

# *CROSSFIRE DB Framework Generator*

USER MANUAL



<http://www.crossfire.jp>

## Index

1.	What is CROSSFIRE DB Framework Generator?	4
1.1.	Easy software based on SQL	4
1.2.	The software writes complex codes instead of you	4
1.3.	The Computer never mistake nor be idle	4
2.	System Environment	5
3.	Setup Project	6
3.1.	Create New Project from wizard	6
3.2.	Necessary Setting	8
3.3.	Project Setting	8
3.4.	Framework	10
4.	Setup TABLE Information	10
4.1.	Take a look at “Hyper Database Editor File” in the Sample Project	10
4.2.	Create a new “Hyper Database Editor File”	12
4.3.	Generating SQL	13
5.	How to design methods executing SQL	14
5.1.	Take a look at “FwCreator Editor File” in the Sample Project	14
5.2.	What’s FwCreator Editor	15
5.3.	Script in FwCreator’s text fields	15
5.3.1.	Valuable	15
5.3.2.	Sub Query	15
5.4.	Hyper DB View help your programming	16
5.4.1.	Showing whole SQL	16
5.4.2.	Showing sample code	17
6.	Creating and using Classes and Methods executing SQL	18
6.1.	Create a new Class	18
6.2.	Create method for SELECT SQL	18
6.3.	Setup ‘WHERE’ Conditions	19
6.4.	Automatic Condition adjustment on executing the method	20
6.5.	Generate source code and executing method	21
6.6.	Getting Result	22
6.7.	Java Bean Mapping (OR Mapping)	23
6.8.	Nesessary conditions to use Java Bean Mapping	25
6.9.	Create method for INSERT	25
6.10.	Create method for UPDATE	27

6.11. Create method for DELETE .....	30
7. Database compatibilities .....	31
7.1. Types .....	31
7.2. LIMIT and OFFSET .....	31

## **1. What is CROSSFIRE DB Framework Generator?**

“CROSSFIRE DB Framework Generator” is a product to generate Java Program instead of human.

You can generate Java Program executing SQL by using JDBC with this product.

### **1.1. Easy software based on SQL**

This product's specification is based on SQL. When you design a System by any computer languages, you probably use any DBMS (Database Management System). And designing is strongly depends on Database Design, and SQL you execute for the designed data.

If you have some experience to design systems using SQL, it is very easy for you to use this product. That is because this software is based on the SQL.

You will be able to understand this software almost intuitive.

### **1.2. The software writes complex codes instead of you**

On Java Program, sometimes it is very complex and difficult to write basic parts of your system. You lost a lot of time to implement modules executing SQL you designed.

But by using this product, you can generate the module in less than a second.

### **1.3. The Computer never mistake nor be idle**

The human takes many mistakes on Program and you have to cost a lot of time to check bugs. But the computer never mistake.

By using this software, you can get source codes without artificial bugs.

## 2. System Environment

This software is developed as Eclipse plug-in, so this software works where Eclipse works.

※About Eclipse, refer to the Web page below

Eclipse official page

<http://www.eclipse.org/>

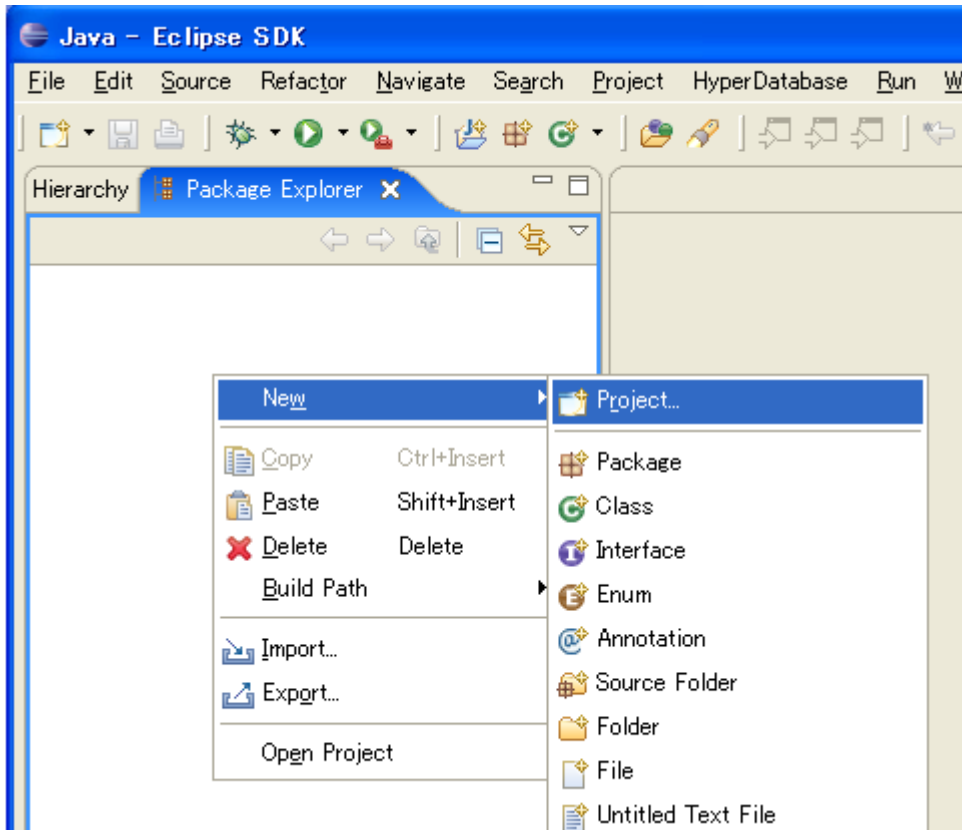
Service	Software	Comment
OS	Windows2000 or XP	Where Eclipse3.1 moves
Java SDK	J2SE ver1.4 2 or later	
Eclipse	Version 3.1	

### 3. Setup Project

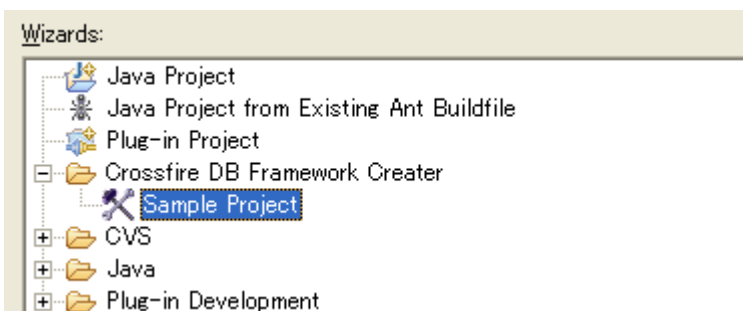
#### 3.1. Create New Project from wizard

You can create a new project with New Project Wizard.

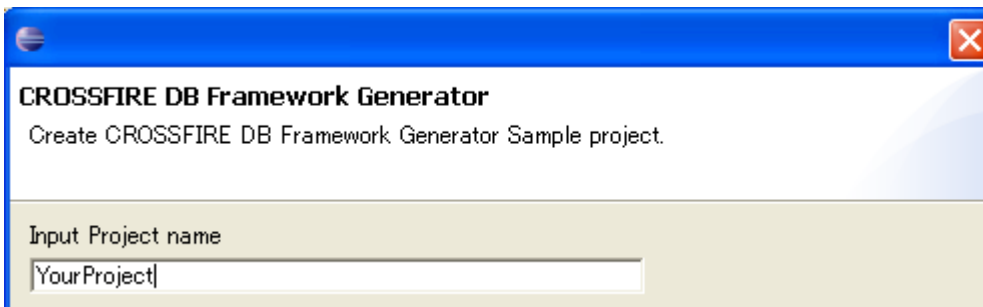
In order to create it, you should right-click in the “Navigator” or “Package Explorer”, and select “New” -> “Project”.



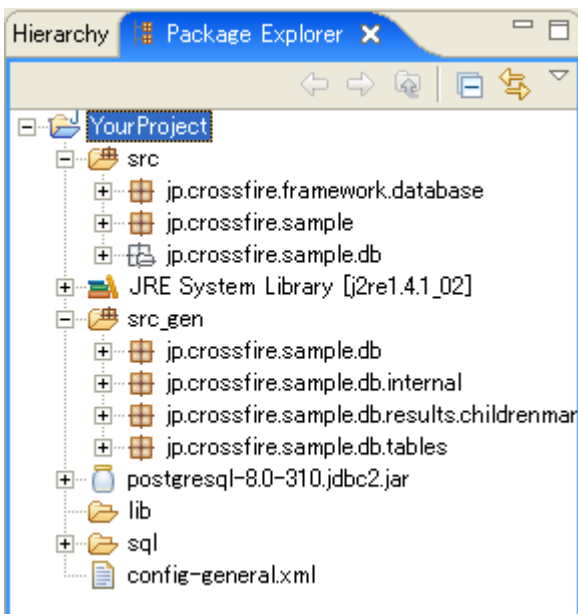
Then the wizard dialog appears. You should select “Sample Project” in the “Crossfire DB Framework Generator”.



At last, you should input project name.



After pushing "Finish" button on the wizard dialog, new project will be created.



The project is set project's configurations automatically.

But you had better know what is necessary to use this software. Next, I am going to refer to the necessary configurations.

### 3.2. Necessary Setting

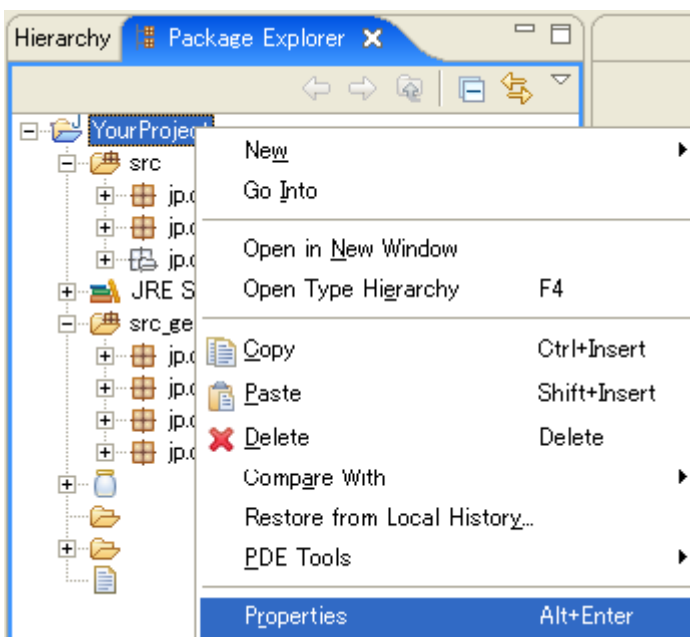
This software requires setting below.

- Project Setting
- Framework

And that's all.

### 3.3. Project Setting

You should open The Property of The Project by right-clicking your project and select “Property”.

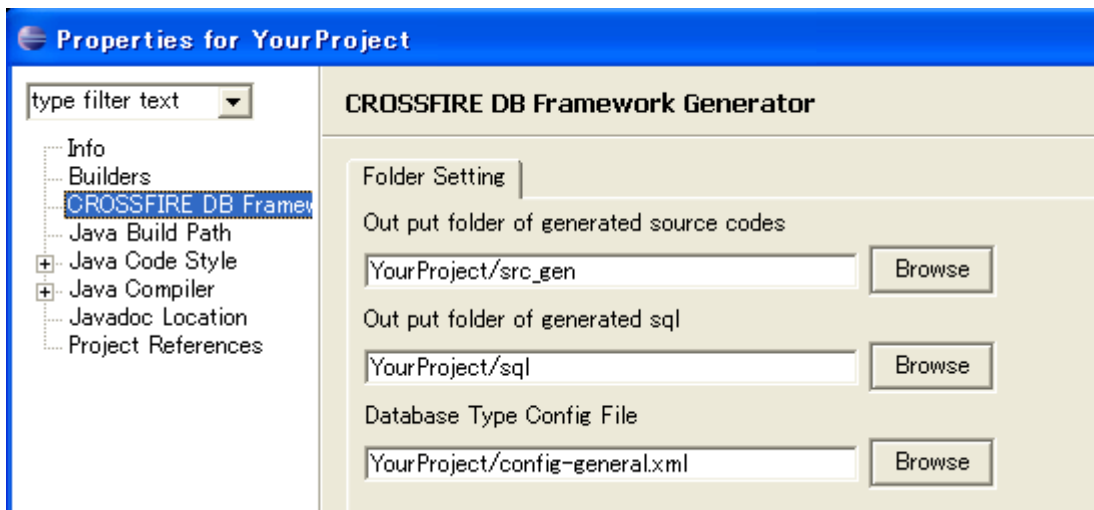


You should select “CROSSFIRE DB Framework Generator” category on the left to show the properties of this software.



Then following page appears.





There are 3 items to input.

#### **“Out put folder of generated source codes”**

Select folder to output source code generated by the “CROSSFIRE DB Framework Generator”.

#### **“Out put folder of generated sql”**

Select folder to output DDL generated by the “CROSSFIRE DB Framework Generator”.

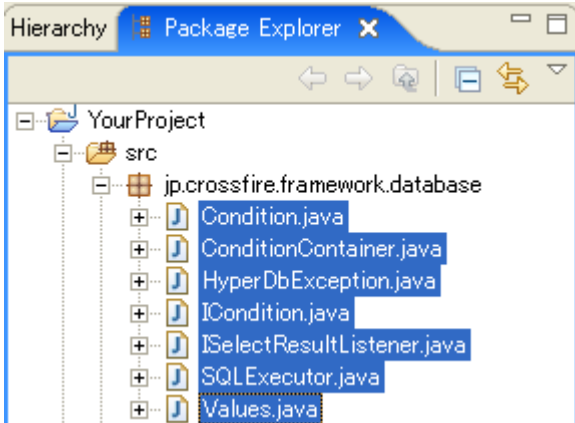
#### **“Database Type Config File”**

Select the “Database Type Configuration File”. This file is written in XML.

This file has information of Mapping between SQL type and Java type. By updating this file, you can use any DBMS.

### 3.4. Framework

The source codes generated by “CROSSFIRE DB Framework Generator” require the framework. But the framework is very tiny. Following 7 files are the all of the framework.



This framework is in the source folder you write, not in source folder for automatic generation. So you can extend this framework, as you like.

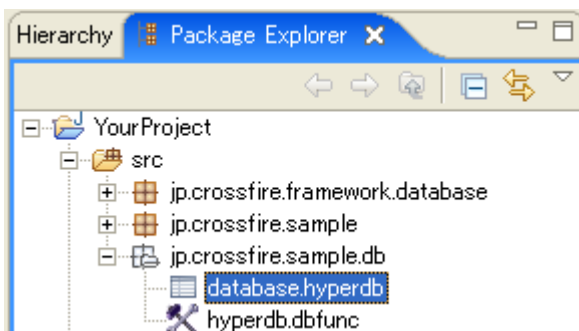
## 4. Setup TABLE Information

In order to use “CROSSFIRE DB Framework Generator”, the first thing you have to do is to setup the TABLE information.

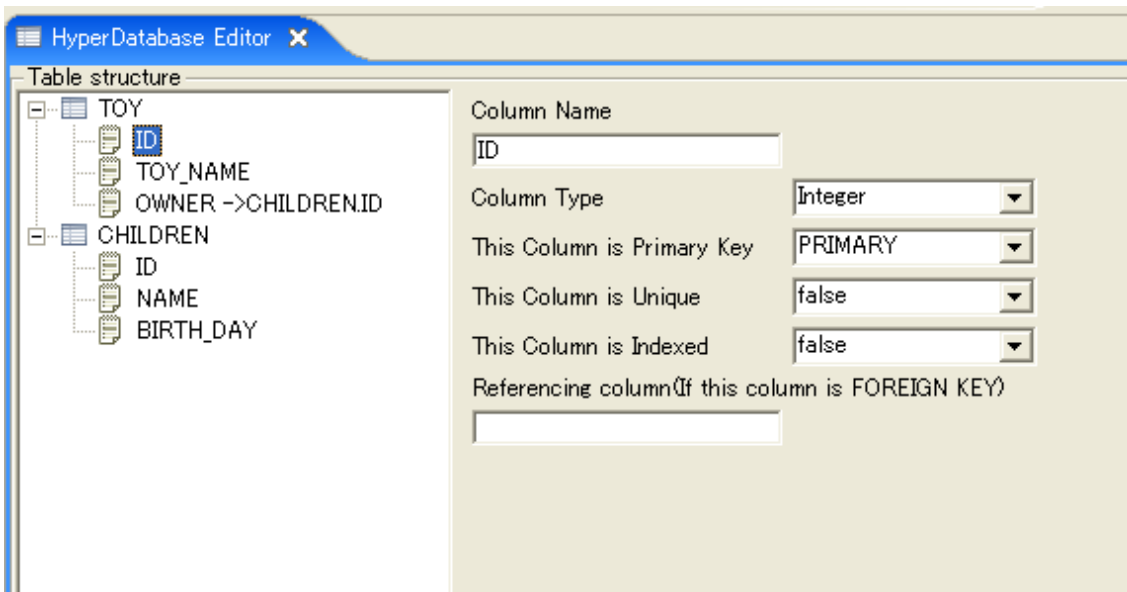
After setup this information, you can generate DDL (SQL) creating the tables.

### 4.1. Take a look at “Hyper Database Editor File” in the Sample Project

After creating the Sample Project, the project has the “Hyper Database Editor File” in the source folder. The “database.hyperdb” is the file on the following picture.



You should double-Click the file to open it with Editor.



The Items you have to input is below.

**TABLE**

Item	Explanation	Comment
name	Name of the Table	Necessary

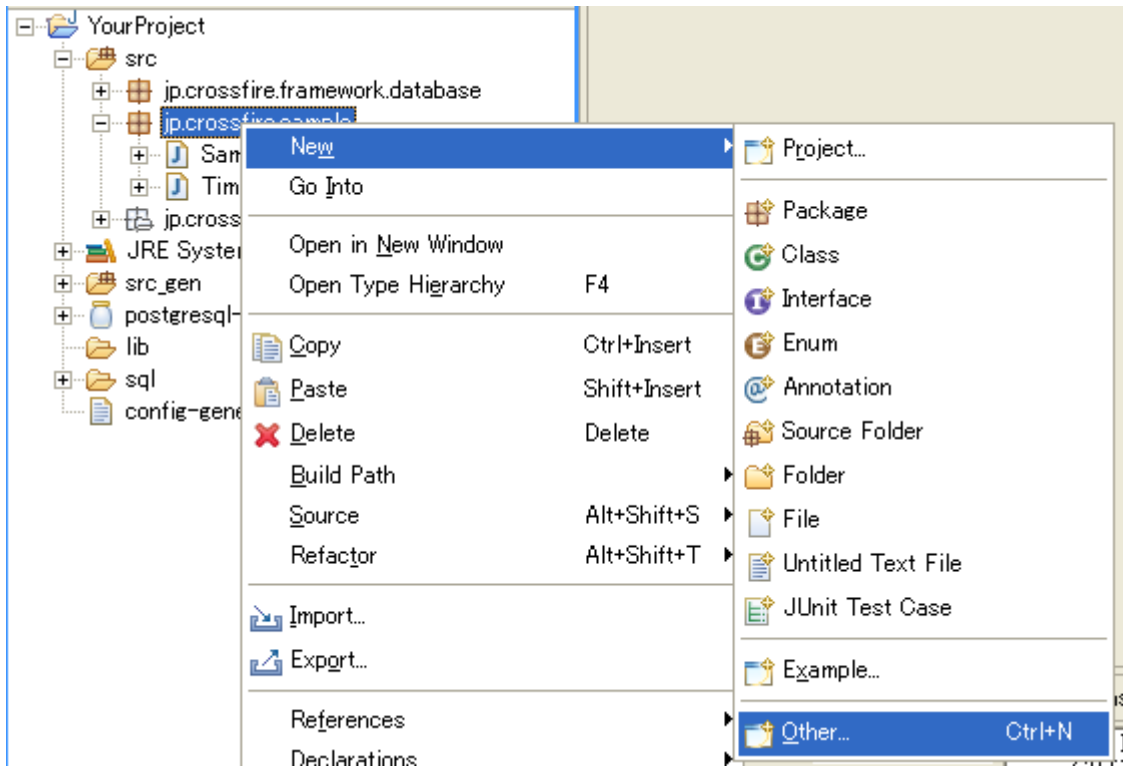
**COLUMN**

Item	Explanation	Comment
Column Name	Name of the Column	Necessary
Column Type	Type of the Column	Necessary
The Column is the Primary Key	If this column is the Primary Key, enter "PRIMARY"	
This Column is Unique	Whether this column is unique or not.	Necessary
This Column is Indexed	Whether this column is indexed or not.	Necessary
Referencing column(if this column is FOREIGN KEY)	If this column is the Foreign Key, enter the table and column this column refers. Ex. 'CHILDREN.ID'	

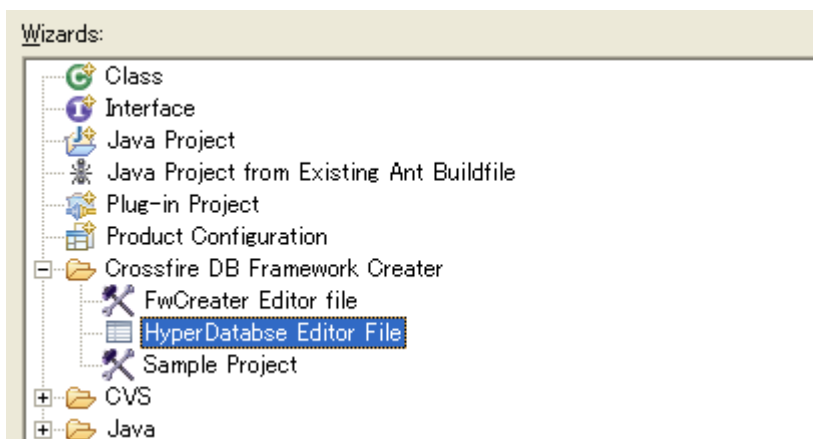
## 4.2. Create a new “Hyper Database Editor File”

You can create a new “Hyper Database Editor File” in your Project.

You should select “package” or folder in the source folder, and right-click, select “New” -> “Other”.



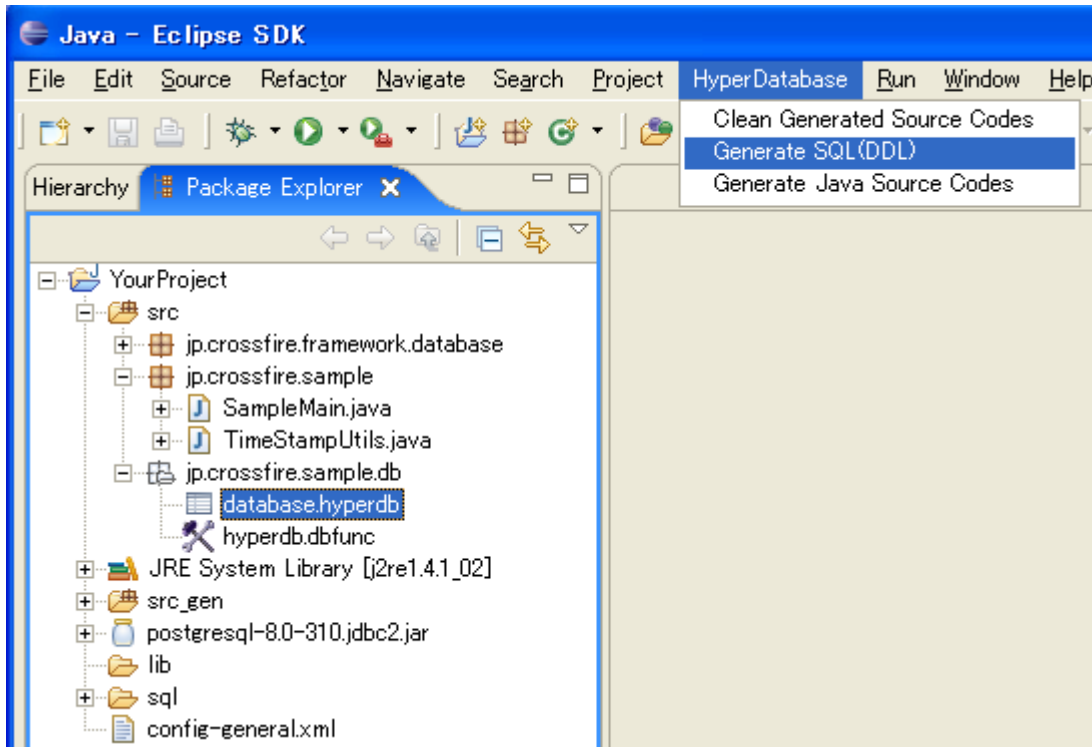
Next, you should select 'Hyper Database Editor File'.



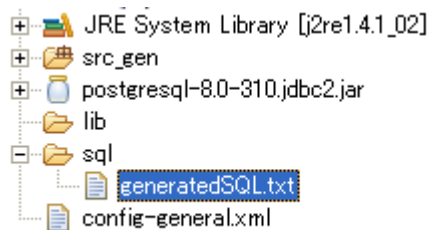
After the selection, dialog box appears, so you should push the “Finish” button.

### 4.3. Generating SQL

You should select “Hyper Database” -> “Generate SQL (DDL)” from the menu of the Eclipse.



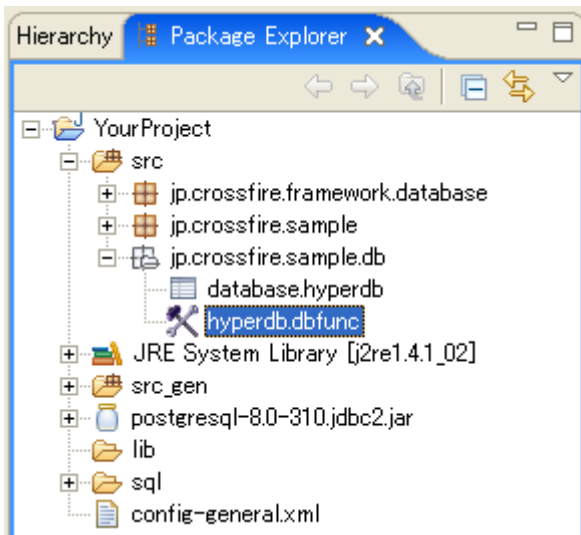
The file named “generatedSQL.txt” will be generated in the folder you selected as “Out put folder of generated sql” at the project’s properties.



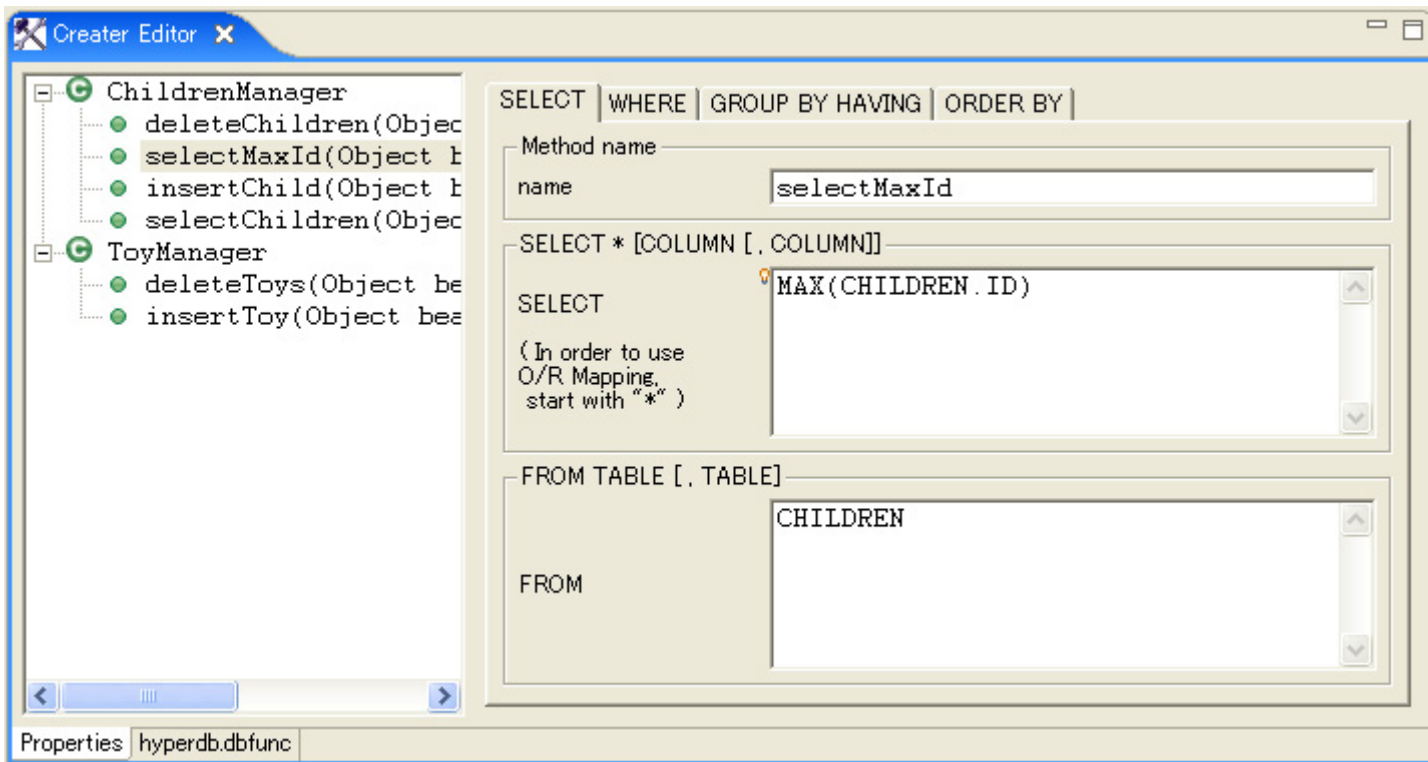
## 5. How to design methods executing SQL

### 5.1. Take a look at “FwCreator Editor File” in the Sample Project

After creating the Sample Project, the project has the “FwCreator Editor File” in the source folder. The “hyperdb.dbfunc” on the following picture is the file.



By double-clicking this file, you can open the file with FwCreator Editor.



## 5.2. What's FwCreator Editor

FwCreator Editor is the Editor to design Java Classes and methods executing SQL. The method has 4 types of SQL, SELECT, UPDATE, INSERT, DELETE.

You can design SQL and the method executing it by inputting into the fields on the Editor.

You can input easy script onto each text field.

## 5.3. Script in FwCreator's text fields

### 5.3.1. Valuable

When you execute the method, your programs have to input data into the method by argument. (Data is for example, id, name, or numbers in the "WHERE" condition)

For example, you designed following SQL.

```
SELECT * FROM TOY WHERE ID = '$ID'
```

\$ID is a valuable you want to input on executing the method.

The term starting with '\$' is the valuable and it is replaced automatically on executing the method.

### 5.3.2. Sub Query

You can call other method executing SELECT SQL as sub query.

For example, you design following SQL.

```
UPDATE CHILDREN SET ID = (@selectMax) + 1
```

'selectMax' is a method name executing SELECT SQL. The term starting with '@' means sub query. If selectMax means SQL below.

```
SELECT MAX(CHILDREN.ID) FROM CHILDREN
```

The first SQL will be transformed like below.

```
UPDATE CHILDREN SET ID = (SELECT MAX(CHILDREN.ID) FROM CHILDREN) + 1
```

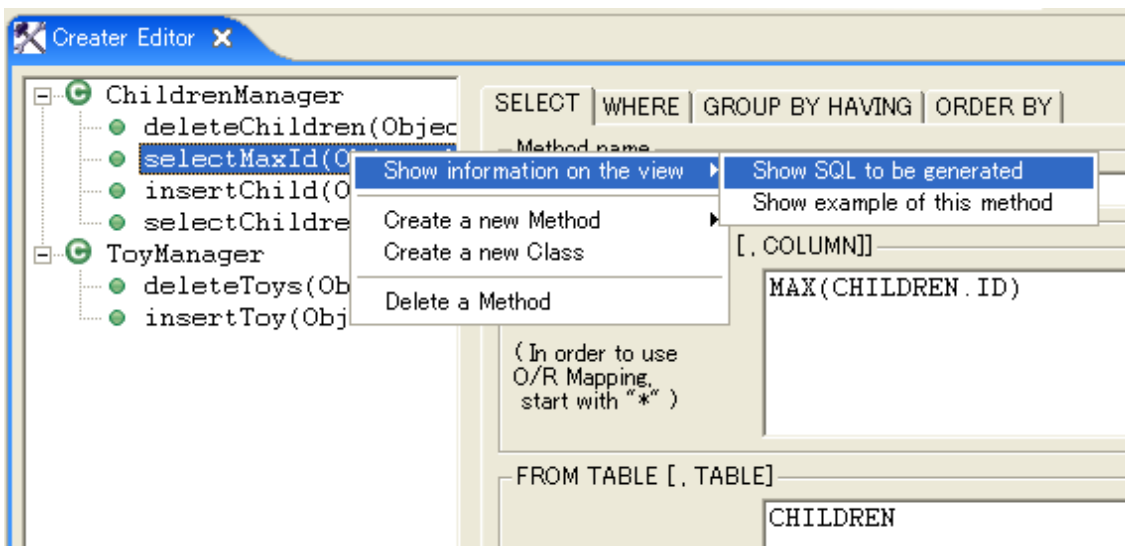
## 5.4. Hyper DB View help your programming

The “Hyper DB View” is a powerful View. You can know

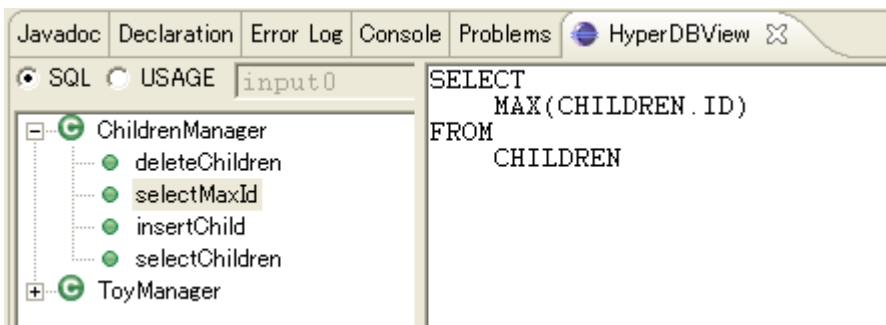
- Whole SQL design
- Sample code using generated Classes and Methods

### 5.4.1. Showing whole SQL

You can see whole SQL with the “Hyper DB View”. You should select the method on the FwCreator Editor and right-Click it. Then select “Show information on the view” -> “Show SQL to be generated”.



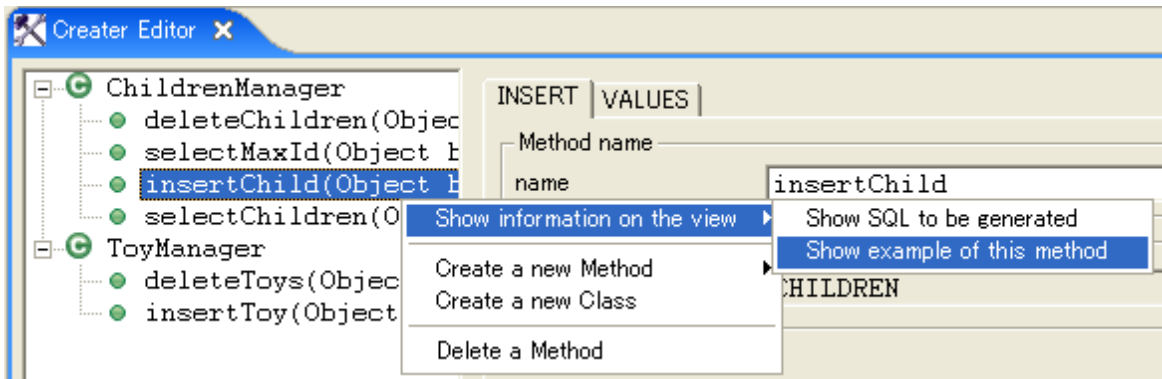
Then “Hyper DB View” appears and shows the whole SQL on it.



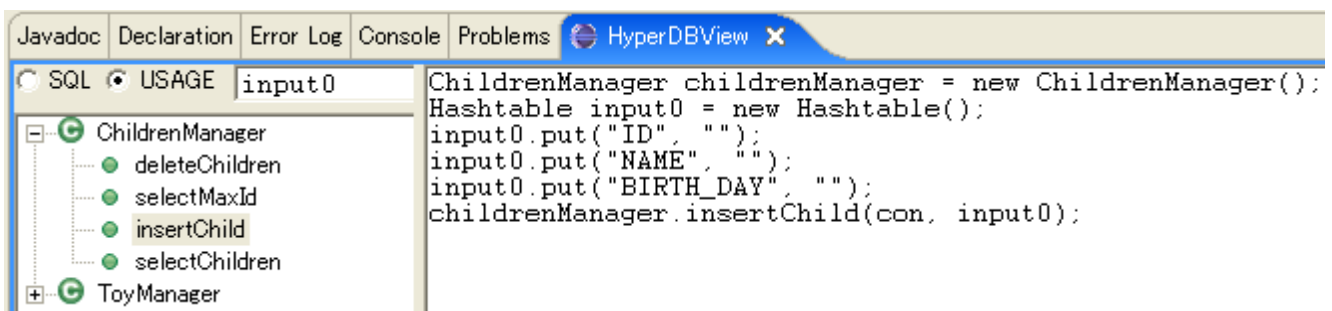


### 5.4.2. Showing sample code

You can see whole SQL with the “Hyper DB View”. You should select the method on the FwCreator Editor and right-Click it. Then select “Show information on the view” -> “Show example of this method”.



Then “Hyper DB View” appears and shows the example code on it.



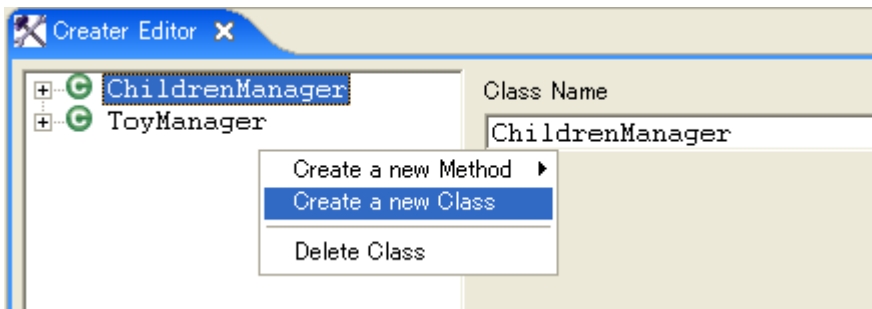
The variables used in the script are listed in the source code. So this is useful tool on using generated methods on programming.

You can copy this code and paste into your Java source codes.

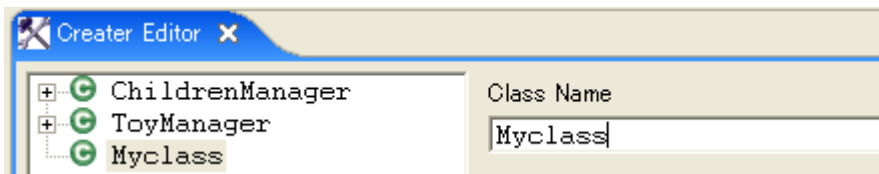
## 6. Creating and using Classes and Methods executing SQL

### 6.1. Create a new Class

You can create a new Class. You should right-Click on the FwCreator Editor and select “Create a new Class”.

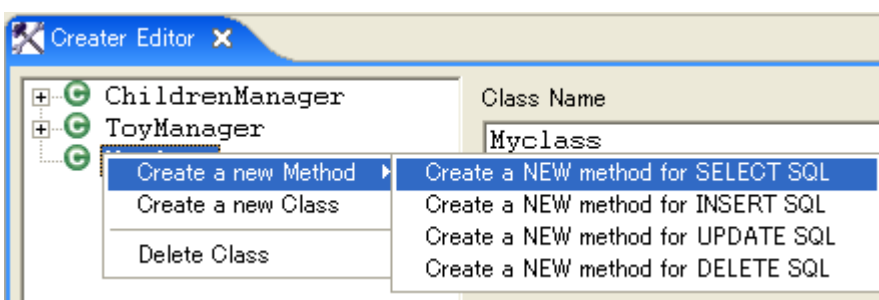


Then a new class will be created, so you have to change the name as you like.



### 6.2. Create method for SELECT SQL

You should select the class where you are going to create method and right-Click it. Then you should select “Create a new Method” -> “Create a NEW method for SELECT SQL”.



Then the method for SELECT method will be created.

Next, you should input each fields. **You can use valuable on each Fields.**

(Valuables begin from '\$'. See [Script in FwCreator's text fields](#))

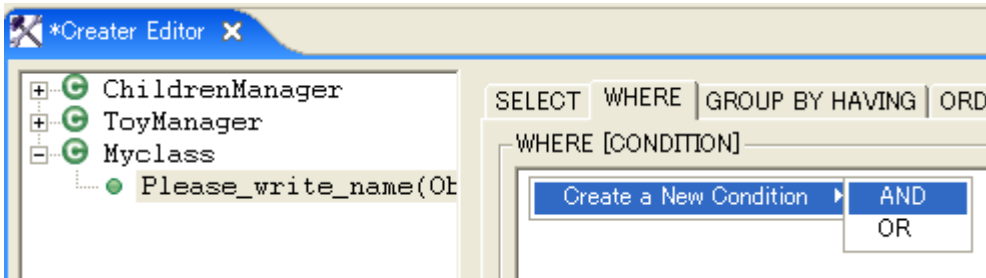
But the way to input 'WHERE' condition is different from other text field, so I'll refer to it in the next section.

### 6.3. Setup 'WHERE' Conditions

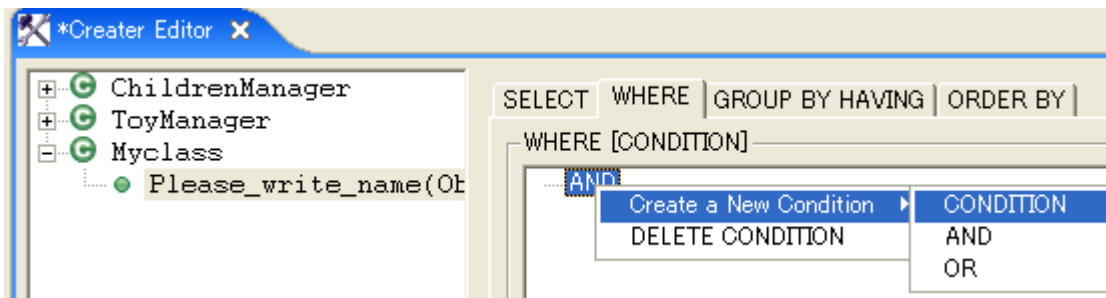
You can setup the 'WHERE' condition by making condition tree.

At first time, there is nothing, so you have to add a new root condition.

You should right-Click in the FwCreator Editor and select "Create a New Condition" -> "AND" ("OR").

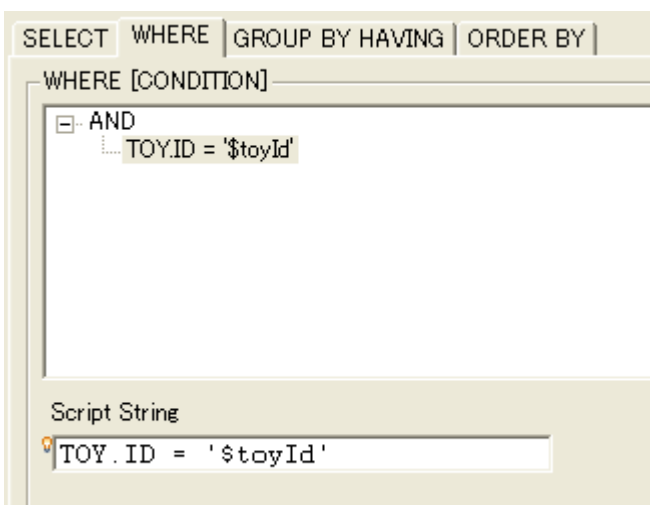


Then first condition is created. Next, you should select the condition you made, and select "Create a New Condition" -> "CONDITION".



Then a condition is added into the first condition.

By selecting the condition, you can edit the condition.

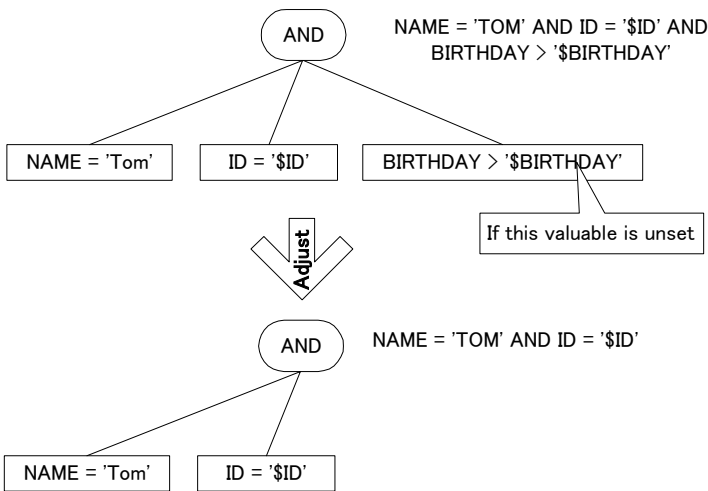


### 6.4. Automatic Condition adjustment on executing the method

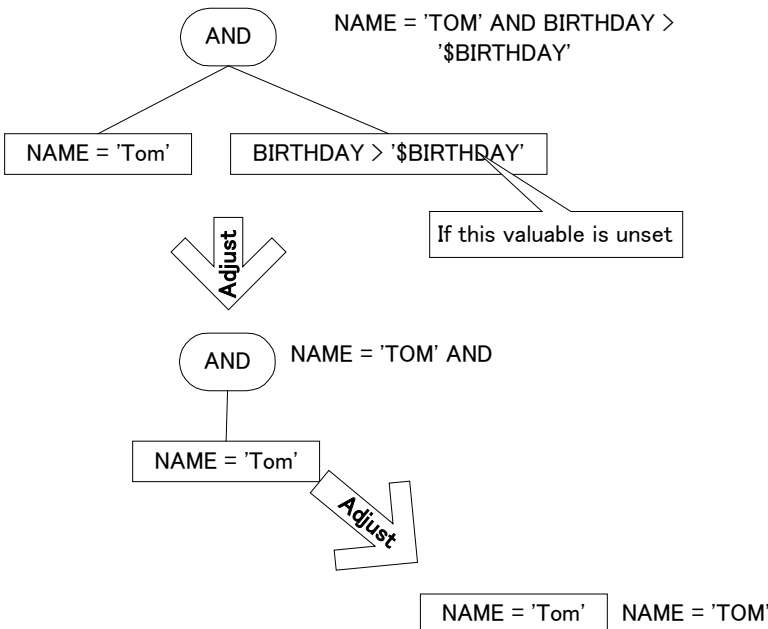
This function is very powerful. You can use valuable in the condition's field. But when you execute the method if the valuable is unset, what happens?

Then the generated program (Framework) automatically adjusts the condition.

**Ex.1.** You designed condition below and 1 of 3 condition's valuable is unset.

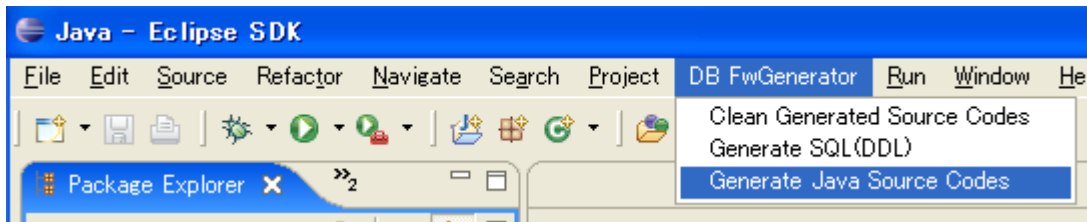


**Ex.2.** You designed condition below and 1 of 2 condition's valuable is unset.



## 6.5. Generate source code and executing method

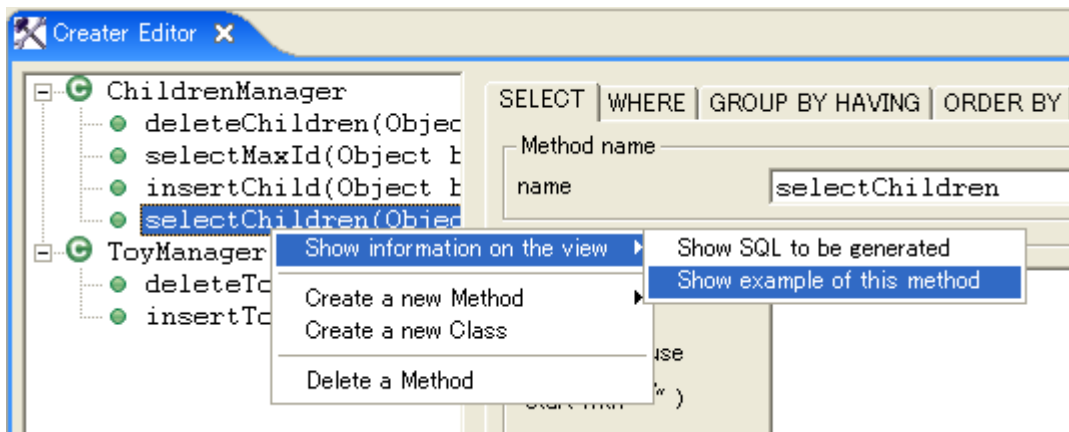
You should select “DB FwGenerator” -> “Generate Java Source Codes” from the menu of the Eclipse to generate source codes.



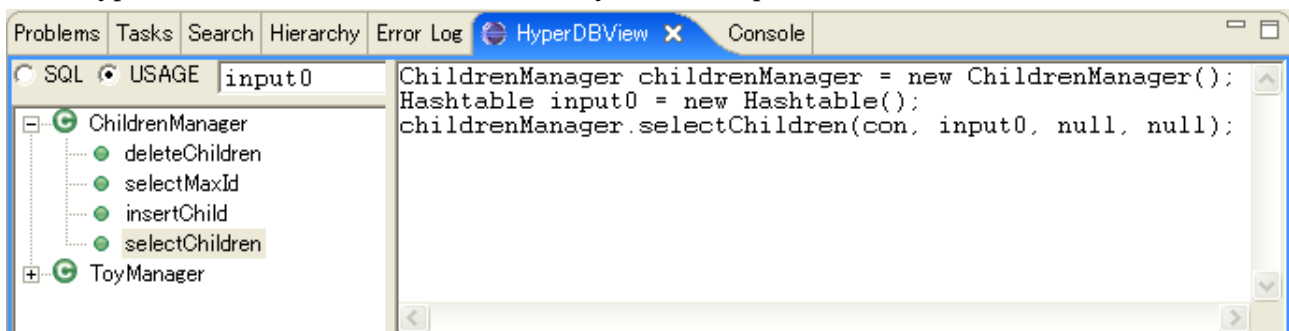
After a moment, source codes are generated in the source folder, which you selected as “Out put folder of generated source codes”.

In order to use the method, you have to know how to use the method.

You can make this product show sample code with the “HyperDb View”. You should select the method you want to use, right-Click it, and select “Show information on the view” -> “Show example of this method”.



The “Hyper DB View” is launched and shows you the sample source code.



## 6.6. Getting Result

The method executing SELECT SQL that you generated returns “java.util.List”. The List contains objects whose type is “java.util.Hashtable”.

### Sample Code

```
// Select
ChildrenManager childrenManager = new ChildrenManager();
Hashtable input = new Hashtable();
List result = childrenManager.selectChildren(con, input, null, null);

// You can get data from List of Hashtable
System.out.println("-----Check RECORDSET RESULT-----");
Iterator it = result.iterator();
while(it.hasNext()){
    Hashtable hash = (Hashtable)it.next();
    System.out.print(hash.get("name"));
    System.out.print(" " + hash.get("birth_day"));
    System.out.println(" " + hash.get("toy_name"));
}
```

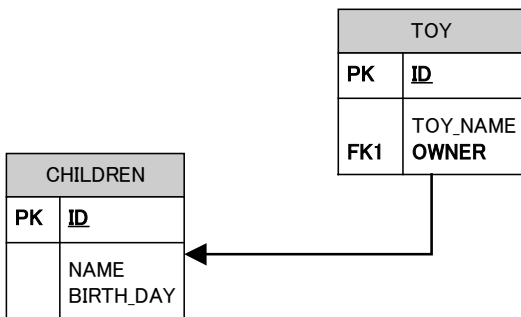
## 6.7. Java Bean Mapping (OR Mapping)

The class Generated class has other way to get the result of SELECT SQL. The generated Class has Java Bean, which has same fields with the TABLE of the Database.

The Java Bean also has reference Lists or Objects. The references are generated from the setting of the Foreign Key.

On this sample, there are 2 TABLES. The OWNER is the FOREIGN KEY making reference to CHILDREN TABLE.

### Structure of Sample's DATABASE TABLES

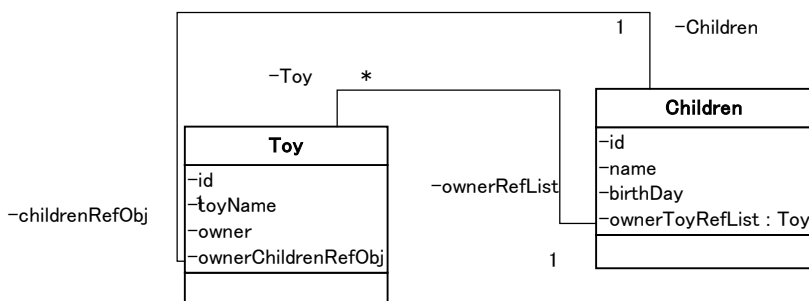


For this structure, the “DB Framework Generator” generates Java Bean, like following picture.

The Bean of TABLE having FOREIGN KEY (Toy) has reference of the Bean of the TABLE to which the FOREIGN KEY makes reference (Children).

And the Bean of the TABLE to which the FOREIGN KEY makes reference (Children) has List of Toy Bean.

### Java Bean



The sample code of using the Mapped Java Bean is below.

### Sample Code

```
// Select
ChildrenManager childrenManager = new ChildrenManager();
Hashtable input = new Hashtable();
childrenManager.selectChildren(con, input, null, null);

// You can also get data from Mapped Java Bean
// JavaBeans are automatically generated and Mapping
// Logic is generated too.
System.out.println("-----Check ORMAPPING RESULT-----");
Iterator it = childrenManager.getChildren().iterator();
while(it.hasNext()){
    Children child = (Children)it.next();
    System.out.println("Child Name : " + child.getName());

    Iterator toyIt = child.getOwnerToyRefList().iterator();
    while(toyIt.hasNext()){
        Toy toy = (Toy)toyIt.next();
        System.out.print("  having: " + toy.getToyName());
        System.out.println("  owner-> "
            + toy.getOwnerChildrenRefObj().getName());
    }
}
```



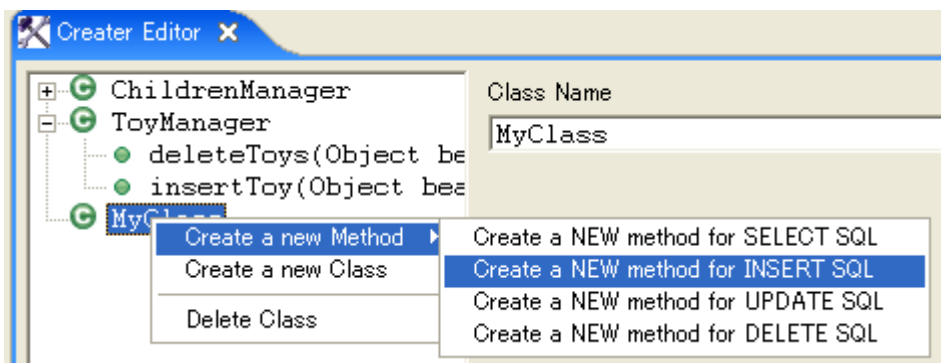
## 6.8. Necessary conditions to use Java Bean Mapping

In order to use Java Bean Mapping function, you have to satisfy following conditions.

- The 'SELECT' section of the SELECT SQL has to start with '\*'
- The 'FROM' section of the SELECT SQL has to contain TABLE name to map

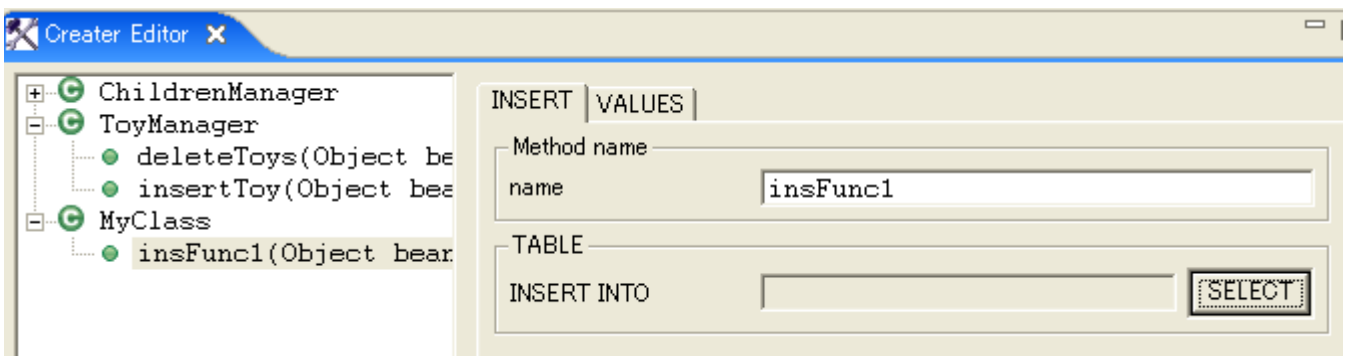
## 6.9. Create method for INSERT

You should select the class where you are going to create method and right-Click it. Then you should select "Create a new Method" -> "Create a NEW method for INSERT SQL".

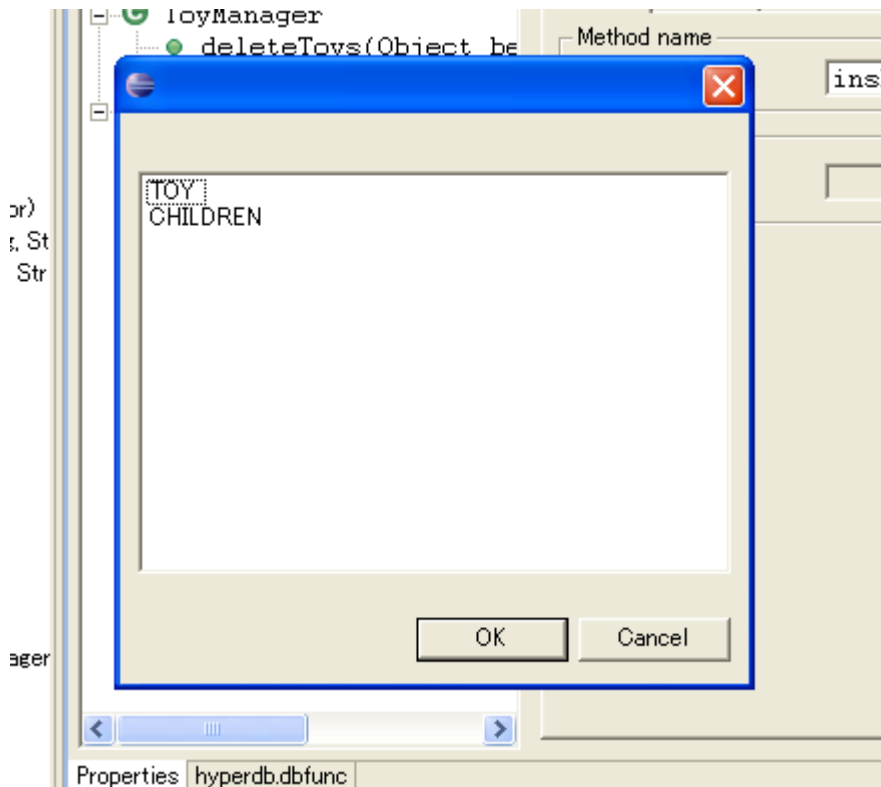


Then the method for INSERT method will be created.

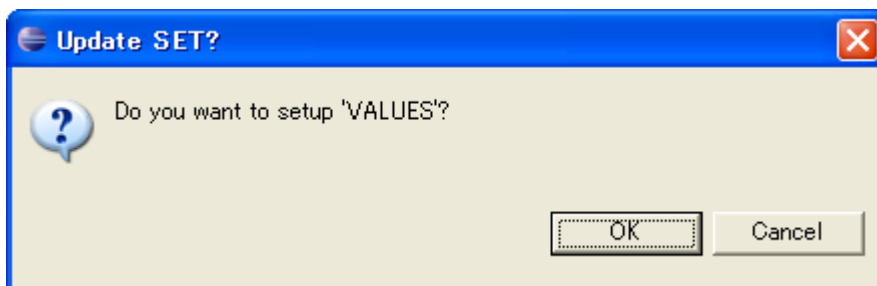
Then you have to select TABLE to INSERT. You should push "SELECT" button and select the TABLE.



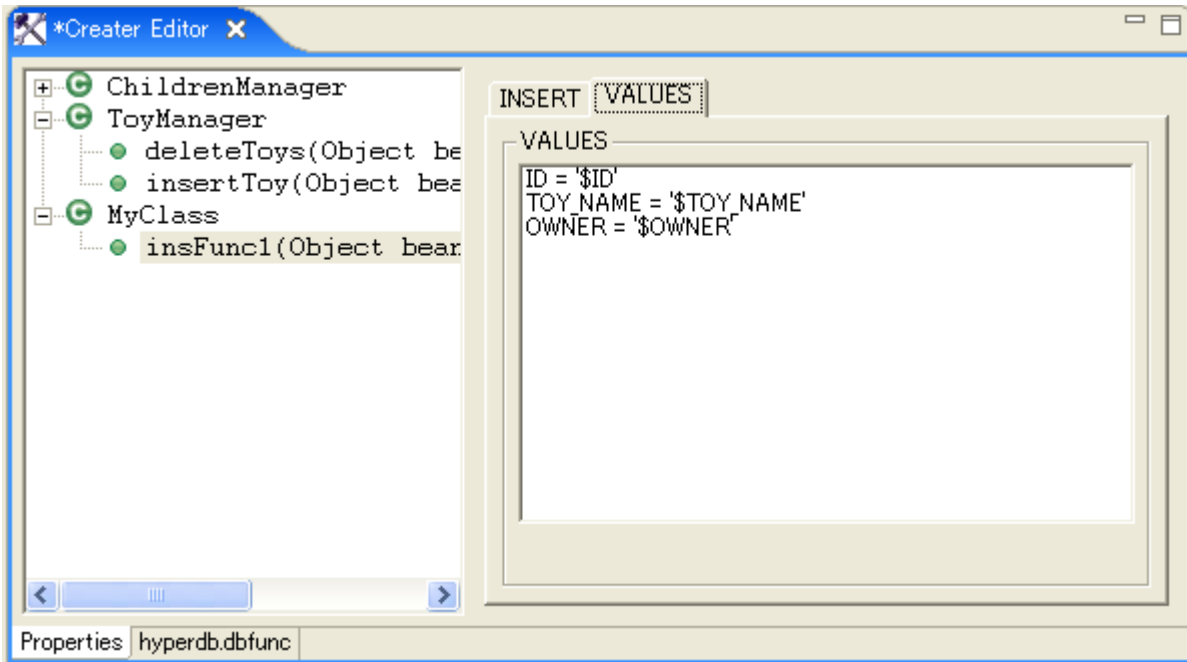
The selection dialog appears, and you should select a TABLE.



After the selection, a dialog to confirm doing default setup appears, then you should push 'OK' button.

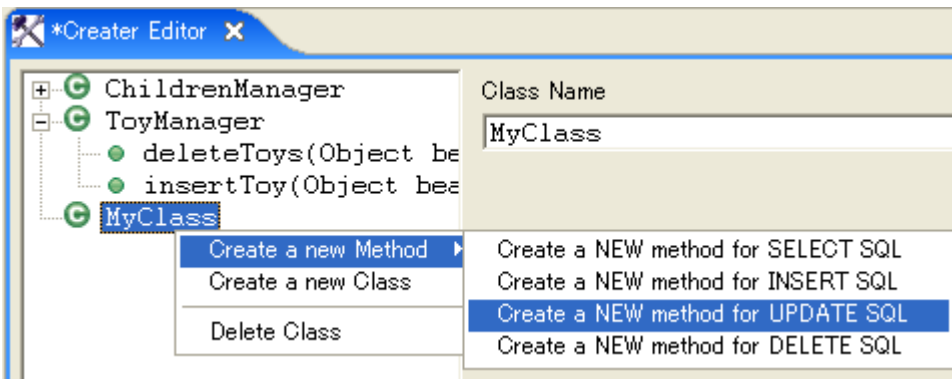


By pushing 'OK' button, 'VALUES' section of the INSERT method will be set automatically. If you want to use Sub Query or some function of the DBMS, you should change the script.



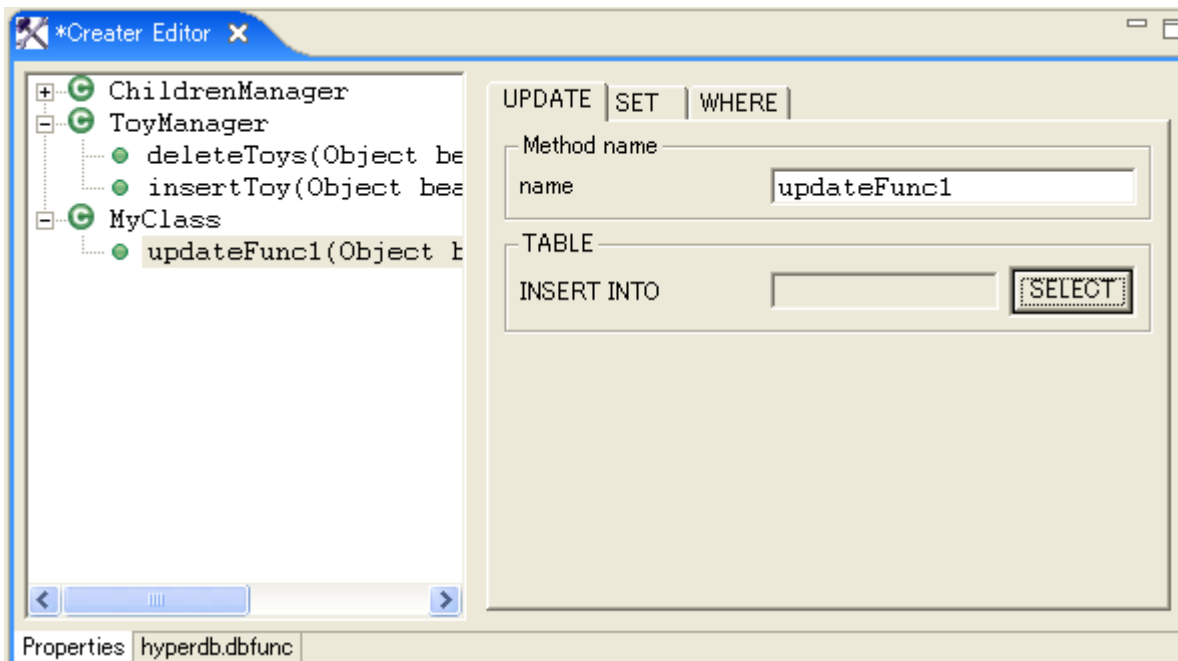
### 6.10. Create method for UPDATE

You should select the class where you are going to create method and right-Click it. Then you should select "Create a new Method" -> "Create a NEW method for UPDATE SQL".

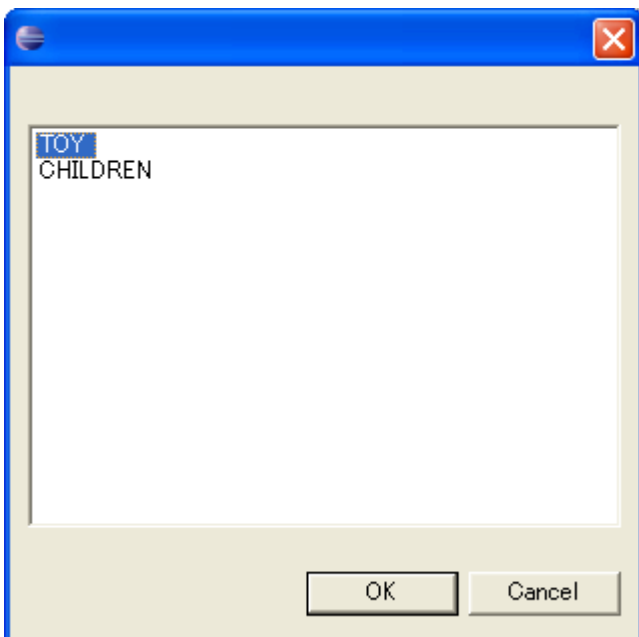


Then the method for UPDATE method will be created.

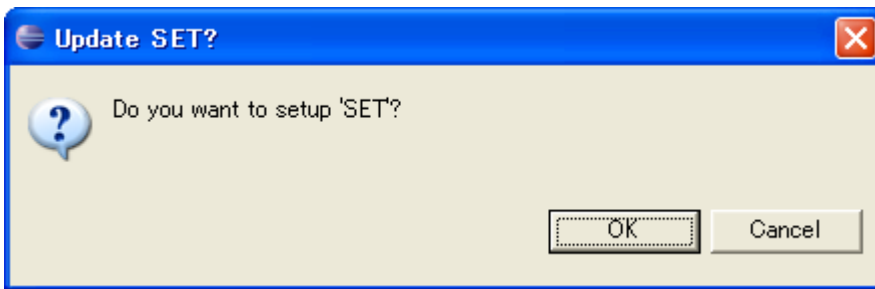
Then you have to select TABLE to UPDATE. You should push "SELECT" button and select the TABLE.



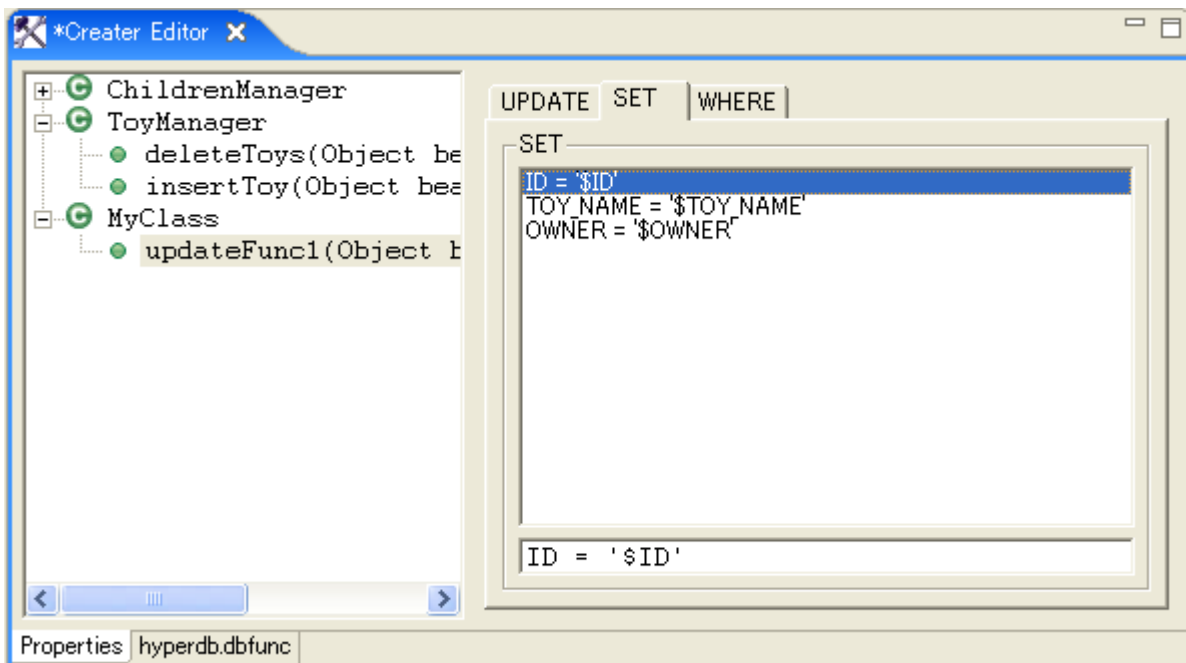
The selection dialog appears, and you should select a TABLE.



After the selection, a dialog to confirm doing default setup arrears, then you should push 'OK' button.



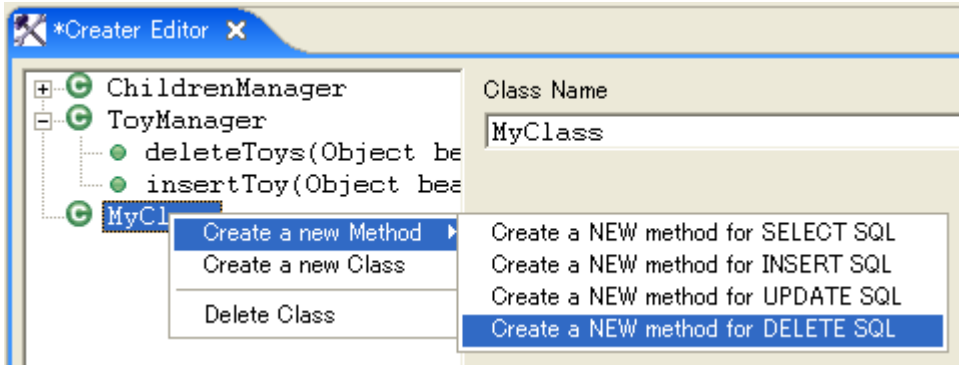
By pushing 'OK' button, 'SET' section of the UPDATE method will be set automatically. If you want to use Sub Query or some function of the DBMS, you should change the script.



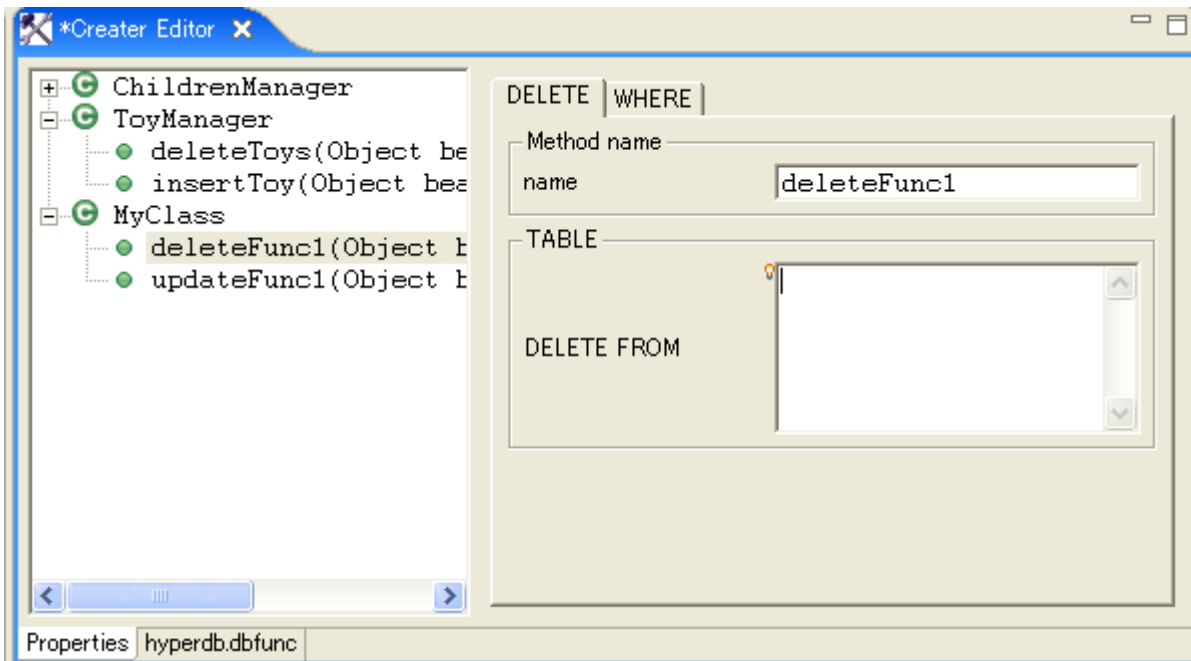
The way to setup the 'WHERE' section is same with "SELECT SQL".

### 6.11. Create method for DELETE

You should select the class where you are going to create method and right-Click it. Then you should select “Create a new Method” -> “Create a NEW method for DELETE SQL”.



After the selection, the method executing DELETE SQL will be created.



The way to setup the ‘WHERE’ section is same with “SELECT SQL”.

## 7. Database compatibilities

### 7.1. Types

You can configure types of the columns by setting up “Database Type Config File”. You can select this file at Properties of your project.

The format of XML is below.

```
<?xml version="1.0" encoding="UTF-8"?>
<HYPERDBCONFIG>
  <TYPE name="[Type name to show in GUI]"
        dbType="[Type of database column]"
        javaType="[Corresponding Java Class]" />
</HYPERDBCONFIG>
```

### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<HYPERDBCONFIG>
  <TYPE name="VARCHAR(16)" dbType="VARCHAR(16)" javaType="java.lang.String" />
  <TYPE name="VARCHAR(128)" dbType="VARCHAR(128)" javaType="java.lang.String" />
  <TYPE name="VARCHAR(255)" dbType="VARCHAR(255)" javaType="java.lang.String" />
  <TYPE name="TEXT" dbType="TEXT" javaType="java.lang.String" />
  <TYPE name="TIMESTAMP" dbType="TIMESTAMP" javaType="java.sql.Timestamp" />
  <TYPE name="DATE" dbType="DATE" javaType="java.sql.Date" />
  <TYPE name="Integer" dbType="INT" javaType="java.lang.Integer" />
  <TYPE name="REAL" dbType="REAL" javaType="java.lang.Double" />
  <TYPE name="BOOL" dbType="BOOL" javaType="java.lang.Boolean" />
</HYPERDBCONFIG>
```

### 7.2. LIMIT and OFFSET

There are some databases do not supports LIMIT and OFFSET. Then you have to set the 3<sup>rd</sup> and 4<sup>th</sup> arguments of method executing select method “null”.