



#### Práctica - Clase 4

14. Crear una consulta que liste todos los clientes que vivan en California ordenados por compañía.

15. Obtener un listado de la cantidad de productos únicos comprados a cada fabricante, 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.

```

SELECT manu_code Fabricante, COUNT(DISTINCT stock_num) productosUnicos
from items
group by manu_code
having sum(unit_price*quantity) >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 (quantity x unit\_price), para los fabricantes cuyo código tiene una "R" como segunda letra. Ordenar el listado por código de fabricante y nro de producto.

```

SELECT manu_code, stock_num producto,
COUNT(quantity)cant_productos, SUM(quantity*unit_price) total_vendidos
from items
WHERE manu_code LIKE '_R%'
group by manu_code,stock_num
order by 1,2
  
```

17. Crear una tabla temporal OrdenesTemp que contenga las siguientes columnas: cantidad de órdenes por cada cliente, primera y última fecha de orden de compra (order\_date) del cliente. 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', ordenada por fechaUltimaCompra en forma descendente.

```

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

```
select count(customer_num) cantClientes, cantidadCompras
from #ordenesTemp
group by cantidadCompras
order by 2 DESC
```

19. Desconectarse de la sesión. Volver a conectarse y ejecutar SELECT \* from #ordenesTemp.  
Que sucede?

20. Se desea 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

```
select c.state, c.city, count(*)
from customer c
where c.company like '%ts%'
and c.zipcode between 93000 and 94100
and city != 'Mountain View'
group by c.state, c.city
order by c.city
```

21. Para cada estado, obtener la cantidad de clientes referidos. Mostrar sólo los clientes que hayan sido referidos cuya compañía empiece con una letra que este en el rango de 'A' a 'L'.

```
select c.state, count(*)
from customer c
where c.company like '[A-L]%'
and c.customer_num_referredBy IS NOT NULL
group by c.state
order by 1
```

22. Se desea obtener el promedio de lead\_time por cada estado, donde los Fabricantes tengan una 'e' en manu\_name y el lead\_time sea 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
```

### Práctica de INSERT, UPDATE y DELETE.

1. Crear una tabla temporal #clientes a partir de la siguiente consulta:

```
SELECT * FROM customer
```

```
select * into #clientes from customer
```

2. Insertar el siguiente cliente en la tabla #clientes

Customer_num	144
Fname	Agustín
Lname	Creevy
Company	Jaguares SA
State	CA
City	Los Angeles

```
INSERT INTO #clientes(customer_num, fname, lname, company, state, city) VALUES
(144, 'Agustin', 'Creevy', 'Jaguares SA', 'CA', 'Los Angeles')
```

3. Crear una tabla temporal #clientesCalifornia con la misma estructura de la tabla customer. Realizar un insert masivo en la tabla #clientesCalifornia con todos los clientes de la tabla customer cuyo state sea CA.

```
CREATE TABLE #clientesCalifornia(
customer_num smallint NOT NULL PRIMARY KEY,
fname varchar(15),
lname varchar(15),
company varchar(20),
address1 varchar(20),
address2 varchar(20),
city varchar(15),
state char(2),
zipcode char(5),
phone varchar(18),
status char(1)
)
```

```
INSERT INTO #clientesCalifornia
SELECT * FROM customer WHERE state='CA'
```

OTRA FORMA MAS RAPIDA

```
SELECT * INTO #clientesCalifornia
FROM customer WHERE state='CA'
```

4. Insertar el siguiente cliente en la tabla #clientes un cliente que tenga los mismos datos del cliente 103, pero cambiando en customer\_num por 155 Valide lo insertado.

```
INSERT INTO #clientes
(customer_num, fname, lname, company, address1, address2, city, state,
zipcode, phone)
SELECT 155, fname, lname, company, address1, address2, city, state, zipcode, phone
FROM customer
WHERE customer_num=103
```

5. Borrar de la tabla #clientes los clientes cuyo campo zipcode esté entre 94000 y 94050 y la ciudad comience con 'M'. Validar los registros a borrar antes de ejecutar la acción.

```
SELECT COUNT(*) FROM #clientes
WHERE zipcode BETWEEN 94000 AND 94050
AND city LIKE 'M%'

DELETE FROM #clientes
WHERE zipcode BETWEEN 94000 AND 94050
AND city LIKE 'M%'
```

6. Borrar de la tabla #clientes todos los clientes que no posean órdenes de compra en la tabla orders. (Utilizar un subquery).

```
SELECT * FROM #clientes
WHERE customer_num NOT IN (SELECT DISTINCT customer_num FROM orders)
DELETE FROM #clientes
WHERE customer_num NOT IN (SELECT DISTINCT customer_num FROM orders)
```

7. Modificar los registros de la tabla #clientes cambiando el campo state por 'AK' y el campo address2 por 'Barrio Las Heras' para los clientes que vivan en el state 'CO'. Validar previamente la cantidad de registros a modificar.

```
SELECT COUNT(customer_num) from #clientes WHERE state='CO'

UPDATE #clientes SET state='AK', address2='Barrio Las Heras'
WHERE state='CO'
```

8. Modificar todos los clientes de la tabla #clientes, agregando un dígito 1 delante de cada número telefónico, debido a un cambio de la compañía de teléfonos.

```
UPDATE #clientes SET phone='1'+phone
```

9. Comenzar una transacción, dentro de ella realizar:
- Insertar un registro en la tabla #clientes con los siguientes 4 datos
    - Customer\_num 166
    - Lname apellido
    - State CA
    - Company nombre empresa
  - Borrar los registros de la tabla #clientesCalifornia

Consultar los datos de las tablas #clientes y #clientesCalifornia, y asegurarse de que se haya realizado las operaciones.

Realizar un ROLLBACK y volver a chequear la información, que pasó??

```
BEGIN TRANSACTION
INSERT INTO #clientes (customer_num, lname, state, company)
VALUES (166, 'apellido', 'CA', 'nombre empresa')
DELETE FROM #clientesCalifornia

SELECT * FROM #clientes WHERE customer_num=166
SELECT COUNT(*) FROM #clientesCalifornia

ROLLBACK TRANSACTION

SELECT * FROM #clientes WHERE customer_num=166
SELECT COUNT(*) FROM #clientesCalifornia
```

10. Ejecutar la misma transacción del punto 9.

Realizar un COMMIT y volver a chequear la información, que pasó??

```
BEGIN TRANSACTION
INSERT INTO #clientes (customer_num, lname, state, company)
VALUES (166, 'apellido', 'CA', 'nombre empresa')
DELETE FROM #clientesCalifornia

SELECT * FROM #clientes WHERE customer_num=166
```

```
SELECT COUNT(*) FROM #clientesCalifornia
```

```
COMMIT TRANSACTION
```

```
SELECT * FROM #clientes WHERE customer_num=166
```

```
SELECT COUNT(*) FROM #clientesCalifornia
```