



Instrucción SELECT para una sola tabla

1. Obtener un listado de todos los clientes y sus direcciones.

```
SELECT customer_num, fname, lname, address1, address2
FROM customer
```

2. Obtener el listado anterior pero sólo los clientes que viven en el estado de California "CA".

```
SELECT customer_num, fname, lname, address1, address2
FROM customer
WHERE state= 'CA'
```

3. Listar todas las ciudades (city) de la tabla clientes que pertenecen al estado de "CA", mostrar sólo una vez cada ciudad.

```
SELECT DISTINCT city
FROM customer
WHERE state= 'CA'
```

4. Ordenar la lista anterior alfabéticamente.

```
SELECT DISTINCT city
FROM customer
WHERE state= 'CA' ORDER BY city
```

5. Mostrar la dirección sólo del cliente 103. (customer_num)

```
SELECT address1, address2
FROM customer
WHERE customer_num=103
```

6. Que productos del fabricante “ANZ” existen en la tabla stock? Mostrar la lista con los datos de la tabla ordenados por el campo descripción. (description)

```
SELECT stock_num, manu_code, description,  
unit_price, unit, unit_descr  
FROM products  
ORDER BY description
```

7. Listar los códigos de fabricantes que tengan alguna orden de pedido ingresada, ordenados alfabéticamente y no repetidos.

```
SELECT DISTINCT manu_code  
FROM items  
ORDER BY manu_code
```

8. Escribir una sentencia SELECT que devuelva el número de orden, fecha de orden, número de cliente y fecha de embarque de todas las órdenes que no han sido pagadas (paid_date es nulo), pero fueron embarcadas (ship_date) durante los primeros nueve meses de 2015.

```
SELECT order_num, order_date, customer_num, ship_date  
FROM orders  
WHERE paid_date IS NULL  
AND ship_date BETWEEN '2015-01-01' AND '2015-09-30'
```

9. Obtener de la tabla cliente (customer) el número del cliente y el nombre de la compañía, la cual debe contener en su nombre la palabra “town”.

```
SELECT customer_num, Company  
FROM customer  
WHERE company LIKE '%town%'
```

10. Obtener el precio máximo y mínimo que se pagó (ship_charge) por todos los embarques. También el promedio de lo pagado. Se pide obtener la información de la tabla ordenes (orders).

```
SELECT MAX(ship_charge) maximo, MIN(ship_charge) minimo,  
AVG(ship_charge) promedio  
FROM orders
```

11. Realizar una consulta que muestre el número de orden, fecha de orden y fecha de embarque de todas que fueron embarcadas (ship_date) en el mismo mes que fue dada de alta la orden (order_date).

```
SELECT order_num, order_date, ship_date  
FROM orders  
WHERE MONTH(order_date)=MONTH(ship_date)
```

12. Escribir una consulta que devuelva la siguiente información:

- ☐ El número de cliente
- ☐ Fecha del embarque
- ☐ El total del costo (ship_charge) por embarque del cliente(ship_date).

Ordenar los resultados por el total de costo en orden inverso

```
SELECT customer_num, ship_date, COUNT(*), SUM(ship_charge)
costoTotal
FROM orders
WHERE ship_date IS NOT NULL
GROUP BY customer_num, ship_date
ORDER BY costoTotal DESC
```

13. Se requiere una lista de cada grupo de clientes por embarque (ship_date) cuya suma total del peso de sus embarques supere las 30 libras. Se requiere mostrar fecha de embarque y cantidad de libras. Ordenar el resultado por el total del peso en orden inverso por cantidad de libras.

```
SELECT ship_date, SUM(ship_charge) pesoTotal
FROM orders
GROUP BY ship_date
HAVING SUM(ship_charge) >30
ORDER BY pesoTotal DESC
```

Práctica - Clase 4

14. Escribir una consulta que liste todos los clientes que vivan en California ordenados por compañía. (En una tabla temporaria llamada forsam) Escribir otra sentencia SELECT que liste los datos de esa tabla para verificar si son correctos.

```
SELECT customer_num, fname, lname, adress1, adress2, company,
state, zipcode, phone, customer_num_referedBy
INTO #forsam
FROM customer WHERE state='CA'
ORDER BY company
```

15. Obtener un listado con la cantidad de productos únicos vendidos de cada fabricante (de todos los productos que el fabricante tiene en stock cuantos fueron comprados), en donde el total comprado a cada fabricante sea mayor a 1500\$. El listado deberá estar ordenado por cantidad de productos comprados de mayor a menor.

Fabricante	ProductosUnicos
XXX	99

```
SELECT manu_code Fabricante, COUNT(DISTINCT stock_num) productosUnicos
FROM items
GROUP BY manu_code
HAVING sum(total_price) >1500
ORDER BY 2 DESC
```

16. Obtener un listado con el código de fabricante, nro de producto, la cantidad vendida (quantity), y el total vendido (total_price), para los fabricantes cuyo código tiene una "R" como segunda letra. Ordenados por código de fabricante y nro de producto.

Fabricante	Producto	Cantidad Vendida	Total Vendido
XXX	999	9999	999.99

```
SELECT manu_code fabricante, stock_num producto, SUM(quantity) cant_productos,
SUM(total_price) 'Total Vendido'
from items
WHERE manu_code LIKE '_R%'
```

```
group by manu_code,stock_num
order by 1,2
```

17. Crear una tabla temporal OrdenesTemp con la cantidad de ordenes por cada cliente, la fecha primera (order_date) y la fecha última de compra. Realizar una consulta de la tabla temp OrdenesTemp en donde la primer fecha de compra sea anterior a '2015-05-23 00:00:00.000'. Ordenar la consulta por fechaUltimaCompra en forma descendente.

Resultado:

Customer_num	CantidadCompras	fechaPrimerCompra	fechaUltimaCompra
106	2	2015-05-22 00:00:00.000	2015-06-25 00:00:00.000
104	4	2015-05-20 00:00:00.000	2015-06-22 00:00:00.000
101	1	2015-05-21 00:00:00.000	2015-05-21 00:00:00.000

```
select customer_num, count(order_num) CantidadCompras,
min(order_date)fechaPrimerCompra, max(order_date) fechaUltimaCompra
into #ordenesTemp
from orders
group by customer_num
```

```
select * from #ordenesTemp where fechaPrimerCompra < '2015-05-23 00:00:00.000'
order by fechaUltimaCompra DESC
```

18. Consultar la tabla temporal del punto anterior y obtener la cantidad de clientes con igual cantidad de compras. Ordenar el listado por cantidad de compras en orden descendente

Resultado:

CantidadCompras	CantidadClientes
4	1
2	3
1	13

```
select CantidadCompras, count(customer_num) CantidadClientes
from #ordenesTemp
group by CantidadCompras
order by CantidadCompras DESC
```

19. Desconectarse de la sesión. Volver a conectarse.

```
Ejecutar SELECT * from #ordenesTemp
Que pasó???
```

20. Se desean obtener la cantidad de clientes por cada state y city, donde los clientes contengan el string 'ts' en el nombre de compañía, el código postal este entre 93000 y 94100 y la ciudad no sea 'Mountain View'. Se desea el listado ordenado por ciudad

Formato

Estado	ciudad	cantidadClientes
xx	xxxxxxxxxxxxxx	999

```
select state,city,count(customer_num) cantidadClientes
from customer
where company like '%ts%' and zipcode BETWEEN 93000 AND 94100
```

```
and city <> 'Mountain View'  
group by state, city  
order by 2
```

21. Se desean obtener la cantidad de clientes referidos que posea cada cliente por cada estado (state) y además solo se mostraran aquellos cuya compañía empiece con una letra que este en el rango de 'A' a 'L'. Tomar sólo los clientes que estén referidos por algún otro cliente (customer_num_referredBy no sea NULL)

```
SELECT customer_num_referredBy, state, COUNT(*)  
FROM customer  
WHERE customer_num_referredBy IS NOT NULL  
AND company LIKE '[A-L]%'  
GROUP BY customer_num_referredBy, state
```

22. Se desea obtener el promedio de lead_time por cada estado, donde los fabricantes deben tener una 'e' en manu_name y el lead_time este entre 5 y 20.

```
SELECT state, AVG(lead_time) FROM manufact  
WHERE manu_name LIKE '%e%' AND lead_time BETWEEN 5 AND 20  
GROUP BY state
```

23. Se tiene la tabla units, de la cual se quiere saber la cantidad de unidades que hay por cada tipo (unit) que no tengan en nulo el descr_unit, y además se deben mostrar solamente los que cumplan que la cantidad mostrada se superior a 5. Al resultado final se le debe sumar 1

```
SELECT unit, COUNT(*)+1 FROM units  
WHERE unit_descr IS NOT NULL  
GROUP BY unit  
HAVING COUNT(*) > 5
```