

UNIVERSIDAD AUTÓNOMA DE MADRID

DEPARTAMENTO DE INFORMÁTICA

Data Structure

1st Assignment

Roberto MARABINI RUIZ

Contents

1	Goals	2
2	Explore the data base	2
3	Queries	4
3.1	Conventions to follow when implementing the Queries	5
3.1.1	What requirements must satisfy a query?	5
4	Data Base Redesign	6
5	Deliverables to be uploaded to <i>Moodle</i>	6
6	Grading Criteria	7

1 Goals

In this assignment we will design, develop and query a DVD rental database. The data base will be managed using PostgreSQL and the queries will be written using the SQL language.

2 Explore the data base

We will use the public domain database created by the “PostgreSQL-tutorial” organization (see URL <http://www.postgresqltutorial.com/postgresql-sample-database/> for details).

From *Moodle* you may download the file `dvdrental.sql` that contains the DVD rental database. In order to restore the database from the `dvdrental.sql` we will use the `make` command which can be used, in addition to compile programs, to automatize tasks. In this assignment we will use it to create, delete and query the database. In the `Makefile` available in *Moodle*, we have define the commands described in table 1.

command	description
<code>createdb</code>	create a database called <i>dvdrental</i>
<code>dropdb</code>	clean the database
<code>dump</code>	backup the database
<code>restore</code>	restore the database from a backup
<code>shell</code>	start the command line client <i>psql</i>
<code>all</code>	execute <code>dropdb</code> , <code>createdb</code> and <code>restore</code>

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 (`pgmanager`) and include in your assignment report the following information:

1. Primary keys.

2. Foreign Keys.
3. Database relational schema
- ...

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

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

where *tableName* and *anotherTableName* are the table names. The primary keys are the attributes called *attrbA* -table *tableName*- and *attrb1* -table *anotherTableName*- respectively. The foreign keys (*attrb2*) are linked by the symbol \rightarrow to the referenced attribute (*attrA* 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, and the referential integrity constraint lines (pointing from a foreign key to the primary key it refers to) are drawn.

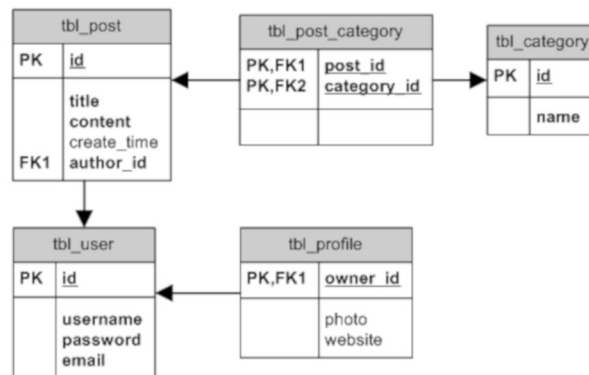


Figure 1: Relational diagram example.

3 Queries

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

1. Number of movies rented each year. The query must return two attributes containing the year and the number of films rented. Order the answer by the rental year in ascending order. Hint: the year can be extracted from a “date” variable using the function `extract`.
2. Client who has rented more movies. If there is a tie between several clients, all clients with the maximum number of rented movies must appear in the response. The query must return the attributes: `customer_id`, `first_name`, `last_name` and the number of rented movies.
3. List the cities where movies, in which “Bob Fawcett” appears, have been rented. Each city must appear just **once**. Sort the output alphabetically by the city name from A to Z. The query must return the `city_id` and the `city.name`.
4. Language in which most of the films have been filmed. The query must return the `language.name` attribute. If there is a tie between several languages, all languages in which the greatest number of films have been filmed should appear in the answer.
5. Language (of the films) in which a greater number of rentals has been done. The query must return the `language.name` attribute. If there is a tie between several languages, all languages in which a greater number of rentals has been done must appear in the answer.
6. Favorite category (`category.name`) of the customer who has rented more movies. By favorite category we refer to the category in which the client has made more rentals. If a client rents the same movie twice it should count as two rentals. The query must return the name (`category.name`) and the identifier (`category_id`) of the category. If there is a tie between several clients, all clients who have rented more films should appear in the response.

3.1 Conventions to follow when implementing the Queries

Makefile

We will use `make` to execute the queries. In this way, `make query1` shown run the first query, `make query2` should run the second query and so on and so forth. 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 at the right side 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 indentations 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 data base.

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 Data Base Redesign

The database schema used in this assignment presents some limitations. If a staff member is transferred from one store to another no record is kept about the stores in which he has worked before. Modify the database so this information is kept. That is, for any given staff member we want to keep records of all the stores in which he has worked.

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 recreate 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. `makefile` file and all the auxiliary files needed to create, delete, populate and query the database. **IMPORTANT: do NOT include the file `dv-drental.sql`.**
2. Assignment report in pdf format. The report must contain the database diagram and answer to all questions posed in this assignment. For each query include de 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 2 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 redesign 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.