**LógicaConsultasSQL**

1. Crea el esquema de la BBDD.

2. Muestra los nombres de todas las películas con una clasificación por

edades de ‘R’.

3. Encuentra los nombres de los actores que tengan un “actor\_id” entre 30

y 40.

4. Obtén las películas cuyo idioma coincide con el idioma original.

5. Ordena las películas por duración de forma ascendente.

6. Encuentra el nombre y apellido de los actores que tengan ‘Allen’ en su

apellido.

7. Encuentra la cantidad total de películas en cada clasificación de la tabla

“film” y muestra la clasificación junto con el recuento.

8. Encuentra el título de todas las películas que son ‘PG-13’ o tienen una

duración mayor a 3 horas en la tabla film.

9. Encuentra la variabilidad de lo que costaría reemplazar las películas.

10. Encuentra la mayor y menor duración de una película de nuestra BBDD.

11. Encuentra lo que costó el antepenúltimo alquiler ordenado por día.

12. Encuentra el título de las películas en la tabla “film” que no sean ni ‘NC-

17’ ni ‘G’ en cuanto a su clasificación.

13. Encuentra el promedio de duración de las películas para cada

clasificación de la tabla film y muestra la clasificación junto con el

promedio de duración.

14. Encuentra el título de todas las películas que tengan una duración mayor

a 180 minutos.

15. ¿Cuánto dinero ha generado en total la empresa?

16. Muestra los 10 clientes con mayor valor de id.

17. Encuentra el nombre y apellido de los actores que aparecen en la

película con título ‘Egg Igby’.

18. Selecciona todos los nombres de las películas únicos.

19. Encuentra el título de las películas que son comedias y tienen una

duración mayor a 180 minutos en la tabla “film”.

20. Encuentra las categorías de películas que tienen un promedio de

duración superior a 110 minutos y muestra el nombre de la categoría

junto con el promedio de duración.

21. ¿Cuál es la media de duración del alquiler de las películas?

22. Crea una columna con el nombre y apellidos de todos los actores y

actrices.

23. Números de alquiler por día, ordenados por cantidad de alquiler de

forma descendente.

24. Encuentra las películas con una duración superior al promedio.

25. Averigua el número de alquileres registrados por mes.

26. Encuentra el promedio, la desviación estándar y varianza del total

pagado.

27. ¿Qué películas se alquilan por encima del precio medio?

28. Muestra el id de los actores que hayan participado en más de 40

películas.

29. Obtener todas las películas y, si están disponibles en el inventario,

mostrar la cantidad disponible.

30. Obtener los actores y el número de películas en las que ha actuado.

31. Obtener todas las películas y mostrar los actores que han actuado en

ellas, incluso si algunas películas no tienen actores asociados.

32. Obtener todos los actores y mostrar las películas en las que han

actuado, incluso si algunos actores no han actuado en ninguna película.

33. Obtener todas las películas que tenemos y todos los registros de

alquiler.

34. Encuentra los 5 clientes que más dinero se hayan gastado con nosotros.

35. Selecciona todos los actores cuyo primer nombre es 'Johnny'.

36. Renombra la columna “first\_name” como Nombre y “last\_name” como

Apellido.

37. Encuentra el ID del actor más bajo y más alto en la tabla actor.

38. Cuenta cuántos actores hay en la tabla “actor”.

39. Selecciona todos los actores y ordénalos por apellido en orden

ascendente.

40. Selecciona las primeras 5 películas de la tabla “film”.

41. Agrupa los actores por su nombre y cuenta cuántos actores tienen el

mismo nombre. ¿Cuál es el nombre más repetido?

42. Encuentra todos los alquileres y los nombres de los clientes que los

realizaron.

43. Muestra todos los clientes y sus alquileres si existen, incluyendo

aquellos que no tienen alquileres.

44. Realiza un CROSS JOIN entre las tablas film y category. ¿Aporta valor

esta consulta? ¿Por qué? Deja después de la consulta la contestación.

45. Encuentra los actores que han participado en películas de la categoría

'Action'.

46. Encuentra todos los actores que no han participado en películas.

47. Selecciona el nombre de los actores y la cantidad de películas en las

que han participado.

48. Crea una vista llamada “actor\_num\_peliculas” que muestre los nombres

de los actores y el número de películas en las que han participado.

49. Calcula el número total de alquileres realizados por cada cliente.

50. Calcula la duración total de las películas en la categoría 'Action'.

51. Crea una tabla temporal llamada “cliente\_rentas\_temporal” para

almacenar el total de alquileres por cliente.

52. Crea una tabla temporal llamada “peliculas\_alquiladas” que almacene las

películas que han sido alquiladas al menos 10 veces.

53. Encuentra el título de las películas que han sido alquiladas por el cliente

con el nombre ‘Tammy Sanders’ y que aún no se han devuelto. Ordena

los resultados alfabéticamente por título de película.

54. Encuentra los nombres de los actores que han actuado en al menos una

película que pertenece a la categoría ‘Sci-Fi’. Ordena los resultados

alfabéticamente por apellido.

55. Encuentra el nombre y apellido de los actores que han actuado en

películas que se alquilaron después de que la película ‘Spartacus

Cheaper’ se alquilara por primera vez. Ordena los resultados

alfabéticamente por apellido.

56. Encuentra el nombre y apellido de los actores que no han actuado en

ninguna película de la categoría ‘Music’.

57. Encuentra el título de todas las películas que fueron alquiladas por más

de 8 días.

58. Encuentra el título de todas las películas que son de la misma categoría

que ‘Animation’.

59. Encuentra los nombres de las películas que tienen la misma duración

que la película con el título ‘Dancing Fever’. Ordena los resultados

alfabéticamente por título de película.

60. Encuentra los nombres de los clientes que han alquilado al menos 7

películas distintas. Ordena los resultados alfabéticamente por apellido.

61. Encuentra la cantidad total de películas alquiladas por categoría y

muestra el nombre de la categoría junto con el recuento de alquileres.

62. Encuentra el número de películas por categoría estrenadas en 2006.

63. Obtén todas las combinaciones posibles de trabajadores con las tiendas

que tenemos.

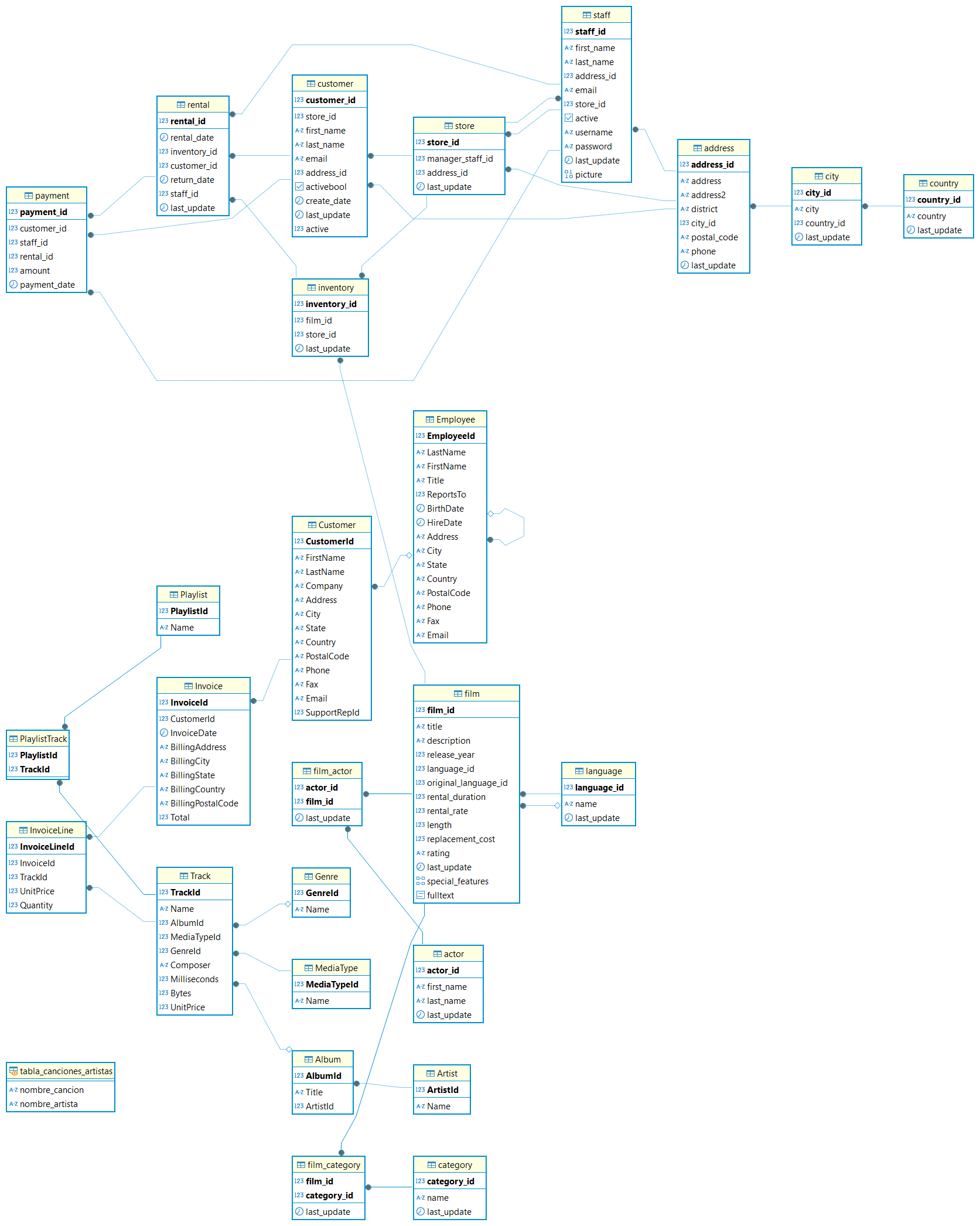
64. Encuentra la cantidad total de películas alquiladas por cada cliente y

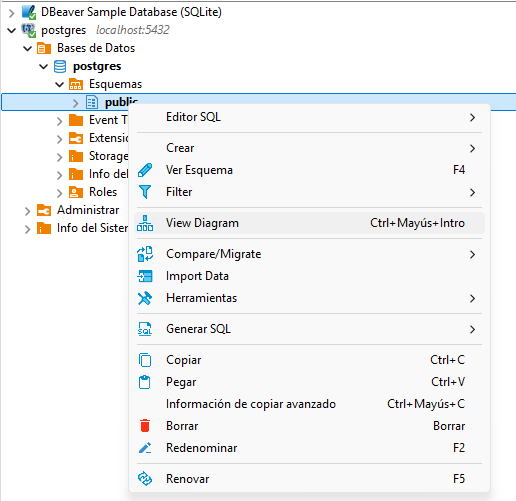
muestra el ID del cliente, su nombre y apellido junto con la cantidad de

películas alquiladas.

**Consulta 1**

Crea el esquema de la BBDD.





Postgres – Bases de datos – postgres – esquemas – botón derecho en public – view diagram.

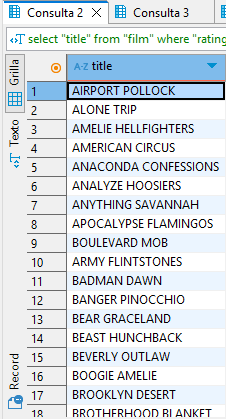
**Consulta 2**

Muestra los nombres de todas las películas con una clasificación por edades de ‘R’.

**select** **"title"**

**from** **"film"**

**where** **"rating"** = 'R';



Estas son las películas con una clasificación por edades de ‘R’ ordenadas por orden alfabético.

**Explicación de la consulta:**

1. **Seleccionar la columna**: Escribimos SELECT “title” porque queremos los nombres de las películas.
2. **Indicar la tabla**: FROM "film" para obtener datos de la tabla de películas.
3. **Filtrar por clasificación**: WHERE "rating" = 'R' selecciona solo las películas con esta clasificación.

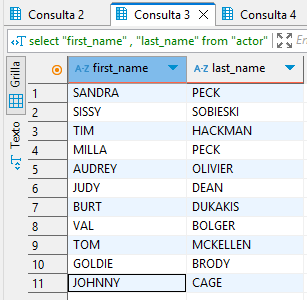
**Consulta 3**

Encuentra los nombres de los actores que tengan un “actor\_id” entre 30 y 40.

**select** **"first\_name"** , **"last\_name"**

**from** **"actor"**

**where** **"actor\_id"** **between** 30 **and** 40;



Nombres de los actores que tiene un “actor\_id” entre 30 y 40.

**Explicación de la consulta:**

1. **Seleccionar las columnas** SELECT “first\_name” y “last\_name” porque queremos los nombres de los actores y los escribimos separados por una coma.
2. **Indicar la tabla**: FROM "actor" porque ahí se encuentran los actores.
3. **Filtrar por rango**: WHERE "actor\_id" BETWEEN 30 AND 40 selecciona solo los actores dentro de ese rango.

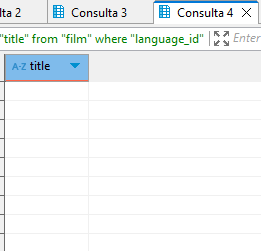
**Consulta 4**

Obtén las películas cuyo idioma coincide con el idioma original.

**select** **"title"**

**from** **"film"**

**where** **"language\_id"** = **"original\_language\_id"**;



No me sale nada, no sé si he realizado la consulta erróneamente.

**Explicación de la consulta:**

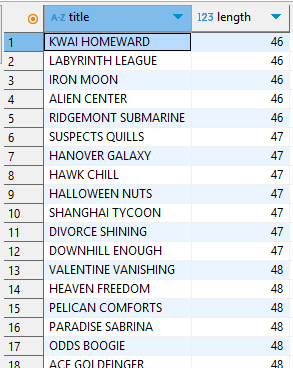
1. **Seleccionar la columna**: SELECT “title” porque queremos los títulos de las películas.
2. **Indicar la tabla**: FROM "film" porque ahí se encuentran los datos de las películas.
3. **Filtrar idioma**: WHERE "language\_id" = "original\_language\_id" para seleccionar solo las películas cuyo idioma coincide con el original.

**Consulta 5**

Ordena las películas por duración de forma ascendente.

**select** **"title"**, **"length"** **from** **"film"**

**order** **by** **"length"** **asc**;



Películas ordenadas por duración de manera ascendente.

**Explicación de la consulta:**

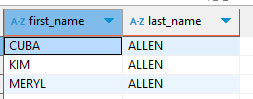
1. **Seleccionar las columnas**: SELECT “title” y “length” porque queremos ver la duración junto con el título separados por una coma.
2. **Indicar la tabla**: FROM "film" porque ahí están los datos que queremos ver.
3. **Ordenar por duración ascendente**: ORDER BY "length" ASC para ordenar de menor a mayor duración.

**Consulta 6**

Encuentra el nombre y apellido de los actores que tengan ‘Allen’ en su apellido.

**select** **"first\_name"**, **"last\_name"** **from** **"actor"**

**where** **"last\_name"** = 'ALLEN';



Nombres de los actores que tienen ‘Allen’ en su apellido.

**Explicación de la consulta:**

1. **Seleccionar las columnas**: SELECT “first\_name” y “last\_name” para obtener los nombres y apellidos de los actores separados por una coma.
2. **Indicar la tabla**: FROM "actor" porque contiene los datos de los actores.
3. **Filtrar por apellido**: WHERE "last\_name" = 'ALLEN' busca apellidos que contengan la palabra "ALLEN" en cualquier posición.

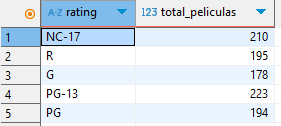
**Consulta 7**

Encuentra la cantidad total de películas en cada clasificación de la tabla “film” y muestra la clasificación junto con el recuento.

**select** **"rating"**, **count**(\*) **as** **"total\_peliculas"**

**from** **"film"**

**group** **by** **"rating"**;



Cantidad total de películas y clasificación de la tabla “film”.

**Explicación de la consulta:**

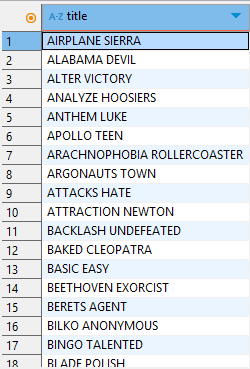
1. **Seleccionar la clasificación**: SELECT “rating” para agrupar las películas por clasificación.
2. **Contar las películas**: COUNT(\*) cuenta cuántas películas hay en cada clasificación. AS “total\_peliculas” para llamar a la columna de esta forma.
3. **Indicar la tabla**: FROM "film" porque contiene los datos de las películas.
4. **Agrupar por clasificación**: GROUP BY "rating" para calcular el total de películas en cada grupo.

**Consulta 8**

Encuentra el título de todas las películas que son ‘PG-13’ o tienen una duración mayor a 3 horas en la tabla film.

**select** **"title"** **from** **"film"**

**where** **"rating"** = 'PG-13' **or** **"length"** > 180;



Películas con clasificación ‘PG-13’ o tienen una duración mayor a 3 horas.

**Explicación de la consulta:**

1. **Seleccionar la columna**: SELECT “title” para obtener solo los títulos de las películas.
2. **Indicar la tabla**: FROM "film" porque ahí están los datos de las películas.
3. **Filtrar por condición**: WHERE "rating" = 'PG-13' OR "length" > 180 para incluir películas con clasificación PG-13 o con duración mayor a 180 minutos.

**Consulta 9**

Encuentra la variabilidad de lo que costaría reemplazar las películas.

**select** **variance**(**"replacement\_cost"**) **as** **"variabilidad\_reemplazo"**

**from** **"film"**;



Variabilidad de lo que costaría reemplazar las películas.

**Explicación de la consulta:**

1. **Seleccionar la varianza**: SELECT variance ("replacement\_cost") para calcular la dispersión de los costes de reemplazo.
2. **Indicar la tabla**: FROM "film" porque ahí están los datos.

**Consulta 10**

Encuentra la mayor y menor duración de una película de nuestra BBDD.

**select** **max**(**"length"**) **as** **"max\_duracion"**, **min**(**"length"**) **as** **"min\_duracion"**

**from** **"film"**;



Mayor y menor duración de las películas de la base de datos.

**Explicación de la consulta:**

1. **Seleccionar el mayor y menor valor**: SELECT max("length") obtiene la mayor duración, y min("length") la menor. As max duracion y min\_duracion para llamar así a la columna.
2. **Indicar la tabla**: FROM "film" porque ahí se encuentran los datos.

**Consulta 11**

Encuentra lo que costó el antepenúltimo alquiler ordenado por día.

**select** **"rental\_id"** **as** **"numero\_alquiler"**, **"amount"** **as** **"alquiler"**

**from** **"payment"**

**order** **by** **"payment\_date"** **desc**

**limit** 1 **offset** 2;



Coste del antepenúltimo alquiler del día.

**Explicación de la consulta:**

1. **Seleccionar las columnas**: SELECT "rental\_id" y "amount" para obtener el identificador del alquiler y el coste del alquiler.
2. **Indicar la tabla**: FROM "payment" porque ahí se registran los pagos de los alquileres.
3. **Ordenar por fecha de pago**: ORDER BY "payment\_date" DESC para ordenar los pagos del más reciente al más antiguo.
4. **Saltar los dos primeros resultados**: LIMIT 1 OFFSET 2 para obtener el tercer registro (antepenúltimo alquiler).

**Consulta 12**

Encuentra el título de las películas en la tabla “film” que no sean ni ‘NC-17’ ni ‘G’ en cuanto a su clasificación.

**select** **"title"**

**from** **"film"**

**where** **"rating"** **not** **in** ('NC-17', 'G');



Títulos de las películas que no son de la clasificación ni ‘NC-17’ ni ‘G’.

**Explicación de la consulta:**

1. **Seleccionar la columna**: SELECT "title" para obtener los títulos de las películas.
2. **Indicar la tabla**: FROM "film" porque ahí están los datos.
3. **Filtrar las clasificaciones**: WHERE "rating" NOT IN ('NC-17', 'G') para excluir las películas con estas clasificaciones.

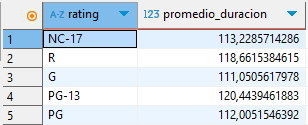
**Consulta 13**

Encuentra el promedio de duración de las películas para cada clasificación de la tabla film y muestra la clasificación junto con el promedio de duración.

**select** **"rating"**, **avg**(**"length"**) **as** **"promedio\_duracion"**

**from** **"film"**

**group** **by** **"rating"**;



Clasificación y promedio de duración de las películas.

**Explicación de la consulta:**

1. **Seleccionar la columna**: SELECT "rating" para obtener las clasificaciones de películas.
2. **Calcular el promedio**: AVG("length") obtiene la duración promedio de las películas en cada clasificación.
3. **Indicar la tabla**: FROM "film" porque queremos los datos de películas.
4. **Agrupar por clasificación**: GROUP BY "rating" para ordenar por clasificación.

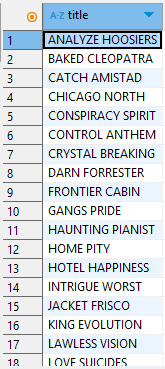
**Consulta 14**

Encuentra el título de todas las películas que tengan una duración mayor a 180 minutos.

**select** **"title"**

**from** **"film"**

**where** **"length"** > 180;



Películas cuya duración es mayor 180 minutos.

**Explicación de la consulta:**

1. **Seleccionar la columna**: SELECT "title" para obtener los títulos de las películas.
2. **Indicar la tabla**: FROM "film" porque queremos los datos de esta tabla.
3. **Filtrar por duración**: WHERE "length" > 180 para seleccionar solo las películas con duración mayor a 180 minutos.

**Consulta 15**

¿Cuánto dinero ha generado en total la empresa?

**select** **sum**(**"amount"**) **as** **"total\_ingresos"**

**from** **"payment"**;



Ingresos totales de la empresa.

**Explicación de la consulta:**

1. **Seleccionar la suma de ingresos**: SUM("amount") calcula el total de pagos recibidos. AS ”total\_ingresos” para que la columna se llame así.
2. **Indicar la tabla**: FROM "payment" porque ahí se registran los pagos.

**Consulta 16**

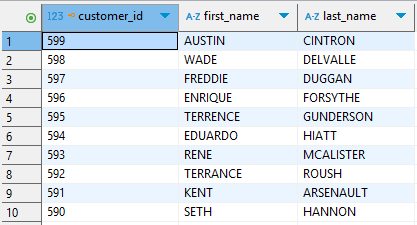
Muestra los 10 clientes con mayor valor de id.

**select** **"customer\_id"**, **"first\_name"**, **"last\_name"**

**from** **"customer"**

**order** **by** **"customer\_id"** **desc**

**limit** 10;



Los 10 clientes con mayor valor de id.

**Explicación de la consulta:**

1. **Seleccionar columnas**: SELECT "customer\_id", "first\_name" y "last\_name" para obtener el ID y nombre de los clientes.
2. **Indicar la tabla**: FROM "customer" porque ahí están los datos de los clientes.
3. **Ordenar en orden descendente**: ORDER BY "customer\_id" DESC para ordenar los clientes con el ID del más alto al primero.
4. **Limitar a 10 resultados**: LIMIT 10 para obtener solo los 10 clientes con el ID más alto.

**Consulta 17**

Encuentra el nombre y apellido de los actores que aparecen en la película con título ‘Egg Igby’.

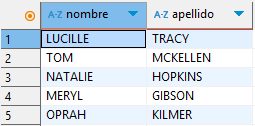
**select** **"actor"**.**"first\_name"** **as** **"nombre"**, **"actor"**.**"last\_name"** **as** **"apellido"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**join** **"film"** **on** **"film\_actor"**.**"film\_id"** = **"film"**.**"film\_id"**

**where** **"film"**.**"title"** = 'EGG IGBY';



Nombres y apellidos de los actores de la película ‘Egg Igby’.

**Explicación de la consulta:**

1. **Seleccionar las columnas de nombre y apellido**: SELECT "first\_name", "last\_name" de la tabla "actor". As para cambiar el nombre de las columnas al español.
2. **Indicar la tabla**: FROM "actor" porque queremos los datos de esta tabla.
3. **Hacer JOIN con "film\_actor"**: JOIN "film\_actor" ON "actor"."actor\_id" = "film\_actor"."actor\_id" para relacionar los actores con las películas en las que han actuado.
4. **Hacer JOIN con "film"**: JOIN "film" ON "film\_actor"."film\_id" = "film"."film\_id" para obtener los títulos de las películas.
5. **Filtrar por título**: WHERE "film"."title" = 'EGG IGBY' para seleccionar solo los actores de esa película.

**Consulta 18**

Selecciona todos los nombres de las películas únicos.

**select** **distinct** **"title"**

**from** **"film"**;



Nombres de todas las películas únicas.

**Explicación de la consulta:**

1. **Seleccionar el título de las películas**: SELECT "title" de la tabla "film".
2. **Usar DISTINCT**: DISTINCT se usa para eliminar duplicados y mostrar solo títulos únicos.
3. **Indicar la tabla**: FROM "film" porque queremos los datos de películas.

**Consulta 19**

Encuentra el título de las películas que son comedias y tienen una duración mayor a 180 minutos en la tabla “film”.

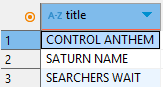
**select** **"film"**.**"title"**

**from** **"film"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"category"**.**"name"** = 'Comedy' **and** **"film"**.**"length"** > 180;



Películas que son comedias y tienen una duración mayor a 180 minutos (3 horas).

**Explicación de la consulta:**

1. **Seleccionar el título**: SELECT "title".
2. **Indicar la tabla**: FROM "film" porque queremos los datos de esta tabla.
3. **Hacer JOIN con "film\_category"**: JOIN "film\_category" ON "film"."film\_id" = "film\_category"."film\_id" para vincular películas con sus categorías.
4. **Hacer JOIN con "category"**: JOIN "category" ON "film\_category"."category\_id" = "category"."category\_id" para obtener los nombres de las categorías.
5. **Filtrar por categoría y duración**: WHERE "category"."name" = 'Comedy' AND "film"."length" > 180 para seleccionar solo comedias con más de 180 minutos de duración.

**Consulta 20**

Encuentra las categorías de películas que tienen un promedio de duración superior a 110 minutos y muestra el nombre de la categoría junto con el promedio de duración.

**select** **"category"**.**"name"** **as** **"categorias\_peliculas"**, **avg**(**"film"**.**"length"**) **as** **"promedio\_duracion"**

**from** **"film"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**group** **by** **"category"**.**"name"**

**having** **avg**(**"film"**.**"length"**) > 110;



Categorías de películas que tienen un promedio de duración superior a 110 minutos.

**Explicación de la consulta:**

1. **Seleccionar la categoría y el promedio**: SELECT "category"."name" y AVG("film"."length") para calcular la duración promedio. Se usa AS para renombrar.
2. **Indicar la tabla**: FROM "film" porque queremos los datos de esta tabla.
3. **Hacer JOIN con "film\_category"**: JOIN "film\_category" ON "film"."film\_id" = "film\_category"."film\_id" para vincular películas con sus categorías.
4. **Hacer JOIN con "category"**: JOIN "category" ON "film\_category"."category\_id" = "category"."category\_id" para obtener los nombres de las categorías.
5. **Agrupar por categoría**: GROUP BY "category"."name" para calcular el promedio por categoría.
6. **Filtrar por duración promedio**: HAVING AVG("film"."length") > 110 para mostrar solo las categorías con un promedio mayor a 110 minutos.

**Consulta 21**

¿Cuál es la media de duración del alquiler de las películas?

**select** **avg**(**"rental\_duration"**) **as** **"media\_duracion"**

**from** **"film"**;



Media de duración del alquiler de las películas.

**Explicación de la consulta:**

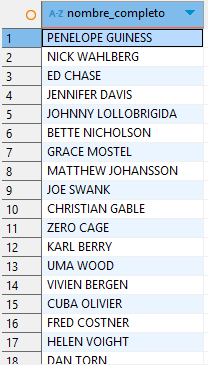
1. **Seleccionar el promedio de duración**: AVG("rental\_duration") para calcular la media de duración de los alquileres. Uso de AS para renombrar.
2. **Indicar la tabla**: FROM "film" porque queremos los datos de esta tabla.

**Consulta 22**

Crea una columna con el nombre y apellidos de todos los actores y actrices.

**select** **"first\_name"** || ' ' || **"last\_name"** **as** **"nombre\_completo"**

**from** **"actor"**;



Nombres y apellidos de todos los actores y actrices.

**Explicación de la consulta:**

1. **Seleccionar el nombre y apellido del actor**: SELECT para unir "first\_name" y "last\_name", agregando un espacio entre ellos || ' ' ||. AS "nombre\_completo" para que la columna tenga el nombre en español.
2. **Indicar la tabla**: FROM "actor" ya que los nombres de los actores están en esta tabla.

**Consulta 23**

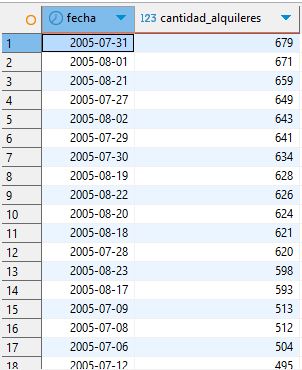
Números de alquiler por día, ordenados por cantidad de alquiler de forma descendente.

**select** **date**(**"rental\_date"**) **as** **"fecha"**, **count**(\*) **as** **"cantidad\_alquileres"**

**from** **"rental"**

**group** **by** **date**(**"rental\_date"**)

**order** **by** **"cantidad\_alquileres"** **desc**;



Número de alquileres por día, ordenados de forma descendente.

**Explicación de la consulta:**

1. **Extraer la fecha del alquiler**: SELECT DATE("rental\_date") para agrupar los alquileres por día.
2. **Indicar la tabla**: FROM "rental" ya que los nombres de los alquileres están en esta tabla.
3. **Contar el número de alquileres por día**: COUNT(\*) AS "cantidad\_alquileres".
4. **Agrupar por fecha**: GROUP BY DATE("rental\_date") para contar los alquileres por cada día específico.
5. **Ordenar en orden descendente**: ORDER BY "cantidad\_alquileres" DESC para mostrar primero los días con más alquileres.

**Consulta 24**

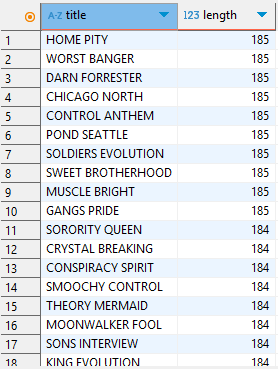
Encuentra las películas con una duración superior al promedio.

**select** **"title"**, **"length"**

**from** **"film"**

**where** **"length"** > (**select** **avg**(**"length"**) **from** **"film"**)

**order** **by** **"length"** **desc**;



Películas con una duración superior al promedio.

**Explicación de la consulta:**

1. **Seleccionar el título y duración**: SELECT "title", "length" de la tabla "film".
2. **Indicar la tabla**: FROM "film" ya que los nombres de las películas están en esta tabla.
3. **Filtrar por duración mayor al promedio**: WHERE "length" > (SELECT AVG("length") FROM "film"), utilizando una subconsulta para obtener la duración promedio.
4. **Ordenar en orden descendente**: ORDER BY "length" DESC para mostrar primero las películas más largas.

**Consulta 25**

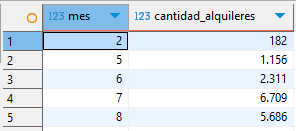
Averigua el número de alquileres registrados por mes.

**select** **extract**(**month** **from** **"rental\_date"**) **as** **"mes"**, **count**(\*) **as** **"cantidad\_alquileres"**

**from** **"rental"**

**group** **by** **extract**(**month** **from** **"rental\_date"**)

**order** **by** **"mes"**;



Número de alquileres por mes. Julio es cuando más películas se alquilan.

**Explicación de la consulta:**

1. **Extraer el mes de la fecha de alquiler**: SELECT EXTRACT(MONTH FROM "rental\_date") para obtener solo el número del mes.
2. **Contar los alquileres por mes**: COUNT(\*) AS "cantidad\_alquileres".
3. **Indicar la tabla**: FROM "rental" ya que los nombres de los alquileres están en esta tabla.
4. **Agrupar por mes**: GROUP BY EXTRACT(MONTH FROM "rental\_date") para agrupar los alquileres en cada mes.
5. **Ordenar por mes**: ORDER BY "mes" para mostrar los meses en orden cronológico.

**Consulta 26**

Encuentra el promedio, la desviación estándar y varianza del total pagado.

**select**

**avg**(**"amount"**) **as** **"promedio"**,

**stddev**(**"amount"**) **as** **"desviacion\_estandar"**,

**variance**(**"amount"**) **as** **"varianza"**

**from** **"payment"**;



Promedio: 4,20.

Desviación estándar: 2,36.

Varianza: 5,59.

**Explicación de la consulta:**

1. **Calcular el promedio**: AVG("amount") AS "promedio" para obtener la cantidad media de pago.
2. **Calcular la desviación estándar**: STDDEV("amount") AS "desviacion\_estandar" para medir la dispersión de los valores.
3. **Calcular la varianza**: VARIANCE("amount") AS "varianza" para obtener la variabilidad de los pagos.
4. **Especificar la tabla**: FROM "payment" porque los pagos están en esta tabla.

**Consulta 27**

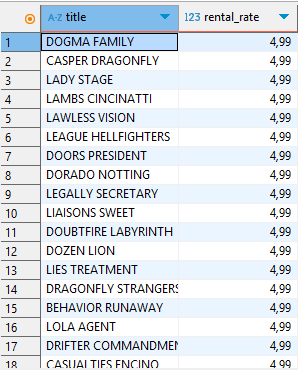
¿Qué películas se alquilan por encima del precio medio?

**select** **"film"**.**"title"**, **"film"**.**"rental\_rate"**

**from** **"film"**

**where** **"film"**.**"rental\_rate"** > (**select** **avg**(**"rental\_rate"**) **from** **"film"**)

**order** **by** **"film"**.**"rental\_rate"** **desc**;



Lista de películas que se alquilan por encima del precio medio.

**Explicación de la consulta:**

1. **Seleccionar el título y tarifa de alquiler**: SELECT “film”."title", “film”."rental\_rate".
2. **Indicar la tabla**: FROM "film" ya que los nombres de las películas están en esta tabla.
3. **Filtrar por precio superior al promedio**: WHERE "rental\_rate" > (SELECT AVG("rental\_rate") FROM "film"), usando una subconsulta para obtener el valor medio.
4. **Ordenar en orden descendente**: ORDER BY "rental\_rate" DESC para mostrar primero las películas más caras.

**Consulta 28**

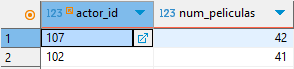
Muestra el id de los actores que hayan participado en más de 40 películas.

**select** **"actor\_id"**, **count**(**"film\_id"**) **as** **"num\_peliculas"**

**from** **"film\_actor"**

**group** **by** **"actor\_id"**

**having** **count**(**"film\_id"**) > 40;



Id de los actores que han participado en más de 40 películas.

**Explicación de la consulta:**

1. **Seleccionar el ID de los actores**: SELECT "actor\_id".
2. **Contar cuántas películas ha hecho cada actor**: COUNT("film\_id") AS "num\_peliculas".
3. **Indicar la tabla**: FROM "film\_actor" ya que los actores de las películas están en esta tabla.
4. **Agrupar por actor**: GROUP BY "actor\_id" para contar correctamente cada actor.
5. **Filtrar por actores con más de 40 películas**: HAVING COUNT("film\_id") > 40.

**Consulta 29**

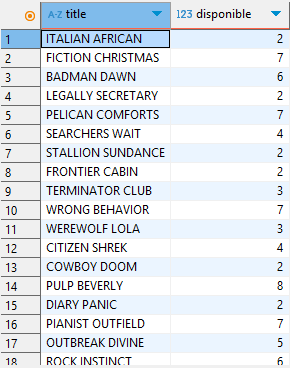
Obtener todas las películas y, si están disponibles en el inventario, mostrar la cantidad disponible.

**select** **"film"**.**"title"**, **count**(**"inventory"**.**"inventory\_id"**) **as** **"disponible"**

**from** **"film"**

**left** **join** **"inventory"** **on** **"film"**.**"film\_id"** = **"inventory"**.**"film\_id"**

**group** **by** **"film"**.**"title"**;



Lista y cantidad de películas disponibles en el inventario.

**Explicación de la consulta:**

1. **Seleccionar el título de la película**: SELECT "film"."title".
2. **Contar cuántas copias hay en el inventario**: COUNT("inventory"."inventory\_id") AS "disponible".
3. **Indicar la tabla**: FROM "film" ya que las películas están en esta tabla.
4. **Unir la tabla de películas con el inventario**: LEFT JOIN "inventory" ON "film"."film\_id" = "inventory"."film\_id" para mostrar todas las películas, incluso si no tienen copias disponibles.
5. **Agrupar por título**: GROUP BY "film"."title" para calcular correctamente la cantidad por película.

**Consulta 30**

Obtener los actores y el número de películas en las que ha actuado.

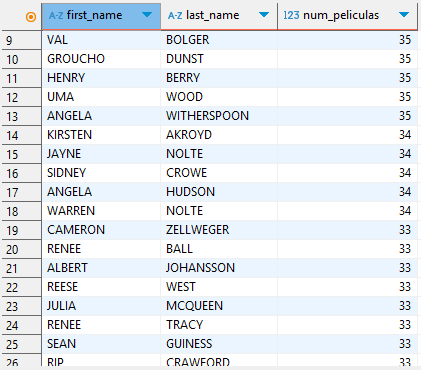
**select** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**, **count**(**"film\_actor"**.**"film\_id"**) **as** **"num\_peliculas"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**group** **by** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**order** **by** **"num\_peliculas"** **desc**;



Actores y número de películas en las que han actuado.

**Explicación de la consulta:**

1. **Seleccionar el nombre y apellido del actor**: SELECT "actor"."first\_name", "actor"."last\_name".
2. **Contar cuántas películas ha hecho cada actor**: COUNT("film\_actor"."film\_id") AS "num\_peliculas".
3. **Indicar la tabla**: FROM "actor" ya que los actores están en esta tabla.
4. **Unir la tabla de actores con la tabla intermedia**: JOIN "film\_actor" ON "actor"."actor\_id" = "film\_actor"."actor\_id".
5. **Agrupar por nombre y apellido**: GROUP BY "actor"."first\_name", "actor"."last\_name".
6. **Ordenar en orden descendente**: ORDER BY "num\_peliculas" DESC para mostrar primero los actores con más películas.

**Consulta 31**

Obtener todas las películas y mostrar los actores que han actuado en ellas, incluso si algunas películas no tienen actores asociados.

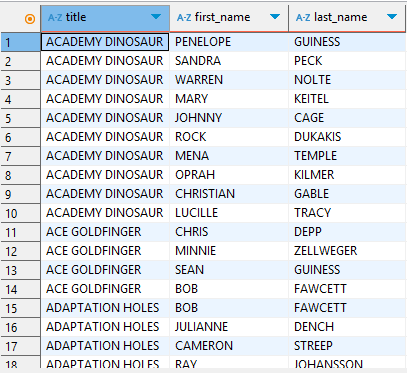
**select** **"film"**.**"title"**, **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"film"**

**left** **join** **"film\_actor"** **on** **"film"**.**"film\_id"** = **"film\_actor"**.**"film\_id"**

**left** **join** **"actor"** **on** **"film\_actor"**.**"actor\_id"** = **"actor"**.**"actor\_id"**

**order** **by** **"film"**.**"title"**;



Total de películas y actores que han actuado en ellas.

**Explicación de la consulta:**

1. **Seleccionar el título de la película y el nombre del actor**: SELECT "film"."title", "actor"."first\_name", "actor"."last\_name".
2. **Indicar la tabla**: FROM "film" ya que las películas están en esta tabla.
3. **Unir la tabla film con film\_actor**: LEFT JOIN "film\_actor" ON "film"."film\_id" = "film\_actor"."film\_id" para mantener todas las películas, aunque no tengan actores asociados.
4. **Unir la tabla film\_actor con actor**: LEFT JOIN "actor" ON "film\_actor"."actor\_id" = "actor"."actor\_id" para obtener el nombre de los actores.
5. **Ordenar alfabéticamente por título de película**: ORDER BY "film"."title".

**Consulta 32**

Obtener todos los actores y mostrar las películas en las que han actuado, incluso si algunos actores no han actuado en ninguna película.

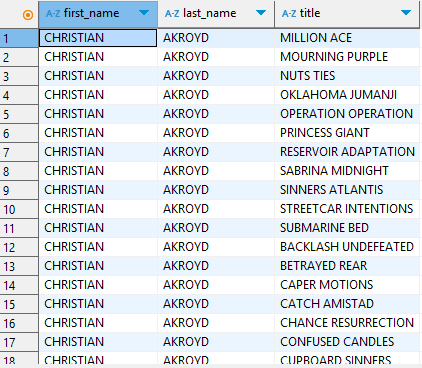
**select** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**, **"film"**.**"title"**

**from** **"actor"**

**left** **join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**left** **join** **"film"** **on** **"film\_actor"**.**"film\_id"** = **"film"**.**"film\_id"**

**order** **by** **"actor"**.**"last\_name"**, **"actor"**.**"first\_name"**;



Total de actores y películas en las que han actuado.

**Explicación de la consulta:**

1. **Seleccionar el nombre del actor y el título de la película**: SELECT "actor"."first\_name", "actor"."last\_name", "film"."title".
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.
3. **Unir la tabla actor con film\_actor**: LEFT JOIN "film\_actor" ON "actor"."actor\_id" = "film\_actor"."actor\_id" para incluir todos los actores, aunque no tengan películas.
4. **Unir la tabla film\_actor con film**: LEFT JOIN "film" ON "film\_actor"."film\_id" = "film"."film\_id" para obtener los títulos de las películas.
5. **Ordenar por apellido y nombre de los actores**: ORDER BY "actor"."last\_name", "actor"."first\_name".

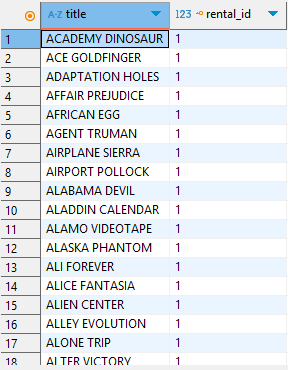
**Consulta 33**

Obtener todas las películas que tenemos y todos los registros de alquiler.

**select** **"film"**.**"title"**, **"rental"**.**"rental\_id"**

**from** **"film"**

**cross** **join** **"rental"**;



Total de películas y registros de alquiler de las mismas.

**Explicación de la consulta:**

1. **Seleccionar el título de la película y el ID del alquiler**: SELECT "film"."title", "rental"."rental\_id".
2. **Indicar la tabla**: FROM "film" ya que las películas están en esta tabla.
3. **Hacer un CROSS JOIN entre film y rental**: CROSS JOIN "rental" generará todas las combinaciones posibles entre películas y alquileres.

**Consulta 34**

Encuentra los 5 clientes que más dinero se hayan gastado con nosotros.

**select** **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**, **sum**(**"payment"**.**"amount"**) **as** **"total\_gastado"**

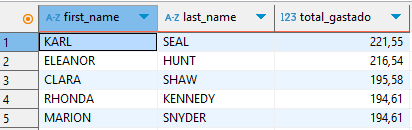
**from** **"customer"**

**join** **"payment"** **on** **"customer"**.**"customer\_id"** = **"payment"**.**"customer\_id"**

**group** **by** **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**order** **by** **"total\_gastado"** **desc**

**limit** 5;



5 clientes que más dinero se han gastado.

**Explicación de la consulta:**

1. **Seleccionar el nombre del cliente y la cantidad total gastada**: SELECT "customer"."first\_name", "customer"."last\_name", SUM("payment"."amount") AS "total\_gastado".
2. **Indicar la tabla**: FROM "customer" ya que las clientes están en esta tabla.
3. **Unir la tabla customer con payment**: JOIN "payment" ON "customer"."customer\_id" = "payment"."customer\_id".
4. **Agrupar por cliente**: GROUP BY "customer"."first\_name", "customer"."last\_name".
5. **Ordenar en orden descendente**: ORDER BY "total\_gastado" DESC para que los clientes que más han gastado aparezcan primero.
6. **Limitar a los 5 primeros clientes que más dinero se hayan gastado con nosotros**: LIMIT 5.

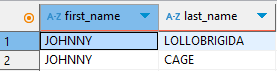
**Consulta 35**

Selecciona todos los actores cuyo primer nombre es 'Johnny'.

**select** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"actor"**

**where** **"actor"**.**"first\_name"** = 'JOHNNY';



Actores cuyo primer nombre es 'Johnny'.

**Explicación de la consulta:**

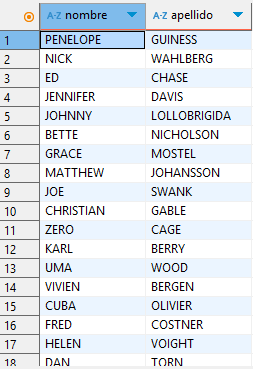
1. **Seleccionar el nombre y apellido del actor**: SELECT "actor"."first\_name", "actor"."last\_name".
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.
3. **Filtrar por nombre 'Johnny'**: WHERE "actor"."first\_name" = 'JOHNNY'.

**Consulta 36**

Renombra la columna “first\_name” como Nombre y “last\_name” como Apellido.

**select** **"first\_name"** **as** **"nombre"**, **"last\_name"** **as** **"apellido"**

**from** **"actor"**;



Renombrar la columna “first\_name” como Nombre y “last\_name” como Apellido.

**Explicación de la consulta:**

1. **Seleccionar las columnas:** SELECT "first\_name" y "last\_name".
2. **Renombrar las columnas:** "first\_name"como "Nombre" y "last\_name" como "Apellido" usando AS.
3. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.

**Consulta 37**

Encuentra el ID del actor más bajo y más alto en la tabla actor.

**select** **min**(**"actor\_id"**) **as** **"id\_actor\_más\_bajo"**, **max**(**"actor\_id"**) **as** **"id\_actor\_más\_alto"**

**from** **"actor"**;



ID del actor más bajo y más alto.

**Explicación de la consulta:**

1. **Seleccionar las columnas**: SELECTMIN(actor\_id) para encontrar el ID más bajo yMAX(actor\_id) para encontrar el ID más alto. Uso de AS para renombrar.
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.

**Consulta 38**

Cuenta cuántos actores hay en la tabla “actor”.

**select** **count**(\*) **as** **"total\_actores"**

**from** **"actor"**;



Número de actores en la tabla “actor”.

**Explicación de la consulta:**

1. **Usar**:SELECT COUNT(\*) para contar todas las filas de la tabla actor. AS "total\_actores".
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.

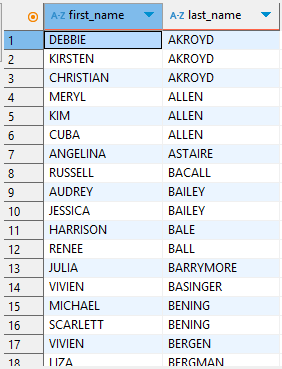
**Consulta 39**

Selecciona todos los actores y ordénalos por apellido en orden ascendente.

**select** **"first\_name"**, **"last\_name"**

**from** **"actor"**

**order** **by** **"last\_name"** **asc**;



Ordenación de todos los actores por apellido de forma ascendente.

**Explicación de la consulta:**

1. **Seleccionar las columnas**:SELECT "first\_name" y "last\_name" de la tabla actor.
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.
3. **Ordenar por:** "last\_name" en orden ascendente (ASC).

**Consulta 40**

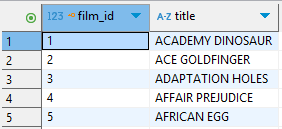
Selecciona las primeras 5 películas de la tabla “film”.

**select** **"film\_id"**, **"title"**

**from** **"film"**

**order** **by** **"film\_id"**

**limit** 5;



5 primeras películas.

**Explicación de la consulta:**

1. **Seleccionar las columnas:** SELECT "film\_id" y "title" de la tabla “film” usando FROM.
2. **Ordenar por:** "film\_id" en orden ascendente para asegurar consistencia.
3. **Limitar el resultado:** a las primeras 5 filas con LIMIT 5.

**Consulta 41**

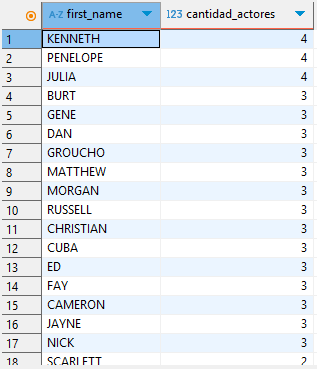
Agrupa los actores por su nombre y cuenta cuántos actores tienen el mismo nombre. ¿Cuál es el nombre más repetido?

**select** **"first\_name"**, **count**(\*) **as** **"cantidad\_actores"**

**from** **"actor"**

**group** **by** **"first\_name"**

**order** **by** **"cantidad\_actores"** **desc**;



Agrupación de los actores por nombre y cantidad.

**Explicación de la consulta:**

1. **Seleccionar la columna:** SELECT "first\_name" y contar cuántos actores tienen ese nombre con COUNT(\*). AS "cantidad\_actores".
2. **Indicar la tabla**: FROM "actor" ya que los actores se encuentran en esta tabla.
3. **Agrupar los resultados:** por "first\_name" con GROUP BY.
4. **Ordenar por:** "cantidad\_actores" en orden descendente (DESC) para ver el más repetido primero.

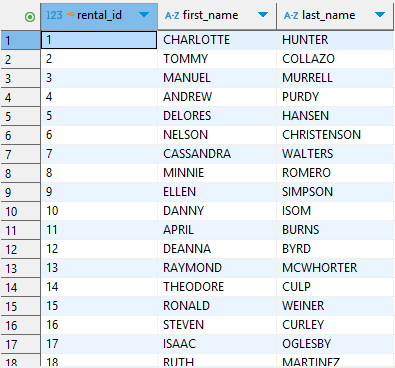
**Consulta 42**

Encuentra todos los alquileres y los nombres de los clientes que los realizaron.

**select** **"rental"**.**"rental\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**from** **"rental"**

**join** **"customer"** **on** **"rental"**.**"customer\_id"** = **"customer"**.**"customer\_id"**;



Alquileres y nombres de los clientes que lo realizaron.

**Explicación de la consulta:**

1. **Seleccionar las columnas:** SELECT "rental"."rental\_id", "customer"."first\_name" y "customer"."last\_name".
2. **Indicar la tabla**: FROM "rental" ya que los alquileres están en esta tabla.
3. **Unir ambas tablas:** mediante un JOIN utilizando "customer\_id" como clave de relación.

**Consulta 43**

Muestra todos los clientes y sus alquileres si existen, incluyendo aquellos que no tienen alquileres.

**select**

**"customer"**.**"customer\_id"**,

**"customer"**.**"first\_name"**,

**"customer"**.**"last\_name"**,

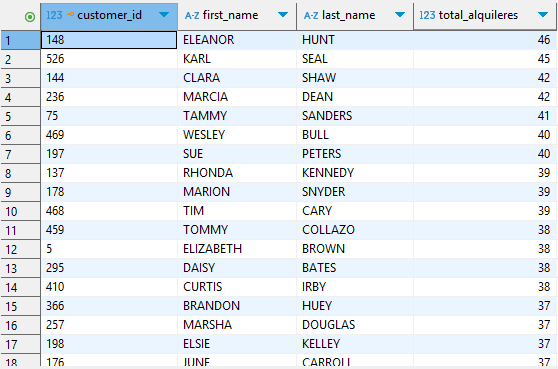
**count**(**"rental"**.**"rental\_id"**) **as** **"total\_alquileres"**

**from** **"customer"**

**left** **join** **"rental"** **on** **"customer"**.**"customer\_id"** = **"rental"**.**"customer\_id"**

**group** **by** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**order** **by** **"total\_alquileres"** **desc**;



Listado de los alquileres de los clientes.

**Explicación de la consulta:**

1. **Seleccionar las columnas:** SELECT "customer\_id", "first\_name" y "last\_name" de "customer".
2. **Contar:** cuántos alquileres (rental\_id) tiene cada cliente con COUNT("rental"."rental\_id") AS "total\_alquileres".
3. **Indicar la tabla**: FROM "customer" ya que los clientes están en esta tabla.
4. **Usar LEFT JOIN:** para incluir clientes sin alquileres (que aparecerán con total\_alquileres = 0).
5. **Agrupar con GROUP BY:** por "customer\_id", "first\_name", y "last\_name" para que el conteo funcione.
6. **Ordenar con ORDER BY "total\_alquileres" DESC** para ver primero los clientes con más alquileres.

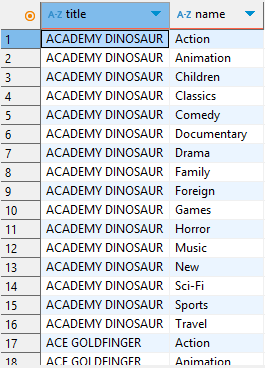
**Consulta 44**

Realiza un CROSS JOIN entre las tablas film y category. ¿Aporta valor esta consulta? ¿Por qué? Deja después de la consulta la contestación.

**select** **"film"**.**"title"**, **"category"**.**"name"**

**from** **"film"**

**cross** **join** **"category"**;



Resultado de la unión cruzada entre las tablas film y category. No, no aporta valor esta consulta. Esta consulta combina cada película con todas las categorías existentes, sin tener en cuenta si realmente están relacionadas.

**Explicación de la consulta:**

1. **Seleccionar columnas:** SELECT"title" de la tabla "film" y "name" de la tabla "category".
2. **Indicar la tabla**: FROM "film" ya que las películas están en esta tabla.
3. **Usar CROSS JOIN:** lo que generará una combinación de todas las películas con todas las categorías.

**Consulta 45**

Encuentra los actores que han participado en películas de la categoría 'Action'.

**select** **distinct** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"actor"**

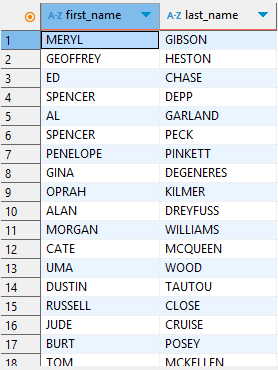
**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**join** **"film"** **on** **"film\_actor"**.**"film\_id"** = **"film"**.**"film\_id"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"category"**.**"name"** = 'Action';



Actores que han participado en películas de la categoría 'Action'.

**Explicación de la consulta:**

1. **Seleccionar las columnas**: SELECT"first\_name" y "last\_name" de la tabla actor. DISTINCT para que sean únicos.
2. **Indicar la tabla**: FROM "actor" ya que los actores están en esta tabla.
3. **Unir:** con "film\_actor" para relacionar actores con películas.
4. **Unir:** con "film" para obtener información de las películas.
5. **Unir:** con "film\_category" y "category" para encontrar la categoría de cada película.
6. **Filtrar**: solo aquellas películas cuya categoría sea ‘Action’.

**Consulta 46**

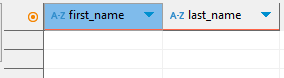
Encuentra todos los actores que no han participado en películas.

**select "actor"."first\_name", "actor"."last\_name"**

**from "actor"**

**left join "film\_actor" on "actor"."actor\_id" = "film\_actor"."actor\_id"**

**where "film\_actor"."film\_id" is null;**



No sé si he ejecutado correctamente esta consulta ya que no hay resultado. Deben de salir todos los actores que no han participado en ninguna película.

**Explicación de la consulta:**

1. **Seleccionar las colunmas:** SELECT"first\_name" y "last\_name" de la tabla actor.
2. **Indicar la tabla**: FROM "actor" ya que los actores están en esta tabla.
3. **Unir con:** "film\_actor" usando LEFT JOIN para obtener los actores y sus películas.
4. **Filtrar con:** WHERE "film\_actor"."film\_id" IS NULL para obtener solo aquellos actores que no tienen películas asociadas.

**Consulta 47**

Selecciona el nombre de los actores y la cantidad de películas en las que han participado.

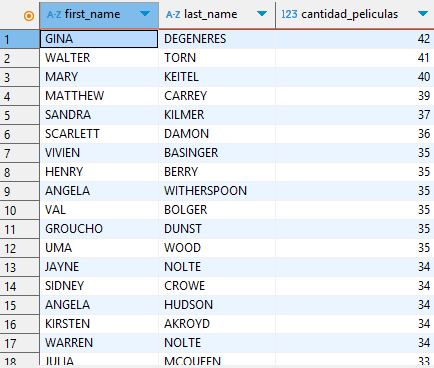
**select** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**, **count**(**"film\_actor"**.**"film\_id"**) **as** **"cantidad\_peliculas"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**group** **by** **"actor"**.**"actor\_id"**, **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**order** **by** **"cantidad\_peliculas"** **desc**;



Nombre de los actores y cantidad de películas en las que han participado.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "first\_name" y "last\_name".
2. **Contar:** cuántas películas ha hecho cada actor usando COUNT(film\_id). Renombrar columna con AS "cantidad\_peliculas".
3. **Indicar la tabla**: FROM "actor" ya que los actores están en esta tabla.
4. **Unir con:** "film\_actor" para obtener la relación entre actores y películas.
5. **Agrupar por:** "actor\_id", "first\_name" y "last\_name" para contar correctamente.
6. **Ordenar por**: "cantidad\_películas" en orden descendente (DESC).

**Consulta 48**

Crea una vista llamada “actor\_num\_peliculas” que muestre los nombres de los actores y el número de películas en las que han participado.

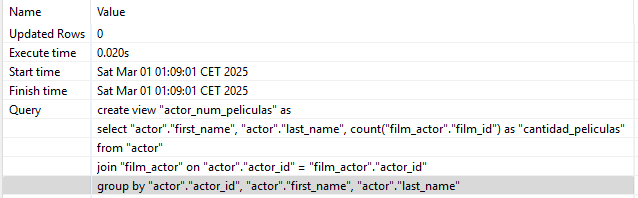
**create** **view** **"actor\_num\_peliculas"** **as**

**select** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**, **count**(**"film\_actor"**.**"film\_id"**) **as** **"cantidad\_peliculas"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**group** **by** **"actor"**.**"actor\_id"**, **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**;



Creación de vista llamada “actor\_num\_peliculas” que muestra los nombres de los actores y el número de películas en las que han participado.

**Explicación de la consulta:**

1. **Crear una vista**: CREATE VIEW para guardar la consulta de forma reutilizable.
2. **Seleccionar**: SELECT "first\_name", "last\_name" y contar cuántas películas ha hecho cada actor con el COUNT.
3. **Indicar la tabla**: FROM "actor" ya que los actores están en esta tabla.
4. **Unir con JOIN:** Unir "film\_actor" para obtener las películas en las que participó cada actor.
5. **Agrupar por**: "actor\_id", "first\_name" y "last\_name".

**Consulta 49**

Calcula el número total de alquileres realizados por cada cliente.

**select** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**, **count**(**"rental"**.**"rental\_id"**) **as** **"total\_alquileres"**

**from** **"customer"**

**join** **"rental"** **on** **"customer"**.**"customer\_id"** = **"rental"**.**"customer\_id"**

**group** **by** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**order** **by** **"total\_alquileres"** **desc**;

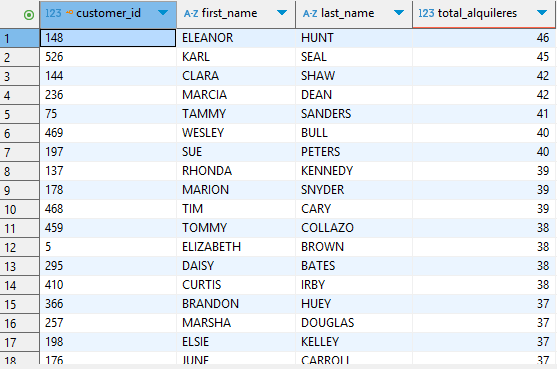


Tabla con el número total de alquileres realizados por cada cliente.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT"customer\_id", "first\_name", "last\_name" de la tabla customer.
2. **Contar:** cuántos alquileres "rental\_id" ha realizado cada cliente.
3. **Unir** con la tabla "rental" usando "customer\_id" como clave de relación.
4. **Agrupar por:** "customer\_id", "first\_name" y "last\_name" para obtener el total de alquileres por cliente.
5. **Ordenar por**: la cantidad de alquileres en orden descendente (DESC).

**Consulta 50**

Calcula la duración total de las películas en la categoría 'Action'.

**select** **sum**(**"film"**.**"length"**) **as** **"duracion\_total"**

**from** **"film"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"category"**.**"name"** = 'Action';



Duración total de las películas en la categoría ‘Action’.

**Explicación de la consulta:**

1. **Seleccionar**: SELECT SUM de la columna "length" (suma de la duración de las películas). Uso de AS para renombrar la columna en español.
2. **Indicar la tabla**: FROM "film" ya que las películas están en esta tabla.
3. **Unir con**: "film\_category" para relacionar las películas con sus categorías.
4. **Unir con**: "category" para obtener el nombre de la categoría.
5. **Filtrar con**: WHERE "category.name" = 'Action' para contar solo las películas de acción.

**Consulta 51**

Crea una tabla temporal llamada “cliente\_rentas\_temporal” para almacenar el total de alquileres por cliente.

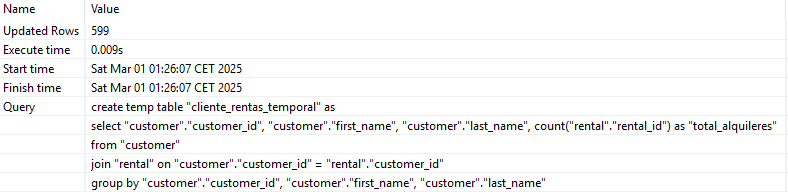
**create** **temp** **table** **"cliente\_rentas\_temporal"** **as**

**select** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**, **count**(**"rental"**.**"rental\_id"**) **as** **"total\_alquileres"**

**from** **"customer"**

**join** **"rental"** **on** **"customer"**.**"customer\_id"** = **"rental"**.**"customer\_id"**

**group** **by** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**;



Creación de una tabla temporal “cliente\_rentas\_temporal” para almacenar el total de alquileres por cliente.

**Explicación de la consulta:**

1. **Crear una tabla temporal**: CREATE TEMP TABLE de "cliente\_rentas\_temporal" y seguido se pone AS.
2. **Seleccionar**: SELECT "customer\_id", "first\_name", "last\_name" y contar la cantidad de alquileres realizados COUNT"rental\_id" de la tabla "costumer".
3. **Unir**: JOIN "rental" y ON usando "customer\_id" como clave de relación.
4. **Agrupar por**: "customer\_id", "first\_name" y "last\_name" para obtener el total de alquileres por cliente.

**Consulta 52**

Crea una tabla temporal llamada “peliculas\_alquiladas” que almacene las películas que han sido alquiladas al menos 10 veces.

**create** **temp** **table** **"peliculas\_alquiladas"** **as**

**select** **"film"**.**"film\_id"**, **"film"**.**"title"**, **count**(**"rental"**.**"rental\_id"**) **as** **"total\_alquileres"**

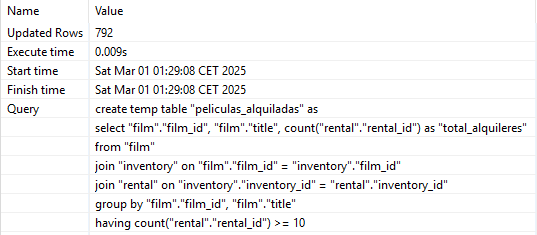
**from** **"film"**

**join** **"inventory"** **on** **"film"**.**"film\_id"** = **"inventory"**.**"film\_id"**

**join** **"rental"** **on** **"inventory"**.**"inventory\_id"** = **"rental"**.**"inventory\_id"**

**group** **by** **"film"**.**"film\_id"**, **"film"**.**"title"**

**having** **count**(**"rental"**.**"rental\_id"**) >= 10;



Creación de una tabla temporal “peliculas\_alquiladas” para almacenar las películas que han sido alquiladas al menos 10 veces.

**Explicación de la consulta:**

1. **Crear una tabla temporal:** que se llame "peliculas\_alquiladas" con CREATE TEMP TABLE. Incluir AS al final.
2. **Seleccionar:** SELECT "film\_id", "title" y contar el número de veces que ha sido alquilada con COUNT"rental\_id".
3. **Unir con:** JOIN"inventory" para relacionar las películas con su inventario.
4. **Unir con:** JOIN"rental" para contar los alquileres de cada película.
5. **Agrupar por:** GROUP BY"film\_id" y "title".
6. **Filtrar con:** HAVING COUNT ("rental"."rental\_id") >= 10 para incluir solo películas con al menos 10 alquileres.

**Consulta 53**

Encuentra el título de las películas que han sido alquiladas por el cliente con el nombre ‘Tammy Sanders’ y que aún no se han devuelto. Ordena los resultados alfabéticamente por título de película.

**select** **distinct** **"film"**.**"title"**

**from** **"film"**

**join** **"inventory"** **on** **"film"**.**"film\_id"** = **"inventory"**.**"film\_id"**

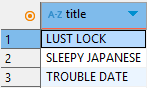
**join** **"rental"** **on** **"inventory"**.**"inventory\_id"** = **"rental"**.**"inventory\_id"**

**join** **"customer"** **on** **"rental"**.**"customer\_id"** = **"customer"**.**"customer\_id"**

**where** **"customer"**.**"first\_name"** = 'TAMMY' **and** **"customer"**.**"last\_name"** = 'SANDERS'

**and** **"rental"**.**"return\_date"** **is** **null**

**order** **by** **"film"**.**"title"** **asc**;



Título de las películas que han sido alquiladas por el cliente ‘Tammy Sanders’ y que aún no se han devuelto, ordenadas alfabéticamente por título de película.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT"title" de la tabla "film" y asegurar que los títulos sean únicos con DISTINCT.
2. **Unir con:** JOIN"inventory" para relacionar películas con inventario.
3. **Unir con:** JOIN"rental" para obtener los alquileres realizados.
4. **Unir con:** JOIN"customer" para identificar al cliente ‘Tammy Sanders’.
5. **Filtrar con:** WHERE "first\_name" = 'TAMMY' AND last\_name = 'SANDERS'.
6. **Filtrar con:** "rental"."return\_date" IS NULL para encontrar alquileres no devueltos.
7. **Ordenar:** los resultados alfabéticamente (ORDER BY "film"."title" ASC).

**Consulta 54**

Encuentra los nombres de los actores que han actuado en al menos una película que pertenece a la categoría ‘Sci-Fi’. Ordena los resultados alfabéticamente por apellido.

**select** **distinct** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

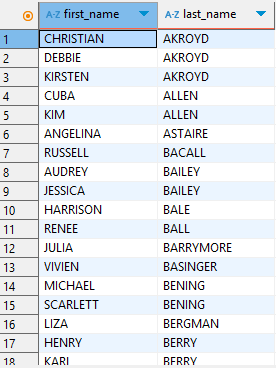
**join** **"film"** **on** **"film\_actor"**.**"film\_id"** = **"film"**.**"film\_id"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"category"**.**"name"** = 'Sci-Fi'

**order** **by** **"actor"**.**"last\_name"** **asc**;



Nombres de los actores que han actuado en al menos una película que pertenece a la categoría ‘Sci-Fi’, ordenadas alfabéticamente por apellido.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT"first\_name" y "last\_name" de la tabla "actor" (FROM).
2. **Asegurar:** que los nombres sean únicos con DISTINCT.
3. **Unir con:** JOIN"film\_actor" para relacionar actores con películas.
4. **Unir con:** JOIN "film" para obtener detalles de las películas.
5. **Unir con:** JOIN "film\_category" para relacionar películas con categorías.
6. **Unir con:** JOIN "category" para obtener el nombre de la categoría.
7. **Filtrar con:** WHERE "category.name" = 'Sci-Fi' para obtener solo películas de ciencia ficción.
8. **Ordenar alfabéticamente por:** ORDER BY "actor".last\_name" ASC.

**Consulta 55**

Encuentra el nombre y apellido de los actores que han actuado en películas que se alquilaron después de que la película ‘Spartacus Cheaper’ se alquilara por primera vez. Ordena los resultados alfabéticamente por apellido.

**select** **distinct** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

**join** **"film"** **on** **"film\_actor"**.**"film\_id"** = **"film"**.**"film\_id"**

**join** **"inventory"** **on** **"film"**.**"film\_id"** = **"inventory"**.**"film\_id"**

**join** **"rental"** **on** **"inventory"**.**"inventory\_id"** = **"rental"**.**"inventory\_id"**

**where** **"rental"**.**"rental\_date"** > (

**select** **min**(**"rental"**.**"rental\_date"**)

**from** **"rental"**

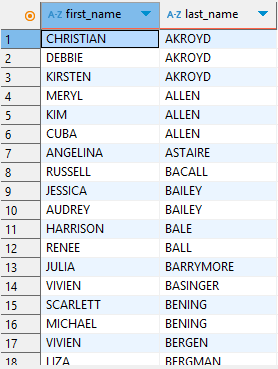
**join** **"inventory"** **on** **"rental"**.**"inventory\_id"** = **"inventory"**.**"inventory\_id"**

**join** **"film"** **on** **"inventory"**.**"film\_id"** = **"film"**.**"film\_id"**

**where** **"film"**.**"title"** = 'SPARTACUS CHEAPER'

)

**order** **by** **"actor"**.**"last\_name"** **asc**;



Nombre y apellido de los actores que han actuado en películas que se alquilaron después de que la película ‘Spartacus Cheaper’ se alquilara por primera vez, ordenados alfabéticamente por apellido.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT"first\_name" y "last\_name" de la tabla "actor".
2. **Asegurar:** que los nombres sean únicos con DISTINCT.
3. **Unir con:** JOIN "film\_actor" para relacionar actores con películas.
4. **Unir con:** JOIN "film" para obtener detalles de las películas.
5. **Unir con:** JOIN "inventory" para relacionar películas con inventario.
6. **Unir con:** JOIN "rental" para obtener los alquileres.
7. **Usar:** una subconsulta para encontrar la fecha del primer alquiler de ‘SPARTACUS CHEAPER’.
8. **Filtrar con:** WHERE "rental"."rental\_date" > (subconsulta) para obtener actores en películas alquiladas después de esa fecha.
9. **Seleccionar:** con el SELECT min("rental"."rental\_date").
10. **De la tabla**: (FROM) "rental".
11. **Unir con:** JOIN "inventory" para relacionar películas con inventario.
12. **Unir con:** JOIN "film" para obtener detalles de las películas.
13. **Donde:** WHERE "film"."title" = 'SPARTACUS CHEAPER'
14. **Ordenar los resultados alfabéticamente por:** ORDER BY"actor"."last\_name" ASC.

**Consulta 56**

Encuentra el nombre y apellido de los actores que no han actuado en ninguna película de la categoría ‘Music’.

**select** **distinct** **"actor"**.**"first\_name"**, **"actor"**.**"last\_name"**

**from** **"actor"**

**where** **"actor\_id"** **not** **in** (

**select** **distinct** **"actor"**.**"actor\_id"**

**from** **"actor"**

**join** **"film\_actor"** **on** **"actor"**.**"actor\_id"** = **"film\_actor"**.**"actor\_id"**

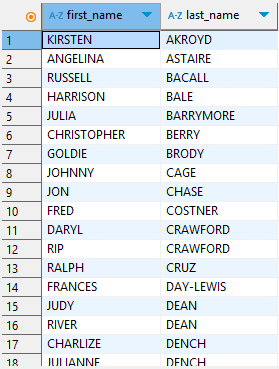
**join** **"film\_category"** **on** **"film\_actor"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"category"**.**"name"** = 'Music'

)

**order** **by** **"actor"**.**"last\_name"**;



Nombre y apellido de los actores que no han actuado en ninguna película de la categoría ‘Music’.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT DISTINCT los actores de la tabla "actor".
2. **Filtrar (WHERE):** aquellos cuyo "actor\_id" NO está en una subconsulta que encuentra los actores que participaron en películas de la categoría 'Music'.
3. **Unir con:** La subconsulta une (JOIN) "film\_actor", "film\_category" y "category" para identificar los actores de películas en la categoría 'Music'.
4. **Ordenar:** los resultados por apellido, ORDER BY "actor"."last\_name".

**Consulta 57**

Encuentra el título de todas las películas que fueron alquiladas por más de 8 días.

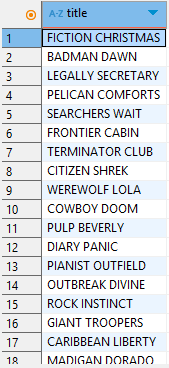
**select** **distinct** **"film"**.**"title"**

**from** **"rental"**

**join** **"inventory"** **on** **"rental"**.**"inventory\_id"** = **"inventory"**.**"inventory\_id"**

**join** **"film"** **on** **"inventory"**.**"film\_id"** = **"film"**.**"film\_id"**

**where** **"rental"**.**"return\_date"** - **"rental"**.**"rental\_date"** > **interval** '8 days';



Películas que fueron alquiladas por más de 8 días.

**Explicación de las consultas:**

1. **Seleccionar:** SELECT DISTINCT el título de las películas de la tabla "rental".
2. **Unir con:** JOIN "inventory" y "film" para relacionar alquileres con películas.
3. **Filtrar por (WHERE):** los registros donde la diferencia entre "return\_date" y "rental\_date" sea mayor a 8 días (interval '8 days').

**Consulta 58**

Encuentra el título de todas las películas que son de la misma categoría que ‘Animation’.

**select** **distinct** **"film"**.**"title"**

**from** **"film"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**where** **"film\_category"**.**"category\_id"** = (

**select** **"category\_id"** **from** **"category"** **where** **"name"** = 'Animation'

);



Películas que son de la categoría ‘Animation’.

**Explicación de las consultas:**

1. **Seleccionar:** SELECT DISTINCT el título de las películas de la tabla "film".
2. **Unir con:** JOIN "film\_category".
3. **Filtrar por:** WHERE "category\_id".
4. **Seleccionar (SELECT):** los títulos de las películas de la categoría de Animation.

**Consulta 59**

Encuentra los nombres de las películas que tienen la misma duración que la película con el título ‘Dancing Fever’. Ordena los resultados alfabéticamente por título de película.

**select** **"title"**

**from** **"film"**

**where** **"length"** = (

**select** **"length"** **from** **"film"** **where** **"title"** = 'DANCING FEVER'

)

**order** **by** **"title"**;



Películas que tienen la misma duración que la película ‘Dancing Fever’, ordenadas alfabéticamente por título de película.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "title" de la tabla "film", usando FROM.
2. **Filtrar por:** la duración ("length") de la película 'DANCING FEVER'.
3. **Ordenar por:** ORDER BY "title"los títulos alfabéticamente.

**Consulta 60**

Encuentra los nombres de los clientes que han alquilado al menos 7 películas distintas. Ordena los resultados alfabéticamente por apellido.

**select** **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**from** **"customer"**

**join** **"rental"** **on** **"customer"**.**"customer\_id"** = **"rental"**.**"customer\_id"**

**join** **"inventory"** **on** **"rental"**.**"inventory\_id"** = **"inventory"**.**"inventory\_id"**

**join** **"film"** **on** **"inventory"**.**"film\_id"** = **"film"**.**"film\_id"**

**group** **by** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**having** **count**(**distinct** **"film"**.**"film\_id"**) >= 7

**order** **by** **"customer"**.**"last\_name"**;



Nombres de los clientes que han alquilado al menos 7 películas distintas, ordenados alfabéticamente por apellido.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "customer"."first\_name", "customer"."last\_name".
2. **Indicar la tabla:** FROM "customer".
3. **Unir con:** JOIN "customer", "rental", "inventory" y "film" para vincular clientes con películas alquiladas.
4. **Agrupar por clientes:** GROUP BY.
5. **Contar:** la cantidad de películas únicas alquiladas por cada cliente y filtrar los clientes que han alquilado al menos 7 películas distintas con COUNT (DISTINCT "film"."film\_id") >= 7.
6. **Ordenar alfabéticamente por apellido:** ORDER BY "customer"."last\_name".

**Consulta 61**

Encuentra la cantidad total de películas alquiladas por categoría y muestra el nombre de la categoría junto con el recuento de alquileres.

**select** **"category"**.**"name"**, **count**(**"rental"**.**"rental\_id"**) **as** recuento\_alquileres

**from** **"rental"**

**join** **"inventory"** **on** **"rental"**.**"inventory\_id"** = **"inventory"**.**"inventory\_id"**

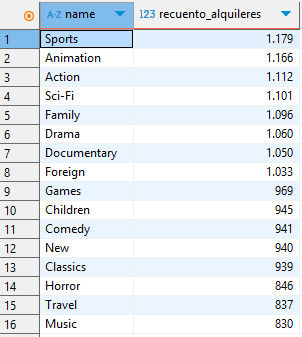
**join** **"film"** **on** **"inventory"**.**"film\_id"** = **"film"**.**"film\_id"**

**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**group** **by** **"category"**.**"name"**

**order** **by** recuento\_alquileres **desc**;



Cantidad total de películas alquiladas por categoría y nombre de la categoría junto con el recuento de alquileres.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "category"."name" y contar COUNT("rental"."rental\_id") AS recuento\_alquileres.
2. **Indicar la tabla:** FROM "rental", porque estos son los datos que queremos.
3. **Unir con:** JOIN "rental", "inventory", "film", "film\_category" y "category" para relacionar alquileres con categorías.
4. **Agrupar por:** categoría ("category"."name").
5. **Ordenar en orden descendente por:** ORDER BY recuento\_alquileres DESC.

**Consulta 62**

Encuentra el número de películas por categoría estrenadas en 2006.

**select** **"category"**.**"name"** **as** categorias, **count**(**"film"**.**"film\_id"**) **as** total\_peliculas

**from** **"film"**

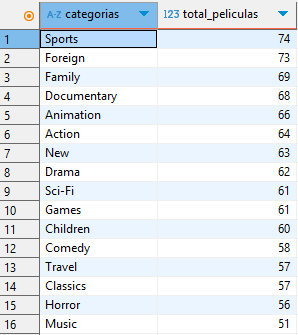
**join** **"film\_category"** **on** **"film"**.**"film\_id"** = **"film\_category"**.**"film\_id"**

**join** **"category"** **on** **"film\_category"**.**"category\_id"** = **"category"**.**"category\_id"**

**where** **"film"**.**"release\_year"** = 2006

**group** **by** **"category"**.**"name"**

**order** **by** total\_peliculas **desc**;



Número de películas por categoría estrenadas en 2006.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "category"."name" y contar: COUNT("film"."film\_id"). Uso de AS para renombrar columnas.
2. **Indicar la tabla:** FROM "film".
3. **Unir con:** JOIN "film\_category" y "category" para relacionar películas con categorías.
4. **Filtrar por:** películas estrenadas en 2006. Uso de WHERE.
5. **Agrupar por:** GROUP BY "category"."name".
6. **Ordenar por:** cantidad de películas en orden descendente (ORDER BY total\_peliculas desc).

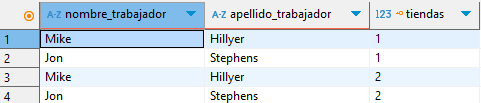
**Consulta 63**

Obtén todas las combinaciones posibles de trabajadores con las tiendas que tenemos.

**select** **"staff"**.**"first\_name"** **as** nombre\_trabajador, **"staff"**.**"last\_name"** **as** apellido\_trabajador, **"store"**.**"store\_id"** **as** tiendas

**from** **"staff"**

**cross** **join** **"store"**;



Todas las combinaciones posibles de trabajadores con las tiendas que hay.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "staff"."first\_name", "staff"."last\_name", "store"."store\_id". Uso de AS para renombrar todas las columnas.
2. **Indicar la tabla:** FROM "staff".
3. **Unir con:** CROSS JOIN para generar todas las combinaciones entre "staff" y "store".

**Consulta 64**

Encuentra la cantidad total de películas alquiladas por cada cliente y muestra el ID del cliente, su nombre y apellido junto con la cantidad de películas alquiladas.

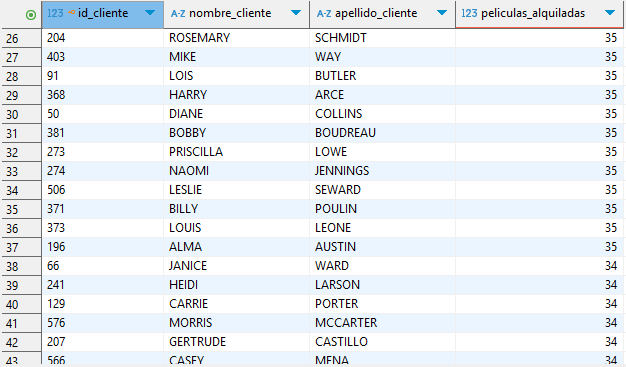
**select** **"customer"**.**"customer\_id"** **as** id\_cliente, **"customer"**.**"first\_name"** **as** nombre\_cliente, **"customer"**.**"last\_name"** **as** apellido\_cliente, **count**(**"rental"**.**"rental\_id"**) **as** peliculas\_alquiladas

**from** **"customer"**

**left** **join** **"rental"** **on** **"customer"**.**"customer\_id"** = **"rental"**.**"customer\_id"**

**group** **by** **"customer"**.**"customer\_id"**, **"customer"**.**"first\_name"**, **"customer"**.**"last\_name"**

**order** **by** peliculas\_alquiladas **desc**;



Cantidad total de películas alquiladas por cada cliente.

**Explicación de la consulta:**

1. **Seleccionar:** SELECT "customer"."customer\_id", "customer"."first\_name", "customer"."last\_name". Uso de AS para remombrar columnas.
2. **Uso de COUNT:** COUNT("rental"."rental\_id"). Uso de AS para remombrar columnas.
3. **Indicar la tabla:** FROM "customer".
4. **Uso de:** LEFT JOIN "rental" ON "customer"."customer\_id" = "rental"."customer\_id".
5. **Agrupar por:** GROUP BY "customer".
6. **Ordenar por:** ORDER BY películas\_alquiladas DESC.