CS 325 - Homework 3

Deadline

11:59 pm on **Friday, September 30, 2022**.

Purpose

To get more practice writing SQL `select` statements, and to practice with database modeling (creating an E-R diagram meeting the required notation for this course)

How to submit

-Part 1 will be completed on the course Canvas site.

-Part 2 will be submitted on nrs-projects.

-Part 3's E-R diagram will be submitted to Canvas.

For Part 2:

Each time you wish to submit Part 2, within the directory `325hw3` on nrs-projects.humboldt.edu (and at the nrs-projects UNIX prompt, **NOT inside** `sqlplus`!) type:

```
~ma548/325submit
```

...to submit your current files, using a homework number of 65.

(**Make sure** that the files you intend to submit are listed as having been submitted!)

Additional notes:

- You are required to use the HSU Oracle `student` database for **Part2** of this homework.
- **DB Reading Packet 4** and **SQL Reading Packet 3**, on the course Canvas site are useful references for this homework.
- Feel free to add additional `prompt` commands to your SQL scripts as desired to enhance the readability of the resulting output.
- You are expected to follow **course style standards** for entity-relationship diagrams and SQL `select` statements.

Part 1

Correctly complete the "HW 3 – Part 1 - Reading Questions for DB Reading Packet 3 and 4 - Entity-Relationship Modeling, Part 1", on the course Canvas site.

Setup for Part 2

Use `ssh` to connect to `nrs-projects.humboldt.edu`, and create, protect, and go to a directory named `325hw3` on nrs-projects:

```
mkdir 325hw3
chmod 700 325hw3
cd 325hw3
```

Put all of your files for Part 2 in this directory. (And it is from this directory that you should type `~ma548/325submit` each time you would like to submit your files for Part 2.)

Part 2

YOU ARE USING ORACLE and SQL FOR THIS PROBLEM.

On the course canvas site, along with this assignment, you will find a SQL script `movies-create.sql`. It creates a collection of tables that can be described in relation structure form as:

```
Movie_category(CATEGORY_CODE, category_name)

Client(CLIENT_NUM, client_lname, client_fname, client_phone,
       client_credit_rtg, client_fave_cat)
   foreign key (client_fave_cat) references movie_category(category_code)

Movie(MOVIE_NUM, movie_title, movie_director_lname, movie_yr_released,
      movie_rating, category_code)
   foreign key(category_code) references movie_category

Video(VID_ID, vid_format, vid_purchase_date, vid_rental_price, movie_num)
   foreign key (movie_num) references movie

Rental(RENTAL_NUM, client_num, vid_id, date_out, date_due, date_returned)
   foreign key (client_num) references client
   foreign key(vid_id) references video
```

(Think of it as a quaint, historical scenario... 8-/)

For your convenience and reference, a handout of these relation structures is posted along with this homework handout.

And, SQL script `movies-pop.sql` initially populates these tables (because it is better style to separate table creation and initial table population).

In your `325hw3` directory on nrs-projects, create a copy of the scripts `movies-create.sql` and `movies-pop.sql`, either by pasting them from the public course web page, or by using these commands at the nrs-projects UNIX prompt while in your directory `325hw3`:

```
cp  ~ma548/movies-create.sql  .    # don't forget the space and period!
cp  ~ma548/movies-pop.sql     .    # don't forget the space and period!
```

Run these SQL scripts `movies-create.sql` and `movies-pop.sql` in `sqlplus`; these 5 tables should be created, and then initially populated. Look them over; check out their contents.

Then, use `nano` (or `vi` or `emacs`) to create a file named `325hw3.sql` within directory `325hw3`:

```
nano 325hw3.sql
```

While within `nano` (or whatever), type in the following within one or more SQL **comments**:

- your name
- `CS 325 - Homework 3 - Part 2`
- the date this file was last modified

Then:

- use `spool` to start writing the results for this script's actions into a file `325hw3-out.txt`
- put in a `prompt` command printing `Homework 3 Part 2`
- put in a `prompt` command printing your name
- include a `spool off` command, at the BOTTOM/END of this file. Type your answers to the problems below BEFORE this `spool off` command!

NOTE!!! READ THIS!!!

Now, within your file `325hw3.sql`, add in SQL statements for the following, **PRECEDING** EACH with a SQL*Plus `prompt` command noting what problem part it is for.

Part 2-A

Perform a **relational selection** of rows of the `client` table for clients with a client credit rating that is higher than 3.4.

Part 2-B

Perform a **"pure"/"true" relational projection** of the movie rating and year released columns from the `movie` table.

Part 2-C

Perform an **equi-join** of the `client` and `movie_category` tables. (Consider: what would be a suitable join condition for an equi-join between these two tables? Which attribute in each has the same domain?)

Part 2-D

Note that, in this scenario, a rental's date returned attribute is empty/null until the rented video has been returned.

Project just the rented video IDs, date out, and date due for rentals that have **not** yet been returned.

Part 2-E

Project just the video IDs, video formats, and rental prices for videos that do **not** have the format Blu-Ray.

Part 2-F

Project just the movie category NAME (**not** the movie category code!), client's last name, and client's credit rating from the equi-join of the `movie_category` and `client` tables.

Part 2-G

Perform a relational selection of videos with a purchase date between July 15, 2008 and December 1, 2011, inclusive. For full credit, appropriately use the `between` operator in your query.

Part 2-H

Perform a relational selection of videos which have a rental price greater than or equal to $3.99 and have a purchase date on or after January 1, 2011.

Part 2-I

Project only the movie title and the movie rating for all movies containing the string `'the'` anywhere in their title (where the case IS significant -- only all-lowercase `'the'` is desired). For full credit, appropriately use the `like` operator in your query.

Part 2-J

Project only the movie rating and movie title for all movies with ratings of PG-13 or R or A only. For full credit, appropriately use the `in` operator in your query.

Submit your files `325hw3.sql` and `325hw3-out.txt`. Using a Homework number of **65**