Universidad Autónoma de Madrid

Departamento de Informática

Data Structures
1st Assignment

Roberto Marabini Ruiz

Contents

1 Goals

In this assignment we will design, develop and query a database containing scale models of classic cars. The database will be managed using PostgreSQL and the queries will be written using the SQL language.

2 Explore the database

We will use the public domain database created by the 'MySQL-tutorial' (`https://www.mysqltutorial.org/mysql-sample-database.aspx`).

In *Moodle* you will find the classic models database (file `classicmodels.sql`) and a `Makefile`. A `Makefile` can be used not only to compile programs, but also to automate all sorts of tasks. In this assignment we will use it to create, delete and query the database. The provided `Makefile` defines the commands described in table 1.

command	**description**
createdb	create a database called `classicmodels`
dropdb	clean the database
dump	backup the database
restore	restore the database from a backup
shell	start the command line client *psql*
all	execute dropdb, createdb and restore

Table 1: List of the tasks defined in the Makefile file.

Create and populate the database using `make all`, explore the database using either a command line client (`make shell`) or a GUI (`pgAdmin4`) and include in your assignment report the following information:

1. Primary keys of each table.

2. Foreign keys of each table.

3. Database relational schema.

In order to answer the first two questions use the following nomenclature.

tableName(**attrbA**, attrbB, ...)
anotherTableName(**attrb1**, attrb2 → tableName.attrA, attr3, ...)
...

where *tableName* and *anotherTableName* are the table names. The primary keys are the attributes called *attrbA* in table *tableName*, and *attrb1* in table *anotherTableName*, respectively. The foreign key (*attrb2*) is linked by the symbol → to the referenced attribute (*attrbA* in table *tableName*).

The database relational schema should follow a model similar to the one shown in Figure 1 where both primary and foreign keys are explicitly marked (PK → primary key, FK → foreign key extranjera), and the referential integrity constraint lines (pointing from a foreign key to the primary key it refers to) are drawn.

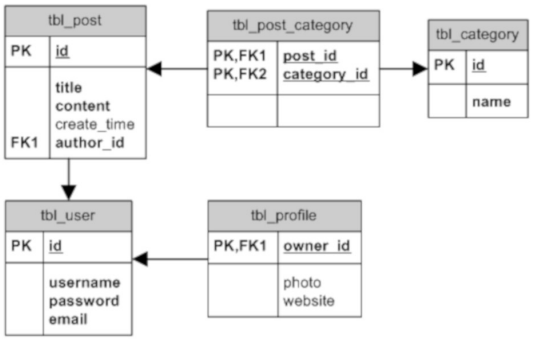

Figure 1: Relational diagram example.

3 Queries

Using SQL, implement the following queries (read subsection 3.1 before implementing them):

1. List the total amount of money paid by the customers that have ordered the "1940 Ford Pickup Truck" (the money may have been paid to purchase other items in addition to this one). Order the result by the amount of money paid. Use descending order. Each row should show customernumber, customername and total amount

2. Average time between order date and ship date for each product line. Each row should show: the productline and the corresponding average time

3. List the employees who report to those employees who report to the director. The director is the person who reports to no one. The output should show the "employeenumber" and the "lastname"

4. Office that sold most products (number of items). Note that in a single order many items, even of the same product, may be sold. The output should show the "officecode" and the number of items sold.

5. Countries with one (or more) offices that did not sell a single item during the year 2003. The output should show each country and the number of offices that did not sell a single item in the year 2003. Order the output by the number of offices. The country with higher number of "no selling offices" should be in the first line.

6. A common retail analytics task is to analyze the orders to learn what products are often purchased together. Report the names of pairs of products that appear together in more than one "shopping cart" (i.e. order). The output should show the IDs of both products and the number of shopping carts in which they appear. Note that given two products with IDs, id1 and Id2, you should not consider the pairs (Id1, Id2) and (Id2, Id1) as different.

3.1 Conventions to follow when implementing the queries

Makefile

We will use `make` to execute the queries. In this way, `make query1` should run the first query, `make query2` should run the second query, etc. In order to achieve this goal, create a file called `queryX.sql` (where X is the query number) that contains the query SQL code and add the following lines to the `Makefile` (repeat these actions for each query):

```
query1:
  @echo query-1: "Movies␣rented␣each␣year"
  @cat query1.sql | $(PSQL) | tee query1.log
```

Remember that in a `Makefile`, the spaces to the left of commands like "@echo" and "@cat" are tabs.

SQL Formatter

A good code must be readable both by the programmer who created it and by any other person who reviews it. Please write the queries consistently using indentation and line breaks to improve their readability. Before delivering the code, pass it through a formatter as the one available at `http://www.dpriver.com/pp/sqlformat.htm`. If you don't like it, look for another one that suits you and has similar functionality, and include the URL in the first line of the `Makefile` as a comment.

3.1.1 What requirements must satisfy a query?

1. It should be possible to execute it in the laboratory computers using the commands defined in the `makefile` and described in this assignment.

2. It should provide the correct result for **any** instance of the database.

3. The code is formatted in such a way that it is easy to read it. The output of the SQL formatted at `http://www.dpriver.com/pp/sqlformat.htm` is an example of code easy to be read.

If any of this items is not satisfied the grade of the query will be zero.

4 Database Redesign

The database schema used in this assignment has a few limitations. For example, if an employee moves from one office to another we loose track of the office in which he previously worked. Another problem is that the payments are not associated to an order and, finally, a customer cannot contact two different employees.

Redesign the database in order to fix these problem. Add to your report: (1) the new relational diagram and comment on the modifications, and (2) add a new option in the `makefile` so the command `makefile newdatabase` deletes all the tables in the database and recreates it with the new design. All SQL instructions needed to create the new tables should be stored in an auxiliary file called `newdatabase.sql`.

5 Deliverables to be uploaded to *Moodle*

Upload to *Moodle* a single file in zip format containing:

1. A single folder named **EDAT_pr1_gX_pY** containing the `makefile` file and all the auxiliary files needed to create, delete, populate and query the database. Here, **X** is your group, and **Y** is your team number. IMPORTANT: **do NOT include the file** `classicmodels.sql`.

2. Assignment report in pdf format. The report must contain the database diagram and answer all questions posed in this assignment. For each query, include the code and a brief comment regarding the implementation. Do not forget to answer the questions raised in section 4

6 Grading Criteria

5 points if the following criteria are met:

- The relational diagram and schema must be correct (they should have no more than 3 errors).
- At least 3 queries must be correct as defined in section 3.1.1.

[5.0-6.9] points if the following criteria are met:

- The criteria listed in the previous paragraphs are fully satisfied.
- At least 5 queries must be correct as defined in section 3.1.1.

[7.0-7.9] points if the following criteria are met:

- The criteria listed in the previous paragraphs are fully satisfied.
- The relational diagram and schema are fully correct. (No mistakes allowed)
- You have uploaded a report that demonstrates your understanding of the assignment.

[8.0-8.9] points if the following criteria are met:

- The criteria listed in the previous paragraphs are fully satisfied.
- All queries must be correct as defined in section 3.1.1.

[9.0-10.] points if the following criteria are met:

- The criteria listed in the previous paragraphs are fully satisfied.
- You have redesigned the database as required in subsection 4.

Late upload of the zip file with the assignment decreases the final mark by one point per day.