

1. **Lista todo los datos en la "classic models database": a) Product Lines (7) b) Product (110); c) Employees (23) d) Oces (7) e) Customers (122) f) Orders (326) g) Orderdetails (2996) h) Payments (273)**

```
Select *  
from productlines , products , employees , offices , customers , orders , orderdetails,  
payments;
```

2. **Seleccione el nombre del cliente (customer name) en la tabla customer. Ordenar por nombre de cliente (customer) (122)**

```
SELECT  
*  
FROM  
    customers  
ORDER BY  
    customerName asc;
```

3. **Muestra cada uno de los diferentes estados (status) en los que puede estar un pedido (order) (6)**

```
SELECT DISTINCT  
    status  
FROM  
    orders;
```

4. **Escriba el nombre (firstname) y apellido (lastname) de cada empleado. Ordenar por apellido y luego nombre (23)**

```
SELECT  
    CONCAT(firstName,'',lastName) fullname  
FROM  
    employees
```

5. **Muestra todos los puestos de trabajo (job titles) de los empleados (7)**

```
SELECT DISTINCT  
    jobTitle  
FROM  
    employees;
```

6. **Muestra todos los productos (products) junto con su escala de productos (product scale) (110)**

```
SELECT productScale  
FROM products;
```

7. Muestra todos los territorios (territories) donde tenemos oficinas (offices) (4)

```
SELECT DISTINCT territory  
FROM offices;
```

8. seleccione el nombre del contacto, el apellido del contacto y el límite de crédito para todos los clientes en los que haya crédito límite > 50000 (85)

```
SELECT contactfirstname, contactlastname, creditlimit  
FROM Customers  
WHERE creditlimit > 50000;
```

9. Seleccione los clientes (customers) que no tienen un límite de crédito (credit limit) (0.00) (24)

```
SELECT contactfirstname, contactlastname  
FROM Customers  
WHERE creditlimit = 0;
```

10. Muestra todas las oficinas (offices) que no están en los EE. UU. (USA) (4)

```
SELECT officecode  
FROM Offices  
WHERE country != 'USA';
```

11. Muestra los pedidos (orders) realizados entre el 16 de junio de 2014 y el 7 de julio de 2014 (8)

```
SELECT ordernumber  
FROM Orders  
WHERE orderdate BETWEEN '2014-06-16' AND '2014-07-07';
```

12. Muestra los productos que necesitamos pedir de nuevo (cantidad de existencias < 1000) (12)

```
SELECT productname  
FROM Products  
WHERE quantityinstock < 1000;
```

13. Muestra todos los pedidos (orders) que se enviaron (shippeddate) después de la fecha requerida (1)

```
SELECT ordernumber  
FROM Orders  
WHERE shippeddate > requireddate;
```

14. Muestra todos los clientes (customers) que tengan la palabra "Mini" en su nombre (10)

```
SELECT customername  
FROM Customers  
WHERE customername like '%Mini%';
```

15. Muestra todos los productos suministrados por "Highway 66 Mini Classics" (9)

```
SELECT productname  
FROM Products  
WHERE productVendor = 'Highway 66 Mini Classics';
```

16. Muestra todos los productos no suministrados por "Highway 66 Mini Classics" (101)

```
SELECT productname  
FROM Products  
WHERE productvendor != 'Highway 66 Mini Classics';
```

17. Lista de todos los empleados(employees) que no tienen un gerente(manager) (1)

```
SELECT employeeNumber, firstname, lastname  
FROM Employees  
WHERE reportsto IS NULL;
```

18. Muestre cada pedido junto con los detalles de ese pedido para los números de pedido 10270, 10272, 10279 (23) Sugerencia: esto se puede hacer de dos maneras. Prueba los dos. ¾Cuál es más fácil si tienes una gran cantidad de criterios de selección?

```
SELECT ordernumber, productcode, quantityordered, priceeach, orderlinenumber  
FROM Orders NATURAL JOIN OrderDetails  
WHERE ordernumber IN(10270, 10272,10279);
```

19. Lista de líneas de productos (productline) y proveedores(vendors) que suministran los productos en esa línea de productos. (65)

```
SELECT DISTINCT productline, productvendor  
FROM ProductLines NATURAL JOIN Products;
```

20. Selecciona los clientes(customeres) que viven en el mismo estado (state) que una de nuestras ocinas (26)

```
SELECT customername  
FROM Customers INNER JOIN Offices  
ON Customers.state = Offices.state;
```

- 21. Selecciona los clientes(customeres) que viven en el mismo estado en que trabaja su representante sindical (employee representative) (26).**

```
SELECT DISTINCT customername
FROM Customers INNER JOIN (Employees NATURAL JOIN Offices AS ER)
ON Customers.state = ER.state;
```

- 22. Seleccione customerName, orderDate, quantityOrdered, productLine, productName para todos pedidos(orders) realizados y enviados (shipped) en 2005 (444)**

```
SELECT customername, orderdate, quantityordered, productline, productname
FROM Orders NATURAL JOIN Customers NATURAL JOIN OrderDetails NATURAL
JOIN Products
WHERE orderdate BETWEEN '2005-01-01' AND '2005-12-31'
AND shippeddate BETWEEN '2005-01-01' AND '2005-12-31';
```

- 23. Lista de productos que no se vendieron (1)**

```
SELECT productname, ordernumber
FROM Products LEFT OUTER JOIN OrderDetails
ON Products.productcode = OrderDetails.productcode
WHERE ordernumber IS NULL;
```

- 24. Muestra todos los clientes(customer) y sus representantes de ventas(sales rep); lístalos incluso si no tienen un representante de ventas (122)**

```
SELECT customername, lastname, firstname
FROM Customers LEFT OUTER JOIN Employees
ON Customers.salesrepemployeenumber = Employees.employeenumber;
```

- 37. Lista de todos los clientes que no hicieron pedidos en 2005 (78)**

aquí presenta un error en el que no llego sacar el resultado que pide. He encontrado ejemplos con EXCEPT pero en mysql no existe. Lo sustituye NOT IN pero no logrado llegar a 78

- 38. SELECT DISTINCT customerName
From Customers NATURAL JOIN Orders
WHERE YEAR(orderDate) = 2005;**

- 39. Haga una lista de todas las personas con las que tratamos (empleados y contactos de clientes (customer contact)). Mostrar nombre (rst name), apellido (last name), nombre de la empresa (company name) (o empleado) (145)**

```
SELECT 'Employee' AS "Company Name", lastName AS "Last Name", firstName AS  
"First Name"  
FROM Employees  
UNION  
SELECT customerName AS "Company Name", contactLastName AS "Last Name",  
contactFirstName AS "First Name"  
FROM Customers;
```

- 40. Muestra el apellido, el nombre y el número de empleado (employee number) de todos los empleados que no tienen ningún cliente. Ordene primero por apellido, luego el nombre. (8)**

```
SELECT E.lastname, E.firstName, E.employeeNumber  
FROM Employees E  
LEFT OUTER JOIN Customers C  
ON E.employeeNumber = C.salesrepemployeenumber  
WHERE C.customerNumber IS NULL  
ORDER BY lastname, firstname;
```

- 41. Muestra el Código y nombre del producto para cada producto que nunca ha estado en un pedido de más de 48 unidades. Ordenar por producto Nombre. (8)**
*******Encontré este resultado**

```
SELECT productCode, productName FROM Products  
EXCEPT  
SELECT DISTINCT productCode, productName FROM OrderDetails NATURAL JOIN  
Products WHERE quantityOrdered > 48;
```

En este ejercicio se presenta un problema porque intento usar NOT IN para sustituir EXCEPT y no da resultado que pide a la hora de realizar la ejecucion.

- 42. Muestra el nombre y el apellido de cualquier cliente que solicitó cualquier producto cd cualquiera de las líneas de productos "Trenes" o "Camiones y autobuses". No use una "or". En su lugar realiza una UNION. Ordenar por el nombre del cliente. (61)**

```
SELECT DISTINCT contactlastname, contactfirstname  
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails NATURAL  
JOIN Products  
WHERE productLine = 'Trains'  
UNION  
SELECT DISTINCT contactlastname, contactfirstname
```

```

FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails NATURAL
JOIN Products
WHERE productLine = 'Trucks and Buses'
ORDER BY contactlastname, contactfirstname;

```

- 43. Indique el nombre de todos los clientes que no viven en el mismo estado y país que cualquier otro cliente. No uses count() para este ejercicio. Ordenar por el nombre del cliente. Devuelve 17 si convierte los estados Null en algo así como "N / A" y devuelve 10 si excluye a todos los clientes sin ningún valor en state. (61)**

```

SELECT DISTINCT contactlastname, contactfirstname
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails NATURAL
JOIN Products
WHERE productLine = 'Trains'
UNION
SELECT DISTINCT contactlastname, contactfirstname
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails NATURAL
JOIN Products
WHERE productLine = 'Trucks and Buses'
ORDER BY contactlastname, contactfirstname;

```

- 44. Qué producto nos hace ganar más dinero (qty * price)? (1)**

```

SELECT productName
FROM Products NATURAL JOIN OrderDetails
GROUP BY productName
HAVING SUM(quantityOrdered*priceEach) =
(
SELECT MAX(LIST_OF_PRODUCT_TOTALS.productTotal)
FROM
(
SELECT productCode, sum(quantityOrdered*priceEach) AS productTotal
FROM OrderDetails
GROUP BY productCode
) AS LIST_OF_PRODUCT_TOTALS
);

```

45. Muestra las líneas de productos (junto con su proveedor) que tienen <5 proveedores (3)

```
SELECT productLine, productVendor
FROM Products
WHERE productLine = (SELECT productLine FROM Products
GROUP BY productLine
HAVING count(productVendor) < 5);
```

46. Haz un listado con los productos de la línea de productos que tiene el mayor número de productos (38)

```
SELECT productName, productLine
FROM Products
WHERE productLine =
(
SELECT productLine AS numOfProductsPerProductLine
FROM Products
GROUP BY productLine
HAVING count(productName) =
(
SELECT MAX(COUNTTABLE.numOfProductsPerProductLine) AS maxcount
FROM
(
SELECT productLine, count(productName) AS numOfProductsPerProductLine
FROM Products
GROUP BY productLine
) AS COUNTTABLE
)
);
```

47. Encuentra el nombre y el apellido de todos los contactos de clientes cuyo cliente se encuentra en el mismo estado que la oficina de San Francisco. (11)

```
SELECT contactfirstname, contactlastname FROM Customers
WHERE state =
(
SELECT DISTINCT state FROM Customers
WHERE city = 'San Francisco'
);
```

48.Cuál es el cliente y el vendedor del pedido más caro? (1)

```
SELECT customerName, salesRepEmployeeNumber
FROM Customers
WHERE customerNumber =
(
SELECT customerNumber
FROM Orders
WHERE orderNumber =
(
SELECT orderNumber
FROM OrderDetails
GROUP BY orderNumber
HAVING sum(priceEach*quantityOrdered) =
(
SELECT MAX(OrderTotals.orderTotal)
FROM
(
SELECT sum(priceEach*quantityOrdered) AS orderTotal
FROM OrderDetails
GROUP BY orderNumber
) AS OrderTotals
)
)
);
```

**49. ¿Cuál es el número de pedido y el costo del pedido, para los pedidos más caros?
Nota que podría haber más de un pedido que suman el mismo costo, y ese mismo costo podría ser el costo más alto entre todos los pedidos. (1)**

```
SELECT orderNumber, sum(priceEach*quantityOrdered) AS "Order Total"
FROM OrderDetails
GROUP BY orderNumber
HAVING sum(priceEach*quantityOrdered) =
(
SELECT MAX(OrderTotals.orderTotal)
FROM
(
SELECT sum(priceEach*quantityOrdered) AS orderTotal
FROM OrderDetails
GROUP BY orderNumber
) AS OrderTotals
);
```

50.Cuál es el nombre del cliente, el número de pedido y el costo total de la mayoría pedidos caros? (1) * Difícil *

```
SELECT customerName, orderNumber, sum(priceEach*quantityOrdered) AS "Order
```



```

Total"
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails
WHERE customerNumber =
(
SELECT customerNumber
FROM Orders
WHERE orderNumber =
(
SELECT orderNumber
FROM OrderDetails
GROUP BY orderNumber
HAVING sum(priceEach*quantityOrdered) =
(
SELECT MAX(OrderTotals.orderTotal)
FROM
(
SELECT sum(priceEach*quantityOrdered) AS orderTotal
FROM OrderDetails
GROUP BY orderNumber
) AS OrderTotals
)
)
)
AND orderNumber =
(
SELECT orderNumber
FROM OrderDetails
GROUP BY orderNumber
HAVING sum(priceEach*quantityOrdered) =
(
SELECT MAX(OrderTotals.orderTotal)
FROM
(
SELECT sum(priceEach*quantityOrdered) AS orderTotal
FROM OrderDetails
GROUP BY orderNumber
) AS OrderTotals
)
)
GROUP BY customerName, orderNumber;

```

51. Realice la consulta anterior utilizando una vista. (1) * Difícil *

```

CREATE VIEW V_ORDERNUMBER_OF_HIGHEST_PRICED_ORDER AS
(
SELECT orderNumber
FROM OrderDetails
GROUP BY orderNumber
HAVING sum(priceEach*quantityOrdered) =
(
SELECT MAX(OrderTotals.orderTotal)

```

```

FROM
(
SELECT sum(priceEach*quantityOrdered) AS orderTotal
FROM OrderDetails
GROUP BY orderNumber
) AS OrderTotals
);

SELECT customerName, orderNumber, sum(priceEach*quantityOrdered) AS "Order
Total"
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails
WHERE customerNumber =
(
SELECT customerNumber
FROM Orders
WHERE orderNumber = (SELECT ordernumber FROM
V_ORDERNUMBER_OF_HIGHEST_PRICED_ORDER)
)
AND orderNumber = (SELECT ordernumber FROM
V_ORDERNUMBER_OF_HIGHEST_PRICED_ORDER)
GROUP BY customerName, orderNumber;

```

- 52. Muestra todos los clientes que han pedido al menos un producto con el nombre "Ford" en él, que también se ha pedido "Dragon Souvenirs, Ltd.". Listarlos en orden alfabético inverso, sin importar mayúsculas y/o minúsculas en el nombre del cliente que aparece en el pedido. Mostrar a cada cliente sólo una vez. (1)**

```

SELECT customerName, orderNumber, sum(priceEach*quantityOrdered) AS "Order
Total"
FROM Customers NATURAL JOIN Orders NATURAL JOIN OrderDetails
WHERE customerNumber =
(
SELECT customerNumber
FROM Orders
WHERE orderNumber = (SELECT ordernumber FROM
V_ORDERNUMBER_OF_HIGHEST_PRICED_ORDER)
)
AND orderNumber = (SELECT ordernumber FROM
V_ORDERNUMBER_OF_HIGHEST_PRICED_ORDER)
GROUP BY customerName, orderNumber;

```

- 53. Qué productos tienen un MSRP dentro del 5% del MSRP (precio de venta aconsejado por el fabricante) promedio en todos los productos? Indique el nombre del producto, el MSRP y el MSRP promedio ordenado por el producto MSRP. (14)**

```

SELECT productname, msrp
FROM products

```

```
where msrp <= 1.05*(SELECT AVG(msrp) FROM Products) AND msrp >=
0.95*(SELECT AVG(msrp) FROM Products);
```

- 54. Encuentre clientes que hayan ordenado lo mismo. Encuentra solo aquellos pares de clientes que han ordenado lo mismo que el otro al menos 201 veces**

```
SELECT employeeNumber, lastName, firstName
FROM Employees
WHERE employeeNumber IN
(
SELECT reportsto
FROM Employees
GROUP BY reportsto
HAVING COUNT(employeeNumber) =
(
SELECT MAX(Manager.numOfEmployeesManaged)
FROM
(
SELECT reportsto, COUNT(employeeNumber) AS numOfEmployeesManaged
FROM Employees
GROUP BY reportsto
) AS Manager
)
);
```

- 62. Crea una vista para los clientes que viven en Nueva York.**

```
CREATE VIEW customerscity
AS
SELECT customerName, city, addressLine1
FROM customers
WHERE city = 'NYC';
```

```
select *
from customerscity;
```

- 63. Escriba una consulta para crear una vista para todos los clientes, con columnas salesmanid, name y city.**

```
CREATE VIEW customersall
As
SELECT salesRepEmployeeNumber, customerName, city
FROM customers;
```

```
select *
from customersall;
```

64. Escribe una consulta para encontrar al cliente que vive en Nueva York con mejor crédito.

```
CREATE VIEW customercity
AS
SELECT customerName, city, creditLimit
FROM customers
WHERE city = 'NYC'
GROUP BY customerName, city ,creditLimit;

SELECT max(creditLimit)
from customercity;
```

65. Escriba una consulta para crear una vista para obtener un recuento de cuántos clientes tenemos en cada país.

```
CREATE VIEW customerscountry
As
SELECT customerName, country
FROM customers;

SELECT country, count(*) as total FROM customerscountry GROUP BY country
```

66. Escriba una consulta para crear una vista para realizar un seguimiento de la cantidad de pedidos de clientes, la cantidad de vendedores adjuntos (salesRepemployeenumber) , la cantidad promedio de pedidos y la cantidad total de pedidos en un día.

```
CREATE VIEW customerdel
AS
select customerName, salesRepEmployeeNumber
from customers;

SELECT salesRepEmployeeNumber, count(*) as total FROM customerdel GROUP
BY salesRepEmployeeNumber;
```

