

## 内容

寧静 DB ツール (njdba_japanese_1.00.xlsm) マニュアル .....	2
1. 概要 .....	2
1. 1 SQL 文の作成 .....	2
1. 2 テーブル及びデータの操作 .....	2
1. 3 CSV 及び DB データの妥当性チェック .....	2
1. 4 Java エンティティクラスの作成 .....	2
2. データベースとの接続の準備 (ODBC) .....	3
2. 1 ODBC ドライバーのインストール .....	3
2. 2 ODBC の設定 .....	3
3. テーブル定義仕様書の用意 .....	11
4. 初期設定 .....	14
4. 1 ワークフォルダ .....	15
4. 2 DB に接続 ODBC .....	15
4. 3 所有者 (スキーマ) .....	15
4. 4 DB 種類 .....	15
4. 5 Java パッケージ .....	15
4. 6 作成する SQL 方式 .....	16
5. ツールの機能操作 .....	17
5. 1 SQL 文の作成 .....	17
5. 2 テーブル及びデータの操作 .....	20
5. 3 CSV 及び DB データの妥当性チェック .....	23
5. 4 Java エンティティクラスの作成 .....	26
5. 5 下記テーブル定義の設定 .....	27
5. 6 全て選択・全て取消 .....	30
6. その他 .....	31

## 寧静 DB ツール (njdba\_japanese\_1.00.xlsm) マニュアル

### 1. 概要

寧静 DB ツールは Excel の VBA で作成、テーブル仕様書を基づいて、DB に関する操作とデータのチェックや、SQL 文と Java エンティティクラスの自動作成ができます。本ツールには PostgreSQL、MySQL、SqlServer、Oracle、DB2 など五つのデータベースがサポートされています。

プロジェクト開発に伴い、本ツールを利用すれば、効率と品質の上がりが期待されます。

#### 1. 1 SQL 文の作成

テーブル仕様書を基づいて、テーブル作成用 DDL と Insert、Update、Delete、Select の動的 SQL 文の自動作成ができます。作成する動的 SQL 文には、nj-soft、Prestatement、MyBatis、Other 四種類のフォーマットの選択は可能です。

#### 1. 2 テーブル及びデータの操作

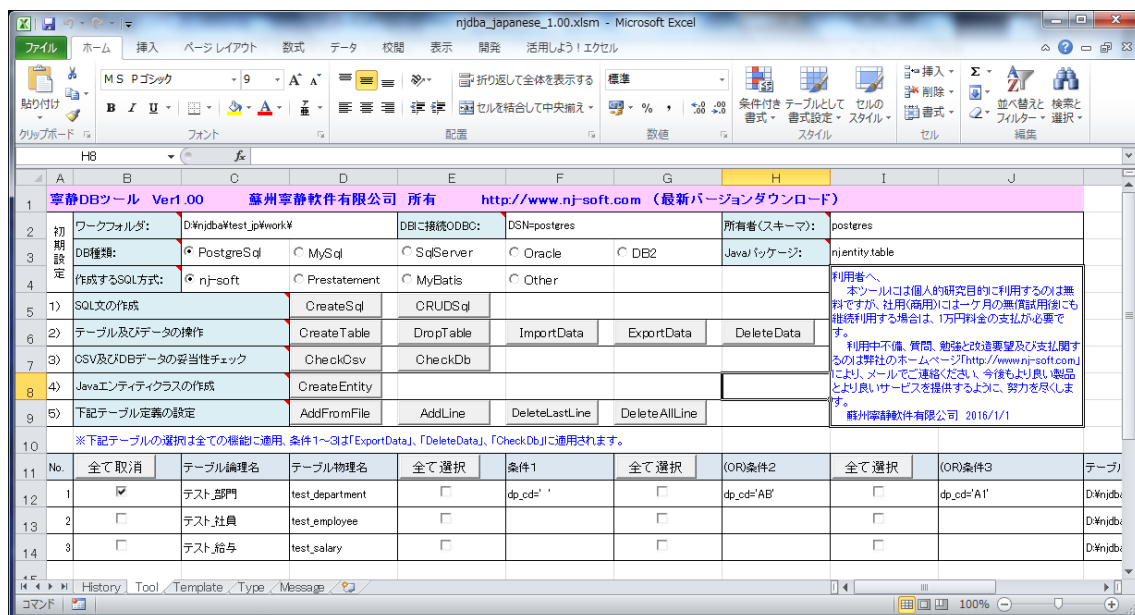
データベースに直接物理テーブルの作成と削除ができます、それとテーブルにデータのインポート、エクスポート、削除もできます。

#### 1. 3 CSV 及び DB データの妥当性チェック

妥当性チェックにはテーブル仕様のキー、NULL 不可のフィールドは NULL でないチェック、文字型設定されたフィールドは文字型チェック、値の正規表現設定されたフィールドは正規表現チェック、追加チェック設定されたフィールドは追加チェックなど 4 種類があります。チェックには CSV データあるいは DB データに対して、どちらもできます。

#### 1. 4 Java エンティティクラスの作成

テーブル仕様のフィールド名、フィールドタイプによって、Java エンティティクラスの自動作成ができます。(図 1-1 本ツールのメイン画面)



## 2. データベースとの接続の準備(ODBC)

本ツールには、Excel が 64 ビットの場合は 64 ビット ODBC しか使えません、その代り、Excel が 32 ビットの場合は 32 ビット ODBC しか使えません。

64 ビット ODBC を利用する場合は、コントロールパネルから ODBC の設定を行います。それと 64 ビット Excel を利用する場合は、データの追加チェックが行いません、追加チェックが必要な場合は 32 ビット Excel を利用してください。

なお、下記の説明は 32 ビット ODBC を基づいて展開します。

利用する DB の 32 ビット ODBC ドライバーをインストールしてから、「¥Windows¥SysWOW64¥odbcad32.exe」を実行して、ODBC の設置を行います、ODBC の設定にはデータベース名までの指定が必要です。

### 2. 1 ODBC ドライバーのインストール

1) PostgreSQL には下記の URL に 32 ビットのドライバーをダウンロードし、解凍し、実行してインストールします。

<http://www.postgresql.org/ftp/odbc/versions/msi/>

2) MySQL には下記の URL に 32 ビットのドライバーをダウンロードし、実行してインストールします。

<http://dev.mysql.com/downloads/connector/odbc/>

3) SqlServer には ODBC ドライバーが Windows 7 に既に存在するため、インストールは必要ありません。

4) Oracle には 32 ビット ODBC ドライバーを配置するために、Oracle の 32 ビット Client のインストールが必要です。下記は 11g の Client です。

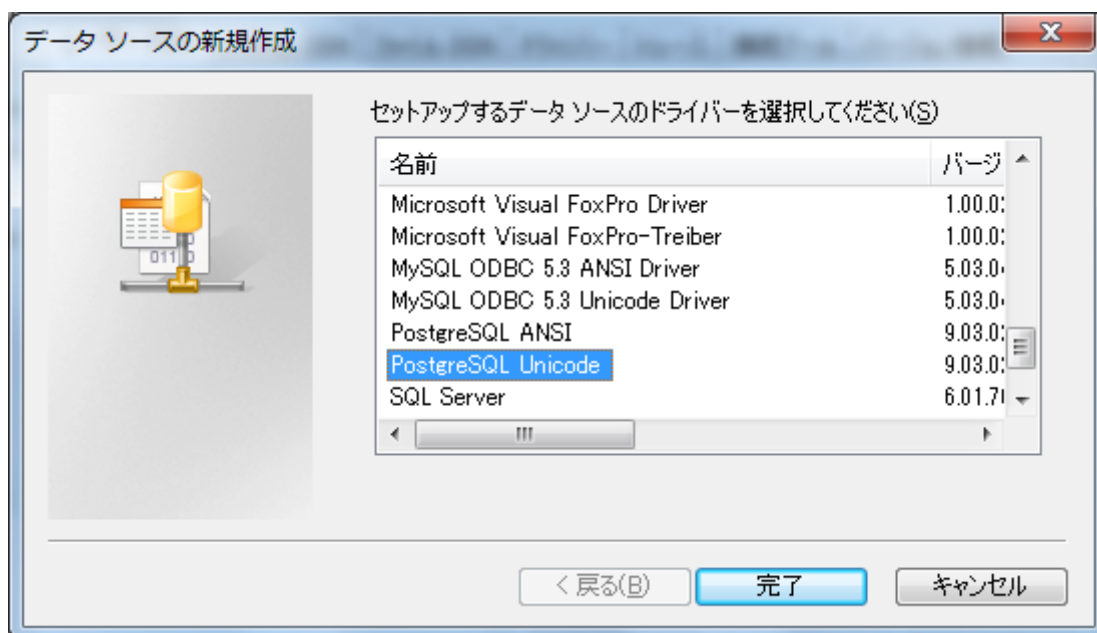
<http://www.oracle.com/technetwork/jp/database/enterprise-edition/downloads/12010-win32soft-098788-ja.html>

5) DB2 には下記の URL で適当な Express-C バージョンを選んでダウンロードしてインストールします、この中に ODBC ドライバーが含まれています。

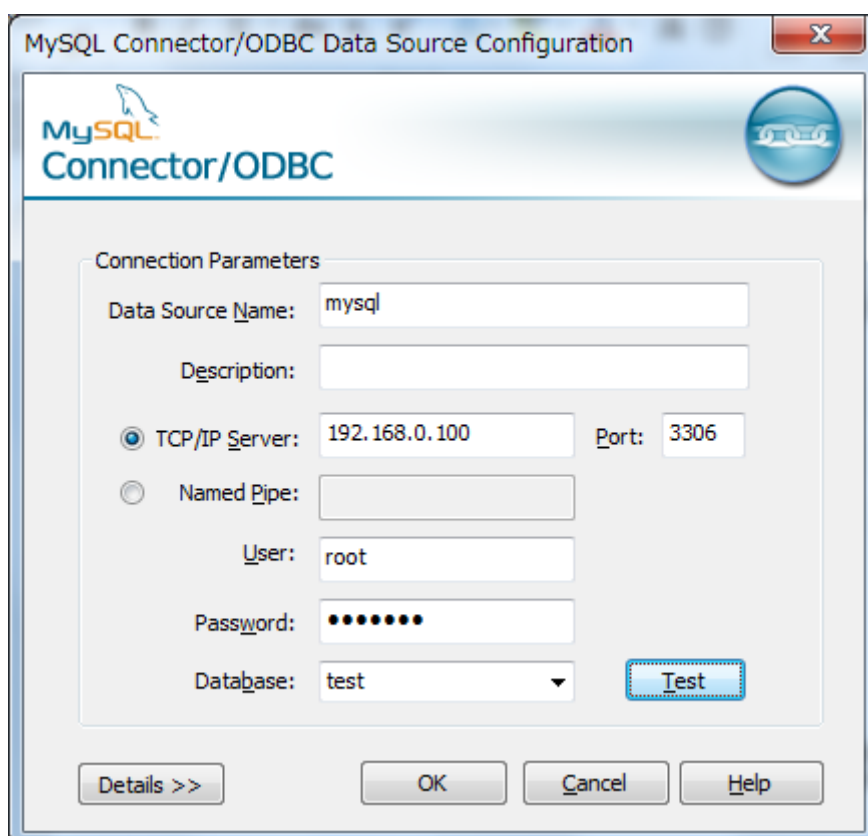
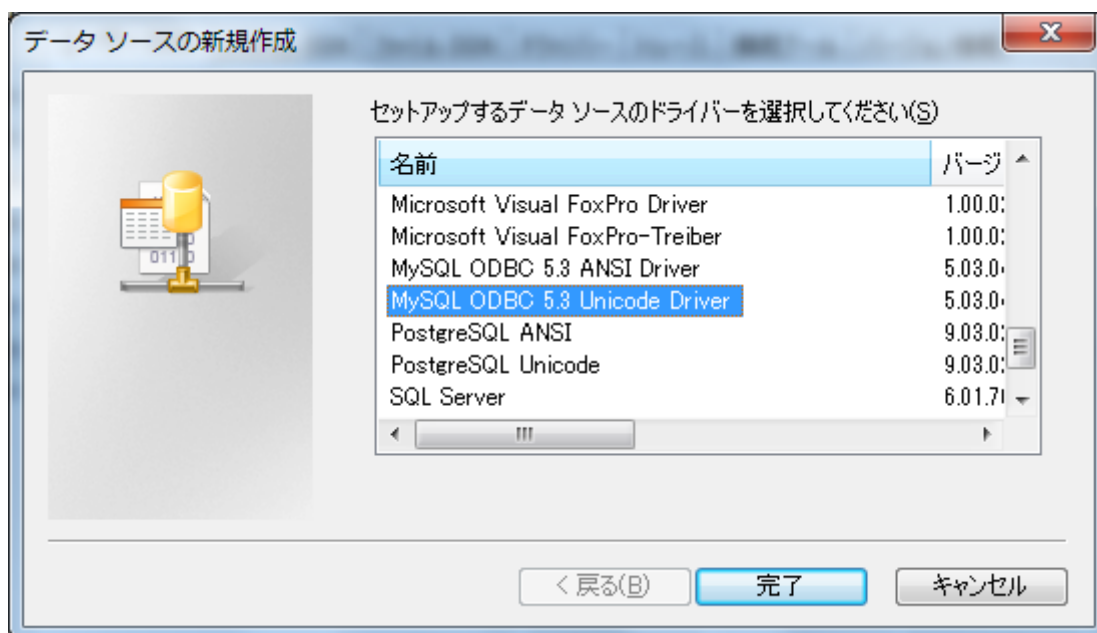
<http://www-01.ibm.com/software/data/db2/express-c/download.html>

### 2. 2 ODBC の設定

## 1) PostgreSQL の ODBC 設定 (図 2 - 1)



## 2) MySql の ODBC 設定 (図 2-2)



## 3) SqlServer の ODBC 設定 (図 2-3)

Microsoft SQL Server 用の DSN の設定

このウィザードは、SQL Server に接続するために使用する ODBC データ ソースを作成するお手伝いをします。

新しいデータ ソースの名前を入力してください。

名前(M): sqlserver

データ ソースについての説明を入力してください。

説明(D):

接続する SQL Server サーバー名を入力してください。

サーバー(S): USER-20150913JK¥SQLEXPRESS

完了 次へ(N) > キャンセル ヘルプ

Microsoft SQL Server 用の DSN の設定

SQL Server が、ログイン ID の権限の確認を行う方法は?

☐ ネットワークへのログイン ID で、Windows NT の認証メカニズムを使う(W)

☒ ユーザーが入力する SQL Server 用のログイン ID とパスワードを使う(S)

SQL Server との通信に使用するネットワーク ライブラリを変更するには、[クライアントの設定] をクリックします。

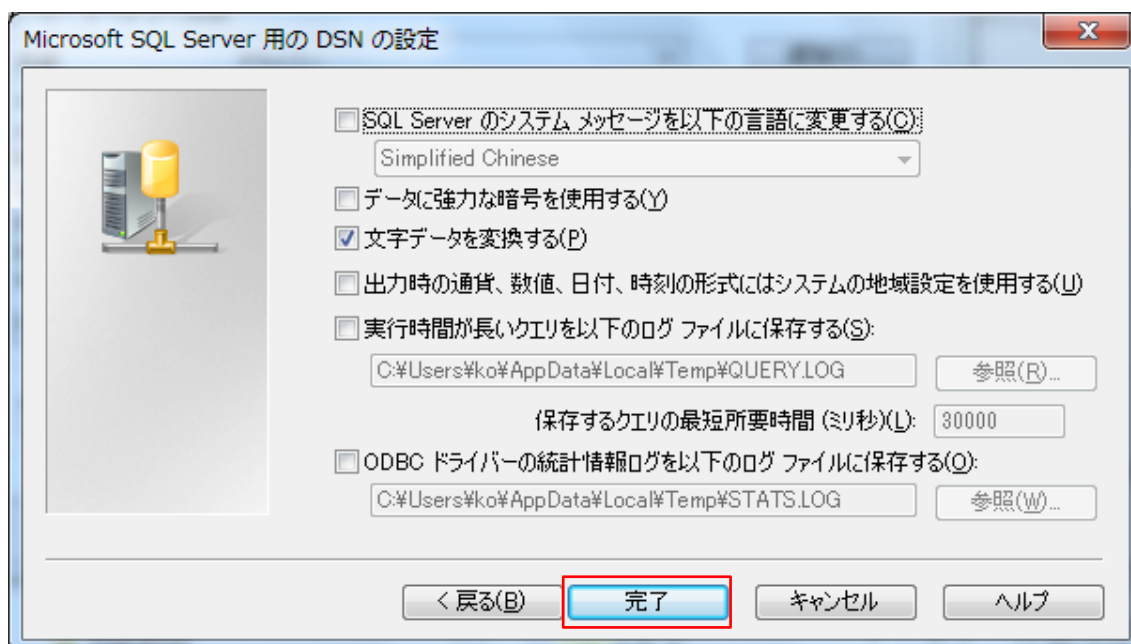
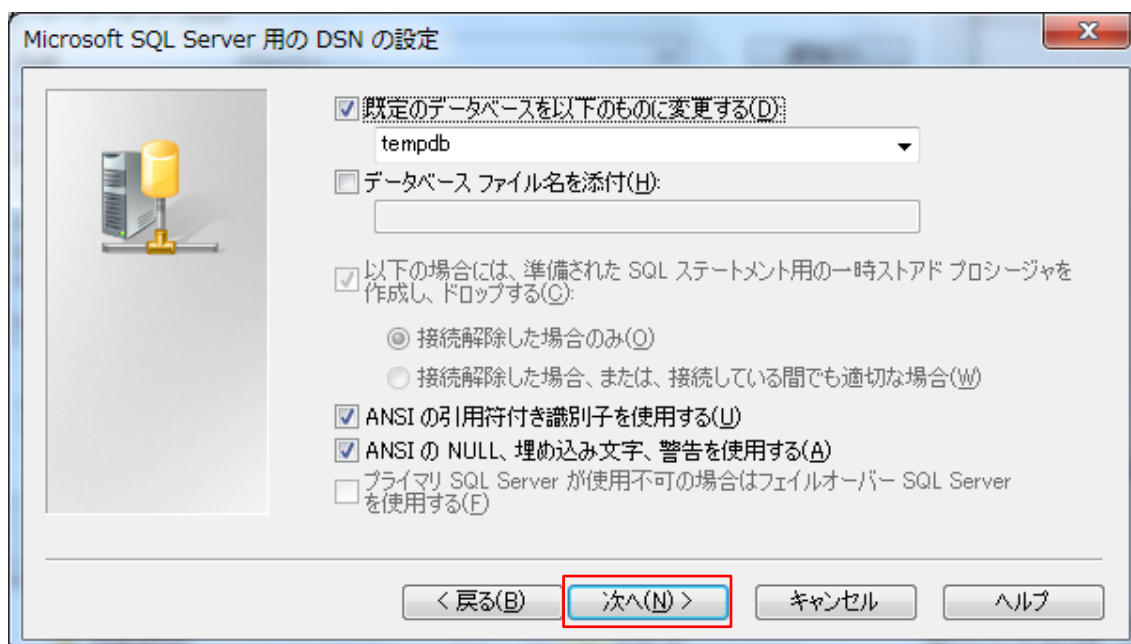
クライアントの設定(T) ...

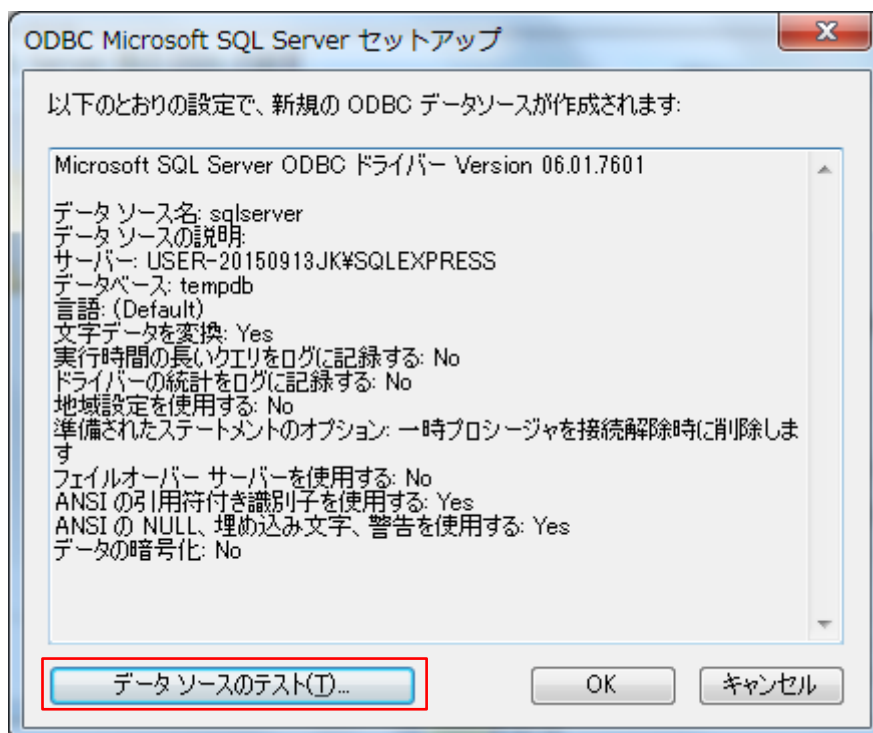
☒ SQL Server に接続して追加の構成オプションの既定設定を取得する(C)

ログイン ID(L): sa

パスワード(P):

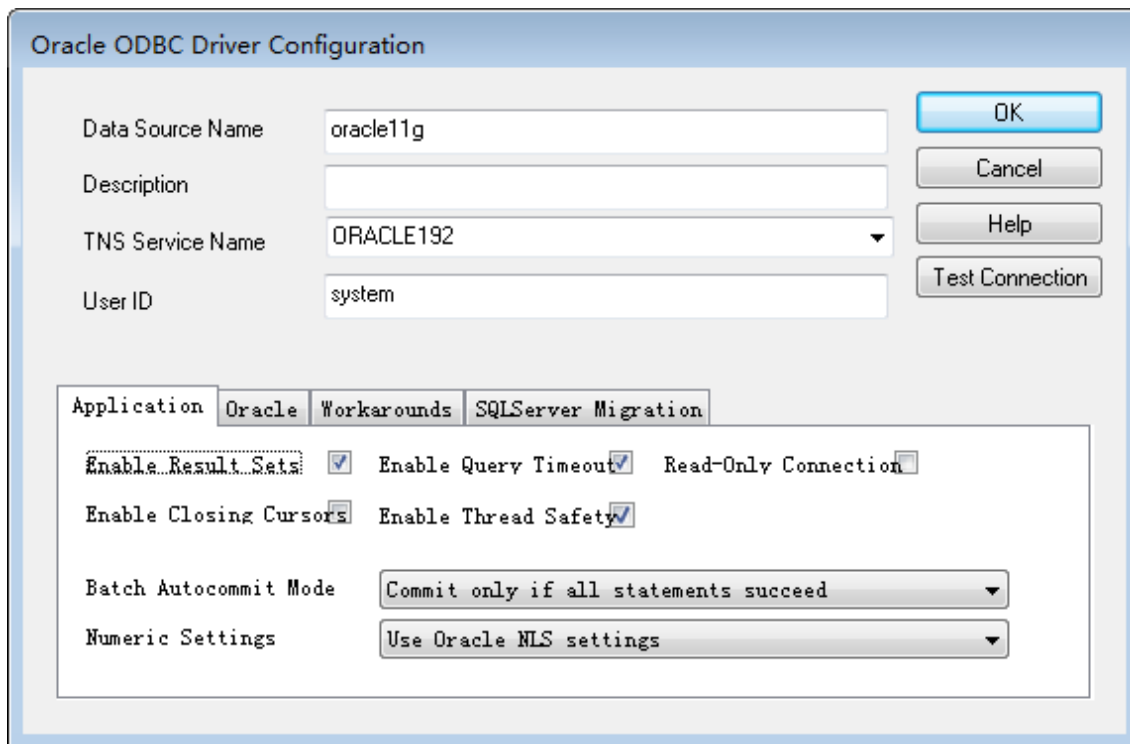
< 戻る(B) 次へ(N) > キャンセル ヘルプ







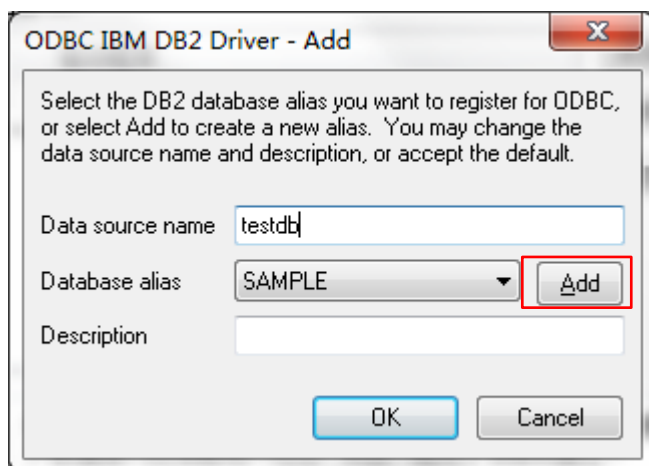
## 4) Oracle の ODBC 設定 (図 2 - 4)



The dialog box is titled "Oracle ODBC Driver Configuration". It contains the following fields and controls:

- Data Source Name: oracle11g
- Description: (empty)
- TNS Service Name: ORACLE192 (dropdown)
- User ID: system
- Buttons: OK, Cancel, Help, Test Connection
- Application tabs: Oracle (selected), Workarounds, SQLServer Migration
- Options:
  - Enable Result Sets: ☒ Enable Query Timeout: ☒ Read-Only Connection: ☐
  - Enable Closing Cursors: ☐ Enable Thread Safety: ☒
  - Batch Autocommit Mode: Commit only if all statements succeed (dropdown)
  - Numeric Settings: Use Oracle NLS settings (dropdown)

## 5) DB2 の ODBC 設定 (図 2 - 5)



The dialog box is titled "ODBC IBM DB2 Driver - Add". It contains the following fields and controls:

- Text: Select the DB2 database alias you want to register for ODBC, or select Add to create a new alias. You may change the data source name and description, or accept the default.
- Data source name: testdb
- Database alias: SAMPLE (dropdown)
- Buttons: Add (highlighted with a red box), OK, Cancel
- Description: (empty)

CLI/ODBC Settings - testdb

Data Source **TCP/IP** Security options Advanced Settings

Data source name: testdb

Description: testdb

User ID: db2admin

Password: ●●●●

☒ Save password

确定 取消 应用 (A) 帮助

CLI/ODBC Settings - testdb

Data Source TCP/IP Security options Advanced Settings

Database name: testdb

Database alias: testdba

Host name: 192.168.0.109

Port number: 50000

☐ The database physically resides on a host or QS/400 system.

☒ Connect directly to the server

☐ Connect to the server via the gateway

☐ DCS Parameters

,, INTERRUPT\_ENABLED, , , ,

Optimize for application

▼

确定 取消 应用 (A) 帮助

### 3. テーブル定義仕様書の用意

本ツールには弊社（蘇州寧静軟件有限公司）提供したテーブル定義書フォーマットしか使えません、それと一つのテーブルに一つ以上のキー項目が必要です。

下記は準備したテーブル定義書のサンプルです。（図3－1）

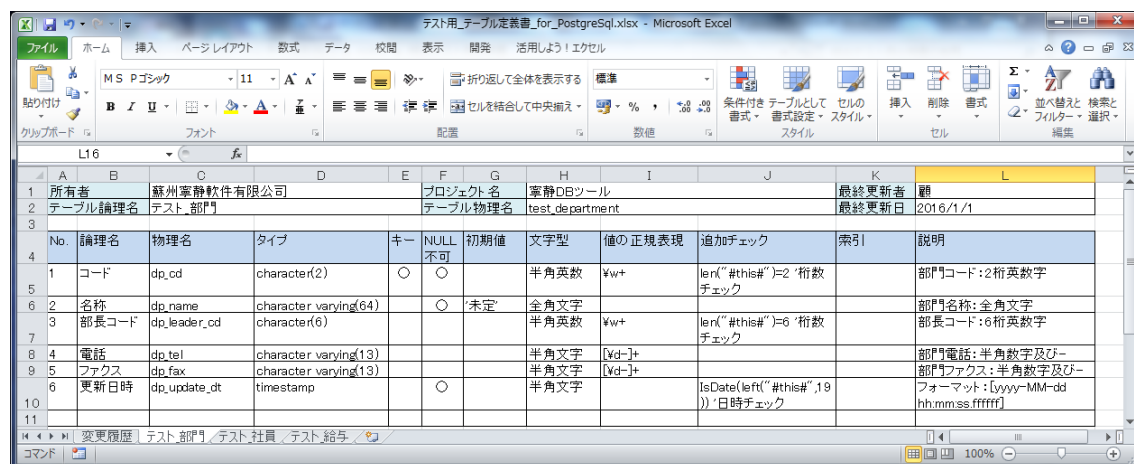


Figure 3-1 shows a screenshot of Microsoft Excel displaying a table definition template for a PostgreSQL database. The template is titled 'テスト用\_テーブル定義書\_for\_PostgreSQL.xlsm'. It contains a table with columns for 'No.', '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description). The table is divided into two sections: '所有者' (Owner) and 'テーブル物理名' (Table Physical Name). The '所有者' section includes fields for '所有者' (Owner), 'プロジェクト名' (Project Name), and '最終更新者' (Last Updated By). The 'テーブル物理名' section includes fields for 'テーブル物理名' (Table Physical Name), '最終更新日' (Last Updated Date), and '最終更新者' (Last Updated By). The table is also divided into two sections: 'No.' and '論理名' (Logical Name). The 'No.' section includes fields for 'No.', '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description). The '論理名' section includes fields for '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description).

No.	論理名	物理名	タイプ	キー	NULL 不可	初期値	文字型	値の正規表現	追加チェック	索引	説明
1	コード	dp_cd	character(2)	○	○		半角英数	¥w+	len("#this#")=2 '桁数チェック		部門コード:2桁英数字
2	名称	dp_name	character_varying(64)		○	未定	全角文字				部門名称:全角文字
3	部長コード	dp_leader_cd	character(6)				半角英数	¥w+	len("#this#")=6 '桁数チェック		部長コード:6桁英数字
4	電話	dp_tel	character_varying(13)				半角文字	[vd-]+			部門電話:半角数字及び-
5	ファクス	dp_fax	character_varying(13)				半角文字	[vd-]+			部門ファクス:半角数字及び-
6	更新日時	dp_update_dt	timestamp		○				IsDate(left("#this#",19)) '日時チェック		フォーマット:[yyyy-MM-dd hh:mm:ss.ffffff]

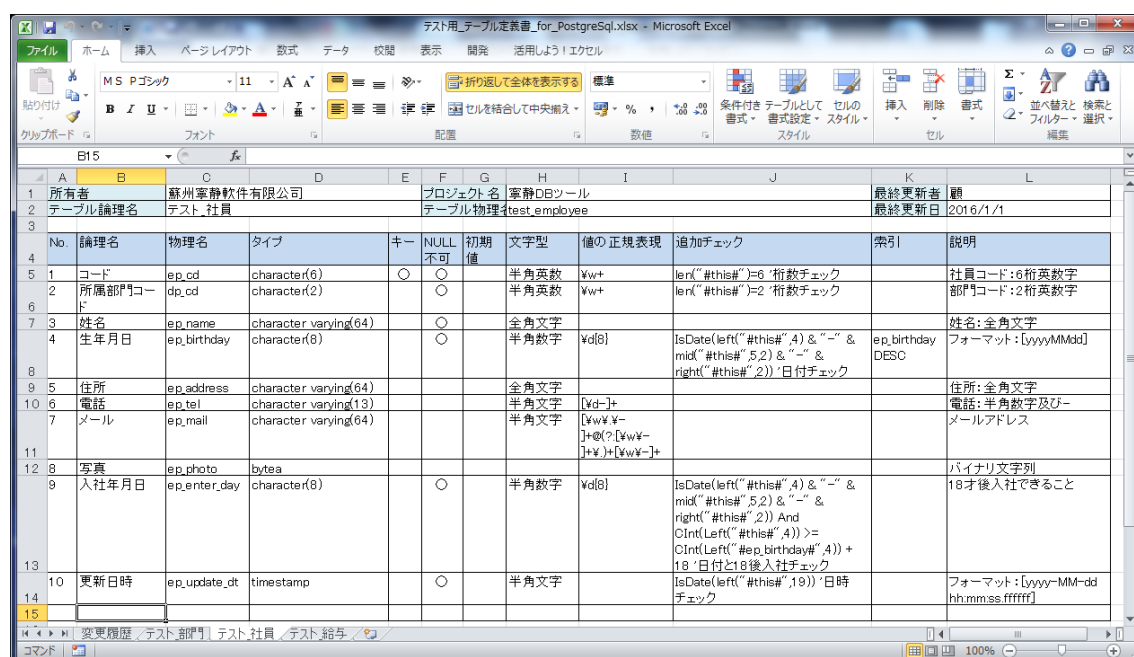


Figure 3-2 shows a screenshot of Microsoft Excel displaying a table definition template for a PostgreSQL database. The template is titled 'テスト用\_テーブル定義書\_for\_PostgreSQL.xlsm'. It contains a table with columns for 'No.', '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description). The table is divided into two sections: '所有者' (Owner) and 'テーブル物理名' (Table Physical Name). The '所有者' section includes fields for '所有者' (Owner), 'プロジェクト名' (Project Name), and '最終更新者' (Last Updated By). The 'テーブル物理名' section includes fields for 'テーブル物理名' (Table Physical Name), '最終更新日' (Last Updated Date), and '最終更新者' (Last Updated By). The table is also divided into two sections: 'No.' and '論理名' (Logical Name). The 'No.' section includes fields for 'No.', '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description). The '論理名' section includes fields for '論理名' (Logical Name), '物理名' (Physical Name), 'タイプ' (Type), 'キー' (Key), 'NULL 不可' (Not Null), '初期値' (Initial Value), '文字型' (Character Type), '値の正規表現' (Value Regular Expression), '追加チェック' (Additional Check), '索引' (Index), and '説明' (Description).

No.	論理名	物理名	タイプ	キー	NULL 不可	初期値	文字型	値の正規表現	追加チェック	索引	説明
1	コード	ep_cd	character(6)	○	○		半角英数	¥w+	len("#this#")=6 '桁数チェック		社員コード:6桁英数字
2	所属部門コード	dp_cd	character(2)		○		半角英数	¥w+	len("#this#")=2 '桁数チェック		部門コード:2桁英数字
3	姓名	ep_name	character_varying(64)		○		全角文字				姓名:全角文字
4	生年月日	ep_birthday	character(8)		○		半角数字	Vd[8]	IsDate(left("#this#",4) & "-" & mid("#this#",5,2) & "-" & right("#this#",2)) '日付チェック	ep_birthday DESC	フォーマット:[yyyyMMdd]
5	住所	ep_address	character_varying(64)				全角文字				住所:全角文字
6	電話	ep_tel	character_varying(13)				半角文字	[vd-]+			電話:半角数字及び-
7	メール	ep_mail	character_varying(64)				半角文字	[vw¥-]+@([vw¥-]+¥)+[vw¥-]+			メールアドレス
8	写真	ep_photo	bytea								バイナリ文字列
9	入社年月日	ep_enter_day	character(8)		○		半角数字	Vd[8]	IsDate(left("#this#",4) & "-" & mid("#this#",5,2) & "-" & right("#this#",2)) And CInt(Left("#this#",4)) >= CInt(Left("#ep_birthday#",4)) + 18 '日付と18歳入社チェック		18才後入社できること
10	更新日時	ep_update_dt	timestamp		○		半角文字		IsDate(left("#this#",19)) '日時チェック		フォーマット:[yyyy-MM-dd hh:mm:ss.ffffff]

No.	論理名	物理名	タイプ	キー	NULL 不可	初期値	文字型	値の正規表現	追加チェック	索引	説明
1	社員コード	ep_cd	character(6)	○	○		半角英数	^[a-zA-Z0-9]{6}\$	len("#this#")=6 '桁数チェック'		社員コード: 6桁英数字
2	給与年月	sl_ym	character(6)	○	○		半角数字	^(19 20)\d{2}(0[1-9] 1[0-2])\$			フォーマット: [yyyyMM]
3	基本給	sl_base	numeric(12,2)		○		半角数値		IsNumeric("#this#") '数字チェック'	sl_base DESC, sl_bonus ASC	基本給: 金額
4	残業時間	sl_over_time	real		○	0	半角数値		IsNumeric("#this#") '数字チェック'		残業時間: 数字
5	残業単価	sl_over_unit	integer		○	0	半角数値	^[0-9]+\$			残業単価: 金額
6	残業代	sl_over_money	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") And #this# = #sl_over_time# * #sl_over_unit# '数字と計算チェック'		残業代: 金額
7	ボーナス	sl_bonus	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") '数字チェック'		ボーナス: 金額
8	税金	sl_tax	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") '数字チェック'		税金: 金額
9	実給	sl_hand_money	numeric(12,2)		○		半角数値		IsNumeric("#this#") And #this# = #sl_base# + #sl_over_money# + #sl_bonus# - #sl_tax# '数字と実給の算出チェック'		実給: 金額
10	累計税金	sl_sum_tax	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") And #this# >= #sl_tax# '数字と累計チェック'		累計税金: 金額
11	更新日時	sl_update_dt	timestamp		○		半角文字		IsDate(left("#this#",19)) '日時チェック'		フォーマット: [yyyy-MM-dd hh:mm:ss.ffffff]

1) 「テーブル論理名」には作成する物理テーブル、SQL 文と Java エンティティクラスにコメントとして利用されます。それとテーブル論理名とシート名が一致する場合は仕様ファイルに該当シートがテーブル定義書かどうかの判定基準となります。

2) 「論理名」列には物理テーブル、SQL 文と Java エンティティクラスにコメント、CSV データの一行目のタイトルとして利用されます。

3) 「物理名」列には物理テーブル、SQL 文と Java エンティティクラスにフィールド名（項目名）、CSV データの二行目のタイトルとして利用されます。

4) 「タイプ」列には物理テーブルのフィールドのデータ型として利用されます、それと本ツールのシート「Type」に、フィールドのデータ型と Java 変数のタイプの対照表を利用して、Java エンティティクラスに項目のデータ型も決められます。

5) 「キー」列に○を付けるのはキー項目であり、キーフィールドとして物理テーブルを作成します。キー項目の CSV データの値には、

- Oracle の場合は、NULL と ” ” がインポートできません。
- それ以外の場合は、NULL がインポートできません。

6) 「NULL 不可」列に○を付けるのは NULL 不可項目であり、NULL 不可フィールドとして物理テーブルを作成します。NULL 不可項目の CSV データの値には、

- Oracle の場合は、NULL と ” ” がインポートできません。
- それ以外の場合は、NULL がインポートできません。

7) 「初期値」列に記入された場合は、該当フィールドの初期値として物理テーブルを作成します。

8) 「文字型」には下記の五つ種類があります、文字型が指定された場合は、CSV データと

DB データの文字型チェックができます。

- 半角数字 = [0-9]の半角数字のみです。
- 半角数値 = [0-9]と[-]と[.]の有効な数値です。
- 半角英数 = [0-9]と[a-z]と[A-Z]です。
- 半角文字 = 全ての半角文字です。
- 全角文字 = 全ての全角文字です。

9)「値の正規表現」は VBA の正規表現式であり、値の正規表現が指定された場合、CSV データと DB データの値の正規表現チェックができます。例)

[¥w+] = [a-zA-Z\_0-9]の一つ以上です。

[¥d{8}] = 8 桁の半角数字です。

1 0)「追加チェック」は自項目及び同じレコードの他項目との VBA の一行比較式であり、値の追加チェックが指定された場合、CSV データと DB データの追加チェックができます。例)

自項目の値は 2 桁です: `len("#this#")=2` '桁数チェック

自項目の値は有効な日付です: `IsDate("#this#")` '日付チェック

二つ項目の値の比較です: `"#from_date#"<="#to_date#"` '日付比較

※「#this#」は自項目、他項目は「#項目物理名#」とします。

※64 ビット Excel を利用する場合は、データの追加チェックが行いません、追加チェックが必要な場合は 32 ビット Excel を利用してください。

1 1)「索引」列に記入された場合は、テーブルの索引として物理テーブルを作成します。記入ルールは下記です。

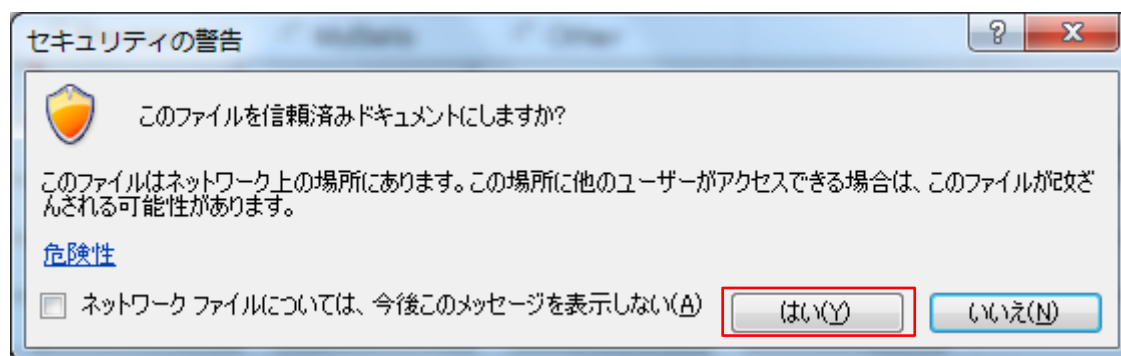
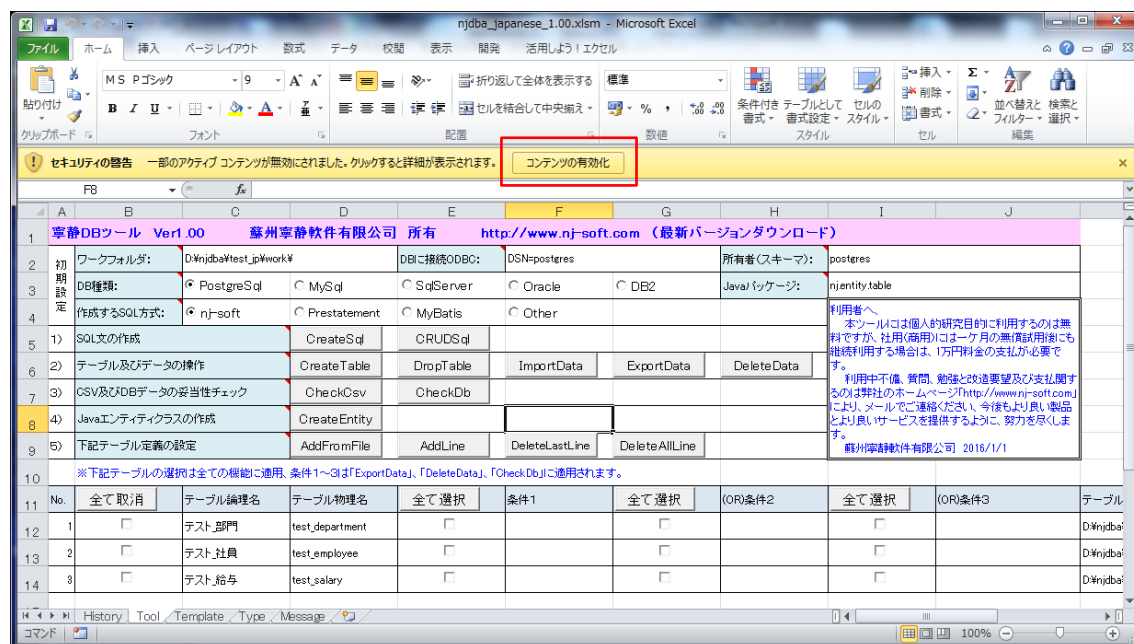
- 一つ項目索引の場合 = [項目物理名 DESC|ASC]
- 複数項目索引の場合 = [項目物理名 DESC|ASC, 項目物理名 DESC|ASC...]

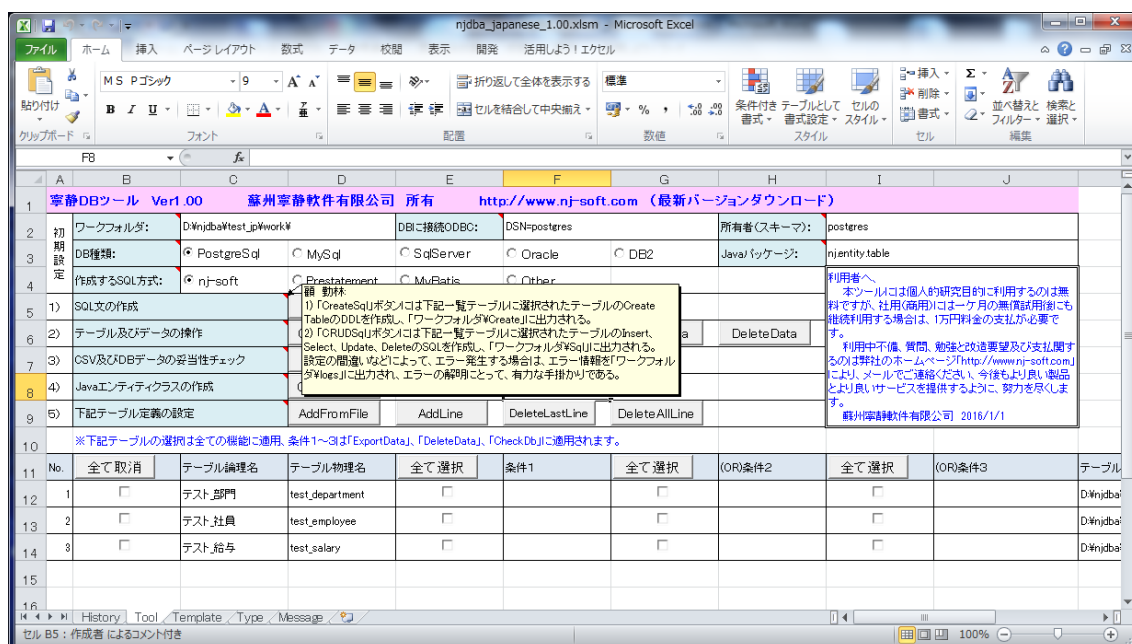
1 2)「説明」列には Java エンティティクラスのコメントとして利用されます。

#### 4. 初期設定

本ツール (njdba\_japanese\_1.00.xlsm) を Excel2007 以上で開いて、シート「Tool」に「初期設定」の下記項目の設定を行います。

なお、本ツールには各タイトルセルにコメントがあります、マウスのカーソルを適当に移動すれば、利用方法などの説明が表示されます。(図 4-1)





#### 4. 1 ワークフォルダ

ワークフォルダは入力と出力の所在フォルダーであり、既に存在することが必要です。

#### 4. 2 DB に接続 ODBC

ODBC は DB との接続用であり、右のセルには「DSN=データソース名」のみを記入すればよいですが、SqlServer と Oracle の場合は「DSN=データソース名;UID=ユーザ名;PWD=パスワード」の記入が必要になります、その原因は SqlServer と Oracle の場合は ODBC の設定にユーザ名とパスワードの保存ができないからです。

#### 4. 3 所有者 (スキーマ)

DB によって、記入は少し違います。PostgreSQL はデータベース名、MySQL はスキーマ名、SqlServer は所有者名、Oracle は所有者名、DB2 はスキーマ名で記入すればいいです。

#### 4. 4 DB 種類

DB 種類には利用するデータベースの選択が必要です。現時点には五つのデータベースがサポートされていますが、バイナリの長文字列フィールドには CSV ファイルのインポートとエクスポートの際、NULL として取り扱います、それと Oracle の「interval day to second、interval year to month、timestamp with local time zone、timestamp with time zone」データ型のフィールドには CSV ファイルのインポートとエクスポートがサポートされていないため、項目自体が省略されています。

#### 4. 5 Java パッケージ

Java エンティティクラス作成する際、パッケージ名として使われています。ご自身に合わせて適宜にご修正ください。

#### 4. 6 作成する SQL 方式

SQL 方式とは作成する Insert、Select、Update、Delete の SQL 文のフォーマットです、「nj-soft」方式には寧静フレームワーク（弊社のメイン製品）が使われているフォーマット、「Prestatement」方式には普段の JDBC の Prestatement が使われているフォーマット、「MyBatis」方式には MyBatis が使われているフォーマット、「Other」方式には利用者に合い、ご自身で設定可能なフォーマットです。「Other」方式の修正には「テンプレート」シートの「2.4)Other テンプレート」を参照。

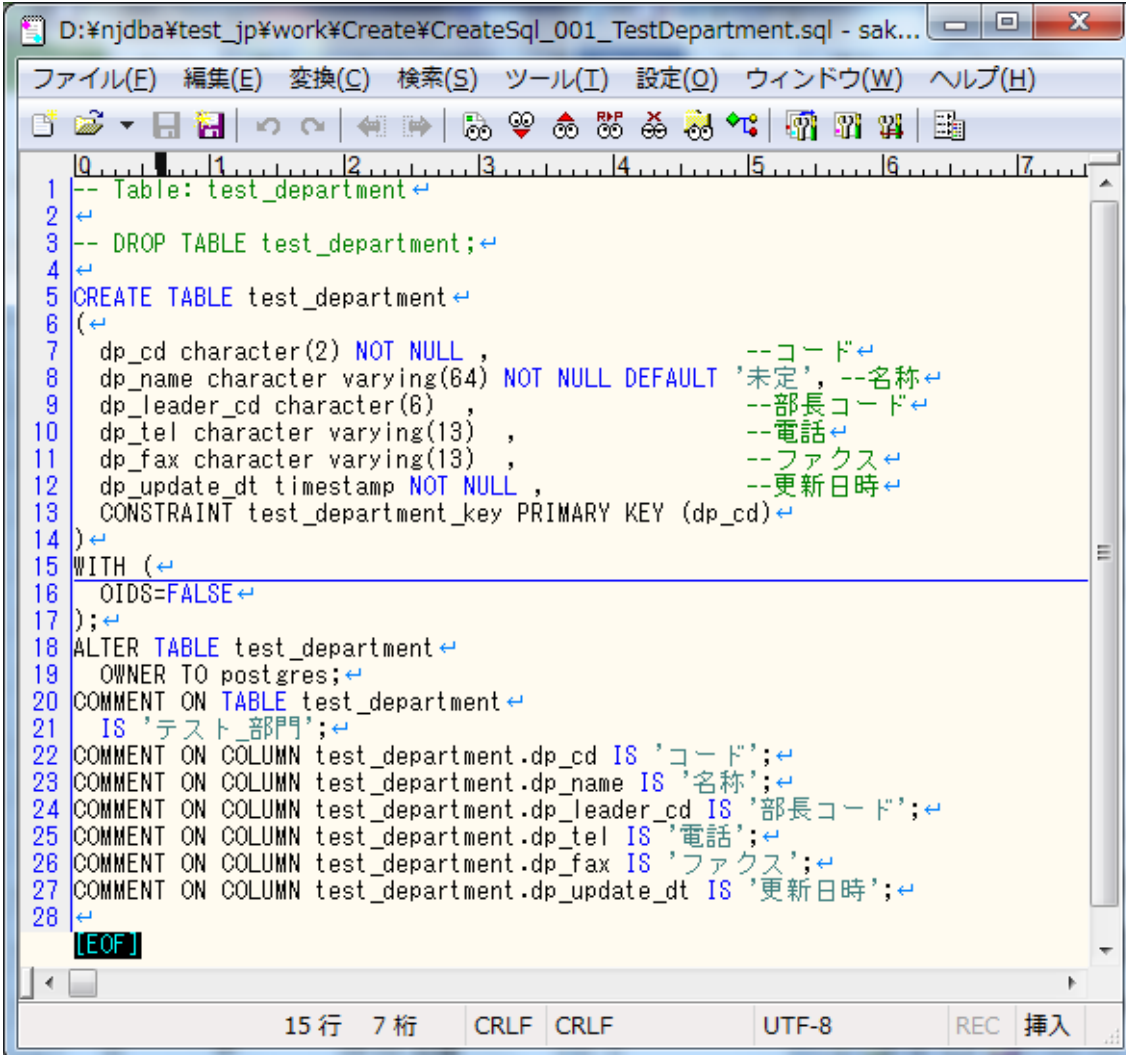


## 5. ツールの機能操作

### 5. 1 SQL 文の作成

1) 「CreateSql」ボタンには下辺の一覧テーブルに選択されたテーブルの Create Table の DDL を作成し、「ワークフォルダ¥Create」に出力されます。設定の間違いなどによって、エラー発生する場合は、エラー情報を「ワークフォルダ¥logs」に出力され、エラー情報はエラーの解明にとって、有力な手掛かりです。

下記は作成した PostgreSQL のサンプルです。(図 5 - 1)



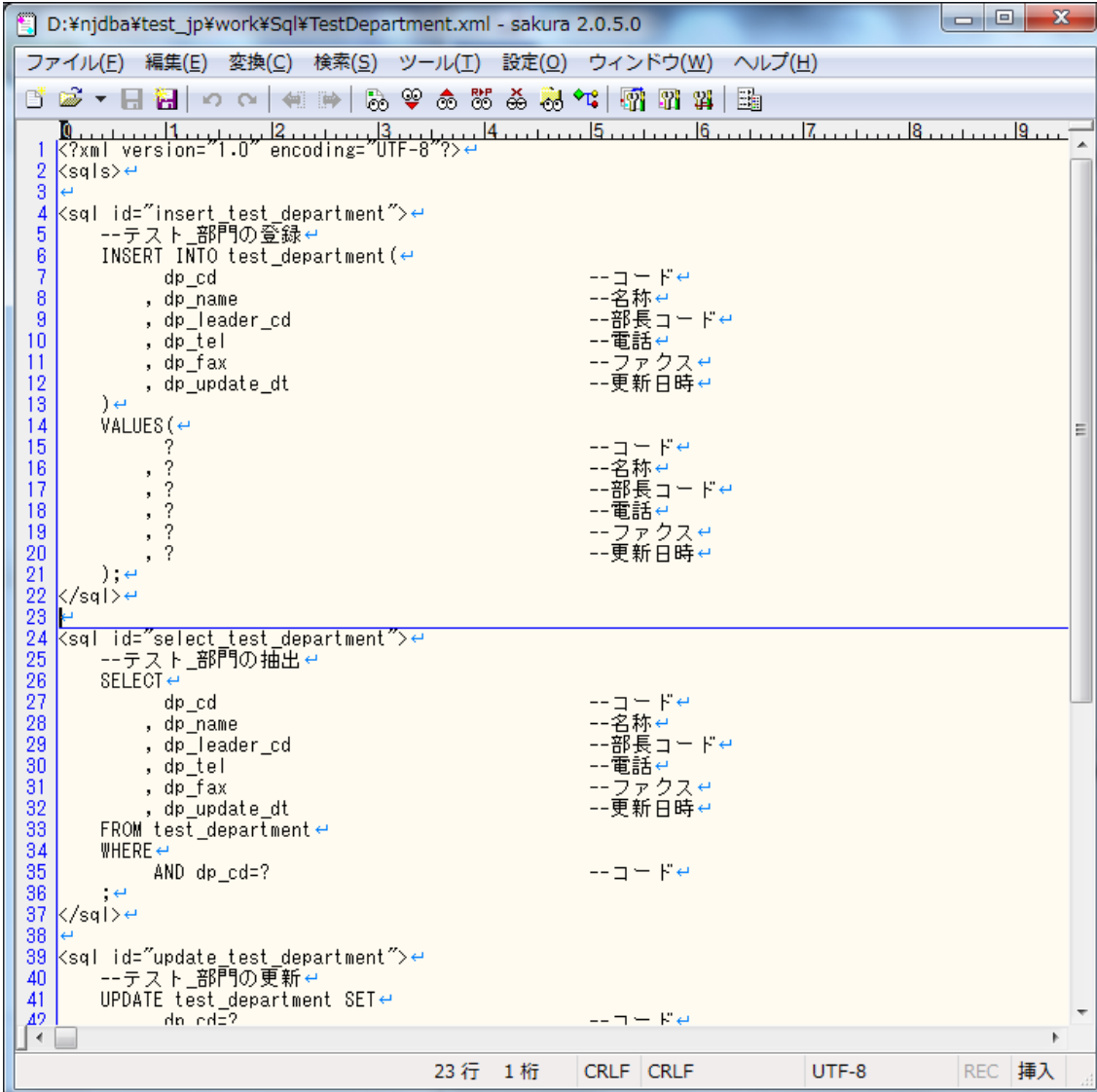
```
1  -- Table: test_department↵
2  ↵
3  -- DROP TABLE test_department;↵
4  ↵
5  CREATE TABLE test_department↵
6  (↵
7      dp_cd character(2) NOT NULL ,           --コード↵
8      dp_name character varying(64) NOT NULL DEFAULT '未定', --名称↵
9      dp_leader_cd character(6) ,             --部長コード↵
10     dp_tel character varying(13) ,          --電話↵
11     dp_fax character varying(13) ,          --ファクス↵
12     dp_update_dt timestamp NOT NULL ,       --更新日時↵
13     CONSTRAINT test_department_key PRIMARY KEY (dp_cd)↵
14 )↵
15 WITH (↵
16     OIDS=FALSE↵
17 );↵
18 ALTER TABLE test_department↵
19     OWNER TO postgres;↵
20 COMMENT ON TABLE test_department↵
21     IS 'テスト_部門';↵
22 COMMENT ON COLUMN test_department.dp_cd IS 'コード';↵
23 COMMENT ON COLUMN test_department.dp_name IS '名称';↵
24 COMMENT ON COLUMN test_department.dp_leader_cd IS '部長コード';↵
25 COMMENT ON COLUMN test_department.dp_tel IS '電話';↵
26 COMMENT ON COLUMN test_department.dp_fax IS 'ファクス';↵
27 COMMENT ON COLUMN test_department.dp_update_dt IS '更新日時';↵
28 ↵
[EOF]
```

2) 「CRUDSql」ボタンには下辺の一覧テーブルに選択されたテーブルの Insert、Select、Update、Delete の SQL 文を作成し、「ワークフォルダ¥Sql」に出力されます。

- ・下記は作成した「nj-soft」方式のサンプルです。(図5-2)

```
<?xml version="1.0" encoding="UTF-8"?>
<sqls>
  <sql id="insert_test_department" parameterClass="nj.entity.table.TestDepartment">
    --テスト_部門の登録
    INSERT INTO test_department(
      dp_cd          --コード
      , dp_name       --名称
      , dp_leader_cd  --部長コード
      , dp_tel        --電話
      , dp_fax        --ファクス
      , dp_update_dt  --更新日時
    )
    VALUES(
      #dp_cd#        --コード
      , #dp_name#     --名称
      , #dp_leader_cd# --部長コード
      , #dp_tel#      --電話
      , #dp_fax#      --ファクス
      , #dp_update_dt# --更新日時
    );
  </sql>
  <sql id="select_test_department" parameterClass="nj.entity.table.TestDepartment" resultClass="n
    --テスト_部門の抽出
    SELECT
      dp_cd          --コード
      , dp_name       --名称
      , dp_leader_cd  --部長コード
      , dp_tel        --電話
      , dp_fax        --ファクス
      , dp_update_dt  --更新日時
    FROM test_department
    WHERE
      1=1
      <if test="#dp_cd# != null and #dp_cd# != ''">
        AND dp_cd=#dp_cd# --コード
      </if>
    ;
  </sql>
</sqls>
```

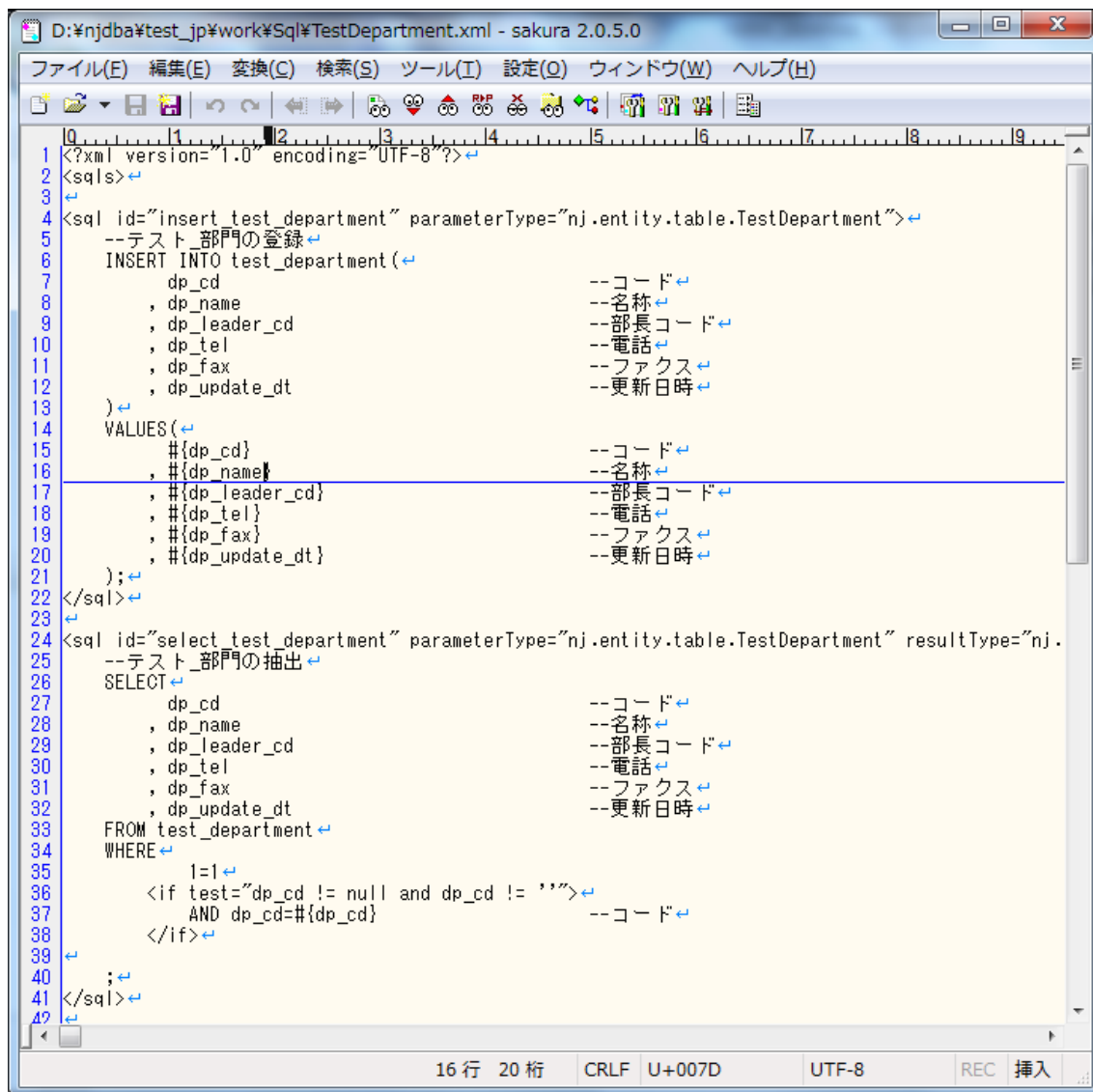
・下記は作成した「Prestatement」方式のサンプルです。(図 5 - 3)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <sqls>
3
4 <sql id="insert test_department">
5   --テスト_部門の登録
6   INSERT INTO test_department(
7     dp_cd           --コード
8     , dp_name       --名称
9     , dp_leader_cd  --部長コード
10    , dp_tel        --電話
11    , dp_fax        --ファクス
12    , dp_update_dt  --更新日時
13  )
14  VALUES(
15    ?               --コード
16    , ?             --名称
17    , ?             --部長コード
18    , ?             --電話
19    , ?             --ファクス
20    , ?             --更新日時
21  );
22 </sql>
23
24 <sql id="select test_department">
25   --テスト_部門の抽出
26   SELECT
27     dp_cd           --コード
28     , dp_name       --名称
29     , dp_leader_cd  --部長コード
30     , dp_tel        --電話
31     , dp_fax        --ファクス
32     , dp_update_dt  --更新日時
33   FROM test_department
34   WHERE
35     AND dp_cd=?     --コード
36   ;
37 </sql>
38
39 <sql id="update test_department">
40   --テスト_部門の更新
41   UPDATE test_department SET
42     dp_cd=?         --コード
```

23 行 1 桁 CRLF CRLF UTF-8 REC 挿入

- ・下記は作成した「MyBatis」方式のサンプルです。(図 5-4)

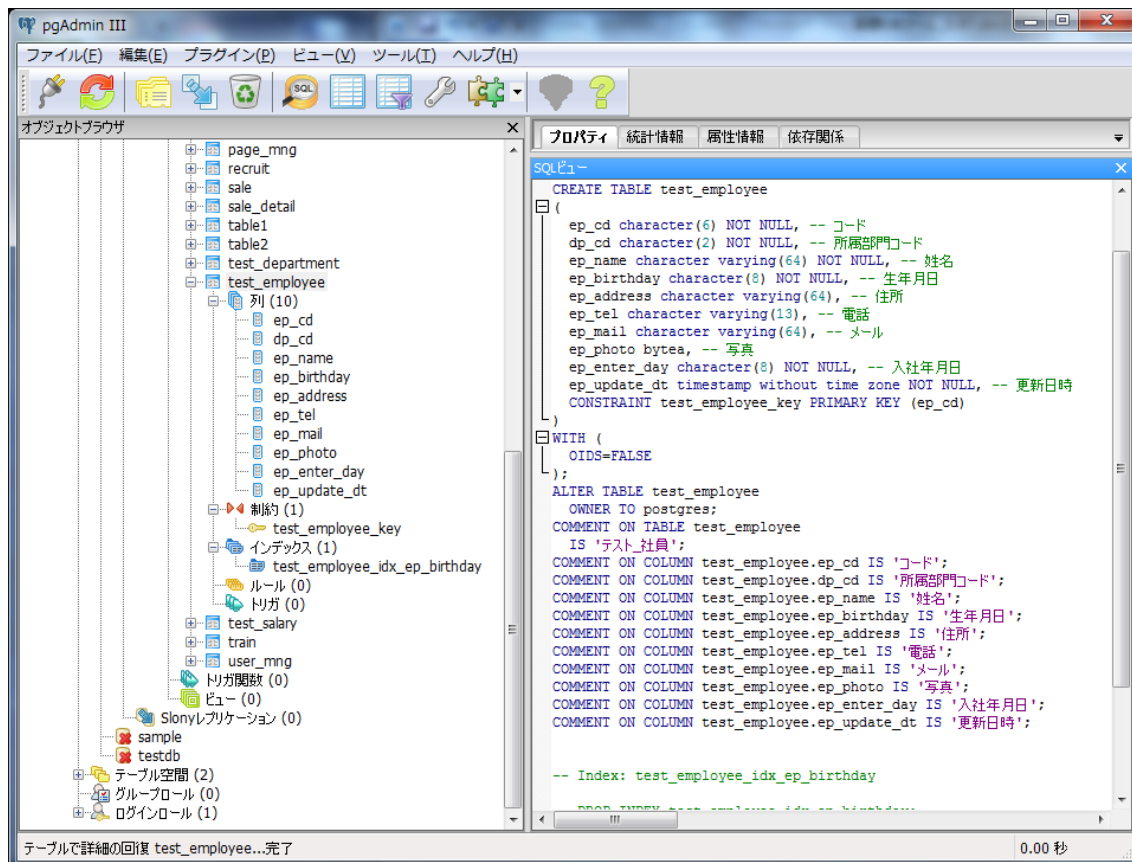


## 5. 2 テーブル及びデータの操作

1) 「CreateTable」ボタンには下辺の一覧テーブルに選択されたテーブルの 5. 1 の「CreateSql」で作成した DDL を利用して、データベースに物理テーブルを作成します。5. 1 の「CreateSql」で作成した DDL を手で修正してから実行も可能です。

※テーブル定義書を修正した場合、修正内容を反映するために、「CreateTable」実行する前に、「CreateSql」の再度実行が必要です。

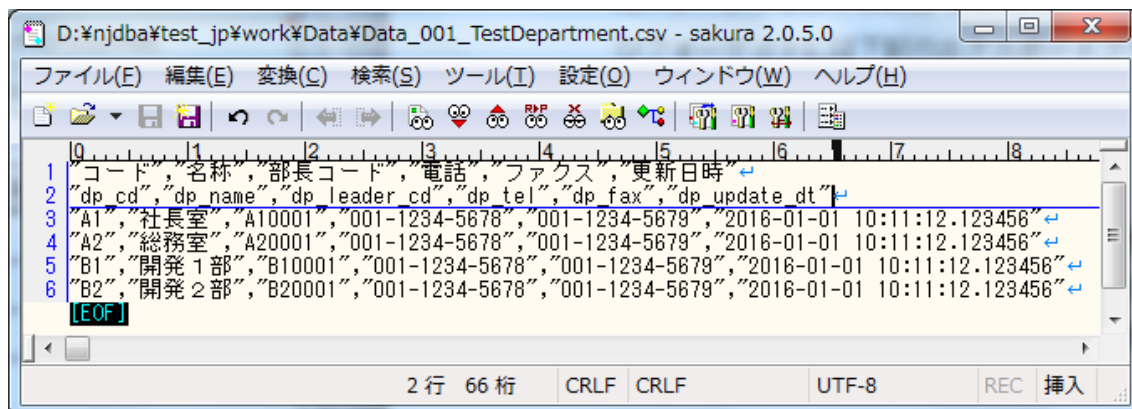
- ・下記は作成した「PostgreSQL」のテーブルです。(図 5 - 5)

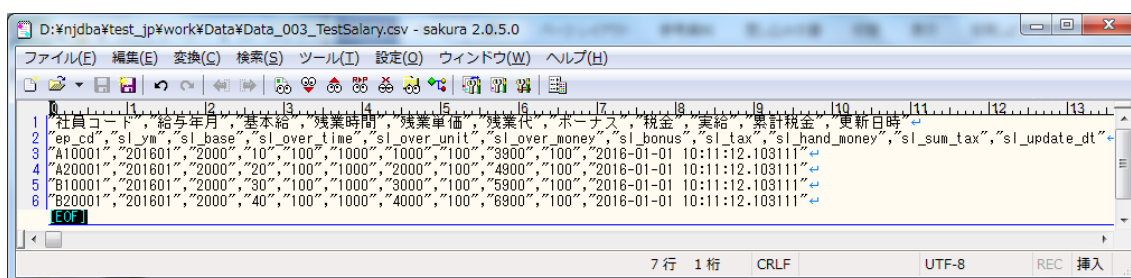
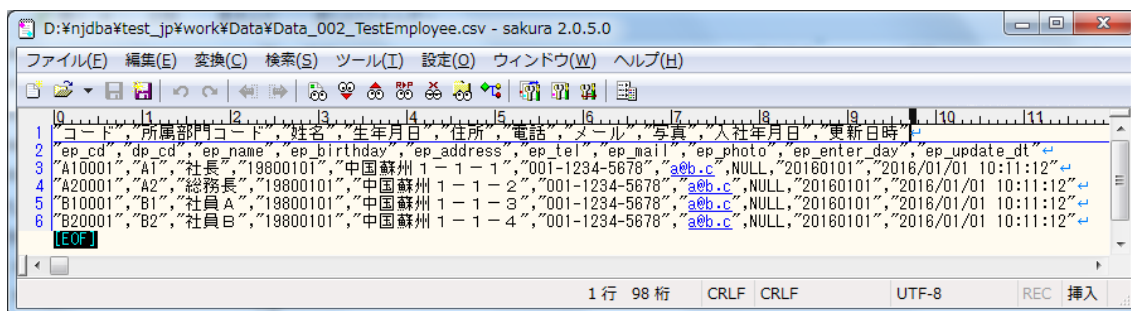


2) 「DropTable」ボタンには下辺の一覧テーブルに選択されたテーブルをデータベースから削除します。

3) 「ImportData」ボタンには下辺の一覧テーブルに選択されたテーブルに、事前に用意した「ワークフォルダ¥Data」にある CSV ファイルデータをインポートします。CSV ファイル名及びフォーマットには下記のエクスポートファイルと同じです。例)

- ・以下はインポートする CSV データ。(図 5 - 6)





・下記は DB にインポートされたデータ。(図 5-7)

編集データ - PostgreSQL 9.3 (localhost:5432) - postgres - test\_department

ファイル(E) 編集(E) ビュー(V) ツール(I) ヘルプ(H)

100 行

	dp_cd [PK] character(2)	dp_name character varying(64)	dp_leader_cd character(6)	dp_tel character varying(13)	dp_fax character varying(13)	dp_update_dt timestamp without time zone
1	A1	社長室	A10001	001-1234-5678	001-1234-5679	2016-01-01 10:11:12.123456
2	A2	総務室	A20001	001-1234-5678	001-1234-5679	2016-01-01 10:11:12.123456
3	B1	開発1部	B10001	001-1234-5678	001-1234-5679	2016-01-01 10:11:12.123456
4	B2	開発2部	B20001	001-1234-5678	001-1234-5679	2016-01-01 10:11:12.123456
*						

4 行

4) 「ExportData」ボタンには下辺の一覧テーブルに選択されたテーブルを、選択された「条件 1」、「(OR) 条件 2」、「(OR) 条件 3」の OR でデータをエクスポートします、「条件 1」、「(OR) 条件 2」、「(OR) 条件 3」の何れにも選択されていない場合は全てのデータをエクスポートします。エクスポートしたデータは「「ワークフォルダ¥Data」」に保存します。保存したファイルには 2 種類があり、UTF-8 文字コードのファイルはインポート、エクスポート用、Local 言語コードは Excel 開いて参照する用です。

※エクスポートされた CSV データはインポートと同じですから、例は省略します。

5) 「DeleteData」ボタンには下辺の一覧テーブルに選択されたテーブルに、選択された「条件1」、「(OR)条件2」、「(OR)条件3」のORでデータを削除します、「条件1」、「(OR)条件2」、「(OR)条件3」の何れにも選択されていない場合は全てのデータを削除します。

### 5. 3 CSV 及びDBデータの妥当性チェック

1) 「CheckCsv」ボタンには下辺の一覧テーブルに選択されたテーブルに対応する「ワークフォルダ¥Data」にある CSV ファイルデータの妥当性チェックを行い、チェック内容は下記です。

- ・テーブル定義書にキー項目の値必須チェック
- ・テーブル定義書に NULL 不可項目の NULL でないチェック
- ・テーブル定義書に文字型欄に設定された以下の文字コードチェック  
半角数字、半角数値、半角英数、半角文字、全角文字
- ・テーブル定義書に値の正規表現と合致するチェック
- ・テーブル定義書に追加チェック、追加チェックには自項目及び同じレコードの他項目との VBA の一行比較式で、チェックを行い、下記はチェック例です：

自項目の値は 2 桁です： `len("#this#")=2` '桁数チェック

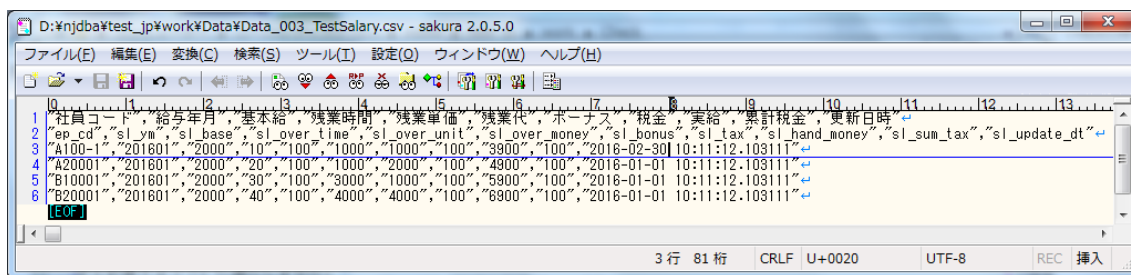
自項目の値は有効な日付です： `IsDate("#this#")` '日付チェック

二つ項目の値の比較です： `"#from_date#"<="#to_date#"` '日付比較

チェック結果は「ワークフォルダ¥Check」に出力されます。

例)

- ・下記は不正の CSV データ。(図 5 - 8)





- ・下記はテーブルフィールドの各種チェック。(図5-9)

No.	論理名	物理名	タイプ	キー	NULL	初期値	文字型	値の正規表現	追加チェック	索引	説明
1	社員コード	ep_cd	character(6)	○	○		半角英数	Yw+	len("#this#")=6 '桁数チェック		社員コード:6桁英数字
2	給与年月	sl_ym	character(6)	○	○		半角数字	(19 20)wd2(0 1-9)[0-2]			フォーマット:[yyyyMM]
3	基本給	sl_base	numeric(12,2)		○		半角数値		IsNumeric("#this#") '数字チェック	sl_base DESC,sl_bon us ASC	基本給:金額
4	残業時間	sl_Over_time	real		○	0	半角数値		IsNumeric("#this#") '数字チェック		残業時間:数字
5	残業単価	sl_Over_unit	integer		○	0	半角数値	Yd+			残業単価:金額
6	残業代	sl_Over_money	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") And #this# = #sl_Over_time# * #sl_Over_unit# '数 字と計算チェック		残業代:金額
7	ボーナス	sl_bonus	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") '数字チェック		ボーナス:金額
8	税金	sl_tax	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") '数字チェック		税金:金額
9	実給	sl_hand_money	numeric(12,2)		○		半角数値		IsNumeric("#this#") And #this#=#sl_base#+#sl_Over_money#+ #sl_bonus#+#sl_tax# '数字と実給 の算出チェック		実給:金額
10	累計税金	sl_sum_tax	numeric(12,2)		○	0	半角数値		IsNumeric("#this#") And #this#=#sl_tax# '数字と累計チェッ ク		累計税金:金額
11	更新日時	sl_update_dt	timestamp		○		半角文字		IsDate(left("#this#",19)) '日時 チェック		フォーマット:[yyyy-MM-dd hh:mm:ss.ffffff]

- ・下記はチェック結果です。(図5-10)

```

1 [テスト 給与]
2 001) データの3行1番目「社員コード」は文字型チェック「半角英数」に満たしません。
3 ["A100-1"]
4 002) データの3行1番目「社員コード」は正規表現「Yw+」に満たしません。
5 ["A100-1"]
6 003) データの3行11番目「更新日時」は追加チェック「IsDate(left("#this#",19)) '日時チェック」に満たしません。
7 [IsDate(left("2016-02-30 10:11:12.103111",19)) '日時チェック]
8 004) データの4行6番目「残業代」は追加チェック「IsNumeric("#this#") And #this# = #sl_Over_time# * #sl_Over_unit# '数字と計算チェック」に満たしません。
9 [IsNumeric("1000") And 1000 = 20 * 100 '数字と計算チェック]
10 005) データの6行9番目「実給」は追加チェック「IsNumeric("#this#") And #this#=#sl_base#+#sl_Over_money#+#sl_bonus#+#sl_tax# '数字と実給の算出チェック」に満たしません。
11 [IsNumeric("6900") And 6900=2000+4000+4000-100 '数字と実給の算出チェック]
[EOF]

```

2) 「CheckDb」ボタンには下辺の一覧テーブルに選択されたテーブルを、選択された「条件1」、「(OR)条件2」、「(OR)条件3」のORでデータを抽出してチェックします、チェック内容は上記1)と同じ、チェック結果も「ワークフォルダ¥Check」に出力されます。

例)



- ・下記はDB データ。(図 5 - 1 1)

	ep_cd [PK] chara	sl_ym [PK] chara	sl_base numeric(12)	sl_over_time real	sl_over_money integer	sl_bonus numeric(12,2)	sl_tax numeric(12,2)	sl_hand_mo numeric(12,2)	sl_sum_tax numeric(12,2)	sl_update_dt timestamp without time zone
1	A100-1	201601	2000.00	10	100	1000.00	1000.00	100.00	3900.00	2016-02-29 10:11:12.103111
2	A20001	201601	2000.00	20	100	1000.00	2000.00	100.00	4900.00	2016-01-01 10:11:12.103111
3	B10001	201601	2000.00	30	100	3000.00	1000.00	100.00	5900.00	2016-01-01 10:11:12.103111
4	B20001	201601	2000.00	40	100	4000.00	4000.00	100.00	6900.00	2016-01-01 10:11:12.103111

- ・下記選定されたデータ条件。(図 5 - 1 2)

寧静DBツール Ver1.00 蘇州寧静軟件有限公司 所有 <http://www.nj-soft.com> (最新バージョンダウンロード)

ワークフォルダ: D:\njdba\test\_jp\work\ DB接続ODBC: DSN=postgres 所有者(スキーマ): postgres

DB種類: PostgreSQL MySQL SqlServer Oracle DB2 Java/JKパッケージ: njentity.table

作成するSQL方式: nj-soft Prestatement MyBatis Other

1) SQL文の作成 CreateSql CRUDSql

2) テーブル及びデータの操作 CreateTable DropTable ImportData ExportData DeleteData

3) CSV及びDBデータの妥当性チェック CheckCsv CheckDb

4) Javaエンティティクラスの作成 CreateEntity

5) 下記テーブル定義の設定 AddFromFile AddLine DeleteLastLine DeleteAllLine

※下記テーブルの選択は全ての機能に適用。条件1〜3は「ExportData」、「CheckDb」に適用されます。

No.	全て取消	テーブル物理名	全て選択	条件1	全て選択	(OR)条件2	全て選択	(OR)条件3	テ...
1	<input type="checkbox"/>	テスト.部門	test_department	<input type="checkbox"/>	dp_cd=' '	<input type="checkbox"/>	dp_cd='AB'	<input type="checkbox"/>	dp_cd='A'
2	<input type="checkbox"/>	テスト.社員	test_employee	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
3	<input checked="" type="checkbox"/>	テスト.給与	test_salary	<input checked="" type="checkbox"/>	ep_cd='A100-1'	<input checked="" type="checkbox"/>	ep_cd='B20001'	<input type="checkbox"/>	

- ・下記はチェック結果。(図 5 - 1 3)

```

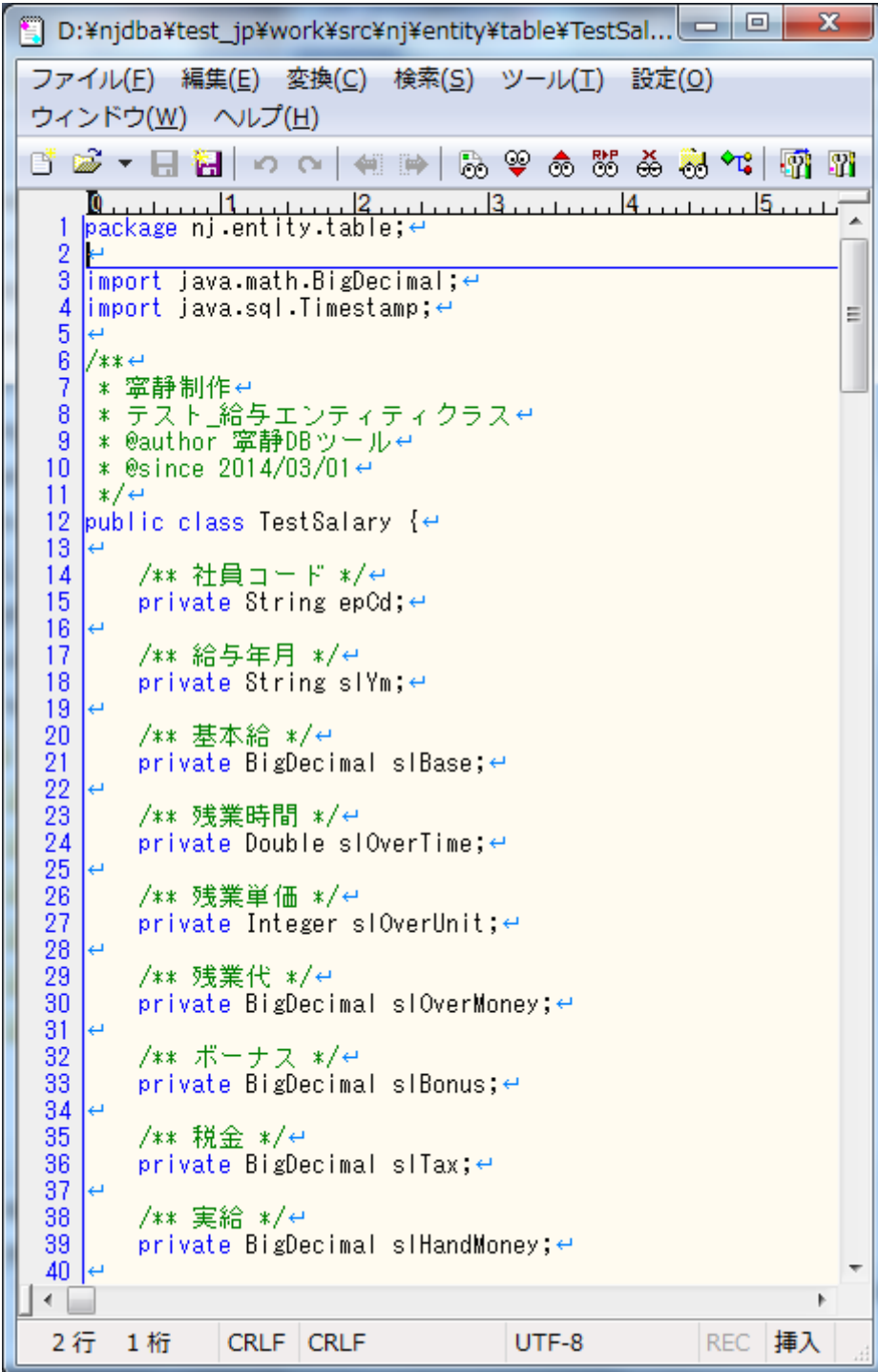
1 [テスト.給与]
2 001) キーの「ep_cd = [A100-1], sl_ym = [201601]」のレコードの項目「社員コード」は文字型チェック「半角英数」に満た
3 ません。
4 [A100-1]
5 002) キーの「ep_cd = [A100-1], sl_ym = [201601]」のレコードの項目「社員コード」は正規表現「^w+」に満たしません。
6 [A100-1]
7 003) キーの「ep_cd = [B20001], sl_ym = [201601]」のレコードの項目「実給」は追加チェック「IsNumeric("##this##") And #
his##=sl_base##sl_over_money##sl_bonus##sl_tax## "数字と実給の算出チェック」に満たしません。
8 [IsNumeric("6900") And 6900=2000+4000+4000-100 "数字と実給の算出チェック]
9 [EOF]

```

#### 5. 4 Java エンティティクラスの作成

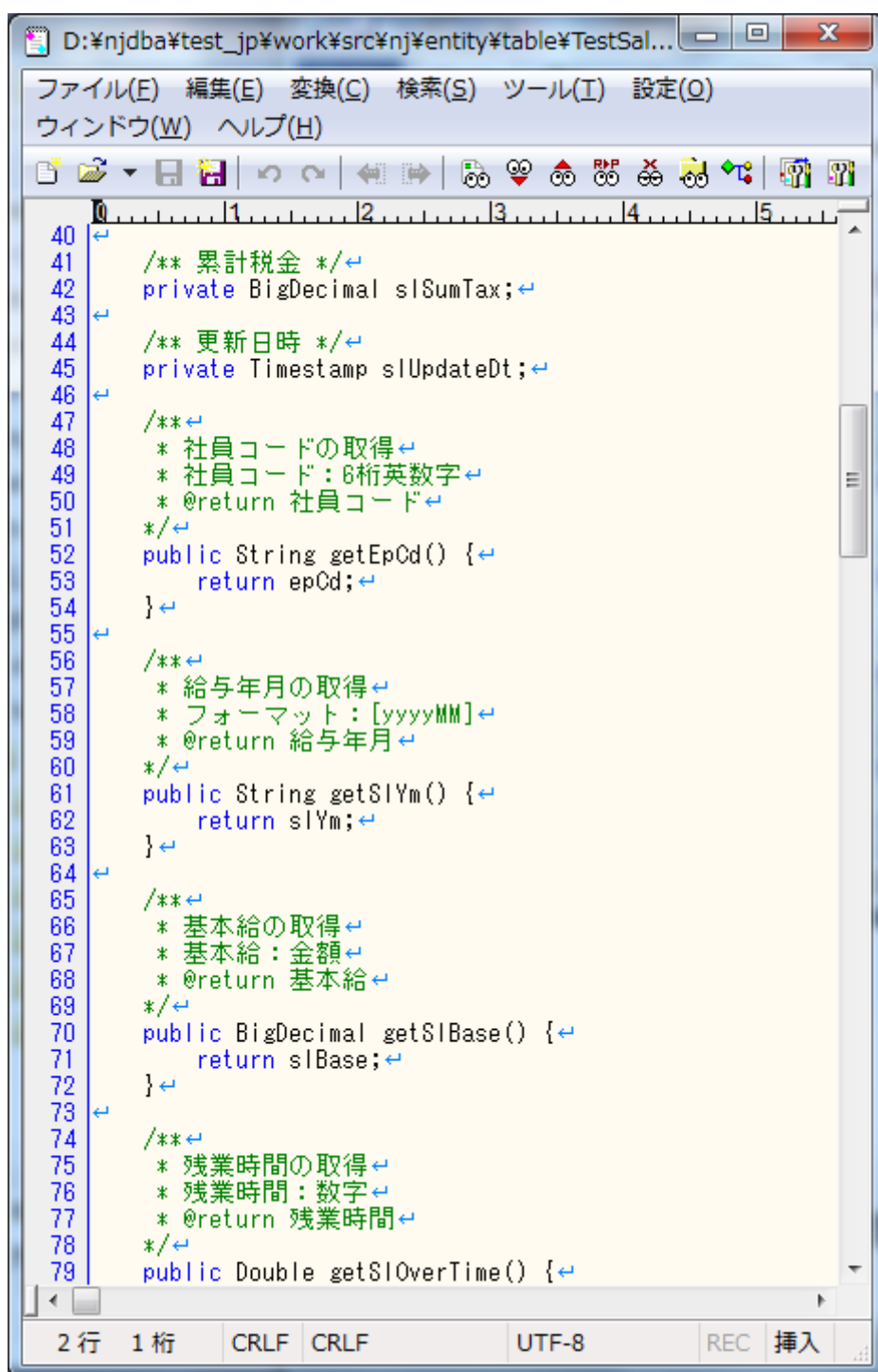
「CreateEntity」ボタンには下辺の一覧テーブルに選択されたテーブルを、プロジェクトに  
使われる Java エンティティクラスを作成します。作成したファイルは「ワークフォルダ¥src」  
に出力されます。作成するエンティティクラスのテンプレートにはご自身の好みで修正可能です、  
修正には「テンプレート」シートの「3)Java エンティティクラス」をご参照ください。

例) 下記は作成した Java エンティティクラスです。(図 5-14)



```
1 package nj.entity.table;
2
3 import java.math.BigDecimal;
4 import java.sql.Timestamp;
5
6 /**
7  * 寧靜制作
8  * テスト_給与エンティティクラス
9  * @author 寧靜DBツール
10  * @since 2014/03/01
11  */
12 public class TestSalary {
13
14     /** 社員コード */
15     private String epCd;
16
17     /** 給与年月 */
18     private String slYm;
19
20     /** 基本給 */
21     private BigDecimal slBase;
22
23     /** 残業時間 */
24     private Double slOverTime;
25
26     /** 残業単価 */
27     private Integer slOverUnit;
28
29     /** 残業代 */
30     private BigDecimal slOverMoney;
31
32     /** ボーナス */
33     private BigDecimal slBonus;
34
35     /** 税金 */
36     private BigDecimal slTax;
37
38     /** 実給 */
39     private BigDecimal slHandMoney;
40 }
```

2行 1桁 CRLF CRLF UTF-8 REC 挿入

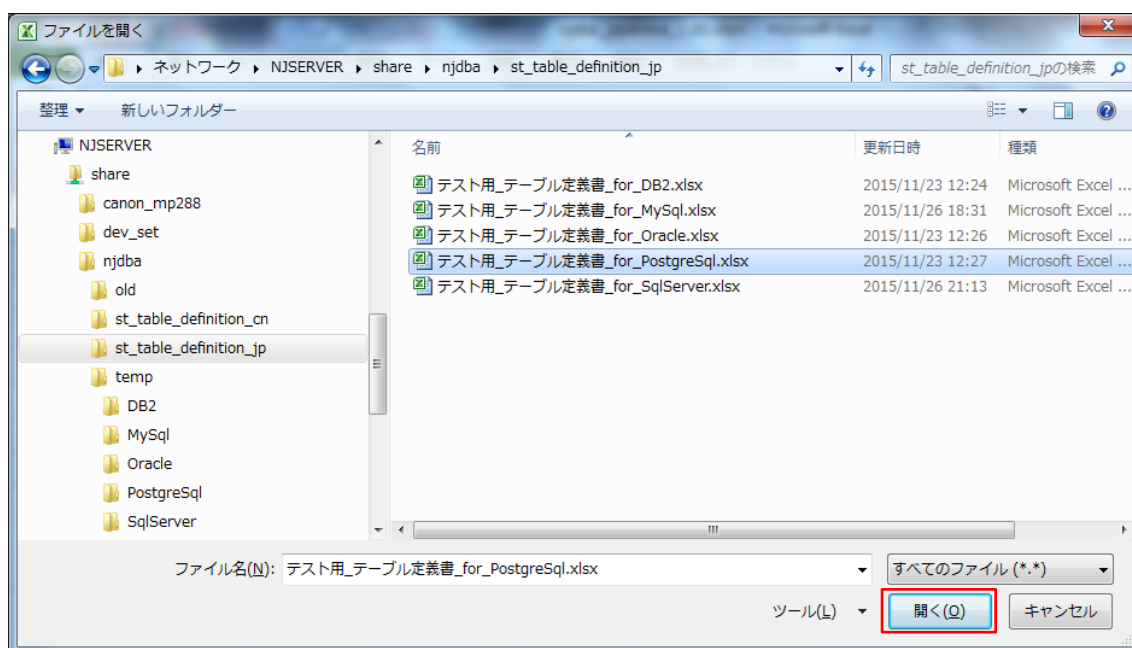
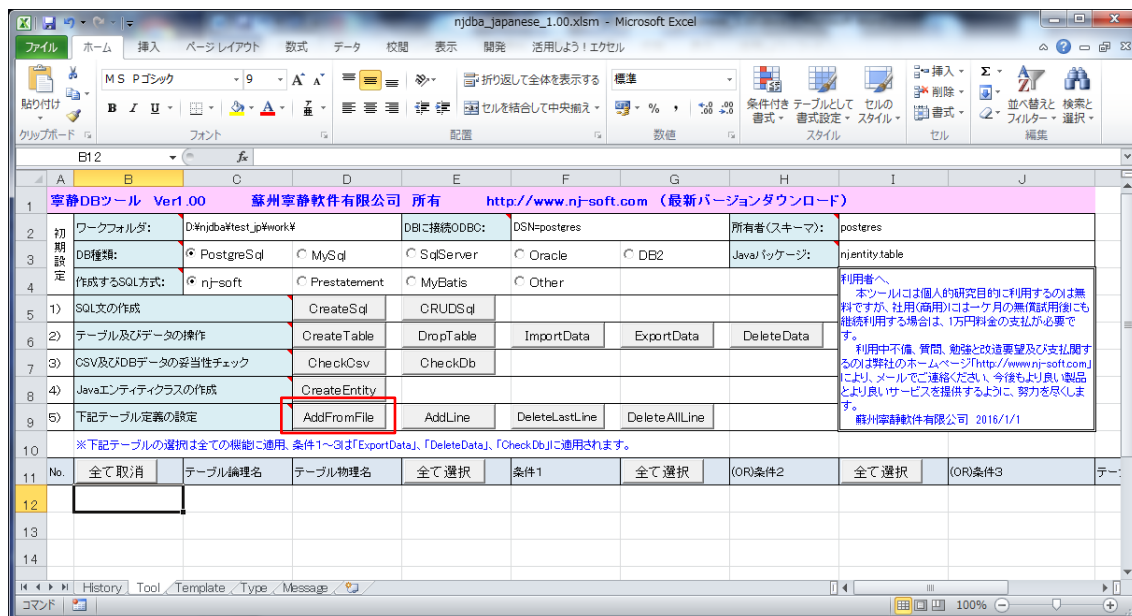


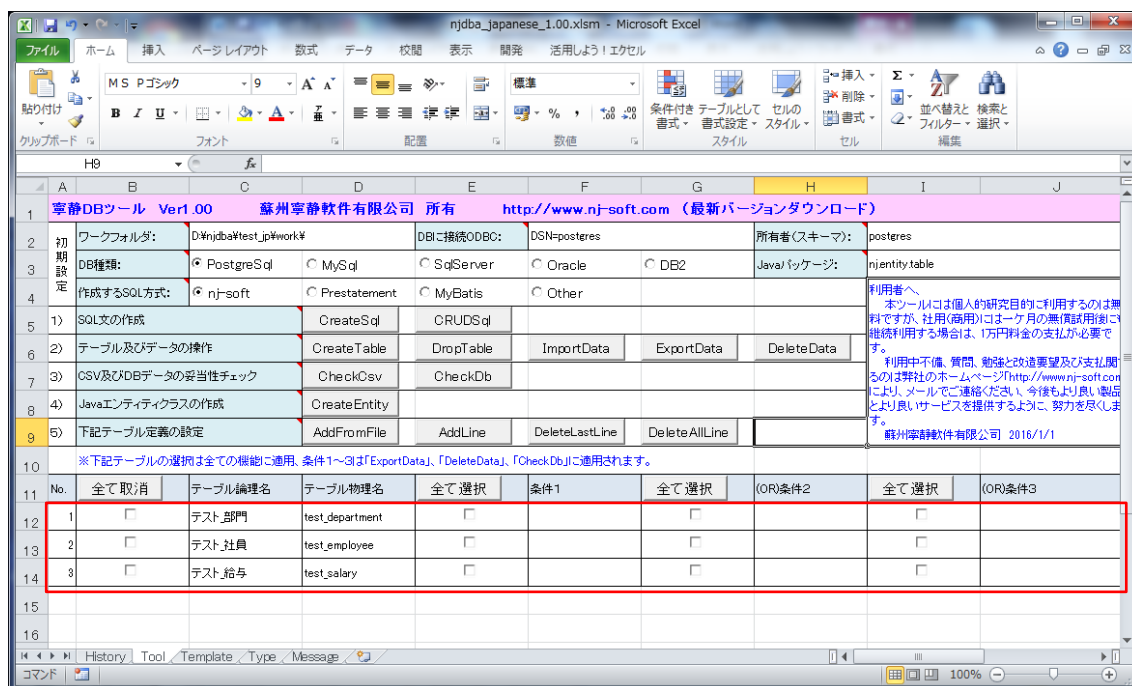
### 5. 5 下記テーブル定義の設定

本機能は下辺の一覧テーブルの設定を行います。

1) 「AddFromFile」ボタンには、入力するテーブル定義書ファイルに全てのテーブルを下辺の一覧に追加します。入力したファイルにそれぞれのシートがテーブル定義かどうかの判定基準はシート名と該当シートにテーブル論理名セル内容が一致することです。

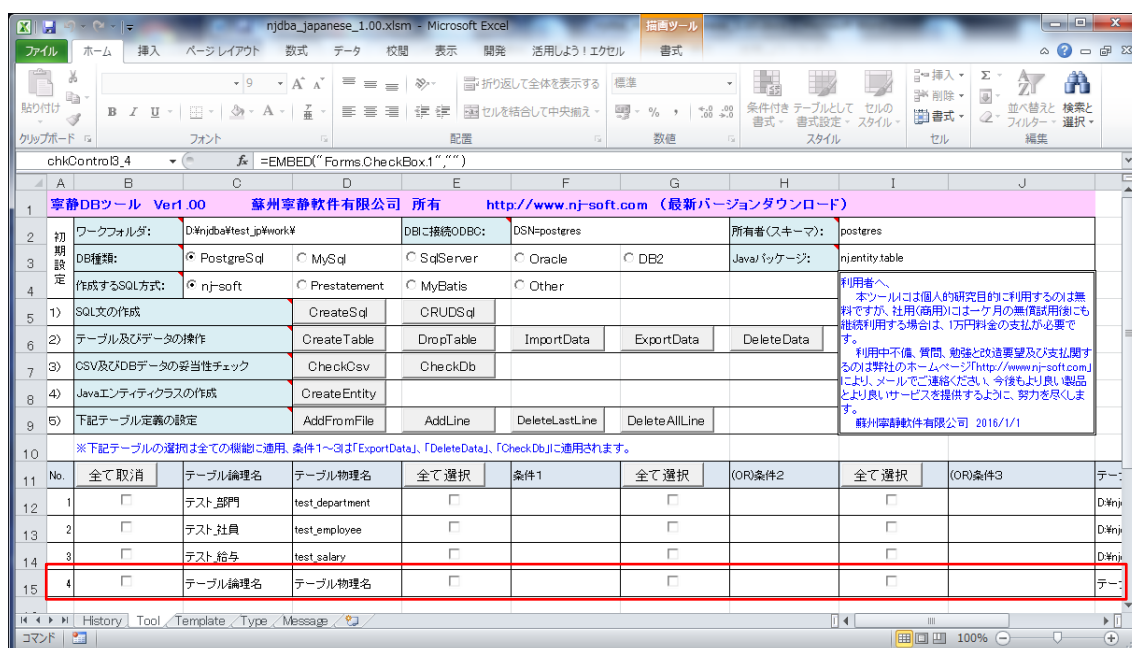
例) 下記は「AddFromFile」をクリックしてからの操作です。(図5-15)





2) 「AddLine」ボタンには、下辺の一覧テーブルの最後に一行が追加されます。但し、追加の後、「テーブル論理名」、「テーブル物理名」、「テーブル仕様書」列の内容を実テーブル定義書の内容と一致するように修正してください。

例) 下記は「AddLine」ボタンをクリックしてから、最後の一行が追加されました。(図5-16)



3) 「DeleteLastLine」ボタンには、下辺の一覧テーブルの最後の一行が削除されます。

4) 「DeleteAllLine」ボタンには、下辺の一覧テーブルに全ての行が削除されます。

## 5. 6 全て選択・全て取消

1) 下辺の一覧テーブルに Excel の B 列の「全て選択・全て取消」をクリックする場合、全てのテーブルが選択されるか選択されないことになります。この列の選択には上記 5. 1 ～ 5. 5 の機能に全部適用されます。

2) 下辺の一覧テーブルに Excel の E 列の「全て選択・全て取消」をクリックする場合、全ての条件 1 が選択されるか選択されないことになります。この列の選択には上記の「ExportData」、「DeleteData」、「CheckDb」機能に適用されます。

3) 下辺の一覧テーブルに Excel の G 列の「全て選択・全て取消」をクリックする場合、全ての条件 2 が選択されるか選択されないことになります。この列の選択には上記の「ExportData」、「DeleteData」、「CheckDb」機能に適用されます。

4) 下辺の一覧テーブルに Excel の I 列の「全て選択・全て取消」をクリックする場合、全ての条件 3 が選択されるか選択されないことになります。この列の選択には上記の「ExportData」、「DeleteData」、「CheckDb」機能に適用されます。

## 6. その他

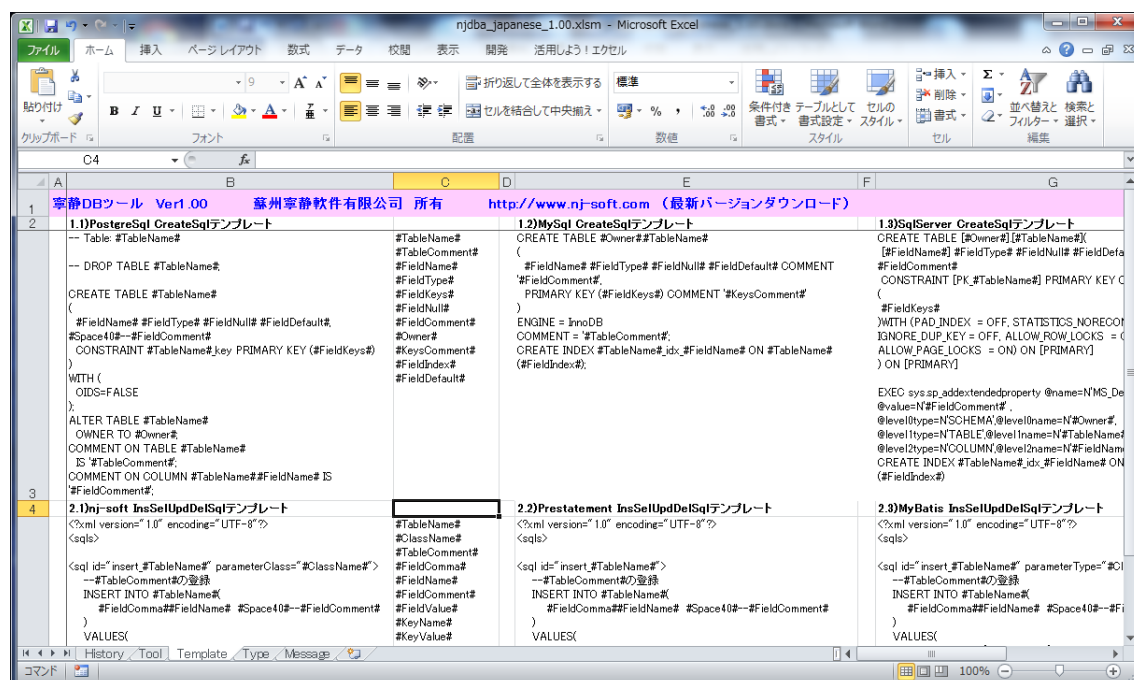
1) 本ツールの「History」シートには Ver1.00 以降の変更履歴です。(図 6-1)

バージョン	契機	シート	変更内容	変更者	変更日	承認者	承認日
1.00			新規作成	顧	2016/1/1	顧	2016/1/1

2) 本ツールの「Template」シートには各種機能のテンプレートです、ご自身の好みに適当に修正も可能です。

- ・ 1.1) PostgreSQL CreateSql テンプレート
- ・ 1.2) MySQL CreateSql テンプレート
- ・ 1.3) SqlServer CreateSql テンプレート
- ・ 1.4) Oracle CreateSql テンプレート
- ・ 1.5) DB2 CreateSql テンプレート
- ・ 2.1) nj-soft InsSelUpdDelSql テンプレート
- ・ 2.2) Prestatement InsSelUpdDelSql テンプレート
- ・ 2.3) MyBatis InsSelUpdDelSql テンプレート
- ・ 2.4) Other テンプレート
- ・ 3) Java エンティティクラス
- ・ 4.1) PostgreSQL テーブル名取得テンプレート
- ・ 4.2) MySQL テーブル名取得テンプレート
- ・ 4.3) SqlServer テーブル名取得テンプレート
- ・ 4.4) Oracle テーブル名取得テンプレート
- ・ 4.5) DB2 テーブル名取得テンプレート

(図 6 - 2)



3) 本ツールの「Type」シートには各種データベースのデータタイプと Java 変数のデータタイプの対照です。この対照を適当に変更すれば、Java エンティティクラスの項目のデータタイプが変更されます、それと「Java インプット」列に入力すれば、Java エンティティクラスにこの入力情報を import されます。

- ・ 1)PostgreSql のタイプと Java のタイプ
- ・ 2)MySql のタイプと Java のタイプ
- ・ 3)SqlServer のタイプと Java のタイプ
- ・ 4)Oracle のタイプと Java のタイプ
- ・ 5)DB2 のタイプと Java のタイプ



(図 6 - 3)

寧静DBツール Ver1.00 蘇州寧静軟件有限公司 所有 http://www.nj-soft.com (最新バージョンダウンロード)							
1) PostgreSQLのタイプとJavaのタイプ			2) MySQLのタイプとJavaのタイプ			3) SQLServerのタイプとJavaのタイプ	
No.	PostgreSQLのタイプの正規式	Java変数のタイプ	Javaインプット	MySQLのタイプの正規式	Java変数のタイプ	Javaインプット	SQLServerのタイプの正規式
1	bigint	Long		int	Integer		bigint
2	bigserial	Long		varchar	String		binary
3	bit	byte[]		decimal	BigDecimal	java.math.BigDecimal	bit
4	boolean	Boolean		datetime	Timestamp	java.sql.Timestamp	char
5	box	Box		blob	byte[]		date
6	bytea	byte[]		binary	byte[]		datetime
7	character	String		longblob	byte[]		datetime2
8	cidr	String		mediumblob	byte[]		datetimeoffset
9	circle	Circle		tinyblob	byte[]		decimal
10	date	Date	java.sql.Date	varbinary	byte[]		float
11	double	Double		date	Date	java.sql.Date	geography
12	inet	String		time	Time	java.sql.Time	geometry
13	integer	Integer		timestamp	Timestamp	java.sql.Timestamp	hierarchyid
14	interval	String		year	Integer		image
15	line	Line2D	java.awt.geom.Line2D	geometry	Geometry	njentity.Geometry	int
16	lseg	Lseg	njentity.Lseg	geometrycollection	GeometryCollection	njentity.GeometryCollection	money
17	macaddr	Macaddr	njentity.Macaddr	linestring	Linearring	njentity.Linearring	nchar
18	money	BigDecimal	java.math.BigDecimal	multilinestring	MultiLinearring	njentity.MultiLinearring	ntext
19	numeric	BigDecimal	java.math.BigDecimal	multipoint	MultiPoint	njentity.MultiPoint	numeric

4) 本ツールの「Message」シートには本ツールの出力メッセージとログに使われています。

(図 6 - 4)

寧静DBツール Ver1.00 蘇州寧静軟件有限公司 所有 http://www.nj-soft.com (最新バージョンダウンロード)		
No.	メッセージID	メッセージ内容
1	csv_required	データの[0]行[1]番目[2]はキーがNULL不可のため、必須入力項目です。
2	db_required	キーの[0]のレコードの項目[1]はキーがNULL不可のため、必須入力項目です。
3	csv_regular	データの[0]行[1]番目[2]は正規表現[3]に満たしません。
4	db_regular	キーの[0]のレコードの項目[1]は正規表現[2]に満たしません。
5	csv_addcheck	データの[0]行[1]番目[2]は追加チェック[3]に満たしません。
6	db_addcheck	キーの[0]のレコードの項目[1]は追加チェック[2]に満たしません。
7	csv_model_check	データの[0]行[1]番目[2]は文字型チェック[3]に満たしません。
8	db_model_check	キーの[0]のレコードの項目[1]は文字型チェック[2]に満たしません。
9	log_1_1_start	<<=====1. 1)SQL文の作成 Create Sql Start=====
10	log_1_1_end	=====1. 1)SQL文の作成 Create Sql End=====
11	log_1_2_start	<<=====1. 2)SQL文の作成 Insert Sql Start=====
12	log_1_2_end	=====1. 2)SQL文の作成 Insert Sql End=====
13	log_2_1_start	<<=====2. 1)テーブル及びデータの操作 Create Table Start=====
14	log_2_1_end	=====2. 1)テーブル及びデータの操作 Create Table End=====
15	log_2_2_start	<<=====2. 2)テーブル及びデータの操作 Drop Table Start=====
16	log_2_2_end	=====2. 2)テーブル及びデータの操作 Drop Table End=====
17	log_2_3_start	<<=====2. 3)テーブル及びデータの操作 Import Data Start=====

(全文完了)

蘇州寧静軟件有限公司 所有

<http://www.nj-soft.com>

2016年1月1日 元旦