

**DTP SQL Development Tools: Database Table Manipulation component
Design Document
Version 0.1
10/26/2006
Takehiro Hosokawa**

Table of Contents

| | |
|--|---|
| 1.Introduction..... | 2 |
| 2.Requirements..... | 2 |
| 3.High-Level Design..... | 2 |
| 3.1.Background..... | 2 |
| 3.1.1.SQL Editor..... | 2 |
| 3.1.2.NEC Soft's JALCEDO..... | 2 |
| 3.2.Database Table Manipulation component..... | 3 |
| 3.2.1.The Interface..... | 3 |
| 3.2.2.Menu..... | 4 |
| 3.2.3.Create Table dialog..... | 5 |
| 3.2.4.Modify Table dialog..... | 6 |
| 3.2.4.1.Insert Column..... | 6 |
| 3.2.4.2.Drop Column..... | 7 |
| 3.2.5.Remove Table dialog..... | 7 |
| 3.3.Dependencies..... | 7 |
| 3.4.Tasks..... | 7 |
| 3.4.1.Add menus to the DSE..... | 8 |
| 3.4.2.Create menu actions..... | 8 |
| 3.4.3.Create dialogs..... | 8 |
| 3.4.4.Create SQL generator..... | 8 |
| 3.4.5.Create inputs for the SQL Editor..... | 9 |
| 3.4.6.Display results..... | 9 |
| 3.4.7.Execute results..... | 9 |
| 3.5.Additional Notes..... | 9 |
| 3.5.1.Extensibility Issues..... | 9 |

1. Introduction

The purpose for this document is to describe the high-level design of the Database Table Manipulation component for the DTP SQL Development Tools project.

2. Requirements

The Database Table Manipulation component will provide functionality for manipulation relational database tables. To accomplish this purpose, the Database Table Manipulation component should:

- 1) Build a specific set of SQL Data Definition Language (DDL) statements.
- 2) Supply a graphical user interface for 1).
- 3) Execute the result of 1).
- 4) Display the result of 1).

Manipulations this component supports are:

- a) Create a database table

The user creates a new table within a specific database schema. Column names, data types (domains), and constraints are specified as necessary.

- b) Modify a database table

The user modifies a specific table in a given database schema, performing actions such as adding/removing columns, changing column data types (domains), and changing constraints.

- c) Remove a database table

The user removes (drops) a specific table from a given database schema.

3. High-Level Design

3.1. Background

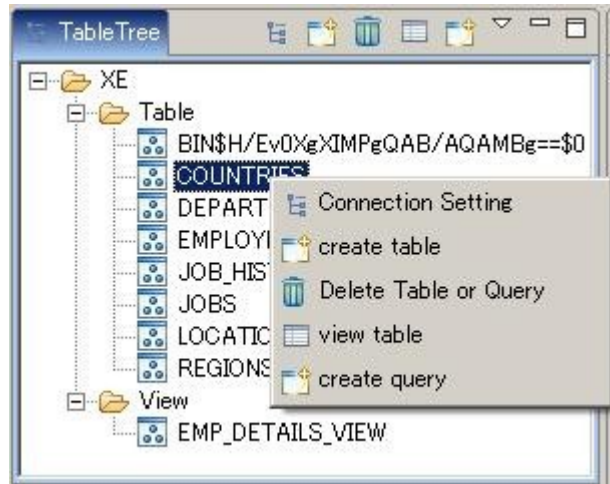
3.1.1. SQL Editor

The SQL Editor of the DTP SQL Development Tools project provides a tool for standard text-based editing of SQL statements and a framework for executing and displaying them.

3.1.2. NEC Soft's JALCEDO

The JALCEDO has functionality for manipulation database tables. As you can guess from the screen shot below, it supplies a graphical mechanism for creating and deleting tables.

The JALCEDO internally builds SQL statements for the specified manipulations such as creating/deleting/modifying tables. So It will make use of the SQL Editor for executing and displaying results.



Clicking create table, the dialog box is shown and you can create a new database table with specified columns.

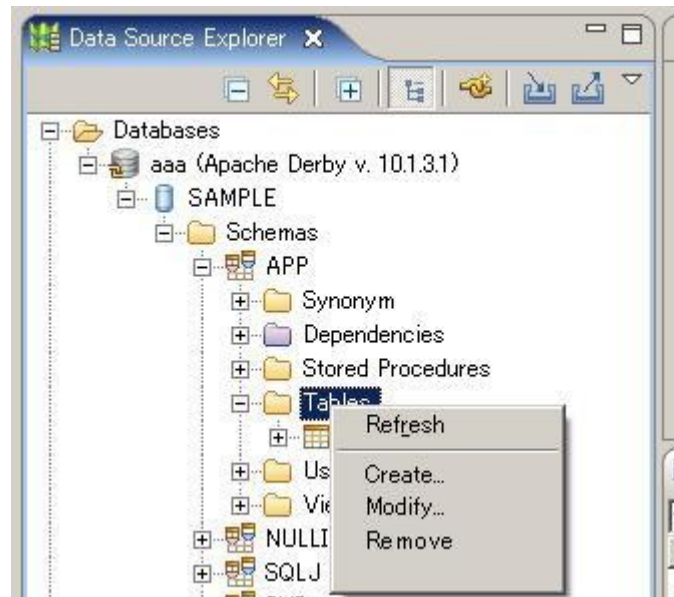
Clicking Delete Table or Query, you can delete the selected database table.

3.2. Database Table Manipulation component

The Database Table Manipulation component will provide the interface through which the user will be interacting with other component, SQL Editor, in the DTP SQL Development Tools project. It will provide database table manipulations such as creating, modifying and removing a database table.

3.2.1. The Interface

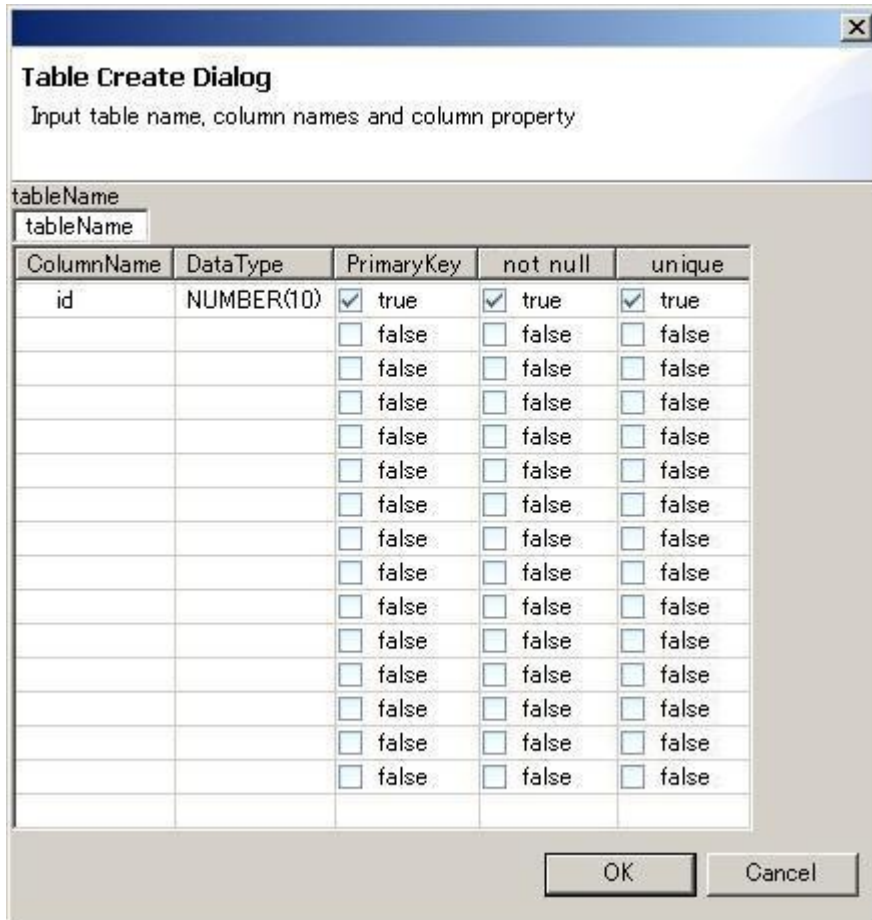
The Database Table Manipulation component will supply a graphical mechanism to build and execute a specific set of SQL Data Definition Language (DDL) statements. It will be available from the menu in the Data Source Explore (DSE). A couple of menus will be added to the menu that is shown as right clicking in the DSE. If Tables folder or tables in a Tables tree is selected, Create/Modify/Remove menus will be available.



3.2.2. Menu

| Menu | Action |
|--------|--|
| Create | Opens the Create Table dialog. |
| Modify | Opens the Modify Table dialog. |
| Remove | Opens the Remove Table dialog and removes the selected database table. |

3.2.3. Create Table dialog



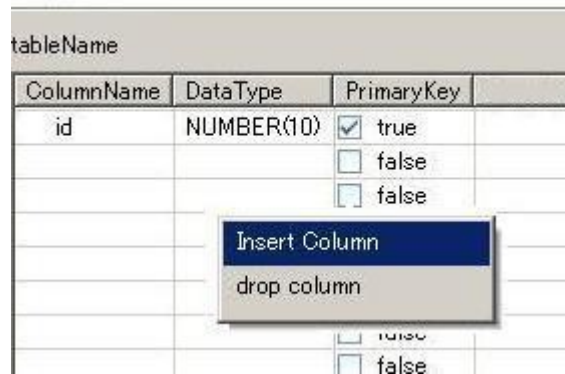
| Widget | Action |
|---|---|
| Table Name text box | Specifies a database table name to create. |
| Column Name cell | Specifies a column name to be added to the created database table. |
| Data Type cell | Specifies a data type for the column. If this cell is clicked, a combo box with list of selectable data type is shown. |
| Constraint cell (Primary key, not null, unique....) | Checks if the constraint is true. |
| OK button | If be clicked, build a create table SQL statement with specified setting in this dialog, and let the SQL statement be shown in the SQL Editor and execute it. |
| Cancel button | Cancel the operation and close the dialog. |

Note1: We will use the Database Definition (DBDefinition) associated with a given connection profile instance to get all the data types available for that specific database. DTP also have a generic DBDefinition for JDBC. Doing it this way will allow users to get data types for specific databases, as long as an appropriate DBDefinition is installed. (If there is not vendor-specific DBDefinition, then using the generic JDBC one is fine.)

Note2: Ideally, we will get information for constraints from the DBDefinition file. While this will not cover all possible constraints, it would be a start. We want to check with the DTP community for additional ideas.

3.2.4. Modify Table dialog

The Modify Table dialog will be almost similar with the Create Table dialog besides the adding/removing columns action. Right clicking at any point on a table, the menu will be shown that has Insert Column/Drop Column menu.

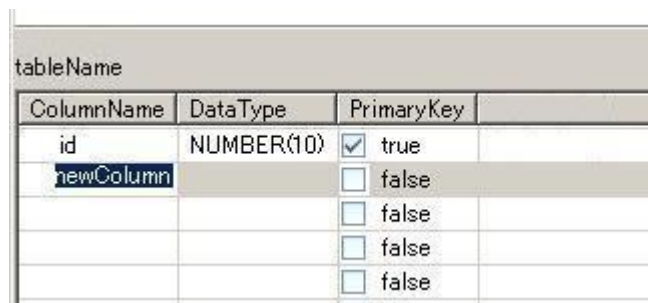


| Menu | Action |
|---------------|--|
| Insert Column | Inserts a new column below the existed column. |
| Drop Column | Drops the specified column. |

Note: Cells are disable besides the existed column's. To be available you have to click Insert Column.

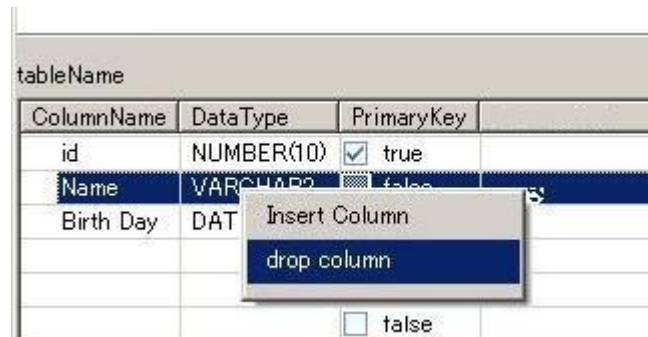
3.2.4.1. Insert Column

Clicking the Insert Column menu, cells below end of the existed column will be available to write data. In default “new Column” will be shown on the Column Name's cell and other cells are empty.



3.2.4.2. Drop Column

Clicking the Drop Column menu, the selected column will be dropped. If a column is removed from a table, then any data associated with it will be dropped.



3.2.5. Remove Table dialog

The Remove Table dialog will be just for confirmation, and just have OK and Cancel button. If OK is clicked, the specified database table is removed.

3.3. Dependencies

We will be dependent on the following components:

- Data Source Explore
- connection profiles (exposed in the Data Source Explore)
- SQL Editor
- SQL model
- components for executing SQL and getting results (maybe the SQL Dev team has)

3.4. Tasks

- 1) Add menus to the DSE.
 - a. Add Create/Modify/Remove menu to the DSE menu.
- 2) Create menu actions.
 - a. Create Create Table action (Create Table dialog).
 - b. Create Modify Table action (Modify Table dialog).
 - c. Create Remove Table action.
- 3) Create dialogs.
 - a. Create Create Table dialog.
 - b. Create Modify Table dialog.
- 4) Create SQL generator.
 - a. Generate SQL DDL statements for the specified menu actions.
- 5) Create inputs for the SQL Editor.
 - a. Create database properties for the SQL Editor.
 - b. Create SQL statements for the SQL Editor.
 - c. Set these results to the SQL Editor.

DTP SQL Development Tools: Database Table Manipulation component Design

- 6) Display results.
 - a. Call the SQL Editor to display SQL statements generated in SQL generator.
- 7) Execute results.
 - a. Call the SQL Editor to execute SQL statements generated in SQL generator.

3.4.1. Add menus to the DSE.

| Part | Dependency | Risks | Time to Complete (min/max) |
|--|------------|-------|----------------------------|
| Add Create/Modify/Remove menu to the DSE menu. | DSE | N/A | 0.5 days/1 days |

3.4.2. Create menu actions.

| Part | Dependency | Risks | Time to Complete (min/max) |
|---|------------|-------|----------------------------|
| Create Create Table action (Create Table dialog). | None | N/A | 0.5 day/1 day |
| Create Modify Table action (Modify Table dialog). | None | N/A | 0.5 day/1 day |
| Create Remove Table action. | None | N/A | 0.5 day/ 1day |

3.4.3. Create dialogs.

| Part | Dependency | Risks | Time to Complete (min/max) |
|-----------------------------|-------------------|---------------------------|----------------------------|
| Create Create Table dialog. | DTP Connectivity? | Dialog design Constraints | 2 days/5 days |
| Create Modify Table dialog. | DTP Connectivity? | Dialog design Constraints | 1 day/ 3days |

3.4.4. Create SQL generator.

| Part | Dependency | Risks | Time to Complete (min/max) |
|---|------------|------------------------|----------------------------|
| Generate SQL DDL statements for the specified menu actions. | ? | Compatibility with DTP | 3 days/5 days |

3.4.5. Create inputs for the SQL Editor.

| Part | Dependency | Risks | Time to Complete (min/max) |
|--|----------------------------|-------|----------------------------|
| Create database properties for the SQL Editor. | DTP Connectivity SQL tools | N/A | 1 days/2 days |
| Create SQL statements for the SQL Editor. | SQL Editor | N/A | 1 days/2 days |
| Set these results to the SQL Editor. | SQL Editor | N/A | 0.5 days/1 days |

3.4.6. Display results.

| Part | Dependency | Risks | Time to Complete (min/max) |
|---|------------|-------|----------------------------|
| Call the SQL Editor to display SQL statements generated in SQL generator. | SQL Editor | N/A | 1 days/3 days |

3.4.7. Execute results.

| Part | Dependency | Risks | Time to Complete (min/max) |
|---|------------|-------|----------------------------|
| Call the SQL Editor to execute SQL statements generated in SQL generator. | SQL Editor | N/A | 1 days/3 days |

3.5. Additional Notes

3.5.1. Extensibility Issues

These are just some rough ideas. We want to talk with the community about this more:

- The DBDefinition and SQL models are specialized for specific databases. They are supposed to contain information about specific database requirements, and some of this information might be useful in these components.
- We would like to design the UI in such a way that people can use its pieces for their own versions of wizards/table/other UI to do the same. In general, a good object-oriented design using design patterns should help here a lot:
 - Push common, generic functionality up to a standard set of base classes
 - Inherit (specialize) these base classes for specific cases
 - Use design patterns like factory, visitor, and strategy to help in reuse

DTP SQL Development Tools: Database Table Manipulation component Design

- Good places for extension points:
 - After the SQL is built, but before it is executed: An extension point allowing for manipulation of the built SQL, so database specific changes could be inserted. In the first version of these tools, we will pass contributors to this extension point simply the String for the built SQL. Once an extensible DDL parser that DTP plans to have in the future is in place, we will be able to pass contributors to the extension point a syntax tree generated by the DDL parser instead of a String.
 - Instead of using the default execution mechanism for the built SQL, have an extension point allowing alternative execution components to be registered.