**Part 1**

Each of the below queries will include at least one Join. Read carefully and be sure to think about which columns you can use to Join the necessary tables.

Run the following queries:

1. Write a query to find the first and last name, customer ID and rental ID for customers who have rented a film.
2. Write a query that finds all films with actors that have an actor\_id 5.
3. Write a query that lists out all information of every film along with the name of the language for each film, even if a language doesn't exist for that film.
4. Write a query that lists out the title of films and the name of the actors who starred in those films. Additionally, only list films that starred artists whose first names start with a vowel.

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Graphical user interface, text, table

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

## Part 2

Below is a real-life scenario. Please read this scenario and run the appropriate queries needed.

You have just been hired as a Data Analyst for a company that rents films to customers.

They would like an inventory list of films that were rented for more than $4.99.

### Tip!

Pay close attention to the column names and give them unique names if a column name is repeated.

Since the highest rental rate is 4.99 and it says “rented for more than 4.99” would mean greater not greater than or equal too than and would result in 0 results I went with “amount” since it would return results. The first doesn’t look like the math adds up, the customers don’t seem to keep to the rental\_duration and the company doesn’t seem to charge late fees or something idk…. I went through it by hand on a few and what they pay doesn’t not equal what they should have paid.

A picture containing table

Description automatically generated