## **SQL Homework Assignment**

**Link to Google Docs for these instructions:**

https://docs.google.com/document/d/1beW3kBnymvUjBdQ\_7ir8X9BjL5CfXPccFe\_BIRIQGuE/edit#

Pavan is happy to do a 1 on 1 with you to go over anything in particular. Please message him on slack

Just in case: Pavan’s cell phone: 732-236-8447

Use the pagilla database to do this.

But Pavan I lost the pagilla database!!!!!!

You can download the sql files and run them here.

<https://drive.google.com/open?id=0Bz0Wzew04n0ubnBTc3YyaUxaYzQ>

To submit you can make

1. A google doc and click share and share public - then submit that google doc
2. Better option: make a github repo, put the sql files for pagilla in it. Make a README.md file and put all the queries and questions in there formated nicely
   1. and give instructions on how to populate the database.
      1. To create this database
         1. Open up psql in the directory of these files and run these commands
         2. \i …
         3. \i …
         4. \i ...

* 1a. You need a list of all the actors’ first name and last name
* 1b. Display the first and last name of each actor in a single column in upper case letters. Name the column Actor Name
* 2a. You need to find the id, first name, and last name of an actor, of whom you know only the first name of "Joe." What is one query would you use to obtain this information?
* 2b. Find all actors whose last name contain the letters GEN. Make this case insensitive
* 2c. Find all actors whose last names contain the letters LI. This time, order the rows by last name and first name, in that order. Make this case insensitive.
* 2d. Using IN, display the country\_id and country columns of the following countries: Afghanistan, Bangladesh, and China:
* 3a. Add a middle\_name column to the table actor. Specify the appropriate column type
* 3b. You realize that some of these actors have tremendously long last names. Change the data type of the middle\_name column to something that can hold more than varchar.

Now we didn’t explicitly cover this in class so exercise your googling skills.

* 3c. Now write a query that would remove the middle\_name column.
* 4a. List the last names of actors, as well as how many actors have that last name.
* 4b. List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors
* 4c. Oh, no! The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.
* 4d. Perhaps we were too hasty in changing GROUCHO to HARPO. It turns out that GROUCHO was the correct name after all!

In a single query,

if the first name of the actor is currently HARPO, change it to GROUCHO.

Otherwise, change the first name to MUCHO GROUCHO, as that is exactly what the actor will be with the grievous error.

BE CAREFUL NOT TO CHANGE THE FIRST NAME OF EVERY ACTOR TO MUCHO GROUCHO

(Hint: update the record using a unique identifier.)

* 5a. You’re answering each of these questions. Please include examples and answer them in plain english.
  + What’s the difference between a left join and a right join.
  + What about an inner join and an outer join?
  + When would you use rank?
  + What about dense\_rank?
  + When would you use a subquery in a select?
  + When would you use a right join?
  + When would you use an inner join over an outer join?
  + What’s the difference between a left outer and a left join
  + When would you use a group by?
  + Describe how you would do data reformatting
  + When would you use a with clause?
* 6a. Use a JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:
* 6b. Use a JOIN to display the total amount rung up by each staff member in January of 2007. Use tables staff and payment.

You’ll have to google for this one, we didn’t cover it explicitly in class.

* 6c. List each film and the number of actors who are listed for that film. Use tables film\_actor and film. Use inner join.
* 6d. How many copies of the film Hunchback Impossible exist in the inventory system?
* 6e. Using the tables payment and customer and the JOIN command, list the total paid by each customer. List the customers alphabetically by last name:
* 7a. The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters K and Q have also soared in popularity. display the titles of movies starting with the letters K and Q whose language is English.
* 7b. Use subqueries to display all actors who appear in the film Alone Trip.
* 7c. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all Canadian customers. Use joins to retrieve this information.
* 7d. Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as a family film.

Now we mentioned family film, but there is no family film category. There’s a category that resembles that. In the real world nothing will be exact.

* 7e. Display the most frequently rented movies in descending order.
* 7f. Write a query to display how much business, in dollars, each store brought in.
* 7g. Write a query to display for each store its store ID, city, and country.
* 7h. List the top five genres in gross revenue in descending order.
* 8a. In your new role as an executive, you would like to have an easy way of viewing the Top five genres by gross revenue. Use the solution from the problem above to create a view.
* 8b. How would you display the view that you created in 8a?
* 8c. You find that you no longer need the view top\_five\_genres. Write a query to delete it.

### **Appendix: List of Tables in the Pagilla DB**

* A schema is also available as sakila\_schema.svg. Open it with a browser to view.

'actor'  
 'actor\_info'  
 'address'  
 'category'  
 'city'  
 'country'  
 'customer'  
 'customer\_list'  
 'film'  
 'film\_actor'  
 'film\_category'  
 'film\_list'  
 'film\_text'  
 'inventory'  
 'language'  
 'nicer\_but\_slower\_film\_list'  
 'payment'  
 'rental'  
 'sales\_by\_film\_category'  
 'sales\_by\_store'  
 'staff'  
 'staff\_list'  
 'store'