

Wize テーブル仕様書 操作説明書

第 1.3 版

作 成 者	株式会社 Wize
作 成 日	2017 年 2 月 10 日

[illegible]

目次

Wize テーブル仕様書 操作説明書	1
1. 概要	4
2. 使用制限	4
3. 動作環境	5
4. 動作環境設定	6
4.1. データベース接続用の設定	6
4.2. データベース用パッケージのインストール	9
4.3. Excel のオプション設定	11
5. 操作説明	13
5.1. 起動方法	13
5.2. 環境設定	14
5.3. 終了方法	15
5.4. シート一覧	16
5.5. テーブル一覧シート	16
5.6. テーブル仕様シート	23
5.7. 日本語-英語変換機能	36
5.8. トリガー設定機能	39
6. 処理実行時のチェック項目	41
6.1. テーブル一覧のチェック項目	41
6.2. テーブル仕様のチェック項目	41
7. おわりに	42

1. 概要

テーブル仕様書は、Excel シート上で動作する、データベースシステムの開発を支援するアプリケーションです。

テーブル仕様書の主な機能は次のとおりです。

- ・ DB インポート機能
データベースに接続してテーブル定義情報を取得し、Excel シートに表示します。
- ・ CREATE 文出力機能
Excel シートに入力されたテーブル仕様書からテーブル生成用の CREATE 文を出力します。
またトリガー設定からインサート/アップデートトリガー文とログテーブル出力用トリガー文を出力します。

2. 使用制限

本ソフトはフリーソフトです。個人・法人にかかわらず自由にご使用ください。
本ソフトへのリンクは自由ですが、転載,配布は禁止します。
なお著作権はすべて株式会社 Wize が保有しています。

本ソフトを使用した事によって生じたすべての障害・損害・不具合等に関して、弊社は一切の責任を負いません。
各自の責任においてご使用ください。

3. 動作環境

本ソフトの動作確認済み環境は以下のとおりです。

OS	Microsoft Windows 7 (32bit, 64bit)	
Excel	Microsoft Excel 2013 (32bit)	Microsoft Excel 2013 (64bit)
データベース・サーバ	Oracle 11g, PostgreSQL 9.3	
データベース・クライアント	Oracle 11g Client (32bit) PostgreSQL 9.3 Client (32bit)	Oracle 11g Client (64bit) PostgreSQL 9.3 Client (64bit)
インターフェース	データソース(ODBC) (32bit)	データソース(ODBC) (64bit)

【注意】

Windows7 64bit 環境で使用する場合、Excel, データベース・クライアント, インターフェースはすべて 32bit 版または 64bit 版で統一してください。

32bit 版, 64bit 版が混在している場合、データベースアクセス機能が正しく動作しません。

なお、Windows7 32bit 環境で使用する場合は、Excel, データベース, インターフェースはすべて 32bit 版となりますので問題ありません。

4. 動作環境設定

使用目的に応じた設定項目は以下のようになっています。

以下の表を参考にして、データベースからテーブル定義情報を取得する DB インポート機能を使用するか、ワークシートに入力したテーブル定義情報から CREATE 文を出力する機能を使用するかにより、必要な項目を設定してください。

設定項目	DB インポート機能	CREATE 文出力機能
ネットサービス名	●	—
データソース(ODBC)	●	—
Excel セキュリティセンター	●	●

● 必須, △ 任意, — 不要

4.1. データベース接続用の設定

Oracle データベースを使用する場合

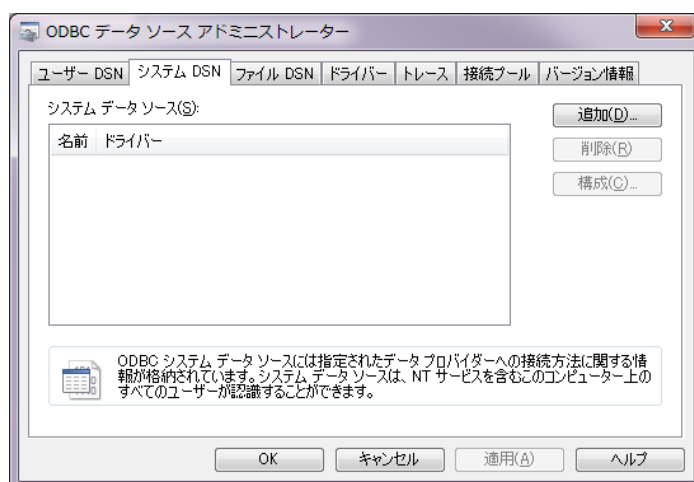
Oracle データベースを使用する場合は、あらかじめ使用するデータベースのネットサービス名(TSN サービス名)を設定しておく必要があります。

スタートメニューから Net Configuration Assistant を起動して設定してください。

設定方法の詳細については、Net Configuration Assistant のマニュアルを参照してください。

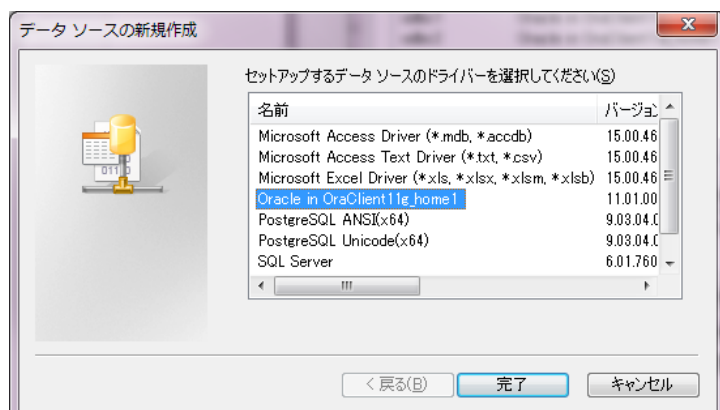
データソースの設定

- 1) コントロールパネル/管理ツール/データソース(ODBC) を起動します。
 なお、Windows7 64bit 環境で、32bit 版の ODBC データアドミニストレータを起動するためには、
`%systemdrive%\Windows\SysWoW64\odbcad32.exe` を実行する必要がありますので、十分ご注意ください。
- 2) ODBC データソースアドミニストレータ画面の、システム DSN のタブを選択します。



- 3) [追加]ボタンをクリックします。

- 4) データソースの新規作成画面で、使用するデータベース用のドライバーを選択して [完了] ボタンをクリックします。



使用するデータベースとドライバの対応は以下のとおりです。

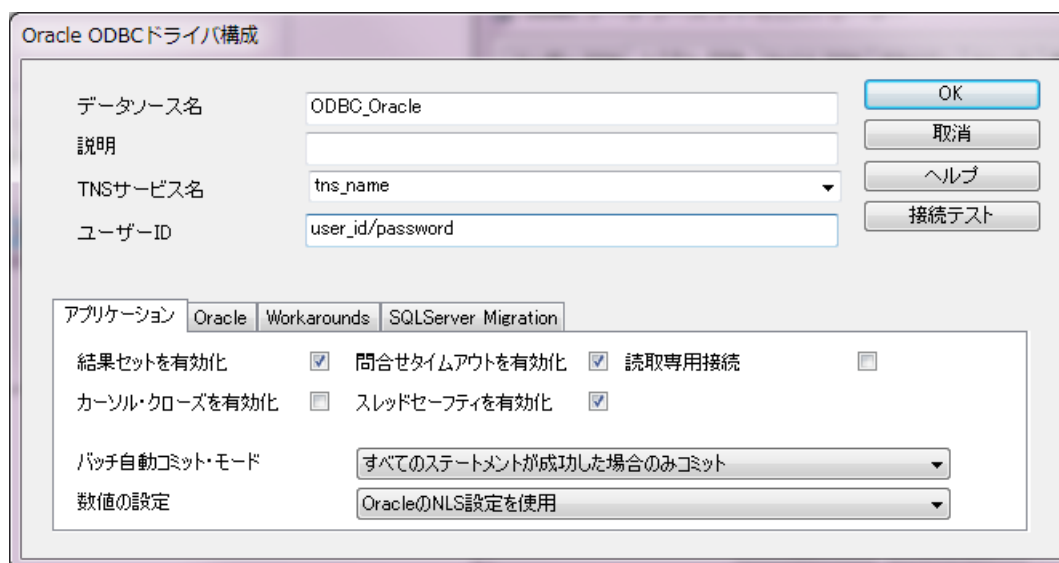
データベース	選択するドライバ
Oracle	Oracle in OraClient11g_home1
PostgreSQL	PostgreSQL Unicode(x32), PostgreSQL Unicode(x32)

- 5) 次に ODBC ドライバ構成画面が表示されるので、必要な項目を入力します。
ODBC ドライバ構成画面は使用するデータベースによって設定内容が異なるので、各 ODBC ドライバ構成画面に従って操作してください。

データベース	ODBC ドライバ設定方法
Oracle	5)-1. Oracle 用 ODBC ドライバ設定 を参照してください。
PostgreSQL	5)-2. PostgreSQL 用 ODBC ドライバ設定 を参照してください。

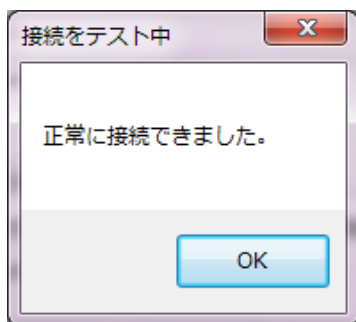
5)-1. Oracle 用 ODBC ドライバ設定

以下の画面で、データソース名, TNS サービス名, ユーザーID の項目にデータを入力します。
ユーザーID には、ユーザ ID とパスワード を “/” で区切って入力してください。



データ入力後に、[接続テスト] ボタンをクリックします。

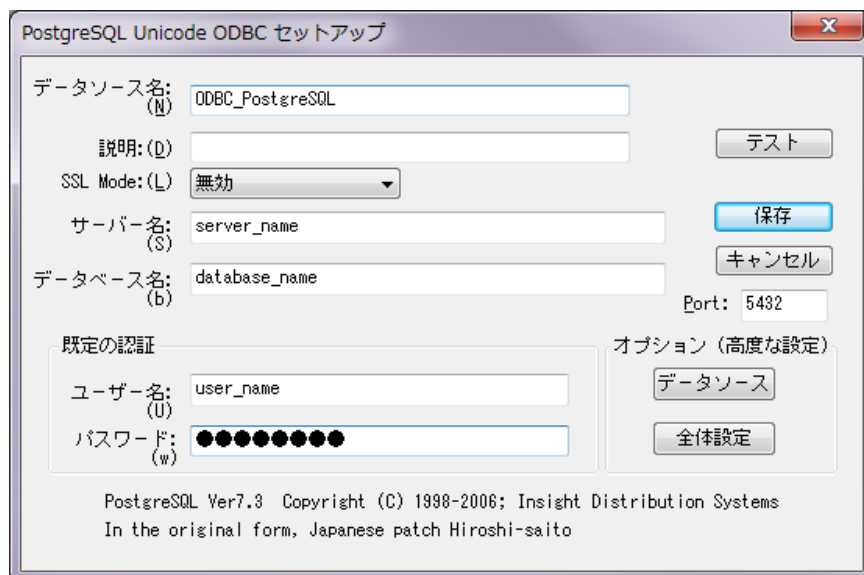
データベースに接続できれば以下のメッセージが表示されます。



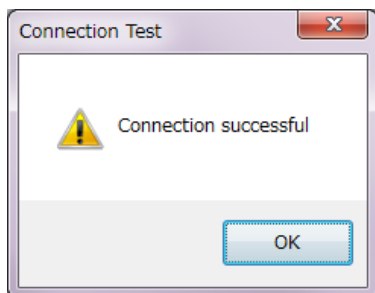
[OK]ボタンをクリックし、Oracle ODBC ドライバ構成画面の[OK]ボタンをクリックします。

5)-2. PostgreSQL 用 ODBC ドライバ設定

以下の画面で、データソース名、サーバ名、データベース名、ユーザー名、パスワード の項目にデータを入力します。

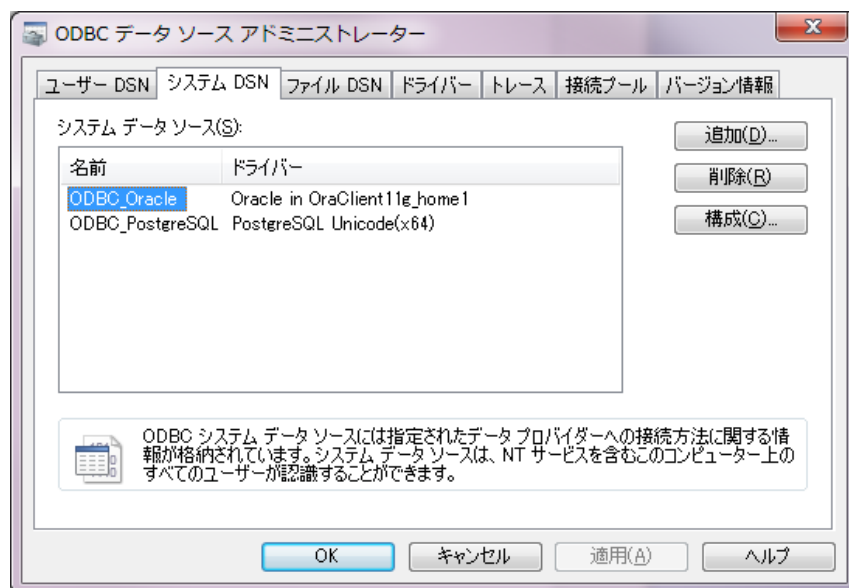


[テスト]ボタンをクリックします。
データベースに正常に接続できれば、下記のメッセージが表示されます。



[OK]ボタンをクリックし、PostgreSQL ODBC セットアップ画面の[保存]ボタンをクリックします。

- 6) ODBC データソースアドミニストレータ画面のシステム DSN タブを選択し、システムデータソースに追加した項目が表示されていることを確認してください。



以上で ODBC データソースの設定は終了です。

4.2. データベース用パッケージのインストール

Oracle データベースを使用する場合

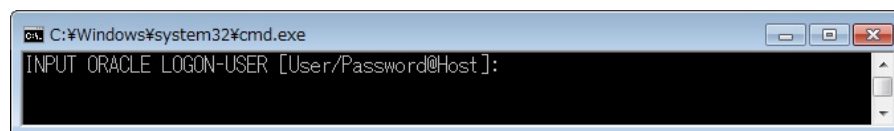
Oracle データベースを使用する場合、以下のパッケージをインストールすることにより、CREATE 文でテーブル生成時に、バックアップしたデータを復帰することができます。

※この機能は現在 Oracle のみの対応となります。ご了承ください。

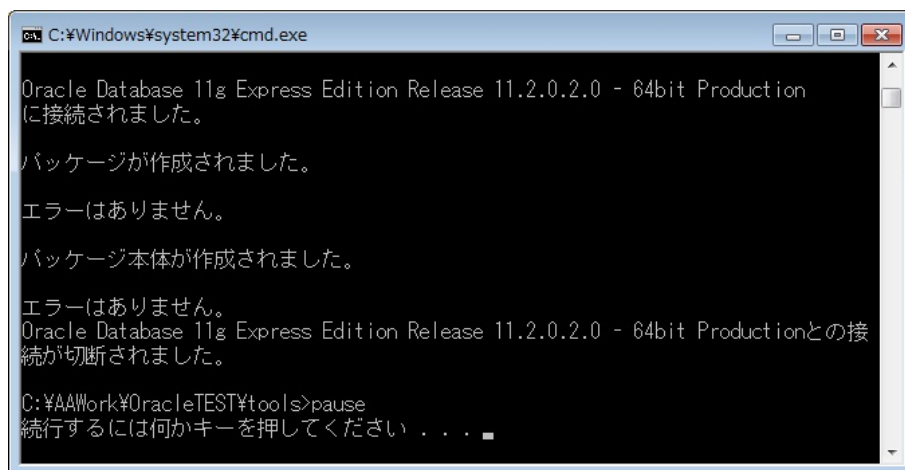
- 1) ダウンロードしたファイルに含まれる WizePkgInstaller フォルダを開くと、以下の 3 つのファイルが格納されています。

```
INSTALL_WIZE_PKG.bat
Instpkg.sql
PKG_WizeDBMaster.plb
```

- 2) コマンドプロンプトを起動して、INSTALL_WIZE_PKG.bat を実行します。



上記のメッセージが表示されるので、Oracle ユーザ名、パスワード、ホスト名を user/password@host 形式で入力してください。



```
C:\Windows\system32\cmd.exe

Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
に接続されました。

パッケージが作成されました。

エラーはありません。

パッケージ本体が作成されました。

エラーはありません。
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Productionとの接
続が切断されました。

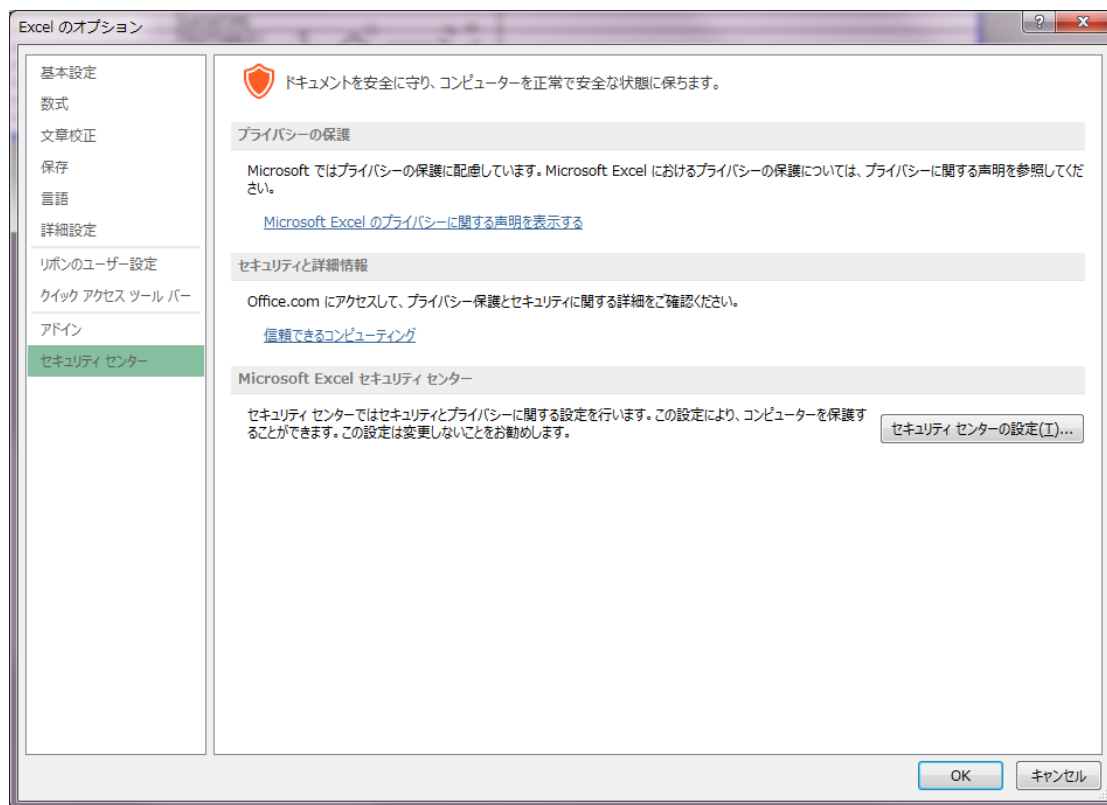
C:\AAWork\OracleTEST\tools>pause
続行するには何かキーを押してください . . .
```

上記のメッセージが表示され、“エラーはありません。” ”パッケージ本体が作成されました。”
が表示されれば、パッケージのインストールは成功です。

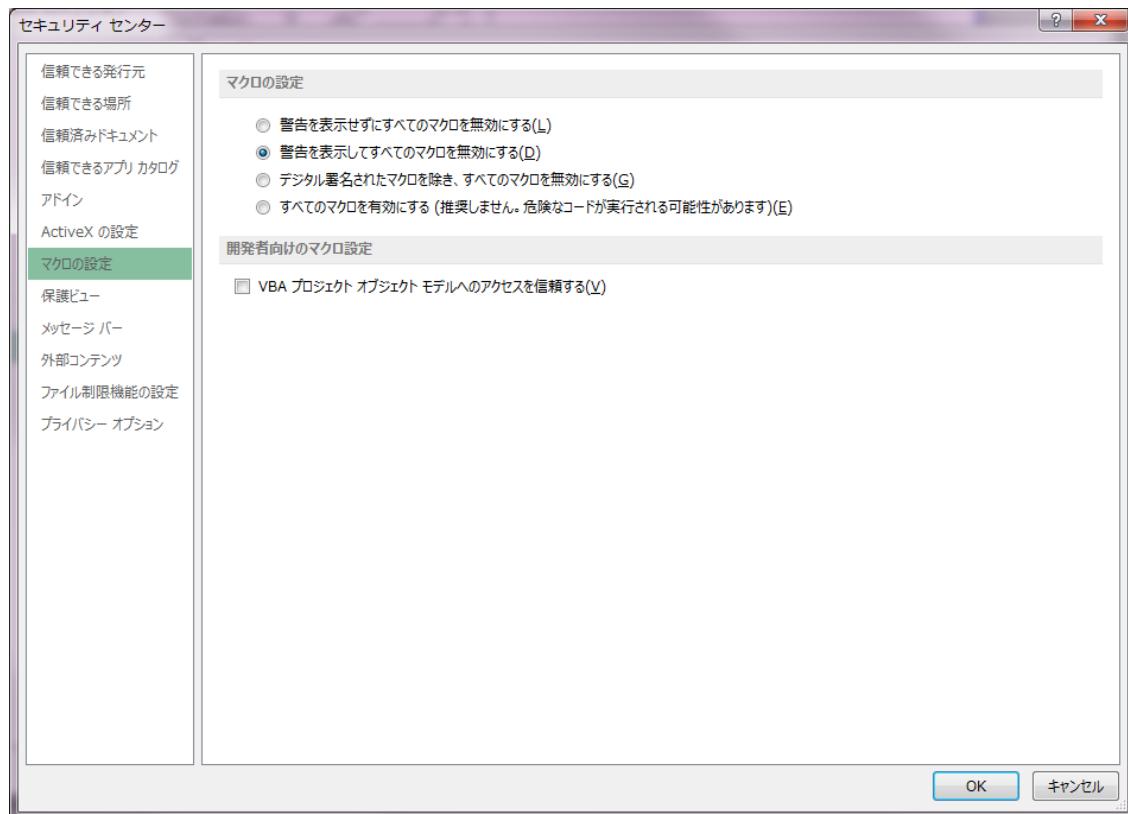
4.3. Excel のオプション設定

セキュリティセンターの設定

- 1) Excel を起動し、メニューの ファイル/オプション を選択して、Excel のオプション画面を表示し、左側のメニューから”セキュリティセンター”を選択します。



- 2) [セキュリティセンターの設定]ボタンをクリックします。
- 3) セキュリティセンターの左側のメニューから、”マクロの設定”を選択します。



- 4) マクロの設定で、“警告を表示してすべてのマクロを無効にする” を選択してください
- 5) [OK]ボタンをクリックします。
- 6) Excel のオプション画面で[OK]ボタンをクリックします。

以上で Excel の環境設定は終了です。

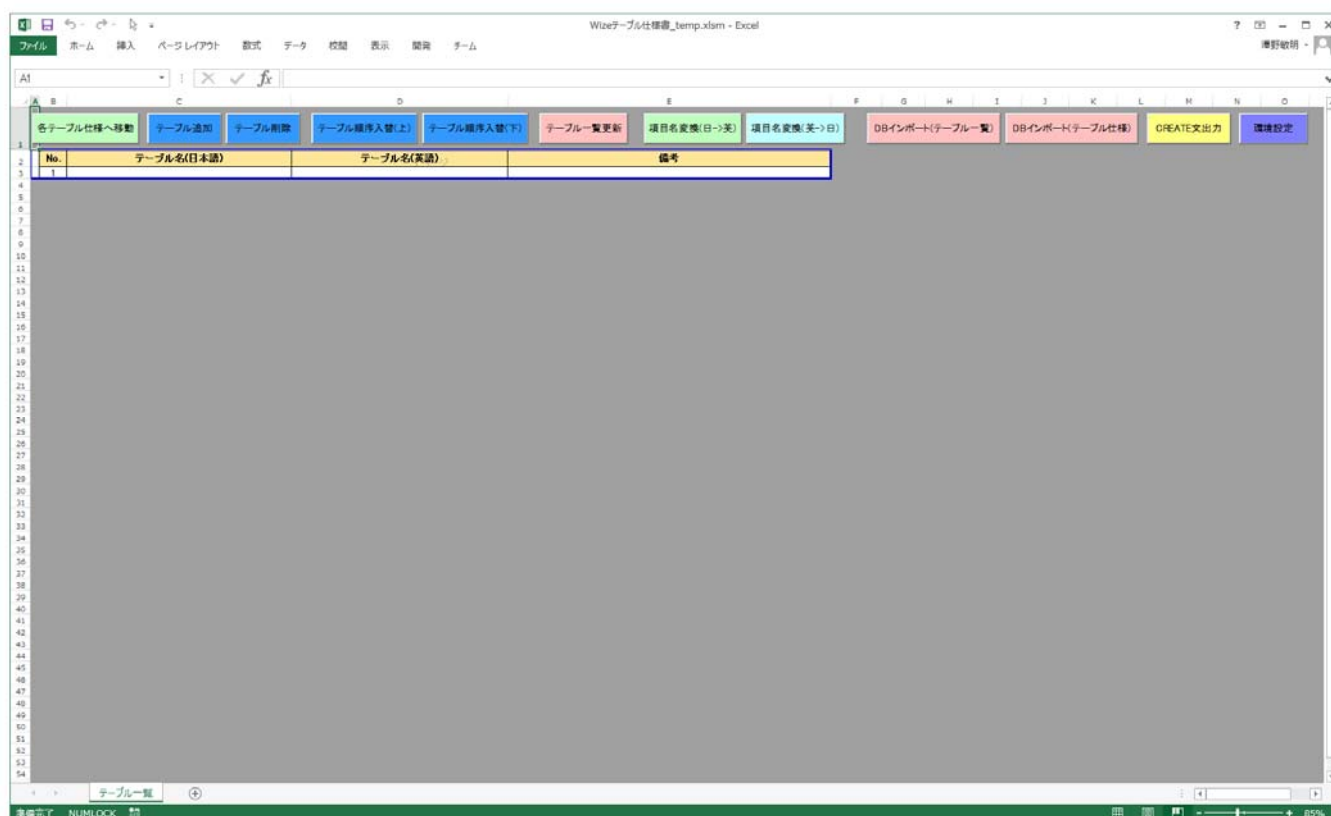
5. 操作説明

以下、テーブル仕様書の操作方法を説明します。

5.1. 起動方法

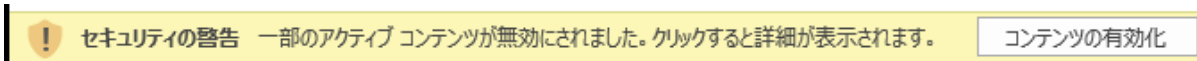
“テーブル仕様書.xlsm”を開く

通常の Excel シートと同様に“テーブル仕様書.xlsm”ファイルを開いて起動してください。
起動すると以下のような画面が表示されます。



セキュリティの警告が表示される場合

起動後、Excel メニューの下に以下のような警告メッセージが表示される場合は、[コンテンツの有効化]ボタンをクリックしてください。



その他

シート上部にあるボタンをクリックしても何も動作しない場合は、Excel の環境設定で、マクロの実行が無効になっている可能性があります。再度 Excel のオプション設定 に従って設定内容を確認してください。

5.2. 環境設定

“テーブル一覧シート” の[環境設定]ボタンをクリックする

テーブル仕様書の機能を使用するためには、最初に環境設定が必要です。
環境設定ボタンをクリックすると、以下のような画面が表示されます。

環境設定

バージョン 1.3.0

データベース種別 ORACLE

ODBCデータソース名(DSN) odbc_oracle

プロジェクト名 DB_TEST

スキーマ名 schema_name

CREATE文のスキーマ名 ☒ 使用しない ☐ 使用する

CREATE文のテーブル名、項目名 ☒ 引用符で囲まない ☐ 引用符で囲む
※PostgreSQLの場合は、必ず「引用符で囲む」の動作となります

マテリアライズドビュー情報の取得 ☒ 取得しない ☐ 取得する

CREATE文出力フォルダ C:\Users\user_name\Documents\参照

キャンセル OK

各項目を設定してください。

設定項目	内容
バージョン	テーブル仕様書のバージョン番号です。
データベース種別	データベースの種別を選択します。 (現在のバージョンでは Oracle, PostgreSQL が選択できます)
ODBC データソース名(DSN)	ODBC データソース名を入力します。
プロジェクト名	プロジェクト名を入力します。 CREATE 文出力時に使用します。
スキーマ名	スキーマ名を入力します。
CREATE 文のスキーマ名	CREATE 文のテーブル名にスキーマ名を付加するかどうかを指定します。
CREATE 文のテーブル名、項目名	CREATE 文のテーブル名、項目名を引用符で囲むかどうかを指定します。

	※ただしデータベース種別を”PostgreSQL”に設定した場合は、ラジオボタンが選択不可となり、「引用符で囲む」の動作となります。
マテリアライズドビュー情報の取得	DB インポート時にマテリアライズドビュー情報を取得するかどうかを指定します。（現在は”取得しない”しか選択できません） マテリアライズドビュー情報のテーブル名は、先頭に MLOG\$_ , RUPD\$_ , USLOG\$_ が付いています。
CREATE 文出力フォルダ	CREATE 文ファイルの出力先フォルダを入力します。 [参照]ボタンで任意のフォルダを指定できます。

必要項目を設定した後、以下のいずれかのボタンをクリックしてください。

ボタン	機能
OK	入力されたデータを環境設定シートに保存します。
キャンセル	環境設定の変更を行いません。

以下の表を参考にして、データベースからテーブル定義情報を取得する DB インポート機能を使用するか、Excel シートに入力したテーブル定義情報から CREATE 文を出力する機能を使用するかにより、必要な項目を設定してください。

設定項目		DB インポート機能	CREATE 文出力機能
テーブル仕様書の 環境設定	データベース種別	●	●
	ODBC データソース名(DSN)	●	—
	プロジェクト名	—	△
	スキーマ名	△	△
	CREATE 文でスキーマ名を使用	—	△
	マテリアライズドビュー情報の取得	△	—
	CREATE 文出力フォルダ	—	●

● 必須, △ 任意, — 不要

5.3. 終了方法

“テーブル仕様書.xlsx”を閉じる

通常の Excel シートと同様に”テーブル仕様書.xlsx”ファイルを閉じて終了してください。

5.4. シート一覧

テーブル仕様書には、あらかじめ以下に示す 7 種類のワークシートが用意されています。

シート名	機能・用途
テーブル一覧	テーブル一覧表示用シートで、各種操作実行用ボタンがあります。(表示)
#トリガー設定	追加/更新用トリガー出力用のデータを保存します。(非表示)
#環境設定	環境設定の内容を保存します。(非表示)
#DIC	日本語-英語変換辞書データを保存します。(非表示)
#Table	テーブル仕様作成用のテンプレートシートです。(非表示)
#CT_Oracle	Oracle 用の CREATE 文出力用のテンプレートです。(非表示)
#CT_PostgreSQL	PostgreSQL 用の CREATE 文出力用のテンプレートです。(非表示)
#CT_SQLServer	SQLServer 用の CREATE 文出力用のテンプレートです。(非表示)

非表示設定のシートは必要に応じて再表示してご使用ください。

テーブル仕様書が作成されると、テーブル 1 つにつきテーブル仕様シートが 1 つ自動的に追加されます。

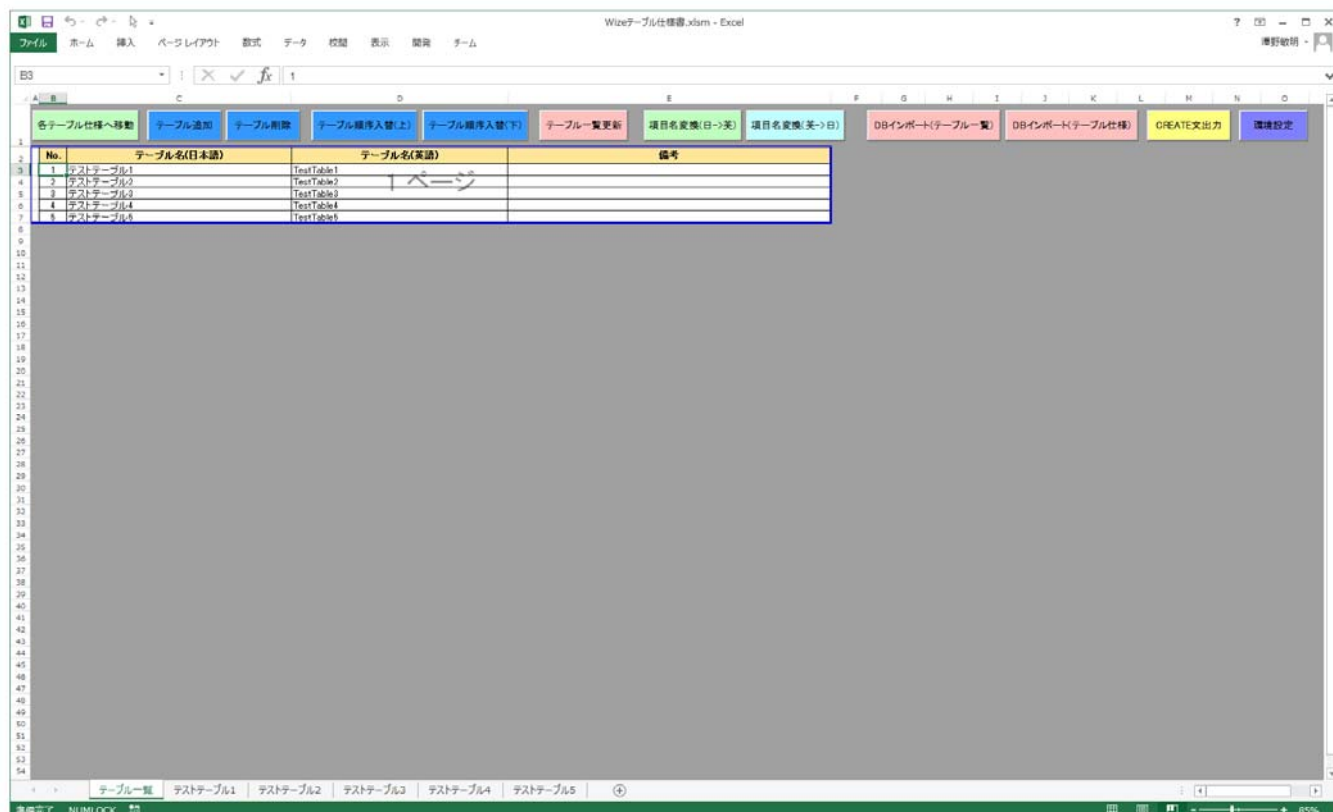
【注意】

これらのシートは絶対に削除しないでください。

また、動作環境によっては、作成したテーブル仕様シート数が多い場合、保存したファイルが正しく開けなくなることがありますのでご注意ください。

5.5. テーブル一覧シート

テーブル一覧シートは、データベースから取得したテーブル定義の一覧表示、テーブル一覧からテーブル仕様書の作成、およびテーブル仕様書から CREATE 文を作成するなど、テーブル仕様書の主要機能が集約されているシートです。



各ボタンの機能

[各テーブル仕様へ移動]

テーブル一覧で選択したテーブル仕様シートへ移動します。

テーブル一覧の任意の行を選択してからこのボタンをクリックすると、選択された行の”テーブル名(日本語)”と同一名のワークシートへ移動します。

[テーブル追加]

テーブル一覧表にテーブルを追加します。

このボタンをクリックすると、以下のダイアログボックスが表示されます。

必要項目を入力してください。

項目名	機能
スキーマ名	スキーマ名を入力します。 環境設定で入力されたスキーマ名が表示されます。
テーブル名(日本語)	テーブル名を日本語で入力します。データベースに定義するテーブルのコメントとして使用されます。省略できません。 [日本語->英語変換]ボタンを押下すると、テーブル名(日本語)に入力されたデータを元に、辞書データを参照してテーブル名(英語)を自動入力することができます(※)。
テーブル名(英語)	テーブル名を英語で入力します。データベースに定義するテーブル名として使用されます。省略できません。 [英語->日本語変換]ボタンを押下すると、テーブル名(英語)に入力されたデータを元に、辞書データを参照してテーブル名(日本語)を自動入力することができます(※)。
テーブル備考	テーブル名の備考を入力します。データベースに定義するテーブルのコメントとして使用されます。

※あらかじめ辞書データを作成しておく必要があります。詳細は『[5.7 日本語-英語変換機能](#)』を参照してください。

上記の項目を入力した後、以下のいずれかのボタンをクリックしてください。

ボタン	機能
選択した行の下に挿入	テーブル一覧表の選択行の下にテーブルを追加します。
最下行に追加	テーブル一覧表の最下行にテーブルを追加します。 ※初期状態では No.1 にテーブルを追加します。
キャンセル	処理を行いません。

[テーブル削除]

テーブル一覧表からテーブルを削除します。

テーブル一覧の任意の行を選択してからこのボタンをクリックすると、選択された行のテーブル仕様シートが削除されます。複数行を選択することも可能です。

【テーブル順序入替(上)】

テーブル一覧表の行を 1 つ上に移動します。

テーブル一覧の任意の行を選択してからこのボタンをクリックすると、選択した行が 1 つ上の行と入れ替わります。

【テーブル順序入替(下)】

テーブル一覧表の行を 1 つ下に移動します。

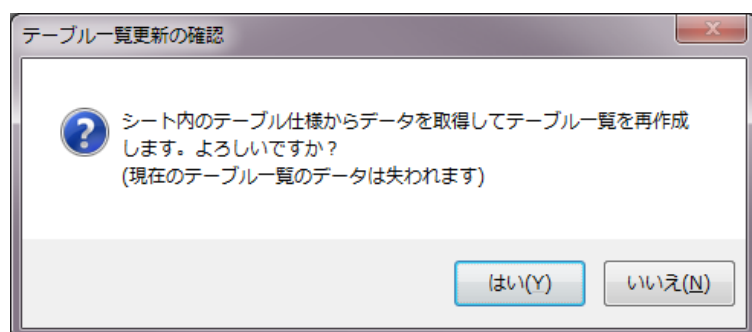
テーブル一覧の任意の行を選択してからこのボタンをクリックすると、選択した行が 1 つ下の行と入れ替わります。

【テーブル一覧更新】

現在のテーブル仕様シートを元にテーブル一覧表を再作成します。

あらかじめテーブル仕様シートが 1 つ以上作成されていることが必要です。

このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください。

ボタン	機能
はい	テーブル仕様書に入力されている情報を元に、テーブル一覧データを再作成します。
いいえ	処理を行いません。

【項目名変換(日->英)】

すべてのテーブル仕様シートに対して、項目名(日本語)に入力されたデータを元に、日本語-英語変換辞書を参照して項目名(英語)を自動入力します(※)。

【項目名変換(英->日)】

すべてのテーブル仕様シートに対して、項目名(英語)に入力されたデータを元に、日本語-英語変換辞書を参照して項目名(日本語)を自動入力します(※)。

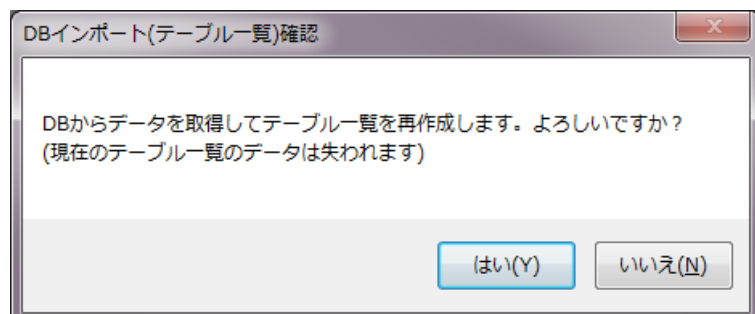
※あらかじめ辞書データを作成しておく必要があります。詳細は『[5.7 日本語-英語変換機能](#)』を参照してください。

[DB インポート(テーブル一覧)]

データベースに接続します。

データベースに定義されているテーブル一覧情報を取得し、テーブル一覧データを再作成します。

このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください

ボタン	機能
はい	データベースからテーブル一覧情報を取得し、テーブル一覧データを再作成します。
いいえ	処理を行いません。

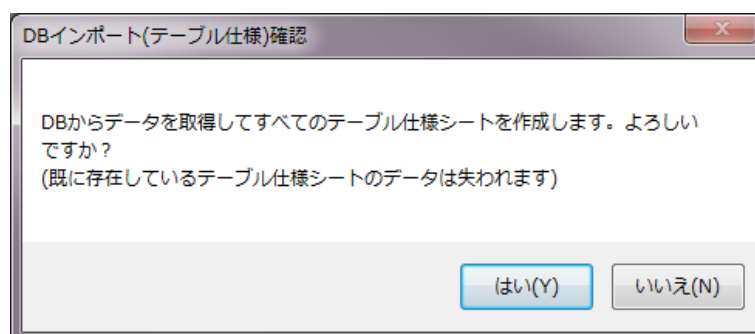
[DB インポート(テーブル仕様)]

データベースに接続します。

テーブル一覧に入力されているすべてのテーブル定義情報をデータベースから取得し、テーブル仕様シートの内容を更新します。

すでにテーブル仕様書シートが存在している場合でも、データを更新するので十分ご注意ください。

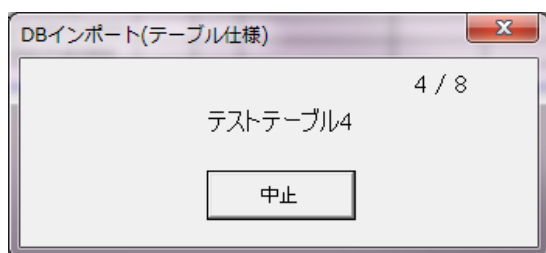
このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください。

ボタン	機能
はい	データベースからすべてのテーブル定義情報を取得し、テーブル仕様データを一括作成します。
いいえ	処理を行いません。

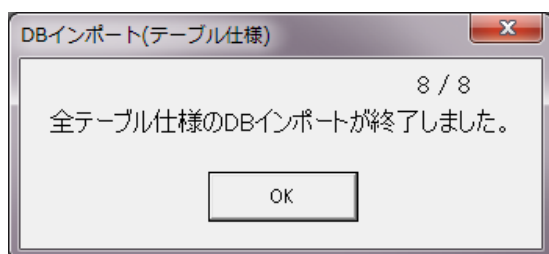
処理実行中は以下のようなメッセージで処理の状況が表示されます。



中央に現在作成中のテーブル名、右上に、“作成中のテーブル数 / 全テーブル数” を表示します。
[中止]ボタンを押すと、処理を途中で中止することができます。

ボタン	機能
中止	テーブル仕様シートの作成処理を中止します。

テーブル仕様の作成処理が最後まで終了すると以下のメッセージが表示されます。



[OK]ボタンを押してこのメッセージを終了してください。

[CREATE 文出力]

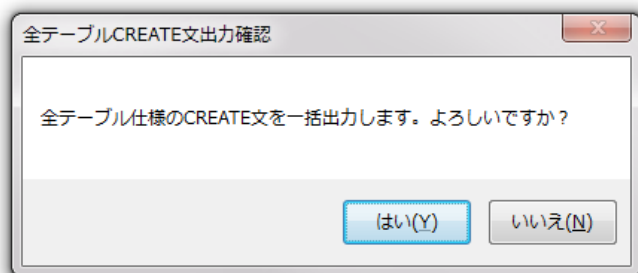
テーブル仕様に定義されている各テーブル定義情報に基づき、**CREATE** 文を一括作成します。
ただし出力するファイルは 1 テーブルごとに 1 ファイルとなります。

出力するファイル名は、“CT_<テーブル名(英語)>.sql” となります。
ファイル出力先は、環境設定の“**CREATE 文出力フォルダ**”で指定した場所になります。

【注意】

出力ファイル名の大文字/小文字は区別されません。たとえば、テーブル名(英語)で、“TableName”と“TABLENAME”を作成した場合、**CREATE** 文出力時にどちらも同じファイル名で出力します。その結果、ファイルの内容はあとから出力した内容で上書きされてしまいます。
この問題を回避するために、環境設定の“**CREATE 文出力フォルダ**”を変更し、対象となるテーブル仕様シートから**[CREATE 文出力]**を実行してください。

このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください

ボタン	機能
はい	すべてのテーブル定義情報から、CREATE 文を一括出力します。
いいえ	処理を行いません

テーブル一覧データ

各項目の内容は以下のようになっています

No.

DB インポート時	CREATE 文出力時
データベースのテーブル一覧から、テーブル情報を取得した順に 1 から始まる連続する数字を表示します。	No.に入力されているデータと関係なく、一番上の行から順に CREATE 文を出力します。

テーブル一覧の、[テーブル追加], [テーブル削除], [テーブル順序(上)], [テーブル順序(下)], [テーブル更新] ボタンを使用した場合は、No.は自動で 1 から始まる数値を割り付けます。

注意

手作業でテーブル一覧データを追加、削除した場合は、No.の表示は自動更新されません。
また No.欄には空白を入力しないでください。

テーブル名(日本語)

DB インポート時	CREATE 文出力時
データベースのテーブル一覧から、テーブルコメントの 1 番目の要素(※)を取得して表示します。	入力された値を、テーブル定義のテーブルコメントの 1 番目の要素(※)にセットします。

テーブル名(英語)

DB インポート時	CREATE 文出力時
データベースのテーブル一覧から、テーブル名を取得して表示します。	入力された値を、テーブル定義のテーブル名にセットします。

備考

DB インポート時	CREATE 文出力時
データベースのテーブル一覧から、テーブルコメントの 2 番目の要素(※)を取得して表示します。	入力された値を、テーブル定義のテーブルコメントの 2 番目の要素(※)にセットします。

※ テーブルコメントの要素

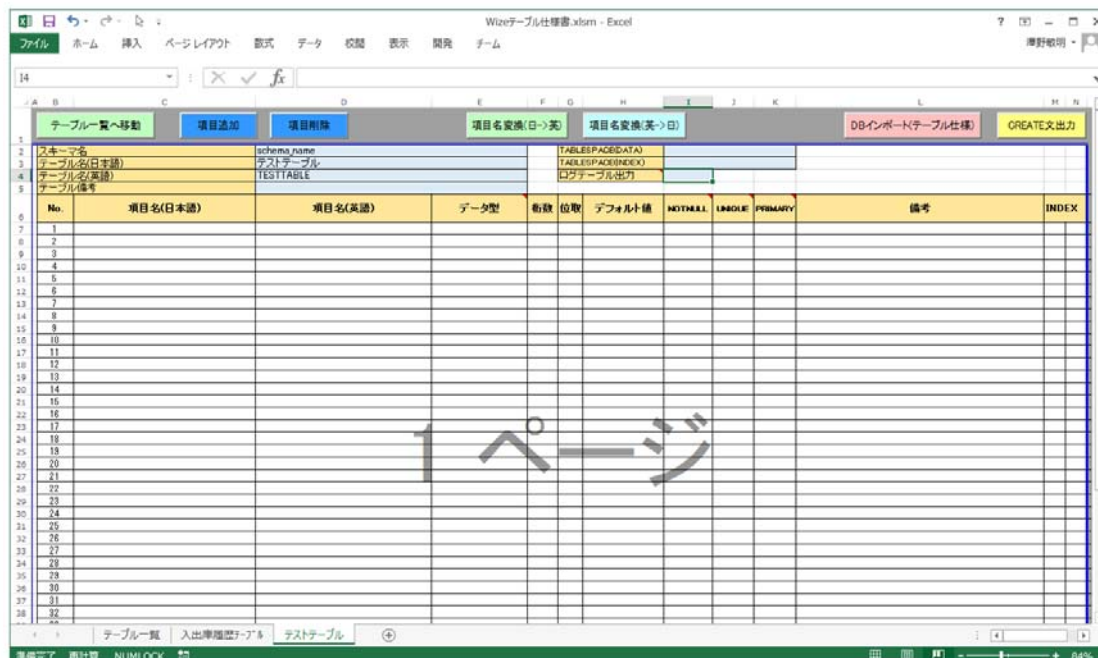
テーブル仕様書では、データベースのテーブルコメントを、1 番目の要素(日本語テーブル名)と 2 番目の要素(テーブルの備考)を、スペースで区切って格納するというルールを設定し、それに従って DB インポート, CREATE 文出力を行います。

シートの印刷

- ・印刷範囲はテーブル一覧データが入力されている領域のみに限定しています。
- ・ヘッダ部に、シート名(テーブル一覧), 印刷日時、ヘッダ部にページ番号/総ページ数、Copyright を表示します。

5.6. テーブル仕様シート

テーブル仕様シートは、作成したテーブル仕様の数に応じて複数作成されますが、テンプレートシート (Table Template) からコピーして作成されるので、各ボタンや表示機能はすべて共通となります。テーブル仕様シートは以下のような形式になっています。



各ボタンの機能

[テーブル一覧へ移動]

テーブル一覧シートへ移動します。

[項目追加]

選択した行の上にデータ項目を追加します。複数行を選択することも可能です。

[項目削除]

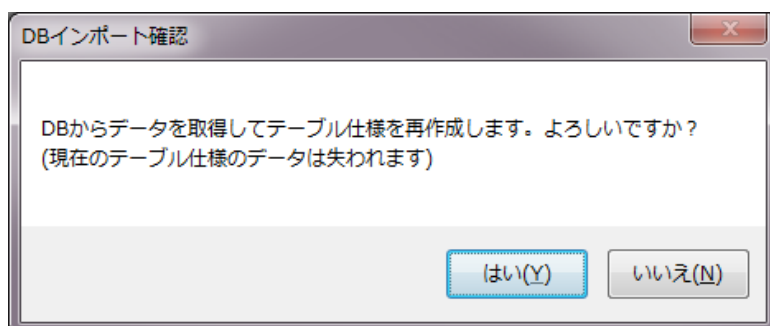
選択した行のデータ項目を削除します。複数行を選択することも可能です。

[DB インポート(テーブル仕様)]

データベースに接続します。

選択したシートに対応するテーブル定義情報をデータベースから取得し、テーブル仕様シートの内容を更新します。

このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください

ボタン	機能
はい	データベースからテーブル定義情報を取得し、テーブル仕様書データを再作成します。
いいえ	処理を行いません。

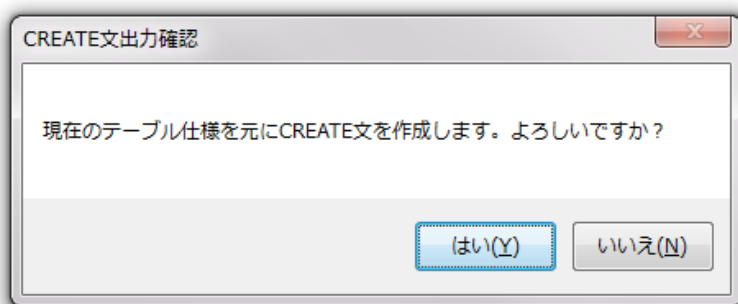
[CREATE 文出力]

テーブル仕様シートに入力されたデータから、CREATE 文を出力します。

出力されるファイル名は、"CT_<テーブル名(英語)>.sql" です。

ファイル出力先は、環境設定の CREATE 文出力フォルダで指定した位置です。

このボタンをクリックすると、以下の確認メッセージが表示されます。



以下のいずれかのボタンをクリックしてください

ボタン	機能
はい	現在のテーブル定義情報から CREATE 文を作成します。
いいえ	処理を行いません。

上記以外に、テーブル仕様に入力されたデータの整合性をチェックしますが、エラーメッセージの内容は、テーブル仕様データの各項目で説明します。

[項目名変換(日->英)]

項目名(日本語)に入力されたデータを元に、日本語-英語変換辞書を参照して項目名(英語)を自動入力することができます(※)。

[項目名変換(英->日)]

項目名(英語)に入力されたデータを元に、日本語-英語変換辞書を参照して項目名(日本語)を自動入力することができます(※)。

※あらかじめ辞書データを作成しておく必要があります。詳細は『[5.7 日本語-英語変換機能](#)』を参照してください。

テーブル仕様データ

スキーマ名

DB インポート時	CREATE 文出力時
データベースからは取得しません。 環境設定のスキーマ名に設定されたデータを表示します。	環境設定で”CREATE 文でスキーマ名を使用する”設定になっている場合、CREATE 文出力時に各テーブル仕様シート内に入力されたスキーマ名を使用します。

テーブル名(日本語)

DB インポート時	CREATE 文出力時
データベースのテーブルコメントの 1 番目の要素(※1)を取得して表示します。	CREATE 文でテーブルコメントの 1 番目の要素(※1)として出力します。

テーブル名(英語)

DB インポート時	CREATE 文出力時
データベースのテーブル名を取得して表示します。	CREATE 文でテーブル名として出力します。

備考

DB インポート時	CREATE 文出力時
データベースのテーブルコメントの 2 番目の要素(※1)を取得して表示します。	CREATE 文でテーブルコメントの 2 番目の要素(※1)として出力します。

※1 テーブルコメントの要素

テーブル仕様書では、データベースのテーブルコメントを、1 番目の要素(日本語テーブル名)と 2 番目の要素(テーブルの備考)を、スペースで区切って格納するというルールを設定し、それに従って DB インポート、CREATE 文出力を行います。

TABLESPACE(DATA)

DB インポート時	CREATE 文出力時
データベースのデータ格納用のテーブルスペース名を取得して表示します。	入力されたデータを、テーブル定義のデータ格納用テーブルスペース名として出力します。

TABLESPACE(INDEX)

DB インポート時	CREATE 文出力時
データベースのインデックス格納用のテーブルスペース名を取得して表示します。	入力されたデータを、テーブル定義のインデックス格納用テーブルスペース名として出力します。

ログテーブル出力

DB インポート時	CREATE 文出力時
参照されません。	○を入力した場合、ログテーブル出力用の SQL 文(ログテーブル生成、ログ出力トリガー)を出力します。

No.

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラム情報を取得した順に 1 から始まる連続する数字を表示します。	No.に入力されているデータと関係なく、1 番上の行から順に CREATE 文のカラム定義を出力します。

項目名(日本語)

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラムコメントの 1 番目の要素(※2)を取得して表示します。	カラムコメントの 1 番目の要素(※2)として出力します。

項目名(英語)

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラム名を取得して表示します。	テーブル定義のカラム名として出力します。

データ型

データベース	DB インポート時	CREATE 文出力時																																
Oracle	<p>データベースのテーブル定義から、データ型を取得して表示します 表示されるデータ型は Create 文で設定した内容と一致します。</p>	<p>データ型定義として出力します。 以下のデータ型は、データ型と桁数,位取りの内容をチェックします。</p> <p>VARCHAR2, NVARCHAR2, CHAR, NCHAR, LONG, CLOB, NUMBER, BINARY_FLOAT, BINARY_DOUBLE, DATE, RAW, BLOB, BFILE, ROWID, UROWID, FLAT</p> <p>上記以外のデータ型はチェックを行いませんので、ワークシートに入力したデータがそのままデータ型として出力されます。</p>																																
PostgreSQL	<p>データベースのテーブル定義から、データ型を取得して表示します。 ただし下記のデータ型は、Create 文で設定した内容と異なる名称で表示されます。</p> <table><tr><th>Create 文</th><th>データ型</th></tr><tr><td>bigint</td><td>int8</td></tr><tr><td>bigserial</td><td>int8 (注)</td></tr><tr><td>bit varying</td><td>varbit</td></tr><tr><td>boolean</td><td>bool</td></tr><tr><td>character varying</td><td>varchar</td></tr><tr><td>character</td><td>char</td></tr><tr><td>double precision</td><td>float8</td></tr><tr><td>integer</td><td>int4</td></tr><tr><td>decimal</td><td>numeric</td></tr><tr><td>real</td><td>float4</td></tr><tr><td>smallint</td><td>int2</td></tr><tr><td>smallserial</td><td>int2 (注)</td></tr><tr><td>serial</td><td>int4 (注)</td></tr><tr><td>time with time zone</td><td>timetz</td></tr><tr><td>Timestamp with time zone</td><td>timestamptz</td></tr></table> <p>(注) Create 文で指定したデータ型が自動増分整数のため、デフォルト値の欄に nextval('テーブル名_カラム名_seq'::regclass) NOTNULL の欄に ○が表示されます。</p>	Create 文	データ型	bigint	int8	bigserial	int8 (注)	bit varying	varbit	boolean	bool	character varying	varchar	character	char	double precision	float8	integer	int4	decimal	numeric	real	float4	smallint	int2	smallserial	int2 (注)	serial	int4 (注)	time with time zone	timetz	Timestamp with time zone	timestamptz	<p>データ型定義として出力します。 以下のデータ型は、データ型と桁数,位取りの内容をチェックします。</p> <p>varchar, character varying, char, character, bpchar, serial, serial4, serial8, bigserial, int, int4, int8, integer, bigint, decimal, numeric, real, float4, time, timestamp</p> <p>上記以外のデータ型はチェックを行いませんので、ワークシートに入力したデータがそのままデータ型として出力されます。</p>
Create 文	データ型																																	
bigint	int8																																	
bigserial	int8 (注)																																	
bit varying	varbit																																	
boolean	bool																																	
character varying	varchar																																	
character	char																																	
double precision	float8																																	
integer	int4																																	
decimal	numeric																																	
real	float4																																	
smallint	int2																																	
smallserial	int2 (注)																																	
serial	int4 (注)																																	
time with time zone	timetz																																	
Timestamp with time zone	timestamptz																																	

データベース	DB インポート時	CREATE 文出力時
SQLServer	未対応	<p>データ型定義として出力します。 以下のデータ型は、データ型と桁数,位取りの内容をチェックします。</p> <p>BIGINT, INT, SMALLINT, TINYINT, BIT, DECIMAL, NUMERIC, MONEY, SMALLMONEY, FLOAT, REAL, DATETIME, SMALLDATETIME, CHAR, VARCHAR, TEXT, NCHAR, NVARCHAR, NTEXT, BINARY, VARBINARY, IMAGE, SQL_VARIANT, TIMESTAMP, UNIQUEIDENTIFIER</p> <p>上記以外のデータ型はチェックを行いませんので、ワークシートに入力したデータがそのままデータ型として出力されます。</p>

桁数 および 位取

Oracle データベースの場合

DB インポート時			CREATE 文出力時			
データベースのテーブル定義から取得したデータ型に応じて、以下のように表示します。			入力されたデータにより、カラム定義のデータ型を以下のように出力します。			
データ型	桁数	位取	データ型	桁数	位取	CREATE 文
VARCHAR2	char_length	空白	VARCHAR2	n		VARCHAR2(n)
NVARCHAR2	char_length	空白	NVARCHAR2	n		NVARCHAR2(n)
CHAR	char_length	空白	CHAR			CHAR
NCHAR	char_length	空白	CHAR	n		CHAR(n)
CLOB	data_length	空白	NCHAR			NCHAR
NUMBER(x, y)	data_precision	data_scale	NCHAR	n		NCHAR(n)
NUMBER(x)	data_precision	空白	CLOB			CLOB
NUMBER	空白	空白	NUMBER			NUMBER
FLOAT(x)	data_precision	空白	NUMBER	p		NUMBER(p)
FLOAT	空白	空白	NUMBER	p	s	NUMBER(p, s)
BINARY_FLOAT	空白	空白	FLOAT			FLOAT
BINARY_DOUBLE	空白	空白	FLOAT	n		FLOAT(n)
DATE	空白	空白	BINARY_FLOAT			BINARY_FLOAT
TIMESTAMP	空白	空白	BINARY_DOUBLE			BINARY_DOUBLE
INTERVAL YEAR TO MONTH	空白	空白	DATE			DATE
INTERVAL DAY TO SECOND	空白	空白	ROWID			ROWID
ROWID	空白	空白	RAW			RAW
RAW	data_length	空白	RAW	n		RAW(n)
UROWID	data_length	空白	UROWID			UROWID
BLOB	空白	空白	UROWID	n		UROWID(n)
BLOB	空白	空白	BLOB			BLOB
BFILE	空白	空白	BFILE			BFILE
LONG	空白	空白	LONG			LONG

上記以外のデータ型が入力された場合は、データ型、桁数,位取りいずれも内容をチェックしません。

PostgreSQL データベースの場合

DB インポート時			CREATE 文出力時			
データベースのテーブル定義から取得したデータ型に応じて、以下のように表示します。			入力されたデータにより、カラム定義のデータ型を以下のように出力します。			
データ型	桁数	位取	データ型	桁数	位取	CREATE 文
int8 (bigint), int4 (integer), int2 (smallint)	空白	空白	bigint, integer, smallint			bigint, integer, smallint
float4 (real)	空白	空白	real, float4			real, float4
serial8 (bigserial), serial4 (serial), serial2 (smallserial)	空白	空白	bigserial, serial, smallserial			bigserial, serial, smallserial
Varchar (cahracter varying), bpchar (character, char)	char_length	空白	character varying, varchar			character varying, varchar
			character varying, varchar	n		character varying(n), varchar(n)
			character, char			character, char
numeric (decimal)	data_precision	data_scale	character, char	n		character(n), char(n)
date	空白	空白	numeric, decimal			numeric, decimal
time	data_precision	空白	numeric, decimal	p		numeric(p), decimal(p)
timetz (time with time zone)	data_precision	空白	numeric, decimal	p	s	numeric(p,s), decimal(p,s)
timestamp	data_precision	空白	date			date
timestamp (timestamp with time zone)	data_precision	空白	time, time with time zone			time, time with time zone
			timestamp, timestamp with time zone			timestamp, timestamp with time zone
			上記以外のデータ型が入力された場合は、データ型、桁数、位取りいずれも内容をチェックしません。			

SQLServer データベースの場合

DB インポート時	CREATE 文出力時																																																								
未対応	入力されたデータにより、カラム定義のデータ型を以下のように出力します。																																																								
	<table><tr><th>データ型</th><th>桁数</th><th>位取</th><th>CREATE 文</th></tr><tr><td>bigint, integer, smallint, tinyint, bit</td><td></td><td></td><td>bigint, integer, smallint, tinyint, bit</td></tr><tr><td>real, float</td><td></td><td></td><td>real, float</td></tr><tr><td>float</td><td>n</td><td></td><td>float(n)</td></tr><tr><td>money, smallmonry</td><td></td><td></td><td>money, smallmonry</td></tr><tr><td>char, varchar nchar, nvarchar, text, ntext</td><td></td><td></td><td>char, varchar nchar, nvarchar, text, ntext</td></tr><tr><td>char, varchar nchar, nvarchar</td><td>n</td><td></td><td>char(n), varchar(n), nchar(n), nvarchar(n)</td></tr><tr><td>numeric, decimal</td><td></td><td></td><td>numeric, decimal</td></tr><tr><td>numeric, decimal</td><td>p</td><td></td><td>numeric(p), decimal(p)</td></tr><tr><td>numeric, decimal</td><td>p</td><td>s</td><td>numeric(p,s), decimal(p,s)</td></tr><tr><td>datetime, smalldatetime</td><td></td><td></td><td>datetime, smalldatetime</td></tr><tr><td>binary, varbinary</td><td></td><td></td><td>binary, varbinary</td></tr><tr><td>binary, varbinary</td><td>n</td><td></td><td>binary(n), varbinary(n)</td></tr><tr><td>image, sql_variant, timestamp, uniqueidentifier</td><td></td><td></td><td>image, sql_variant, timestamp, uniqueidentifier</td></tr></table>	データ型	桁数	位取	CREATE 文	bigint, integer, smallint, tinyint, bit			bigint, integer, smallint, tinyint, bit	real, float			real, float	float	n		float(n)	money, smallmonry			money, smallmonry	char, varchar nchar, nvarchar, text, ntext			char, varchar nchar, nvarchar, text, ntext	char, varchar nchar, nvarchar	n		char(n), varchar(n), nchar(n), nvarchar(n)	numeric, decimal			numeric, decimal	numeric, decimal	p		numeric(p), decimal(p)	numeric, decimal	p	s	numeric(p,s), decimal(p,s)	datetime, smalldatetime			datetime, smalldatetime	binary, varbinary			binary, varbinary	binary, varbinary	n		binary(n), varbinary(n)	image, sql_variant, timestamp, uniqueidentifier			image, sql_variant, timestamp, uniqueidentifier
	データ型	桁数	位取	CREATE 文																																																					
	bigint, integer, smallint, tinyint, bit			bigint, integer, smallint, tinyint, bit																																																					
	real, float			real, float																																																					
	float	n		float(n)																																																					
	money, smallmonry			money, smallmonry																																																					
	char, varchar nchar, nvarchar, text, ntext			char, varchar nchar, nvarchar, text, ntext																																																					
	char, varchar nchar, nvarchar	n		char(n), varchar(n), nchar(n), nvarchar(n)																																																					
	numeric, decimal			numeric, decimal																																																					
	numeric, decimal	p		numeric(p), decimal(p)																																																					
	numeric, decimal	p	s	numeric(p,s), decimal(p,s)																																																					
	datetime, smalldatetime			datetime, smalldatetime																																																					
	binary, varbinary			binary, varbinary																																																					
	binary, varbinary	n		binary(n), varbinary(n)																																																					
	image, sql_variant, timestamp, uniqueidentifier			image, sql_variant, timestamp, uniqueidentifier																																																					
		上記以外のデータ型が入力された場合は、データ型、桁数、位取りいずれも内容をチェックしません。																																																							

デフォルト値

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラムの DATA DEFAULT を取得してその値を表示します。	データが入力されている場合、カラム定義に”DEFAULT” 情報を出力します。

NOTNULL

DB インポート時	CREATE 文出力時	
データベースのテーブル定義から、カラムの NULLABLE を取得し、NULLABLE 指定されている場合は○を表示します。	Oracle	○が入力されている場合、カラム定義に”NOT NULL ENABLE”を出力します。
	PostgreSQL SQLServer	○が入力されている場合、カラム定義に”NOT NULL”を出力します。

※DB インポート時に、Oracle では PRIMARY KEY に指定されているカラムは NOTNULL に○が表示されません。

UNIQUE

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラムの制限情報を取得し、UNIQUE 指定されている場合は○を表示します。	○が入力されている場合、カラム定義に”CONSTRAIN T UNIQUE” 情報を出力します。

PRIMARY

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラムの制限情報を取得し、PRIMARY 指定されている場合はそのポジション番号(1, 2, 3, ...)を表示します。	1 から始まる数値が入力されている場合、カラム定義に”CONSTRAINT PRIMARY” 情報を出力します。

備考

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、カラムコメントの 2 番目の要素(※2)を取得して表示します。	データが入力されている場合、カラムコメントの 2 番目の要素(※2)として”COMMENT” 情報を出力します。

INDEX

DB インポート時	CREATE 文出力時
データベースのテーブル定義から、INDEX 情報、および UNIQUE INDEX 情報を取得し、INDEX 情報はポジション番号(1, 2, 3, ...)を表示します。 UNIQUE INDEX 情報はポジション番号を(U1, U2, ...)のように表示します。	1 から始まる数値が入力されている場合、”INDEX”を出力します。 U1, U2, U3, ... のように U が付加された数値が入力されている場合、”UNIQUE INDEX” 情報を出力します。

※2 カラムコメントの要素

Wize テーブル仕様書では、データベースのカラムコメントを、1 番目の要素(日本語項目名)と 2 番目の要素(カラムの備考)を、スペースで区切って格納するというルールを設定し、それに従って DB インポート、CREATE 文出力を行います。

出力される CREATE 文の構成

CREATE 文を生成する処理では、Create 文作成用テンプレート内の各キーワードを所定の文字列と置換えます。下記に示したテンプレートは Oracle 用のものですが、テンプレートはデータベース種別ごとに個別に用意されているので、内容をカスタマイズすることが可能です。

```
-----
-- プロジェクト名: %project%
--   データベース: Oracle
--   テーブル名: %table_name_j%
--   説明: %table_remark%
--
-- %copyright%
-----

-----
-- %table_name_j% のバックアップを作成(%table_name_e%_SV)
-----

DROP TABLE %schema_name%"%table_name_e%_SV";
CREATE TABLE %schema_name%"%table_name_e%_SV" NOLOGGING PARALLEL AS SELECT * FROM %schema_name%"%table_name_e%";

-----
-- %table_name_j% を作成(%table_name_e%)
-----

DROP TABLE %schema_name%"%table_name_e%";
%CREATE_TABLE%

-----
-- バックアップからデータをコピー(%table_name_e%_SV -> %table_name_e%)
-----

SET serveroutput ON
DECLARE
  varret NUMBER;
  varmsg VARCHAR2(255);
BEGIN
  -- 文字のサイズが大→小となる場合はエラー
  varret := PKG_WIZEDBMASTER.COPYSAMECOLUMNData('%table_name_e%_SV', '%table_name_e%', 'E');
  SYS.DBMS_OUTPUT.PUT_LINE('return = ' || varret);
  IF (varret = 0) THEN
    varmsg := 'テーブルのコピーが成功しました。';
  ELSE
    varmsg := 'テーブルのコピーが失敗しました。';
  END IF;
  SYS.DBMS_OUTPUT.PUT_LINE(varmsg);
END;
/

-----
-- コメント
-----

COMMENT ON TABLE  %schema_name%"%table_name_e%" IS '%table_comment%';
%COMMENT_COLUMN_LIST%
```



```

-----
-- インデックス
-----

%INDEX_COLUMN_LIST%

#INSERT_UPDATE_TRG_START
-----
-- インサート, アップデートトリガー
-----

CREATE OR REPLACE TRIGGER TGIU_%table_name_e%
BEFORE INSERT OR UPDATE ON %table_name_e%
FOR EACH ROW
BEGIN
    IF (INSERTING) THEN
        %INSERT_TRIGGER_LIST%
    ELSIF (UPDATING) THEN
        %UPDATE_TRIGGER_LIST%
    END IF;
END;
/
#INSERT_UPDATE_TRG_END

#LOGTABLE_OUT_START
-----
-- ログ出力用テーブルを作成(LOG_%table_name_e%)
-----

DROP TABLE %schema_name%"LOG_%table_name_e%";
CREATE TABLE %schema_name%"LOG_%table_name_e%" (
    "UPDDATE" DATE
    , "MODULE" VARCHAR2(100)
    , "HOST" VARCHAR2(100)
    , "TERMINAL" VARCHAR2(100)
    , "IP_ADDRESS" VARCHAR2(15)
    , "OS_USER" VARCHAR2(100)
    , "OPEFLG" CHAR(1)
    , %LOG_CREATE_TABLE_CLM_LIST%
);
-----
-- ログ出力トリガー
-----

CREATE OR REPLACE TRIGGER TGL_%table_name_e%
AFTER INSERT OR UPDATE OR DELETE ON %table_name_e%
FOR EACH ROW
BEGIN
    IF (INSERTING) THEN
        INSERT INTO LOG_%table_name_e% (
            UPDDATE
            , MODULE
            , HOST
            , TERMINAL
            , IP_ADDRESS

```

```
, OS_USER
, OPEFLG
, %LOG_CLM_NAME_LIST%
) VALUES (
    CURRENT_DATE
, SYS_CONTEXT('userenv', 'module')
, SYS_CONTEXT('userenv', 'host')
, SYS_CONTEXT('userenv', 'terminal')
, SYS_CONTEXT('userenv', 'ip_address')
, SYS_CONTEXT('userenv', 'os_user')
, 'I'
, %LOG_CLM_DATA_NULL_LIST%
, %LOG_NEW_CLM_DATA_LIST%
);
ELSIF (UPDATING) THEN
    INSERT INTO LOG_%table_name_e% (
        UPDDATE
    , MODULE
    , HOST
    , TERMINAL
    , IP_ADDRESS
    , OS_USER
    , OPEFLG
    , %LOG_CLM_NAME_LIST%
    ) VALUES (
        CURRENT_DATE
    , SYS_CONTEXT('userenv', 'module')
    , SYS_CONTEXT('userenv', 'host')
    , SYS_CONTEXT('userenv', 'terminal')
    , SYS_CONTEXT('userenv', 'ip_address')
    , SYS_CONTEXT('userenv', 'os_user')
    , 'U'
    , %LOG_OLD_CLM_DATA_LIST%
    , %LOG_NEW_CLM_DATA_LIST%
    );
ELSIF (DELETING) THEN
    INSERT INTO LOG_%table_name_e% (
        UPDDATE
    , MODULE
    , HOST
    , TERMINAL
    , IP_ADDRESS
    , OS_USER
    , OPEFLG
    , %LOG_CLM_NAME_LIST%
    ) VALUES (
        CURRENT_DATE
    , SYS_CONTEXT('userenv', 'module')
    , SYS_CONTEXT('userenv', 'host')
    , SYS_CONTEXT('userenv', 'terminal')
    , SYS_CONTEXT('userenv', 'ip_address')
    , SYS_CONTEXT('userenv', 'os_user')
```

```

, 'D'
, %LOG_OLD_CLM_DATA_LIST%
, %LOG_CLM_DATA_NULL_LIST%
);
END IF;
END;
/
#LOGTABLE_OUT_END

%EOF%

```

※テーブル作成時にバックアップからデータを復帰する処理は、現在 Oracle 版のみの対応となります。
 またこの機能を使用するためには、データベース用パッケージ **WIZE_PKG** をインストールしておく必要があります。
 インストール方法は『[4.2 データベース用パッケージのインストール](#)』を参照してください。

キーワードと置換える文字の対応

SQL Template 内キーワード	置換える文字
%project%	環境設定の”プロジェクト名”
%table_name_j%	各テーブル仕様の”テーブル名(日本語)”
%table_remark%	各テーブル仕様の”テーブル備考”
%copyright%	コピーライト (内部固定)
%table_name_e%	各テーブル仕様の”テーブル名(英語)”
%CREATE_TABLE%	各テーブル仕様から生成した CREATE TABLE 句全体
%schema_name%	各テーブル仕様の”スキーマ名”
%table_comment%	各テーブルの”テーブル名(日本語)”と”テーブル備考”をスペースで連結したもの
%COMMENT_COLUMN_LIST%	各テーブル仕様の”項目名(日本語)”と”備考”をスペースで連結したもの
%INSERT_TRIGGER_LIST%	“トリガー設定”シートから生成したインサートトリガー設定文
%UPDATE_TRIGGER_LIST%	“トリガー設定”シートから生成したアップデートトリガー設定文
%LOG_CREATE_TABLE_CLM_LIST%	ログテーブル生成用の列定義
%LOG_CLM_NAME_LIST%	ログ出力トリガー用の列名
%LOG_CLM_DATA_NULL_LIST%	ログ出力トリガー用の NULL データ
%LOG_OLD_CLM_DATA_LIST%	ログ出力トリガー用の更新前データ(OLD)
%LOG_NEW_CLM_DATA_LIST%	ログ出力トリガー用の更新後データ(NEW)
%EOF%	置換えなし (ファイルの終りを判定するため)

シートの印刷

- 印刷範囲はテーブル仕様データが入力されている領域のみに限定しています。
 ただしテーブル仕様で INDEX 定義がひとつもない場合でも、INDEX 領域の 2 列分は印刷範囲に含まれます。
- ヘッダ部の表示
 テーブル名(日本語)、印刷日時を表示します。
- ヘッダ部の表示
 ページ番号/総ページ数、Copyright を表示します。

5.7. 日本語-英語変換機能

“テーブル一覧シート”のテーブル名(日本語)、テーブル名(英語)、および“テーブル仕様”シートの項目名(日本語)、項目名(英語)の入力作業を自動化するための機能です。

#DICシートに入力された変換用のデータを参照して、変換元に指定したデータを日本語→英語、または英語→日本語に一括変換することが可能です。

また、データ型、桁数、位取、デフォルト値、NOTNULL、UNIQUE、備考の値を定義しておくことにより、日本語-英語変換時に指定データが完全一致したときに、これらの設定値をテーブル仕様書にコピーすることが可能です。

1) テーブル一覧表

- ① [テーブル追加]ボタンを押下して表示されるテーブル追加ダイアログで、以下の操作が可能になります。
 - ・テーブル名(日本語)を入力し、[日本語→英語変換]ボタンを押下すると、テーブル名(英語)に対応する英語名が自動的に入力されます。
 - ・テーブル名(英語)を入力し、[英語→日本語変換]ボタンを押下すると、テーブル名(日本語)に対応する日本語名が自動的に入力されます。

- ② [項目名変換(日→英)], [[項目名変換(英→日)]]ボタンで、すべてのテーブル仕様シートの項目名について日本語-英語変換ができます。詳細は [2\) テーブル仕様書](#) を参照してください。

2) テーブル仕様書

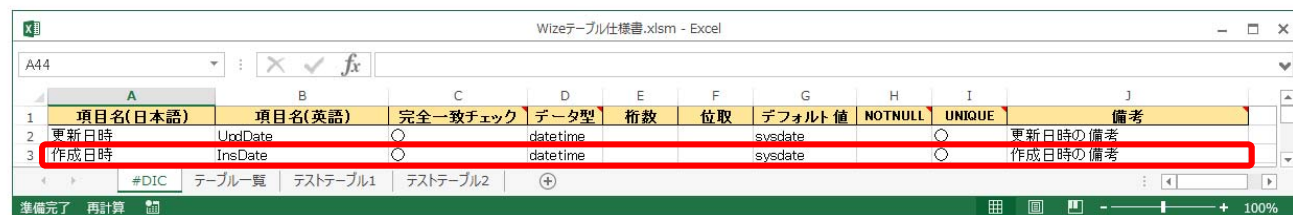
テーブル仕様書シートで、以下の操作が可能になります。

- ・項目名(日本語)にデータを入力後、[項目名変換(日->英)]ボタンを押下すると、項目名(英語)に対応する英語名が自動的に入力されます。
- ・項目名(英語)にデータを入力後、[項目名変換(英->日)]ボタンを押下すると、項目名(日本語)に対応する日本語名が自動的に入力されます。

<例> 項目名変換(日->英)ボタンを押下し、No.4 の項目を自動入力



<辞書データの設定>



3) 日本語-英語変換用辞書データの設定

日本語-英語変換機能を使用するためには、あらかじめ“#DIC”ワークシートに変換用データを入力しておく必要があります。

(1) 日本語-英語変換用データ

項目名(日本語), 項目名(英語)の欄に変換用データを入力します。

項目名(日本語), 項目名(英語)の両方にデータが入力されている場合のみ、変換対象となります。

(2) データ型等の設定用データ

「完全一致チェック, データ型, 桁数, 位取, デフォルト値, NOTNULL, UNIQUE, 備考」にデータを入力します。

完全一致チェック欄に“○”を入力した場合は、変換元に指定したデータが完全一致したとき、日本語-英語変換を行い、同時に「データ型, 桁数, 位取, デフォルト値, NOTNULL, UNIQUE, 備考」に入力された値がテーブル仕様書にコピーされます。

ただし、「備考」については以下の処理を行います。

#DIC シートの”備考”	テーブル仕様書の”備考”
空白	変更しない
データが入力されている	#DIC の”備考”がコピーされる

※各テーブル仕様書ごとに”備考”を個別に設定したい場合は、#DIC の”備考”は空白にしてください。

完全一致チェック欄に“○”以外を入力した場合は「データ型, 桁数, 位取, デフォ値, NOTNULL, UNIQUE, 備考」のコピーは行いません。

#DIC シートの設定は以下の例を参照してください。

項目名(日本語)	項目名(英語)	完全一致チェック	データ型	桁数	位取	デフォルト値	NOTNULL	UNIQUE	備考
更新日時	UpdDate	○	datetime						
作成日時	InsDate	○	datetime	16		sysdate	○	○	作成日時の備考
ロット番号	LotNo		varchar2						
実績	RSLT								
在庫	STOCK								
入庫	STORE								
出庫	DELIVERY								
履歴	HISTORY								
構成	CONSTITUTION								
原因	CAUSE								
報告	REPORT								
仕様	INPROC								
状態	STATUS								
設備	EQUIPMNT								
テーブル	TABLE								
計画	PLAN								
詳細	DETAIL								
マスター	MASTER								
工程	PROCESS								
テスト	TEST								
更新	UPDATE								
日	DATE								
モジュール	MODULE								
ホスト	HOST								
端末	TERMINAL								
アドレス	ADDRESS								
ユーザ	USER								
名	NAME								

5.8. トリガー設定機能

CREATE 文出力時に、以下の 2 種類のトリガー設定文を出力する機能です。

1) インサート/アップデートトリガー

あらかじめ”#トリガー設定”シートに必要なデータを入力しておくことにより、各テーブルに対して”作成”(INSERT)、“更新”(UPDATE) が実行されたときに、指定した列に対して自動的にデータをセットするためのトリガー設定文を出力します。

“#トリガー設定”シートの設定例を以下に示します。

No.	名称	列名(英語)	追加	更新	設定データ	DB種別
1	作成日時	InsertDate	<input type="radio"/>	<input type="radio"/>	NEW.InsertDate := SYSDATE	Oracle
2	作成モジュール	InsertModule	<input type="radio"/>	<input type="radio"/>	NEW.InsertModule := SYS_CONTEXT('userenv','module')	Oracle
3	作成ホスト	InsertHost	<input type="radio"/>	<input type="radio"/>	NEW.InsertHost := SYS_CONTEXT('userenv','host')	Oracle
4	作成ターミナル	InsertTerminal	<input type="radio"/>	<input type="radio"/>	NEW.InsertTerminal := SYS_CONTEXT('userenv','terminal')	Oracle
5	作成IPアドレス	InsertIPAddress	<input type="radio"/>	<input type="radio"/>	NEW.InsertIPAddress := SYS_CONTEXT('userenv','ip_address')	Oracle
6	作成作業員	InsertUserName	<input type="radio"/>	<input type="radio"/>	NEW.InsertUserName := SYS_CONTEXT('userenv','os_user')	Oracle
7	更新日時	UpdateDate	<input type="radio"/>	<input type="radio"/>	NEW.UpdateDate := SYSDATE	Oracle
8	更新モジュール	UpdateModule	<input type="radio"/>	<input type="radio"/>	NEW.UpdateModule := SYS_CONTEXT('userenv','module')	Oracle
9	更新ホスト	UpdateHost	<input type="radio"/>	<input type="radio"/>	NEW.UpdateHost := SYS_CONTEXT('userenv','host')	Oracle
10	更新ターミナル	UpdateTerminal	<input type="radio"/>	<input type="radio"/>	NEW.UpdateTerminal := SYS_CONTEXT('userenv','terminal')	Oracle
11	更新IPアドレス	UpdateIPAddress	<input type="radio"/>	<input type="radio"/>	NEW.UpdateIPAddress := SYS_CONTEXT('userenv','ip_address')	Oracle
12	更新作業員	UpdateUserName	<input type="radio"/>	<input type="radio"/>	NEW.UpdateUserName := SYS_CONTEXT('userenv','os_user')	Oracle
13	作成日時	InsertDate	<input type="radio"/>	<input type="radio"/>	new."INSERTDATE" := current_date	PostgreSQL
14	作成IPアドレス	InsertIPAddress	<input type="radio"/>	<input type="radio"/>	new."INSERTIPADDRESS" := inet_client_addr()	PostgreSQL
15	作成作業員	InsertUserName	<input type="radio"/>	<input type="radio"/>	new."INSERTUSERNAME" := current_user	PostgreSQL
16	更新日時	UpdateDate	<input type="radio"/>	<input type="radio"/>	new."UPDATEDATE" := current_date	PostgreSQL
17	更新IPアドレス	UpdateIPAddress	<input type="radio"/>	<input type="radio"/>	new."UPDATEIPADDRESS" := inet_client_addr()	PostgreSQL
18	更新作業員	UpdateUserName	<input type="radio"/>	<input type="radio"/>	new."UPDATEUSERNAME" := current_user	PostgreSQL

“#トリガー設定”シートの設定項目と機能の対応

列名	機能
No.	行番号(トリガー設定文に出力されないで、任意の値を入力してください)
名称	各行の名称(トリガー設定文に出力されないで、任意の値を入力してください)
列名(英語)	トリガー出力対象の列名を入力します。列名がテーブルに存在する場合にトリガー設定文を出力します。
追加	<input type="radio"/> が入力された場合、INSERT トリガー設定文(TGIU_{テーブル名(英語)})を出力します。
更新	<input type="radio"/> が入力された場合、UPDATE トリガー設定文(TGIU_{テーブル名(英語)})を出力します。
設定データ	トリガー設定文で実行する処理を入力します。
DB 種別	データベース種別をドロップダウンリストから選択します。 “環境設定”のデータベース種別と一致する項目のみトリガー設定文を出力します。

トリガー設定文の名称は INSERT, UPDATE 共通で、TGIU_{テーブル名(英語)} となります。

2) ログ出力トリガー

テーブル仕様書のログテーブル出力に”○”を設定した場合、”作成” (INSERT), “更新” (UPDATE), “削除” (DELETE)が実行されたときに、対応するログ出力テーブルを生成し、自動的にログデータを出力するためのトリガー設定文を出力します。

“テーブル仕様書”のログテーブル出力指定



ログ出力テーブルの構成は以下のようになります。

<Oracle 用>

列名	データ型	設定データ
UPDDATE	DATE	CURRENT_DATE
MODULE	VARCHAR(100)	SYS_CONTEXT('userenv', 'module')
HOST	VARCHAR(100)	SYS_CONTEXT('userenv', 'host')
TERMINAL	VARCHAR(100)	SYS_CONTEXT('userenv', 'terminal')
IP_ADDRESS	VARCHAR(15)	SYS_CONTEXT('userenv', 'ip_address')
OS_USER	VARCHAR(100)	SYS_CONTEXT('userenv', 'os_user')
OPEFLG	CHAR(1)	T (INSERT 時), U (UPDATE 時), D (DELETE) 時
BEF_{列名}	{列名}に対応	NULL (INSERT 時) or :OLD.{列名} (UPDATE 時, DELETE 時)
AFT_{列名}	{列名}に対応	:NEW.{列名} (INSERT 時, UPDATE 時) or NULL (DELETE 時)

<PostgreSQL 用>

列名	データ型	設定データ
UPDDATE	DATE	CURRENT_DATE
IP_ADDRESS	VARCHAR(15)	INET_CLIENT_ADDR0
DB_USER	VARCHAR(100)	CURRENT_USER
OPEFLG	CHAR(1)	T (INSERT 時), U (UPDATE 時), D (DELETE) 時
BEF_{列名}	{列名}に対応	NULL (INSERT 時) or OLD.{列名} (UPDATE 時, DELETE 時)
AFT_{列名}	{列名}に対応	NEW.{列名} (INSERT 時, UPDATE 時) or NULL (DELETE 時)

6. 処理実行時のチェック項目

- ・ Wize テーブル仕様書では、各処理実行時に以下のようなチェックを行います。
- ・ エラーが検出されると対応するメッセージを表示します。

6.1. テーブル一覧のチェック項目

[テーブル一覧更新] 実行時

- ・ テーブル仕様が 1 つ以上作成されていること。
- ・ テーブル仕様の “テーブル名(日本語)” がシート名と一致していること。

[DB インポート(テーブル仕様)] 実行時

- ・ テーブル一覧の “テーブル名(日本語)” が空白でないこと。
- ・ テーブル一覧の “テーブル名(日本語)” に、Excel シート名に使用できない文字が含まれていないこと。

Excel のシート名に使用できない文字は以下のとおりです(小文字, 大文字ともに)
“:”, “¥”, “/”, “?”, “*”, “[”, “”

- ・ テーブル一覧の “テーブル名(日本語)” が重複していないこと。

6.2. テーブル仕様のチェック項目

[DB インポート(テーブル仕様)] 実行時

- ・ テーブル仕様の “テーブル名(英語)” が空白でないこと。
- ・ テーブル仕様の “テーブル名(日本語)” が空白でないこと。

[CREATE 文出力] 実行時

- ・ テーブル仕様の “テーブル名(英語)” が空白でないこと。
- ・ テーブル仕様の “テーブル名(日本語)” が空白でないこと。
- ・ テーブル仕様の “項目名(英語)” が空白でないこと。
- ・ テーブル仕様の “データ型” が空白でないこと。

7. おわりに

本ソフトをダウンロードしていただきまして、ありがとうございます。

感想や要望、バグなどありましたらお手数ですが info@wizejp.com までご連絡ください。