EECS-3421, Winter 2017
Project I
Due March 19 at 12PM

SQL Jeopardy

Querying the York River Bookseller Database

Overview

In this project, you will work with an existing database, the *York River Bookseller's Database* (YRB DB). Actually, each will install his or her own private copy of the database on which to work. For *SQL Jeopardy*, you are to implement a number of SQL queries over the YRB DB.

There are 10 queries that you are to write. Each is worth 2 points for a total of 20 points for **Project I**. I am the manager and I provide you with English specifications for the queries that I want answered from the database. You are providing SQL implementations of these queries for YRB DB to do the job.

Each query is given a name for bookkeeping. You have successfully implemented the SQL query if you get the *same* result when run in DB2 on YRB DB as the correct query does. (And, of course, you have not cheated by making a simple query on purpose that manages to print the same results!) Otherwise, your query is unsuccessful. Grading per query is *all-or-nothing*. Either your query produces the right results, or it does not. You should make sure that your output of a query is exactly in the specified format. **If the format of the output is incorrect, you will get 0 points even if the query is otherwise correct**.

Some of the queries are straightforward, and some are difficult. Do not get too discouraged if you are unable to do a few.

Honesty: For projects, you are permitted to confer with others, seek advice, and (to a reasonable extent) help. However, remember that copying someone else's queries and claiming them as your own work is plagiarism. You must do your own work.

The YRB Database

Two scripts are provided:

- `yrb-create` and
- `yrb-drop`.

Note: if running the script returns an error, try another version of the same script: `yrb-create`.

The script `yrb-create` will create the YRB DB schema for you. It will also populate the tables with mock data. The script `yrb-drop` is provided for convenience. It will drop your copy of YRB DB from your DB2 schema space. If you mess things up, you can always drop YRB DB and then re-create it easily.

To create the YRB DB in your DB2 schema space, copy the script into your home directory and then issue the commands:

- `% setenv DB2DBDFT c3421m`
- `% db2 -tf yrb-create`

Read the schema definition in `yrb-create` for YRB DB to understand the design and what the YRB DB is about. Also, please read the instructions on how to use DB2 before you ask questions.

The Queries

General comment: do not remove duplicates unless specifically instructed to do so.

one

Which categories have names starting with 'h'

Show the category.

Answer.

Hint: See the SQL operator `like` in the textbook.

two

For each customer, find the largest number of purchases from a single club. Do not show customers who never bought any books.

Show the name of the customer and the number.

Answer.

three

Find the oldest book published for each category

Show the title, year and category.

Answer.

four

Select distinct titles which are not offered by 'Readers Digest' club.

Show the title.

Answer.

five

Select distinct book titles offered by all clubs.

Show the title.

Answer.

six

Select customers with no purchases.

Show all columns.

Answer.

seven

Select distinct names and descriptions of clubs from which books have been purchased at the most expensive price available for a particular title (that is, a particular book).

Show the club and description.

Answer.

eight

Select average price of the cheapest books offered in each category. (Find the cheapest books in categories and than find the average for this set). Round the amount to 2 decimal places.

Show the avg_cost cast as decimal(5,2).

Answer.

nine

Calculate the average number of days between purchases for each customer (If a customer purchased books only once, do not return him). Count multiple purchases in one day as one purchase.

Show the cid and avg_gap (cast as decimal (7,2)).

Answer.

Hint: *The attribute `when` is of type `timestamp`. This is different than type `date`. So to use it say in a `WHERE` condition to compare it with, say, '1998-1-1'. you will need to cast it to type `date`: `cast(when as date)`. Likewise, `'1998-1-1'` is a string! So for DB2 to consider it as a date, you must cast it as well: `cast('1998-1-1' as date)`.*

ten

Find the book which is ranked 22nd (from heaviest to lightests) in weight among all books (for simplicity, assume there is just one such book).

Show title and year.

Answer.

Deliverables

For *each* query, write your query in a file with the corresponding name as follows (`one.sql`, `two.sql`, etc.).

Do not include anything else in the files (in particular, the answers generated by each of your queries). **Do not end your queries with a ";".** We will be testing your queries by running in DB2 the content of each file. If the file contains anything that is not recognizable by DB2, your query will fail and you will get no credit for it.

When you are finished, use the `submit` command to turn in your work. The course is 3421 (ignore the section number for this submission); the directory is `a2`.

Let us know if you have problems submitting the files. **If you do not submit the files electronically, you will get no credit for this project**. There will be over 1,000 queries to verify and we cannot possibly do it by hand.

Hints and DB2 Comments

DB2 SQL provides a useful clause called `with`. It is easiest to show it's use by example. Here is an SQL query for DB2 to find the purchases for which folks are owed refunds, as related to what you are doing in your refund APP. Query **refund**:

```
with
    best (cid, title, year, lowest) as
        (select distinct M.cid, O.title, O.year, min(price)
             from yrb_member M, yrb_purchase P, yrb_offer O
             where M.club = O.club and
                   M.cid = P.cid and
                   P.title = O.title and P.year = O.year
             group by M.cid, O.title, O.year)
select C.name, P.title, P.year, qnty, price, lowest
    from yrb_customer C, yrb_purchase P, Best B, yrb_offer O
    where P.cid = B.cid and P.title = B.title and
          P.year = B.year and P.title = O.title and
          P.year = O.year and P.club = O.club and
          C.cid = P.cid and
          O.price > B.lowest
    order by C.name, P.title, P.year;
```

One may have more tables in the `with` clause, if need be:

```
with
    first (...) as
        (...),
    second (...) as
        (...),
    third (...) as
        (...)
select ...
    ...;
```

Note that you can use the (temporary) view `first` inside the definition for `second`, and the views `first` and `second` inside the definition for `third`, and so forth.

DB2 is picky about nested `SELECTS` always being named via `AS`. For example,

```
select ...
    from (select ...) as A,
         (select ...) as B,
         ...
    ...;
```

Even if you do not need the alias names `A` or `B` later on, give the sub-queries (nested `SELECTS`) names anyway. Otherwise, DB2 will complain.