Part 1: Reading Assignment. Read the following sections in the textbook. No reading questions assigned this week.

- Ch. 3: 3.8

Part 2: More Database Queries. Use the database (cpsc321) that has been installed for the course to answer the following questions. Within the cpsc321 database are a number of tables that have been created and populated to represent simple movie rental data. For this assignment, you’ll use the following tables in the database:

- Actor(actor_id, first_name, last_name, last_update)
- Category(category_id, name, last_update)
- Customer(customer_id, store_id, first_name, last_name, email, address_id, active, create_date, last_update)
- Film(film_id, title, description, release_year, language_id, original_language_id, rental_duration, rental_rate, length, replacement_cost, rating, special_features, last_update)
- Film_Actor(actor_id, film_id, last_update)
- Film_Category(film_id, category_id, last_update)
- Inventory(inventory_id, film_id, store_id, last_update)
- Payment(payment_id, customer_id, staff_id, rental_id, amount, payment_date, last_update)
- Rental(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id, last_update)
- Staff(staff_id, first_name, last_name, address_id, picture, email, store_id, active, username, password, last_update)
- Store(store_id, manager_staff_id, address_id, last_update)

For each question below, write a single SQL query to answer the question and show the result of the query answer against the above database. If the answer has more than 10 rows, you only need to show the first 10 rows (however, your query should work generally over the instance).

1. Find the total number of films by category ordered from most to least. Give the name of each category along with the number of corresponding films.

2. Find the number of films acted in by each actor ordered from highest number of films to lowest. For each actor, give their first and last name and the number of films they acted in.

3. For each ‘PG’ rated films that costs 2.99 to rent find the number of times it has been rented. The result should be sorted from most rented to least rented. For each film, give the film title and the corresponding number of rentals.

4. Find all first and last names of customers that have rented at least six ‘R’ rated films costing 0.99 each to rent. For each customer give the number of such films they’ve rented.
5. Find the total sales (of payments) for each film category sorted from highest total payments to least. Give the name of each category and the total payments.

6. Find the film category (or categories if there is a tie) with the most number of ‘R’ rated films. Your query cannot use limit. Return the category name and the corresponding number of R rated films.

7. Find the ‘G’ rated film (or films if there is a tie) that have been rented the most number of times. You cannot use limit and your query must return only the film(s) rented the most number of times (not the second, third, etc). Return the film id, title, and the number of times the film has been rented.

8. Find the store (or stores if there is a tie) that have the most rentals. You cannot use limit and your query must return only the store(s) that have the most rentals (not the second most, third most, etc). Return the store id and the number of store rentals.

9. For each staff member, find the movies they rented for 0.99 and the total number of times that they were rented to customers by the staff member. Return the first and last name of each staff member, the title of each movie, and the number of times each movie was rented by the staff member. The results should be ordered by staff member last name followed by first name. For each staff member, the movies that they have rented should be ordered from most rented to least rented.

10. Come up with your own “interesting” query over the database schema. Your query should involve group by, having, and (necessary) subqueries. Give your query, the query result, and explain in plain English what the purpose of the query is.

Turn in a hard-copy print out of the cover sheet, your reflection, a script containing the queries answering the above questions, and the result of running your script (queries) against the cpsc321 database. In addition, submit your script to the online class submission page by the due date.