Assignment 8 Objects in Procedures

Objectives

1. Use procedures and functions to interact with the database
2. Design custom types to be used with procedures
3. Design procedures that use custom types
4. Use procedures in Java as opposed to insert statements

Instructions

Follow the instructions below to define types for procedures, create procedures, and use Java code to interact with these procedures and functions in the database.

Database

Data

For this assignment you will be working with the table of data representing the courses taken by computer science students in the 420.B0 program. Reuse your code from assignment 7 to create the tables and insert data. Ensure that you include this code in your final submission.

Custom Types

As seen in class, application code should interact with the database through procedures and functions as opposed to writing insert and update statements. However, procedures can have many parameters if we want to expose every column in a table. Instead, we will define a custom type that maps to an object in our Java code. This will make writing both the procedure and Java code easier.

Create a new object type called course_typ that contains the same fields and types as the courses table. Ensure, the order of the fields is the same as the columns in the table. Note, types persist like tables, so they should be dropped at the top of your script.

Writing the procedure

Create a procedure called add_course which takes a single parameter course, of type course_typ. Inside the procedure right the code necessary to insert the data into the courses table. Note, you can access elements of the course_typ using the dot notation.

Creating the Java Application

Setting up Java Project

Reuse your code and java project from assignment 7. You will update aspects of your code to work with the procedures you designed above.

Adding data using procedures

Modify the AddToDatabase method in Course to use the procedure we have defined above. You should no longer use a PreparedStatement with ‘insert’ to update the DB.

1. Add a field called typeName to the Course class. Set the value to ‘COURSE_TYP’, i.e., the name of the type in our database.

2. Implement the SQLData interface for the Course class.
    - Note, you can use VS Code to automatically generate the unimplemented methods
    - In the getSQLTypeName method return the type name
    - In the readSQL method, set the type name and use the stream object with setter methods to read the values of the object in order.
    - In the writeSQL method, use the stream object with the getter methods to write the values of the object in order.
3. Modify AddToDatabase to call the procedure add_course.
    - You first must define a mapping between the object and oracle. See the sample code below:

```
Map map = conn.getTypeMap();
        conn.setTypeMap(map);
        map.put(typeName, Class.forName("Course"));
```

    - Create a callable statement that calls the add_course procedure. Use the setObject method to provide the Course object (this). Note, much like PreparedStatements, CallableStatements must closed when finished.

Testing the code

Reuse the App class that contains a main method. In the main method prompt the user to provide their username and password. Use the **CourseListServices** to add the Programming IV course to the database.

Deliverable

Submit a zip called assignment8.zip. Additionally, submit the SQL Script file assignment8.sql. Be sure your script is runnable from start to finish (i.e., it drops tables, types, etc. at the start) and has the necessary comments to be well understood. Finally include your Java code for assignment8. Ensure it is well structured, compiles, and runs without errors.

Resources