**DBAS 1007 – Assignment 2 - Basic SQL Queries   
Value:** 10% of your overall mark.

**Instructions:**Create a single script file that will contain SQL statements to perform the following requirements against the **Sakila** database. Read the instructions carefully to ensure that you understand the requirements for each query. If you do not understand a question, it is your responsibility to clarify the requirements with your instructor.  
Be sure to attempt all questions as you may be awarded partial marks for your work.  
When you are finished, submit your script file on BrightSpace as **[YourName]\_DBAS\_Assignment2.sql**.

**In your script, please remember to clearly label your solution statement for each question with a comment indicating for which question it is the solution. Example:**

**#Query 1**

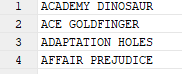
**SELECT…**

**Query Requirements, using the Sakila database:**

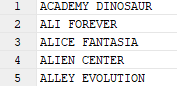
1. Create a query that returns all film titles in alphabetical order for the language with ID = 1.
2. Create a query that will list all the film titles containing the word “Drama” in the description. (You need only to accommodate the displayed spelling of the word.)
3. 'Create a query that lists all films with the ratings 'G', 'PG, and 'PG-13' and also do not have an original language ID. Build the query to check for all conditions, regardless of the data. Sort the results alphabetically by rating, then title.
4. Create a query that lists all payment ids for amounts less than $2.99 and after the June 1st, 2005. Sort the results by payment date, then amount.
5. Create a query that lists all cities who have 'Canada' as a country. Build the query as if you do not know Canada's country id. Hint: Remember you can have selects within select queries. Sort the results reverse alphabetically by city name.
6. Create a query that returns all films that either start with the letter ‘A’, or end with the letter ‘t’. Suppress any duplicate film names and sort the results in reverse alphabetical order.
7. Create a query that returns each customer ID and the count of payments they have made. Make sure you call the second column "count". Sort results by count in reverse order.
8. Create a query that lists the CustomerId and maximum paid by each customer. Then modify the query to display only those records whose amount was more than $5.99. Sort from highest total to lowest total, then sort by CustomerId from lowest to highest.
9. Create a query that returns the CustomerId and the total number of items for each rental. Sort the results from the most number of items rented to least number of items rented. **Note**: this is using a different table from the previous two queries.
10. Create a query that returns all customers who have payments with a total over $7.00 and were served by the staff with an ID of 1. Sort the results by customer ID from highest to lowest, then amount lowest to highest.
11. Create a query that lists the CategoryId and count of films (name this column Cat\_Count). Restrict your results to only those categories with more than 60 films. Sort the results by the Category Count in descending order.
12. Create a query that returns the names of the films with Actor ID 107. Sort the movie titles alphabetically.
13. Create a query that only lists the actor ids from actors that appear in more than 36 films. Sort from largest to smallest amount of films per actor. Rename columns as indicated.
14. Create a query that displays the total of films are not in inventory. Call the single column "Total Not In Inventory".
15. Create a query that lists all of the film ids for any movies that contain the actor with the title "RIP CRAWFORD". Do not use the actor's ID in the query. Display Film IDs in Ascending order.

**Snapshots of Expected Results**

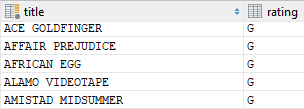
**Question 1**

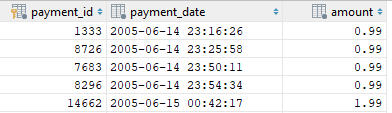


**Question 2** (The order of your results may be different but the titles should be the same)

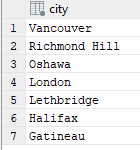


**Question 3**

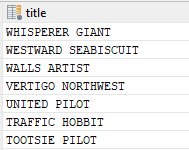
  
**Question 4**



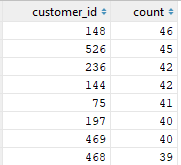
**Question 5**



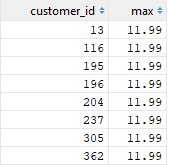
**Question 6**



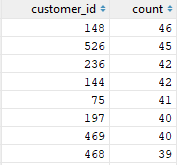
**Question 7**



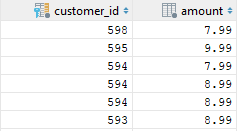
**Question 8**



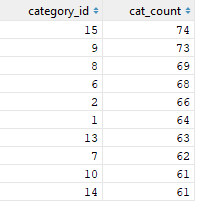
**Question 9**



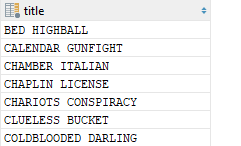
**Question 10**



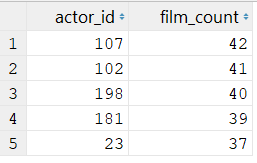
**Question 11**



**Question 12**



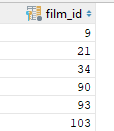
**Question 13**



**Question 14**



**Question 15**



## Rubric

|  |  |  |
| --- | --- | --- |
| Questions 1-15 | **Yes \_\_\_\_\_\_\_ No \_\_\_\_\_\_\_ Partial \_\_\_\_\_\_\_** | 2 points each, 1 point for partial |
|  | **Total** | 30 points |