1) Realizar una consulta donde se listen todos los clientes (customer) con su nombre y apellido, y sus direcciones, las direcciones deberan mostrar los campos "address" y "address", el nombre de la ciudad y el pais.

2) Mostrar el listado del punto 1 ordenado por ciudad y pais

3) Mostrar el listado del punto anterior agregando una columna con la cantidad de alquileres (rental) por cliente.

4) Mostrar el listado del pto. anterior agregando una columna con el total de pagos hecho por cada cliente, (no tomar en cuenta el atributo "customer\_id" de la tabla "payments")

5) Mostrar el listado del punto anterior dejando solo aquellos registros que correspondan a alquileres realizados durante el mes de mayo.

6) Mostrar el listado del punto anterior mostrando solo los clientes de ese listado que gastaron mas de 10 pesos.

7) Mostrar el listado del punto anterior mostrando solo los 10 clientes de ese listado que mas gastaron.

DIFERENCIA ENTRE ARCHIVO “QUERY” Y ARCH “sql fILE” ¿?????

1. Realizar una consulta donde se listen todos los clientes (customer)

**select \* from customer**

**599 filas**

1. Realizar una consulta donde se listen todos los clientes (customer) con su nombre y apellido, y sus direcciones

**select first\_name, last\_name, address\_id**

**from customer**

**AQUÍ SE MUESTRAN NOMBRE, APELLIDO Y EL ID DE DIRECCION, NO LA DIRECCION EN SI**

1. Realizar una consulta donde se listen todos los clientes (customer) con su nombre y apellido, y sus direcciones, las direcciones deberan mostrar los campos "address" y "address", el nombre de la ciudad y el pais.

**~~select first\_name, last\_name, customer.address\_id~~**

**~~from customer~~**

**~~join address~~**

~~AL AGREGAR UNA SEGUNDA TABLA, DEMAND SE ESPECIFIQUEN LA TABLA DE COLUMNAS CON IGUAL DENOMINACION~~

~~HASTA AQUÍ, SOLO UNI TABLAS~~

~~CANTIDAD EXESIVA DE FILAS, MAS DE 599, CANT DE CLIENTES. MAL~~

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**from customer, address, city, country**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

2) Mostrar el listado del punto 1 ordenado por ciudad y pais

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**from customer, address, city, country**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**order by country , city**

3) Mostrar el listado del punto anterior agregando una columna con la cantidad de alquileres (rental) por cliente.

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**, count(\*) as cant\_alq**

**#, rental.rental\_id**

**from customer, address, city, country, rental**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**AND customer.customer\_id = rental.customer\_id**

**# 16047 alq**

**#599 clientes**

**group by customer.customer\_id**

**order by country , city**

4) Mostrar el listado del pto. anterior agregando una columna con el total de pagos hecho por cada cliente, (no tomar en cuenta el atributo "customer\_id" de la tabla "payments")

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**, count(\*) as cant\_alq**

**#, rental.rental\_id**

**, sum( py.amount) as total\_pagos**

**from customer, address, city, country, rental r, payment py**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**AND customer.customer\_id = r.customer\_id**

**# al juntar por esta condicion, se reduce a 16044 filas, esto implica que existen 3 alquileres sin pagos, y 5 pagos no asociados a alquileres**

**# 16047 alq**

**# 599 clientes**

**AND r.rental\_id = py.rental\_id**

**# 16049 pagos**

**group by customer.customer\_id**

**order by country , city**

5) Mostrar el listado del punto anterior dejando solo aquellos registros que correspondan a alquileres realizados durante el mes de mayo.

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**, count(\*) as cant\_alq**

**#, rental.rental\_id**

**, sum( py.amount) as total\_pagos**

**from customer, address, city, country, rental r ,payment py**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**AND customer.customer\_id = r.customer\_id**

**# 16047 alq**

**# 599 clientes**

**AND r.rental\_id = py.rental\_id**

**AND month(r.rental\_date) = 5**

**# 16049 pagos**

**group by customer.customer\_id**

**#having month(address.city\_id) = 5**

**#al hacer junta AND r.rental\_id = py.rental\_id la cantidad de registros baja**

**# a 16044, esto quiere decir que hay 3 alquileres sin pagar**

**order by country , city**

**520 filas**

6) Mostrar el listado del punto anterior mostrando solo los clientes de ese listado que gastaron mas de 10 pesos.

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**, count(\*) as cant\_alq**

**#, rental.rental\_id**

**, sum( py.amount) as total\_pagos**

**from customer, address, city, country, rental r ,payment py**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**AND customer.customer\_id = r.customer\_id**

**# 16047 alq**

**# 599 clientes**

**AND r.rental\_id = py.rental\_id**

**AND month(r.rental\_date) = 5**

**# 16049 pagos**

**group by customer.customer\_id**

**having total\_pagos < 10**

**#al hacer junta AND r.rental\_id = py.rental\_id la cantidad de registros baja**

**# a 16044, esto quiere decir que hay 3 alquileres sin pagar**

**order by country , city**

**182 filas**

7) Mostrar el listado del punto anterior mostrando solo los 10 clientes de ese listado que mas gastaron.

**select \* from (**

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

**#customer.address\_id , address.address\_id**

**, address.address #, address.address2 # address2 es una columna vacia con 4 nulls**

**#, address.city\_id , city.city\_id**

**, city.city**

**#, city.country\_id , country.country\_id**

**, country**

**, count(\*) as cant\_alq**

**#, rental.rental\_id**

**, sum( py.amount) as total\_pagos**

**from customer, address, city, country, rental r ,payment py**

**#limit 10**

**where customer.address\_id = address.address\_id**

**AND address.city\_id = city.city\_id**

**AND city.country\_id = country.country\_id**

**AND customer.customer\_id = r.customer\_id**

**# 16047 alq**

**# 599 clientes**

**AND r.rental\_id = py.rental\_id**

**AND month(r.rental\_date) = 5**

**# 16049 pagos**

**group by customer.customer\_id**

**#having total\_pagos > 10**

**#al hacer junta AND r.rental\_id = py.rental\_id la cantidad de registros baja**

**# a 16044, esto quiere decir que hay 3 alquileres sin pagar**

**order by country , city**

**) as a**

**where total\_pagos > 10**

**limit 10**

**trabajando con JOIN**

1) Realizar una consulta donde se listen todos los clientes (customer) con su nombre y apellido, y sus direcciones, las direcciones deberan mostrar los campos "address" y "address", el nombre de la ciudad y el pais.

2) Mostrar el listado del punto 1 ordenado por ciudad y pais

3) Mostrar el listado del punto anterior agregando una columna con la cantidad de alquileres (rental) por cliente.

4) Mostrar el listado del pto. anterior agregando una columna con el total de pagos hecho por cada cliente, (no tomar en cuenta el atributo "customer\_id" de la tabla "payments")

5) Mostrar el listado del punto anterior dejando solo aquellos registros que correspondan a alquileres realizados durante el mes de mayo.

6) Mostrar el listado del punto anterior mostrando solo los clientes de ese listado que gastaron mas de 10 pesos.

7) Mostrar el listado del punto anterior mostrando solo los 10 clientes de ese listado que mas gastaron.