Uppsala University
Department of Information Technology
Kjell Orsborn

DATABASE DESIGN I - 1DL300

Assignment 3 - SQL queries, views and JDBC API

1 Part one – SQL queries and views

The purpose of this exercise is to practice writing both simple and more advanced queries in SQL, including the use of aggregate functions and views. This part of the assignment should be completed using the Mimer database management system.

There is an supervised introduction to the assignment in your schedule. Each group should sign up for one lab time on lists posted on the board outside 1346, to avoid overcrowded labs.

1.1 Preparation

1. If you want to perform the assignment on your own PC, install Mimer.
2. In order to have a uniform starting point, the initial Johnson Brothers database is used here, and not the outcome of your assignment2. Drop all tables that were added to Jonson Brothers database during assignment 2.
3. Set up the Jonson Brothers database. Instructions and scripts can be found at the assignment page found from the course webpage.

Related study material:

- Elmasri and Navathe [EN10]: Chapter 8, 9.2.
- Padron-McCarthy and Risch [PMR05]: Chapter 7, 8, and 9.
- MIMER SQL documentation [M10].

1.2 Assignment

In this assignment you should practice to use SQL to pose simple and more advanced queries towards The Jonson Brothers retail company database that you were introduced to in the Assignment 2. You will try several SQL features including for instance subqueries, aggregated queries and views.

1.3 Exercises

Use SQL to find the answers to the questions below towards your Jonson Brothers company database. Whenever a question requests information about entities that have both a number and a name, select both the number and the name to make your results more useful. For example, in Question (3) return both the number and the name of the parts that were asked for.

1. List all employees, i.e. all tuples in the `EMPLOYEE` relation.
2. List names of all departments, i.e. the `NAME` attribute for all tuples in the `DEPT` relation.
3. What parts are not in store, i.e. `QOH=0`? (QOH = Quantity On Hand).
4. Which employees have a salary between 9000 and 10000 (inclusive)?
5. What was the age of each employee (i.e. `STARTAGE`) when they started working here?
6. Which employees have a last name ending with “`son`” (in queries you can also use simple quotes: `'son'`)? Retrieve employee names and numbers.
7. Which items have been delivered by a supplier called “`Playskool`”? Formulate this query using a subquery in the where-clause.
8. Formulate the same query as above, but without a subquery.
9. What is the name and the color of the parts that are heavier than a black tape drive? Formulate this query using a subquery in the where-clause. (The SQL query should not contain the part’s weight as a constant.)
10. Formulate the same query as above, but without a subquery. (Again, the query should not contain the weight as a constant.)
11. What is the average weight of black parts?

12. What is the total weight of all parts that each supplier in the state Massachusetts (i.e. “Mass”) has delivered? Retrieve the total weight for each of these suppliers.

13. Create a new relation (a table) that contains the items that cost less than the average price for all items. List the contents of the new table, including item names.

14. Create a view that contains the items that cost less than the average price for all items. Query the view and show the result. What is the difference between (13) and (14)?

15. Create a view that calculates the total cost for all sales contained in each transaction (from debit table) by considering price and quantity of each bought item. (To be used for charging customer accounts). The view should return transaction number (from debit table) and total cost. Query the view and show the result.

16. Suddenly, an earthquake strikes. Explain, how you would proceed to remove all suppliers in Los Angeles from the table SUPPLIERS? (How would the corresponding SQL statements look like?) Do **not** execute!

17. A database manager in the company has tried to find out which suppliers have delivered items that are sold. He has created a help view and can find how many items are sold from each supplier of the items:

```
1 SQL> create view sale_supply(supplier, item, quantity) as
       select supplier.name, item.name, sale.quantity
       from supplier, item, sale
       where supplier.number = item.supplier and
             sale.item = item.number;
Ok

2 SQL> select supplier,sum(quantity) from sale_supply
       group by supplier;

6 rows selected
----------------------------------
|      SUPPLIER      |    SUM    |
----------------------------------
|     Cannon         |     6     |
|     Koret          |     1     |
|     Levi-Strauss   |     1     |
|     Playskool      |     2     |
|     White Stag     |     4     |
|     Whitman's      |     2     |
----------------------------------
```

Now he would also like to find out suppliers that have delivered some items none of which have been sold. Help him! Drop and redefine the sale_supply view in such a way that items that have been delivered by suppliers, but have never been sold, are considered as well. Repeat the above query on your view.

Hint: The above definition of `sale_supply` uses an (implicit) inner join. An inner join removes suppliers that have not had any of their delivered items sold. Consider to replace the inner join with some other types of join.

1.4 Examination

The following should be handed in:

1. All SQL commands issued
2. All command results from the database server
3. Answers to questions, and explanations where appropriate

You should hand in the solutions and answers to questions as one report per group.

2 Part two – Java Database Connection using the JDBC API

2.1 Objectives

Create and run a JDBC-based client application using client drivers. This part of the assignment demonstrates the ease to connect to a DBMS through a JDBC-based client.

2.2 Preparations

It should be noted that in this part of the assignment we will only be concerned with one table, the Employee table, of the Jonson Brothers database.

1. Read tutorials and related material about JDBC provided on the course website.

2. locate the `Mimer Database administrator` from the start menu. Make sure that the default "JB" database is running. You can also use any other database instance, but then you need to set dbName variable in Demo program accordingly.

3. If there is an employee table in the database, open a SQL Batch of Mimer and drop it. Use `cascade` option to remove any potential dependencies too.

4. Create a folder for Assignment 3 somewhere in your home directory (e.g. C:/Users/YOUR_NAME/Documents).

5. Download the demo program `demoJDBCclient.java` from the course website and save it in the "Assignment3" folder. The demo program is available at assignment page from course home page.

2.3 The assignment

The assignment covers the use of JDBC database API to connect to a database (a Mimer database in this case). While your Java program is connected to the database, you should also create and populate the the employee table. Furthermore, you should update the Salary for one or several employees and use a query to extract the identity number, name and salary of the employees at Jonson Brothers from a Java program. The Java program should end by deleting the Employee table and closing the connection. Most of the work of writing a JDBC client has already been done for you and can be found in demoJDBCclient.java.

1. Open a command window.

2. Change to the "Assignment3" directory.

3. Make a copy of the demo program called demoJDBCclient.java. Operating System Command (Windows): `copy demoJDBCclient.java TheBrosClient.java`

4. Open the TheBrosClient.java file in a text editor and update the class name to reflect the new file name; Original declaration: `public class demoJDBCclient` New declaration: `public class TheBrosClient`

5. In TheBrosClient.java, verify that the DEFINE VARIABLES SECTION of the program include the correct definitions:

   ```
   String driver = "com.mimer.jdbc.Driver";
   String protocol = "jdbc:mimer://SYSADM:foo@localhost:1360/";
   String dbName = "JB";
   ```

6. In the command window, compile the application using: `javac TheBrosClient.java`. A command prompt appears if the compilation is successful. The binary file TheBrosClient.class is created. If an error message appears, modify the line so that it is identical to the example.

7. Make sure that Mimer database instance ”JB” is running: Run Mimer administrator program form start menue and start JB database, if JB is not running.

8. run the client application: in the command window type

   ```
   java -classpath .;"C:\Program Files\Mimer SQL 10.0\mimjdbc3.jar"
   TheBrosClient
   ```

   Notice that the -classpath option sets the classpath to include the location of the file mimjdbc3.jar, therefore the path to mimjdbc.jar must be changed if Mimer is installed in a different directory. Important: Include the dot (.) at the beginning of the path so that your current working directory is included in the classpath as well. After executing the program, you must see the program output without any errors.

9. Now that you can run the demo program, it is the time to change it: In the TheBrosClient.java, following the comment ”// We create a table...”, which you can find halfway into the file, you will see the table called mydemo is created. You should change this to create the following Employee table:

   ```
   CREATE TABLE employee
   (number INTEGER CONSTRAINT pk_employee PRIMARY KEY,
    name VARCHAR(20),
    salary INTEGER,
    manager INTEGER,
    birthyear INTEGER,
    startyear INTEGER);
   ```

   Please also note that DDL (for example drop and creat commands in sql) and DML (for example insert,update and select commands in sql) statements can not be committed together. That is the reason why there are several commit points in the code.

10. Further down in TheBrosClient.java some rows are inserted into mydemo. Change this to insert these Employees instead:

    ```
    (157, ’Collins, Joanne’, 12000, 199, 1940, 1960)
    (1110, ’Hayes, Evelyn’, 16000, 33, 1952, 1973)
    (35, ’Evans, Michael’, 5000, 32, 1952, 1974)
    ```

11. Further down in TheBrosClient.java, the contents of mydemo table are printed. Change it to to retrieve the identity number, the name and the salary of the employees at Jonson Brothers. Print out the information.
12. In TheBrosClient.java, in a similar way to table creation section, update the salary for one or several employees by your own criterion. Then use a query to extract the identity number, the name and the salary of the employees at Jonson Brothers again and print it out.

You may have noticed that the client program does not shut down the database. This is because the database is a shared resource in a client/server environment and, in most cases, should be shut down only when the Network Server is shut down. If multiple clients are accessing the database and one client shuts down the database, the remaining clients will encounter a failure the next time they attempt an SQL command.

2.4 Examination of part 2

Hand in the source code of your Java program to your appointed assistant and also include a printout from executing the program.

References

[EN10] Elmasri, R. and Navathe, S. B.: Fundamentals of Databases, 6th Edition, Addison-Wesley, 2010 (available e.g. at Akademibokhandeln).

[PMR05] Padron-McCarthy, T. and Risch, T.: Databasteknik, Studentlitteratur, 2005 (available e.g. at Akademibokhandeln).

[M10] Mimer documentation, Version 9.2 (html) (include User's Manual, Reference Manual and Programmer's Manual), `http://developer.mimer.com/documentation/html_92/Mimer_SQL_Engine_DocSet/Mimer_SQL_Engine.htm`.