ

インポートするオブジェクトを選択します。TABLE モードを選択する場合、**スキーマテキストボックス**にスキーマを指定します。

ダンプファイル

インポートに関するダンプファイルセットにダンプファイルを追加します。

ネットワーク

データベースリンク

現在のジョブのデータやメタデータのソースとなるリモートデータベースへのデータベースリンク名。

見積もり

テーブルサイズの見積もりメソッドがジョブ開始前に実行されるべきであることを指定します。

フラッシュバック SCN

ユーザーデータを読みこむためのトランザクション一貫性ポイントとなるシステム変更番号(SCN)。

フラッシュバック時刻

ユーザーデータを読みこむための一貫性ポイントの決定に使用される日付と時間、または、TO_TIMESTAMP(...)形式の文字列のいずれか。

トランスポータブル

選択されたテーブルスペース一式の中のテーブル(とそれらに依存するオブジェクト)に関するメタデータに対して作用し、トランスポータブルテーブルスペースエクスポートを実行します。

データファイルパス

トランスポータブルテーブルスペースセットのデータファイルに関する完全なファイル仕様を指定します。

フィルター

INCLUDE/EXCLUDE

インポート操作からオブジェクト一式を **INCLUDE** または **EXCLUDE** します。**オブジェクトタイプ**を選択し、**名前句**を指定します。

クエリ

テーブルに対する SELECT 文の最後に追加されるサブクエリを指定します。サブクエリの中に WHERE 句を指定する場合、選択される行を制限することができます。

データを再マップ

データを再マップ

フィールド	説明
テーブルスキーマ	再マップされる列を含むスキーマ。
テーブル名	再マップされる列を含むテーブル。
列名	再マップされる列の名前。
パッケージスキーマ	パッケージのスキーマ。
パッケージ名	パッケージ名。
パッケージファンクション	指定された列に関するデータを修正するために呼び出される PL/SQL パッケージファンクション。

データファイルを再マップ

オブジェクトが指定ジョブで処理されるものとして適用されるための再マップを指定します。**ソースデータファイル**と**ターゲットデータファイル**を入力します。

オブジェクトを再マップ

スキーマを再マップ

スキーマが指定ジョブで処理されるものとして適用されるための再マップを指定します。**ソーススキーマ**を入力し、**ターゲットスキーマ**を選択します。

テーブルスペースを再マップ

テーブルスペースが指定ジョブで処理されるものとして適用されるための再マップを指定します。**ソーステーブルスペース**を入力し、**ターゲットテーブルスペース**を選択します。

テーブルを再マップ

テーブルが指定ジョブで処理されるものとして適用されるための再マップを指定します。**ソーステーブル**を入力し、**ターゲットテーブル**を選択します。

詳細プロパティ

スレッド数

ジョブのために使用され得るワーカプロセスの最大数。

データファイルを再利用

テーブルスペースの作成に既存のデータファイルを再利用するためには、このボックスにチェックをつけます。

使用不可インデックスをスキップ

(システムまたはユーザーのいずれかによって)インデックスが使用不可状態に設定されたインデックスを持つテーブルのロードをスキップするためには、このボックスにチェックをつけます。

ストリーム設定

エクスポートダンプファイルに存在する可能性のある一般的な Streams メタデータをインポートするためには、このボックスにチェックをつけます。

データオプション

ジョブの処理に関して特別オプションを提供するビットマスク。

パーティションのオプション

インポート操作中に、パーティション化されたテーブルを処理する方法を指定します。

NONE	パーティションは、ソースデータベースに存在していたとしてターゲットデータベースに再生成されます。
DEPARTITION	ジョブのストレージを含む各パーティションまたはサブパーティションは、パーティション化されていない別々のテーブルとして再生成されます。
MERGE	パーティション化された各テーブルは、パーティション化されていないテーブルとしてターゲットデータベースに再作成されます。

バージョン

抽出されるデータベースオブジェクトのバージョン。

COMPATIBLE	メタデータのバージョンは、データベース互換性レベルと機能の互換性リリースレベルに対応しています。
LATEST	メタデータのバージョンは、データベースのバージョンに対応しています。

暗号化パスワード

暗号化されたテーブル列やメタデータ、テーブルデータがダンプファイルセットにクリアテキストで書き込まれないように、再暗号化するためのキーを指定します。

セグメント属性

トランスフォームが適用されるセグメント属性を指定します。

ストレージ

トランスフォームが適用されるストレージを指定します。

OID

トランスフォームが適用される OID を指定します。

PCT スペース

エクステントの割り当てやデータファイルのサイズを変更するために使用される百分率の乗数を指定します。大きいテーブルスペースをテスト用に縮小する場合に使用します。

ログファイルディレクトリ

ログファイルディレクトリを選択します。

ログファイル名

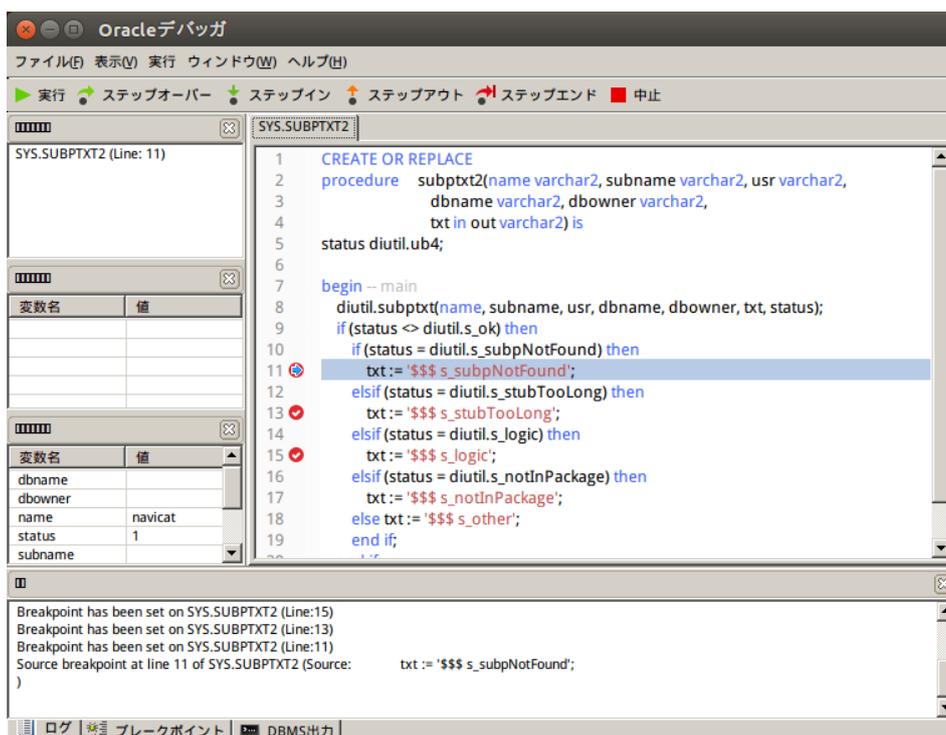
ログファイルの名前を入力します。

Oracle デバッガ (フルバージョンで使用可能)

Navicat は、Oracle ファンクション、プロシージャ、パッケージ、クエリをデバッグするための Oracle PL/SQL デバッガを提供しています。

ワークスペースのアイテムを再編成するためには、ワークスペースのペインをクリックし、カーソルを押したままそのペインを希望の領域にドラッグし、カーソルを放します。選択されたペインはその新しい位置に表示されるはずですが。

ヒント: ペインを新しい位置にドラッグする時、**ペインスティッカー**が表示され、明るい青色のインジケータが挿入ポイントを示します。



ツールバーまたはメニューから、デバッグで最もよく使用されるアクションを実行することができます:

ボタン	説明
実行	デバッグモードでコードの実行を開始します。デバッガは、コードの最後または次のブレークポイントに達するまで、コードを実行します。キーボードショートカット: F9
ステップオーバー	実行を再開します。現在行が実行されます。その行がプロシージャコールまたはファンクションコールの場合、プロシージャまたはファンクションを実行せずに、カウンターはコードの次の行へ移動します。キーボードショートカット: F8
ステップイン	実行を再開します。現在行が実行されます。その行がプロシージャコールまたはファンクショ

	ンコールの場合、カウンターはそのプロシージャまたはファンクション内の最初のステートメントへ進みます。そうでない場合、カウンターはコードの次の行へ移動します。キーボードショートカット: F7
 ステップアウト	実行を再開します。現在のプロシージャまたはファンクション内のコードの残りの部分が実行されます。キーボードショートカット: SHIFT+F7
 ステップエンド	実行を再開します。カウンターはプロシージャまたはファンクションの最後の行にジャンプします。
 中止	コードのステップ実行を止めます。実行は止まり、再開することはできません。

コードペインは、プロシージャ/ファンクションのコードを表示します。各行横のグレーのエリアの  をクリックすることによって、デバッグ用のブレークポイントを追加/削除することができます。ウォッチリストに変数を追加するためには、強調表示されているコードを右クリックし、**ウォッチリストに追加**を選択します。デバッグヒントを表示するためには、コード上にマウスをのせます。

ブレークポイントペインには、全てのブレークポイントが表示され、そのブレークポイントを削除したり有効/無効にすることができます。ブレークポイントを有効/無効にするためには、そのチェックボックスにチェックをつけます/外します。また、ブレークポイントを右クリックし、**ブレークポイントを削除**または**全てのブレークポイントを削除**を選択することによって、1つのブレークポイントまたは全てのブレークポイントを削除することも可能です。ブレークポイントの行にジャンプするためには、それを右クリックし**ファンクションに進む**を選択します。

コールスタックペインには、現在の行のプロシージャコールまたはファンクションコールが表示されます。プロシージャまたはファンクションにジャンプするためには、それを右クリックし**ファンクションに進む**を選択します。

DBMS 出力ペインには、ファンクションまたはプロシージャがその実行を完了した後、結果が表示されます。

ログペインには、コードをデバッグする時に、メッセージログが表示されます。

スマートデータペインには、ブレークポイントに関連する変数についての情報が表示されます。ウォッチリストに変数を追加するためには、変数を右クリックし**ウォッチリストに追加**を選択します。ウォッチ変数の値を調整するためには、変数を右クリックし**値を調整**を選択します。

ウォッチリストペインには、ウォッチされている変数についての情報が表示されます。ウォッチ変数は追加、削除、編集することが可能です。ウォッチ変数を追加するためには、ウォッチリストビューを右クリックし**変数を追加**を選択します。その後、**変数名**を入力します。また、コードウィンドウで強調表示されているコードまたはスマートデータビューの変数を右クリックし、**ウォッチリストに追加**を選択することも可能です。ウォッチ変数の値を調整するためには、変数を右クリックし**値を調整**を選択します。1つのウォッチ変数または全てのウォッチ変数を削除するためには、変数を右クリックし、**変数を削除**または**全ての変数を削除**を選択します。

Oracle 物理属性/デフォルトストレージ特性の編集

% 空き

オブジェクトの行への将来の更新のために予約された、データベースオブジェクトの各データブロック内の領域の割合を、整数表記で指定してください。

% 使用済み

Oracle がデータベースオブジェクトの各データブロックのために維持する使用済み領域の最小割合を整数表記で指定してください。使用済み領域がこの値以下となった時、ブロックは行挿入の対象になります。

初期トランザクション

データベースオブジェクトに割り当てられた各データブロックに割り当てられた同時トランザクションエントリの初期値を指定してください。

最大トランザクション

セグメント内の各データブロックに許可された同時更新トランザクションの最大数を指定してください。

初期値

オブジェクトの初期エクステントのサイズを指定してください。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

次

オブジェクトに割り当てられる次期エクステントのサイズを指定してください。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

最小エクステント

オブジェクトが作成される時に割り当てられるエクステントの総数を指定してください。

最大エクステント

Oracle がオブジェクトに割り当て可能なエクステントの総数を、初期値も含めて指定してください。必要に応じて自動的にエクステントを割り当てたい場合には、**無制限**にチェックをつけてください。

最大サイズ

ストレージエレメントの最大サイズを指定してください。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。ストレージエレメントのディスク領域を制限したくない場合には、**無制限**にチェックをつけてください。

% 増分

3 回目以降のエクステントが、どれくらいの割合まで、それ以前のエクステントを超えて増加するのか、その割合を指定してください。

フリーリスト

テーブルスペースとロールバックセグメント以外のオブジェクト用に、テーブル、パーティション、クラスタ、インデックスの各フリーリストグループ用のフリーリスト数を指定してください。

フリーリストグループ

作成中のデータベースオブジェクト用のフリーリストグループの数を指定してください。

最適値

ロールバックセグメント用の最適値のサイズを指定してください。サイズを指定するためには、ドロップダウンリスト(K(キロバイ

ト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。ロールバックセグメントの最適値サイズを設定しない場合には、**Null** にチェックをつけてください。

バッファプール

DEFAULT	デフォルトバッファプールを指定するためには、これを選択してください。これは、KEEP または RECYCLE に指定されないオブジェクトのためのデフォルトです。
KEEP	ブロックをセグメントから KEEP バッファプールに置くためには、これを選択してください。適切なサイズの KEEP バッファプールを維持することによって、Oracle は、I/O 操作を避けるためにスキーマオブジェクトをメモリ内に保持するようになります。KEEP は、テーブル、クラスタ、マテリアライズド・ビュー、マテリアライズド・ビュー・ログに対して指定した NOCACHE 節よりも優先されます。
RECYCLE	ブロックをセグメントから RECYCLE プールに置くためには、これを選択してください。適切なサイズの RECYCLE プールは、不必要なキャッシュ領域を取り、デフォルトプールが RECYCLE プールであるオブジェクトの数を減らします。

Oracle テーブル

リレーショナルデータベースは、データを保存するためにテーブルを使用します。全てのデータ操作は、テーブル自身の上で行われるか、結果として別のテーブルを生成します。テーブルは行と列の組み合わせで、その交わる部分がフィールドです。一般的な観点から、テーブル内の列は、その列のフィールドに関して行によって検出されるデータの名前やタイプを表します。テーブル内の行は、対応する列の名前とタイプに従って、左から右へ記述されるフィールドから成るレコードを表します。行の各フィールドは、その行の他の各フィールドと暗黙的に関係しています。  をクリックして、**テーブル用オブジェクトリスト**を開きます。

新しいテーブルを作成するためには、オブジェクトリストツールバーから  **新規テーブル**の  下向き矢印をクリックして、**標準 / 外部 / 索引構成**から種類を選択します。

グラフィカルフィールドを含むテーブルを開く場合、テーブルを右クリックし、ポップアップメニューから  **テーブルを開く (高速)** を選択します。セルがクリックされるまで BLOB フィールド(イメージ)はロードされませんので、より高速なパフォーマンスでグラフィカルテーブルを開くことができます。Navicat でテーブルを開く時にすべてのイメージをロードしたい場合、  **テーブルを開く**を使用してください。

テーブルを空にするためには、選択したテーブルを右クリックし、ポップアップメニューから**テーブルをクリア**を選択します。このオプションは、自動増加倍はリセットせずに既存のすべてのレコードをクリアしたい場合のみ適用されます。テーブルを空にする時に自動増加倍をリセットするためには、**テーブルをトランケート**を使用してください。

Oracle 標準テーブル

Oracle データベースにおいて、テーブルはデータストレージの基本単位です。データは、行と列に格納されます。テーブル名と列を使って、テーブルを定義します。

標準(ヒープ構成)テーブルでは、データは順序のない集合(ヒープ)として格納されます。

Oracle テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成や選択したフィールドの削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+Fを押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。 注意: 新規テーブルの作成時のみ使用可能です。
 フィールドを削除	選択したフィールドを削除します。
 プライマリーキー	フィールドにプライマリーキーを設定します。
 上に移動/  下に移動	選択したフィールドを上/下に移動します。 注意: 新規テーブルの作成時のみ使用可能です。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならぬことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプを定義します。詳細については、[Oracle Built-in Datatypes](#) を参照してください。

長さ編集ボックスを使用してフィールドの精度（総桁数）を定義し、スケール編集ボックスを使用して数値列のスケール（小数点の右側の数字の桁数）を定義します。

注意: フィールドの長さを短くする時には、データが失われる可能性があることに注意してください。

ヌルでない

フィールドに NULL の値を許可しない場合、このオプションにチェックをつけます。

プライマリーキー

プライマリーキーは、レコードを一意に定義するための、1つのフィールドまたはフィールドの組み合わせです。プライマリーキーの一部であるフィールドはいずれも、null 値を含むことはできません。

フィールドのポップアップオプション

フィールドを複製	既存のフィールドの1つに変更を加えた新しいフィールドを追加します。
プライマリーキー名	プライマリーキー制約の名前を入力します。

フィールドのプロパティ

注意: 以下のオプションは選択されたフィールドによって異なります。

デフォルト

フィールドのデフォルト値を設定します。

注釈行

現在のフィールドを説明する任意のテキストを設定します。

先行フィールド精度

先行フィールドの桁数を設定します。

小数秒精度

SECOND 日付時間フィールドの小数部分の桁数を設定します。

年精度

年の桁数を設定します。

単位

BYTE または CHAR のいずれかの単位を設定します。

オブジェクトスキーマ

フィールドタイプのオブジェクトスキーマを設定します。

オブジェクトタイプ

フィールドのオブジェクトタイプを設定します。

Oracle テーブルのインデックス

インデックスとは、テーブルやクラスタに関連付けられた任意の構造です。テーブルに対する SQL 文の実行の速度を上げるために、そのテーブルの 1 つ以上の列にインデックスを作成することができます。Oracle データベースのインデックスは、テーブルデータへの高速なアクセスパスを提供します。インデックスの主要目的は、インデックスが適切に使用された時のディスク I/O を削減することです。

各インデックスの列の組み合わせが異なる限り、テーブルに対して複数のインデックスを作成することができます。全く異なる列の組み合わせを指定する場合、同じ列を使用したインデックスを複数作成できます。

インデックスタブにおいて、編集を行うためにはインデックスフィールドをクリックします。インデックスツールバーを使用して、インデックスフィールドの新規作成や、選択したインデックスフィールドの編集、削除を行うことができます。

ボタン	説明
 インデックスを追加	テーブルにインデックスを追加します。
 インデックスを削除	選択したインデックスを削除します。

名前編集ボックスを使用してインデックス名を設定します。

インデックスにフィールドを追加するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

インデックスタイプ

テーブルインデックスのタイプを定義します。

Non-unique	Non-unique インデックスは、列の値に対してインデックスの一意制約はありません。
Unique	Unique インデックスは、テーブルの 2 つの行がキー列に重複する値をもたないことを保証します。
Bitmap	Bitmap インデックスでは、各キー値のビットマップが rowid のリストの代わりに使用されます。

並列度

パラレルインデックスは、大量のデータを持つ時や複数の CPU を持つ時に、インデックスのパフォーマンスを向上させることができます。それぞれのインデックス処理数を決める度数を入力します。

テーブルスペース

インデックスを作成するテーブルスペースです。インデックスは、インデックスを設定するテーブルと同じまたは異なるテーブルスペースに作成できます。

スキーマ

インデックスを作成するスキーマです。

注意: ユーザー自身のスキーマにインデックスを作成するためには、少なくとも次の条件の 1 つを満たさなければなりません:

- インデックスを設定されるテーブルまたはクラスタは、ユーザー自身のスキーマにある。
- ユーザーは、インデックスを設定されるテーブルに対して INDEX 権限を持っている。
- ユーザーは、CREATE ANY INDEX システム権限を持っている。

別のスキーマにインデックスを作成するためには、次の条件すべてを満たさなければなりません:

- ユーザーは、CREATE ANY INDEX システム権限を持っている。
- 他のスキーマの所有者は、インデックスまたはインデックスパーティションを含むテーブルスペースに対するクォータ、または、UNLIMITED TABLESPACE システム権限を持っている。

Oracle テーブルの外部キー

外部キーは、列（または列の集まり）の値が別テーブルの行に存在している値と一致しなければならないことを指定します。これは、2 つの関連テーブル間の参照整合性を維持します。

外部キータブにおいて、編集を行うためには外部キーフィールドをクリックします。外部キーツールバーを使用して、外部キーフィールドの新規作成や、選択した外部キーフィールドの編集、削除を行うことができます。

ボタン	説明
 外部キーを追加	テーブルに外部キーを追加します。
 外部キーを削除	選択した外部キーを削除します。

名前編集ボックスを使用して新しいキーの名前を入力します。

被参照スキーマ、被参照テーブル、被参照制約ドロップダウンリストを使用して、外部のスキーマ、テーブル、制約をそれぞれ選択することができます。

キーにフィールドを追加するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

削除時ドロップダウンリストで、削除時に行われる処理の種類を定義します。

No Action	これはデフォルトの処理です。参照されるキー値は更新も削除もされません。
CASCADE	削除された行を参照している全ての行を削除するか、参照している列の値をそれぞれ被参照列の新しい値に更新します。
SET NULL	参照している列を null に設定します。

有効にする

このボックスにチェックをつける/はずすことによって、外部キー制約を有効/無効にすることができます。

関連トピック:

[外部キーデータ選択](#)

Oracle テーブルの UNIQUE 制約

UNIQUE 制約は、列または列の集まりに含まれるデータが、そのテーブルの全ての行に対して一意であることを保証します。

UNIQUE 制約タブにおいて、編集を行うためには UNIQUE 制約フィールドをクリックします。UNIQUE 制約ツールバーを使用して、UNIQUE 制約フィールドの新規作成や、選択した UNIQUE 制約フィールドの編集、削除を行うことができます。

ボタン	説明
 UNIQUE 制約の追加	テーブルに UNIQUE 制約を追加します。
 UNIQUE 制約の削除	選択した UNIQUE 制約を削除します。

名前編集ボックスを使用して UNIQUE 制約名を設定します。

UNIQUE 制約としてフィールドを設定するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

有効にする

このボックスにチェックをつける/はずすことによって、UNIQUE 制約を有効/無効にすることができます。

Oracle テーブルのチェック制約

チェック制約は最も汎用的な制約です。特定の列の値が論理(真値)式を満たさなければならないことを指定できます。

チェック制約タブにおいて、編集を行うためにはチェック制約をクリックします。チェック制約ツールバーを使用して、チェック制約の新規作成や、選択したチェック制約の編集、削除を行うことができます。

ボタン	説明
 チェック制約の追加	テーブルにチェック制約を追加します。
 チェック制約の削除	選択したチェック制約を削除します。

名前編集ボックスを使用してチェック制約名を設定します。

チェック

チェック編集ボックスにチェック制約の条件を設定します。例えば、"field_name1 > 0 AND field_name2 > field_name1"。列制約として指定されたチェック制約は、その列の値のみ参照します。一方、テーブル制約の式は、複数の列を参照する可能性があります。

定義

チェック制約の定義を入力します。

有効にする

このボックスにチェックをつける/はずすことによって、チェック制約を有効/無効にすることができます。

Oracle テーブルのトリガー

トリガーとは、特定の種類の操作が行われる度にデータベースが特定の関数を自動実行しなければならないという仕様です。INSERT、UPDATE、DELETE 操作の前後のいずれかや、行が変更される度または SQL 文毎に、トリガーを実行するように定義することができます。

トリガータブにおいて、編集を行うためにはトリガーをクリックします。トリガーツールバーを使用して、トリガーフィールドの新規作成や、選択したトリガーフィールドの編集、削除を行うことができます。

ボタン	説明
 トリガーの追加	テーブルにトリガーを追加します。
 トリガーの削除	選択したトリガーを削除します。

名前

トリガー名を設定します。

複合

トリガーを複合トリガーとして設定する場合、チェックをつけます。

注意: Oracle 11g 以降でサポートされています。

行トリガー

トリガーを行トリガーとして設定する場合、チェックをつけます。

起動

トリガーの動作がトリガーを起動する文の前後のどちらに起動されるのか、トリガーのタイミングを指定します。

挿入

INSERT 文がテーブルに行を追加する時、または、ネストされたテーブルに要素を追加する時はいつも、トリガーを起動します。

更新

UPDATE 文が**フィールドの更新**に指定された列のうちの 1 つの列の値を更新する時はいつも、トリガーを起動します。**フィールドの更新**に設定がなければ、UPDATE 文がテーブルまたはネストされたテーブルのいずれかの列の値を変更する時はいつも、トリガーが起動されます。

削除

DELETE 文がテーブルから行を削除する時、または、ネストされたテーブルから要素を削除する時はいつも、トリガーを起動します。

フィールドの更新

必要に応じて、UPDATE 文トリガーに対するフィールドを指定します。

有効にする

このボックスにチェックをつける/はずすことによって、トリガー制約を有効/無効にすることを選択できます。

定義

トリガーの定義を入力します。例:

```
BEGIN
  add_job_history(:old.employee_id, :old.hire_date, sysdate,
    :old.job_id, :old.department_id);
END;
```

条件式

トリガーの条件を指定します。これは、トリガーを起動するためにデータベースに対して満たさなければならない SQL 条件です。この条件は、相関名を含まなければなりません、クエリを含むことはできません。

実行前を参照

相関名を指定します。デフォルトの相関名は OLD です。

実行後を参照

相関名を指定します。デフォルトの相関名は NEW です。

Follows

同じタイプのトリガーの相対的な起動順序を指定します。

注意: Oracle 11g 以降でサポートされています。

スキーマ

指定するスキーマにトリガーを定義します。

Oracle テーブルのオプション

テーブルスペース

テーブルを作るためのデフォルトテーブルスペースとは異なるテーブルスペースを定義します。

ロギング

データベースオブジェクトの作成が REDO ログファイルに記録されるか (LOGGING)、記録されないか (NOLOGGING) を指定します。

圧縮

ディスク使用量を削減するためにデータセグメントを圧縮するかどうかを指定します。ヒープ構成テーブルに対してのみ有効です。

COMPRESS	テーブル圧縮を有効にします。
COMPRESS FOR ALL OPERATIONS	テーブルに対する全ての DML 操作中にデータの圧縮を試みます。
COMPRESS FOR DIRECT_LOAD OPERATIONS	ダイレクトパスの INSERT 操作中にデータの圧縮を試みます (そうすることによって効果がある場合)。
NOCOMPRESS	テーブル圧縮を無効にします。

キャッシュ

バッファキャッシュにどの程度ブロックが格納されるかを指定します。

CACHE	フルテーブルスキャンが実行される時、この表のために取り出されたブロックは、バッファキャッシュ内の最低使用頻度 (LRU) リストの最高使用頻度側に置かれます。
NOCACHE	フルテーブルスキャンが実行される時、この表のために取り出されたブロックは、バッファキャッシュ内の LRU リストの最低使用頻度側に置かれます。

並列度

並列度を指定します。並列度とは、並列処理で使われる並列スレッドの数です。

行移動

このオプションを有効にすると、データベースはテーブル行を移動できるようになります。例えば、テーブル圧縮中やパーティションデータの更新処理中に、行を移動できるようになります。

物理属性

[物理属性/デフォルトストレージ特性](#)を参照してください。

Oracle 外部テーブル

外部テーブルは、外部ソースのデータに対して、それらがデータベース内のテーブルに存在しているかのように、アクセスします。外部テーブルを作成する時、実際は、データディクショナリ内に外部データへのアクセスを可能にするメタデータを作成しているのです。

外部テーブルは、読み込み専用であることに注意してください。DML 操作は不可能です。インデックスを作成することもできません。

Oracle 外部テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成や選択したフィールドの削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+Fを押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。 注意: 新規テーブルの作成時のみ使用可能です。
 フィールドを削除	選択したフィールドを削除します。
 上に移動/  下に移動	選択したフィールドを上/下に移動します。 注意: 新規テーブルの作成時のみ使用可能です。

既存のフィールドの1つに変更を加えた新しいフィールドを追加するためには、右クリックしてポップアップメニューから**フィールドを複製**を選択します。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならないことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプを定義します。詳細については、[Oracle Built-in Datatypes](#)をご覧ください。

長さ編集ボックスを使用してフィールドの**精度**（総桁数）を定義し、**スケール**編集ボックスを使用して**数値列のスケール**（小数点の右側の数字の桁数）を定義します。

注意: フィールドの長さを短くする時には、データが失われる可能性があることに注意してください。

フィールドのプロパティ

注意: 以下のオプションは選択されたフィールドによって異なります。

先行フィールド精度

先行フィールドの桁数を設定します。

小数秒精度

SECOND 日付時間フィールドの小数部分の桁数を設定します。

年精度

年の桁数を設定します。

単位

BYTE または CHAR のいずれかの単位を設定します。

オブジェクトスキーマ

フィールドタイプのオブジェクトスキーマを設定します。

オブジェクトタイプ

フィールドのオブジェクトタイプを設定します。

Oracle 外部テーブルの外部プロパティ

デフォルトディレクトリ

外部テーブルのデフォルトディレクトリを指定します。

ディレクトリ

外部ディレクトリを設定します。

ロケーション

外部ソースロケーションを設定します。

アクセスドライバ

外部テーブルへのアクセスドライバを指定します。外部テーブルのデフォルトタイプは ORACLE_LOADER です。

拒否制限

外部データへの問合せ中に発生可能なエラー数の限界を指定します。

並列度

データソースに対する並列クエリを有効にする場合は、チェックをつけて、並列アクセス数を指定します。

Oracle 外部テーブルへのアクセスパラメータ

Oracle データベースのデータ列への外部データのマッピングを記述します。

CLOB を使用

返されるクエリの CLOB データ値を取得するためには、このオプションにチェックをつけます。

Oracle 索引構成テーブル

索引構成テーブルは、プライマリーB-ツリーの一種であるストレージ構成を持ちます。索引構成テーブルのデータは、プライマリーキー順にソートされた B-ツリーインデックス構造で格納されます。インデックス構造の各リーフブロックは、キー列と非キー列の両方を格納しています。

索引構成テーブルは、完全なテーブル機能を備えています。制約、トリガーなどをサポートし、キー圧縮のような付加機能もサポートします。

注意: 索引構成テーブルのテーブルデザイナーは、**オプションタブ**以外は**標準テーブル**と同じです。したがって、同様のタブに関しては、[標準テーブル](#)を参考にしてください。

Oracle 索引構成テーブルのオプション

テーブルスペース

テーブルを作るためのデフォルトテーブルスペースとは異なるテーブルスペースを定義します。

ロギング

データベースオブジェクトの作成が REDO ログファイルに記録されるか (LOGGING)、記録されないか (NOLOGGING) を指定します。

並列度

並列度を指定します。並列度とは、並列処理で使われる並列スレッドの数です。

行移動

このオプションを有効にすると、データベースはテーブル行を移動することができます。例えば、テーブルの圧縮中もしくはパーティション化されたデータの更新操作中に、行の移動が可能になります。

物理属性

[物理属性/デフォルトストレージ特性](#)を参照してください。

キー圧縮

キー圧縮を有効にするためには、このオプションにチェックをつけます。必要に応じて、プレフィックス長 (キー列の数) も指定可能で、これによってキー列がどのようにプレフィックスエントリとサフィックスエントリに分けられるかを特定します。

マッピングテーブル

索引構成テーブルのマッピングテーブルが存在するかどうかを指定します。マッピングテーブルは、索引構成テーブルにビットマップインデックスを作成する場合に必要とされることに、注意してください。

% 閾値

オーバーフローセグメントが使用されている時に、インデックスブロックに格納される行の一部の最大サイズを、ブロックサイズのパーセンテージで定義します。

オーバーフロープロパティ

オーバーフローストレージ領域を有効にするために、チェックをつけます。

注意: テーブルを保存した後、このオプションのチェックをはずすことはできません。

オーバーフローコラム

別のオーバーフローデータセグメントに配置される列を指定します。

オーバーフローテーブルスペース

オーバーフローセグメントが格納されるテーブルスペースを指定します。

オーバーフローロギング

データベースオブジェクトの作成が、REDO ログファイルに記録されるか(LOGGING)、記録されないか(NOLOGGING)を指定します。

オーバーフロー物理属性

[物理属性/デフォルトストレージ特性](#)を参照してください。

Oracle ビュー

ユーザーがまるで単一テーブルにアクセスしているかのように複数の関連テーブルにアクセスできるようにしたり、そのアクセスを制限したりするために、ビューが役に立ちます。ビューは、行(特定テーブルの一部)へのアクセスを制限するために使用されることもあります。  をクリックして、**ビュー**用のオブジェクトリストを開きます。

ボタン	説明
 プレビュー	ビューの結果をプレビューします。
 説明	ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: **表示** -> **結果を表示** -> **クエリエディタの下** または **新規タブ**を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQLの知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

詳細プロパティ

クエリを制限する

名前を指定します。この識別子を省略する場合、Oracle は自動的に SYS_Cn 形式の名前を割り当てます。この n はデータベース内の制限名を一意にする整数です。

読み取り専用

テーブルまたはビューが更新不可であることを示します。

チェックオプション

Oracle データベースが、サブクエリに含まれない行を生成するテーブルまたはビューへの変更を禁止することを示します。

強制的に作成

ビューの元になるテーブルまたは関連するオブジェクトタイプが存在するかどうかや、ビューを含むスキーマの所有者がそれらに対する権限を持っているかどうかに関わらず、ビューを作成したい場合、このオプションにチェックをつけます。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

Oracle ファンクション/プロシージャ

プロシージャおよびファンクションとは、SQL 文および他の PL/SQL 構造の集合から構成されており、グループ化され、データベースに格納され、単体として実行されるスキーマオブジェクトです。これは、特定の問題の解決したり、関連する一連の処理を実行したりするためのものです。ファンクションが呼び出しに対して常に 1 つの値を返すのに対して、プロシージャは必ずしもそうではないことを除けば、プロシージャとファンクションは同一です。 をクリックして、**ファンクション**用のオブジェクトリストを開きます。

ファンクションウィザード

オブジェクトリストツールバーから  **新規ファンクション** をクリックします。ファンクションウィザードがポップアップ画面で開きます。このウィザードで、簡単にプロシージャ/ファンクションを作成することができます。ファンクションウィザードを表示せずに、プロシージャ/ファンクションを新規作成することもできます。

1. ルーチンの**名前**を指定します。ルーチンの種類を選択します：**プロシージャ** または **ファンクション**
2. プロシージャ/ファンクションのパラメータを定義します。該当する列に、**名前**、**タイプ**、**モード**、**デフォルト値**パラメータを設定します。
3. ファンクションを作成する場合、リストから**リターンタイプ**を選択します。

ヒント: **次回ウィザードを表示**のチェックを外した場合、[オプション](#)で再度有効にすることができます。

定義

コードアウトラインウィンドウには、パラメータやコード本体などを含むファンクション/プロシージャに関する情報が表示されます。コードアウトラインウィンドウを表示するためには、**表示** -> **コードアウトライン**を選択します。

注意: フルバージョンで使用可能です。

ボタン	説明
	コードアウトラインを更新します。
	コードアウトラインの詳細ビューを表示します。
	マウスオーバー時の強調表示を on または off に変更します。
	選択アイテムを展開します。
	選択アイテムを折りたたみます。
	位置によるソートを切り替えます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

結果

プロシージャ/ファンクションを実行するために、ツールバーの  **実行**をクリックします。SQL 文が正しい場合、その文が実行されます。そして、その文がデータを返すことになっている場合、プロシージャ/ファンクションによって返されたデータを表示した **DBMS 出力**タブが開きます。プロシージャ/ファンクションの実行時にエラーが発生した場合、実行は中止され、適切なエラーメッセージが表示されます。ファンクション/プロシージャがパラメータの入力を必要とする場合、**パラメータ入力**ボックスがポップアップ表示されます。

ヒント: Navicat は、10 個の結果セットを返すことに対応しています。

デバッグ (フルバージョンで使用可能)

ファンクション/プロシージャをデバッグするために、ツールバーの  **デバッグ**をクリックして、[Oracle デバッガ](#)を起動します。必要に応じて、入力パラメータを入力します。

各行横のグレーのエリアで  をクリックすることによって、デバッグ用のブレークポイントを追加/削除することができます。

Oracle データベースリンク

データベースリンクは、名前のついたスキーマオブジェクトで、あるデータベースから別のデータベースへのパスを表します。データベースリンクは、分散データベースのグローバルオブジェクト名に対して、参照が作成される場合に、暗黙的に使用されます。データベースリンクを作成した後、他のデータベースのテーブルとビューを参照するために、データベースリンクを使用することができます。  -> **データベースリンク**をクリックして、**データベースリンク**用のオブジェクトリストを開きます。

基本プロパティ

サービス名

リモートデータベースのサービス名を指定します。

ユーザー名

設定済みのユーザーのデータベースリンクを使用してリモートデータベースに接続する時に使用されるユーザー名。

パスワード

リモートデータベースへの接続用のパスワード。

現在のユーザー

このオプションにチェックをつけると、現在のユーザーのデータベースリンクが作成されます。現在のユーザーは、リモートデータベースに有効なアカウントを持つグローバルユーザーでなければなりません。

共有

共有オプションが有効である場合、**認証ユーザー名**と **認証パスワード**を入力します。

Oracle インデックス

インデックスによって、より高速にテーブルデータにアクセスすることができます。インデックスは、テーブルの 1 つ以上の列を使用して作成され、そのテーブルでの SQL 文の実行スピードを上げます。🏗️ -> **インデックス**をクリックして、**インデックス**用のオブジェクトリストを開きます。

インデックスの**タイプ**を選択することができます：

Normal(標準)	標準インデックスは、列の値に対する制約がありません。
Unique(一意)	一意インデックスは、テーブルの 2 つの行がキー列に重複する値を持ちません。
Bitmap(ビットマップ)	各行別々にインデックス化されるのではなく、各個別キーのビットマップで作成されるビットマップインデックス。ビットマップインデックスは、ビットマップとしてキー値に関連付けられた rowid を保存します。ビットマップ内の各ビットは、使用可能な rowid に対応しています。
Domain(ドメイン)	空間処理やイメージ処理など、特化されたドメイン向けに設計されたインデックスです。設計者がインデックスタイプを作成した後、ユーザーが任意のタイプのドメインインデックスを作成できます。
Cluster(クラスター)	クラスタインデックスは、クラスタ向けに設計されたインデックスです。

基本プロパティ(Normal/Unique インデックス)

テーブルスキーマ

インデックスを含むスキーマ。

テーブル名

テーブルの名前。

列

名前ドロップダウンリストを使用して、フィールド名を選択します。そして、**順序**ドロップダウンリストを使用して、インデックスの順序を定義します(ASC または DSC)。

基本プロパティ(Bitmap インデックス)

テーブルスキーマ

インデックスを含むスキーマ。

テーブル名

テーブルの名前。

Bitmap Join インデックス

単一テーブルのビットマップインデックスだけでなく、ビットマップジョインインデックスを作成することもできます。これは、2つ以上のテーブルの結合用のビットマップインデックスです。ビットマップジョインインデックスは、事前に制限が実行されることによって結合されるデータ量を削減するという、スペースに関して効果的な方法です。

列

スキーマ、テーブル、名前ドロップダウンリストを使用して、スキーマ、テーブル、フィールド名を選択します。**順序**ドロップダウンリストを使用して、インデックスの順序を定義します(ASC または DSC)。

Bitmap Join

インナースキーマ、インナーテーブル、インナーフィールド、アウタースキーマ、アウターテーブル、アウターフィールドドロップダウンリストを使用して、結合されるスキーマ、テーブル、フィールドをそれぞれ選択します。

基本プロパティ(Domain インデックス)

テーブルスキーマ

インデックスを含むスキーマ。

テーブル名

テーブルの名前。

列

インデックスの元となる列。

スキーマ

インデックスタイプのスキーマ。

タイプ

作成または組み込まれたインデックスタイプを選択します。

パラメータ

パステーブルに関する情報や、XML インデックスの構成要素に対応するセカンダリインデックスに関する情報。

基本プロパティ(Cluster インデックス)

テーブルスキーマ

インデックスを含むスキーマ。

クラスタ名

クラスタの名前。

詳細プロパティ

使用不可

使用不可のインデックスは、使用前に、再構築されるか、削除され再作成されなければなりません。

テーブルスペース

インデックスを保持するテーブルスペース名。

圧縮

キー列の値の繰り返し発生を排除し、実質的にストレージを少なくするために、キーの圧縮を有効にします。

注意: Bitmap インデックスに関しては圧縮できません。

パラレル

インデックスの作成が並列で処理されます。

リバース

rowid を除き、インデックスブロックのバイトを逆の順番で格納します。

ロギング

REDO ログファイルにインデックスの作成を記録するためには、**Logging** を選択します。ログに書き込まない場合は、**No Logging** を選択します。

可視性

インデックスがオプティマイザに対して **Visible**(可視)または **Invisible**(不可視)を指定します。

オンライン

インデックス作成中にテーブルに対する DML 操作が可能であることを表します。

ソートしない

インデックスの作成時に Oracle データベースが行のソートを行わなくてもいいように、行が既に昇順で格納されているということを、データベースに示します。

物理属性

インデックスの[物理属性](#)を設定します。

Oracle Java

Java は、アプリケーションレベルのプログラムに有用なオブジェクト指向プログラム言語です。データベース内に、アプリケーションを書き込んだり、ロードすることができます。  -> **Java** をクリックして、**Java** 用のオブジェクトリストを開きます。

タイプを選択します: Java Source(ソース)、Java Class(クラス)、Java Resource(リソース)のいずれか。

基本プロパティ (Java Source)

BFile

ディレクトリを選択し、**サーバーファイル名**を入力します。

Load from file

Java ソースファイルの**ファイルパス**を表示します。

Plain source

Source ボックスにソースコードを入力します。

基本プロパティ (Java Class/Java Resource)

BFile

ディレクトリを選択し、**サーバーファイル名**を入力します。

Load from file

Java クラスファイルまたは Java リソースファイルの**ファイルパス**を表示します。

詳細プロパティ

Invoker 権限

CURRENT_USER を選択すると、クラスの方法が CURRENT_USER の権限を使用して実行することを表します。**DEFINER** を選択すると、クラスの方法がクラスの属するスキーマの所有者の権限を使用して実行し、外部名がクラスの属するスキーマ内で解決することを表します。

Resolver

Java スキーマオブジェクトに対する完全修飾 Java 名のマッピングを指定します。

Compile / Resolve

Oracle データベースがこのステートメントの成功時に作成される Java スキーマオブジェクトを解決しようとするべきであることを指定する場合、チェックをつけます。

No Force

上記の Compile / Resolve が有効な時に、解決またはコンパイルに失敗した場合、Java の CREATE コマンドの結果をロールバックするためには、これにチェックをつけます。このオプションを指定しない場合は、解決またはコンパイルに失敗すると、Oracle データベースは何の対処もせず、作成されたスキーマオブジェクトがそのまま残ります。

Oracle マテリアライズド・ビュー

マテリアライズド・ビューは、データの集約、計算、複製、分散に使用されるスキーマオブジェクトです。  -> **マテリアライズド・ビュー** をクリックして、**マテリアライズド・ビュー**用のオブジェクトリストを開きます。

ボタン	説明
-----	----

 プレビュー	マテリアライズド・ビューの結果をプレビューします。
 説明	マテリアライズド・ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: 表示 -> 結果を表示 -> クエリエディタの下 または **新規タブ**を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQL の知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

詳細プロパティ

条件

Demand	3 つの DBMS_MVIEW 更新プロシージャの 1 つを呼び出すことによって、マテリアライズド・ビューは要求に応じて更新されます。
Commit	マテリアライズド・ビューのマスターテーブルに作用するトランザクションをデータベースがコミットする時はいつも、高速更新されます。
Automatic	データベースは、自動更新時間を使い、マテリアライズド・ビューを自動的に更新します。
Never	マテリアライズド・ビューは、Oracle データベース更新メカニズムまたはパッケージプロシージャを使って更新されません。

メソッド

FORCE	更新発生時、Oracle データベースは、可能な場合、高速更新(FAST)を実行します。高速更新が可能でない場合、完全更新(COMplete)をします。
FAST	マスターテーブルに起こった変更に従って更新を実行するという、インクリメンタルな更新方法です。
COMPLETE	マテリアライズド・ビューの定義クエリを実行することによって実装される、完全な更新方法です。

開始

最初の自動更新時間のための日付時間式。

次

自動更新の間隔を計算するための日付時間式。

注意: 日付時間を編集するためには、... をクリックして、希望のデータを選択 / 入力します。

タイプ

Primary Key	プライマリーキー・マテリアライズド・ビューです。
Row ID	rowid マテリアライズド・ビューです。

マスター

リモートのロールバックセグメントは、個々のマテリアライズド・ビュー用のリモートマスターサイトで使用されます。

ローカル

リモートのロールバックセグメントは、マテリアライズド・ビューが含まれるローカルリフレッシュグループで使用されます。

制約

Enforced	Oracle データベースは、更新操作中、強制的に制約を使用します。
Trusted	Oracle データベースは、データベースによって確認されていないがデータベース管理者によって信頼性を宣言されたディメンションと制約情報を使用します。

No Index

デフォルトインデックスの作成を抑えるためには、これにチェックをつけます。

ビルドタイプ

Immediate	マテリアライズド・ビューはすぐに投入されます。
Deferred	マテリアライズド・ビューは次回の更新操作によって投入されます。
Prebuilt	すでに初期化されたマテリアライズド・ビューとして既存のテーブルを登録します。

Reduced Precision

With	テーブルまたはマテリアライズド・ビューのカラムの精度がサブクエリによって返される精度と正確に一致しない場合に生じる精度の損失を許可します。
Without	テーブルまたはマテリアライズド・ビューのカラムの精度がサブクエリによって返される精度と正確に一致するように要求します。一致しない場合、作成の操作は失敗します。

圧縮

データセグメントを圧縮して、ディスクとメモリの使用を削減します。

パラレル

直列実行の場合、**No Parallel** を選択します。Oracle が、初期化パラメータ `PARALLEL_THREADS_PER_CPU` の値と全参加インスタンスで使用可能な CPU の数とをかけた値に等しい並列度を選択するようにしたい場合、**Parallel** を選択します。

並列度

作成後のマテリアライズド・ビューにおけるクエリと DML のデフォルトの並列度を設定します。

ロギング

REDO ログファイルにマテリアライズド・ビューの作成をロギングするためには、**LOGGING** を選択します。ロギングしない場合、**NOLOGGING** を選択します。

テーブルスペース

マテリアライズド・ビューが作成されるテーブルスペースを選択します。

物理属性

マテリアライズド・ビューの[物理属性](#)を設定します。

キャッシュを有効にする

フルテーブルスキャンが実行される時、テーブルに対して取得されたブロックは、バッファキャッシュ内の最低使用頻度リスト (LRU)の中で、ごく最近に使用された部分の終端に置かれます。

For Update

サブクエリ、プライマリーキー、オブジェクトまたは rowid のマテリアライズド・ビューを更新するためには、このオプションにチェックをつけます。高度なレプリケーションと組み合わせて使用される場合、これらの更新はマスターに伝えられます。

クエリ・リライトを有効にする

マテリアライズド・ビューが、クエリの再書き込みに使用されます。

Using Index 節

オプション	説明
テーブルスペース	インデックスのテーブルスペースを選択します。
物理属性	Oracle データベースがマテリアライズド・ビューのデータを保守するために使用するデフォルトインデックスの 物理属性 を設定します。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

Oracle マテリアライズド・ビュー・ログ

マテリアライズド・ビュー・ログは、マスターテーブルのデータへの変更を記録するスキーマオブジェクトです。これによって、マスターテーブルに定義された[マテリアライズド・ビュー](#)は増分更新されることが可能となります。-> **マテリアライズド・ビュー・ログ**をクリックして、**マテリアライズド・ビュー・ログ**用のオブジェクトリストを開きます。

基本プロパティ

テーブル

マテリアライズド・ビュー・ログのテーブル。

テーブルスペース

マテリアライズド・ビュー・ログのテーブルスペース。

ロギング

LOGGING または **NOLOGGING** を指定して、マテリアライズド・ビュー・ログのロギング特性を設定します。

キャッシュ

CACHE	フルテーブルスキャンが実行される時、このログ用に取得されたブロックは、バッファキャッシュ内の最低使用頻度リスト(LRU)の中で、ごく最近に使用された部分の終端に置かれます。
NOCACHE	ブロックは、LRU リストの中で、最も長い間使われていない部分の終端に置かれます。

新しい値

INCLUDING	新しい値と古い値の両方をログに保存します。
EXCLUDING	ログに新しい値を記録することを無効にします。

並列度

並列操作で使用される並列スレッドの数を決定します。

物理属性

マテリアライズド・ビュー・ログの[物理属性](#)を設定します。

オブジェクト ID

変更された全ての行のシステム生成オブジェクト識別子またはユーザー定義オブジェクト識別子が、マテリアライズド・ビュー・ログに記録されます。

プライマリーキー

変更された全ての行のプライマリーキーが、マテリアライズド・ビュー・ログに記録されます。

Row ID

変更された全ての行の rowid が、マテリアライズド・ビュー・ログに記録されます。

シーケンス

追加順番情報を提供するシーケンス値が、マテリアライズド・ビュー・ログに記録されます。

フィールド

フィールドを選択すると、その行の値が変更される時、マテリアライズド・ビュー・ログに記録されます。

Oracle パッケージ

パッケージは、関連するプロシージャ、ストアドファンクションや、データベースと一緒に格納される他のプログラムオブジェクトの集まりで、カプセル化されています。パッケージに続いて指定されるパッケージボディが、これらのオブジェクトを定義しま

す。パッケージは、2つの部分から構成されています：仕様とボディです。 -> **パッケージ**をクリックして、**パッケージ**用のオブジェクトリストを開きます。

コードアウトラインウィンドウには、ファンクションやプロシージャ、パラメータ、コード本体などを含むパッケージ/パッケージボディに関する情報が表示されます。**コードアウトライン**ウィンドウを表示するためには、**表示** -> **コードアウトライン**を選択します。

注意: フルバージョンで使用可能です。

ボタン	詳細
	コードアウトラインを更新します。
	コードアウトラインの詳細ビューを表示します。
	マウスオーバー時の強調表示を on または off に変更します。
	選択アイテムを展開します。
	選択アイテムを折りたたみます。
	位置によるソートを切り替えます。

パッケージの定義

パッケージの定義を入力します。パッケージを保存した後、パッケージボディを編集することができます。 **新規パッケージボディ**または  **パッケージボディを設計**をクリックして、パッケージボディデザイナーを開きます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

パッケージボディの定義

パッケージボディの定義を入力します。パッケージ仕様を編集するためには、 **パッケージ仕様を設計**をクリックし、パッケージデザイナーを開きます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

結果

パッケージを実行するためには、ツールバーの  **実行**をクリックします。SQL 文が正しい場合、その文が実行されます。そして、その文がデータを返すことになっている場合、パッケージによって返されたデータを表示した **DBMS 出力**タブが開きます。パッケージの実行中にエラーが発生した場合、実行は中止され、適切なエラーメッセージが表示されます。ファンクション/プロシージャがパラメータの入力を必要とする場合、ファンクション/プロシージャを選択し、パラメータを入力します。

デバッグ(フルバージョンで使用可能)

パッケージをデバッグするために、ツールバーの  **デバッグ**をクリックして、[Oracle デバッグ](#)を起動させます。ファンクション/プロシージャがパラメータの入力を必要とする場合、ファンクション/プロシージャを選択し、パラメータを入力します。

各行横のグレーのエリアで  をクリックすることによって、デバッグ用のブレークポイントを追加/削除することができます。

Oracle シーケンス

シーケンスは、行が 1 行だけの新しい特別なテーブルを作成し、初期化します。通常は、テーブルの行に対して一意な識別子を生成するために使用されます。  -> **シーケンス**をクリックして、**シーケンス**用のオブジェクトリストを開きます。

基本プロパティ

インクリメント

新しい値を作成するために、現在のシーケンス値に追加される値を指定します。正の値は、昇順のシーケンスを作成し、負の値は、降順のシーケンスを作成します。デフォルト値は 1 です。

開始

生成される最初のシーケンス番号を指定します。

最小

シーケンスが生成できる最小値です。

最大

シーケンスの最大値です。

キャッシュ

より高速なアクセスを可能にするために、データベースによってメモリに事前に割り当てられ保存されるシーケンス番号の数を指定します。このパラメータの最小値は 2 です。

キャッシュなし

このオプションは、シーケンスの値が事前に割り当てられていないことを示します。

循環

このオプションにチェックをつけると、シーケンスは、その最大値または最小値に達した後、値を生成し続けることができます。昇順のシーケンスがその最大値に達した後は、最小値を生成します。降順のシーケンスがその最小値に達した後は、最大値を生成します。

順序

このオプションにチェックをつけると、シーケンス番号がリクエスト順に生成されることが保証されます。

Oracle シノニム

シノニムは、テーブル、ビュー、マテリアライズド・ビュー、シノニム、プロシージャ、ファンクション、パッケージ、タイプ、Java クラスのスキーマオブジェクト、ユーザー定義オブジェクトタイプ、または、別のシノニムに対するエイリアスです。シノニムは単なるエイリアスであるため、データディレクトリに、その定義以外に保存領域を必要としません。  -> **シノニム**をクリックして、**シノニム**用のオブジェクトリストを開きます。

基本プロパティ

オブジェクトデータベースリンク

スキーマオブジェクト用のシノニムをそのオブジェクトが置かれているリモートデータベース上に作成するための、完全データベースリンクまたは部分データベースリンクです。

オブジェクトスキーマ

オブジェクトが存在するスキーマです。

オブジェクトタイプ

オブジェクトのタイプです。

オブジェクト

このオブジェクトに対してシノニムが作成されます。

Oracle トリガー

トリガーは、プロシージャと似ています。データベースに格納されたトリガーは、SQL や PL/SQL、または、一体となって実行される Java ステートメントを含むことが可能です。また、トリガーは、プロシージャを呼び出すことができます。🏭 -> **トリガー** をクリックして、**トリガー**用のオブジェクトリストを開きます。

詳細については、[トリガー](#)を参照してください。

トリガーのタイプを選択します: TABLE、VIEW、SCHEMA または DATABASE。

テーブルトリガーの基本プロパティ

有効にする

トリガー文が発行され、トリガー制限が(存在する場合に)true と判断された場合、有効なトリガーはそのトリガーアクションを実行します。

テーブルスキーマ

トリガーのテーブルスキーマです。

テーブル名

トリガーを作成したいテーブルです。

複合

複合トリガーとは、テーブルに対する単一トリガーで、4つのタイミングポイントそれぞれに対してアクションを指定することができます:

タイミングポイント	節
トリガー文の実行前	BEFORE STATEMENT
トリガー文の実行後	AFTER STATEMENT

トリガー文が影響を及ぼす各行の前	BEFORE EACH ROW
トリガー文が影響を及ぼす各行の後	AFTER EACH ROW

注意: Oracle 11g 以降でサポートされています。トリガーの定義で SQL を編集することができます。

起動

トリガーを定義する時、トリガーアクションがトリガー文の**前(BEFORE)**または**後(AFTER)**のどちらに実行されるか、トリガーのタイミングを指定できます。

For Each

Oracle データベースは、トリガー文の影響を受ける行ごとに 1 回ずつ、**ROW** トリガーを実行します。また、トリガー文が発行され、かつ、その任意のトリガー制約が満たされる場合のみ、発行ごとに **STATEMENT** トリガーを実行します。

条件

トリガーの条件文を指定します。これは、トリガーを起動するために、データベースに対して満たされなければならない SQL 条件です。

挿入

トリガーは、テーブルに行を追加またはネストされたテーブルに要素を追加する時、常に起動されます。

削除

トリガーは、テーブルから行を削除またはネストされたテーブルから要素を削除する時、常に起動されます。

更新

トリガーは、**フィールドの更新**で選択されたフィールドの 1 つの値を変更する時、常に起動されます。

ビュートリガーの基本プロパティ

有効にする

トリガー文が発行され、トリガー制限が(存在する場合に)true と判断された場合、有効なトリガーはそのトリガーアクションを実行します。

テーブルスキーマ

トリガーのビュースキーマです。

テーブル名

トリガーを作成したいビューです。

ネストされたテーブルフィールド

ネストされたテーブルのフィールドを選択します。

複合

Instead Of Trigger を指定します。

注意: Oracle 11g 以降でサポートされています。トリガーの定義で SQL を編集することができます。

挿入

トリガーは、テーブルに行を追加またはネストされたテーブルに要素を追加する時、常に起動されます。

削除

トリガーは、テーブルから行を削除またはネストされたテーブルから要素を削除する時、常に起動されます。

更新

トリガーは、行の値を変更する時はいつも起動されます。

スキーマトリガーの基本プロパティ

有効にする

トリガー文が発行され、トリガー制限が(存在する場合に)true と判断された場合、有効なトリガーはそのトリガーアクションを実行します。

テーブルスキーマ

トリガーのテーブルスキーマです。

起動

トリガーを定義する時、トリガーアクションがトリガー文の**前(BEFORE)**または**後(AFTER)**のどちらに実行されるか、トリガーのタイミングを指定できます。

条件

トリガーの条件文を指定します。これは、トリガーを起動するために、データベースに対して満たされなければならない SQL 条件です。

イベント

トリガーを起動するステートメントの種類を示します。

データベーストリガーの基本プロパティ

有効にする

トリガー文が発行され、トリガー制限が(存在する場合に)true と判断された場合、有効なトリガーはそのトリガーアクションを実行します。

起動

トリガーを定義する時、トリガーアクションがトリガー文の**前(BEFORE)**または**後(AFTER)**のどちらに実行されるか、トリガーのタイミングを指定できます。

条件

トリガーの条件文を指定します。これは、トリガーを起動するために、データベースに対して満たされなければならない SQL 条件です。

イベント

トリガーを起動するステートメントの種類を示します。

テーブル/ビュートリガーの詳細プロパティ

実行前を参照

ネストされた古いテーブルの関連名です。

実行後を参照

ネストされた新しいテーブルの関連名です。

親を参照

親テーブルの関連名です。

Follows

指定したトリガーの後に起動するトリガーを示します。**スキーマ**ドロップダウンリストを使用してスキーマ名を選択し、**トリガー**ドロップダウンリストを使用してトリガーを選択します。

注意: Oracle 11g 以降でサポートされています。

定義

BEGIN と *END* 内のトリガー定義で、有効な SQL 文またはプロシージャ文を編集することができます。

Oracle タイプ

タイプは、アプリケーションでデータの構造や動作を形作るユーザー定義のデータタイプです。オブジェクトタイプは、2つの部分から構成されています：仕様とボディです。タイプボディは、そのタイプの仕様に常に依存します。コレクションタイプは、名前付きの可変配列またはネストされたテーブルタイプです。🔧 -> **タイプ**をクリックして、**タイプ**用のオブジェクトリストを開きます。

コードアウトラインウィンドウには、宣言などを含むオブジェクトタイプ/オブジェクトタイプボディに関する情報が表示されます。コードアウトラインウィンドウを表示するためには、**表示** -> **コードアウトライン**を選択します。

注意: フルバージョンで使用可能です。

ボタン	説明
	コードアウトラインを更新します。
	コードアウトラインの詳細ビューを表示します。
	マウスオーバー時の強調表示を on または off に変更します。
	選択アイテムを展開します。
	選択アイテムを折りたたみます。
	位置によるソートを切り替えます。

オブジェクトタイプの定義

オブジェクトタイプの定義を入力します。オブジェクトタイプを保存後、オブジェクトタイプボディを編集することができます。
新規オブジェクトタイプボディまたは  オブジェクトタイプボディの設計をクリックして、タイプボディデザイナーを開きます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

オブジェクトタイプボディの定義

オブジェクトタイプボディの定義を入力します。オブジェクトタイプの仕様を編集するためには、 オブジェクトタイプ仕様の設計をクリックして、オブジェクトタイプデザイナーを開きます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

コレクションタイプ

ネストされたテーブル

ネストされたテーブルタイプを作成します。

VArray

VArray タイプを作成します。

配列サイズ

VArray タイプの配列サイズを決定します。

データタイプ

Oracle データベース固有のデータタイプまたはユーザー定義の属性のタイプを選択します。

データタイプパラメータ

対応するデータタイプパラメータを決定します。

Oracle XML スキーマ

XML スキーマは、XML で記述されたスキーマ定義言語です。適合インスタンスドキュメントの構造や他のさまざまな意味合いを記述するために使用されます。 -> **XML スキーマ**をクリックして、**XML スキーマ**用のオブジェクトリストを開きます。

スキーマドキュメント

スキーマドキュメントタブに、有効な XML スキーマドキュメントを入力します。

詳細プロパティ

ローカル

これにチェックをつけると、ローカルスキーマとして登録します。

スキーマを強制登録

これにチェックをつけると、スキーマの変更中に発生するエラーを無視します。

オブジェクトタイプ

これにチェックをつけると、スキーマコンパイラは、オブジェクトタイプを生成することができます。

Java Beans

これにチェックをつけると、スキーマコンパイラは、Java beans を生成することができます。

デフォルトテーブル

これにチェックをつけると、スキーマコンパイラは、デフォルトテーブルを生成することができます。

REGISTER_NODOCID

これにチェックをつけると、ユーザーがストレージの最適化をしたい場合、この列を作成しません。

REGISTER_BINARYXML

これにチェックをつけると、バイナリ XML 用のスキーマを登録します。

REGISTER_NT_AS_IOT

これにチェックをつけると、スキーマの登録中に作成されるネストされたテーブルを索引構成テーブルとして格納します。

REGISTER_AUTO_OOL

これにチェックをつけると、大きいタイプを行の外に自動的に移動します。

Hierarchy を有効にする

ENABLE_HIERARCHY_NONE	Enable Hierarchy は、そのスキーマの登録中に作成されるどのテーブルでも呼び出されません。
ENABLE_HIERARCHY_CONTENTS	Enable Hierarchy は、スキーマ登録中に作成される全てのテーブルで、 hierarchy_type が DBMS_XDBZ.ENABLE_CONTENTS として、呼び出されます。
ENABLE_HIERARCHY_RESMETADATA	Enable Hierarchy は、スキーマ登録中に作成される全てのテーブルで、 hierarchy_type が DBMS_XDBZ.ENABLE_RESMETADATA として、呼び出されま す。

Oracle リサイクルビン

リサイクルビンは、削除されたオブジェクトに関する情報を含むデータディクショナリテーブルです。削除されたテーブルと関連したオブジェクト(インデックスや制約、ネストされたテーブルなど)は、削除されておらず、まだ領域を占有しています。リサイクルビンから明確に削除されるか、テーブルスペースのスペース制約が原因でデータベースによって削除されなければならないというまれな状況になるまで、それらはユーザーのスペースクォータに作用し続けます。  -> **リサイクルビン**をクリックして、**リサイクルビン**用のオブジェクトリストを開きます。

テーブルを復元するために

1. リサイクルビンにあるテーブルを選択します。
2. オブジェクトリストツールバーから  **テーブルをフラッシュバック**をクリックします。

オブジェクトを削除するために

1. オブジェクトリストペインから、削除するオブジェクトを選択します。
2. オブジェクトリストツールバーから  **オブジェクトを削除**をクリックします。
3. ダイアログウィンドウが開くので、確認して削除します。

すべてのオブジェクトを削除するために

1. 右クリックし、ポップアップメニューから**リサイクルビンを削除**を選択します。
2. ダイアログウィンドウが開くので、確認して削除します。

全ユーザーのすべてのオブジェクトを削除するために

1. **SYSDBA** 権限を持つユーザーでログインします。
2. 右クリックし、ポップアップメニューから **DBA リサイクルビンを削除**を選択します。
3. ダイアログウィンドウが開くので、確認して削除します。

Oracle ディレクトリ

ディレクトリオブジェクトは、外部バイナリファイル LOB (BFILE) と外部テーブルデータが置かれたサーバーファイルシステム上に、ディレクトリ用のエイリアスを指定します。すべてのディレクトリは単一のネームスペースに作成され、個々のスキーマによって所有されません。  -> **ディレクトリ**をクリックして、**ディレクトリ**用のオブジェクトリストを開きます。

基本プロパティ

ディレクトリパス

ファイルが置かれているサーバーのオペレーティングシステムディレクトリのフルパス名を指定します。パス名は、大文字小文字を区別します。

Oracle テーブルスペース

テーブルスペースは、スキーマオブジェクトを含むことができる、データベース内の領域の割り当てです。  -> **テーブルスペース**をクリックして、**テーブルスペース**用のオブジェクトリストを開きます。

基本プロパティ

テーブルスペースのタイプ

PERMANENT	永続テーブルスペースは、永続的なスキーマオブジェクトを含みます。永続テーブルスペースのオブジェクトは、データファイルに格納されます。
TEMPORARY	一時テーブルスペースは、セッションの存続時間のみ、スキーマオブジェクトを含みます。一時テーブルスペース

	ースのオブジェクトは、一時ファイルに格納されます。
UNDO	Undo テーブルスペースは、自動 Undo 管理モードでデータベースを実行する場合に、Undo データを管理するために Oracle データベースによって使用される永続的なテーブルスペースタイプです。

名前

データファイル / 一時ファイルの名前を設定します。

サイズ

データファイル / 一時ファイルのサイズを設定します。

単位

データファイル / 一時ファイルのサイズの単位を定義します。データファイルの自動拡張に許容される最大ディスクスペースを指定します。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

再利用

Oracle が既存のファイルを再利用できるようにします。

パス

データファイル / 一時ファイルのパスを指定します。

自動拡張

新規または既存のデータファイルまたは一時ファイルの自動拡張を **ON** (有効) または **OFF** (無効) にします。

次サイズ

より多くの領域が必要となる場合に自動的に割り当てられるディスクスペースの、次の増分のサイズをバイトで指定します。デフォルトは、1 データブロックのサイズです。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

最大制限なし

Oracle がデータファイルまたは一時ファイルに割り当てることができるディスクスペースは、制限されません。

最大サイズ

データファイルの自動拡張に許容される最大ディスクスペースを指定します。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

ストレージ

ファイルタイプ

BIGFILE	BIGFILE テーブルスペースは、およそ 40 億(2^{32})ブロックまで含むことが可能なデータファイルまたは一時ファイルを 1 つだけ含みます。そのたった 1 つのデータファイルまたは一時ファイルの最大サイズは、32K ブロックのテーブルスペースでは 128 テラバイト(TB) であり、8K ブロックのテーブルスペースでは 32TB となります。
SMALLFILE	SMALLFILE テーブルスペースは、それぞれがおよそ 400 万(2^{22})ブロックまで含むことができる、1022 のデ

ータファイルまたは一時ファイルを含むことが可能な、従来の Oracle のテーブルスペースです。

最小エクステントサイズ

テーブルスペースの領域の最小サイズです。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

ブロックサイズ

テーブルスペースのブロックサイズです。

デフォルトストレージオプション

テーブルスペースに作成されたオブジェクトの[デフォルトストレージ特性](#)を設定します。

テーブル圧縮

ドロップダウンリストを使用して、ディスクの使用を軽減するためのデータセグメントの圧縮タイプを選択します。

手動セグメント管理

空きリストを使用して、テーブルスペース内で、セグメントの空きスペースを管理します。

エクステント管理

DICTIONARY	データディクショナリによるエクステント管理です。
LOCAL	ビットマップによるエクステント管理です。

ローカルエクステント

AUTOALLOCATE	テーブルスペースは、システム管理されます。
UNIFORM	テーブルスペースは、均一な領域のサイズで管理されます。

均一サイズ

均一な領域のサイズです。デフォルトサイズは、1 メガバイトです。サイズを指定するためには、ドロップダウンリスト(K(キロバイト), M(メガバイト), G(ギガバイト), T(テラバイト), P(ペタバイト), E(エクサバイト))を使用してください。

詳細プロパティ

ロギング

LOGGING	REDO ログファイルに、テーブルスペース内の全てのオブジェクトの記録をとります。
NOLOGGING	どの操作も記録されません。

強制ロギング

Oracle データベースは、個々のオブジェクトの NOLOGGING 設定を無効にし、一時セグメントに対する変更を除く、テーブルスペースの全てのオブジェクトに対する全ての変更を記録します。

オフライン

テーブルスペースは、作成後すぐに利用不可(オフライン)となります。

保存保証

Oracle データベースは、テーブルスペースの全ての UNDO セグメントにある失効していない UNDO データを保存することによって、それらのセグメント内の UNDO スペースを必要とする進行中操作を強制的に失敗させる場合であっても、保存するべきです。

テーブルスペースグループ

テーブルスペースがテーブルスペースグループのメンバーであるかどうかを決定します。

フラッシュバック

ON	Oracle データベースは、このテーブルスペースに関するフラッシュバックログデータを保存し、テーブルスペースは、FLASHBACK DATABASE の操作に加わることができます。
OFF	Oracle データベースは、このテーブルスペースに関するいかなるフラッシュバックログデータも保存しません。

暗号化を使用

テーブルスペースの暗号化プロパティを有効にします。

アルゴリズム

暗号化アルゴリズムを選択します。

Oracle パブリックデータベースリンク

パブリックデータベースリンクは、DBA によってローカルデータベース上に作成されるデータベースリンクです。そのデータベース上の全ユーザーがアクセスできます。  -> **パブリックデータベースリンク** をクリックして、**パブリックデータベースリンク** 用のオブジェクトリストを開きます。

詳細については、[データベースリンク](#) を参照してください。

Oracle パブリックシノニム

パブリックシノニムは、PUBLIC という特別なユーザーグループに所有されるシノニムで、データベースの全ユーザーがアクセスすることができます。  -> **パブリックシノニム** をクリックして、**パブリックシノニム** 用のオブジェクトリストを開きます。

詳細については、[シノニム](#) を参照してください。

PostgreSQL オブジェクト

サーバーオブジェクトの操作を開始する前に、接続を作成し、接続を開かなければなりません。サーバーが空の場合、接続ペインでその接続を右クリックし、**新規データベース** を選択して、新しいデータベースを作成する必要があります。

基本プロパティ

データベースを作成するためには、**データベースを作成できる** (usecreatedb) 権限が必要です。ユーザープロパティの設定方法については、[ロールデザイナー](#)または[ユーザーデザイナー](#)を参照してください。

データベース名

データベースの名前を定義します。

エンコード

データベースのエンコーディングを定義します。定義しない場合、デフォルトでテンプレートデータベースのエンコーディングになります。

所有者

データベースの所有者を定義します。定義しない場合、デフォルトでコマンドを実行するユーザーになります。スーパーユーザーのみがスーパーユーザー以外のユーザーによって所有されるデータベースを作成することができます。

テンプレート

テンプレートデータベースからデータベースを作成します。

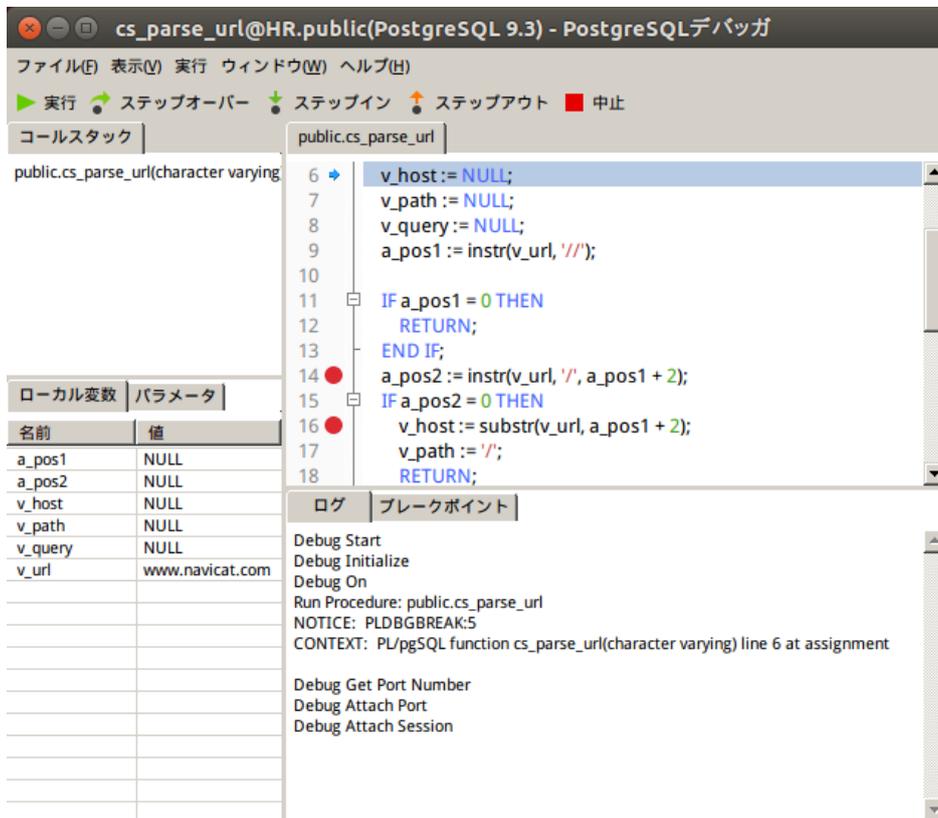
注意: コピー操作中、コピー元のデータベースは、待機状態(進行中でデータ変更不可のトランザクション)でなければなりません。CREATE DATABASE は、操作開始時に、(それ自身以外の)セッションがコピー元のデータベースに接続されていないことを確認します。しかし、これは、コピー処理中に変更が行われないことを保証するものではなく、コピーされたデータベースが不整合となる場合があります。このような理由から、テンプレートとして使用されるデータベースは、読み込みのみ可能な状態で取り扱われることを推奨します。

テーブルスペース

データベースのテーブルスペースを定義します。定義しない場合、デフォルトで pg_default になります。

PostgreSQL デバッガ (フルバージョンで使用可能)

Navicat は、PostgreSQL の PL/pgSQL ファンクションをデバッグするための PostgreSQL デバッガを提供しています。



ツールバーまたはメニューから、デバッグで最もよく使用されるアクションを実行することができます：

ボタン	説明
	デバッグモードでコードの実行を開始します。必要に応じて、入力パラメータを入力します。デバッガは、コードの最後または次のブレークポイントに達するまで、コードを実行します。キーボードショートカット: F9
	実行を再開します。現在行が実行されます。その行がファンクションコールの場合、ファンクションを実行せずに、カウンターはコードの次の行へ移動します。キーボードショートカット: F8
	実行を再開します。現在行が実行されます。その行がファンクションコールの場合、カウンターはそのファンクション内の最初のステートメントへ進みます。そうでない場合、カウンターはコードの次の行へ移動します。キーボードショートカット: F7
	実行を再開します。現在のファンクション内のコードの残りの部分が実行されます。キーボードショートカット: SHIFT+F7
	コードのステップ実行を止めます。実行は止まり、再開することはできません。

コードペインには、ファンクションのコードが表示されます。各行横のグレーのエリアの をクリックすることによって、デバッグ用のブレークポイントを追加/削除することができます。

ブレークポイントタブには、全てのブレークポイントが表示されます。ブレークポイントを右クリックし、**ブレークポイントを削除**または**全てのブレークポイントを削除**を選択することによって、1つまたは全てのブレークポイントを削除することができます。また、ブレークポイントの行にジャンプするためには、それを右クリックし、**ファンクションに進む**を選択します。

コールスタックタブには、現在の行のファンクションコールが表示されます。

結果タブには、ファンクションの実行が完了した後、結果が表示されます。

ログタブには、コードをデバッグする時に、メッセージログが表示されます。

ローカル変数タブには、ファンクションの変数が表示されます。

パラメータタブには、入力されたパラメータが表示されます。

PostgreSQL スキーマ

スキーマは、原則的にネームスペースです：名前がつけられたオブジェクト(テーブル、データタイプ、関数、演算子)を含み、それらの名前は、他のスキーマに存在している他のオブジェクトの名前と重複する可能性があります。スキーマ名は、現在のデータベースにある既存のスキーマ名と異なる名前であればなりません。

基本プロパティ

スキーマ名

作成するスキーマの名前。システムスキーマに用意されている名前のような pg_ で始まる名前をつけることはできません。

所有者

スキーマを所有するユーザーの名前。定義しない場合、デフォルトで、コマンドを実行するユーザーになります。

PostgreSQL テーブル

リレーショナルデータベースは、データを保存するためにテーブルを使用します。全てのデータ操作は、テーブル自身に対して行われるか、結果として別のテーブルを生成します。テーブルは行と列のセットで、その交わる部分がフィールドです。一般的な観点から、テーブル内の列は、その列のフィールドに関して行によって検出されるデータの名前やタイプを表します。テーブル内の行は、対応する列の名前とタイプによって、左から右へ記述されるフィールドから成るレコードを表します。行の各フィールドは、その行の他の各フィールドと暗黙的に関係しています。 をクリックして、**テーブル用オブジェクトリスト**を開きます。

グラフィカルフィールドを含むテーブルを開く場合、テーブルを右クリックし、ポップアップメニューから  **テーブルを開く (高速)** を選択します。セルがクリックされるまで BLOB フィールド(イメージ)はロードされませんので、より高速なパフォーマンスでグラフィカルテーブルを開くことができます。Navicat でテーブルを開く時にすべてのイメージをロードしたい場合、 **テーブルを開く** を使用してください。

新しいテーブルを作成するためには、オブジェクトリストツールバーの  **新規テーブル** をクリックし、同時に  下向き矢印を使用して、タイプ **標準 / 外部** を選択します。

テーブルを空にするためには、選択したテーブルを右クリックし、ポップアップメニューから**テーブルをクリア**を選択します。このオプションは、自動増加倍はリセットせずに既存のすべてのレコードをクリアしたい場合のみ適用されます。テーブルを空にする時に自動増加倍をリセットするためには、**テーブルをトランケート**を使用してください。

PostgreSQL 標準テーブル

PostgreSQL データベースにおいて、テーブルはデータストレージの基本単位です。データは、行と列に格納されます。テーブル名と列を使って、テーブルを定義します。

PostgreSQL テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成や選択したフィールドの削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+Fを押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。 注意: 新規テーブルの作成時のみ使用可能です。
 フィールドを削除	選択したフィールドを削除します。
 プライマリーキー	フィールドをプライマリーキーとして設定し、 プライマリーキーのプロパティ を設定します。
 上に移動/  下に移動	選択したフィールドを上/下に移動します。 注意: 新規テーブルの作成時のみ使用可能です。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならないことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプを定義します。詳細については、[PostgreSQL Data Types](#)を参照してください。

長さ編集ボックスを使用してフィールドの長さを定義します。小数編集ボックスを使用して浮動小数点データタイプの小数点後の桁数(スケール)を定義します。

注意: フィールドの長さを短くする時には、データが失われる可能性があることに注意してください。

ヌルでない

フィールドに NULL の値を許可しない場合、このオプションにチェックをつけます。

プライマリーキー

プライマリーキーは、レコードを一意に定義するための、1つのフィールドまたはフィールドの組み合わせです。プライマリーキーの一部であるフィールドはいずれも、null 値を含むことはできません。

フィールドのポップアップオプション

フィールドを複製	既存のフィールドの1つに変更を加えた新しいフィールドを追加します。
----------	-----------------------------------

プライマリーキーのプロパティ	<p>名前</p> <p>プライマリーキー制約の名前。</p> <p>テーブルスペース</p> <p>プライマリーキー制約のインデックスのテーブルスペース。</p> <p>Fill Factor</p> <p>プライマリーキー制約のインデックスのフィルファクターストレージパラメータ。</p> <p>遅延可能</p> <p>プライマリーキー制約を遅延させることが可能です。 注意: PostgreSQL 9.0 以降でサポートされています。</p> <p>遅延</p> <p>プライマリーキー制約はトランザクションの最後にのみ検査されます。 注意: PostgreSQL 9.0 以降でサポートされています。</p>
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

フォールドのプロパティ

注意: 以下のオプションは選択されたフィールドによって異なります。

デフォルト

フィールドのデフォルト値を設定します。

注釈行

現在のフィールドを説明する任意のテキストを設定します。

照合

列の照合を設定します (照合可能なデータタイプでなければなりません)。指定しない場合、列のデータタイプのデフォルトの照合が使用されます。

注意: PostgreSQL 9.1 以降でサポートされています。

ディメンション

配列指定子のディメンションを設定します。

オブジェクトスキーマ

フィールドのオブジェクトスキーマを設定します。

オブジェクトタイプ

フィールドのオブジェクトタイプを設定します。

PostgreSQL テーブルのインデックス

インデックスは、主に、データベースのパフォーマンスの向上のために使用されます（しかし、不適切に使用すると、パフォーマンスが悪くなる可能性があります）。インデックスのフィールドは、テーブルの行の 1 つ以上の列の値から計算される式です。この機能は、基本データの変形に基づくデータへの高速アクセスを行うために使用されます。

インデックスタブにおいて、編集を行うためにはインデックスフィールドをクリックします。インデックスツールバーを使用して、インデックスフィールドの新規作成や、選択したインデックスフィールドの編集、削除を行うことができます。

ボタン	説明
 インデックスを追加	テーブルにインデックスを追加します。
 インデックスを削除	選択したインデックスを削除します。

名前編集ボックスを使用してインデックス名を設定します。ここにスキーマ名を含むことはできません；インデックスは、その親テーブルと同じスキーマに常に作成されます。

インデックスにフィールドを追加するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用エディタを開きます。

注意: 複数フィールドによるインデックス化を許可しないフィールドタイプもあります。

インデックスメソッドドロップダウンリストで、テーブルインデックスのタイプを定義します。

UNIQUE 制約

インデックスを一意にします。（データが既に存在する状態で）インデックスが作成される時とデータが追加される度に、システムがテーブル内の値の重複を検査します。

クラスタ化

`CLUSTER` は、tablename で指定されたテーブルを indexname で指定されたインデックスに基づいてクラスタ化するように、PostgreSQL に指示します。インデックスは tablename に対して事前に定義されていなければなりません。

テーブルがクラスタ化されると、PostgreSQL はクラスタ化に使用されたインデックスを記憶します。CLUSTER tablename という構文によって、以前にクラスタ化された時と同じインデックスを使用してテーブルを再クラスタ化します。

同時

このオプションを使用すると、PostgreSQL は、テーブルに対する同時挿入、更新、削除を防止するようなロックを獲得せずにインデックスを作成します。通常のインデックス作成処理では、完了するまでテーブルへの書き込みはロックされます（読み取りはロックされません）。

テーブルスペース

インデックスを作成するテーブルスペースです。

Fill Factor

インデックス用のフィルファクター。

バッファリング

インデックスを構築するためにバッファリング構築技術を使用します。

注意: PostgreSQL 9.2 以降でサポートされています。

高速更新

高速更新技法の使用。

注意: PostgreSQL 8.4 以降でサポートされています。

制約

部分インデックスを作成したい場合、この編集ボックスに制約の条件を入力します。部分インデックスは、テーブルの一部分のみに対するエントリーを含むインデックスです。通常、テーブルの一部分とは、テーブルの残りの部分よりも、インデックス化に有用な部分です。

注釈行

インデックスに関する注釈を定義します。

フィールドエディタ

名前リストからフィールドを選択します。また、矢印ボタンを使用して、インデックスフィールドの順番を変更することができます。

照合

インデックスの照合を選択します。

注意: PostgreSQL 9.1 以降でサポートされています。

ソート順

ソート順を指定します - 昇順"ASC"または降順"DESC"。

Null の順番

null が、null ではないものの前(NULLS FIRST)/後(NULLS LAST)にソートされることを指定します。

PostgreSQL テーブルの外部キー

外部キーは、列（または列の集合）の値が別テーブルの行に存在する値と一致しなければならないことを指定します。これは、2つの関連するテーブル間の参照整合性を維持すると言えます。

外部キータブにおいて、編集を行うためには外部キーフィールドをクリックします。外部キーツールバーを使用して、外部キーフィールドの新規作成や、選択した外部キーフィールドの編集、削除を行うことができます。

ボタン	説明
 外部キーを追加	テーブルに外部キーを追加します。
 外部キーを削除	選択した外部キーを削除します。

名前編集ボックスを使用して新しいキーの名前を入力します。

被参照スキーマと被参照テーブルドロップダウンリストを使用して外部スキーマと外部テーブルをそれぞれ選択します。

キーにフィールド/参照されるフィールドを追加するためには、**フィールド/被参照フィールド**フィールドをダブルクリックするか、

 をクリックして、編集用のエディタを開きます。

削除時/更新時ドロップダウンリストで、削除時/更新時に行われる処理の種類を定義します。

RESTRICT	削除または更新が外部キー制約違反を起こす可能性がある場合に、それを表示するエラーを生成します。これは、その確認が遅延不可であること以外は、NO ACTIONと同じです。
NO ACTION	削除または更新が外部キー制約違反を起こす可能性がある場合に、それを表示するエラーを生成します。制約が遅延される場合、制約を確認する時にまだ参照行が存在しているなら、このエラーは生成されます。これがデフォルトの動作です。
CASCADE	削除された行を参照している全ての行を削除するか、または、参照している列の値を参照元の列の新しい値に更新します。
SET NULL	参照している列を null に設定します。
SET DEFAULT	参照している列をデフォルト値に設定します。

完全一致

このオプションにチェックをつけると、全ての外部キー列が NULL となる場合を除き、複数列外部キーのある列が NULL となることを許可しません。

遅延可能

外部キー制約を遅延させることが可能です。

遅延

外部キー制約はトランザクションの最後にのみ検査されます。

注釈行

外部キーに関する注釈を定義します。

関連トピック:

[外部キーデータ選択](#)

PostgreSQL テーブルの UNIQUE 制約

UNIQUE 制約は、列または列の集まりに含まれるデータが、そのテーブルの全ての行に対して一意であることを保証します。

UNIQUE 制約タブにおいて、編集を行うためには UNIQUE 制約フィールドをクリックします。UNIQUE 制約ツールバーを使用して、UNIQUE 制約フィールドの新規作成や、選択した UNIQUE 制約フィールドの編集、削除を行うことができます。

ボタン	説明
 UNIQUE 制約の追加	テーブルに UNIQUE 制約を追加します。
 UNIQUE 制約の削除	選択した UNIQUE 制約を削除します。

名前編集ボックスを使用して UNIQUE 制約名を設定します。

UNIQUE 制約としてフィールドを設定するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

テーブルスペース

UNIQUE 制約のインデックスのテーブルスペース。

Fill Factor

UNIQUE 制約のインデックスのフィルファクターストレージパラメータ。

注意: PostgreSQL 8.2 以降でサポートされています。

遅延可能

UNIQUE 制約を遅延させることが可能です。

遅延

UNIQUE 制約はトランザクションの最後にのみ検査されます。

注釈行

UNIQUE 制約に関する注釈を定義します。

PostgreSQL テーブルのチェック制約

チェック制約は最も汎用的な制約です。特定の列の値が論理(真値)式を満たさなければならないことを指定できます。

チェック制約タブにおいて、編集を行うためにはチェック制約をクリックします。チェック制約ツールバーを使用して、チェック制約の新規作成や、選択したチェック制約の編集、削除を行うことができます。

ボタン	説明
 チェック制約の追加	テーブルにチェック制約を追加します。
 チェック制約の削除	選択したチェック制約を削除します。

名前編集ボックスを使用してチェック制約名を設定します。

チェック

チェック編集ボックスにチェック制約の条件を設定します。例えば、"field_name1 > 0 AND field_name2 > field_name1"。列制約として指定されたチェック制約は、その列の値のみ参照します。一方、テーブル制約の式は、複数の列を参照する可能性があります。

定義

チェック制約の定義を入力します。

継承なし

チェック制約は、子テーブルに反映されません。

注意: PostgreSQL 9.2 以降でサポートされています。

注釈行

チェック制約の注釈を入力します。

PostgreSQL テーブルの除外制約

除外制約は、2つの行が指定した演算子を使用して指定される列または式を比較される場合、その比較のすべてが TRUE を返すわけではないことを保証します。

除外制約タブにおいて、編集を行うためには除外制約フィールドをクリックします。除外制約ツールバーを使用して、除外制約フィールドの新規作成や、選択した除外制約フィールドの編集、削除を行うことができます。

注意: 除外制約は、PostgreSQL 9.0 以降でサポートされています。

ボタン	説明
 除外制約を追加	テーブルに除外制約を追加します。
 除外制約を削除	選択した除外制約を削除します。

名前編集ボックスを使用して、除外制約名を設定します。

インデックスメソッド

使用されるインデックスアクセスメソッドの名前。

要素

除外される要素を選択し、演算子を指定します。

テーブルスペース

除外制約のインデックスのテーブルスペース。

Fill Factor

除外制約のインデックスのフィルファクターストレージパラメータ。

バッファリング

除外制約のインデックスを構築するためにバッファリング構築技術を使用します。

注意: PostgreSQL 9.2 以降でサポートされています。

述語

テーブルの部分集合の除外制約を指定します。

遅延可能

除外制約を遅延させることが可能です。

遅延

除外制約はトランザクションの最後にのみ検査されます。

注釈行

除外制約に関する注釈を定義します。

PostgreSQL テーブルのルール

PostgreSQL のルールシステムでは、データベーステーブルへの挿入、更新、削除時に本来の操作の代替として実行される動作を定義することができます。簡単に言うと、指定されたテーブルに対して指定されたコマンドが実行される時、ルールによって追加のコマンドが実行されるということです。

注意: テーブルの所有者でなければ、テーブルへのルールの作成や変更を行うことはできません。

ルールタブにおいて、編集を行うためにはルールフィールドをクリックします。ルールツールバーを使用して、ルールフィールドの新規作成や、選択したルールフィールドの編集、削除を行うことができます。

ボタン	説明
 ルールの追加	テーブルにルールを追加します。
 ルールの削除	選択したルールを削除します。

名前編集ボックスを使用してルール名を設定します。同じテーブルの他のルール名と別の名前であればなりません。同じテーブルと同じイベントタイプのルールが複数ある場合は、名前のアルファベット順に適用されます。

イベント

イベントは、*SELECT*、*INSERT*、*UPDATE*、*DELETE* のいずれかです。

代わりに実行

これにチェックをつけると、コマンドが元のコマンドの代わりに実行されます。チェックをつけないと、コマンドは元のコマンドに追加されて実行されます。

条件

SQL の条件式(戻り値 boolean)。条件式は、NEW や OLD 以外のテーブルを参照できません。集約関数を含むこともできません。

定義

ルールの動作を構成する 1 つ以上のコマンド。使用可能なコマンドは、*SELECT*、*INSERT*、*UPDATE*、*DELETE* または *NOTIFY* です。

条件やコマンド内では、被参照テーブルの値を参照するために、特別なテーブル名 *NEW* と *OLD* が使用されます。*NEW* は、挿入または更新される新しい行を参照するために、*ON INSERT* と *ON UPDATE* ルールで有効です。*OLD* は、更新または削除される既存の行を参照するために、*ON UPDATE* と *ON DELETE* ルールで有効です。

注釈行

ルールに関する注釈を定義します。

PostgreSQL テーブルのトリガー

トリガーとは、指定された種類の操作が実行された時はいつでもデータベースが特定の関数を自動実行しなければならないという仕様です。トリガーは、*INSERT*、*UPDATE*、*DELETE* 操作の前後のどちらかに実行されるか、変更行に対して 1 回または SQL 文に対して 1 回実行されるように、定義することができます。

トリガータブにおいて、編集を行うためにはトリガーフィールドをクリックします。トリガーツールバーを使用して、トリガーフィールドの新規作成や、選択したトリガーフィールドの編集、削除を行うことができます。

注意: テーブルにトリガーを作成するために、ユーザーはテーブルに対して [TRIGGER](#) 権限がなければなりません。

ボタン	説明
 トリガーの追加	テーブルにトリガーを追加します。
 トリガーの削除	選択したトリガーを削除します。

名前編集ボックスを使用してトリガー名を設定します。同じテーブルの他のトリガー名と別の名前であればなりません。

行トリガー

トリガー処理が、トリガーのイベントによって影響を受けた全ての行に 1 回実行されるべきか、1 つの SQL ステートメントに 1 回実行されるべきかを指定します。チェックがつけられていない場合、*FOR EACH STATEMENT* がデフォルトになります。

起動

トリガーの起動時期を定義します。**Before** または **After** を選択することによって、トリガーを起動させるステートメントの前または後に、トリガーが起動することを表します。

挿入

新しい行がテーブルに挿入される時はいつでも、トリガーが起動されます。

更新

行が修正される時はいつでも、トリガーが起動されます。

削除

行がテーブルから削除される時はいつでも、トリガーが起動されます。

フィールドの更新

列のリストを指定します。トリガーは、リストにあげられた列のうち少なくとも 1 つの列が UPDATE コマンドのターゲットとして挙げられている場合のみ、起動されます。

注意: PostgreSQL 9.1 以降でサポートされています。

条件式

Boolean トリガーが起動されるべきかどうか調べるのに分析される WHEN 条件を指定します。

注意: PostgreSQL 9.0 以降でサポートされています。

トリガー関数のスキーマ と トリガー関数

引数無し、かつ、タイプを返すトリガーとして宣言されているユーザー提供関数で、トリガーの起動時に実行されます。

引数

トリガーが実行される時に関数に与えられる引数の任意のコンマ区切りのリストです。引数は、リテラル文字列定数です。単純な名前や数値の定数をここに記入することも可能ですが、全て文字列に変換されます。トリガーの引数が関数内でどのように使用されるかについては、トリガー関数のインプリメンテーション言語の説明を確認してください; 通常関数の引数とは異なるかもしれません。

注釈行

トリガーに関する注釈を定義します。

制約

制約トリガーを作成します。

遅延可能

トリガー制約を遅延させることが可能です。

遅延

トリガー制約はトランザクションの最後にのみ検査されます。

被参照テーブルスキーマと被参照テーブル名

制約によって参照されるもう 1 つのテーブルのスキーマと名前。

PostgreSQL テーブルのオプション

ロギングしない

ロギングされないテーブルとしてテーブルが作成されます。ロギングされないテーブルに書き込まれるデータは、先行書き込みログに書き込まれます。それは、通常のテーブルよりもかなり高速になります。

注意: PostgreSQL 9.1 以降でサポートされています。

所有者

このテーブルを所有するユーザーを定義します。

テーブルスペース

デフォルトテーブルスペースとは異なる、テーブルを作成するためのテーブルスペースを定義します。

注意: PostgreSQL 8.0 以降でサポートされています。

継承元

このオプションでは、新規テーブルが全ての列を自動的に継承する元のテーブルを一覧で指定します。継承を使用すると、新しい子テーブルとその親テーブルの間には、永続的な関係が作成されます。親テーブルへのスキーマ変更は、通常、同様に子テーブルにも及びます。そして、親テーブルのスキャンに、子テーブルのデータがデフォルトで含まれます。

1 つ以上の既存のテーブルから継承されるように新規テーブルを設定するためには、 をクリックして、編集用のエディタを開きます。

OID を持つ

新規テーブルの行がそれらに割り当てられる OID(オブジェクト識別子)を持つべきかどうかを指定したい場合、このオプションにチェックをつけます。

Fill Factor

テーブルの fillfactor は、10~100%です。100(完全に全て使用)がデフォルトです。小さな値を指定すると、INSERT 操作は指定した割合までしかテーブルページを使用しません。各ページの残りの部分は、そのページ内の行更新用に予約されます。これにより、UPDATE は、元の行と同じページ上に更新済みの行の複製を格納することができるようになります。これは、別のページに格納するよりも効率的です。項目の更新がまったくないテーブルでは、完全に全てを使用することが最善の選択です。しかし、更新が非常に多いテーブルでは、より小さめの fillfactor にすることが適当です。

注意: PostgreSQL 8.2 以降でサポートされています。

PostgreSQL 外部テーブル

外部テーブルは、リモートデータの構造を定義します。外部テーブルは、標準テーブルのようにクエリで使用されることが可能です。しかし、外部テーブルは、PostgreSQL サーバーにストレージを持ちません。外部テーブルが使用されるときはいつも、PostgreSQL は外部データラッパーに、外部ソースからデータを取り出すように、または、更新コマンドの場合は外部ソースにデータを送るように、要求します。

注意: PostgreSQL 9.1 以降でサポートされています。

PostgreSQL 外部テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成や選択したフィールドの削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+Fを押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。 注意: 新規テーブルの作成時のみ使用可能です。
 フィールドを削除	選択したフィールドを削除します。
 上に移動/  下に移動	選択したフィールドを上/下に移動します。 注意: 新規テーブルの作成時のみ使用可能です。

既存フィールドの1つを修正した新しいフィールドを追加するためには、右クリックし、ポップアップメニューから**フィールドを複製**を選択します。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならないことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプを定義します。詳細については、[PostgreSQL Data Types](#)を参照してください。

長さ編集ボックスを使用してフィールドの長さを定義します。**小数**編集ボックスを使用して浮動小数点データタイプの小数点後の桁数(スケール)を定義します。

注意: フィールドの長さを短くする時には、データが失われる可能性があることに注意してください。

ヌルでない

フィールドに NULL の値を許可しない場合、このオプションにチェックをつけます。

フィールドプロパティ

注意: 以下のオプションは、選択したフィールドのタイプによって異なります。

オプション

外部テーブルの列に関連付けられたオプション。許可されるオプション**名前**と**値**は、それぞれの外部データラッパーに固有であり、かつ、外部データラッパーのバリデータファンクションを使用して認証されます。

デフォルト

フィールドのデフォルト値を設定します。

注釈行

現在のフィールドを説明する任意のテキストを設定します。

照合

列(照合可能なデータタイプでなければならない)の照合を設定します。指定しない場合、列のデータタイプのデフォルトの照合が使用されます。

注意: PostgreSQL 9.1 以降でサポートされています。

ディメンション

配列指定子のディメンション j を設定します。

オブジェクトスキーマ

フィールドのオブジェクトスキーマを設定します。

オブジェクトタイプ

フィールドのオブジェクトタイプを設定します。

PostgreSQL 外部テーブルのチェック制約

チェック制約は最も一般的な制約です。特定の列の値が論理(真値)式を満たさなければならないことを指定できます。

チェック制約タブにおいて、編集を行うためにはチェック制約をクリックします。チェック制約ツールバーを使用して、チェック制約の新規作成や、選択したチェック制約の編集、削除を行うことができます。

注意: PostgreSQL 9.5 以降でサポートされています。

ボタン	説明
 チェック制約の追加	テーブルにチェック制約を追加します。
 チェック制約の削除	選択したチェック制約を削除します。

名前編集ボックスを使用してチェック制約名を設定します。

チェック

チェック編集ボックスにチェック制約の条件を設定します。例えば、"field_name1 > 0 AND field_name2 > field_name1"。列制約として指定されたチェック制約は、その列の値のみ参照します。一方、テーブル制約の式は、複数の列を参照する可能性があります。

定義

チェック制約の定義を入力します。

継承なし

チェック制約は、子テーブルに反映されません。

注釈行

チェック制約の注釈を入力します。

PostgreSQL 外部テーブルのトリガー

トリガーとは、指定された種類の操作が実行された時はいつでもデータベースが特定の関数を自動実行しなければならないという仕様です。トリガーは、INSERT、UPDATE、DELETE 操作の前後のどちらかに実行されるか、変更行に対して 1 回または SQL 文に対して 1 回実行されるように、定義することができます。

トリガータブにおいて、編集を行うためにはトリガーフィールドをクリックします。トリガーツールバーを使用して、トリガーフィールドの新規作成や、選択したトリガーフィールドの編集、削除を行うことができます。

注意: PostgreSQL 9.4 以降でサポートされています。

注意: テーブルにトリガーを作成するために、ユーザーはテーブルに対して [TRIGGER](#) 権限がなければなりません。

ボタン	説明
 トリガーの追加	テーブルにトリガーを追加します。
 トリガーの削除	選択したトリガーを削除します。

名前編集ボックスを使用してトリガー名を設定します。同じテーブルの他のトリガー名と別の名前ではなければなりません。

行トリガー

トリガー処理が、トリガーのイベントによって影響を受けた全ての行に 1 回実行されるべきか、1 つの SQL ステートメントに 1 回実行されるべきかを指定します。チェックがつけられていない場合、*FOR EACH STATEMENT* がデフォルトになります。

起動

トリガーの起動時期を定義します。**Before** または **After** を選択することによって、トリガーを起動させるステートメントの前または後に、トリガーが起動することを表します。

挿入

新しい行がテーブルに挿入される時はいつでも、トリガーが起動されます。

更新

行が修正される時はいつでも、トリガーが起動されます。

削除

行がテーブルから削除される時はいつでも、トリガーが起動されます。

フィールドの更新

列のリストを指定します。トリガーは、リストにあげられた列のうち少なくとも 1 つの列が UPDATE コマンドのターゲットとして挙げられている場合のみ、起動されます。

条件式

Boolean トリガーが起動されるべきかどうか調べるのに分析される WHEN 条件を指定します。

トリガー関数のスキーマ と トリガー関数

引数無し、かつ、タイプを返すトリガーとして宣言されているユーザー提供関数で、トリガーの起動時に実行されます。

引数

トリガーが実行される時に関数に与えられる引数の任意のコンマ区切りのリストです。引数は、リテラル文字列定数です。単純な名前や数値の定数をここに記入することも可能ですが、全て文字列に変換されます。トリガーの引数が関数内でどのように使用されるかについては、トリガー関数のインプリメンテーション言語の説明を確認してください; 通常の間数の引数とは異なるかもしれません。

注釈行

トリガーに関する注釈を定義します。

PostgreSQL 外部テーブルのテーブルオプション

外部サーバー

外部テーブル用の既存サーバーの名前。

注意: PostgreSQL 9.1 以降でサポートされています。

オプション

外部テーブルに関連付けられたオプション。許可されるオプション**名前**と**値**は、それぞれの外部データラッパーに固有であり、かつ、外部データラッパーのバリデータファンクションを使用して認証されます。

所有者

このテーブルを所有するユーザーを定義します。

継承元

このオプションは、新しいテーブルが自動的にすべての列を継承する元となるテーブル一覧を指定します。1 つ以上の既存テーブルから継承される新しいテーブルを設定するためには、 をクリックして、編集用のエディタを開きます。

注意: PostgreSQL 9.5 以降でサポートされています。

OID を持つ

新しいテーブルの行がその行に割り当てられる OID(オブジェクト識別子)を持つことを指定したい場合、このオプションにチェックを付けます。

注意: PostgreSQL 9.5 以降でサポートされています。

PostgreSQL ビュー

ユーザーがまるで単一テーブルにアクセスしているかのように複数の関連テーブルにアクセスできるようにしたり、そのアクセスを制限したりするために、ビューが役に立ちます。ビューは、行(特定テーブルの一部)へのアクセスを制限するために使用されることもあります。🔑 をクリックして、**ビュー用のオブジェクトリスト**を開きます。

ボタン	説明
 プレビュー	ビューの結果をプレビューします。
 説明	ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: **表示** -> **結果を表示** -> **クエリエディタの下** または **新規タブ**を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQL の知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

ルール

名前編集ボックスを使用してルールの名前を設定します。

イベント

イベントは、*SELECT*、*INSERT*、*UPDATE*、*DELETE* のうちの 1 つです。

代わりに実行

これは、コマンドが元のコマンドの代わりに実行されるべきであることを示します。無効の場合、コマンドは元のコマンドに追加して実行されます。

条件

(boolean を返す)SQL 条件式。条件式は、NEW と OLD を除くいかなるテーブルも参照できず、集計関数を含むこともできません。

定義

ルールのアクションを構成するコマンド。有効なコマンドは、*SELECT*、*INSERT*、*UPDATE*、*DELETE* または *NOTIFY* です。

条件とコマンドの中では、NEW と OLD という名前の特別なテーブルが被参照テーブルの値を参照するために使用される可能性があります。NEW は、挿入または更新された新しい行を参照するために、ON INSERT と ON UPDATE ルールに有効です。OLD は、更新または削除された既存の行を参照するために、ON UPDATE と ON DELETE ルールに有効です。

注釈行

ルールの注釈を定義します。

詳細プロパティ

所有者

ビューの所有者。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

PostgreSQL 関数(ファンクション)

PostgreSQL は、4 種類の関数を提供しています：

- 問い合わせ言語関数 (SQL で記述された関数)
- 手続き言語関数 (例えば、PL/Tcl や PL/pgSQL で記述された関数)
- 内部関数
- C 言語関数

どの種類の関数も、引数(パラメータ)として、基本型、複合型、または、これらの組み合わせを受け取ります。加えて、どの種類の関数も、基本型または複合型を返します。多くの種類の関数が、擬似データ型(多様型のような)を受け取り、返すことができますが、使用可能な機能は様々です。  をクリックし、**ファンクション用のオブジェクトリスト**を開きます。

PL/pgSQL ファンクションをデバッグするための `pldbgapl` 拡張機能をインストールするためには、オブジェクトリストペインで右クリックし、**pldbgapl 拡張機能をインストール**を選択します。

ファンクションウィザード

オブジェクトリストツールバーから  **新規ファンクション**をクリックします。**ファンクションウィザード**がポップアップ画面で開き、このウィザードを使用して簡単に関数を作成することができます。新規ファンクションを作成する時、**ファンクションウィザード**を表示しないようにもできます。

1. 関数のパラメータを定義します。対応する列に、パラメータの**モード**、**タイプのスキーマ**、**タイプ**、**名前**、**デフォルト値**を設定します。
2. リストから**スキーマ**と**リターンタイプ**を選択します。

ヒント: **次回ウィザードを表示**のチェックを外した場合、[オプション](#)で再度有効にすることができます。

定義

定義は、有効な SQL 手続き文で構成されます。これは、*SELECT* または *INSERT* のような単純な文の場合も、*BEGIN* と *END* を使用して記述された複合文の場合もあります。複合文には、宣言やループ、他の制御構造文を含めることが可能です。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

パラメータ

関数のパラメータを定義します。

リターンタイプのスキーマ と リターンタイプ

関数のリターンタイプを表します。

テーブル列

リターンタイプが TABLE に設定されている場合、テーブル列を入力します。

詳細プロパティ

所有者

関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

言語

関数の実装に使用された言語名。SQL 言語、C 言語、内部言語、または、ユーザー定義の手続き言語名。下位互換性の場合、名前をシングルクォーテーションで囲みます。

変動性

これらの属性は、クエリオプティマイザーに、関数の動作に関する情報を提供します。1 つの選択肢のみ指定できます。指定がない場合、デフォルトで *VOLATILE* と解釈されます。

IMMUTABLE	関数がデータベースを変更できないこと、そして、同じ引数を与えると、常に同じ結果を返すことを表します; つまり、データベースを検索したり、引数リストに直接存在しない情報を使用したりしないということです。このオプションが設定されると、引数が全て定数である関数の呼び出しは、直ちに関数値と置き換えられます。
STABLE	関数がデータベースを変更できないこと、そして、単一テーブルスキャン内では、同じ引数値に対して、常に同じ結果を返しますが、SQL 文が異なると結果が変わる可能性があることを表します。関数の結果がデータベース検索やパラメータ変数(現在のタイムゾーンのような)などに依存する関数については、これを選択することが適当です。また、 <i>current_timestamp</i> 系の関数は、その値がトランザクション内で変化しないので、不変であるとみなされることに注意してください。
VOLATILE	単一テーブルスキャン内で関数値が変化する可能性があること、そのため、最適化できないことを表します。このような意味で、変動しやすいデータベース関数は、比較的少ないです; 例えば、 <i>random()</i> 、 <i>currval()</i> 、 <i>timeofday()</i> です。しかし、副作用のある関数は、その結果を完全に予測できる場合でも、最適化によって呼び出しが取り消されないために、 <i>volatile</i> に分類されなければなら

ないことに注意してください; 例えば、setval()のような関数です。

定義者の安全性

関数を作成したユーザーの権限で、関数が実行されることを指定します。

セットを返す

関数が、1つの項目ではなく、項目のセットを返すことを表します。

Strict

関数は、その引数に null がある時は常に、null を返すことを表します。このパラメータが指定されると、null 引数がある場合、関数は実行されません; 代わりに、結果は null であると自動的にみなされます。

推定コスト

関数の推定実行コストを表す正の数で、単位は `cpu_operator_cost` です。関数がセットを返す場合、この値は、返される行の1行あたりのコストとなります。コストを指定しない場合、C言語関数と内部関数では1、他の全ての言語の関数では100とみなされます。値をより大きくすると、プランナーは必要以上に頻繁に関数を評価しないようになります。

注意: PostgreSQL 8.3以降でサポートされています。

推定行数

プランナーが予測する、関数が返すであろう推定行数を表す正の数。これは、関数がセットを返すと宣言されている場合のみ、許可されます。

注意: PostgreSQL 8.3以降でサポートされています。

設定パラメータ

ここに指定する設定パラメータは、関数が開始する時、指定する値に設定され、関数が終了する時、以前の値に戻されます。

注意: PostgreSQL 8.3以降でサポートされています。

結果

ツールバーの  **実行**をクリックして、関数を実行します。SQL文が正しい場合、その文が実行されます。そして、その文がデータを返すことになっている場合、関数によって返されたデータを表示した**結果**タブが開きます。関数の実行時にエラーが発生した場合、実行は停止し、適切なエラーメッセージが表示されます。関数がパラメータの入力を必要とする場合、**パラメータ入力**ボックスがポップアップ表示されます。パラメータを区切る場合は、`'`を使用します。

ヒント: Navicat は、10個の結果セットの返信に対応しています。

デバッグ (フルバージョンで使用可能)

PL/pgSQL ファンクションをデバッグするためには、ツールバーから  **ファンクションをデバッグ**をクリックし、[PostgreSQL デバッガ](#)を起動します。

PostgreSQL 集約関数

PostgreSQL の集約関数は、状態値と状態遷移関数として表されます。つまり、入力項目が処理される度に修正される状態を用いて、集約関数は定義されます。新しい集約関数を定義するためには、状態値のデータタイプ、状態の初期値と状態遷移関数を選択します。状態遷移関数は一般的な関数で、集約以外でも使用されることがあります。期待される集約の結果が動いている状態値にあるべきデータと異なる場合も、最終関数を指定することが可能です。🏭-> **集約関数**をクリックして、**集約関数用のオブジェクト**リストを開きます。

プロパティ

所有者

集約関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

入力タイプ

この集約関数が演算する入力データタイプ。

注意: PostgreSQL 8.2 以降でサポートされています。8.2 以前のバージョンについては、ドロップダウンリストから**入力タイプ**のスキーマと**入力タイプ**を選択するだけです。

状態タイプのスキーマ と 状態タイプ

集約関数の状態値のデータタイプ。

状態関数のスキーマ と 状態関数

入力行ごとに呼び出される状態遷移関数。引数 N の集約関数については、状態関数は、引数 N+1 を受け取らなければならない、最初が `state_data_type` タイプで、残りは集約関数の指定入力データタイプに一致するものです。関数は、`state_data_type` タイプの値を返さなければなりません。この関数は、現在の状態値と現在の入力データ値を受け取り、次の状態値を返します。

最終関数のスキーマ と 最終関数

全ての入力行が検索された後、集約関数の結果を計算するために呼び出される最終関数。関数は、`state_data_type` タイプの単一引数を受け取らなければなりません。集約関数の戻りデータのタイプは、この関数のリターンタイプとして定義されます。最終関数が指定されない場合、最後の状態値が集約関数の結果として使用され、リターンタイプは `state_data_type` です。

初期状態

状態値の初期設定。これは、`state_data_type` データタイプに受け入れられる形式の文字列定数でなければなりません。指定されない場合、状態値は null から始まります。

ソート演算子のスキーマ と ソート演算子

MIN- や MAX- のような集約関数に関連するソート演算子。演算子は、集約関数(単一引数の集約関数でなければならない)と同じ入力データタイプを持つことを前提とされています。

注意: PostgreSQL 8.1 以降でサポートされています。

PostgreSQL 符号化方式変換

変換は、文字コードエンコーディング間の新しい変換を定義します。変換名は、特定のエンコーディングの変換を指定するために、`convert` 関数で使用することができます。また、DEFAULT として指定される変換は、クライアントとサーバーの間での自動エンコーディング変換に使用することができます。このために、エンコーディング A から B へ、および、エンコーディング B から A へという 2 つの変換が、定義されなければなりません。  -> **変換** をクリックして、**変換用** のオブジェクトリストを開きます。

プロパティ

所有者

変換関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

ソースのエンコーディング

変換元のエンコーディング名。

ターゲットエンコーディング

変換後のエンコーディング名。

ファンクションのスキーマ と ファンクション

変換の実行に使用される関数。関数名は、スキーマ修飾することができます。スキーマ修飾しない場合、関数はパスで検索されます。

関数は、以下のような形式でなければなりません:

```
conv_proc(  
integer, -- source encoding ID  
integer, -- destination encoding ID  
cstring, -- source string (null terminated C string)  
internal, -- destination (fill with a null terminated C string)  
integer -- source string length  
) RETURNS void;
```

デフォルト

元のエンコーディングからターゲットエンコーディングへのこの変換をデフォルトとする場合は、このボックスにチェックをつけます。1 つのスキーマ内では、エンコーディングのペアに対して、デフォルトエンコーディングは 1 つだけです。

PostgreSQL ドメイン

ドメインは、基本的に、任意の制約(設定可能な値の集合に対する制限)を持つデータタイプです。ドメインを定義するユーザーが、その所有者となります。ドメインは、フィールドに対する共通の制約を一箇所にまとめることができ、メンテナンス時に役立ちます。例えば、複数のテーブルが e メールアドレスの列を持ち、その全ての列がアドレス構文検証のための `CHECK` 制約を必要とし

ているような場合です。各テーブルの制約を個々に設定するのではなく、ドメインを定義します。🔧 -> **ドメイン**をクリックして、**ドメイン**用のオブジェクトリストを開きます。

基本プロパティ

基本タイプの分類

基本となるデータタイプの分類を選択します: **Base Type**、**Composite Type**、**Enum Type**、**Domain**。

注意: PostgreSQL 8.2 以降でサポートされています。

基本タイプのスキーマ

基本となるデータタイプのスキーマを選択します。

基本タイプ

ドメインの基本データタイプをドロップダウンリストから選択します。

ディメンション

配列指定子のディメンション。

長さ と スケール

長さ編集ボックスを使用して、フィールドの長さを定義します。**スケール**編集ボックスを使用して、小数点後の桁数を定義します(選択したデータタイプに必要な場合)。

照合

ドメインの照合。

注意: PostgreSQL 9.1 以降でサポートされています。

デフォルト

DEFAULT 句では、ドメインのデータタイプの列に対するデフォルト値を指定します。値は、変数のない式です(サブクエリは許可されません)。デフォルト式のデータタイプは、ドメインのデータタイプと一致していなければなりません。デフォルト値が指定されない場合、デフォルト値は NULL 値となります。

デフォルト式は、列に対して値を指定しない insert 操作に使用されます。特定の列に対してデフォルト値が定義された場合、ドメインに関連するデフォルト値を上書きします。逆に、ドメインのデフォルト値は、基本となるデータタイプに関連するデフォルト値を上書きします。

ヌルでない

このドメインの値は、NULL が許可されていません。

所有者

ドメイン関数の所有者。ドメインを定義するユーザーがその所有者となります。

注意: PostgreSQL 7.4 以降でサポートされています。

チェック制約

チェック制約タブは、ドメインのチェック制約を管理するために提供されています。このタブで、チェック制約を新規作成、選択したチェック制約を編集、削除することができます。

CHECK 句には、ドメインの値が満たさなければならない整合性制約やテストを指定します。各制約は、Boolean 型の結果を生成する式でなければなりません。検査される値を参照するためには、VALUE キーワードを使用してください。

PostgreSQL インデックス

インデックスによって、より高速にテーブルデータにアクセスすることができます。インデックスは、テーブルの 1 つ以上の列を使用して作成され、そのテーブルでの SQL 文の実行スピードを上げます。🏗️ -> **インデックス**をクリックして、**インデックス**用のオブジェクトリストを開きます。

基本プロパティ

UNIQUE 制約

インデックスを一意にします。システムは、インデックスが作成される時(データがすでに存在している場合)またはデータが追加されるごとに、テーブルで値が重複しているかどうか確認します。

テーブル名

インデックス化されるテーブルの名前(スキーマ修飾可能)。

メソッド

インデックスのタイプを定義します。

名前/式

テーブルの列の名前または、テーブルの 1 つ以上の列を基にした式。

照合

インデックスの照合を選択します。

注意: PostgreSQL 9.1 以降でサポートされています。

演算子クラスのスキーマ と 演算子クラス

オペレータークラスのスキーマと名前。

ソート順 (B-Tree インデックスに関してのみ)

ソート順を指定します - 昇順"ASC"または降順"DESC"。

Null の順番 (B-Tree インデックスに関してのみ)

null が、null ではないものの前(NULLS FIRST)/後(NULLS LAST)にソートされることを指定します。

詳細プロパティ

クラスタ化

`CLUSTER` は、tablename で指定されたテーブルを indexname で指定されたインデックスに基づいてクラスタ化するように、PostgreSQL に指示します。インデックスは tablename に対して事前に定義されていなければなりません。

テーブルがクラスタ化されると、PostgreSQL はクラスタ化に使用されたインデックスを記憶します。CLUSTER tablename という構文によって、以前にクラスタ化された時と同じインデックスを使用してテーブルを再クラスタ化します。

同時

このオプションが使用される時、PostgreSQL はテーブルへの同時挿入や更新、削除を防ぐロックを使用せず、インデックスを作成します;一方、標準インデックスの作成は、それが終わるまで、テーブルへの書き込み(読み込みではない)をロックアウトします。

テーブルスペース

インデックスを作成するテーブルスペースです。

Fill Factor (%)

インデックス用のフィルファクターは、インデックスメソッドがインデックスページをどれくらい圧縮しようとするかを決定する割合(パーセント)です。

バッファリング

インデックスを構築するためにバッファリング構築技術を使用します。

注意: PostgreSQL 9.2 以降でサポートされています。

高速更新

この設定は、高速な更新技術の使用を管理します。

注意: PostgreSQL 8.4 以降でサポートされています。

制約

部分インデックスを作成したい場合、この編集ボックスに制約の条件を入力します。部分インデックスは、テーブルの一部分のみに対するエントリーを含むインデックスです。通常、テーブルの一部分とは、テーブルの残りの部分よりも、インデックス化に有用な部分です。

PostgreSQL マテリアライズド・ビュー

マテリアライズド・ビューは、物理的に実体化されたクエリのビューです。クエリは実行され、コマンド発行時にビューにデータを投入するために使用されます。そして、将来、更新されるかもしれません。  -> **マテリアライズド・ビュー**をクリックして、**マテリアライズド・ビュー**用のオブジェクトリストを開きます。

注意: PostgreSQL 9.3 以降でサポートされています。

マテリアライズド・ビューの内容を更新したり、完全に置き換えるためには、オブジェクトリストペインのマテリアライズド・ビューを右クリックし、ポップアップメニューから**マテリアライズドビューを更新** -> **データ** または **データなし**を選択します。

ボタン	説明
 プレビュー	マテリアライズド・ビューの結果をプレビューします。
 説明	マテリアライズド・ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: **表示** -> **結果を表示** -> **クエリエディタの下** または **新規タブ** を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQLの知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

詳細プロパティ

所有者

マテリアライズド・ビューの所有者。

テーブルスペース

新しいマテリアライズド・ビューが作成されるテーブルスペース名。

Fill Factor

ビューの fillfactor は、10~100%です。100(完全に全て使用)がデフォルトです。

データを含む

マテリアライズド・ビューは、作成時にデータを投入されます。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

PostgreSQL 演算子

PostgreSQL は、左単項、右単項、バイナリの演算子をサポートしています。演算子は、オーバーロードされることが可能です。少なくとも *LEFTARG* と *RIGHTARG* のどちらか 1 つを定義する必要があります。バイナリ演算子に対しては、両方とも定義され

なければなりません。右単項演算子に対しては、*LEFTARG* のみを定義し、一方、左単項演算子に対しては、*RIGHTARG* のみを定義する必要があります。🔧 -> **演算子**をクリックして、**演算子用のオブジェクトリスト**を開きます。

注意: *LEFTARG* = 左のタイプ; *RIGHTARG* = 右のタイプ。

基本プロパティ

所有者

演算子関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

左のタイプのスキーマ と 左のタイプ

演算子の左オペランドのデータタイプです（左オペランドがある場合）。このオプションは、左単項演算子に対しては省略されます。

右のタイプのスキーマ と 右のタイプ

演算子の右オペランドのデータタイプです（右オペランドがある場合）。このオプションは、右単項演算子に対しては省略されます。

演算子関数のスキーマ と 演算子関数

この演算子の実行に使用される関数。

詳細プロパティ

制限選択評価関数のスキーマ と 制限選択評価関数

この演算子の制限選択評価関数。

結合選択評価関数のスキーマ と 結合選択評価関数

この演算子の結合選択評価関数。

交代演算子のスキーマ と 交代演算子

この演算子の交代演算子。

否定子のスキーマ と 否定子

この演算子の否定子。

ハッシュ

このオプションにチェックをつけると、演算子はハッシュ結合をサポートできます。

マージ

このオプションにチェックをつけると、演算子はマージ結合をサポートできます。

バージョン 8.3 以前の PostgreSQL に関する追加情報

左のソート演算子のスキーマ と 左のソート演算子

この演算子がマージ結合をサポートできる場合、この演算子の左辺のデータタイプをソートする左ソート演算子。

右のソート演算子のスキーマ と 右のソート演算子

この演算子がマージ結合をサポートできる場合、この演算子の右辺のデータタイプをソートする右ソート演算子。

小なり演算子のスキーマ と 小なり演算子

この演算子がマージ結合をサポートできる場合、この演算子の入力データタイプを比較する小なり演算子。

大なり演算子のスキーマ と 大なり演算子

この演算子がマージ結合をサポートできる場合、この演算子の入力データタイプを比較する大なり演算子。

PostgreSQL 演算子クラス

演算子クラスは、特定のデータタイプがインデックスでどのように使用されるかを定義します。ある演算子がこのデータタイプとインデックスメソッドに対して特定の役割または"戦略"を遂行するというのが、演算子クラスによって指定されます。また、演算子クラスは、演算子クラスがインデックス列に対して選択される時にインデックスメソッドによって使用されるサポートプロシージャも指定します。演算子クラスによって使用される全ての演算子と関数は、演算子クラスが作成される前に定義されなければなりません。  -> **演算子クラス**をクリックして、**演算子クラス**用のオブジェクトリストを開きます。

注意: 同じスキーマ内の2つの演算子クラスは、別のインデックスメソッドに対するものである場合のみ、同じ名前を使用できません。

注釈行タブは PostgreSQL 8.0 以降でサポートされています。

基本プロパティ

所有者

演算子クラス関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

データタイプのスキーマ と データタイプ

この演算子クラスを使用する列のデータタイプ。

インデックスメソッド

この演算子クラスを使用するインデックスメソッドの名前。

ストレージタイプのスキーマ と ストレージタイプ

実際にインデックスに保存されるデータのタイプ。通常、これは列のデータタイプと同じですが、異なるものを許可しているインデックスメソッドもあります(現時点ではGINとGiST)。インデックスメソッドが異なるタイプの使用を許可しない限り、STORAGE句は省略しなければなりません。

演算子族

この演算子クラスを追加する既存の演算子族の名前。指定しない場合、演算子クラスと同じ名前の演算子族が使用されます(そのような演算子族が存在しない場合は、作成されます)。

注意: PostgreSQL 8.3 以降でサポートされています。

デフォルト演算子クラス

このオプションが選択されると、演算子クラスは、そのデータタイプのデフォルトの演算子クラスとなります。指定のデータタイプとインデックスメソッドに対しては、最大でも 1 つの演算子クラスしかデフォルトとすることはできません。

演算子

ストラテジナンバー

演算子クラスに関連する演算子のインデックスメソッドの戦略番号。

演算子のスキーマ と 演算子名

演算子クラスに関連付けられた演算子。

再確認

インデックスがこの演算子に対して"損失が多い"状態であり、したがって、インデックスを使用して抽出される行が実際にこの演算子に関する条件句を満たしていることを検証するために再確認される必要がある場合、このオプションを選択します。

注意: PostgreSQL 8.4 より前のバージョンでは、OPERATOR 句に、RECHECK オプションを追加することが可能でした。インデックス演算子に"損失がある"かどうかは実行時にその場で決定されるようになりましたので、これはもうサポートされていません。これによって、演算子に損失があるかもしれない場合または損失がないかもしれない場合に効果的な処理を行えるようになりました。

ファンクション

サポートナンバー

演算子クラスに関連付けられた関数のインデックスメソッドのサポートプロシージャ番号です。

ファンクションのスキーマ と ファンクション名

演算子クラスに関するインデックスメソッドサポートプロシージャである関数。

PostgreSQL シーケンス

シーケンスは、1 行だけの新しい特別なテーブルを作成し、初期化します。通常は、テーブルの行に対して一意な識別子を生成するために使用されます。  -> **シーケンス**をクリックして、**シーケンス**用のオブジェクトリストを開きます。

基本プロパティ

所有者

シーケンス関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

インクリメント

新しい値を作成するために現在のシーケンス値に追加される値を指定します。正の値は、昇順のシーケンスを作成し、負の値は、降順のシーケンスを作成します。デフォルト値は 1 です。

現在の値

シーケンスの開始値。

最小

シーケンスが生成できる最小値です。

最大

シーケンスの最大値です。

キャッシュ

より高速なアクセスを可能にするために、データベースによってメモリに事前に割り当てられ保存されるシーケンス番号の数を指定します。

循環

このオプションを使用すると、シーケンスは、昇順シーケンスまたは降順シーケンスがそれぞれ最大値または最小値に達した時、値を元に戻すことができます。上下限に達すると、生成される次の番号はそれぞれ最小値または最大値になります。指定しない場合、シーケンスがその最大値に達した後の nextval の呼び出しはエラーを返します。

所有元を追加

テーブルによって所有および**列によって所有**を選択すると、シーケンスは指定されたテーブル列と関連付けられます。その列(またはそのテーブル全体)が削除されると、シーケンスも自動的に削除されます。指定されたテーブルは、シーケンスと同じ所有者であり、同じスキーマになければなりません。

注意: PostgreSQL 8.2 以降でサポートされています。

PostgreSQL トリガー

トリガーとは、指定されたデータベースイベントが発生する時に自動的に実行されるデータベースの操作です。  -> **トリガー** をクリックして、**トリガー用のオブジェクトリスト**を開きます。

詳細については、[トリガー](#)を参照してください。

基本プロパティ

制約

制約トリガーを作成するためには、このボックスにチェックをつけます。

トリガーのタイプ

トリガーの種類を選択します: Table または View。

注意: PostgreSQL 9.0 以降でサポートされています。

テーブル名 または ビュー名

テーブルかビューを選択します。

BEFORE

その操作が行に試行される前に、トリガーが起動するように指定できます。

AFTER

その操作が行に試行された後に、トリガーが起動するように指定できます。

INSTEAD OF

その操作が行に試行される代わりに、トリガーが起動するように指定できます。

INSERT/UPDATE/DELETE

トリガーを起動するイベントを選択します。

INSERT

新しい行が挿入される時はいつも、トリガーが起動されます。

UPDATE

行が修正される時はいつも、トリガーが起動されます。

DELETE

行が削除される時はいつも、トリガーが起動されます。

TRUNCATE

トリガーが TRUNCATE に対して起動するように定義されます。

フィールドの更新

列のリストを指定します。列挙された列の少なくとも 1 つの列が UPDATE コマンドの対象として指定される場合に、トリガーが起動します。

STATEMENT

トリガーのプロシージャは、各 SQL 文につき 1 回起動されることを指定します。

ROW

トリガーのプロシージャは、トリガーイベントによって影響を受けるすべての行に対して 1 回起動されることを指定します。

条件

トリガーが起動されるべきかどうか確認するために検証される Boolean の WHEN 条件を指定します。

注意: PostgreSQL 9.0 以降でサポートされています。

ファンクションのスキーマ と ファンクション名

引数をとらず、トリガータイプを返すと宣言されるユーザー提供関数で、トリガーが起動するときに起動されます。

ファンクションの引数

トリガーが実行されるときにファンクションに提供される、任意のカンマ区切りの引数リスト。引数はリテラル文字列定数です。シンプルな名前と数値定数もここに書き込まれますが、すべて文字列に変換されます。トリガー引数がファンクション内でどのようにアクセス可能かについては、トリガー関数の実装言語の説明を確認してください。それは、標準のファンクションの引数とは異なるかもしれません。

制約

遅延可能

制約は、遅らせることが可能です。

INITIALLY IMMEDIATE

制約は、各ステートメントの後、確認されます。

INITIALLY DEFERRED

制約は、トランザクションの最後に、確認されます。

被参照テーブルスキーマ と 被参照テーブル名

制約によって参照されるもう 1 つのテーブルのスキーマと名前

PostgreSQL トリガー関数

トリガー関数は、PL/pgSQL を使って作成され、PostgreSQL のトリガーの定義内で参照されます。"トリガー関数"とは、トリガーによって起動されることを目的としている関数を指します。トリガーは、指定のイベントがデータベース内で発生する時に実行される操作を定義します。トリガーは、トリガーのイベント発生時に実行される操作として、PL/pgSQL のトリガー関数を参照することができます。

トリガーの定義とそれに関連付けられたトリガー関数の定義は違います。トリガーは、SQL の `CREATE TRIGGER` コマンドを使用して定義されます。一方、トリガー関数は、SQL の `CREATE FUNCTION` コマンドを使用して定義されます。🔧 -> **トリガー関数**をクリックして、**トリガー関数用のオブジェクトリスト**を開きます。

詳細につきましては、[トリガー](#)を参照してください。

定義

定義は、有効な SQL 手続き文で構成されます。これは、`SELECT` または `INSERT` のような単純な文の場合も、`BEGIN` と `END` を使用して記述された複合文の場合もあります。複合文には、宣言やループ、他の制御構造文を含めることが可能です。

パラメータ

トリガー関数のパラメータを定義します。

リターンタイプのスキーマ と リターンタイプ

トリガー関数のリターンタイプ。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)を参照してください。

詳細プロパティ

所有者

トリガー関数の所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

言語

関数の実装に使用された言語名。C 言語、内部言語、または、ユーザー定義の手続き言語名。下位互換性の場合、名前をシングルクォーテーションで囲みます。

変動性

これらの属性は、クエリオプティマイザーに、関数の動作に関する情報を提供します。1 つの選択肢のみ指定できます。指定がない場合、デフォルトで VOLATILE と解釈されます。

IMMUTABLE	関数がデータベースを変更できないこと、そして、同じ引数を与えると、常に同じ結果を返すことを表します。つまり、データベースを検索したり、引数リストに直接存在しない情報を使用したりしないということです。このオプションが設定されると、引数が全て定数である関数の呼び出しは、直ちに関数値と置き換えられます。
STABLE	関数がデータベースを変更できないこと、そして、単一テーブルスキャン内では、同じ引数値に対して、常に同じ結果を返しますが、SQL 文が異なると結果が変わる可能性があることを表します。関数の結果がデータベース検索やパラメータ変数(現在のタイムゾーンのような)などに依存する関数については、これを選択することが適当です。また、current_timestamp 系の関数は、その値がトランザクション内で変化しないので、不変であるとみなされることに注意してください。
VOLATILE	単一テーブルスキャン内で関数値が変化する可能性があること、そのため、最適化できないことを表します。このような意味で、変動しやすいデータベース関数は、比較的少ないです; 例えば、random()、currval()、timeofday()です。しかし、副作用のある関数は、その結果を完全に予測できる場合でも、最適化によって呼び出しが取り消されないために、volatile に分類されなければならないことに注意してください; 例えば、setval()のような関数です。

定義者の安全性

関数を作成したユーザーの権限で、関数が実行されることを指定します。

セットを返す

関数が、1 つの項目ではなく、項目のセットを返すことを表します。

Strict

関数は、その引数に null がある時は常に、null を返すことを表します。このパラメータが指定されると、null 引数がある場合、関数は実行されません; 代わりに、結果は null であると自動的にみなされます。

推定コスト

関数の推定実行コストを表す正の数で、単位は `cpu_operator_cost` です。関数がセットを返す場合、この値は、返される行の 1 行あたりのコストとなります。コストを指定しない場合、C 言語関数と内部関数では 1、他の全ての言語の関数では 100 とみなされます。値をより大きくすると、プランナーは必要以上に頻繁に関数を評価しないようになります。

注意: PostgreSQL 8.2 以降でサポートされています。

推定行数

プランナーが予測する、関数が返すであろう推定行数を表す正の数。これは、関数がセットを返すと宣言されている場合のみ、許可されます。

注意: PostgreSQL 8.2 以降でサポートされています。

設定パラメータ

ここに指定する設定パラメータは、関数が開始される時、指定する値に設定され、関数が終了する時、以前の値に戻されます。

注意: PostgreSQL 8.2 以降でサポートされています。

デバッグ (フルバージョンで使用可能)

PL/pgSQL ファンクションをデバッグするためには、ツールバーの  **ファンクションをデバッグ** をクリックして、[PostgreSQL デバッガ](#) を起動します。

PostgreSQL タイプ

タイプは、現在のデータベースで使用する新しいデータタイプを登録します。スキーマ名が与えられている場合、タイプはその指定されたスキーマに作成されます。スキーマ名が与えられていない場合、現在のスキーマに作成されます。タイプ名は、同じスキーマにある既存のタイプまたはドメインと異なる名前であればなりません。(テーブルはデータタイプと関連しているため、タイプ名は、同じスキーマにある既存のテーブルとも異なる名前であればなりません。)  -> **タイプ** をクリックして、**タイプ** 用のオブジェクトリストを開きます。

基本タイプ は、`int4` のように、SQL 言語レベル以下で実装されるものです(通常、C 言語のような低レベル言語で)。一般的に、抽象データタイプとして知られるものに相当します。PostgreSQL は、ユーザーによって提供された関数を介さなければ、このようなタイプに対して操作を行うことができません。そして、PostgreSQL は、ユーザーが記述する範囲までのみ、このようなタイプの動作を理解します。さらに、基本タイプは、スカラタイプと配列タイプに分類されます。それぞれのスカラタイプに対して、対応する配列タイプが自動的に作成され、配列タイプはスカラタイプの可変長の配列を保持することができます。

複合タイプ または行タイプは、ユーザーがテーブルを作成する時に作成されます; 関連付けられたテーブルがない"スタンドアロン"な複合タイプを定義することも可能です。複合タイプは、関連付けられたフィールド名を持つ基本タイプのリストです。複合タイ

プの値は、行またはフィールド値のレコードです。ユーザーは、SQL クエリからその構成要素フィールドにアクセスすることができます。

列挙(Enum)タイプは、静的で、事前に定義された、特定の順序付きの値の集合です。多くのプログラミング言語の Enum タイプと同じです。Enum タイプの例として、曜日や、一部のデータに関するステータス値の集合があります。

範囲(Range)タイプは、いくつかの要素タイプ (Range のサブタイプと呼ばれる) の値の範囲を表すデータタイプです。

注意: 列挙タイプは、PostgreSQL 8.3 で追加されました。範囲タイプは、PostgreSQL 9.2 で追加されました。

基本タイプの基本プロパティ

入力のスキーマ と 入力

タイプの外部テキスト形式から内部形式へデータを変換する関数。

出力のスキーマ と 出力

タイプの内部形式から外部テキスト形式へデータを変換する関数。

長さ

新規タイプの内部表現の長さをバイトで指定する数値定数。デフォルトで、変数の長さであると解釈されます。

変数

タイプの長さがわからない場合、このオプションを選択します。

デフォルト

データタイプのデフォルト値。これが省略されると、デフォルトは null となります。

要素

作成中のタイプは配列です; これは配列の要素のタイプを指定します。

デリミタ

このタイプで作成される配列で、値と値の間に使用される区切り文字。

整列

データタイプの格納整列条件。指定する場合、char、int2、int4、double のいずれかでなければなりません; デフォルトは int4 です。

ストレージ

データタイプの格納戦略です。指定する場合、plain、external、extended、main のいずれかでなければなりません; デフォルトは plain です。

値渡し

このデータタイプの値が、参照によってではなく、値として渡されなければならないことを示します。

所有者

タイプの所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

基本タイプの詳細プロパティ

詳細タブは、PostgreSQL 7.4 以降でサポートされています。

受信のスキーマ と 受信

タイプの外部バイナリ形式から内部形式へデータを変換する関数。

送信のスキーマ と 送信

タイプの内部形式から外部バイナリ形式へデータを変換する関数。

分析のスキーマ と 分析

データタイプに関する統計分析を実行する関数。

注意: PostgreSQL 8.0 以降でサポートされています。

入カタイプ修飾子のスキーマ と 入カタイプ修飾子

タイプの修飾子の配列を内部形式に変換する関数。

注意: PostgreSQL 8.3 以降でサポートされています。

出カタイプ修飾子のスキーマ と 出カタイプ修飾子

タイプの修飾子の内部形式を外部テキスト形式に変換する関数。

注意: PostgreSQL 8.3 以降でサポートされています。

複合タイプの基本プロパティ

名前

複合タイプの属性名(列名)。

タイプ

複合タイプの列となる既存のデータタイプ名。

長さ と スケール

長さ編集ボックスを使用して、フィールドの長さを定義します。**スケール**編集ボックスを使用して、小数点後の桁数を定義します(選択したデータタイプに必要な場合)。

ディメンション

配列指定子のディメンション。

照合

複合タイプの列と関連付けられる既存の照合の名前。

所有者

タイプの所有者。

注意: PostgreSQL 8.0 以降でサポートされています。

Enum タイプの基本プロパティ

ラベル

Enum タイプの 1 つの値に関連付けられるテキスト形式ラベルを表す文字列リテラル。

所有者

タイプの所有者。

範囲タイプの基本プロパティ

サブタイプスキーマ と サブタイプ

範囲タイプが範囲を表す要素タイプのスキーマと名前。

演算子クラスのスキーマ と 演算子クラス

サブタイプ用の b-tree 演算子クラスのスキーマと名前。

照合

範囲タイプの列と関連付けられる既存の照合の名前。

Canonical スキーマ と Canonical

範囲タイプ用の正規化関数のスキーマと名前。

サブタイプの差異関クションのスキーマ と サブタイプの差異関クション

サブタイプ用の差異関クションのスキーマと名前。

所有者

タイプの所有者。

PostgreSQL テーブルスペース

テーブルスペースを使用して、スーパーユーザーは、(テーブルやインデックスのような)データベースオブジェクトを含むデータファイルが格納されるファイルシステム上の場所を、別途定義することができます。  -> **テーブルスペース**をクリックして、**テーブルスペース**用のオブジェクトリストを開きます。

注意: テーブルスペースは、PostgreSQL 8.0 で追加されました。

注釈行タブは、PostgreSQL 8.2 以降でサポートされています。

基本プロパティ

ロケーション

テーブルスペースのために使用されるディレクトリ。ディレクトリは空でなければならず、PostgreSQL のシステムユーザーによって所有されていなければなりません。ディレクトリは絶対パス名で指定しなければなりません。

所有者

テーブルスペースを所有するユーザーの名前。省略すると、デフォルトでコマンドを実行するユーザーになります。スーパーユーザーのみがテーブルスペースを作成することができますが、スーパーユーザーでないユーザーに対してテーブルスペースの所有権を割り当てることができます。

PostgreSQL キャスト

キャストは、2つのデータタイプ間の変換実行方法を指定します。🔧 -> **キャスト**をクリックして、**キャスト用のオブジェクトリスト**を開きます。

注意: **注釈行タブ**は、PostgreSQL 8.0以降からサポートされています。

基本プロパティ

ソースタイプのスキーマ と ソースのタイプ

キャストの変換元のデータタイプのスキーマと名前。

ターゲットタイプのスキーマ と ターゲットのタイプ

キャストの変換後のデータタイプのスキーマと名前。

ファンクションのスキーマ と ファンクション

キャストの実行に使用される関数。関数名は、スキーマ修飾することができます。スキーマ修飾されない場合、関数はスキーマ検索パスで検索されます。関数の結果データタイプは、キャストの変換先のタイプと一致しなければなりません。

全く関数が指定されない場合、元のタイプとターゲットタイプはバイナリ互換性があるため関数がキャストを実行する必要はないということを表します。

暗黙

キャストが、任意のコンテキストで、暗黙的に呼び出されることを表します。

代入

キャストが、代入コンテキストで、暗黙的に呼び出されることを表します。

PostgreSQL 外部サーバー

外部サーバーは、通常、外部データラッパーが外部のデータリソースにアクセスするために使用する接続情報をカプセル化します。さらに、ユーザーマッピングによりユーザー指定の接続情報が指定される可能性があります。**外部サーバー**のオブジェクトリストを開くためには、🔧 -> **外部サーバー**をクリックします。

注意: PostgreSQL 8.4 以降でサポートされています。

外部 PostgreSQL サーバーに保存されたデータにアクセスするための postgres_fdw 拡張機能をインストールするためには、オブジェクトリストペインで右クリックし、**postgres_fdw 拡張機能をインストール**を選択します。

基本プロパティ

外部データラッパー名

サーバーを管理する外部データラッパーの名前。

オプション

サーバーの接続の詳細を定義します。しかし、実際の名前と値はサーバーの外部データラッパーに依存します。

サーバーのタイプ

サーバーのタイプを指定します。

サーバーバージョン

サーバーのバージョンを指定します。

所有者

外部サーバーの所有者。

ユーザーマッピング

名前

外部サーバーにマップされる既存ユーザーの名前。

オプション

ユーザーマッピングのオプションを指定します。オプションは、通常は、マッピングの実際のユーザー名とパスワードを定義します。

PostgreSQL 言語

言語では、PostgreSQL データベースに新しい手続き言語を登録することができます。登録後、この新しい言語で、関数やトリガープロシージャを定義することができます。新しい言語を登録するために、ユーザーは PostgreSQL のスーパーユーザー権限がなければなりません。  -> **言語**をクリックして、**言語用のオブジェクトリスト**を開きます。

注意: **注釈行**タブは、PostgreSQL 8.0 以降でサポートされています。

基本プロパティ

所有者

言語の所有者。

注意: PostgreSQL 8.3 以降でサポートされています。

ハンドラのスキーマ と ハンドラ

呼び出しハンドラは、事前登録された関数の名前で、手続き言語関数を実行するために呼び出されます。手続き言語の呼び出しハンドラは、Cのようなコンパイルされた言語でバージョン 1 呼び出し規則を使って記述されなければなりません。そして、引数を取らない、*language_handler* タイプを返す関数として、PostgreSQL に登録されなければなりません。*language_handler* タイプとは、単に関数を呼び出しハンドラとして識別するために使用されるプレースホルダータイプです。

バリデータのスキーマ と バリデータ

バリデータ関数(検証用関数)は、事前登録された関数の名前で、新しい関数とその言語を使って作成される時に呼び出され、新しい関数を検証します。検証用関数が指定されない場合、新しい関数は作成時に検証されません。検証用関数は、oid タイプの引数を 1 つ取らなければなりません。oid タイプとは、作成される関数の OID であり、通常は void を返します。

検証用関数は、通常、関数本体の構文上の正確性を調べるものですが、例えば言語が特定の引数のタイプを処理できない場合に、関数の他のプロパティを参照することも可能です。エラーを示すために、検証用関数は ereport()関数を使用すべきです。関数の戻り値は無視されます。

信頼性がある

言語の呼び出しハンドラが安全であることを指定します。つまり、権限のないユーザーにアクセス制限を無視する機能を与えません。言語の登録時にこのキーワードが省略されると、PostgreSQL のスーパーユーザー権限を持つユーザーのみがこの言語を使って新しい関数を作ることができます。

SQLite オブジェクト

サーバーオブジェクトの操作を開始する前に、接続を作成し、接続を開かなければなりません。

データベースを追加するためには、開いた接続を右クリックし、**データベースを追加**を選択し、以下の情報を入力します。

オプション	説明
データベースファイル	データベースへのファイルパスを設定します。
データベース名	Navicat に表示するデータベース名を入力します。
暗号化	暗号化された SQLite データベースに接続する場合は、このオプションを有効にし パスワード を与えます。

データベースを切り離すためには、接続ペインのデータベースを右クリックし、**データベースを切り離す**を選択します。

データベースを暗号化または復号化したい場合、接続ペインのデータベースを右クリックし、**データベースを暗号化** または **データベースを復号**を選択します。

sqlite_master という名前の特別なテーブルが、完全なデータベーススキーマを保存します。sqlite_master テーブルを見るためには、データベースを右クリックし、ポップアップメニューから**マスターテーブルを表示**を選択します。

SQLite テーブル

リレーショナルデータベースは、データを保存するためにテーブルを使用します。全てのデータ操作は、テーブル自身に対して行われるか、結果として別のテーブルを生成します。テーブルは行と列のセットで、その交わる部分がフィールドです。一般的な観点から、テーブル内の列は、その列のフィールドに関して行によって検出されるデータの名前やタイプを表します。テーブル内の行は、対応する列の名前とタイプによって、左から右へ記述されるフィールドから成るレコードを表します。行の各フィールドは、その行の他の各フィールドと暗黙的に関係しています。 をクリックして、**テーブル用オブジェクトリスト**を開きます。

グラフィカルフィールドを含むテーブルを開く場合、テーブルを右クリックし、ポップアップメニューから  **テーブルを開く (高速)** を選択します。セルがクリックされるまで BLOB フィールド(イメージ)はロードされませんので、より高速なパフォーマンスでグラフィカルテーブルを開くことができます。Navicat でテーブルを開く時にすべてのイメージをロードしたい場合、 **テーブルを開く** を使用してください。

テーブルを空にするためには、選択したテーブルを右クリックし、ポップアップメニューから**テーブルをクリア**を選択します。

SQLite テーブルのフィールド

フィールドタブにおいて、編集を行うためにはフィールドをクリックします。右クリックしてポップアップメニューを表示するか、フィールドのツールバーを使用して、フィールドの新規作成、挿入や、選択したフィールドの移動、削除を行うことができます。フィールド名を検索するためには、**編集** -> **検索**を選択するか、CTRL+F を押します。

ボタン	説明
 フィールドを追加	テーブルにフィールドを追加します。
 フィールドを挿入	既存のフィールドの上にフィールドを挿入します。
 フィールドを削除	選択したフィールドを削除します。
 プライマリーキー	フィールドにプライマリーキーを設定します。
 上に移動/  下に移動	選択したフィールドを上/下に移動します。

名前編集ボックスを使用してフィールド名を設定します。フィールド名はテーブルのフィールド名の中で一意でなければならないことに注意してください。

タイプドロップダウンリストを使用してフィールドデータのタイプ(ストレージクラス)を定義します。詳細については、[SQLite 2 Data Types](#) や [SQLite 3 Data Types](#) を参照してください。

長さ編集ボックスを使用してフィールドの長さを定義します。**小数**編集ボックスを使用して浮動小数点データタイプの小数点後の桁数(スケール)を定義します。

ヌルでない

このボックスにチェックをつけると、フィールドに null 値が許可されます。

プライマリーキー

プライマリーキーは、レコードを一意に定義するための単一フィールドまたは複数のフィールドの組み合わせです。プライマリーキーの一部であるフィールドはいずれも、null 値を含むことはできません。

フィールドのプロパティ

注意: 以下のオプションは選択されたフィールドによって異なります。

デフォルト

フィールドのデフォルト値を設定します。

照合

列のテキストエントリーを比較する時に使用するテキスト照合関数を指定します。

BINARY	テキストエンコーディングにかかわらず、memcmp()を使用して文字列データを比較します。
NOCASE	比較実行前に ASCII の 26 個の大文字が小文字と同等に変換されること以外は、binary と同様です。ASCII 文字だけが小文字に変換されることに注意してください。SQLite は、必要とされるテーブルのサイズが原因で、完全な UTF 大文字小文字変換を行いません。
RTRIM	末尾のスペース文字が無視されること以外は、binary と同様です。

注意: SQLite 3 でサポートされています。

Not null ON CONFLICT

Not null オプションにチェックをつけると、制約の衝突を解決するために使用されるアルゴリズムを指定します。デフォルトの衝突解決アルゴリズムは ABORT です。

ROLLBACK	制約違反が発生すると、即時 ROLLBACK が起き、現在のトランザクションを終了させ、そのコマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。(全てのコマンドに作成される暗黙トランザクション以外に)アクティブなトランザクションがない場合、このアルゴリズムは ABORT と同様に機能します。
ABORT	制約違反が発生すると、コマンドは事前に行った変更を全て取り消し、SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、ROLLBACK は実行されないため、同トランザクション内で前のコマンドによって行われた変更は保持されます。これがデフォルトの動作です。
FAIL	制約違反が発生すると、コマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、コマンドが制約違反に衝突する前に行ったデータベースへの変更は保持され取り消されません。例えば、UPDATE 文が更新しようとする 100 行目で制約違反に衝突した場合、最初の 99 行の変更は保持され、100 行目以降は全く変更されません。
IGNORE	制約違反が発生すると、制約違反を含むその 1 行は挿入も変更もされません。しかし、コマンドは普通に実行し続けます。制約違反を含む行の前後の他の行は、通常通り挿入も更新もされ続けます。IGNORE 衝突解決アルゴリズムが使用される場合、エラーは全く返されません。
REPLACE	UNIQUE 制約違反が発生すると、現在行の挿入や更新の前に、制約違反を起こしている既存の行が削除されます。従って、挿入や更新は常に発生します。コマンドは REPLACE に続いて普通に実行し続けます。REPLACE 衝突解決では、エラーは全く返されません。NOT NULL 制約違反が発生すると、NULL 値はその列のデフォルト値で置き換えられます。その列がデフォルト値を持たない場合、ABORT アルゴリズムが使用されます。CHECK 制約違反が発生すると、IGNORE アルゴリズムが使用されます。

自動増加

自動増加属性は、新しい行に一意的固有性を生成するために使用されます。1 以外の値で自動増加値を開始するためには、その値をオプションタブで設定することができます。

SQLite テーブルのインデックス

インデックスは、より早くテーブルデータにアクセスできるようにテーブルデータへのパスを提供します。テーブル上での SQL 文の実行を高速化するために、そのテーブルの 1 つ以上の列を使用して作成されます。

インデックスタブにおいて、編集を行うためにはインデックスフィールドをクリックします。インデックスツールバーを使用して、インデックスフィールドの新規作成や、選択したインデックスフィールドの編集、削除を行うことができます。

ボタン	説明
 インデックスを追加	テーブルにインデックスを追加します。
 インデックスを削除	選択したインデックスを削除します。

名前編集ボックスを使用してインデックス名を設定します。

インデックスにフィールドを追加するためには、**フィールド**をダブルクリックするか、 をクリックして、編集用のエディタを開きます。

UNIQUE 制約

インデックス設定された列の値はすべて、一度しか発生することができません。

フィールドエディタ

名前リストからフィールドを選択します。また、矢印ボタンを使用して、インデックスフィールドの順番を変更することができます。

照合

列のテキストエントリに対して使用される照合シーケンスを定義します。デフォルトの照合シーケンスは、その列に定義された照合シーケンスです。

BINARY	テキストエンコーディングにかかわらず、memcmp()を使用して文字列データを比較します。
NOCASE	比較実行前に ASCII の 26 個の大文字が小文字と同等に変換されること以外は、binary と同様です。ASCII 文字だけが小文字に変換されることに注意してください。SQLite は、必要とされるテーブルのサイズが原因で、完全な UTF 大文字小文字変換を行いません。
RTRIM	末尾のスペース文字が無視されること以外は、binary と同様です。

注意: SQLite 3 でサポートされています。

ソート順

ソート順を示します - 昇順"ASC"または降順"DESC"。

SQLite テーブルの外部キー

外部キーは、別のテーブルのプライマリーキー列と一致するリレーショナルテーブルにあるフィールドです。

外部キータブにおいて、編集を行うためには外部キーフィールドをクリックします。外部キーツールバーを使用して、外部キーフィールドの新規作成や、選択した外部キーフィールドの編集、削除を行うことができます。

ボタン	説明
 外部キーを追加	テーブルに外部キーを追加します。
 外部キーを削除	選択した外部キーを削除します。

名前編集ボックスを使用して新しいキーの名前を入力します。

被参照テーブルドロップダウンリストを使用して外部テーブルを選択します。

キーにフィールド/参照されるフィールドを追加するためには、**フィールド/被参照フィールド**フィールドをダブルクリックするか、



をクリックして、編集用のエディタを開きます。

削除時/更新時ドロップダウンリストで、削除時/更新時に行われる処理の種類を定義します。

RESTRICT	"RESTRICT"アクションは、親キーにマップされている子キーが 1 つ以上存在する時に、アプリケーションがその親キーの削除(削除時 RESTRICT の場合)または更新(更新時 RESTRICT の場合)を禁止されていることを意味します。
NO ACTION	"NO ACTION"を設定するという事は、親キーが更新またはデータベースから削除される時に、特別な処理が行われないことを意味します。
CASCADE	"CASCADE"アクションは、親キーに対する削除または更新処理を個々の子キーに反映します。"削除時 CASCADE"アクションでは、削除された親キーに関連付けられた子テーブルの各行も削除されることを意味します。"更新時 CASCADE"アクションでは、個々の子キーに保存された値が親キーの新しい値に一致するように更新されることを意味します。
SET NULL	設定アクションが"SET NULL"の場合、親キーが削除(削除時 SET NULL の場合)または更新(更新時 SET NULL の場合)される時に、親キーにマップされた子テーブルにある全ての行の子キー列が SQL NULL 値を持つように設定されます。
SET DEFAULT	"SET DEFAULT"アクションは、それぞれの子キー列が NULL の代わりに列のデフォルト値を持つように設定されること以外は、"SET NULL"と同じです。

遅延

遅延外部キー制約は、トランザクションが COMMIT しようとするまで確認されません。

関連トピック:

[外部キーデータ選択](#)

SQLite テーブルの UNIQUE 制約

UNIQUE 制約は、列または列の集まりに含まれるデータが、そのテーブルの全ての行に対して一意であることを保証します。

UNIQUE 制約タブにおいて、編集を行うためには UNIQUE 制約フィールドをクリックします。UNIQUE 制約ツールバーを使用して、UNIQUE 制約フィールドの新規作成や、選択した UNIQUE 制約の編集、削除を行うことができます。

ボタン	説明
 UNIQUE 制約の追加	テーブルに UNIQUE 制約を追加します。
 UNIQUE 制約の削除	選択した UNIQUE 制約を削除します。

名前編集ボックスを使用して UNIQUE 制約名を設定します。

UNIQUE 制約としてフィールドを設定するためには、**フィールド**フィールドをダブルクリックするか、 をクリックして、編集用のエディタを開きます。

ON CONFLICT

制約の衝突を解決するために使用されるアルゴリズムを指定します。

ROLLBACK	制約違反が発生すると、即時 ROLLBACK が起き、現在のトランザクションを終了させ、そのコマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。(全てのコマンドに作成される暗黙トランザクション以外に)アクティブなトランザクションがない場合、このアルゴリズムは ABORT と同様に機能します。
ABORT	制約違反が発生すると、コマンドは事前に行った変更を全て取り消し、SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、ROLLBACK は実行されないため、同トランザクション内で前のコマンドによって行われた変更は保持されます。これがデフォルトの動作です。
FAIL	制約違反が発生すると、コマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、コマンドが制約違反に衝突する前に行ったデータベースへの変更は保持され取り消されません。例えば、UPDATE 文が更新しようとする 100 行目で制約違反に衝突した場合、最初の 99 行の変更は保持され、100 行目以降は全く変更されません。
IGNORE	制約違反が発生すると、制約違反を含むその 1 行は挿入も変更もされません。しかし、コマンドは普通に実行し続けます。制約違反を含む行の前後の他の行は、通常通り挿入も更新もされ続けます。IGNORE 衝突解決アルゴリズムが使用される場合、エラーは全く返されません。
REPLACE	UNIQUE 制約違反が発生すると、現在行の挿入や更新の前に、制約違反を起こしている既存の行が削除されます。従って、挿入や更新は常に発生します。コマンドは REPLACE に続いて普通に実行し続けます。REPLACE 衝突解決では、エラーは全く返されません。NOT NULL 制約違反が発生すると、NULL 値はその列のデフォルト値で置き換えられます。その列がデフォルト値を持たない場合、ABORT アルゴリズムが使用されます。CHECK 制約違反が発生すると、IGNORE アルゴリズムが使用されます。

フィールドエディタ

名前リストからフィールドを選択します。UNIQUE 制約からフィールドを削除するためには、同じ方法でチェックをはずします。また、矢印ボタンを使用して、UNIQUE 制約フィールドの順番を変更することができます。

照合

列のテキストエントリに対して使用される照合シーケンスを定義します。デフォルトの照合シーケンスは、その列に定義された照合シーケンスです。

BINARY	テキストエンコーディングにかかわらず、memcmp()を使用して文字列データを比較します。
NOCASE	比較実行前に ASCII の 26 個の大文字が小文字と同等に変換されること以外は、binary と同様です。ASCII 文字だけが小文字に変換されることに注意してください。SQLite は、必要とされるテーブルのサイズが原因で、完全な UTF 大文字小文字変換を行いません。
RTRIM	末尾のスペース文字が無視されること以外は、binary と同様です。

注意: SQLite 3 でサポートされています。

ソート順

ソート順を示します - 昇順"ASC"または降順"DESC"。

SQLite テーブルのチェック制約

チェック制約を使用して、ある列の値が Boolean(真偽値)式を満たさなければならないことを指定できます。

チェック制約タブにおいて、編集を行うためにはチェックフィールドをクリックします。チェック制約ツールバーを使用して、チェック制約フィールドの新規作成や、選択したチェック制約フィールドの編集、削除を行うことができます。

注意: チェック制約は、SQLite version 3.3.0 以降でサポートされています。

ボタン	説明
 チェック制約の追加	テーブルにチェック制約を追加します。
 チェック制約の削除	選択したチェック制約を削除します。

名前編集ボックスを使用してチェック制約名を設定します。

チェック

チェック編集ボックスにチェック制約の条件を設定します。例えば、"field_name1 > 0 AND field_name2 > field_name1"。

定義

チェック制約の定義を入力します。

SQLite テーブルのトリガー

トリガーは、指定のデータベースイベントが発生する時に自動で実行されるデータベース処理です。

トリガータブにおいて、編集を行うためにはトリガーフィールドをクリックします。トリガーツールバーを使用して、トリガーフィールドの新規作成や、選択したトリガーフィールドの編集、削除を行うことができます。

ボタン	説明
 トリガーの追加	テーブルにトリガーを追加します。
 トリガーの削除	選択したトリガーを削除します。

名前

トリガー名を設定します。

起動

関連行の挿入または修正、削除に対して、トリガーの動作が実行されるタイミングを決定します。

挿入

INSERT 文がテーブルに行を追加する時はいつも、トリガーを起動します。

更新

UPDATE 文が**フィールドの更新**に指定された列の 1 つの列の値を更新する時はいつも、トリガーを起動します。**フィールドの更新**に設定がなければ、UPDATE 文がテーブルの列の値を変更する時は常に、トリガーが起動されます。

削除

DELETE 文がテーブルから行を削除する時はいつも、トリガーを起動します。

フィールドの更新

必要に応じて、UPDATE 文トリガーに対して、フィールドを指定します。

定義

トリガーの定義を入力します。

条件式

トリガーの条件を指定します。これは、トリガーを起動するためにデータベースに対して満たさなければならない SQL 条件です。

SQLite テーブルのオプション

Primary Key ON CONFLICT

プライマリーキー制約の衝突を解決するために使用されるアルゴリズムを指定します。

ROLLBACK	制約違反が発生すると、即時 ROLLBACK が起き、現在のトランザクションを終了させ、そのコマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。(全てのコマンドに作成される暗黙トランザクション以外に)アクティブなトランザクションがない場合、このアルゴリズムは ABORT と同様に機能します。
ABORT	制約違反が発生すると、コマンドは事前に行った変更を全て取り消し、SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、ROLLBACK は実行されないため、同トランザクション内で前のコマンドによって行われた変更は保持されます。これがデフォルトの動作です。

FAIL	制約違反が発生すると、コマンドは SQLITE_CONSTRAINT のリターンコードとともに終了します。しかし、コマンドが制約違反に衝突する前に行ったデータベースへの変更は保持され取り消されません。例えば、UPDATE 文が更新しようとする 100 行目で制約違反に衝突した場合、最初の 99 行の変更は保持され、100 行目以降は全く変更されません。
IGNORE	制約違反が発生すると、制約違反を含むその 1 行は挿入も変更もされません。しかし、コマンドは普通に実行し続けます。制約違反を含む行の前後の他の行は、通常通り挿入も更新もされ続けます。IGNORE 衝突解決アルゴリズムが使用される場合、エラーは全く返されません。
REPLACE	UNIQUE 制約違反が発生すると、現在行の挿入や更新の前に、制約違反を起こしている既存の行が削除されます。従って、挿入や更新は常に発生します。コマンドは REPLACE に続いて普通に実行し続けます。REPLACE 衝突解決では、エラーは全く返されません。NOT NULL 制約違反が発生すると、NULL 値はその列のデフォルト値で置き換えられます。その列がデフォルト値を持たない場合、ABORT アルゴリズムが使用されます。CHECK 制約違反が発生すると、IGNORE アルゴリズムが使用されます。

自動増加

編集フィールドに**自動増加値**を設定/リセットします。自動増加値は、次レコードに対する値を表します。

WITHOUT ROWID

rowid (と"oid"と"_rowid_")を含めません。

注意: SQLite 3.8.2 以降でサポートされています。

SQLite ビュー

ユーザーがまるで単一テーブルにアクセスしているかのように複数のテーブルにアクセスできるようにしたり、そのアクセスを制限したりするために、ビューが役に立ちます。ビューは、行(特定テーブルの一部)へのアクセスを制限するために使用されることもあります。🔍 をクリックして、**ビュー用のオブジェクトリスト**を開きます。

ボタン	説明
 プレビュー	ビューの結果をプレビューします。
 説明	ビューのクエリプランを表示します。
 SQL を整形	エディタで SQL 整形の設定を使用してコードをフォーマットします。

注意: **表示** -> **結果を表示** -> **クエリエディタの下** または **新規タブ**を選択することによって、エディタの下または新しいタブのいずれに結果タブを表示するかを選択することができます。

ビュービルダ (フルバージョンで使用可能)

ビュービルダを使用して、視覚的にビューを作成することができます。SQLの知識がなくてもビューの新規作成や編集を行えます。詳細については、[クエリビルダ](#)をご覧ください。

定義

ビューの定義を SQL ステートメント (実行する SELECT 文) として編集できます。

ヒント: エディタのビューをカスタマイズしたり、SQL 編集用の様々な機能を調べるためには、[エディタの高度機能](#)をご覧ください。

ビュービューワー

ビュービューワーはグリッドでビューデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

SQLite インデックス

インデックスによって、より高速にテーブルデータにアクセスすることができます。インデックスは、テーブルの 1 つ以上の列を使用して作成され、そのテーブルでの SQL 文の実行スピードを上げます。 をクリックして、**インデックス**用のオブジェクトリストを開きます。

基本プロパティ

タイプ

インデックスのタイプ。

Normal	標準インデックスは、列の値に制約を設けません。
Unique	Unique インデックスは、テーブルの 2 つの行がキー列に重複した値を持たないことを表します。

テーブル名

インデックスを含むテーブル。

名前

フィールドを定義します。

照合

列のテキストエントリーに対して使用される照合シーケンスを定義します。デフォルトの照合シーケンスは、その列に定義された照合シーケンスです。

BINARY	テキストエンコーディングにかかわらず、memcmp()を使用して文字列データを比較します。
NOCASE	比較実行前に ASCII の 26 個の大文字が小文字と同等に変換されること以外は、binary と同様です。ASCII 文字だけが小文字に変換されることに注意してください。SQLite は、必要とされるテーブルのサイズが原因で、完全な UTF 大文字小文字変換を行いません。
RTRIM	末尾のスペース文字が無視されること以外は、binary と同様です。

注意: SQLite 3 でサポートされています。

ソート順

ソート順を示します - 昇順"ASC"または降順"DESC"。

SQLite トリガー

トリガーは、指定のデータベースイベントが発生する時に自動で実行されるデータベース処理です。🚩 をクリックして、トリガー用のオブジェクトリストを開きます。

詳細については、[トリガー](#)を参照してください。

基本プロパティ

トリガーのタイプ

トリガーのタイプを定義します: TABLE または VIEW。

テーブル名またはビュー名

テーブルまたはビューを選択します。

BEFORE

トリガーは、操作が行に試行される前に起動されるように指定されます。

AFTER

トリガーは、操作が行に試行された後に起動されるように指定されます。

INSTEAD OF

トリガーは、操作が行に試行される代わりに起動されるように指定されます。

条件

トリガーを起動するためのデータベースに対するトリガー条件を指定します。

挿入

テーブルに行を追加する時はいつも、トリガーが起動されます。

削除

テーブルから行を削除する時はいつも、トリガーが起動されます。

更新

フィールドの更新で選択されたフィールドのうちの 1 つのフィールドの値を更新する時はいつも、トリガーが起動されます。

フィールドの更新

必要に応じて、UPDATE 文トリガーに関するフィールドを指定します。

定義

BEGIN と *END* の中で、トリガー定義の有効な SQL ステートメントを編集することができます。

SQL プレビュー

SQL プレビュータブは、データベースオブジェクトまたはスキーマオブジェクトの CREATE 文と必要な SQL 文を表示します。

いくつかのデータベースオブジェクトまたはスキーマオブジェクトでは、下のドロップダウンリストを使用して、 **保存**または  **名前を付けて保存**を押すと実行される SQL を表示することができます。

保守

Navicat は、MySQL、Oracle、PostgreSQL、SQLite、MariaDB のデータベースやデータベースオブジェクトを保守するための完全なソリューションを提供しています。

サーバーオブジェクトを保守するためには、オブジェクトを右クリックし、ポップアップメニューから**保守**を選択します。

MySQL/MariaDB の保守

テーブルの保守

テーブルの分析

テーブルのキー分布を分析し、保存します。分析中、MyISAM と BDB の場合、テーブルに読み込みロックがかかります。InnoDB の場合、テーブルに書き込みロックがかかります。現在、MySQL は、MyISAM、BDB、InnoDB テーブルに対してのみ、分析をサポートしています。

テーブルの検査

エラーに関して、1 つまたは複数のテーブルを検査します。現在、MySQL は、MyISAM、InnoDB、ARCHIVE テーブルに対してのみ検査をサポートしています。MyISAM テーブルの場合、さらにキーの統計値が更新されます。

標準	追加オプションなしで CHECK TABLE 文を実行します。
簡単	不正リンクをチェックするための行スキャンを実行しない。
高速	適切に閉じられなかったテーブルだけを検査します。
変更された	前回の検査以降に変更されたテーブル、または、適切に閉じられなかったテーブルのみを検査します。
詳細	各行の全てのキーに対して、完全キー参照を行います。これにより、テーブルの一貫性が 100%保証されます。しかし、長時間かかります。

テーブルの最適化

テーブルを最適化する主な理由は、不使用スペースの再利用や、データファイルの最適化です。テーブルの大部分を削除した場合や、可変長の行を持つテーブル (VARCHAR、BLOB、TEXT の列を持つテーブル) に対して多くの変更を行った場合、テーブルを最適化すべきです。削除されたレコードはリンクリストに保持され、その後の INSERT 処理が古い行の場所を再利用します。現在、MySQL は、MyISAM、InnoDB、BDB テーブルに対してのみ最適化をサポートしています。

テーブルの修復

壊れたテーブルをできる限り修復し、結果セットを返します。

簡単	テーブルの修復は、インデックスツリーのみ修復しようとしています。
詳細	MySQL は、ソートごとに 1 つのインデックスを作成する代わりに、各行ごとにインデックスを作成します。

行数の取得

テーブルの行数を数えます。

Oracle の保守

テーブルの保守

テーブルロックを有効にする

テーブルロックを有効にするために、`テーブルロックを有効にする` を選択します。そうすることによって、テーブルへの DDL 操作が許可されます。現在実行中のすべてのトランザクションは、Oracle データベースがテーブルロックを有効にする前に、コミットまたはロールバックしなければなりません。

テーブルロックを無効にする

テーブルロックを無効にするために、`テーブルロックを無効にする` を選択します。そうすることによって、テーブルへの DDL 操作が禁止されます。

行移動を有効にする

データベースが行移動することを許可するために、`行移動を有効にする` を選択します。そうすることによって、RowID が変更されます。

行移動を無効にする

データベースが行移動することを禁止したい場合、`行移動を無効にする` を選択します。そうすると、RowID の変更を避けられます。

領域の圧縮

領域の圧縮 は、テーブルセグメントを圧縮することです。この命令は自動セグメント管理が行なわれているテーブルスペースのセグメントに対してのみ有効です。デフォルトで、Oracle データベースは、セグメントを圧縮し、最高水準に調整し、回復した領域をすぐに開放します。

セグメントの圧縮には、行移動が必要になります。そのため、領域を圧縮する前に、縮小したいテーブルに対して行移動を有効にしなければなりません。さらに、アプリケーションが rowid ベースのトリガーを使っているならば、この命令を発行する前にそれらを無効にすべきです。

移動

移動 は、非パーティションテーブルのデータもしくはパーティションテーブルの 1 つのパーティションにあるデータを新しいセグメントに再割り当てします。オプションとして、別のテーブルスペースのセグメントへの配置や、ストレージ属性の変更を行うこともできます。

統計情報を収集

統計情報を収集 は、テーブルの内容を分析します。テーブルを分析する時、データベースは、さらに、関数ベースのインデックスに存在する式についての統計情報も収集します。したがって、テーブルを分析する前に、必ず、テーブル上に関数ベースのインデックスを作成してください。

構造を確認

構造を確認 は、テーブル構造の整合性を検証します。この命令によって収集された統計情報は、Oracle データベースオペティマイザーによって使用されません。構造に問題がなければ、エラーは返されません。しかし、構造に問題がある場合には、エラーメッセージが表示されます。

Oracle データベースは、テーブルに関して、データブロックと行それぞれの整合性を検証します。

ビューの保守

コンパイル

ビューの仕様またはボディを再コンパイルします。

ファンクション/プロシージャの保守

コンパイル

ファンクション/プロシージャの仕様またはボディを再コンパイルします。

デバッグ用にコンパイル

ファンクション/プロシージャの仕様またはボディを再コンパイルし、そして、PL/SQL デバッガが使用するためにコードを生成し保存するように PL/SQL コンパイラに対して指示します。

インデックスの保守

再構築

既存のインデックス、または、そのパーティションまたはサブパーティションの 1 つを再作成します。インデックスが使用不可の場合、再構築に成功すると、使用可能になります。

使用不可にする

インデックスを使用不可にします。使用不可のインデックスは、使用前に、再構築されるか、削除され再作成される必要があります。

結合

再利用のためにブロックを解放することができるインデックスブロックの内容をマージするように、Oracle データベースに指示します。

統計情報を算出

インデックスの統計情報を算出します。

使用状況を監視する

インデックスの監視を開始します。Oracle データベースは、まずインデックス使用に関する既存情報を消去し、その後、「使用状況を監視しない」を選択するまで、使用に関してインデックスを監視します。

使用状況を監視しない

インデックスの監視を終了します。

Java の保守

Compile / Resolve

主要な Java クラスのスキーマオブジェクトを解決します。

AuthID Current User を設定

AUTHID CURRENT_USER に Invoker 権限を設定します。

AuthID Definer を設定

AUTHID DEFINER に Invoker 権限を設定します。

マテリアライズド・ビューの保守

行移動を有効にする

行移動を有効にします。

圧縮

マテリアライズド・ビューのセグメントを圧縮します。デフォルトで、Oracle データベースは、セグメントを圧縮し、最高水準に調整し、回復した領域をすぐに開放します。

コンパイル

マテリアライズド・ビューを明示的に再度有効にします。マテリアライズド・ビューの依存するオブジェクトが削除または修正された場合、マテリアライズド・ビューはアクセス可能なままですが、クエリの再書き込みは無効になります。このオプションを選択すると、クエリの再書き込みを可能にするために、マテリアライズド・ビューを明示的に再度有効にすることができます。

強制更新

更新を実行します。

マテリアライズド・ビュー・ログの保守

行移動を有効にする

行移動を有効にします。行移動は、フラッシュバック発生後に rowid が変わることを表します。

行移動を無効にする

行移動を無効にします。

領域の圧縮

マテリアライズド・ビュー・ログのセグメントを圧縮します。デフォルトで、Oracle データベースは、セグメントを圧縮し、最高水準に調整し、回復した領域をすぐに開放します。

パッケージの保守

コンパイル

パッケージの仕様またはボディを再コンパイルします。

デバッグ用にコンパイル

パッケージの仕様またはボディを再コンパイルし、そして、PL/SQL デバッガが使用するためにコードを生成し保存するように PL/SQL コンパイラに対して指示します。

トリガーの保守

有効にする

トリガーを有効にします。

無効にする

トリガーを無効にします。

コンパイル

トリガーが有効でも無効でも、明示的にトリガーをコンパイルします。明示的な再コンパイルによって、あいまいなランタイムの再コンパイルの必要がなくなり、関連するランタイムコンパイルエラーやパフォーマンスのオーバーヘッドを防止します。

コンパイル用にデバッグ

トリガーを再コンパイルします。そして、PL/SQL コンパイラに対して、PL/SQL デバッガが使用するコードを生成し保存するように指示します。

タイプの保守

コンパイル

タイプの仕様とボディをコンパイルします。

デバッグ用にコンパイル

タイプの仕様またはボディを再コンパイルし、そして、PL/SQL デバッガが使用するためにコードを生成し保存するように PL/SQL コンパイラに対して指示します。

XML スキーマの保守

コンパイル

すでに登録された XML スキーマを再コンパイルします。スキーマを無効な状態から有効な状態にするのに役立ちます。

削除

Oracle 11g において、Oracle XML データベースから XML スキーマを完全に削除します。

テーブルスペースの保守

読み取り専用

テーブルスペースを推移読み取り専用モードにセットします。この状態で、既存のトランザクションは完了(コミットまたはロールバック)することができますが、いかなる DML 操作もテーブルスペースに対して許可されません。前にテーブルスペースのブロックを変更した既存トランザクションのロールバックは除きます。

読み書き

以前読み取り専用だったテーブルスペースで書き込み操作が可能になることを示します。

オンライン

テーブルスペースをオンラインにします。

オフライン

テーブルスペースをオフラインにします。

標準	システムグローバル領域 (SGA) 外のテーブルスペースにある全データファイルの全ブロックをフラッシュします。
一時性	Oracle データベースは、テーブルスペースの全てのオンラインデータファイルに対してチェックポイントを実行しますが、全てのファイルに必ず書き込まれるわけではありません。
即時	Oracle データベースは、テーブルスペースファイルが使用可能であることを保証せず、チェックポイントを実行しません。

結合

テーブルスペースの各データファイルに関して、全ての隣接する空き領域を、より大きな隣接領域に結合します。

領域の圧縮

テーブルスペースが取っている総領域を削減します。これは、Oracle 11g の一時テーブルスペースに対してのみ有効です。

ユーザーの保守

パスワードを失効

ユーザーアカウントのパスワードが期限切れになるように設定します。

アカウントをロック

ユーザーアカウントをロックします。

アカウントのロック解除

ユーザーアカウントのロックを外します。

PostgreSQL の保守

データベース、テーブル、マテリアライズド・ビューの保守

データベースの分析、テーブルの分析とマテリアライズド・ビューの分析

データベースにあるテーブルの内容について統計を集計し、その結果をシステムテーブル `pg_statistic` に保存します。その後、クエリプランナーがこれらの統計を使用して、もっとも効率的なクエリの実行計画を決定します。**データベースの分析**は、現在のデータベースのすべてのテーブルを調べます。

VERBOSE 指定の場合、分析はどのテーブルが現在処理中であることを示す進行メッセージを出します。同様に、そのテーブルについての様々な統計情報が表示されます。Navicat では、デフォルトで有効になっています。

データベースのバキューム、テーブルのバキュームとマテリアライズド・ビューのバキューム

削除されたタプルによって占有された保存領域を回収します。PostgreSQL の標準動作では、削除されたタプルや更新によって不要となったタプルは、テーブルから物理的に削除されません; それらは バキュームが実行されるまで存在したままです。したがって、特に更新頻度が高いテーブルでは、定期的にバキュームを実行する必要があります。**データベースのバキューム**は、現在のデータベースのすべてのテーブルを調べます。

VERBOSE 指定の場合、バキュームはどのテーブルが現在処理中であることを示す進行メッセージを出します。同様に、そのテーブルについての様々な統計情報が表示されます。Navicat では、デフォルトで有効になっています。

バキューム	追加オプションなしで、VACUUM VERBOSE ステートメントを実行します。
バキューム Analyze	クエリを実行する最も効果的な方法を決定するためにプランナーが使用する統計情報を更新します。
バキューム Full	"完全な"バキュームを選択します。これは、より多くの領域を回収できますが、より長い時間を必要とし、テーブルを排他的にロックします。
バキューム Full Analyze	"完全な"バキュームを選択します。これは、より多くの領域を回収できますが、より長い時間を必要とし、テーブルを排他的にロックします。そして、クエリを実行する最も効果的な方法を決定するためにプランナーが使用する統計情報を更新します。
バキューム Freeze	タプルの積極的な"凍結"を選択します。
バキューム Freeze Analyze	タプルの積極的な"凍結"を選択します。そして、クエリを実行する最も効果的な方法を決定するためにプランナーが使用する統計情報を更新します。

データベースのインデックス再構築、テーブルのインデックス再構築とマテリアライズド・ビューのインデックス再構築

インデックステーブルに保存されたデータを使用して、インデックスを再構築し、インデックスの旧コピーを置き換えます。インデックス再構築は、以下のような様々な状況で使用されます:

- インデックスが破損してしまい、もはや有効なデータを持たない。
- インデックスは"膨張状態"、つまり、空またはほぼ空の多くのページを持っている。
- インデックスのストレージパラメータ(Fill Factor など)を変更し、その変更を確実に有効にしたい。
- CONCURRENTLY オプション付きのインデックス作成が失敗し、"無効な"インデックスが残ってしまった。

SQLite の保守

データベースとテーブルの保守

データベースの分析 と テーブルの分析

テーブルとインデックスに関する統計情報を収集します。そして、クエリオプティマイザーがその収集した情報にアクセスでき、より良いクエリプラン選択を行うためにその情報を使用できる、データベースの内部テーブルにその情報を保存します。

データベースのバキューム

データベース全体を再ビルドします。これはメインのデータベースに対してのみ行われます。追加データベースファイルをバキュームすることは不可能です。

データベースのインデックス再構築 と テーブルのインデックス再構築

データベースのすべてのインデックス または スクラッチからテーブルに追加されたすべてのインデックス を削除し、再作成します。これは、照合シーケンスの定義が変更された時に役立ちます。

インデックスの保守

インデックス再構築

スクラッチからのインデックス を削除し、再作成します。これは、照合シーケンスの定義が変更された時に役立ちます。

テーブルビューワー

テーブルビューワーは、グリッドでテーブルのデータを表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**です。

テーブルビューワーのツールバーから、データを管理するための以下の機能を利用することができます：

- **トランザクション開始/コミット/ロールバック**

 **トランザクション開始**をクリックして、トランザクションを開始します。そのトランザクションで実行された全ての変更を恒久的にするためには、 **コミット**をクリックします。または、 **ロールバック**をクリックして、現在のトランザクションで行われた作業を元に戻します。

ヒント: **コミット**と**ロールバック**ボタンは、[オプション設定](#)で**自動トランザクション開始**が有効になっている場合もしくは**トランザクション開始**ボタンをクリックした後のみ、使用可能です。

- [TEXT/BLOB/BFile の編集](#)

TEXT、BLOB、BFile フィールドの内容を閲覧、編集することができます。

注意: Oracle のみ BFile をサポートしています。

- [データにフィルターをかける](#)

データグリッドにフィルター条件を作成し適用することによって、レコードにフィルターをかけることができます。

- [レコードのソート](#)

ユーザー指定の順番でレコードをソートします。

- [データのインポート](#)

ファイルからデータをインポートします。

- [データのエクスポート](#)

データをファイルにエクスポートします。

グリッドビュー

 グリッドビューでは、テーブル内のデータを表示、更新、追加、削除することができます。グリッドのポップアップメニューから、以下の付加機能を利用することができます：フィールド値にヌル/空文字列を設定する、現在のフィールド値をフィルターとして使用する、グリッドビューの書式設定をするなど。

ナビゲーションバーの使用

テーブルビューワーでは、**レコード/ページナビゲーションバー**ボタンを使用して、レコード/ページの間を簡単に行き来することができます。全てのボタンは、前または次のレコード/ページに、左右に行き来するために使用されます。



レコードナビゲーションバー

ボタン	説明
+	新規レコード: 新しいレコードを入力します。グリッドビューでテーブルを操作するどの時点であっても、このボタンをクリックして、レコード用の空白表示部を出すことができます。
-	レコードの消去: 既存レコードを削除します。
✓	変更の適用: 変更を適用します。
✕	変更のキャンセル: 現在のレコードに加えられた全ての編集を削除します。
↻	更新: テーブルを更新します。
⏹	中止: サーバーから非常に大きいデータを読み込んでいる時に、中止します。

フォームビュー用追加レコードナビゲーションバー

ボタン	説明
⏪	最初のレコード: 最初のレコードに移動します。
⏩	前のレコード: 現在のレコードから 1 つ前のレコードに移動します(1 つ前にレコードがある場合)。
➡	次のレコード: 1 つ先のレコードに移動します。
⏩	最後のレコード: 最後のレコードに移動します。

注意: レコードナビゲーションバーの下に表示される SQL 文は、今実行された SQL 文を表します。

ページナビゲーションバー

ボタン	説明
⏪	最初のページ: 最初のページに移動します。
⏩	前のページ: 前のページに移動します。
➡	次のページ: 次のページに移動します。
⏩	最後のページ: 最後のページに移動します。
⚙️	レコード制限設定: 各ページに表示するレコード数を設定します。
📊	グリッドビュー: グリッドビューに切り替えます。
📄	フォームビュー: フォームビューに切り替えます。

編集モードにするためには、**レコード制限設定** ⚙️ ボタンを使用します。

レコード制限 レコード(1 ページあたり)

各ページに表示されるレコード数を制限したい場合、このオプションにチェックをつけます。チェックをつけない場合、すべてのレコードが 1 ページに表示されます。**レコード(1 ページあたり)**の値を編集フィールドに設定します。各ページに表示されるレコード数を表す数です。

注意: この設定モードは、現在のテーブルのみに対して有効です。グローバル設定への適用については[オプション](#)を参照してください。



レコード a/b(ページ c)

レコード/ページ インジケータは、選択されたレコードやページを表す数を表示します。

- a. 選択されたレコード。
- b. 現在のページにあるレコード数。
- c. 現在のページ。

レコードの編集

ナビゲーションバーを使って、レコードをすばやく入れ替えたり、レコードを挿入、更新、削除することができます。グリッドでのデータ表示は、テーブルへの新規レコードの入力や古いレコードの編集に、大変役立ちます。

レコードを挿入するために

1. カーソルがテーブルの最初の空白セルにあることを確認したら、追加したいデータを入力します。既存のテーブルに新しいレコードを追加する場合、既存のレコードをクリックし、ナビゲーションバーから **+** をクリックするか、CTRL+N を押し、レコード用の空白表示部を出します。
2. レコードの左側のレコード選択ボックスにあるグラフィック記号を見てください。カレントレコードであることを表す  から、変更中であることを表す  に変わります。
3. レコードを保存するためには、別のレコードに移動するか、ナビゲーションバーから **✓** をクリックします。

レコードを編集するために

1. 変更したい特定のフィールドをクリックして、編集したいレコードを選択します。
2. そのフィールドに対して、新しいデータを入力します。
3. 別のレコードに移動すると、新しいデータが前のデータを上書きします。または、ナビゲーションバーから **✓** をクリックします。

注意: 別の方法として、テーブルを閉じてレコードを保存することができます。

同じデータで複数のセルを編集するためには

1. データグリッドでひとまとまりのセルを選択します。
2. 新しいデータを入力します。

注意: 変更は、互換性のあるデータタイプを使って複数のフィールドに適用されます。

レコードを削除するために

1. 削除したいレコードを選択します。
2. 右クリックし、**削除レコード**を選択するか、ナビゲーションバーから **✖** をクリックします。

特別な操作によるレコードの編集

セルに**空文字列**を設定するためには、選択したセルを右クリックし、**空文字列にする**を選択します。

セルに**ヌル値**を設定するためには、選択したセルを右クリックし、**ヌルにする**を選択します。

テキストフィールドのレコードを編集するためには、**表示メニューからグリッドにメモを表示**をクリックします。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

customer_id	store_id	first_name	last_name
1	1	(WIDEMEMO)	SMITH
2	1	(WIDEMEMO)	JOHNSON
3	1	(WIDEMEMO)	WILLIAMS
4	2	(WIDEMEMO)	JONES
5	1	(WIDEMEMO)	BROWN

ヒント: テキストフィールドのレコードを簡単に表示/編集する方法については、[メモエディタ](#)を参照してください。

グリッドに画像を表示するためには、**表示メニューからグリッドにイメージを表示**をクリックします。

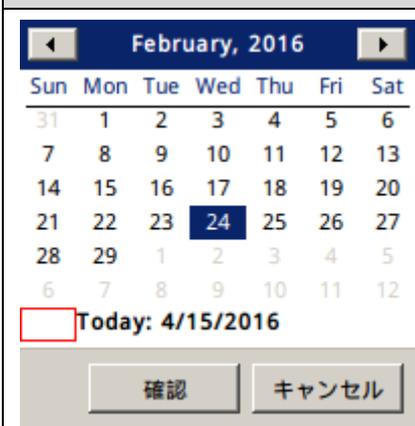
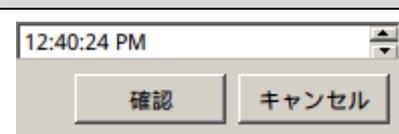
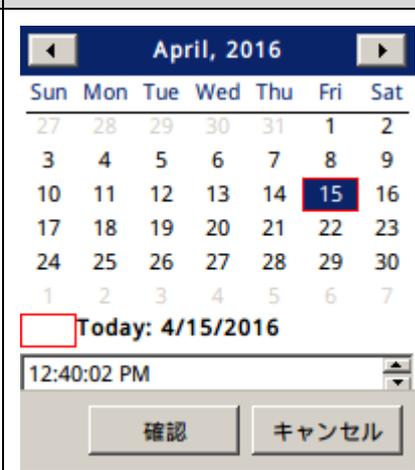
注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

id	image
1	
2	
3	

ヒント: 画像を簡単に表示/編集する方法については、[イメージエディタ](#)を参照してください。

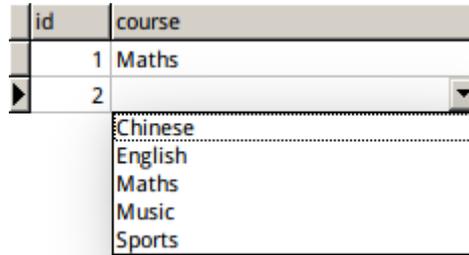
日付/時刻レコードを編集するためには、 をクリックするか、CTRL+ENTER を押して、編集用エディタを開きます。編集したいデータを選択/入力します。セルで使用されるエディタは、列に割り当てられたフィールドタイプによって決定されます。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

Date	Time	DateTime/Timestamp
		

Enum レコードを編集するためには、ドロップダウンリストからレコードを選択します。

注意: MySQL、PostgreSQL、MariaDB で使用可能。



id	course
1	Maths
2	<ul style="list-style-type: none">ChineseEnglishMathsMusicSports

Set レコードを編集するためには、 をクリックするか、CTRL+ENTER を押して、編集用エディタを開きます。リストからレコードを選択します。レコードを削除するためには、同様にチェックを外します。

注意: MySQL、MariaDB で使用可能。



id	course
1	English, Maths, Music, Sports
2	Apply Maths, Sports 

- Chinese
- English
- Maths
- Music
- Apply Maths
- Sports

確認 キャンセル

BFile の内容を表示するためには、表示メニューから **BFile をプレビュー** を有効にします。

注意: Oracle で使用可能。

UUID/GUID を生成するためには、選択したセルを右クリックし、**UUID を生成** を選択します。

注意: PostgreSQL で使用可能。

外部キーを持つレコードの編集 (外部キーのデータ選択 - フルバージョンで使用可能)

外部キーデータの選択は、選択可能な値を参照テーブルから簡単に取得するために役立つツールです。参照テーブルから追加レコードを表示したり、特別なレコードを探したりすることができます。

レコードにデータを追加するためには、 をクリックするか、CTRL+ENTER を押して、編集用エディタを開きます。

payment_id	customer_id	staff_id	rental_id	amount	payment_date	last_update
1	1	1	76	2.99	2005-05-25 11:30:37	2006-02-15 22:12:30
2	3	...	1	573	0.99	2005-05-28 10:35:23
3	fk_payment_customer : payment (customer_id) - customer (customer_id)					6-03-31 13:12:33
4	customer_id	store_id	first_name	last_name		6-03-31 13:09:30
5						6-02-15 22:12:30
6	1		1 MARY	SMITH		6-02-15 22:12:30
7	2		1 PATRICIA	JOHNSON		6-02-15 22:12:30
8	3		1 LINDA	WILLIAMS		5-01-12 17:15:18
9	4		2 BARBARA	JONES		6-02-15 22:12:30
10	5		1 ELIZABETH	BROWN		6-02-15 22:12:30
11	6		2 JENNIFER	DAVIS		6-02-15 22:12:30
12	7		1 MARIA	MILLER		6-02-15 22:12:30
13	8		2 SUSAN	WILSON		6-02-15 22:12:30
14	9		2 MARGARET	MOORE		6-02-15 22:12:30
15						6-02-15 22:12:30
16						6-02-15 22:12:30
17						6-02-15 22:12:30
18						6-02-15 22:12:30

ダブルクリックして、編集したいデータを選択します。

ヒント デフォルトでは、1 ページに表示されるレコード数は、**100** です。全てのレコードを表示するためには、グリッド上で右クリックし、**全て表示**を選択します。グローバル設定への適用については[オプション](#)を参照してください。

レコードを更新するためには、グリッド上で右クリックし、**更新**を選択します。または、F5 を押します。

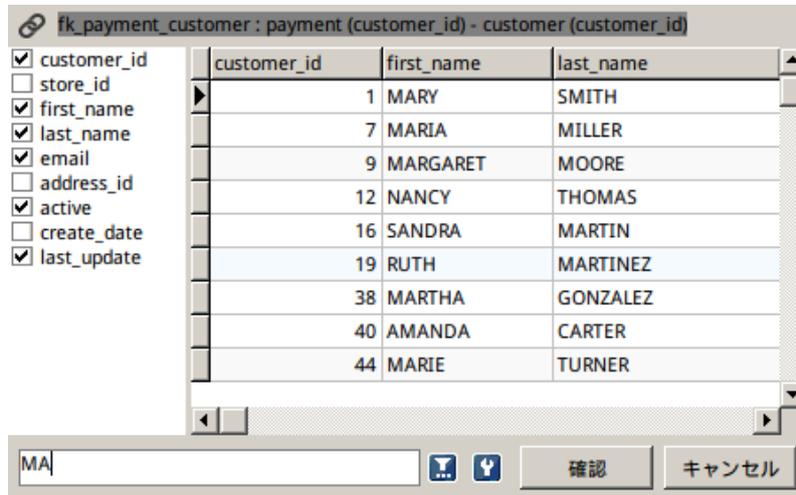
 をクリックすると、左側に列名一覧が表示されます。追加する列をクリックすると表示されます。同様にしてチェックを外すとその列は除かれます。

fk_payment_customer : payment (customer_id) - customer (customer_id)			
<input checked="" type="checkbox"/> customer_id	customer_id	first_name	last_name
<input type="checkbox"/> store_id			
<input checked="" type="checkbox"/> first_name	1	MARY	SMITH
<input checked="" type="checkbox"/> last_name	2	PATRICIA	JOHNSON
<input checked="" type="checkbox"/> email	3	LINDA	WILLIAMS
<input type="checkbox"/> address_id	4	BARBARA	JONES
<input checked="" type="checkbox"/> active	5	ELIZABETH	BROWN
<input type="checkbox"/> create_date	6	JENNIFER	DAVIS
<input checked="" type="checkbox"/> last_update	7	MARIA	MILLER
	8	SUSAN	WILSON
	9	MARGARET	MOORE

ヒント 列に昇順または降順モードを設定するためには、列のどこかで右クリックし、**ソート** ->  昇順ソート /  降順ソート を選択します。

エディタウィンドウでテキストを検索するためには、グリッド上で右クリックし、**検索**を選択するか、CTRL+F を押します。

特定のレコードにフィルターをかけるには、編集ボックスに値を入力して  をクリックします。



ヒント: フィルター結果を削除するためには、グリッド上で右クリックし、**全て表示**を選択します。

Navicat からのデータのコピー

Navicat からコピーされたデータは、タブによって区切られたフィールドとキャリッジリターンによって区切られたレコードとして、Windows のクリップボードに入ります。これによって、希望のアプリケーションにクリップボードの内容を簡単に貼り付けることができます。一般的なスプレッドシートアプリケーションは、フィールド間のタブ文字を認識し、クリップボードのデータを行と列にきちんと分けます。

キーボードのショートカットでデータを選択するために

CTRL+A	データグリッドにある全ての行と列の選択を切り替えます。
SHIFT+ARROW	データグリッドで上/下/左/右に移動しながら、セルの選択を切り替えます。

マウス操作でデータを選択するために

- CTRL キーを押したまま、それぞれの行をクリックすることによって、希望のレコードを選択します。
- ひとまとまりのセルを選択します。

注意: 希望のレコードを選択した後、CTRL+C を押すか、右クリックしてポップアップメニューから  **コピー**を選択します。

Navicat のデータの貼り付け

クリップボードにコピーされたデータは、以下の形式で配置されます:

- データは行と列に配置されます。
- 行と列は、それぞれ、キャリッジリターン/タブによって区切られます。
- クリップボードの列は、選択したデータグリッドの列と同じ順番になります。

Navicat のデータ貼り付けでは、現在のレコードの内容を置き換えたり、クリップボードのデータをテーブルに追加することができます。テーブルの現在のレコードの内容を置き換えるためには、クリップボードのデータに置き換えられるべき内容のセルをデータグリッドで選択しなければなりません。CTRL+V を押すか、右クリックしてポップアップメニューから  **貼り付け**を選択

します。Navicat は、クリップボードのすべての内容を選択されたセルに貼り付けます。トランザクションを有効にしていない場合、貼り付けは必ず実行されます。

Insert/Update ステートメントとしてレコードをコピー

Insert/Update ステートメントとしてレコードをコピーするためには、レコードを選択し、メニューから**編集 -> 以下としてコピー -> Insert ステートメント** または **Update ステートメント** をクリックします。その後、どのエディタにでもステートメントを貼り付けることができます。

フィールド名をコピー

タブで区切られた値としてフィールド名をコピーするためには、**コラム/データ**を選択し、メニューから**編集 -> 以下としてコピー -> タブ区切り (フィールド名のみ)** を選択します。データのみまたはフィールド名とデータの両方をコピーしたい場合、**タブ区切り (データのみ)** または **タブ区切り (フィールド名とデータ)** を選択します。

ファイルとしてデータをコピー

テーブルグリッドのデータをファイルに保存することができます。セルを右クリックして、**名前を付けてデータを保存**を選択します。名前を付けて保存ダイアログウィンドウに、ファイル名とファイルの拡張子を入力します。

注意: 複数選択時は使用できません。

レコードのソート/検索/置換

レコードのソート

サーバーは、テーブルに追加された順番にレコードを保存します。Navicat のソートを使用して、レコードを一時的に並び替え、違う順番でレコードを表示または更新することができます。

ソートしたい列の見出しに移動して、その列の右の方をクリックし、**昇順ソート**、**降順ソート** または **ソートを解除** を選択します。

payment_id	customer_id	staff_id	rental_id	amount
1		昇順ソート(X)	76	2.99
2		降順ソート(Y)	573	0.99
3		ソートを解除(Z)	185	5.99
4			422	0.99
5	1	2	1476	9.99

複数フィールドを希望の順序にソートするためには、ツールバーから  **ソート** をクリックします。

レコードの検索

検索ダイアログは、エディタウィンドウにおけるテキストのクイック検索のために提供されています。メニューから、**編集 ->  検索** をクリックするか、CTRL+F を押します。そして、**データの検索** を選択し、検索文字列を入力します。

customer_id	store_id	first_name	last_name	email	address_id
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer	5
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer	6
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer	7
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer	8
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer	9
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer	10
7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer	11
8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer	12
9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer	13
10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer	14
11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer	15
12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer	16
13	2	KAREN	JACKSON	KAREN.JACKSON@sakilacustomer	17
14	2	BETTY	WHITE	BETTY.WHITE@sakilacustomer	18

データの検索: MA 次 すべてを強調 置換

SELECT * FROM `customer` LIMIT 0, 1000 レコード 1 / 599 (ページ1)

検索は、カーソルの現在の位置から始まり、ファイルの最後で終わります。大文字/小文字のどちらで検索しても、違いはありません。

次のテキストを検索するためには、**次**をクリックするか、F3を押します。

レコードの置換

置換ダイアログを開くためには、**置換**ボックスにチェックを付け、検索したいテキストと置換したいテキストを入力します。

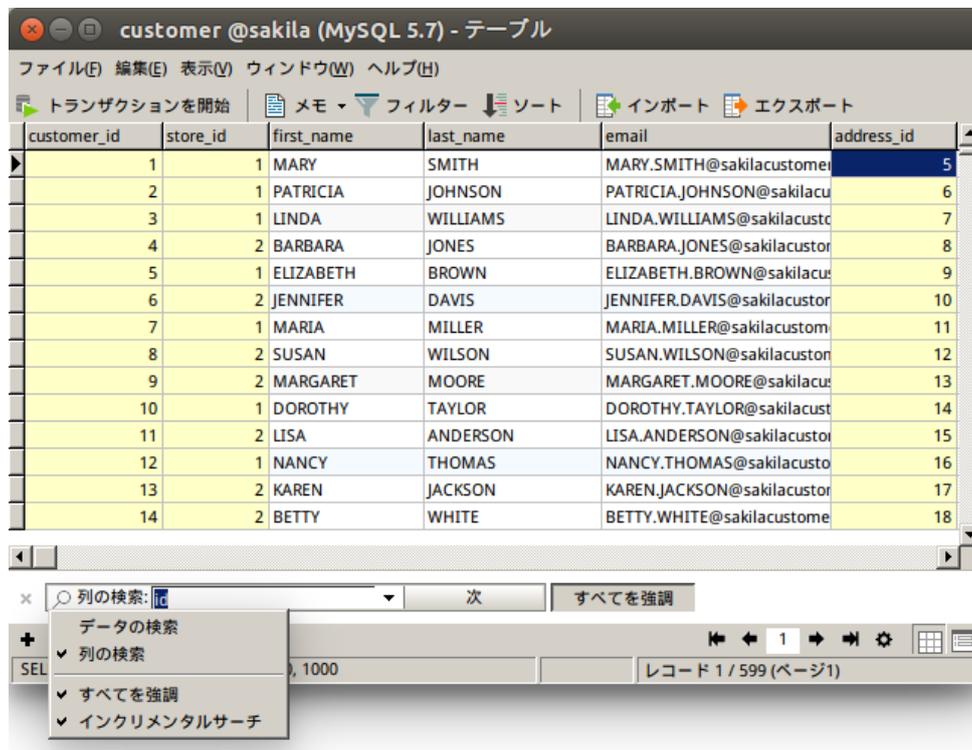
置換 または **全て置換** ボタンをクリックすることによって、最初の置換対象または全ての置換対象を置換することができます。

全て置換ボタンをクリックする場合、**適用**ボタンをクリックして変更を適用するか、**キャンセル**ボタンをクリックして変更をキャンセルすることができます。



列の検索

列を検索するためには、メニューから、**編集** -> **検索**をクリックするか、CTRL+Fを押します。そして、**列の検索**を選択し、検索文字列を入力します。



レコードにフィルターをかける

以下のいずれかの方法を使い、グリッドのデータにフィルターをかけることができます：

- 選択した列の現在の値を用いてレコードにフィルターをかけるためには、フィールドを右クリックし、ポップアップメニューから**フィルター**を選択します。
- 単純なフィルターを素早く作成するために、**カスタムフィルター**ダイアログが提供されています。フィールドを右クリックし、ポップアップメニューから**フィルター -> カスタムフィルター**を選択します。条件内で任意の一文字を表すためには、文字 '_' を使用します。条件内で任意の文字列を表すためには、文字 '%' を使用します。
- フィルターをより複雑にカスタマイズすることも可能です。フィールドを右クリックし、ポップアップメニューから**フィルター -> フィルター**を選択するか、ツールバーから **フィルター** をクリックします。フィルターウィザードがグリッドの一番上に表示され、現在有効なフィルターの条件を見たり、左側のチェックボックスをクリックすることで簡単に有効/無効にすることができます。

Raw データの操作

Navicat は、通常、ユーザーがグリッドに入力したものを普通の文字列とみなし、特別な文字または関数は単純なテキストとして処理されます(つまり、その機能性は省かれます)。

Raw モードでのデータ編集を使用すると、サーバー組み込み関数を簡単に直接適用することができます。Raw モード機能にアクセスするためには、メニューから**表示 -> Raw モード**を選択します。

注意: MySQL、PostgreSQL、SQLite、MariaDB で使用可能

customer_id	store_id	first_name	last_name
'1'	'1'	'MARY'	'SMITH'
'2'	'1'	'PATRICIA'	'JOHNSON'
'3'	'1'	'LINDA'	'WILLIAMS'
'4'	'2'	CONCAT('BARBARA', ' ', 'A')	'JONES'
'5'	'1'	'ELIZABETH'	'BROWN'

テーブルグリッドの書式設定

以下の方法を使用して、テーブルグリッドを書式設定します：

列の移動

1. 列のヘッダーをクリックし、マウスの左ボタンを押した状態にします。
2. 2 倍の太さの黒線が希望の場所に表示されるまで、ポインターを移動します。
3. マウスを放すと、列が移動します。

customer_id	store_id	first_name	last_name
1	1	MARY	SMITH
2	1	PATRICIA	JOHNSON
3	1	LINDA	WILLIAMS
4	2	BARBARA	JONES
5	1	ELIZABETH	BROWN
6	2	JENNIFER	DAVIS
7	1	MARIA	MILLER

選択した列を凍結する

テーブルにたくさんの列があり、1 つ以上の列を凍結してレコードを特定したい場合、凍結したい列を右クリックし、**表示 -> 選択中の列を凍結する**を選択するか、メニューから**表示 -> 選択中の列を凍結する**を選択します。

凍結された列は、テーブルグリッドの左端に移動します。この操作によって、凍結された列はロックされ、編集されなくなります。

列を解凍するためには、テーブルグリッド上で右クリックし、**表示 -> 列を非凍結**を選択するか、メニューから**表示 -> 列を非凍結**を選択します。

表示書式を設定

表示書式の設定ダイアログは、選択した列に出力されるデータに適用される書式をカスタマイズするために提供されています。書式を編集したい列を右クリックし、**表示 -> 表示書式を設定**を選択するか、メニューから**表示 -> 表示書式を設定**を選択します。書式のスタイルを編集して、結果書式を必要な書式に変更します。例: dd-mm-yyyy。

ヒント: この動作は、選択中の列だけに適用されます。グローバル設定への適用については[オプション](#)を参照してください。

列の幅を設定

列の一番上で右端をクリックし、左か右にドラッグします。

その列にとって最適な幅にするためには、列の一番上で右端をダブルクリックします。

列の幅を設定したい列を右クリックし、**表示 -> 列幅を設定**を選択するか、メニューから**表示 -> 列幅を設定**を選択します。**列幅を設定**ダイアログで幅を指定します。デフォルト値は 120 です。

ヒント: 結果は、選択中の列だけに適用されます。グローバル設定への適用については[オプション](#)を参照してください。

行の高さを設定

テーブルグリッド上で右クリックし、**表示 -> 行の高さを設定**を選択するか、メニューから**表示 -> 行の高さを設定**を選択します。**行の高さを設定**ダイアログで行の高さを指定します。デフォルト値は 17 です。

ヒント: この動作は、現在のテーブルグリッドだけに適用されます。グローバル設定への適用については[オプション](#)を参照してください。

列を表示/隠す

テーブルにたくさんの列があり、そのいくつかをテーブルグリッドから隠したい場合、テーブルグリッドで右クリックし、**表示 -> 列を表示/隠す**を選択するか、メニューから**表示 -> 列を表示/隠す**を選択します。隠したい列を選択します。

隠された列は、テーブルグリッドから消えます。

隠された列を表示するためには、テーブルグリッドで右クリックし、**表示 -> 列を表示/隠す**を選択するか、メニューから**表示 -> 列を表示/隠す**を選択します。再表示したい列を選択します。

<input checked="" type="checkbox"/> customer_id	customer_id	store_id	last_name	address_id
<input checked="" type="checkbox"/> store_id	1	1	SMITH	5
<input type="checkbox"/> first_name	2	1	JOHNSON	6
<input checked="" type="checkbox"/> last_name	3	1	WILLIAMS	7
<input type="checkbox"/> email	4	2	JONES	8
<input checked="" type="checkbox"/> address_id	5	1	BROWN	9
<input checked="" type="checkbox"/> active	6	2	DAVIS	10
<input checked="" type="checkbox"/> create_date				
<input checked="" type="checkbox"/> last_update				

ROWID を表示/隠す

全行の rowid(アドレス)を表示または隠したい場合には、テーブルグリッドで右クリックし、**表示 -> ROWID を表示/隠す**を選択するか、メニューから**表示 -> ROWID を表示/隠す**を選択します。

ROWID 列は最後の列に表示されます。

注意: Oracle、SQLite で使用可能。

フォームビュー(フルバージョンで使用可能)

 フォームビューでは、現在のレコードのフィールド名とその値が表示されており、記入用紙のようにデータを表示、更新、挿入、削除することができます。フォームのポップアップメニューは、以下の付加機能を提供しています：フィールド値をヌル/空文字に設定、現在のフィールド値をフィルターとして使用、フォームビューの書式設定など。

ナビゲーションバーを使用して、レコードをすばやく入れ替えたり、レコードを挿入、更新、削除することができます。

関連トピック:

[レコードのソート/検索/置換](#)

[レコードにフィルターをかける](#)

[Raw データの操作](#)

[テーブルグリッドの書式設定](#)

補助エディタ

Navicat は、TEXT/BLOB/BFile フィールドの内容を表示、編集するために、メモ/16 進数/イメージ/ウェブペインを提供しています。エディタを使用して、テーブルにあるデータを表示、更新、挿入、削除することができます。ツールバーから  **メモ**、 **16 進数**、 **イメージ**、 **動的カラム**をクリックして、適当なビューワー/エディタを開くことができます。

注意: Oracle BFile フィールドは編集できません。

メモペインでは、シンプルなテキストとしてデータを編集することができます。ナビゲーションバーの  ボタンを使用して、変更されたレコードをテーブルに更新します。

16 進数ペインでは、16 進モードでデータを編集することができます。ナビゲーションバーの  ボタンを使用して、変更されたレコードをテーブルに更新します。

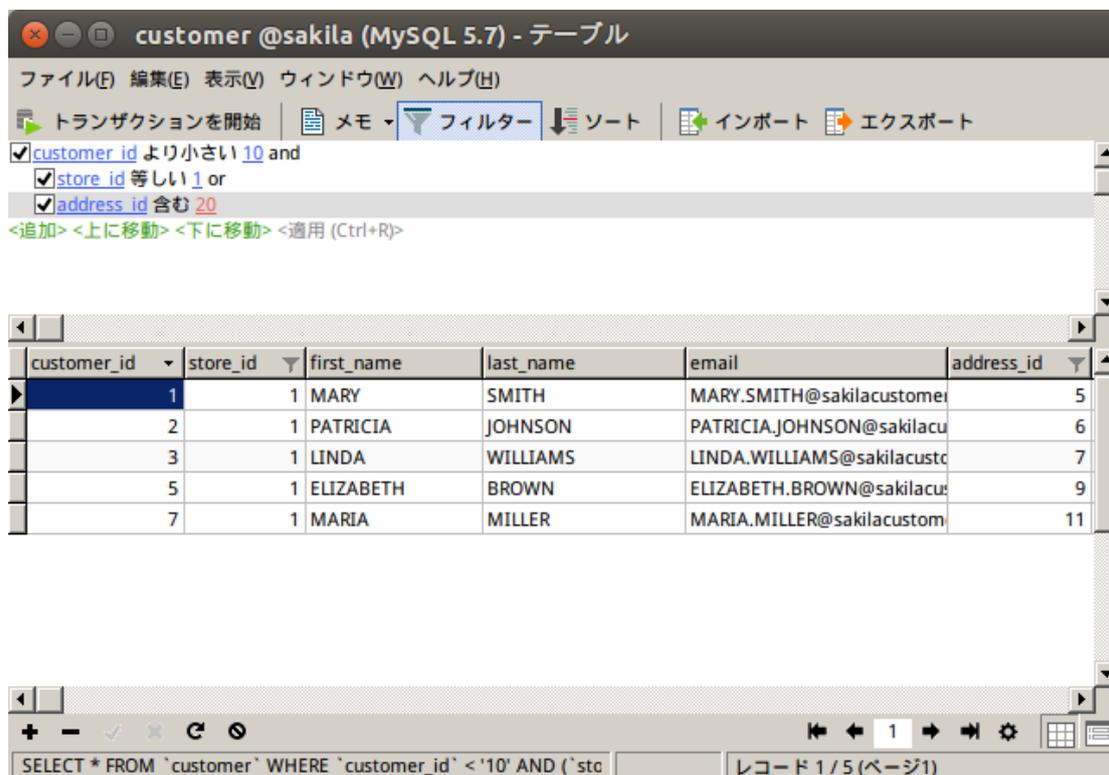
注意: キーボードの INSERT キーを使用して、挿入モードと上書きモードを切り替えることができます。

イメージペインでは、イメージとしてデータを表示することができます。  **ロード**、 **ディスクに保存**、 **クリア**ボタンを使用して、ファイルからイメージをロード/削除したり、イメージをファイルに保存することができます。

動的カラムペインでは、MariaDB の動的カラムとしてデータを編集することができます。左側の  と  ボタンを使用して、値を追加したり削除することができます。

フィルターウィザード

フィルターウィザードでは、テーブルグリッドに対して指定するフィルター条件を簡単に作成し適用することができます。さらに、将来使用する時のために、フィルター条件をプロファイルとして保存しておくことができます。ツールバーから  **フィルター**をクリックして、エディタを起動します。



1. 新しい条件を追加するためには、<追加>をクリックします。
2. (チェックボックスの隣の)列ボックスをクリックし、テーブル列を選択します。リストから[追加フィルター]を選択すると、条件を手動で入力することができます。
3. (列ボックスの隣の)演算子ボックスをクリックし、フィルター演算子を選択します。

フィルター演算子	結果
等しい <?>	フィールド = 値
等しくない <?>	フィールド <> 値
より小さい <?>	フィールド < 値
より小さいもしくは等しい <?>	フィールド <= 値
より大きい <?>	フィールド > 値
より大きいもしくは等しい <?>	フィールド >= 値
含む <?>	フィールド LIKE '%値%'
含む (大文字小文字を区別しない) <?> 注意: PostgreSQLのみ使用可能です。	フィールド ILIKE '%値%'
含まない <?>	フィールド NOT LIKE '%値%'
含まない (大文字小文字を区別しない) <?> 注意: PostgreSQLのみ使用可能です。	フィールド NOT ILIKE '%値%'
から始まる <?>	フィールド LIKE '値%'
で終わる <?>	フィールド LIKE '%値'
ヌルである	フィールド IS NULL
ヌルでない	フィールド IS NOT NULL
空である	フィールド = ''
空でない	フィールド <> ''

の間にある <?> <?>	((フィールド >= 値 1) AND (My_Field <= 値 2))
の間に無い <?> <?>	NOT ((フィールド >= 値 1) AND (My_Field <= 値 2))
リストに含まれる <?>	フィールド IN ('aaa','bbb',...)
リストに含まれない <?>	フィールド NOT IN ('aaa','bbb',...)

4. (演算子ボックスの隣の)条件値ボックスをクリックすると、適切なエディタが起動します。条件値を入力します。条件値ボックスで使用されるエディタは、該当する列に割り当てられるエディタの種類によって決まります。
5. (条件値ボックスの隣の)論理演算子ボックスをクリックし、**and** または **or** を選択します。
6. 別の新たな条件を追加する場合は、ステップ 1-5 を繰り返します。
7. **<適用 (Ctrl+R)>** をクリックするか、CTRL+R を押して、作成されたフィルターの結果を見ます。

ヒント: 複合フィルターを設定するためには、選択した条件を右クリックし、**インデント** または **アウトデント** を選択します。

将来使用する時のために、フィルター条件をプロファイルに保存したり、プロファイルから読み込むことができます。フィルターウィザード上で右クリックし、**プロファイルを開く**、または、**プロファイルを保存**、**名前を付けて保存**、**プロファイルを削除** を選択します。

クエリ

クエリは、ユーザーの要求に従い、読み取り可能な形式で、データベースからデータを抽出するために使用されます。Navicat は、SQL クエリを操作する 2 つの高性能なツールを提供しています：クエリ文を直接編集するクエリエディタと、クエリを視覚的に作成するクエリビルダです。また、[スケジュール](#)設定用にクエリを保存することができます。 をクリックして、**クエリ用のオブジェクトリスト**を開きます。もしくは、メインウィンドウの  ボタンをクリックして、クエリビルダまたはクエリエディタを開きます。

ヒント: クエリ(.sql)は、[設定の保存場所](#)の下に保存されます。接続が [Navicat Cloud](#) に同期される場合、クエリは Cloud に保存されます。

ボタン	説明
 実行	クエリを実行します：実行、選択部分を実行、ここからステートメントを実行する
 中止	クエリを止めます。
 説明	クエリのクエリプランを表示します。
 SQL を整形	エディタの SQL 整形設定を使って、コードを書式設定します。
 結果をエクスポート	クエリの結果をエクスポートします。

クエリビルダ(フルバージョンで使用可能)

Navicat は、視覚的にクエリを作成するのに役立つ**クエリビルダ**と呼ばれるツールを提供しています。このツールを使うと、SQL の知識がなくても、クエリを作成し編集することができます。データベースオブジェクトが左パネルに表示されます。一方、右パネルは 2 つの部分に分かれています：上部の**ダイアグラムデザイン**ペインと下部の**構文**ペイン。

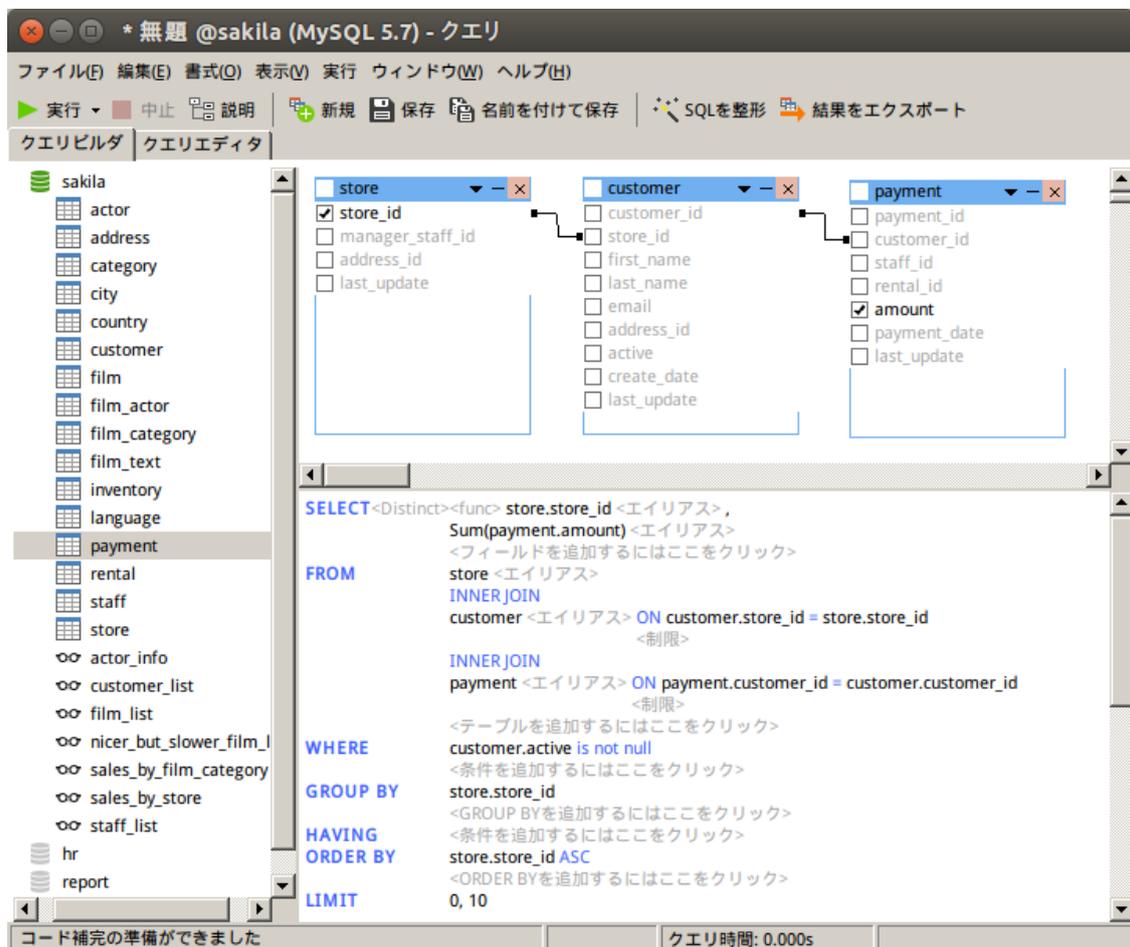
注意: クエリビルダは、*SELECT* 文のみをサポートします。複合クエリを作成するためには、クエリエディタを使用します。

クエリにテーブルまたはビューを追加するためには、左パネルのテーブルまたはビューをダイアグラムデザインペインにドラッグするか、ダブルクリックします。クエリにフィールドを追加するためには、ダイアグラムデザインペインのフィールド名の左にチェックを付けます。全てのフィールドを追加するためには、オブジェクトキャプションの左をクリックします。

ダイアグラムデザインペインからオブジェクトを削除するためには、オブジェクトキャプションの×ボタンをクリックします。

テーブル/ビューのエイリアスを追加するためには、テーブル/ビューの名前をダブルクリックし、ダイアグラムデザインペインにエイリアスを入力します。

ヒント: ダイアグラムデザインペインでフィールドを右クリックして、条件を設定することもできます。



フィールドの関連付けの設定

2つのフィールドによってデータベースオブジェクトを関連付けするためには、オブジェクトから1つのフィールドを他のオブジェクトのフィールドにドラッグします。そうすると、リンクされたフィールド間に1本の線が表示されます。

ヒント: オブジェクトの全てのリンクを削除するためには、オブジェクトエイリアスの隣の'-'ボタンをクリックします。

リンク間の関連付けを変更するためには、構文ペインを使用します。演算子をクリックし、ポップアップメニューからプロパティ要素を選択します。リスト(=, <>, <, <=, >, >=)から選択することによって、関連条件を変更できます。変更を確認するためには、**確認**をクリックします。また、Joinの種類を変更することもできます。

出力フィールドの設定

ダイアグラムデザインペインで選択されたフィールドが、構文ペインに表示されます。構文ペインでは、<Distinct>、<func>、<Alias>を使用して、表示順を設定したり、クエリの実出力フィールドを変更することができます。

<Distinct>

重複しているレコードがクエリの結果に含まれないようにしたい場合、このオプションを有効にします。

<func>

各フィールドに集約関数(SUM, MAN, MIX, AVG, COUNT)を設定します。

<エイリアス>

出カクエリフィールド名を変更します。

条件の設定

条件を追加するためには、構文ペインの **WHERE** 句から <--> = <--> をクリックします。<--> をクリックして、クエリで使用できる全てのテーブルフィールド一覧からフィールドを選択します。独自の条件を定義するためには、編集タブで値を直接入力します。条件演算子を設定するためには、= をクリックします。

グループ化条件の設定

構文ペインの **GROUP BY** 句でクエリレコードをグループ化するための条件を設定することができます。条件の設定と同じ方法で設定します。条件は、現在のクエリの **HAVING** ステートメントに含まれます。

ソート条件の設定

データベース/スキーマに問合せを行う時、クエリの最後に **ORDER BY** を追加することによって、結果を任意のフィールドの昇順または降順にソートすることができます。

クエリビルダで、構文ペインの **ORDER BY** 句でクエリレコードのソート方法を設定することができます。ソートの向きを変更するためには、**ASC** または **DESC** のいずれかをクリックします。

制限条件の設定

クエリの結果を指定範囲内に収まるように制限するためには、**LIMIT** 句が使用されます。結果の最初の X 個を表示したり、X - Y の範囲の結果を表示することができます。Limit X, Y のように表され、クエリの最後に追加されます。X が開始点(最初のレコードが 0 であることに注意)で、Y が幅 (表示するレコード数) です。

注意: MySQL、PostgreSQL、SQLite、MariaDB で使用可能。

クエリエディタ

Navicat は、クエリの作成や実行に役立つ **クエリエディタ** と呼ばれるツールを提供しています。このツールによって、クエリ用の SQL 文を作成・編集したり、選択したクエリを準備・実行することができます。

ヒント: クエリビルダでクエリを作成すると、クエリテキストが自動作成されます。

クエリの選択した部分を実行することができます。強調表示されたクエリを右クリックし、**選択部分を実行** を選択します。

1 つのエディタウィンドウで、複数の SQL 文を定義することができます。そして、カーソルが上にある (文の前にカーソルを置きます) 文を実行できます。**ここからステートメントを実行する** を選択するか、F7 を押します。

注意: **ここからステートメントを実行する** を選択するか、F7 を押すと、次の文が続いて実行されていきます。

エディタの高度機能

Navicat は、様々な高度機能を提供しています。例えば、強力なコード編集機能、スマートコード補完、SQL の書式設定など。

SQL の書式設定

SQL 文の書式を変更するためには、**書式**メニューから下記の項目を選択します。

インデント

コードの選択行のインデントを大きく/小さくします。

注釈行

コードの選択行をコメント化/非コメント化します。

文字を切り替える

選択されたコードを大文字/小文字に書式変更します。

SQL を整形(フルバージョンで使用可能)

SQL 整形の設定を使い、選択されたコードの書式を設定します。

SQL の整形の設定(フルバージョンで使用可能)

SQL の整形の設定を変更します。

オプション	説明
タブ文字を使用	タブ文字を使用する場合、このオプションにチェックをつけます。
タブのサイズ	タブのサイズを設定します。
短い括弧の長さ	短い括弧の長さを設定します。
大文字キーワード	すべての SQL キーワードを大文字に書式設定します。
設定を保存	整形 ボタンをクリックすると、SQL 整形オプションの設定を保存します。

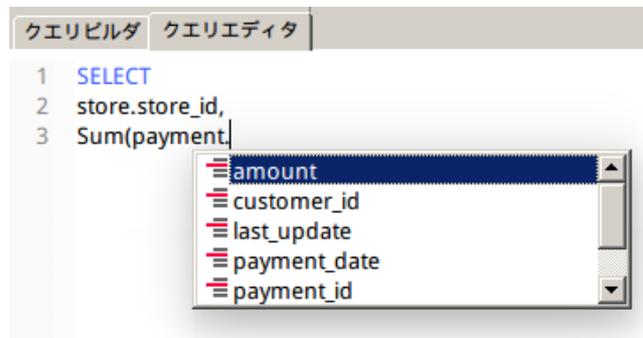
SQL を縮小(フルバージョンで使用可能)

SQL エディタで SQL の書式を縮小します。

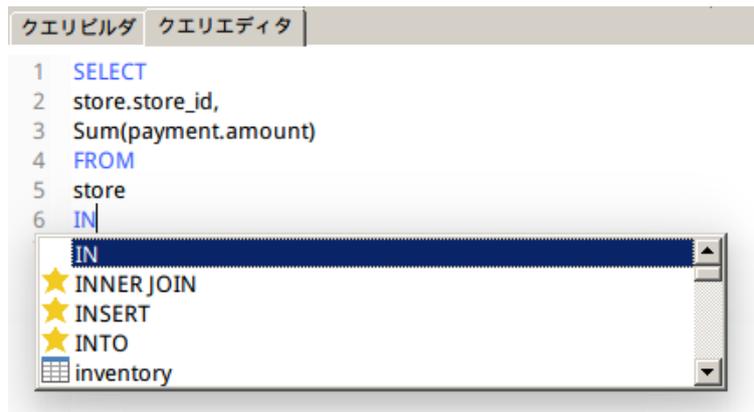
コード補完(フルバージョンで使用可能)

Navicat のコード補完機能は、エディタで SQL 文を入力する時に、ドロップダウンリストで情報を表示します。それは文の完成を補助し、データベースオブジェクト(例えば、データベース、テーブル、フィールド、ビューなど)の使用可能なプロパティが適切なアイコンとともに表示されます。

データベースオブジェクトの使用可能なプロパティについては、スコープ内で'.'を入力すると、コード補完機能が起動します。



ヒント: SQL キーワードについては、2 文字を入力するか、キーボードの CTRL+SPACE を押すことによって、コード補完機能を起動することができます。



ヒント: スマートコード補完は、単語を完成するための補完一覧を自動的にポップアップ表示します。

注意: コード補完機能は、ビューやファンクション/プロシージャなどにも適用されます。

関連トピック:

[コード入力支援オプション](#)

コードの折りたたみ

コードの折りたたみ機能を使うと、ひとまとまりのコードを折りたたんで、その一番最初の行だけが**エディタ**に表示されるようにすることができます。

折りたたみ可能なひとまとまりのコードは、その最初の行の左側にアイコン  が表示されます。そのアイコンから折りたたみ可能なコードの一番下まで、垂直線が伸びています。対照的に、折りたたまれているコードは、そのコードブロックの左側にアイコン  が表示されます。**エディタ**で、コードを折りたたむには  をクリックし、展開するには  をクリックします。

```

5 BEGIN
6   v_host := NULL;
7   v_path := NULL;
8   v_query := NULL;
9   a_pos1 := instr(v_url, '/');
10
11 IF...END IF;
14   a_pos2 := instr(v_url, '/', a_pos1 + 2);
15 IF...END IF;
20
21   v_host := substr(v_url, a_pos1 + 2, a_pos2 - a_pos1 - 2);
22   a_pos1 := instr(v_url, '?', a_pos2 + 1);
23
24 IF a_pos1 = 0 THEN

```

括弧の強調表示

Navicat は、エディタにおいて、対応する括弧の強調表示に対応しています。例えば、()。

注意: カーソルが括弧上に来た時に強調表示されます。

```

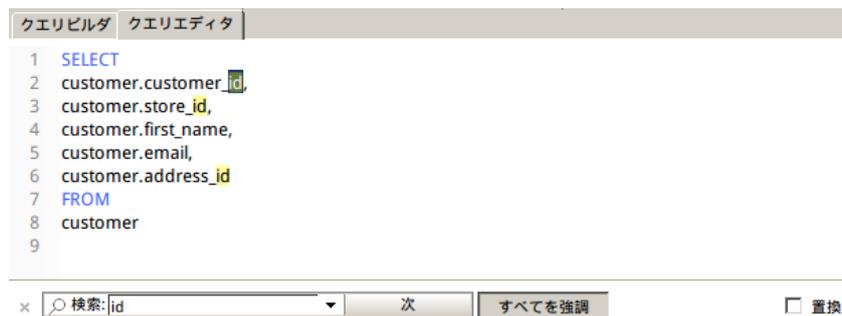
22
23 IF
24   status = diutil.s_subpNotFound
25 THEN
26   txt := '$$$ s_subpNotFound';
27 ELSIF (
28   status = diutil.s_stubTooLong
29 THEN

```

検索と置換

検索

エディタウィンドウには、テキストのクイック検索のために、**検索バー**が用意されています。メニューから**編集 -> 検索**をクリックするか、CTRL+F を押して、検索文字列を入力します。



検索は、カーソルの現在の位置から開始され、ファイルの最後で終わります。大文字小文字のどちらで検索を実行しても違いはありません。

次のテキストを見つけるためには、**次**をクリックするか、F3 を押します。

置換

置換バーを開くためには、**置換**ボックスにチェックを付けます。そして、検索/置換するテキストを入力します。

置換ボタンをクリックすると、最初の出現箇所が置換されます。

全て置換ボタンをクリックすると、全ての出現箇所が自動的に置換されます。



検索と置換用にいくつかの追加オプションがあります：



引用符付きコピー

SQL 文を引用符付きでコピーするためには、強調表示された SQL を右クリックしてから、**引用符付きでコピー**を選択し、書式を選択します。

注意: クエリ、ビュー、マテリアライズド・ビューで使用可能。

ズームイン/ズームアウト

Navicat は、エディタで SQL をズームイン/ズームアウトすることが可能です。ズームのオプションは**表示メニュー**から使用できます。下記のキーボードのショートカットを使って同様の効果を得ることができます。

- ズームイン: [CTRL+=]
- ズームアウト: [CTRL+-]
- リセット: [CTRL+0]

ヒント: 範囲は-10~+20 です。

注意: 別のタブで開いているファイルはズームの影響を受けません。

クエリの結果

クエリを実行するためには、ツールバーの  **実行**をクリックします。クエリ文が正確な場合、クエリが実行されます。そして、クエリ文がデータを返すことになっている場合、**結果**タブが開き、クエリによって返されたデータを表示します。クエリの実行中にエラーが発生した場合、実行は停止し、適切なエラーメッセージが表示されます。

結果タブは、クエリによって返された結果データをグリッドで表示します。2つのモードでデータを表示することができます：**グリッドビュー**と**フォームビュー**。詳細については、[テーブルビューワー](#)をご覧ください。

ヒント: Navicat は、10 個の結果セットを返すことをサポートしています。

注意: エディタの下または新しいタブから、結果タブの表示場所を選択することができます。**表示 -> 結果を表示 -> クエリエディタの下または新規タブ**を選択します。

結果タブに名前をつける

結果タブに名前をつけるためには、クエリエディタで、各 SELECT 文の前に `-- NAME:tab_name` または `/*NAME:tab_name*/` を追加します。

例:

```
-- NAME:Q1
SELECT * from table1;
/*NAME:Q2*/
SELECT * from table2;
```

クエリのプロファイルと状態(MySQL、MariaDB で使用可能)

クエリの実行時にプロファイルと状態を表示するためには、**表示 -> プロファイルと状態を表示**を選択し、ツールバーの  **実行**をクリックします。

プロファイルタブには、クエリのプロファイルが表示されます: テーブルロック、システムロック、統計情報など。

注意: MySQL 5.0 では、5.0.37 以上でサポートされています。
MySQL 5.1 では、5.1.24 以上でサポートされています。

状態タブには、クエリの状態が表示されます: 受信バイト、送信バイトなど。

クエリパラメータ

クエリビルダとクエリエディタのどちらも、クエリ文内でのパラメータの使用をサポートしています。クエリパラメータを設定することによって、実行時はいつも、クエリに変数値が追加されます。パラメータを識別子として示すためには、前に **\$**を付け、**[]**で囲んでください。例えば、`[$any_name]`。

クエリを実行すると、検索したい希望のデータを入力するための**パラメータを入力**ダイアログが表示されます。

Oracle クエリのデバッグ(フルバージョンで使用可能)

Oracle クエリをデバッグするために、ツールバーの  **デバッグ**をクリックして、[Oracle デバッガ](#)を起動させます。

クエリが入力パラメータを持つ場合、パラメータを入力します

モデル(Navicat Premium と Enterprise バージョンで使用可能)

モデルは、物理データベースモデルを作成し、操作するための高性能なツールです。  をクリックして、**モデル**用のオブジェクトリストを開きます。重要な特徴は下記のとおりです。

- 図で物理モデルを作成し操作する。
- データベース/スキーマまたはテーブル、ビューを物理モデルに変換する。(リバースエンジニアリング)
- 物理モデルを SQL ファイルやデータベース/スキーマに変換する。(フォワードエンジニアリング)
- テーブル構造を直接作成し編集する。

モデルを作成するためには、オブジェクトリストツールバーから  **新規モデル**をクリックします。新規モデルウィンドウがポップアップで表示されるので、ターゲットの**データベース**と**バージョン**を選択します。

ヒント: モデルファイル(.ndm)は、[プロファイルの保存場所](#)の下に保存されます。モデルが [Navicat Cloud](#) に同期される場合、モデルは Cloud に保存されます。

モデルサイドバー

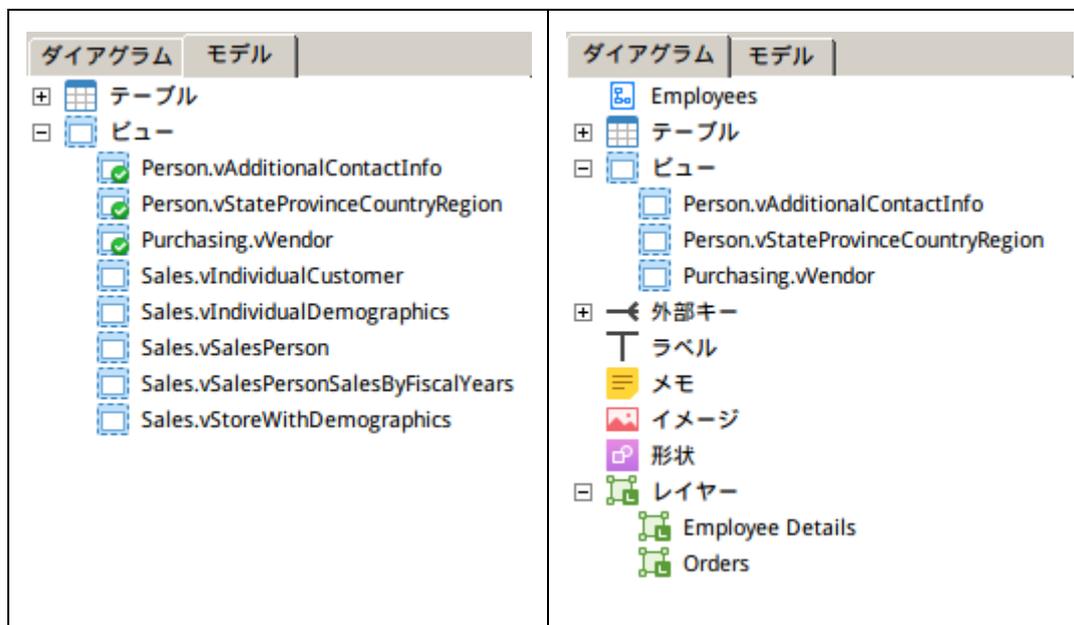
モデルのサイドバーには、モデル/ダイアグラムの全てのオブジェクトとそのプロパティ、操作履歴が一覧表示されます。

サイドバーは、以下のコンポーネントで構成されます:

- エクスプローラー
- 履歴
- プロパティ
- 概要

モデルエクスプローラーペイン

エクスプローラーペインには、2つのタブがあります。**モデル**と**ダイアグラム**です。モデルタブには、それぞれの個別ダイアグラムで使用されるものを含む、モデルの全てのテーブルまたはビューが含まれます。モデルタブからオブジェクトをドラッグし、アクティブなダイアグラムキャンバスにドロップすることができます。ダイアグラムタブには、アクティブダイアグラムに追加された全てのオブジェクト(テーブル、ビュー、外部キー、レイヤー、ノート、イメージなど)が含まれます。エクスプローラーペインが非表示の場合、モデルのメニューから**表示** -> **エクスプローラーを表示**を選択します。



モデル履歴ペイン

履歴ペインは、行われた全ての処理を表示します。その状態を復元するためには、その処理をクリックします。履歴ペインが非表示の場合、モデルのメニューから**表示 -> エクスプローラを表示**と**履歴を表示**を選択します。

モデルプロパティペイン

プロパティペインには、モデルのデフォルトプロパティを設定するための**モデル**、**ダイアグラム**、**オブジェクトタブ**があります。モデルやアクティブなダイアグラム、選択されたオブジェクトのプロパティ設定をすばやく編集することができます。プロパティペインが非表示の場合、モデルのメニューから**表示 -> プロパティを表示**を選択します。

矢印始点スタイル

矢印の後ろのスタイル。

白黒

ダイアグラムの色を白黒に変更する場合、このボックスにチェックをつけます。

太字

テーブルやビュー、外部キー、形状を太字にする場合、このボックスにチェックをつけるか、CTRL+B を押します。

境界の色

形状の境界の色。

キャップスタイル

線/矢印のキャップスタイル。

大文字小文字の区別

テーブルまたはビューの名前の大文字小文字の区別。MySQL/MariaDB の物理モデルで使用可能です。

色

オブジェクトの色。

ダッシュスタイル

線/矢印のダッシュスタイル。

データベースのタイプ

モデルのデータベースのタイプ

データベースのバージョン

モデルのデータベースのバージョン。

ダイアグラムノーターション

ダイアグラムのノーターション。この値は、デフォルト、シンプル、IDEF1X、UML、IE (Crow's Foot)、Classic のいずれかです。

矢印終点スタイル

矢印の前のスタイル。

フォント 太字

ノート/ラベルのフォントを太字にする場合、このボックスにチェックをつけます。

フォント色

ノート/ラベル/レイヤーのフォント色。

フォント 斜体

ノート/ラベルのフォントに斜体スタイルを適用する場合、このボックスにチェックをつけます。

フォント名

ノート/ラベル/レイヤーのフォント名。

フォントサイズ

ノート/ラベル/レイヤーのフォントサイズ。

高さ

オブジェクトの高さ。

高さ (ページ)

ダイアグラムの高さ (用紙数)。

ジョインスタイル

線/矢印のジョインスタイル。

左

オブジェクトからキャンバスの左端までのピクセル数。

モデルタイプ

モデルのタイプ。

名前

オブジェクト名。

不透明度

画像/形状の透明度。この値は、0~100 の間です。不透明の場合は 100 を、透明の場合は 0 を使用します。

参照されるカーディナリティ

参照される（親）テーブルの外部キーカーディナリティ。

参照カーディナリティ

参照する（子）テーブルの外部キーカーディナリティ。

スキーマ名

テーブル/ビューのスキーマ名。

名前を表示

外部キー/形状の名前を表示する場合、このボックスにチェックをつけます。

スキーマ名を表示

ダイアグラムにテーブル/ビューのスキーマ名を表示する場合、このボックスにチェックをつけます。

ビューのリレーションシップを表示

ビューのリレーションの線を表示する場合、このボックスにチェックをつけます。

テーブルフォント名

テーブルのフォント名。

テーブルフォントサイズ

テーブルのフォントサイズ。

上

オブジェクトからキャンバスの上までのピクセル数。

可視

外部キーの線を表示する場合、このボックスにチェックをつけます。

幅

オブジェクトの幅。

幅 (ページ)

ダイアグラムの幅（用紙数）。

モデル概要ペイン

概要ペインには、キャンバスのアクティブダイアグラム全体が表示されます。ダイアグラムの選択領域をズームインまたはズームアウトするためには、スライドバーで調整します。概要ペインが非表示の場合、モデルのメニューから**表示** -> **プロパティを表示** や **概要を表示** を選択します。キーボードショートカットを使用して、同様のことが行えます:

ズームイン: [CTRL++] または [CTRL+Mousewheel Up]

ズームアウト: [CTRL+-] または [CTRL+Mousewheel Down]

ダイアグラムキャンバス

ダイアグラムキャンバスは、オブジェクトの追加やダイアグラムの書式設定、モデルの印刷など、ダイアグラムを設計するためのキャンバスとツールバーから構成されます。モデルファイルは、1つ以上のダイアグラムを持ちます。各ダイアグラムは、モデル内で、タブを使用して表示されます。新しいダイアグラムを作成するためには、モデルのメニューから**ファイル** ->  **新規ダイアグラム** を選択します。

テーブルの作成

新しいテーブルを作成するためには、ダイアグラムツールバーから  ボタンをクリックし、キャンバス上でクリックします。エクスプローラーのモデルタブから既存のテーブルを追加するためには、モデルタブで選択されたテーブルをキャンバスにドラッグアンドドロップします。

デフォルトのダイアグラムノテーションについて、 アイコンは、フィールドがプライマリーキーであることを意味します。◆ アイコンは、フィールドがインデックスとして扱われることを表します。

注意: フィールドを右クリックすると、フィールドの追加や挿入、削除、名前の変更、フィールドをプライマリーキーとして設定することを選択できます。

キャンバスのテーブルオブジェクトのポップアップメニューオプション:

テーブルを設計

デザイナーでテーブル構造を編集します。例えば、フィールド、インデックス、外部キーなど。デザイナーのタブやオプションは、選択されたダイアグラムデータベースタイプによって異なります。それぞれのタブの設定については、[サーバーオブジェクト](#)を参照してください。

関連オブジェクトを追加

選択されたテーブルに関連する全てのテーブル/ビューを追加します。

フィールドを追加

既存のテーブルにフィールドを追加します。

切り取り

テーブルをダイアグラムから削除し、クリップボードに置きます。

コピー

テーブルをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全テーブルを選択

ダイアグラムの全てのテーブルを選択します。

削除

ダイアグラムから または ダイアグラムとモデルから、テーブルを削除します。

名前を変更

テーブルの名前を変更します。

色

テーブルの色を変更します。

適合サイズ

テーブルをその内容に合わせて自動的にリサイズします。

最前面へ移動

テーブルを最前面に移動します。

最背面へ移動

テーブルを最背面に移動します。

ビューの作成

新しいビューを作成するためには、ダイアグラムツールバーから  ボタンをクリックし、キャンバス上でクリックします。エクスプローラーのモデルタブから既存のビューを追加するためには、モデルタブで選択されたビューをキャンバスにドラッグアンドドロップします。

注意: ビューコネクターを右クリックすると、頂点の追加または削除、色の変更を選んだり、元のビューやターゲットテーブルに進むことができます。

キャンバスのビューオブジェクトのポップアップメニューオプション:

ビューを設計

デザイナーでビューの構造を編集します。デザイナーのタブやオプションは、選択されたダイアグラムデータベースタイプによって異なります。それぞれのタブの設定については、[サーバーオブジェクト](#)を参照してください。

関連オブジェクトを追加

選択されたビューに関連する全てのテーブル/ビューを追加します。

切り取り

ビューをダイアグラムから削除し、クリップボードに置きます。

コピー

ビューをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全ビューを選択

ダイアグラムの全てのビューを選択します。

削除

ダイアグラムから または ダイアグラムとモデルの両方から、ビューを削除します。

名前を変更

ビューの名前を変更します。

色

ビューの色を変更します。

適合サイズ

ビューをその内容に合わせて自動的にリサイズします。

最前面へ移動

ビューを最前面に移動します。

最背面へ移動

ビューを最背面に移動します。

外部キーの作成

外部キーを追加するためには、ダイアグラムツールバーから  ボタンをクリックし、子テーブルから親テーブルにフィールドをドラッグアンドドロップします。リンクされた名前ラベルを表示/非表示にするためには、プロパティペインの**名前を表示**オプションにチェックをつけます/外します。

キャンバスの外部キーオブジェクトのポップアップメニューオプション:

リレーションを設計

デザイナーで外部キーを編集します。デザイナーのオプションは、選択されたダイアグラムデータベースタイプによって異なります。設定については、[サーバーオブジェクト](#)を参照してください。

カーディナリティ テーブル 1

テーブル 1 のカーディナリティを設定します: なし、1 つだけ、複数、1 つまたは複数、0 または 1 つ、0 または複数。

カーディナリティ テーブル 2

テーブル 2 のカーディナリティを設定します: なし、1 つだけ、複数、1 つまたは複数、0 または 1 つ、0 または複数。

頂点を追加

外部キーコネクタに頂点を追加します。

頂点を削除

外部キーコネクタの頂点を削除します。

全頂点を削除

外部キーコネクタの全ての頂点を削除します。

ソースへ進む

元のテーブル(子テーブル)に進み、選択します。

ターゲットへ進む

ターゲットテーブル(親テーブル)に進み、選択します。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全リレーションを選択

ダイアグラムの全ての外部キーを選択します。

削除 ダイアグラムとモデルから

ダイアグラムとモデルから外部キーを削除します。

色

外部キーの色を変更します。

ラベルの作成

ラベルは、主にダイアグラム設計プロセスを文書で記録するために使用されます。例えば、テーブルオブジェクトのグループ分けを説明するために。新しいラベルを作成するためには、ダイアグラムツールバーから **T** ボタンをクリックし、キャンバス上でクリックします。

キャンバスのラベルオブジェクトのポップアップメニューオプション:

編集

ラベルの内容を変更します。

切り取り

ラベルをダイアグラムから削除し、クリップボードに置きます。

コピー

ラベルをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全て選択

ダイアグラムの全てのラベルを選択します。

削除

ダイアグラムからラベルを削除します。

適合サイズ

ラベルをその内容に合わせて自動的にリサイズします。

最前面へ移動

ラベルを最前面に移動します。

最背面へ移動

ラベルを最背面に移動します。

ノートの作成

ノートは、主にダイアグラム設計プロセスを文書で記録するために使用されます。例えば、テーブルオブジェクトのグループ分けを説明するために。新しいノートを作成するためには、ダイアグラムツールバーから  ボタンをクリックし、キャンバス上でクリックします。

キャンバスのノートオブジェクトのポップアップメニューオプション:

編集

ノートの内容を変更します。

切り取り

ノートをダイアグラムから削除し、クリップボードに置きます。

コピー

ノートをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全ノートを選択

ダイアグラムの全てのノートを選択します。

削除

ダイアグラムからノート削除します。

色

ノートの色を変更します。

適合サイズ

ノートその内容に合わせて自動的にリサイズします。

最前面へ移動

ノートを最前面に移動します。

最背面へ移動

ノートを最背面に移動します。

イメージの作成

新しいイメージを作成するためには、ダイアグラムツールバーから  ボタンをクリックし、キャンバス上でクリックします。それから、開くダイアログボックスでイメージファイルを選択します。

キャンバスのイメージオブジェクトのポップアップメニューオプション:

サイズをリセット

イメージのサイズを元のサイズにリセットします。

縦横比をリセット

イメージの元の高さとの比率を維持します。

切り取り

イメージをダイアグラムから削除し、クリップボードに置きます。

コピー

イメージをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全イメージを選択

ダイアグラムの全てのイメージを選択します。

削除

ダイアグラムからイメージを削除します。

最前面へ移動

イメージを最前面に移動します。

最背面へ移動

イメージを最背面に移動します。

形状の作成

新しい形状（線、矢印、長方形、楕円、ユーザー、データベース、クラウド、トリガー、サーバー、デスクトップ、モバイル）を作成するためには、ダイアグラムツールバーから  ボタンをクリックし、形状の種類を選択します。その後、キャンバス上でクリックします。リンクされた名前ラベルを表示/非表示にするためには、プロパティペインの**名前を表示**オプションにチェックをつけます/外します。

キャンバスの形状のポップアップメニューオプション:

縦横比をリセット (長方形、楕円、ユーザー、データベース、クラウド、トリガー、サーバー、デスクトップ、モバイルのみ)

形状の幅と高さの元の比率を維持します。

切り取り

形状をダイアグラムから削除し、クリップボードに置きます。

コピー

形状をダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全形状を選択

ダイアグラムの全ての形状を選択します。

削除

ダイアグラムから形状を削除します。

色

形状の色を変更します。

境界の色 (長方形、楕円、ユーザー、データベース、クラウド、トリガー、サーバー、デスクトップ、モバイルのみ)

形状の境界の色を変更します。

矢印始点スタイル (矢印のみ)

矢印の後ろのスタイルを変更します。

矢印終点スタイル (矢印のみ)

矢印の前のスタイルを変更します。

頂点を追加 (線と矢印のみ)

線または矢印に頂点を追加します。

頂点を削除 (線と矢印のみ)

線または矢印の頂点を削除します。

全頂点を削除 (線と矢印のみ)

線または矢印の全ての頂点を削除します。

最前面へ移動

形状を最前面に移動します。

最背面へ移動

形状を最背面に移動します。

レイヤーの作成

レイヤーは、キャンバス上のオブジェクト（例えば、テーブル、ノート、イメージなど）を整理するために使用されます。全ての関連オブジェクトを同じレイヤーに追加することができます。例えば、販売に関連する全てのテーブルを一つのレイヤーに追加するといった選択が可能です。新しいレイヤーを作成するためには、ダイアグラムツールバーから  ボタンをクリックし、キャンバス上でクリックします。

キャンバスのレイヤーオブジェクトのポップアップメニューオプション:

切り取り

レイヤーをダイアグラムから削除し、クリップボードに置きます。

コピー

レイヤーをダイアグラムからクリップボードにコピーします。

貼り付け

クリップボードの内容をダイアグラムに貼り付けます。

全レイヤーを選択

ダイアグラムの全てのレイヤーを選択します。

削除

ダイアグラムからレイヤーを削除します。

色

レイヤーの色を変更します。

適合サイズ

レイヤーをその内容に合わせて自動的にリサイズします。

最前面へ移動

レイヤーを最前面に移動します。

最背面へ移動

レイヤーを最背面に移動します。

ダイアグラムの書式設定

グリッドを表示

ダイアグラムキャンパスにグリッドを表示するためには、メニューから**表示** -> **グリッドを表示**を選択します。

グリッドにスナップ

グリッドが表示されたキャンパスでオブジェクトを一直線に並べるためには、メニューから**表示** -> **グリッドにスナップ**を選択します。

ダイアグラムノートーションを変更

ダイアグラムのノートーションを変更するためには、メニューから**ダイアグラム** -> **ダイアグラムノートーション**を選択し、ノートーションを選択します。

デフォルト	Navicat で使用されるデフォルトのノートーションスタイル。
シンプル	シンプルなノートーションスタイル。テーブルとビューのオブジェクトは名前を表示するだけです。
IE (Crow's Foot)	Crow's Foot ノートーションスタイル。
IDEF1X	ICAM DEFinition 言語情報モデリングメソッド。
UML	統一モデリング言語スタイル。
クラシック	クラシックノートーションスタイル。
白黒	ダイアグラムの色を白黒に変更します。
スキーマ名を表示	ダイアグラムにテーブルとビューのスキーマ名を表示します。

ダイアグラムディメンションを変更

ダイアグラムで使用されるページ数を変更するためには、メニューから**ダイアグラム** -> **ダイアグラムディメンション**を選択し、幅と高さを設定します。

オブジェクトの整列

キャンバス上のオブジェクトを整列するためには、2つ以上のオブジェクト(テーブル、ビュー、ノート、ラベル、イメージ、形状)を選択した後、右クリックし、**整列** -> **左揃え/中央揃え/右揃え/上揃え/中央揃え/下揃え**を選択します。

オブジェクトの配置を変更

キャンバス上のオブジェクトの配置を変更するためには、2つ以上のオブジェクト(テーブル、ビュー、ノート、ラベル、イメージ、形状)を選択した後、右クリックし、**配置** -> **横/縦**を選択します。

ページ設定を変更

紙のサイズや向き、余白を変更するためには、**ファイル** -> **ページのセットアップ**を選択します。

自動レイアウトを適用

キャンバスでオブジェクトを自動的に整えて配置するためには、ツールバーから  **自動レイアウト** をクリックします。自動レイアウトの書式を変更するためには、メニューから **ダイアグラム** -> **自動レイアウトの設定** を選択し、以下のオプションを設定します:

オプション	説明
自動ダイアグラムディメンション	適切なダイアグラムディメンションを自動選択します。
テーブルを自動サイズ調整	テーブルをその内容に合わせて自動的にリサイズします。
質	自動レイアウト出力の質。
オブジェクト間隔	ダイアグラムのオブジェクト間の距離。

モデルのプレビューと印刷

印刷する前にページをプレビューするためには、 **印刷のプレビュー** ボタンをクリックします。モデルは、プリンターまたは様々なファイル形式に印刷することができます。

プリンターに印刷

ダイアグラムを直接プリンターに送るためには、**ファイル** ->  **印刷** を選択します。ポップアップウィンドウで印刷オプションを設定できます。

ファイルに印刷(PDF/PNG/SVG)

ダイアグラムの PDF/PNG/SVG ファイルを作成するためには、**ファイル** -> **次の形式で印刷** -> **PDF/PNG/SVG** を選択します。

リバースエンジニアリング

リバースエンジニアリング は、モデルの重要な機能の 1 つです。この機能を使って、既存のデータベースの構造を読み込み、新しいダイアグラムを作成することができます。MySQL、PostgreSQL、Oracle、SQLite、MariaDB のデータベース/スキーマ/テーブル/ビューのインポートに対応しています。

Navicat では、ウィザードを使って、この処理を実行できます:

1. **ツール** -> **データベースからインポート** を選択します。
2. 接続を選択します。
3. インポートしたいデータベースまたはスキーマ、テーブルを選択します。
4. **開始** をクリックします。

Navicat メインウィンドウのリバースエンジニアリングを使用してモデルを新規作成することも可能です。開いているデータベース/スキーマまたはテーブルを右クリックし、ポップアップメニューから  **データベースをモデルに置き換え** または  **スキーマをモデルに置き換え**、 **テーブルをモデルに置き換え**、 **ビューをモデルに置き換え** を選択します。

スクリプトの生成

モデルの作成後、モデルのテーブル構造とビュー、外部キーをスクリプトファイルに保存することができます。**SQL をエクスポート**機能を使って、スクリプト用の SQL ファイルを作成します。**ツール** -> **SQL をエクスポート**を選択します。

SQL のエクスポートの基本設定

ファイル

出力ファイルの名前と場所を設定します。

オブジェクト

エクスポートしたい現在のモデルのオブジェクトを選択します。

SQL のエクスポートの詳細設定

以下のオプションは、選択されたダイアグラムデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、SQLite、MariaDB。

サーバーバージョン

SQL ファイル用に、サーバーのバージョンを選択します。

スキーマを含む

このオプションを選択すると、スキーマ名をファイルに追加します。選択しない場合、オブジェクト名だけが SQL 文に含まれます。

デフォルトスキーマ

オブジェクトのスキーマ名を設定します。スキーマの設定はなし。

Drop SQL を含む

このオプションを選択すると、drop object SQL 文をファイルに追加します。

Drop With CASCADE を含む

このオプションを選択すると、cascade オプション付きの drop object SQL 文をファイルに追加します。

プライマリーキーを含む

このオプションを選択すると、プライマリーキーをファイルに追加します。

外部キーを含む

このオプションを選択すると、外部キーをファイルに追加します。

UNIQUE 制約を含む

このオプションを選択すると、UNIQUE 制約をファイルに追加します。

インデックスを含む

このオプションを選択すると、インデックスをファイルに追加します。

チェック制約を含む

このオプションを選択すると、チェック制約をファイルに追加します。

除外制約を含む

このオプションを選択すると、除外制約をファイルに追加します。

ルールを含む

このオプションを選択すると、ルールをファイルに追加します。

トリガーを含む

このオプションを選択すると、トリガーをファイルに追加します。

キャラクターセットを含む

このオプションを選択すると、テーブルとフィールドのキャラクターセットをファイルに追加します。

自動増加を含む

このオプションを選択すると、テーブルの自動増加値をファイルに追加します。

照合を含む

このオプションを選択すると、テーブルの照合をファイルに追加します。

フォワードエンジニアリング

フォワードエンジニアリングは、モデルの重要な機能の 1 つです。この機能を使って、モデルを既存のデータベースまたはスキーマと比較し、両者間の構造の違いを表示することができます。そして、モデルの構造をターゲット接続に同期することができます。

Navicat では、ウィザードを使って、この処理を実行できます。

1. **ツール** -> **データベースに同期...**を選択します。
2. 同期の種類を選択します。
3. 同期元のデータベース、スキーマ、テーブルまたはビュー、そして、既存の接続からターゲット接続を選択します。
4. 比較と実行のオプションを選択します。
5. **比較**をクリックすると、ソーステーブルとターゲットテーブル間の違いを示すスクリプトが生成されます。
6. 実行したいスクリプトを選択します。
7. **クエリの実行**をクリックします。

同期の種類を選択

選択スキーマを同期

選択したスキーマにある全てのオブジェクトに対して機能する同期を設定します。

選択オブジェクトを同期

選択したオブジェクトのみに対して機能する同期を設定します。

スキーマ/オブジェクトと接続の選択

このステップでは、ターゲットスキーマまたはオブジェクトと比較する、モデルの 1 つ以上のスキーマまたはオブジェクトを選択します。モデルのオブジェクトが既存のスキーマからの場合、その既存のスキーマを選択することができます。そうでない場合、比較する元のモデルオブジェクトについて、**スキーマなしのオブジェクトは、このスキーマに同期しません**にターゲットスキーマの名前を入力してください。

それから、既存の接続からターゲット接続とデータベースを選択します。

比較/実行オプションの選択

以下のオプションは、選択されたダイアグラムデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、SQLite、MariaDB。

大文字小文字を区別して識別子と比較

大文字小文字を区別してテーブルの識別子と比較したい場合、このオプションにチェックをつけます。

テーブルを比較

テーブルを比較したい場合、このオプションにチェックをつけます。

プライマリーキーを比較

テーブルのプライマリーキーを比較したい場合、このオプションにチェックをつけます。

外部キーを比較

テーブルの外部キーを比較したい場合、このオプションにチェックをつけます。

インデックスを比較

インデックスを比較したい場合、このオプションにチェックをつけます。

トリガーを比較

トリガーを比較したい場合、このオプションにチェックをつけます。

キャラクターセットを比較

テーブルのキャラクターセットを比較したい場合、このオプションにチェックをつけます。

自動増加値を比較

テーブルの自動増加値を比較したい場合、このオプションにチェックをつけます。

UNIQUE 制約を比較

UNIQUE 制約を比較したい場合、このオプションにチェックをつけます。

チェック制約を比較

チェック制約を比較したい場合、このオプションにチェックをつけます。

除外制約を比較

除外制約を比較したい場合、このオプションにチェックをつけます。

ルールを比較

ルールを比較したい場合、このオプションにチェックをつけます。

照合を比較

テーブルの照合を比較したい場合、このオプションにチェックをつけます。

ビューを比較

ビューを比較したい場合、このオプションにチェックをつけます。

定義者を比較

ビューの定義者を比較したい場合、このオプションにチェックをつけます。

オブジェクト作成用 SQL

新しいデータベースオブジェクトをターゲットデータベースに作成する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

オブジェクト変更用 SQL

ターゲットデータベースのデータベースオブジェクトを変更する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

オブジェクト削除用 SQL

ターゲットデータベースからデータベースオブジェクトを削除する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

エラーが発生しても続ける

同期処理中に発生したエラーを無視します。

比較結果の表示

ソースオブジェクト/ターゲットオブジェクト

構造の比較後、ツリービューに、モデルとデータベース/スキーマオブジェクトとの差異が表示されます。**修正用クエリ**リストに、詳細な SQL 文が提供されます。

赤色のアイテムは、他のデータベース/スキーマに存在しないことを表します。

青色のアイテムは、他のデータベース/スキーマに存在するが、異なる定義が検出されたことを表します。



修正用クエリリストにおいて、デフォルトで、全てのスクリプトにチェックはつけられていません。ターゲットに適用したいスクリプトを選択してください。

複数行のスクリプトを強調表示し、右クリックするとポップアップメニューが表示されます。チェックされたスクリプトの詳細を選択すると、選択したスクリプトの完全な SQL 文を見ることができます。



クエリの実行ボタンをクリックして、選択したクエリを実行します。

モデルの変換

モデルをあるデータベースタイプから別のデータベースタイプに変換することができます。たとえば、MariaDB 10.0 の物理モデルを PostgreSQL 9.0 の物理モデルに。

変換中に、全てのデータタイプが自動的に変換されます。あるデータベースタイプから別のデータベースタイプに変換する場合、変換のプロセスはビューの SQL 構文を変更しません。ターゲットデータベースのバージョンが MySQL 4.0 以前の場合、全てのビューが削除されます。

開かれているモデルファイルを変換する場合、**ファイル** -> **モデル変換**を選択します。そして、ターゲットの**データベースとバージョン**を選択します。

モデル ヒントと Tips

Navicat は、より効率的にモデルを操作できるように、役立つヒントを提供しています。

動作	説明
ダイアグラムキャンバスのオブジェクトを探す	<ul style="list-style-type: none"> - エクスプローラーのダイアグラムタブで選択されたオブジェクトが、ダイアグラムキャンバスで強調表示されます。 - エクスプローラーのダイアグラムタブでオブジェクトをダブルクリックすると、ダイアグラムキャンバスの対応するオブジェクトにジャンプします。
モデルからオブジェクトを削除する	<ul style="list-style-type: none"> - ダイアグラムキャンバスでオブジェクトを選択し、SHIFT+DELETE を押します。
テーブル/ビューデザイナーを開く	<ul style="list-style-type: none"> - エクスプローラーのモデルタブまたはダイアグラムキャンバスでテーブル/ビューをダブルクリックします。
Navicat のメインウィンドウからテーブル/ビューを追加する	<ul style="list-style-type: none"> - Navicat メインウィンドウからテーブル/ビューをドラッグし、ダイアグラムキャンバスにドロップします。
テーブル/ビューの構造を取得する (SQL 文)	<ul style="list-style-type: none"> - ダイアグラムキャンバスでテーブル/ビューを選択しコピーして、他のテキストエディタに貼り付けます。
テーブルデザイナーなしでフィールドを設計する	<ul style="list-style-type: none"> - テーブル名を選択しクリックします。そして、タブ/下向き矢印を押すことによって、フィールドを追加/編集できます。 <p>Navicat は、入力されたフィールド名によってフィールドタイプを予測します。</p> <p>INTEGER/int/int4/NUMBER</p> <ul style="list-style-type: none"> - 接尾語 "id", "no" (最初の列の場合、プライマリーキーと予測されます) - 接尾語 "num" - "qty", "number" - "age", "count" <p>DECIMAL(10,2)/decimal(10,2)/NUMBER/REAL/money</p> <ul style="list-style-type: none"> - 接尾語 "price", "cost", "salary" <p>FLOAT/double/float8/NUMBER/REAL/float</p> <ul style="list-style-type: none"> - "size", "height", "width", "length", "weight", "speed", "distance" <p>DATE/datetime/date/TEXT/datetime2</p> <ul style="list-style-type: none"> - "date", "time" <p>VARCHAR(255)/varchar(255)/VARCHAR2(255)/TEXT</p> <ul style="list-style-type: none"> - 他のフィールド名 <p>フィールド名の前に*を入力すると、プライマリーキーとして認識されます。例えば、*itemNo:int。</p> <p>フィールド名とフィールドタイプの間には:を入力すると、フィールドタイプをカスタマ</p>

	イズできます。例えば、itemName:varchar(255)。
フィールドの順番を変更する	- ダイアグラムキャンバスでテーブルを選択した後、SHIFT キーを押した状態にします。  を使用して、希望の位置にフィールドをドラッグします。
フィールドを削除する	- ダイアグラムキャンバスでテーブルを選択した後、SHIFT キーを押した状態にします。  を使用して、希望のフィールドをテーブルの外にドラッグします。
外部キー/線/矢印に頂点を追加する	- ダイアグラムキャンバスで外部キー/線/矢印を選択した後、SHIFT キーを押した状態にします。外部キー/線/矢印の上でクリックすると、頂点が追加されます。
外部キー/線/矢印の頂点を削除する	- ダイアグラムキャンバスで外部キー/線/矢印を選択した後、SHIFT キーを押した状態にします。頂点をクリックします。
ハンドモードに切り替える	- SPACE キーを押した状態にし、ダイアグラムを移動します。
印刷のプレビューでページを選択する	- SHIFT キーを押した状態にし、ページにポイントを当てると、ページ番号が表示されます。 - SHIFT キーを押した状態にし、ページをクリックすると、ダイアグラムキャンバスの対応するページにジャンプします。

高度なツール

Navicat は、**インポートウィザード**、**エクスポートウィザード**、**SQL ファイルにダンプ**、**SQL ファイルの実行**などを含む、データを操作するための高性能なツールを多く提供しています。

インポートウィザード

インポートウィザードを使い、CSV、TXT、XML、DBF などから、テーブルにデータをインポートすることができます。スケジュールの設定用にプロファイルとして設定を保存することができます。

注意: Navicat Essentials バージョンは、TXT、CSV、HTML、XML、JSON ファイルといったテキストベースのファイルのインポートのみに対応しています。

インポートウィザードを開くためには、テーブルオブジェクトリストツールバーから  **インポートウィザード**をクリックします。

ヒント: サポートされているファイルをテーブルのオブジェクトリストペインや接続ペインのデータベース/スキーマにドラッグすることができます。Navicat は、**インポートウィザード**ウィンドウをポップアップで表示します。既存のテーブルが強調表示されている場合、Navicat は強調表示されているテーブルにファイルをインポートします。そうでない場合は、新しいテーブルにファイルをインポートします。

インポートファイル形式の設定 (ステップ 1)

使用可能なインポート元のファイルの種類を選択します。

ソースファイルの名前の設定 (ステップ 2)

インポート元ファイル名を表示します。**インポート元**テキストボックスのファイル名の拡張子は、ステップ 1 で選択されたインポートタイプに従って変わります。インポート元ファイルの**エンコード**を選択します。

注意: TXT と XML ファイルの場合は、インポートするファイルを 1 つ以上選択することができます。

区切り文字の設定 (ステップ 3) - TXT、XML

TXT

ファイルの**レコード分離記号**、**フィールド区切り文字**、**テキスト修飾子**を定義します。

固定長でテキストファイルをインポートするためには、**固定長**を選択します。インポート元のファイルの列の範囲を区切るためには、希望の位置をクリックして、区切り線を作成します。その線をドラッグすると、移動できます。ダブルクリックすると、削除できます。



XML

テーブル行を識別するタグを定義します。

タグ属性をテーブルフィールドとみなす

例:

```

<row age="17">
<id>1</id>
<name>size</name>
</row>

```

このオプションを有効にすると、Navicat は、"id" や "name" とともに、"age" をテーブルフィールドとみなします。有効にしない場合、"id" と "name" のみがテーブルフィールドとしてインポートされます。

注意: Navicat は、多階層の XML ファイルをサポートしていません。

追加オプションの設定 (ステップ 4) - TXT、XML、HTML

以下のオプションは、ステップ 1 で選択されたファイル形式によって異なります。

フィールド名の行

フィールド名の行は、Navicat が列のタイトルとしてみなすべき行を示します。

先頭データ行

先頭データ行は、Navicat が実データを読み始めるべき行を示します。

最終データ行

最終データ行は、Navicat が実データの読み込みをやめるべき行を示します。

注意: ファイルに列のタイトルを全く定義しない場合、先頭データ行に **1** を、フィールド名の行に **0** を入力してください。

日付形式, 日付区切り, 時間区切り, 日付時間順

日付と時間の書式を定義します。

小数点

小数の書式を定義します。

バイナリデータエンコーディング

バイナリデータがファイルから **Base64** でエンコード または **エンコードなし**のどちらとしてインポートされるかを設定します。

ターゲットテーブルの設定 (ステップ 5)

新しいテーブル名を定義するか、既存のテーブルにインポートする場合はドロップダウンリストからテーブルを選択します。

注意: **ターゲットテーブル**に新しいテーブル名を入力すると、**新規テーブル**に自動的にチェックがつけられます。

ソーステーブル	ターゲットテーブル	新規テーブル
items	items	<input checked="" type="checkbox"/>

複数のテーブルをインポートする場合、全てのテーブルがリストに表示されます。

ソーステーブル	ターゲットテーブル	新規テーブル
clients	clients	<input type="checkbox"/>
items	items	<input type="checkbox"/>

フィールド構造の適用とフィールドのマッピング (ステップ 6)

Navicat は、ソーステーブルのフィールドのタイプと長さを推測します。ドロップダウンリストから希望のタイプを選択することができます。

ヒント: 複数のテーブルをインポートする場合、**ソーステーブル**ドロップダウンリストから他のテーブルを選択します。

ソーステーブル:	items				
ターゲットテーブル:	clients customer items orders				
	ターゲットフィールド				
<input checked="" type="checkbox"/>	OrderNo				
<input checked="" type="checkbox"/>	ItemNo				
<input checked="" type="checkbox"/>	PartNo	varchar	255	0	
<input checked="" type="checkbox"/>	Qty	varchar	255	0	
<input checked="" type="checkbox"/>	Discount	varchar	255	0	

既存のテーブルにデータをインポートする場合、ソースフィールド名をインポート先テーブルに手動でマップするか、右クリックメニューから、**すべてスマートマッチ**、**すべてダイレクトマッチ**または**すべてアンマッチ**を選択する必要があります。

ターゲットフィールド	ソースフィールド	プライマリーキー
Species No		
Category		
Common_Name		
Species Name	Species No	
Length (cm)	Category	
Length_In	Common_Name	
Graphic	Species Name	
	Length (cm)	
	Length_In	

ターゲットフィールド	ソースフィールド	プライマリーキー
Species No		
Category		
Common_Name		
Species Name		
Length (cm)		
Length_In		
Graphic		

すべてスマートマッチ

すべてダイレクトマッチ

すべてアンマッチ

インポートモードの選択 (ステップ 7)

データのインポート方法を定義するインポートモードを選択します。

インポート形式

追加：インポート先テーブルにレコードを追加します

更新：インポート先に既に存在しているレコードのみを更新します

追加/更新：インポート先にレコードが存在する場合は更新し、ない場合は追加します

削除：インポート元レコードに一致する、インポート先レコードを削除します

コピー：インポート先レコードを全て削除し、インポート元から追加します

詳細

ヒント: 残りのオプションを有効にするためには、ステップ 6 でプライマリーキーを有効にしなければなりません。

ターゲットフィールド	ソースフィールド	プライマリーキー
Species No	Species No	🔑
Category	Category	
Common_Name	Common_Name	
Species Name	Species Name	
Length (cm)	Length (cm)	
Length_In	Length_In	
Graphic	Graphic	

その他の設定に関しては、**詳細**ボタンをクリックします：

以下のオプションは、選択されたデータベースの種類によって異なります：MySQL、Oracle、PostgreSQL、SQLite、MariaDB。

各実行時に複数クエリを実行

各実行時に複数のクエリを実行したい場合は、このオプションにチェックをつけます。

extended-insert ステートメントを使用

extended-insert 構文を使用してレコードを挿入します。

例:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindsy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

空文字を NULL として使用

元のデータフィールドが空文字を含む場合、**NULL** 値をインポートします。

外部キー制約を使用

テーブル間に外部キーリレーションがある場合、外部キーを追加します。

エラーが発生しても続ける

インポート処理中に発生したエラーを無視します。

自動増加フィールドを作成

インポート処理中、自動増加フィールドを作成します。

注意: ファイルタイプが Paradox ファイル、DBase ファイルの場合のみサポートされます。

削除されたレコードをインポート

インポート処理中、DBase ファイルの削除済みのレコードをインポートします。

注意: ファイルタイプが DBase ファイルの場合のみサポートされます。

インポートの保存と確認 (ステップ 8)

開始 ボタンをクリックして、インポート処理を開始します。成功または失敗を示す実行中プロセスを見ることができます。このメッセージは、ファイル - LogImport.txt に保存されます。

ヒント: **保存** ボタンをクリックして、スケジュール設定用にプロファイルとして設定を保存することができます。

ログ ボタンをクリックして、ログファイルを見ることができます。

エクスポートウィザード

エクスポートウィザード を使用して、テーブル、ビュー、クエリの結果からのデータを、使用可能な形式にエクスポートすることができます。スケジュールの設定用にプロファイルとして設定を保存することができます。

注意: Navicat Essentials バージョンは、TXT、CSV、HTML、XML、JSON ファイルといったテキストベースのファイルのエクスポートのみに対応しています。

エクスポートウィザードを開くためには、オブジェクトリストツールバーから  **エクスポートウィザード**をクリックします。

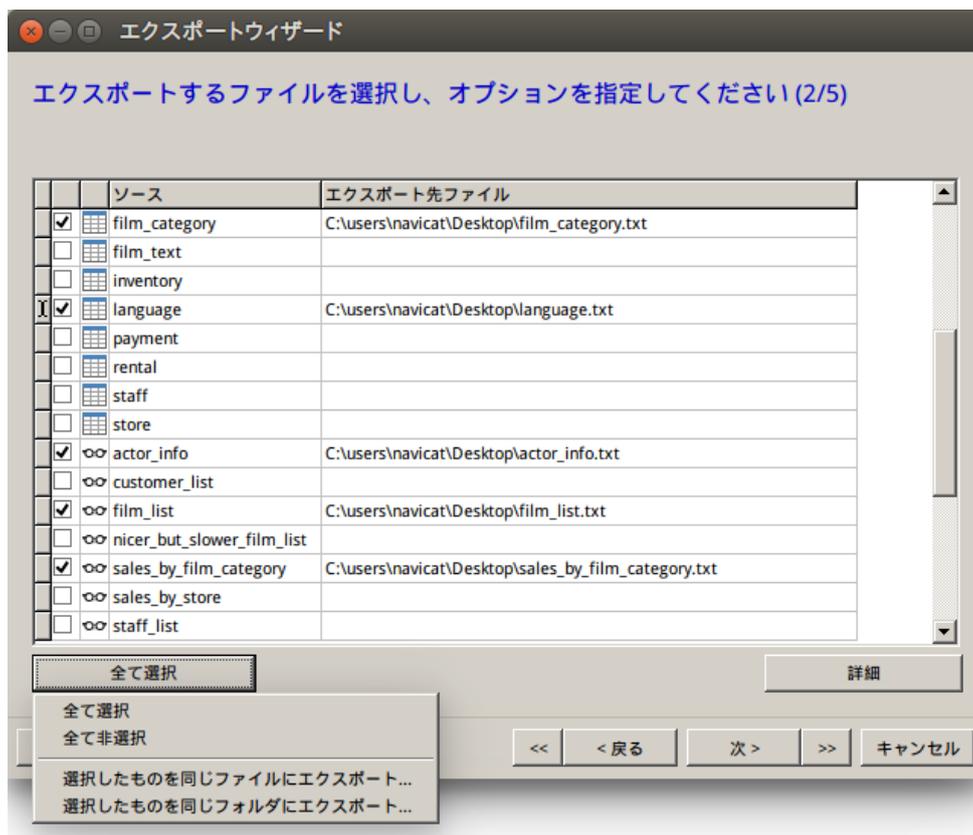
エクスポートファイル形式の設定 (ステップ 1)

ターゲットファイルに関する使用可能なエクスポート形式の 1 つを選択します。

エクスポート先のファイル名の設定 (ステップ 2)

エクスポートファイルの名前とロケーションを設定します。**ソース**のテーブル名の隣のボックスにチェックをつけて、直接デフォルトファイルの名前とロケーションを割り当てることができます。既存のテーブルが強調表示されている場合、そのテーブルに自動的にチェックがつけられ、デフォルトのファイル名とロケーションが割り当てられます。**エクスポート先ファイル**テキストボックスのファイルの拡張子は、ステップ 1 で選択されたエクスポートタイプによって変わります。

注意: クエリ結果をエクスポートする場合は、エクスポートウィザードを実行する前にクエリを必ず保存してください。そうしないと、元のテーブルはここに表示されません。



選択したテーブルを同じターゲットファイルにエクスポートする場合、クイックマッピング用の**全て選択**ボタンから、**選択したものを同じファイルにエクスポート**を選択します。

選択したテーブルを同じディレクトリにエクスポートする場合、クイックマッピング用の**全て選択**ボタンから、**選択したものを同じフォルダにエクスポート**を選択します。

その他の設定に関しては、**詳細**ボタンをクリックします：

エンコード

エクスポートファイルのエンコーディングを選択します。

タイムスタンプを追加

ファイル名にエクスポート実行のタイムスタンプを入れたい場合、このオプションにチェックをつけます。ドロップダウンリストから日付/時刻の書式を選択します。

エクスポート対象フィールドの選択 (ステップ 3)

エクスポート対象のテーブルフィールドを選択します。デフォルトで、**利用可能なフィールド**リストにある全てのフィールドが選択されています。エクスポートしたくないフィールドがある場合、まず**全てのフィールド**ボックスのチェックをはずしてから、**利用可能なフィールド**リストにあるフィールドのチェックを外します。

注意: クエリ結果のエクスポートの場合、ウィザードはこのステップを飛ばします。

追加オプションの設定 (ステップ 4)

以下のオプションは、ステップ 1 で選択されたファイル形式によって異なります。

列タイトルを含める

このオプションが有効な場合、フィールド名がエクスポートファイルに含まれます。

追加

ステップ 2 で複数のテーブルに対して**選択したものを同じターゲットファイルにエクスポート**オプションを選択した場合は、エクスポートファイルにレコードを追加するためにこのオプションにチェックをつけてください。

エラーが発生しても続ける

エクスポート処理中に発生したエラーを無視します。

XML で属性フォーマットを使用

属性フォーマット
<pre><RECORDS> <RECORD OrderNo="1003" ItemNo="1" PartNo="1313" Qty="5" Discount="0"></RECORD> <RECORD OrderNo="1004" ItemNo="1" PartNo="1313" Qty="10" Discount="50"></RECORD> </RECORDS></pre>
非属性フォーマット
<pre><RECORDS> <RECORD> <OrderNo>1003</OrderNo> <ItemNo>1</ItemNo></pre>

```
<PartNo>1313</PartNo>
<Qty>5</Qty>
<Discount>0</Discount>
</RECORD>
<RECORD>
  <OrderNo>1004</OrderNo>
  <ItemNo>1</ItemNo>
  <PartNo>1313</PartNo>
  <Qty>10</Qty>
  <Discount>50</Discount>
</RECORD>
</RECORDS>
```

ファイルのレコード分離記号、フィールド区切り文字、テキスト修飾子を定義します。

日付形式, 日付区切り, 時間区切り, 日付時間順

日付と時間の書式を定義します。

小数点

小数の書式を定義します。

バイナリデータエンコーディング

バイナリデータがファイルの **Base64** でエンコード または **エンコードなし**のどちらとしてエクスポートされるかを設定します。

エクスポートの保存と確認 (ステップ 5)

開始ボタンをクリックし、エクスポート処理を開始します。成功または失敗を示す実行中プロセスを見ることができます。このメッセージは、ファイル - LogExport.txt に保存されます。

ヒント: **保存**ボタンをクリックして、スケジュールの設定用にプロファイルとして設定を保存することができます。

開くボタンをクリックして、ログファイルを開くことができます。

データ転送(フルバージョンで使用可能)

Navicat を使用して、ひとつのデータベース/スキーマから他のデータベース/スキーマに、または sql ファイルに、データベースオブジェクトを転送することができます。ターゲットのデータベース/スキーマは、元のデータベース/スキーマと同じサーバーにあっても、別のサーバーにあっても構いません。スケジュール設定用にプロファイルとしてデータ転送設定を保存することができます。メインメニューから **ツール** ->  **データ転送**を選択します。

ヒント: テーブルを接続ペインのデータベース/スキーマにドラッグすることができます。ターゲットのデータベース/スキーマが同じ接続内にある場合、Navicat は直接テーブルをコピーします。そうでない場合、Navicat は**データ転送**ウィンドウをポップアップで開きます。

保存済みのプロファイルを開くためには、**プロファイル**タブで、プロファイルを選択して**ロード**ボタンをクリックするか、プロファイルをダブルクリックします。

データ転送の基本設定

ソース(転送元)

ソースの接続とデータベース/スキーマを定義します。

デフォルトで、**データベースオブジェクト**リストにある全てのデータベースオブジェクトが選択されています。転送したくないデータベースオブジェクトがある場合、それらのチェックをはずします。

このオプションが有効な場合、チェックされたデータベースオブジェクトだけが転送されます。しかし、データ転送プロファイルを作成後にソースデータベース/スキーマに新しいデータベースオブジェクトを追加すると、**データベースオブジェクト**リストを手動で修正しない限り、新しく追加されたデータベースオブジェクトは転送されません。

 全てのデータベースオブジェクトをターゲットデータベース/スキーマに転送したい場合、このオプションを選択します。データ転送プロファイルを修正しなくても、新しく追加されたデータベースオブジェクトは全て転送されます。

ターゲット

接続

選択されたデータベースオブジェクトを接続、データベース/スキーマに直接転送します。

ファイル

選択されたデータベースオブジェクトをテキストファイルに直接転送します。ファイルの **SQL フォーマット**と**エンコード**を選択することができます。

データ転送の詳細設定 (同種サーバー間)

このタブでは、同種サーバー間もしくは MySQL と MariaDB 間の転送に関する詳細設定を選択することができます。

以下のオプションは、選択されたデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、SQLite、MariaDB。

テーブルを作成

このオプションを有効にすると、ターゲットデータベース/スキーマにテーブルを作成します。

例えば、このオプションが非選択で、ターゲットデータベース/スキーマにテーブルがすでに存在している場合、全てのデータは目的のテーブルに追加されます。

インデックスを含む

このオプションを有効にすると、テーブルにインデックスを含めます。

外部キー制約を含む

このオプションを有効にすると、テーブルに外部キーを含めます。

エンジン/テーブルタイプを含む

このオプションを有効にすると、テーブルタイプを含めます。

キャラクターセットを含む

このオプションを有効にすると、テーブルにキャラクターセットを含めます。

自動増加を含む

このオプションを有効にすると、テーブルに自動増加を含めます。

他のテーブルオプションを含む

このオプションを有効にすると、テーブルに他のオプションを含めます。

UNIQUE 制約を含む

このオプションを有効にすると、テーブルにユニーク制約を含めます。

ルールを含む

このオプションを有効にすると、テーブルにルールを含めます。

チェック制約を含む

このオプションを有効にすると、テーブルにチェック制約を含めます。

トリガーを含む

このオプションを有効にすると、テーブルにトリガーを含めます。

除外制約を含む

このオプションを有効にすると、テーブルに除外制約を含めます。

オブジェクト名を変換

処理中にオブジェクトの名前を**小文字**または**大文字**に変換する必要がある場合、このオプションにチェックをつけます。

レコードの挿入

全てのレコードを目的のデータベース/スキーマに転送する必要がある場合、このオプションにチェックをつけます。

ターゲットテーブルをロック

データ転送処理中、ターゲットデータベース/スキーマにあるテーブルにロックをかけます。

トランザクションを使用

データ転送処理中、トランザクションを使用する場合、このオプションにチェックをつけます。

complete-insert ステートメントを使用

完全な insert 構文を使用して、レコードを挿入します。

例:

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('1', 'Peter McKindsy', '23');
```

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('2', 'Johnson Ryne', '56');
```

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('0', 'katherine', '23');
```

extended-insert ステートメントを使用

拡張 insert 構文を使用して、レコードを挿入します。

例:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

delayed-insert ステートメントを使用

DELAYED insert SQL 文を使用して、レコードを挿入します。

例:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');
```

```
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');
```

```
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

複数の insert 文を実行

それぞれの実行中に複数の insert 文を実行したい場合、このオプションにチェックをつけます。データ転送処理はより早くなるでしょう。

BLOB の場合 16 進形式を使用

16 進形式で BLOB データを挿入します。

エラーが発生しても続ける

転送処理中に発生したエラーを無視します。

ソーステーブルをロック

データ転送がいったん始まると、元のデータベース/スキーマにあるテーブルへの更新ができないように、ロックをかけます。

作成前にターゲットオブジェクトを削除

データベースオブジェクトがすでにターゲットデータベース/スキーマに存在する場合、このオプションにチェックをつけます。データ転送が開始されるとすぐに、既存のオブジェクトは削除されます。

ターゲットデータベース/スキーマがなければ作成

ターゲットサーバーに指定されたデータベース/スキーマが存在しない場合、新しいデータベース/スキーマを作成します。

SHOW CREATE TABLE から DDL を使用する

このオプションを有効にすると、DDL は SHOW CREATE TABLE から使用されます。

sqlite_master から DDL を使用する

このオプションを有効にすると、DDL は *SQLITE_MASTER* テーブルから使用されます。

データ転送の詳細設定 (クロスサーバー)(Navicat Premium のみ使用可能)

Navicat Premium は、種類の異なるサーバー間でテーブルとデータを転送することができます。例えば、MySQL から Oracle へ。MySQL と MariaDB 間で転送を行う場合は、[データ転送の詳細設定 \(同種サーバー間\)](#)を参照してください。

以下のオプションは、選択されたデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、SQLite、MariaDB。

テーブルを作成

このオプションを有効にすると、ターゲットデータベース/スキーマにテーブルを作成します。

例えば、このオプションが非選択で、ターゲットデータベース/スキーマにテーブルがすでに存在している場合、全てのデータは目的のテーブルに追加されます。

インデックスを含む

このオプションを有効にすると、テーブルにインデックスを含めます。

外部キー制約を含む

このオプションを有効にすると、テーブルに外部キーを含めます。

オブジェクト名を変換

処理中にオブジェクトの名前を**小文字**または**大文字**に変換する必要がある場合、このオプションにチェックをつけます。

レコードの挿入

全てのレコードを目的のデータベース/スキーマに転送する必要がある場合、このオプションにチェックをつけます。

ターゲットテーブルをロック

データ転送処理中、ターゲットデータベース/スキーマにあるテーブルにロックをかけます。

トランザクションを使用

データ転送処理中、トランザクションを使用する場合、このオプションにチェックをつけます。

complete-insert ステートメントを使用

完全な insert 構文を使用して、レコードを挿入します。

例:

```
INSERT INTO `users` (` ID Number`, `User Name`, `User Age`) VALUES ('1', 'Peter McKindsy', '23');  
INSERT INTO `users` (` ID Number`, `User Name`, `User Age`) VALUES ('2', 'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, `User Name`, `User Age`) VALUES ('0', 'katherine', '23');
```

extended-insert ステートメントを使用

拡張 insert 構文を使用して、レコードを挿入します。

例:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindsy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

delayed-insert ステートメントを使用

DELAYED insert SQL 文を使用して、レコードを挿入します。

例:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindsy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

複数の insert 文を実行

それぞれの実行中に複数の insert 文を実行したい場合、このオプションにチェックをつけます。データ転送処理はより早くなるでしょう。

BLOB の場合 16 進形式を使用

16 進形式で BLOB データを挿入します。

エラーが発生しても続ける

転送処理中に発生したエラーを無視します。

ソーステーブルをロック

データ転送がいったん始まると、元のデータベース/スキーマにあるテーブルへの更新ができないように、ロックをかけます。

作成前にターゲットオブジェクトを削除

データベースオブジェクトがすでにターゲットデータベース/スキーマに存在する場合、このオプションにチェックをつけます。データ転送が開始されるとすぐに、既存のオブジェクトは削除されます。

ターゲットデータベース/スキーマがなければ作成

ターゲットサーバーに指定されたデータベース/スキーマが存在しない場合、新しいデータベース/スキーマを作成します。

データの同期 (フルバージョンで使用可能)

Navicat では、詳細な分析処理で、1 つのデータベース/スキーマから別のデータベース/スキーマにデータを転送することができます。つまり、Navicat は、それぞれのレポジトリが同じ情報を持つために、異なるデータベース/スキーマにあるデータを最新の状態にすることが可能です。転送処理をロールバックすることも、転送先へレコードを挿入したり、転送先にあるレコードを削除、更新することもできます。スケジュールの設定用に、プロファイルとしてデータ同期設定を保存することも可能です。メインメニューから **ツール** ->  **データの同期** を選択します。

全てのテーブルがプライマリーキーを含んでいなければなりません。そして、全てのテーブル構造が、同期元とターゲットの間で、全く同じでなければなりません。データの同期の前に、構造の同期を適用するといいでしょう。

実行中プロセスの成功または失敗を示すメッセージは、ファイル - LogSynchronize.txt に保存されます。

保存済みのプロファイルを開くためには、**プロファイル** タブで、プロファイルを選択して **ロード** ボタンをクリックするか、プロファイルをダブルクリックします。

注意: Oracle サーバーに関しては、データの同期処理中、BLOB、CLOB、NCLOB、LONG、LONG RAW データはスキップされます。TIMESTAMP プライマリーキーは、9i サーバーにデータベースリンクを使用して同期(挿入、更新)することができません。RAW プライマリーキーは、エラーなしに、どのサーバーへもデータベースリンクを使用して同期(挿入、更新、削除)することはできません。

Navicat Premium と Navicat for MySQL は、MySQL と MariaDB 間の同期に対応しています。

データの同期の基本設定

ソース/ターゲット

ソースとターゲットの接続、データベース、スキーマを定義します。

注意: Oracle サーバーに関しては、ターゲットの Oracle データベースに[パブリック/プライベート](#)データベースリンクを事前に作成しておく必要があります。

ソーステーブル/ターゲットテーブル

デフォルトで、同期元とターゲットで同じテーブル名のテーブルだけが、リストにマップされます。同期したくないテーブルがある場合、ドロップダウンリストから手動で無効にします。

ヒント: 実行前に、結果をプレビューすることができます。

データの同期の詳細設定

トランザクションを使用

エラーが発生した時、全てのデータをロールバックします。

同期の詳細を見る

同期中に**メッセージログ**タブに詳細なプロセスを一覧表示したい場合、このオプションにチェックをつけます。

注意: このオプションのチェックを外すと、処理は早くなります。

レコードの挿入、レコードの削除、レコードの更新

これらのオプションにチェックをつけると、データが同期される時にターゲットに対してそのアクションを実行します。

構造の同期 (フルバージョンで使用可能)

Navicat では、詳細な分析処理で、テーブル構造を比較し変更することができます。つまり、Navicat は、2 つのデータベース/スキーマ間でテーブルを比較し、構造の違いを示します。メインメニューから**ツール** ->  **構造の同期**を選択します。

保存済みのプロファイルを開くためには、**プロファイル**タブで、プロファイルを選択して**ロード**ボタンをクリックするか、プロファイルをダブルクリックします。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能です。Navicat Premium と、Navicat for MySQL は、MySQL と MariaDB 間の同期に対応しています。

構造の同期の基本設定

以下のオプションは、選択されたデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、MariaDB。

ソース/ターゲット

ソースとターゲットの接続、データベース/スキーマを定義します。

テーブルを比較

ソースとターゲットのデータベース間でテーブルを比較したい場合、このオプションにチェックをつけます。

プライマリーキーを比較

テーブルのプライマリーキーを比較したい場合、このオプションにチェックをつけます。

外部キーを比較

テーブルの外部キーを比較したい場合、このオプションにチェックをつけます。

キャラクターセットを比較

テーブルのキャラクターセットを比較したい場合、このオプションにチェックをつけます。

自動増加値を比較

テーブルの自動増加値を比較したい場合、このオプションにチェックをつけます。

パーティションを比較

テーブルのパーティションを比較したい場合、このオプションにチェックをつけます。

UNIQUE 制約を比較

UNIQUE 制約を比較したい場合、このオプションにチェックをつけます。

チェック制約を比較

チェック制約を比較したい場合、このオプションにチェックをつけます。

除外制約を比較

除外制約を比較したい場合、このオプションにチェックをつけます。

ビューを比較

ビューを比較したい場合、このオプションにチェックをつけます。

ファンクションを比較

ファンクションを比較したい場合、このオプションにチェックをつけます。

イベントを比較

イベントを比較したい場合、このオプションにチェックをつけます。

定義者を比較

定義者を比較したい場合、このオプションにチェックをつけます。

インデックスを比較

インデックスを比較したい場合、このオプションにチェックをつけます。

シーケンスを比較

シーケンスを比較したい場合、このオプションにチェックをつけます。

トリガーを比較

トリガーを比較したい場合、このオプションにチェックをつけます。

ルールを比較

ルールを比較したい場合、このオプションにチェックをつけます。

テーブルスペースと物理属性を比較

テーブルスペースと物理属性を比較したい場合、このオプションにチェックをつけます。

オブジェクト作成用 SQL

新しいデータベースオブジェクトをターゲットに作成する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

オブジェクト変更用 SQL

ターゲットのデータベースオブジェクトを変更する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

オブジェクト削除用 SQL

ターゲットからデータベースオブジェクトを削除する場合、関連する全ての SQL 文を生成するために、このオプションにチェックをつけます。

CASCADE オプションで削除する

従属オブジェクトを削除するためにカスケードしたい場合、このオプションにチェックをつけます。

実行後に比較する

同期が実行された後、テーブルを比較します。

エラーが発生しても続ける

同期処理中に発生したエラーを無視します。

親を継承して作成する

同期処理中、親を継承したテーブルを作成します。

構造の同期の結果

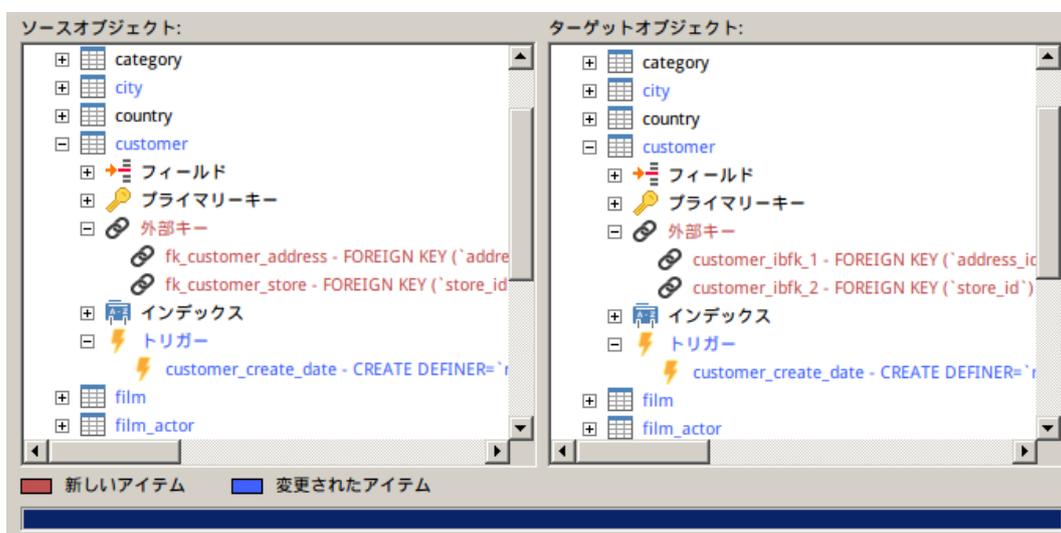
ソースオブジェクト/ターゲットオブジェクト

ツリービューには、構造の比較後、ソースとターゲットのデータベース/スキーマ間の差異が表示され、**修正用クエリ**リストに詳細な SQL ステートメントが表示されます。

赤色のアイテムは、他のデータベース/スキーマに存在しないことを表します。

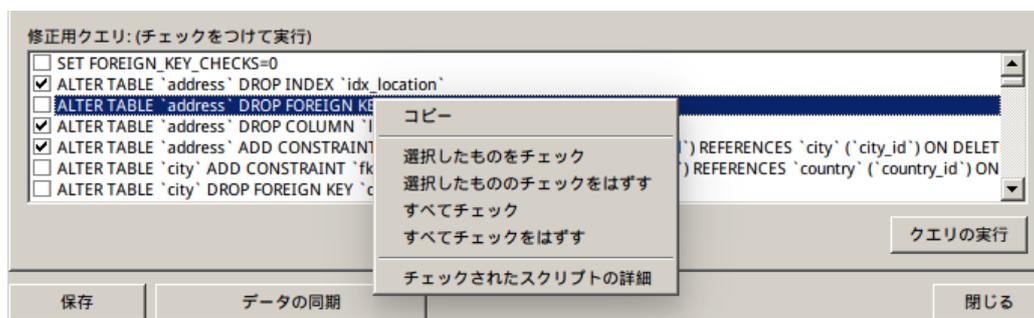
青色のアイテムは、他のデータベース/スキーマに存在するが、異なる定義が検出されたことを表します。

手動でオブジェクト構造を編集することができます。ツリービューでオブジェクトを右クリックし、ポップアップメニューから**編集**を選択すると、関連するデザイナーが開きます。



デフォルトで、**修正用クエリ**リストの全てのスクリプトにチェックはつけられていません。ターゲットに適用したいスクリプトにチェックをつけます。

複数行のスクリプトを強調表示し、右クリックしてポップアップメニューを表示することができます。選択したスクリプトの完全な SQL 文を見るためには、**チェックされたスクリプトの詳細**を選択します。



選択したクエリを実行するためには、**クエリの実行**ボタンをクリックします。

バックアップ/復元 (フルバージョンで使用可能)

安全で信頼性のあるサーバーは、定期的なバックアップの実行にしっかりと対応しています。それは、攻撃やハードウェア障害、人為的ミス、電源異常などが原因となり障害が発生する可能性があるためです。

Navicat では、データベースのデータベースオブジェクトをバックアップ/復元することができます。スケジュール設定のために、プロファイルとして設定を保存することができます。🗄️ をクリックして、**バックアップ**用のオブジェクトリストを開きます。

ヒント: バックアップファイルは、[設定の保存場所](#)の下に保存されます。

注意: MySQL、PostgreSQL、SQLite、MariaDB で使用可能です。Oracle のバックアップについては、[Oracle データポンプ](#)をご覧ください。

バックアップ

基本プロパティ

バックアップファイルに関する注釈を入力することができます。

オブジェクトの選択

バックアップしたいデータベースオブジェクトを選択します。

詳細プロパティ

以下のオプションは、選択されたデータベースの種類によって異なります: MySQL、PostgreSQL、SQLite、MariaDB。

圧縮

より小さなサイズのバックアップを作成したい場合、このオプションにチェックをつけます。

バックアップオブジェクトリストのバックアップファイルを圧縮または解凍するためには、バックアップファイルを右クリックし、ポップアップメニューから**バックアップを圧縮**または**バックアップを解凍**を選択します。

ヒント: 圧縮 (.psc)、解凍 (.psb)。

全テーブルをロック

バックアップ処理が進行している間、全てのオブジェクトにロックをかけます。

単一トランザクションを使用 (InnoDB のみ)

テーブルが InnoDB ストレージエンジン使用の場合にこのオプションにチェックをつけると、Navicat はバックアップ処理開始前にトランザクションを使用します。

指定したファイル名を使用

バックアップ用のファイル名を定義します。定義しない場合、バックアップファイルは、例えば、"2007-05-10 17:38:20"という名前がつけられます。

復元

復元機能は、まず最初に、選択されたデータベースオブジェクトを削除し、その後、ユーザーのバックアップに従って新しいオブジェクトを再作成します。最後に、データを挿入します。

現行のデータベースにバックアップを復元するためには、データベースを開き、既存のバックアップファイルを選択します。バックアップオブジェクトリストツールバーから  **バックアップから復元** をクリックします。

新しいデータベースにバックアップを復元するためには、新しいデータベースを作成して開き、バックアップオブジェクトリストツールバーから  **バックアップから復元** をクリックします。バックアップファイルを選択します。

注意: 復元を実行するためには、Create と Drop、Insert 権限 ([MySQL/MariaDB](#) または [PostgreSQL](#)) を持っていないとできません。

基本プロパティ

バックアップファイルの情報を表示します。

オブジェクトの選択

復元したいデータベースオブジェクトを選択します。

詳細プロパティ

以下のオプションは、選択されたデータベースの種類によって異なります: MySQL、PostgreSQL、SQLite、MariaDB。

トランザクションを使用

エラー発生時、全データをロールバックします。

エラーが発生しても続ける

復元処理中に発生したエラーを無視します。

書き込みに対してテーブルをロック

復元処理中に、ユーザーがテーブル修正できないように、テーブルにロックをかけます。

extended-insert ステートメントを使用

拡張 insert 構文を使用して、レコードを挿入します。

例:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindsy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

各実行時に複数クエリを実行

各実行時に複数のクエリを実行したい場合、このオプションにチェックをつけます。これによって復元処理がより早くなります。

テーブルを作成

このオプションを有効にすると、復元処理中にテーブルを作成します。

レコードを作成

このオプションを有効にすると、テーブルレコードを復元します。無効にすると、テーブル構造だけが復元されます。

インデックスを作成

このオプションを有効にすると、復元されたテーブルのインデックスを作成します。

トリガーの作成

このオプションを有効にすると、復元されたテーブルのトリガーを作成します。

既存のテーブルを上書き

データベース/スキーマにテーブルがすでに存在する場合、上書きします。

既存のビューを上書き

データベース/スキーマにビューがすでに存在する場合、上書きします。

既存のファンクションを上書き

データベース/スキーマにファンクションがすでに存在する場合、上書きします。

既存のイベントを上書き

データベース/スキーマにイベントがすでに存在する場合、上書きします。

既存のシーケンスを上書き

データベース/スキーマにシーケンスがすでに存在する場合、上書きします。

既存のインデックスを上書き

データベース/スキーマにインデックスがすでに存在する場合、上書きします。

既存のトリガーを上書き

データベース/スキーマにトリガーがすでに存在する場合、上書きします。

自動増加値の挿入

データベース/スキーマに自動増加値を挿入します。

SQL の抽出

SQL の抽出は、バックアップファイルから SQL ファイルに SQL を抽出することができます。

バックアップファイルから SQL を抽出するためには、データベースを開き、既存のバックアップファイルを選択します。バックアップオブジェクトリストツールバーから  **SQL を取り出す**をクリックします。

バッチジョブ/スケジュール (フルバージョンで使用可能)

Navicat では、バッチジョブを作成し、指定日付や時間に開始/終了し、定期的な間隔で 1 回以上そのバッチジョブを実行するためのスケジュールを設定することが可能です。バッチジョブは、データベースからのクエリやバックアップ、データ転送、データの同期、インポート、エクスポートに対して作成することができます。1 つのバッチジョブで実行される動作のリストを定義し、手動または指定された時間/定期的に、そのバッチジョブを実行できます。📅 をクリックして、**スケジュール**用のオブジェクトリストを開きます。

実行中プロセスの成功または失敗を示すメッセージは、ファイル - LogCmd.txt に保存されます。

🕒 **スケジュール設定**をクリックして、バッチジョブのスケジュールを設定します。

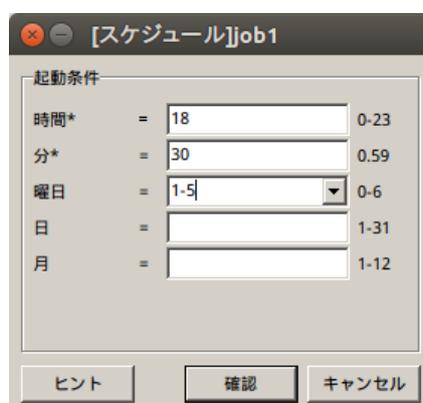
注意: スケジュールを設定する前に、バッチジョブを保存してください。スケジュールを起動する前に、[接続情報](#)にてパスワードが保存されている必要があります。

接続情報	
ユーザー名:	<input type="text" value="root"/>
パスワード:	<input type="password" value="*****"/>
	<input checked="" type="checkbox"/> パスワード保存

スケジュールの設定

パターンは、crontab エントリーの書式に従います。フィールドが値なしのままにされると、その値が使用されます。例えば、"weekday"フィールドが空の場合、システムはそのフィールドを"0, 1, 2, 3, 4, 5, 6"が入力されているものとして処理します。コンマを使用して値を分けます。例えば、"0, 1, 3, 6"。ハイフンを使用すると、スペースを使用せず値を表すことができます。例えば、"0-4"。

例: バッチジョブは、全ての平日の 6:30pm に実行されます。



起動条件		
時間*	=	<input type="text" value="18"/> 0-23
分*	=	<input type="text" value="30"/> 0-59
曜日	=	<input type="text" value="1-5"/> 0-6
日	=	<input type="text"/> 1-31
月	=	<input type="text"/> 1-12

ヒント 確認 キャンセル

Tips: crontab とは?

繰り返し実行される必要があるコマンドについて(つまり、時間毎、日毎、週毎)、crontab コマンドを使用することができます。crontab コマンドは、cron デーモンが実行するコマンドと命令を含む crontab ファイルを作成します。以下のオプションを使って、crontab コマンドを使用することができます:

crontab -a filename	crontab ファイルとして filename を設定します。多くのシステムで、このコマンドは、単に crontab filename で実行されます (つまり、-a オプション無しで)。
crontab -e	crontab ファイルを編集、または、まだ存在していない場合は作成します。
crontab -l	crontab ファイルを表示します。
crontab -r	crontab ファイルを削除します。
crontab -v	crontab ファイルを最後に編集した日時を表示します。(このオプションは、ほんの数システムでのみ使用可能です。)
crontab -u user	他のオプションと連動して使用されます。このオプションによって、ユーザーの crontab ファイルを修正または表示することができます。使用可能な場合、管理者のみこのオプションを使用することができます。

crontab ファイルの各エントリは、6 つのフィールドからなり、以下の順に指定します:

minute(s) hour(s) day(s) month(s) weekday(s) command(s)

フィールドは、スペースまたはタブで分けられています。最初の 5 つは、整数パターンで、6 番目は、実行するコマンドです。

フィールド	値	説明
minute	0-59	コマンドシーケンスが実行する日時の"分"
hour	0-23	コマンドシーケンスが実行する日時の"時"
day	1-31	コマンドシーケンスが実行する日時の"日"
month	1-12	コマンドシーケンスが実行する日時の"月"
weekday	0-6	コマンドシーケンスが実行する日時の"曜日" (日曜日 = 0、月曜日 = 1、火曜日 = 2 など)
command	特定	実行コマンドの完全なシーケンス。コマンド文字列は、 Bourne シェル 構文に準拠していなければなりません。コマンド、実行ファイル(スクリプトのような)、または、その併用が許容されます。

バッチジョブ/スケジュールの基本設定

 **選択**ボタンを使用する、または、ダブルクリックすることによって、**実行可能なジョブ**リストから**選択されたジョブ**リストへオブジェクトを移動します。同じ方法で、選択されたジョブリストからオブジェクトを削除することができます。1 つのバッチジョブ/スケジュールで、別々のサーバーからのプロファイルを実行することができます。

選択されたジョブの順番を並べ替えるには、**↑ 上に移動** または **↓ 下に移動** ボタンを使用します。

サーバー全体をバックアップするためには、接続を選択し、**Backup Server xxx** を選択します。(接続設定をバックアップするためには、[Navicat を新しいコンピュータに移動する](#)をご覧ください。)

エクスポートファイルをメール添付としてバッチジョブに追加することができます。**選択されたジョブ**リストのジョブを選択し、 **添付を追加** または  **添付を削除** をクリックして、メール添付を追加または削除します。

データ転送またはデータ同期のプロファイルを見つけるためには、左パネルの一番上の  **Navicat** を選択します。

バッチジョブ/スケジュールの詳細設定

メールを送信

Navicat では、スケジュールから返された結果付きの個別のメールを作成し送信することができます。結果セットを複数の受信者に E メール送信することが可能です。このオプションにチェックをつけ、必要な情報を入力します。

From

送信者の E メールアドレスを指定します。例えば、someone@navicat.com。

To, CC

各受信者の E メールアドレスを指定します。複数の場合、コンマまたはセミコロン(;)で区切ります。

サブジェクト

お好きな形式で E メール の件名を指定します。

ボディ

E メール の内容を書きます。

ホスト (SMTP サーバー)

送信メッセージ用 Simple Mail Transfer Protocol (SMTP) サーバーを入力します。

ポート

送信 E メール (SMTP) サーバーに接続するポート番号を入力します。デフォルト値は、**25** です。

認証を使用する

SMTP サーバーが E メールを送信するために認証を必要とする場合、このオプションにチェックをつけ、**ユーザー名とパスワード** を入力します。

安全な接続

接続が、**TLS**、**SSL** の安全な接続を使用する、または、**使用なし**を指定します。

テストメールを送信

Navicat は、成功または失敗を示すテストメールを送信します。

バッチジョブコンバーター (Navicat Premium で使用可能)

Navicat Premium は、Navicat for MySQL、Navicat for Oracle、Navicat for PostgreSQL、Navicat for SQLite、Navicat for MariaDB から、保存済みのバッチジョブを Navicat Premium にコンバートすることができます。オブジェクトリストペインで右クリックし、ポップアップメニューから**バッチジョブコンバーター**を選択します。

元のバッチジョブを削除

Navicat の元のバッチジョブを削除したい場合、このオプションにチェックをつけます。元のバッチジョブが削除されると、スケジュール設定済みのバッチジョブは、Navicat Premium または元のアプリケーションで再設定されるまで、動作しません。

既存のバッチジョブを上書き

Navicat Premium の既存のバッチジョブを上書きしたい場合、このオプションにチェックをつけます。

バッチジョブが存在する場合は追加

バッチジョブの詳細を Navicat Premium の既存のバッチジョブに追加したい場合、このオプションにチェックをつけ、既存のバッチジョブ名を入力します。

インポート処理を開始するためには、**開始**ボタンをクリックします。

SQL ファイルにダンプ/SQL ファイルの実行

Navicat では、**SQL ファイルにダンプ**と **SQL ファイルの実行**機能を使用して、データベース/スキーマ/テーブルのバックアップと復元を行うことができます。

データベース/スキーマ/テーブルをバックアップするためには、それを右クリックし、ポップアップメニューから **SQL ファイルにダンプ** -> **構造とデータ** または **構造のみ**を選択します。

データベース/スキーマ/テーブルを復元または SQL ファイルを実行するためには、右クリックし、ポップアップメニューから **SQL ファイルを実行する**を選択します。

ヒント: テーブルのオブジェクトリストペインまたは接続ペインのデータベース/スキーマに、.sql ファイルをドラッグすることができます。Navicat は、**SQL ファイルの実行**画面をポップアップで表示します。

データベース/スキーマ/テーブルの構造の印刷 (フルバージョンで使用可能)

Navicat では、データベース、スキーマ、テーブルの構造を表示、印刷することができます。データベース/スキーマ/テーブルを右クリックし、 **データベースを印刷する** または  **スキーマを印刷する** または  **テーブルを印刷する**を選択します。

サーバーセキュリティ

Navicat は、サーバーのセキュリティ管理ツールを提供しています。選択されたデータベースとそのデータベースオブジェクトに関して、ユーザーの追加や編集、削除、権限の付与や取り消しを行えます。👤 をクリックして、**ユーザー用のオブジェクトリスト**を開きます。オブジェクトリストペインはサーバーに存在する全てのユーザーを表示します。

MySQL/MariaDB セキュリティ

ユーザー権限に関する情報は、**mysql** データベース(つまり、mysql という名前のデータベース)の **user**、**db**、**host**、**tables_priv**、**columns_priv**、**procs_priv** テーブルに保存されます。MySQL サーバーは、起動時にこれらのテーブルの内容を読み込みます。

サーバーに接続するクライアントプログラムを実行する時、MySQL のアクセスコントロールは 2 つの段階を必要とします：

段階 1: サーバーは、あなたがサーバーに接続することを許可すべきかどうかを確認します。

段階 2: あなたが接続可能であると仮定すると、サーバーは、あなたが発行するそれぞれの命令文を確認し、あなたがその命令文の実行権限を満たしているかどうかを判定します。例えば、テーブル作成権限、テーブル削除権限、テーブル修正権限。

サーバーは、アクセスコントロールの両段階で、**mysql** データベースの **user**、**db**、**host** テーブルを使用します。

MySQL/MariaDB ユーザーデザイナー

基本プロパティ

ユーザー名

ユーザーの名前を設定します。

ホスト

データベースが配置されているホスト名、または、サーバーの IP アドレス。

パスワード

パスワードを設定し、**パスワード確認**テキストボックスにもう一度入力します。

詳細プロパティ

毎時ごとに最大実行されるクエリ、毎時ごとに最大実行される更新、毎時ごとの最大接続数

これらのオプションは、1 時間でユーザーが実行することができるクエリやアップデート、ログインの数を制限します。0 に設定されている場合 (デフォルト)、そのユーザーに対しては制限がないことを意味します。

最大ユーザー接続数

このオプションは、アカウントが作成できる最大同時接続数を制限します。0 に設定されている場合 (デフォルト)、**max_user_connections** システム変数がアカウントの同時接続数を決定します。

OLD_PASSWORD オプションの暗号化を使う

パスワードハッシングメカニズムは、より良いセキュリティを提供するために、そして、パスワードがインターセプトされるとい
うリスクを削減するために、MySQL 4.1 で更新されました。しかしながら、この新しいメカニズムは、MySQL 4.1(以上)のサー
バーとクライアントでのみ理解されるので、互換性の問題をもたらす可能性があります。4.1 以上のクライアントは新旧のパスワ
ードハッシングメカニズムの両方を理解するので、4.1 より古いサーバーに接続できます。しかしながら、4.1 より古いクライ
アントが 4.1 以上のサーバーに接続しようとする、問題が起きるかもしれません。

サーバーが長いパスワードハッシュを生成する状況で、4.1 より古いクライアントとの下位互換性を維持したい場合は、このオプ
ションを有効にします。オプションは認証に影響しません (4.1 以上のクライアントは、今でも、長いパスワードハッシュを持つ
アカウントを使用することができます) が、パスワードの変更操作で *user* テーブルに長いパスワードハッシュを作成することがで
きなくなります。

SSL タイプ

MySQL は、ユーザー名とパスワードに基づく標準認証に加えて、X509 認証属性を確認することができます。MySQL アカウント
に関する SSL 関連オプションを指定するためには、*GRANT* 文の *REQUIRE* 節を使用します。

ANY	このオプションは、アカウントに対して SSL 暗号化接続のみを許可することを、サーバーに伝えます。
X509	これは、クライアントが有効な証明書を持っていないが、正確な証明書、発行人、対象は問題ではな いことを意味します。唯一必要なことは、CA 証明書のひとつとその署名が照合可能でなければならないことです。
SPECIFIED	発行者 クライアントが CA 発行者から発行された有効な X509 証明書を提示しなければならない接続を制限します。有効 だが発行者が異なる証明書をクライアントが示すと、サーバーは接続を拒否します。X509 証明書の使用は常に暗 号化を意味しますので、この場合 SSL オプションは不必要です。 サブジェクト クライアントがサブジェクト <i>subject</i> を含む有効な X509 証明書を提示しなければならない接続を制限します。有 効だが対象が違う証明書をクライアントが示すと、サーバーは接続を拒否します。 暗号 十分な強度の暗号文とキーの長さが確実に使用されるようにするために必要です。短い暗号キーを用いた古いアル ゴリズムが使用されている場合、SSL 自体は弱くなる可能性があります。このオプションを使用すると、指定する 暗号方法が接続許可に使用されることを要求できます。

サーバー権限

グリッドで、**権限**に一覧表示されるサーバー権限に対して**付与**オプションを選択することにより、そのユーザーがその権限を持つ
ことを指定できます。複数の権限を付与することが可能です。

権限

ユーザーの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実
行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。

2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。
3. グリッドで、**権限**に一覧表示される権限に対して**状態オプション**を選択することにより、そのユーザーがその権限を持つことを指定できます。複数の権限を付与することが可能です。

Oracle セキュリティ

Oracle は、ユーザーとロールを使用してデータベースのアクセス許可を管理します。ユーザーはスキーマオブジェクト(例えば、テーブル、ビュー)を所有しており、どのユーザーがどのオブジェクトにアクセスするのかを制御するために、そのオブジェクトに対する権限を他のユーザーに割り当てることができます。

作成するユーザーアカウントに加えて、データベースにはインストール時に自動作成される多くのユーザーが含まれます。管理アカウント: **SYS**、**SYSTEM**、**SYSMAN**、**DBSNMP** です。管理アカウントは、データベースの開始と終了、データベースのメモリとストレージの管理、データベースユーザーの作成と管理などの管理タスクを行なうための特権アカウントです。データベースには、サンプルスキーマ(**SCOTT**、**HR**、**OE**、**OC**、**PM**、**IX**、**SH**)も作成されているかもしれません。それらは相互に関連したスキーマで、一般的なデータベースタスクを説明するための Oracle ドキュメントと Oracle 教材です。

データベースオブジェクトを作成すると、作成者はその所有者になります。デフォルトでは、オブジェクトの所有者のみがオブジェクトに対して様々な操作を行なえます。他のユーザーへ使用権限を与えるには、権限を付与しなければなりません。(ただし、スーパーユーザー属性を持つユーザーは、常にすべてのオブジェクトにアクセスできます)。

通常、オブジェクト所有者(もしくはスーパーユーザー)のみがオブジェクトの権限の付与、もしくは取消を行なうことができます。しかし、**Admin オプション/Grant オプション**権限を付与することによって、この権限を付与された人に、他者に対して権限を付与することが可能な権利を与えることができます。その後、権限オプションが取り消された場合は、(直接、もしくは権限の連鎖を通して)権限を付与された人から権限を与えられたすべてのユーザーは、その権限を失います。

注意: 特別な名前 **PUBLIC** は、全てのデータベースユーザーが使用可能です。そして、**PUBLIC** に付与される全ての権限とロールは、全てのデータベースユーザーが使用可能です。

Oracle ユーザーデザイナー

基本プロパティ

ユーザー名

ユーザーの名前を設定します。

認証

認証方法を選択します。

パスワード	<p>パスワード パスワードを設定し、パスワード確認テキストボックスにもう一度入力します。</p> <p>パスワードを失効 ユーザーのパスワードを失効します。この設定は、ユーザーまたは DBA に対してデータベースにログインする前にパ</p>
-------	--------------------------------------------------------------------------------------------------------------------------------------------------------

	スワードの変更を強制します。
外部	外部ユーザーは、オペレーティングシステムや第三者サービスのような外部サービスによって認証されなければなりません。
グローバル	グローバルユーザーは、エンタープライズディレクトリサービス(Oracle Internet Directory)によって認証されなければなりません。 X.500 名 このユーザーを識別するエンタープライズディレクトリサービスでの X.509 名を入力します。

デフォルトテーブルスペース

ユーザーが作成するオブジェクトのデフォルトテーブルスペースを選択します。

一時テーブルスペース

ユーザーの一時セグメントのテーブルベースもしくはテーブルスペースグループを選択します。

プロファイル

ユーザーに割り当てるプロファイルを選択します。

アカウントをロック

ユーザーアカウントをロックし、アクセス不可にします。

どこのメンバーか

グリッドで、**ロール名**に一覧表示されるロールに対して、**付与**、**Admin オプション**または**デフォルトとして**を選択することにより、そのユーザーが選択されたロールのメンバーであることを指定できます。複数のロールを付与することが可能です。

クォータ

グリッドで、そのユーザーがテーブルスペース内で割り当てることができるスペースの最大容量を指定します。**クォータ**を入力し、**テーブルスペースの単位**を選択します。**無制限**を使用すると、ユーザーはテーブルスペース内のスペースを限りなく割り当てることができます。複数のテーブルスペースに設定することができます。

サーバー権限

グリッドで、**権限**に一覧表示されるサーバー権限に対して**付与**または**Admin オプション**オプションを選択することにより、そのユーザーがその権限を持つことを指定できます。複数の権限を付与することが可能です。

権限

ユーザーの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。

3. グリッドで、**権限**に一覧表示される権限に対して**付与**または **Grant オプション** オプションを選択することにより、そのユーザーがその権限を持つことを指定できます。複数の権限を付与することが可能です。

Oracle ロールデザイナー

基本プロパティ

ロール名

ロールの名前を設定します。

認証

認証方法を選択します。

パスワード	パスワード パスワードを設定し、 パスワード確認 テキストボックスにもう一度入力します。
外部	外部ユーザーは、ロールを有効にする前に、オペレーティングシステムや第三者サービスなどの外部サービスによって認証されなければなりません。
グローバル	グローバルユーザーは、ロールがログイン時に有効となる前に、エンタープライズディレクトリサービスによって、ロールの使用を認証されなければなりません。
匿名	ロールは、データベースによって認証されます。ロールを有効化するために、パスワードは要求されません。

どこのメンバーか

グリッドで、**ロール名**に一覧表示されるロールに対して**付与**または **Admin オプション**を選択することにより、そのロールが選択されたロールのメンバーであることを指定できます。複数のロールを付与することが可能です。

メンバー

グリッドで、**メンバー**に一覧表示されるユーザーに対して**付与**または **Admin オプション**を選択することにより、選択されたユーザーがそのロールのメンバーであることを指定できます。複数のユーザーに付与することが可能です。

サーバー権限

グリッドで、**権限**に一覧表示されるサーバー権限に対して**付与**または **Admin オプション** オプションを選択することにより、そのロールがその権限を持つことを指定できます。複数の権限を付与することが可能です。

権限

ロールの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。

3. グリッドで、**権限**に一覧表示される権限に対して**付与オプション**を選択することにより、そのロールがその権限を持つことを指定できます。複数の権限を付与することが可能です。

PostgreSQL セキュリティ

PostgreSQL は、ユーザーやグループを使用してデータベースアクセス許可を管理します。ユーザーはデータベースオブジェクト(例えばテーブル)を所有し、他のユーザーにそれらのオブジェクトに対する権限を割り当てることによって、だれがどのオブジェクトにアクセスできるかを制御することができます。

注意: PostgreSQL version 8.1 以降、はっきりと区別された種類のエンティティではなかったユーザーとグループが、ロールのみになりました。ロールは、ユーザー、グループ、両方として機能することができます。ロールの概念は、ユーザーとグループの概念を含みます。

ユーザーの追加/削除を行うことができるのは、スーパーユーザー(全ての権限を許可されているユーザー)だけです。PostgreSQL は、デフォルトで **postgres** という名前の 1 つのスーパーユーザーをインストールします。このユーザーまたは後で追加された別のスーパーユーザーが、他の全ユーザーを追加しなければなりません。

データベースオブジェクトを作成する時、あなたがその所有者になります。デフォルトで、オブジェクトの所有者だけがそのオブジェクトに対して何でも行うことができます。他のユーザーがそれを使用するためには、権限が付与されなければなりません。(しかしながら、スーパーユーザー属性を持つユーザーは常にどのオブジェクトにもアクセス可能です。)

通常、そのオブジェクトの所有者(またはスーパーユーザー)だけが、オブジェクトに対する権限の付与/取り消しを行うことができます。しかし、**Grant オプション付き**で権限を付与することが可能で、その権限の受け手は他の人に権限を付与する権利を同様に与えられることとなります。Grant オプションがその後取り消されると、その受け手から(直接または付与のつながりを通して)権限を受けた全員がその権限を失います。

注意: 特別な名前 **public** を使用すると、システム上のどのロール(ユーザー/グループ)にも権限を付与することができます。

PostgreSQL Server 7.3 ~ 8.0

PostgreSQL バージョン 7.3~8.0 は、**ユーザー**と**グループ**を使用してデータベースアクセス許可を管理します。

PostgreSQL ユーザーデザイナー

基本プロパティ

ユーザー名

ユーザーの名前を設定します。

ユーザーID

ユーザーの ID を指定します。これは、通常必要ではありませんが、オブジェクトの所有者を再作成する必要が生じた場合に役立

ちます。これを指定しない場合、デフォルトで、割り当て済みの最大ユーザーIDに1を加えたID(100を最小とする)が使用されます。

パスワード

パスワードを設定し、**パスワード確認**テキストボックスにもう一度入力します。

注意: パスワード認証を使用する予定がない場合、このオプションを省略することができます。しかし、パスワード認証に切り替えることを決定した場合、ユーザーは接続することができなくなります。

パスワードの暗号化

このオプションは、**ENCRYPTED**(暗号化)か**UNENCRYPTED**(非暗号化)のどちらでパスワードをシステムカタログ内に保存するかを制御します。(どちらも指定しない場合、デフォルトの動作は設定パラメータ `password_encryption` によって決まります。)

有効期限

ユーザーのパスワードが有効でなくなる日時を設定します。この節が省略されると、パスワードはいつまでも有効となります。

スーパーユーザー

ユーザーをスーパーユーザーとして定義する場合、このオプションにチェックをつけます。

データベースを作成可能

データベースの作成を許可されたユーザーを定義する場合、このオプションにチェックをつけます。

どこのメンバーか

グリッドで、**グループ名**に一覧表示されるグループに対して**付与**オプションを選択することにより、そのユーザーが選択されたグループのメンバーであることを指定できます。複数のグループを付与することが可能です。

権限

ユーザーの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。
3. グリッドで、**権限**に一覧表示される権限に対して**付与**または**Grant オプション**オプションを選択することにより、そのユーザーがその権限を持つことを指定できます。複数の権限を付与することが可能です。

PostgreSQL グループデザイナー

基本プロパティ

グループ名

グループの名前を設定します。

グループ ID

グループの ID を指定します。これは、通常必要ではありませんが、オブジェクトの許可で参照されるグループを再作成する必要が生じた場合に役立ちます。これを指定しない場合、デフォルトで、割り当て済みの最大グループ ID に 1 を加えた ID(100 を最小とする)が使用されます。

メンバー

グリッドで、**メンバー**に一覧表示されるユーザーに対して**付与オプション**を選択することにより、選択されたユーザーがそのグループのメンバーであることを指定できます。複数のユーザーに付与することが可能です。

権限

グループの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実行します：

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。
3. グリッドで、**権限**に一覧表示される権限に対して**付与オプション**を選択することにより、そのグループがその権限を持つことを指定できます。複数の権限を付与することが可能です。

PostgreSQL Server 8.1 以降

PostgreSQL version 8.1 以降、はっきりと区別された種類のエンティティではなかったユーザーとグループが、**ロールのみ**になりました。ロールは、ユーザー、グループ、両方として機能することができます。ロールの概念は、ユーザーとグループの概念を含みます。

PostgreSQL ロールデザイナー

基本プロパティ

ロール名

ロールの名前を設定します。

ロール ID

ロールの ID を指定します。これは、通常必要ではありませんが、オブジェクトの所有者を再作成する必要が生じた場合に役立ちます。これを指定しない場合、デフォルトで、割り当て済みの最大ロール ID に 1 を加えた ID(100 を最小とする)が使用されます。

注意: PostgreSQL version 8.1 以降では、指定された ID は無視されますが、下位互換性のために受け入れられます。

ログイン可能

ログインを許可するロールを作成する場合、このオプションにチェックをつけます。このオプションがあるロールは、ユーザーとしてみなされます。この属性のないロールは、データベース権限の管理に使用できますが、言葉の普通の意味においてユーザーではありません。

パスワード

パスワードを設定し、**パスワード確認**テキストボックスにもう一度入力します。

注意: パスワード認証を使用する予定がない場合、このオプションを省略することができます。しかし、パスワード認証に切り替えることを決定した場合、ロールは接続することができなくなります。

パスワードの暗号化

このオプションは、**ENCRYPTED**(暗号化)か **UNENCRYPTED**(非暗号化)のどちらでパスワードをシステムカタログ内に保存するかを制御します。(どちらも指定しない場合、デフォルトの動作は設定パラメータ `password_encryption` によって決まります。)

接続制限

ロールがログインできる場合、ロールが作成可能な同時接続数を指定します。-1(デフォルト)は、制限がないことを意味します。

有効期限

ロールのパスワードが有効でなくなる日時を設定します。この節が省略されると、パスワードはいつまでも有効となります。

スーパーユーザー

新しいロールが、データベース内の全てのアクセス制限をオーバーライドすることができるスーパーユーザーであることを決定する場合、このオプションにチェックをつけます。

データベースを作成可能

ロールがデータベースを作成可能であることを定義する場合、このオプションにチェックをつけます。

ロールを作成可能

このオプションにチェックをつけると、ロールを作成することが可能になります。

権限を継承

ロールが、あるロールのメンバーとなっていて、そのロールの権限を継承する場合、このオプションにチェックをつけます。

システムカタログを更新可能

このオプションにチェックをつけると、ロールがシステムカタログを更新することが可能になります。

どこのメンバーか

グリッドで、**ロール名**に一覧表示されるロールに対して**付与**または **Admin オプション**を選択することにより、そのロールが選択されたロールのメンバーであることを指定できます。複数のロールを付与することが可能です。

メンバー

グリッドで、**メンバー**に一覧表示されるロールに対して**付与**または **Admin オプション**を選択することにより、選択されたロールがそのロールのメンバーであることを指定できます。複数のロールに付与することが可能です。

権限

ロールの特定のオブジェクト権限を編集するためには、 **権限の追加**をクリックしてウィンドウを開き、以下のステップを実行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. そのオブジェクトにチェックをつけると、右パネルにグリッドが表示されます。
3. グリッドで、**権限**に一覧表示される権限に対して**付与**または **Grant オプション** オプションを選択することにより、そのロールがその権限を持つことを指定できます。複数の権限を付与することが可能です。

SQLite セキュリティ

デフォルトで、SQLite データベースはユーザー認証を必要としません (no-authentication-required database)。ユーザーを作成すると、データベースは認証が必要なデータベース (authentication- required database) としてマークされます。その後は、データベースファイルに接続するとき、ユーザーはユーザー名とパスワードを提供する必要があります。

SQLite ユーザーデザイナー

ユーザー名

ユーザーの名前を設定します

パスワード

パスワードを設定し、**パスワード確認** テキストボックスにもう一度入力します。

アドミニストレーター

ユーザーに admin 権限を与える場合、このオプションにチェックをつけます。

権限マネージャー

権限マネージャーは、各ユーザーの権限の設定以外に、接続とそのデータベースオブジェクトに対する権限の設定も行えます。

注意: MySQL/Oracle/PostgreSQL/MariaDB で使用可能です。

ユーザーオブジェクトリストツールバーから  **権限マネージャー** をクリックして、以下のステップを実行します:

1. ターゲットオブジェクトにたどり着くまで、ツリービューのノードを開きます。
2. オブジェクトを選択し、 **権限の追加** をクリックしてウィンドウを開きます。
3. そのユーザーにチェックをつけると、右パネルにグリッドが表示されます。
4. グリッドで、**権限**に一覧表示される権限に対して該当する権限を選択することによって、選択されたユーザーがそのオブジェクト権限を持つことを指定できます。複数の権限を付与することが可能です。

便利なツール

Navicat は、Navicat をより便利に使えるように、様々なツールを用意しています。例えば、**オブジェクト情報**、**接続の色付け**、**検索フィルター**など。

リスト/詳細/ER ダイアグラム ビュー

Navicat は、メインウィンドウのオブジェクトに関して、3種類のビューを提供しています。デフォルトで、Navicat はオブジェクトリストペインの**リストビュー**を使用します。このビューはオブジェクト名のみを表示します。メインメニューから**表示 -> リスト**を選択するか、右下の  をクリックします。

詳細ビューは、列のオブジェクトのプロパティを表示します。詳細ビューに変更するためには、メインメニューから**表示 -> 詳細**を選択するか、右下の  をクリックします。

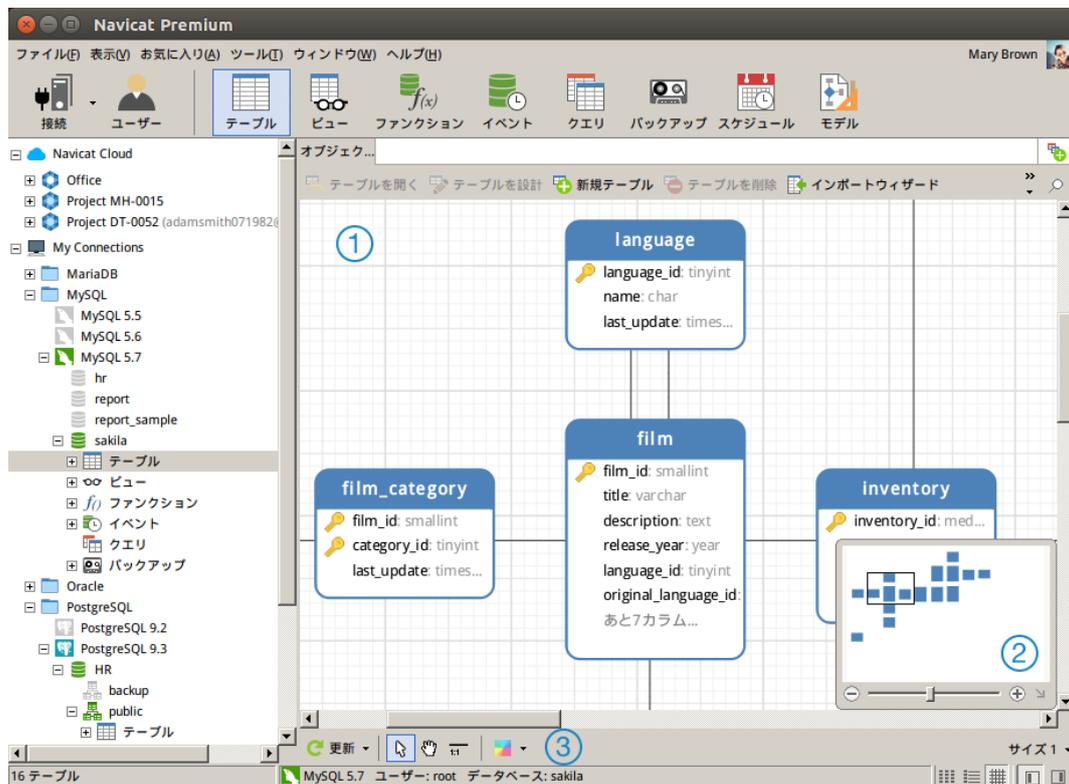
プロパティの表示列を変更するためには、メインメニューから**表示 -> カラムの選択**を選択し、ポップアップウィンドウからそれぞれのオブジェクトの表示列を選択します。

リストと詳細ビュー以外に、Navicat はテーブル表示の新しい **ER ダイアグラム**ビューに対応しました。この ER ダイアグラムビューでは、データベース/スキーマのテーブルフィールドやテーブル間のリレーションを図で見ることができます。また、テーブルに外部キー制約を直接追加することもできます。

注意: フルバージョンで使用可能。ER ダイアグラムビューはテーブルのみに対応しています。他のデータベースオブジェクトはリストビューと詳細ビューのみです。

メインメニューから**表示 -> ER ダイアグラム**を選択するか、右下の  をクリックします。選択したデータベース/スキーマにテーブルが含まれる場合、ER ダイアグラムは自動的に作成されます。

ヒント: ER ダイアグラムファイルは、[設定の保存場所](#)に保存されます。



① オブジェクトリスト

データベース/スキーマのテーブルフィールドとテーブル間のリレーションを表示します。

注意: ER ダイアグラムにあるテーブルをダブルクリックすると、テーブルデザイナーが開きます。その一方、リストビューや詳細ビューにあるテーブルをダブルクリックすると、テーブルビューが開きます。デザイナーに表示されるタブとオプションは、選択されているダイアグラムデータベースタイプによって異なります。それぞれのタブの設定に関しては、[サーバーオブジェクト](#)を参照してください。

リレーションを追加するためには、下のツールバーから  をクリックします。ソーステーブルフィールドをドラッグし、ターゲットテーブルフィールドにドロップします。

リレーションを編集するためには、リレーションを右クリックし、ポップアップメニューから**外部キーを設計**を選択します。

リレーションを削除するためには、リレーションを右クリックし、ポップアップメニューから**外部キーを削除**を選択します。

リレーションに頂点を追加するためには、リレーションを右クリックし、ポップアップメニューから**頂点を追加**を選択します。

リレーションの1つまたは全ての頂点を削除するためには、リレーションを右クリックし、ポップアップメニューから**頂点を削除**または**全頂点を削除**を選択します。

② 概要

ダイアグラムの選択領域をズームインまたはズームアウトするためには、概要ペインのスライダーで調整します。キーボードショートカットを使って同様の効果を得ることができます：

ズームイン: [CTRL++] または [CTRL+Mousewheel up]

ズームアウト: [CTRL+-] または [CTRL+Mousewheel down]

③ ツールバー

更新

クリックして ER ダイアグラムを更新します。

ER ダイアグラムを再生成

 更新ボタンから **ER ダイアグラムを再生成** を選択します。これは、自動レイアウト機能を使用して ER ダイアグラムを再作成します。

ダイアグラムを移動

クリックして、ハンドモードに切り替えます。SPACE キーを押したままの状態にし、ダイアグラムを移動します。

新規リレーション

クリックして、2 つのテーブルフィールド間にリレーションを作成します。

色

選択テーブルとリレーションに色を設定します。

用紙サイズ

ドロップダウンリストから用紙サイズを選択します。対応する用紙サイズが概要ペインに反映されます。

オブジェクト情報

オブジェクトリストペインでは、選択オブジェクトの情報を見ることも可能です。メインメニューから **表示** -> **オブジェクト情報** を **表示** を選択するか、オブジェクトを右クリックしてポップアップメニューから **オブジェクト情報** を選択します。

注意: タブは、選択されているオブジェクトタイプによって異なります。

基本

オブジェクト情報を表示します。

DDL

オブジェクトの DDL 文を表示します。

使用先

現在のオブジェクトが使用したオブジェクトを表示します。

使用元

現在のオブジェクトがどのオブジェクトによって使用されたかを表示します。

オブジェクト

テーブルスペースにあるオブジェクトを表示します。

プレビュー

クエリの SQL 文を表示します。

どこのメンバーか

ユーザーまたはロールが割り当てられているロールを表示します。

メンバー

ロールのメンバーを表示します。

サーバー監視 (フルバージョンで使用可能)

Navicat は、選択サーバーのプロパティを閲覧できる**サーバー監視**を提供しています。**ツール** ->  **サーバー監視**を選択し、メニューからサーバータイプを選択します。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

プロセスリスト

選択された全てのサーバーからのプロセスを一覧表示します。

選択されたプロセスを停止するためには、 **プロセスを終了**ボタンをクリックします。

自動更新

指定した秒でサーバーを自動更新するという処理を行いたい場合、**表示** -> **自動更新時間を設定**を選択し、自動更新の値を入力します。自動更新機能を無効にするためには、**表示** -> **自動更新**を選択します。

注意: 値を指定すると、すぐにその値が有効になります。

プロセスリストは、以下の情報を提供します。情報は選択されたデータベースの種類によって異なります: MySQL、Oracle、PostgreSQL、MariaDB。

- 接続設定時に与えられたサーバー名。
- サーバー上のプロセス ID。
- プロセスのシリアル番号。
- サーバーにログインしている現在のユーザー。
- ユーザーが接続している接続元ホスト。
- ユーザーが現在使用しているデータベース。
- ユーザーが最後に発行したコマンド。
- プロセスの時間、状態、情報。
- プロセスの CPU 時間と状態。

変数

全てのサーバー変数とその値を一覧表示します。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

ヒント: MySQL と Oracle サーバーの変数の値を編集するためには、 をクリックするか、CTRL+ENTER を押し、編集用のエディタを開きます。PostgreSQL サーバーの値をここで編集することはできません。(設定ファイル `postgresql.conf` を編集することにより、SET 文を使用してそれらの変数を設定することができます。)

状態

全てのサーバー状態とその値を一覧表示します。

注意: MySQL、Oracle、PostgreSQL、MariaDB で使用可能。

バーチャルグルーピング (フルバージョンで使用可能)

バーチャルグループの目的は、全てのオブジェクトを効率的に維持するために、カテゴリによってオブジェクトを論理的に分類するためのプラットフォームを提供することです。

バーチャルグルーピングは、接続、テーブル、ビュー、ファンクション、クエリ、バックアップ、スケジュールとモデルに適用されます。

新しいグループを作成するためには、接続ペイン/オブジェクトリストペインで右クリックし、ポップアップメニューから**新規グループ**または**グループを管理** -> **新規グループ**を選択します。

オブジェクトをグループに移動するためには、オブジェクトを右クリックし、ポップアップメニューから**グループを管理** -> **移動**を選択するか、オブジェクトをグループにドラッグアンドドロップします。

オブジェクトを最上層に戻すためには、オブジェクトを右クリックし、ポップアップメニューから**グループを管理** -> **グループから外す**を選択するか、オブジェクトを接続ペインの目的のレベルにドラッグアンドドロップします。

メインメニューから**表示** -> **接続を折りたたむ/オブジェクトリストを折りたたむ**を選択して、接続/オブジェクトグループを隠すことができます。

接続の色付け

Navicat は、接続とそのデータベースオブジェクトを識別するために、色によって接続を強調表示することができます。強調表示色は、接続ペインとそのデータベースオブジェクトウィンドウのメニューバーに表示されます。

接続を強調表示するためには、接続を右クリックし、ポップアップメニューから**色**を選択します。

お気に入り (フルバージョンで使用可能)

お気に入りは、頻繁にアクセスするデータベースオブジェクトへのリンクです。お気に入りリストにパスを追加することによって、接続ペインの接続やデータベース/スキーマを操作することなく、1 クリックでデータベースオブジェクトにアクセスすることが可能です。

お気に入りリストにリンクを追加するためには、データベースオブジェクトを開き、**ファイル** -> **お気に入り**に**追加**を選択するか、SHIFT+CTRL+#を押します。データベースオブジェクトがタブウィンドウで開かれている場合、そのタブを右クリックし、ポップアップメニューから**お気に入り**に**追加**を選択することも可能です。**お気に入り** **名前**を入力し、**お気に入り** **ID**を選択します。

お気に入りリストからデータベースオブジェクトを開くためには、メインメニューから**お気に入り** -> **お気に入りの名前**を選択するか、Ctrl+#を押します。

お気に入りリストからリンクを削除するためには、メインメニューから**お気に入り** -> **お気に入りをクリア** -> **お気に入りの名前**を選択します。

お気に入りリストから全てのリンクを削除するためには、メインメニューから**お気に入り** -> **お気に入りをクリア** -> **全てをクリア**を選択します。

注意: #は、0、1、2、3、4、5、6、7、8または9を表します。

データベース/スキーマ内検索 (フルバージョンで使用可能)

Navicat は、データベース/スキーマ内のテーブルとビューのレコードまたはオブジェクト構造を検索する**データベース/スキーマ内検索**機能を提供しています。メインメニューから**ツール** -> **データベース/スキーマ内検索**を選択します。

1. ターゲットの**接続**、**データベース**や**スキーマ**を選択します。
2. **検索文字**に検索文字列を入力します。
3. データまたは構造のどちらを検索するか、**検索対象**ドロップダウンリストから選択します。
4. **検索モード**を選択します: 含む、単語全体、前方、または、正規表現。正規表現は、構造を検索する時のみ選択可能です。
5. 必要に応じて、大文字小文字を区別して検索することを無効にするために**大文字小文字を区別しない**にチェックをつけます。
6. 構造を検索する場合、検索するオブジェクトを選択することができます: テーブル、ビュー、ファンクション、クエリ、インデックス、トリガー、イベント、マテリアライズド・ビュー。
7. **検索ボタン**をクリックし、その後、**検索結果**リストのオブジェクトをダブルクリックして、そのレコードまたは構造を見ることができます。

検索フィルター

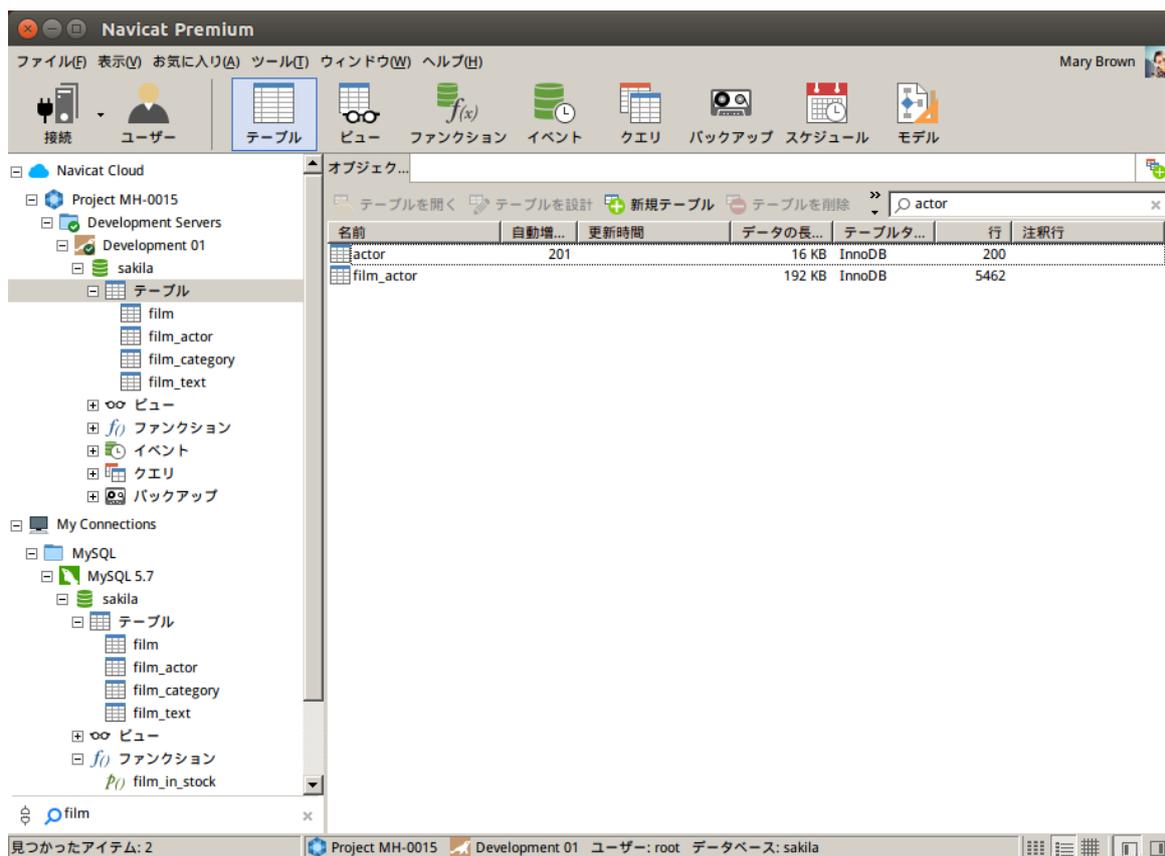
Navicat は、接続ペインやオブジェクトリストペイン、モデルデザイナーウィンドウ、他のツリー構造のオブジェクトを検索するための検索フィルター機能を提供します。

接続ペインまたは他のツリー構造では、ペインまたはツリーをクリックして焦点を合わせ、フィルター文字列を直接入力します。接続ペインで接続が開いている場合、フィルターはそのデータベースオブジェクトにも適用されます。

オブジェクトリストペインでは、Navicat メインウィンドウの  をクリックし、検索ボックスにフィルター文字列を入力します。

モデルデザイナーウィンドウでは、検索ボックスにフィルター文字列を入力するだけです。

フィルター文字列を削除することによって、フィルターを取り消すことができます。



オプション

Navicat では、全てのツールの様々なオプションを設定することにより、ユーザーインターフェースを完全にカスタマイズすることができます。メインメニューから、**ツール -> オプション**を選択します。

基本オプション

複数のフォームインスタンスを許可

このオプションを有効にすると、同じものが選択された時にそのウィンドウのインスタンスを複数開くことができます。

複数の Navicat インスタンスを許可

このオプションにチェックをつけない場合、Navicat のショートカットをクリックすると、起動中の Navicat インスタンスが再びアクティブとなります。新しいコピーは起動されません。

クリックで更新

オブジェクトをクリックする時はいつでも、そのオブジェクトリストペインのリストを更新します。

ファンクションウィザードを表示

ファンクション/プロシージャの新規作成時、ファンクションウィザード([MySQL/MariaDB](#)、[Oracle](#)、[PostgreSQL](#))を表示します。

閉じる前に新規のクエリ/プロファイルを保存するか尋ねる

このオプションを有効にすると、Navicat は、クエリまたはプロファイル用のサブウィンドウを終了する時はいつも、新しいクエリまたはプロファイルを保存するように促します。

Windows

ウィンドウを開く

新しいウィンドウを、**メインウィンドウに/開いているウィンドウに/新しいウィンドウとして**開きます。

コード入力支援(フルバージョンで使用可能)

コード補完を使用

オブジェクト名の中に . (ドット)を入力すると、SQL エディタは、コードを補完するための様々な候補を表示したポップアップリストを表示します。[コード補完](#)を参照してください。

遅延

ポップアップリストが表示されるまでの時間を変更できます。

単語補完を使用

単語の最初の文字を入力すると、SQL エディタは、単語を補完するための様々な候補を表示したポップアップリストを表示します。

遅延

ポップアップリストが表示されるまでの時間を変更できます。

構文の強調表示を使用

構文の強調表示は、コードを見やすく表示するのに役立ちます。SQL エディタで、コードは、属するカテゴリーに応じて、異なる色とフォントで強調表示されます。パフォーマンス向上のために、**次の値より小さいサイズのステートメントに対して構文強調表示を適用(MB)**に最大ファイルサイズ(例えば 10)を設定することによって、構文の強調表示機能を制限することが可能です。

自動保存

自動保存を使用

自動保存の間隔(s)を定義することによって(例えば 30)、SQL エディタでの修正後、その修正を自動的に保存します。

表示オプション

ツールバーの説明を表示

サブウィンドウにあるツールバーのボタンにテキストを表示します。チェックをつけない場合、ボタンだけが表示されます。

ヒント: 変更を有効にするためには、Navicat を再起動します。



メインツールバーキャプションを表示

メインウィンドウのツールバーボタンにテキストを表示します。無効にすると、ボタンだけが表示されます。

メインツールバーに大きいアイコンを使用

メインウィンドウのツールバーボタンに大きいアイコンを使用します。

フォント

GUI

Navicat インターフェースによって使用されるフォントとそのサイズを定義します。

ヒント: 変更を有効にするためには、データベース/スキーマを開きなおします。

グリッド用フォント

テーブルビューワのグリッドで使用されるフォントとそのサイズを定義します。

エディタ用フォント

エディタで使用されるフォントとそのサイズを定義します。

ER ダイアグラム用フォント

ER ダイアグラムで使用されるフォントとそのサイズを定義します。

アンチエイリアス

表示フォントの質を上げるためには、このオプションにチェックをつけます。

色

グリッドの色

テーブルグリッドの背景色を定義します。

3色を使用する

データを見やすくするために、それぞれ異なる3つの色を使用して、テーブルグリッドの背景を表示します。

テキストの色

この色設定を使用すると、SQL エディタの SQL クエリを色付け強調表示された構文で書式表示することができるので、SQL 文がより読みやすくなります。

異なるテキストフラグメントを区別するために使用する SQL エディタのフォント色を設定します：共通、キーワード、注釈行、文字列、数字。色ボックスをクリックし、**色の設定**ダイアログウィンドウから設定したい色を選択します。

メインウィンドウ

テーブルの情報を表示

オブジェクトリストペイン内のテーブル上をマウスポインターが移動する時、テーブル構成についての詳細を表示するポップアップヒントが表示されます。

接続ペインにオブジェクトを表示

ツリー構造を使用して接続ツリーにデータベースオブジェクトを表示します。ノードを展開するためには、ノードをダブルクリックします。

ヒント: 変更を有効にするためには、データベース/スキーマを開きなおします。

システム項目を表示する(PostgreSQL)

information_schema や *pg_catalog* スキーマのようなシステムオブジェクトを全て表示するためには、このオプションにチェックをつけます。

ヒント: 変更を有効にするためには、データベース/スキーマを開きなおします。

オートインデックスを表示(SQLite)

インデックスのオブジェクトリストペインに SQLite テーブルに対して生成されたオートインデックスを表示するためには、このオプションにチェックをつけます。

エディタ

行番号の表示

参照しやすくするために、エディタの端に行番号を表示します。

コードの折りたたみを使用

コードの折りたたみ機能を使うと、エディタで、ひとまとまりのコードを折りたたんで最初の一行だけを表示することができます。

[コードの折りたたみ](#)を参照してください。

括弧の強調表示を使用

参照しやすくするために、カーソルが片方の括弧に移動すると、その一組の括弧を強調表示します。[括弧の強調表示](#)を参照してください。

タブ幅

タブがとる文字数を入力します。例えば、5。

データ/グリッド

TEXT Blob フィールドをデータグリッドに表示

このオプションを有効にすると、フィールドのタイプが TEXT に設定されているデータがテーブルグリッドに表示されます。無効の場合は、[\(WIDEMEMO\)](#)と表示されます。

レコード制限

テーブルグリッド/外部キーのデータの選択の各ページに表示されるレコード数をグローバルに制限したい場合は、このオプションにチェックをつけます。チェックをつけない場合は、1 ページに全てのレコードが表示されます。

レコード(1 ページあたり)

編集フィールドにある **レコード(1 ページあたり)** の値を設定します(例えば 1000)。この数が、テーブルグリッドの 1 ページあたり表示されるレコード数になります。

注意: 特定のテーブルに対する設定を修正する方法は、[テーブルビューワー](#)をご覧ください。

レコード(外部キーエディタの 1 ページあたり)

編集フィールドの **レコード(外部キーエディタの 1 ページあたり)** の値を設定します(例えば 100)。この数が、[外部キーデータの選択](#)の 1 ページあたり表示されるレコード数になります。

現在のレコードを同期する

(プライマリーキーが存在している場合に)テーブルグリッドで行を更新または挿入する時、このレコードをサーバーからリロードします。

例: 3 つのカラム(id、name、timestamp)をもつテーブルがあるとします。name カラムを更新すると、timestamp カラムがグリッド内ですぐに更新されます。

行の高さ

エディタで使用される行の高さを定義します(例えば 17)。

注意: 特定のテーブルに対する設定を修正する方法は、[テーブルグリッドの書式](#)をご覧ください。

列幅

エディタで使用される列幅を定義します(例えば 150)。

注意: 特定のテーブルに対する設定を修正する方法は、[テーブルグリッドの書式](#)をご覧ください。

自動トランザクション開始

テーブルグリッドでレコード変更の自動コミットを希望する場合は、このオプションにチェックをつけます。チェックをつけない場合、 **トランザクションを開始**ボタンから  **コミット**または  **ロールバック**ボタンをクリックして、変更をコミットまたはロールバックします。[テーブルビューワー](#)をご覧ください。

表示書式

データタイプの整数や浮動小数、日付、時刻、日付時刻は、データグリッドに表示される時、書式表示されます。ここに書式を入力することによって、その書式を変更できます。書式が空白の場合、デフォルト書式が使用されます。日付、時刻、日付時刻フィールドに関して、デフォルト書式はシステムの日付時刻書式になります。

表示書式	
整数:	<input type="text" value="#,###"/>
浮動小数:	<input type="text" value="#.##"/>
日付:	<input type="text" value="dd/mm/yyyy"/>
時刻:	<input type="text" value="hh:mm:ss"/>
日付時刻:	<input type="text" value="dd/mm/yyyy hh:mm:ss"/>
例:	4/15/2016 3:18:00 PM
出力:	15/04/2016 15:18:00

これらの書式指定子を使用して文字列を構成することにより、書式は定義されます:

数値フィールド

指定子	表示
0	数値置換子。書式表示されている値が書式文字列中の"0"がある位置に数値を持つ場合、その数値は出力文字列にコピーされます。そうでない場合は、出力文字列のその位置に"0"が格納されます。(例えば、整数フィールドに 0000 がある場合、テーブルから出力されるすべての整数は、0012 というような書式になります)
#	数値置換子。書式表示されている値が書式文字列中の"#"がある位置に数値を持つ場合、その数値は出力文字列にコピーされます。そうでない場合は、出力文字列のその位置には何も格納されません。(例えば、整数フィールドに ## ## がある場合、テーブルから出力されるすべての整数は、12 34 というような書式になります)
.	小数点。書式文字列にある最初の"."は、書式表示される値の小数点記号の位置を決定します; その後の"."はすべて無視されます。出力文字列内で小数点記号として実際に使用される文字は、グローバル変数 <code>DecimalSeparator</code> によって指定されます。
,	桁区切り。書式文字列が 1 つ以上の","を含む場合、出力文字列は、小数点の左側に 3 桁ごとに桁区切りが挿入されます。桁区切りが必要な時以外、書式文字列内の","の位置や数は出力文字列に影響しません。出力文字列内で桁区切りとして実際に使用される文字は、グローバル変数 <code>ThousandSeparator</code> によって指定され

	ます。
E+	科学的表記法。文字列 "E+", "E-", "e+", "e-"のいずれかが書式文字列に含まれる場合、数字は科学的表記法を用いて書式表示されます。"E+"または"E-", "e+", "e-"のすぐ後に最大 4 つの"0"を続けることで、指数の最小桁数を決定します。書式"E+"と"e+"は、正の指数に対してプラスサインを出力し、負の指数に対してマイナスサインを出力します。書式"E-"と"e-"は、負の指数に対してマイナスサインを出力するだけです。
'xx'/"xx"	シングルのまたはダブルクォートで囲まれた文字はそのまま表示されます。表示書式の影響を受けません。
;	書式文字列内で、正、負、ゼロの数値用にセクションを区切ります。

書式文字列の小数点の前の左端にある"0"や小数点の後の右端にある"0"の位置は、出力文字列に常に表示される指数の幅を決定します。

書式表示された数値は、常に、小数点の右にある数値置換子 ("0" or "#")と同数の少数位に四捨五入されます。書式文字列が小数点を含まない場合、書式表示されている値は最も近い整数に四捨五入されます。

書式表示された数値が、書式文字列内の"."の左側にある数値置換子の数よりも多い指数を小数点記号の左側に持つ場合、余分の数値が最初の数値置換子の前に出力されます。

以下の表は、様々な書式文字列の効果です:

表示書式	値	結果	注釈
#.##	12.2	12.2	小数の左側にある余分な数値が表示されていることに注目してください。
#.00	2.5	2.50	余分な"0"に注目してください: フィールドは常に少数第 2 位まで表示します。
00.##	.006	00.01	小数点の右側にある余分な"0"と、少数第 2 位に四捨五入されていることに注目してください。

正、負、ゼロの値それぞれに異なる書式を適用するためには、セミコロンによって書式文字列を 1~3 つの区分に分けます。

区分が 1 つ: 書式文字列はすべての値に適用されます。

区分が 2 つ: 第 1 区分が正の値とゼロに適用され、第 2 区分が負の値に適用されます。

区分が 3 つ: 第 1 区分が正の値に適用され、第 2 区分が負の値に適用され、第 3 区分がゼロに適用されます。

負の値用の区分またはゼロ用の区分が空白の場合、つまり、区分を区切るセミコロン間に何も無い場合、代わりに正の値用の区分が使用されます。

正の値用の区分が空白の場合、または、書式文字列全体が空白の場合、値は一般的な 15 桁の浮動小数点の書式で書式表示されます。値が小数点の左側に 18 桁以上あり、書式文字列が科学的表記法を指定していない場合も、一般的な浮動小数点の書式が使用されます。

日付と時刻のフィールド

指定子	表示
c	グローバル変数 ShortDateFormat に指定された書式を使用した日付、その後に、グローバル変数 LongTimeFormat に指定された書式を使用した時刻が続いて表示されます。日付時刻値の小数部分がゼロの場合、時刻は表示されません。

d	日をゼロ詰めしない数字で表します(1-31)。
dd	日をゼロ詰めした数字で表します(01-31)。
ddd	グローバル変数 ShortDayNames で指定された文字列を使用して、日を短縮名称(日-土)で表します。
dddd	グローバル変数 LongDayNames で指定された文字列を使用して、日を完全名称(日曜日-土曜日)で表します。
dddddd	グローバル変数 ShortDateFormat で指定された書式を使用して、日付を表します。
ddddddd	グローバル変数 LongDateFormat で指定された書式を使用して、日付を表します。
m	月をゼロ詰めしない数字で表します(1-12)。m 指定子が、h あるいは hh 指定子の直後に続く場合、月ではなく、分が表示されます。
mm	月をゼロ詰めした数字で表します(01-12)。mm 指定子が、h あるいは hh 指定子の直後に続く場合、月ではなく、分が表示されます。
mmm	グローバル変数 ShortMonthNames で指定された文字列を使用して、月を短縮名称(Jan-Dec)で表します。
mmmm	グローバル変数 LongMonthNames で指定された文字列を使用して、月を完全名称(1月-12月)で表します。
yy	年を2桁の数字で表します(00-99)。
yyyy	年を4桁の数字で表します(0000-9999)。
h	時間をゼロ詰めしないで表します(0-23)。
hh	時間をゼロ詰めして表します(00-23)。
n	分をゼロ詰めしないで表します(0-59)。
nn	分をゼロ詰めして表します(00-59)。
s	秒をゼロ詰めしないで表します(0-59)。
ss	秒をゼロ詰めして表します(00-59)。
t	グローバル変数 ShortTimeFormat で指定された書式を使用して、時刻を表します。
tt	グローバル変数 LongTimeFormat で指定された書式を使用して、時刻を表します。
am/pm	前述の h または hh 指定子を12時間制の時刻で表し、その後、正午前の時間には"am"、正午以降の時間には"pm"が続いて表示されます。am/pm 指定子は、大文字、小文字、またはそれらの混合形でも指定することができ、それによって、結果が表示されます。
a/p	前述の h または hh 指定子を12時間制の時刻で表し、その後、正午前の時間には"a"、正午以降の時間には"p"が続いて表示されます。a/p 指定子は、大文字、小文字、またはそれらの混合形でも指定することができ、それによって、結果が表示されます。
ampm	前述の h または hh 指定子を12時間制の時刻で表し、その後、正午前の時間にはグローバル変数 TimeAMString、正午以降の時間にはグローバル変数 TimePMString の内容が続いて表示されます。
/	グローバル変数 DateSeparator に指定された日付区切り文字。
:	グローバル変数 TimeSeparator に指定された時刻区切り文字。
'xx'/'xx'	シングルまたはダブルクォートに囲まれた文字列は、書式変更されずに、そのまま表示されます。

書式の指定子は、大文字でも小文字でも記述されます；どちらでも同じ結果になります。

モデルオプション(フルバージョンで使用可能)

オブジェクトを強調表示

このオプションを選択すると、マウスカーソルがオブジェクトの上に乗った時、Navicat はその縁を青色で強調表示します。

ヒント: 変更を有効にするためには、Navicat を再起動します。

リレーションも含め強調表示

このオプションを選択すると、マウスカーソルがテーブルまたはビューの上に乗った時、Navicat はその外部キーまたはビューのリレーションを青色または緑色で強調表示し、他のオブジェクトとの関係を示します。

ヒント: 変更を有効にするためには、Navicat を再起動します。

フィールドタイプを推測

このオプションを選択すると、テーブルのフィールドを設計する時、Navicat は [フィールドのタイプを予測します](#)。

ヒント: 変更を有効にするためには、Navicat を再起動します。

その他のオプション

ファイルの保存場所

異なる種類のファイル用のフォルダを変更できます。デフォルトで、ほとんどのファイルは、[設定の保存場所](#)に保存されますが、いくつかのプロファイルは [profiles](#) ディレクトリに保存され、すべての [ログファイル](#) は、**logs** ディレクトリに保存されます。

アクティベーション&アップデーター

起動時に更新を自動チェック

Navicat の起動時に新しいバージョンの確認を行いたい場合はこのオプションにチェックをつけます。

プロキシ

プロキシの使用

アクティベーション処理にプロキシを使用したい場合はこのオプションにチェックをつけ、**ホストとポート**、**ユーザー名**、**パスワード**を入力します。

OCI (Oracle で使用可能)

OCI library (oci.dll)

[Basic](#) 接続するために、OCI ライブラリ (oci.dll)を含む **Oracle クライアント/Oracle インスタントクライアント**フォルダを選択します。それは、Navicat インストールフォルダに含まれていますので、直接指定することができます。

Oracle インスタントクライアントを使用すると、OCI または OCCI、JDBC-OCI、ODBC ドライバを用いて作られた完全な Oracle クライアントアプリケーションを非常に簡単に展開できます。Oracle インスタントクライアントは、必要な Oracle クライアント

ライブラリを少数のファイルで提供します。以下からも、**Oracle クライアント** / **Oracle インスタントクライアント**をダウンロードできます -

Oracle クライアント

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>

Oracle インスタントクライアント

<http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

お使いのプラットフォームと CPU に適したインスタントクライアントパッケージをダウンロードしてください。すべてのインストールレーションにおいて、Basic または Basic Lite パッケージが必ず必要です。パッケージを解凍し、それに対してパスポイントを設定してください。

Oracle クライアントインストールガイド

http://download.oracle.com/docs/cd/B28359_01/install.111/b32302/toc.htm

ヒント: 変更を有効にするためには、Navicat を再起動します。

コマンド(フルバージョンで使用可能)

Navicat オブジェクト	サーバーの種類	コマンドライン
バックアップ	MySQL、 PostgreSQL、 SQLite、MariaDB	start_navicat -backup [ProfileName] -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
インポート	全て	start_navicat -import ProfileName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
テーブルのエクスポート	全て	start_navicat -export ProfileName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
ビューの結果のエクスポート	全て	start_navicat -exportview ProfileName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
マテリアライズドビューの結果のエクスポート	Oracle、 PostgreSQL	start_navicat -exportmview ProfileName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
クエリの結果のエクスポート	全て	start_navicat -exportquery ProfileName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
クエリの実行	全て	start_navicat -query QueryName -u NavicatID -p ProjectName -t ConnectionType -c ConnectionName -d DatabaseName -s SchemaName
データ転送	全て	start_navicat -datatransfer ProfileName -t ProfileType
データの同期	全て	start_navicat -datasync ProfileName -t ConnectionType
バッチジョブ	全て	start_navicat -batchjob BatchJobName

注意:

NavicatID - 接続が Navicat Cloud に保存される場合、Navicat ID が必要です。例 : user@example.com

ProjectName - 接続が Navicat Cloud に保存される場合、プロジェクト名が必要です。

ConnectionType- 接続の種類: MySQL、Oracle、PostgreSQL、SQLite、MSSQL、MariaDB

ProfileType - データ転送プロファイルの種類: MySQL、Oracle、PostgreSQL、SQLite、MSSQL、MariaDB、Premium

例:

```
start_navicat -exportquery MyQueryExport1 -u test@navicat.com -p Project1 -t MySQL -c "MySQL 5.6" -d sakila
```

ホットキー

Navicat メインウィンドウ

キー	アクション
CTRL+# (# 0~9 を表す)	お気に入りリストからオブジェクトウィンドウを開く
CTRL+H	履歴ログ
CTRL+Q	新規クエリ
F12	アクティブなオブジェクトのみを表示

共通

キー	アクション
CTRL+N	新規オブジェクト
SHIFT+CTRL+# (# 0~9 を表す)	お気に入りに追加
F8	Navicat メインウィンドウ
CTRL+TAB または SHIFT+CTRL+TAB	次のウィンドウ
F1	ヘルプ
CTRL+F1	オンラインドキュメント

テーブルデザイナー

キー	アクション
CTRL+O	テーブルを開く
CTRL+F	フィールドを検索
F3	次のフィールドを検索
SHIFT+F3	前のフィールドを検索

テーブルビューワー/ビュービューワー

キー	アクション
CTRL+D	テーブルを設計/ビューを設計
CTRL+Q	テーブルへのクエリを行う/ビューへのクエリを行う
CTRL+F	テキストを検索
F3	次のテキストを検索
CTRL+G	行へ進む
CTRL+LEFT ARROW	現在のレコードの最初のデータカラム
CTRL+RIGHT ARROW	現在のレコードの最後のデータカラム
CTRL+HOME	現在のカラムの最初のデータ行
CTRL+END	現在のカラムの最後のデータ行

CTRL+PAGE UP または CTRL+UP ARROW	現在のウィンドウの最初のデータ行
CTRL+PAGE DOWN または CTRL+DOWN ARROW	現在のウィンドウの最後のデータ行
CTRL+R	フィルターウィザードでフィルターを適用
SHIFT+ARROW	セルを選択
CTRL+ENTER	エディタを開いてデータを編集
INSERT または CTRL+N	新規レコード
CTRL+DELETE	レコードを削除
CTRL+S	レコードの変更を適用
ESC	レコードの変更をキャンセル
CTRL+T	データのロードを停止

ビュー/クエリ

キー	アクション
CTRL+O	ビューをロード/クエリをロード
CTRL+/	コメント行
SHIFT+CTRL+/	非コメント行
CTRL+E	定義を見る/クエリエディタ
CTRL+R	実行
SHIFT+CTRL+R	選択したものを実行
F7	ここからステートメントを実行
CTRL+T	停止

SQL エディタ

キー	アクション
CTRL+F	テキストを検索
F3	次のテキストを検索
CTRL+= または CTRL+Mousewheel Up	ズームイン
CTRL+- または CTRL+Mousewheel Down	ズームアウト
CTRL+0	ズームをリセット

デバッガ

キー	アクション
F9	実行
F8	ステップオーバー
F7	ステップイン
SHIFT+F7	ステップアウト

モデル

キー	アクション
CTRL+D	モデルに新規ダイアグラム
CTRL+P	印刷
ESC	選択
H	ダイアグラムを移動
T	新規テーブル
V	新規ビュー
L	新規レイヤー
A	新規ラベル
N	新規ノート
I	新規イメージ
R	新規外部キー
CTRL+B	選択されたテーブルまたはビュー、外部キー、形状を太字にする
CTRL+= または CTRL+Mousewheel Up	ズームイン
CTRL+- または CTRL+Mousewheel Down	ズームアウト
CTRL+0	ズームをリセット

ログファイル

Navicat は、Navicat で実行された動作を記録するために、いくつかのログファイルを提供しています。それらは **logs** ディレクトリに置かれます。例えば、Y:¥.navicat¥Navicat¥MySQL¥logs¥。 [オプション](#) で、ログファイルの場所を変更することができます。

HttpDump.log

HTTP サーバーから返された情報を保存します。

LogHistory.txt

Navicat のデータベースやデータベースオブジェクトに対して実行された全操作の全 SQL 文を保存します。履歴ログビューワーで LogHistory.txt ファイルを開くためには、メインメニューから **ツール** -> **履歴ログ** を選択するか、CTRL+H を押します。

注意: このログは、Navicat が再起動されると上書きされます。

LogImport.txt

インポート処理中に発生した全てのエラーに関する詳細情報を記録します (成功または失敗を示す)。

注意: このログは、各インポート毎に上書きされます。

LogExport.txt

エクスポート処理中に発生した全てのエラーに関する詳細情報を記録します (成功または失敗を示す)。

注意: このログは、各エクスポート毎に上書きされます。

LogSynchronize.txt

データの同期処理中に発生した全てのエラーに関する詳細情報を記録します (成功または失敗を示す)。

注意: このログは、同期ごとに上書きされます。

LogCmd.txt

Navicat コマンドラインプロセスとスケジュール実行中の全ての操作に関する情報を保存します。