

Consultas en SQL SERVER

NORTHWIND

1. Listar el id de la categoría, el id del producto y el nombre, de aquellos productos de categorías 2 a 4, clasificados por categoría y dentro de la categoría por id del producto.

```
SELECT CategoryID, ProductID, ProductName  
FROM Products  
WHERE CategoryID BETWEEN 2 and 4  
ORDER BY CategoryID, ProductID
```

2. Listar la categoría, el id y el nombre de los productos de categorías 1,4,5,7, clasificados por categoría y dentro de la categoría por id del producto.

```
SELECT CategoryID, ProductID, ProductName  
FROM Products  
WHERE CategoryID IN (1,4,5,7)  
ORDER BY CategoryID, ProductID
```

3. Listar la categoría, el id y el nombre de los productos que empiecen por Ch, clasificados por categoría y dentro de la categoría por id del producto.

```
SELECT CategoryID, ProductID, ProductName  
FROM Northwind.dbo.Products  
WHERE ProductName LIKE 'Ch%'  
ORDER BY CategoryID, ProductID
```

4. Listar el nombre de la compañía, ciudad, región y country de la tabla Clientes donde la región del cliente no es NULL (función Is Null/Is Not Null).

```
SELECT CompanyName, City, Region, Country  
FROM Northwind.dbo.Customers  
WHERE Region IS NOT NULL  
ORDER BY CompanyName
```

5. Recuperar todos los productos en los que la cantidad de productos almacenados (UnitsInStock) sea inferior al valor ReorderLevel, o el producto venga del proveedor 15 (SupplierID) y sea de la categoría 4.

```
SELECT ProductID, ProductName  
FROM Northwind.dbo.Products  
WHERE UnitsInStock < ReorderLevel  
OR (SupplierID = 15 AND CategoryID = 4)
```

PUBS

6. Recuperar los autores ordenados alfabéticamente, que viven en la misma ciudad y estado en la que tiene sede alguna editorial

USE pubs

```
SELECT au_fname, au_lname, pub_name  
FROM authors as a INNER JOIN publishers as p ON (a.city = p.city  
AND a.state = p.state)  
ORDER BY a.au_lname
```

7. Listar los identificadores de títulos, el título y el número de libros vendidos (qty)

```
SELECT t.title_id, t.title, s.qty  
FROM titles AS t inner JOIN sales AS s ON t.title_id = s.title_id
```

8. Listar el identificador, el nombre y el estado de la editorial con todos los datos del autor que sean de la misma ciudad. (Se trata de obtener el id de editorial, nombre y estado de las editoriales que tengan sede o sucursal en ciudades en las que resida algún autor. Y obtener además todos los datos de dicho autor)

USE pubs

```
SELECT p.pub_id, p.pub_name, p.state, a.*  
FROM publishers p INNER JOIN authors a ON p.city = a.city  
ORDER BY a.au_lname ASC, a.au_fname ASC
```

9. Listar los autores de Oakland, que viven en el mismo área de código postal(zip) ordenados por código postal. (En definitiva se trata de obtener los autores de Oakland con colegas de igual zip)

USE pubs

```
SELECT distinct(au1.au_fname), au1.au_lname, au1.zip  
FROM authors au1 INNER JOIN authors au2 ON au1.zip = au2.zip  
WHERE au1.city = 'Oakland' and au1.au_fname <> au2.au_fname  
ORDER BY au1.zip ASC
```

```
select au_fname  
from authors au1  
where exists  
    (select au_fname  
     from authors au2  
     where au1.city = 'Oakland' and au1.au_fname <> au2.au_fname  
     and au1.zip = au2.zip)
```

10. Listar los títulos ordenados de manera alfabética y los nombres de sus autores, de todos los libros del tipo 'trad_cook'.

```
USE pubs  
SELECT a.au_lname, a.au_fname, t.title  
FROM authors a INNER JOIN titleauthor ta ON a.au_id = ta.au_id  
           JOIN titles t      ON ta.title_id = t.title_id  
WHERE t.type = 'trad_cook'  
ORDER BY t.title ASC
```

11. Calcular el promedio del adelanto(advance) y la suma de las ventas (ytd_sales), sólo de los libros de empresa(business).

```
USE pubs  
SELECT AVG(advance), SUM(ytd_sales)  
FROM titles  
WHERE type = 'business'
```

12. Contar el n° de adelantos (advance) de la tabla titles. En otra consulta contar los nombres de título. ¿Por qué tienen distintos resultados?.

```
USE pubs  
SELECT COUNT(advance)  
FROM titles --16
```

```
USE pubs  
SELECT COUNT(title)  
FROM titles
```

El resultado es distinto porque las funciones de agregación (count, sum, avg,...) ignoran los valores nulos, excepto count(*) que cuenta filas.

13. Listar los autores que viven en la misma ciudad en la que hay alguna editorial.

```
USE pubs  
SELECT au_lname, au_fname  
FROM authors  
WHERE city = ANY (SELECT city FROM publishers)
```

```
SELECT au_lname, au_fname  
FROM authors  
WHERE exists (SELECT *  
              FROM publishers  
              WHERE authors.city = publishers.city)
```

14. Listar los nombres de los editores que han publicado libros de psychology

```
USE pubs  
SELECT pub_name  
FROM publishers  
WHERE pub_id IN  
    (SELECT pub_id  
        FROM titles  
        WHERE type = 'psychology')
```

```
select distinct(pub_name)  
from publishers inner join titles on publishers.pub_id = titles.pub_id  
where type = 'psychology'
```

15. Listar el nombre de los editores que no han publicado libros de psychology.

```
SELECT pub_name  
FROM publishers  
WHERE NOT EXISTS (SELECT *  
                    FROM titles  
                    WHERE pub_id = publishers.pub_id AND type =  
'psychology')
```

16. Para cada tipo de título obtener el promedio del precio de todos los títulos con adelantos (advance) mayor de 5000.

```
USE pubs  
SELECT type, AVG(price)  
FROM titles  
WHERE advance > $5000  
GROUP BY type
```

17. Listar las ciudades en las que viven autores pero no editores

```
SELECT DISTINCT city  
FROM authors  
WHERE NOT EXISTS(SELECT *  
                  FROM publishers  
                  WHERE authors.city = publishers.city)
```

```

SELECT city
FROM Authors
WHERE NOT EXISTS
(SELECT * FROM Publishers
    WHERE Authors.city=Publishers.city)
GROUP BY city

```

18. Listar los identificativos de los editores que hayan tenido ventas (ytd_sales) hasta la fecha superiores a 4000 dólares.

```

SELECT pub_id, total = SUM(ytd_sales)
FROM titles
GROUP BY pub_id
HAVING SUM(ytd_sales) > 4000

```

19. Listar los identificativos de los editores y la suma de ventas (ytd_sales) que como mínimo tengan seis libros involucrados en los cálculos para cada editor. *(Se trata de obtener el id de la editorial y el total de ventas de aquellas editoriales que tengan publicados al menos 6 libros)*

```

USE pubs
SELECT pub_id, total = SUM(ytd_sales)
FROM titles
GROUP BY pub_id
HAVING COUNT(*) > 5

```

20. Busca los nombres de todos los libros con un precio superior al precio mínimo actual.

```

USE pubs
SELECT DISTINCT title
FROM titles
WHERE price >
    (SELECT MIN(price)
    FROM titles)

```

21. Listar los nombres de todos los libros con un precio superior al precio mínimo del tipo psychology.

```

USE pubs
SELECT DISTINCT title
FROM titles
WHERE price >
    (SELECT MIN(price)
    FROM titles
    GROUP BY type
    HAVING type = 'psychology')

```

22. Busca los títulos que recibieron un adelanto superior a alguno de los recibidos por los títulos publicados por 'New Moon Books'

```
USE pubs  
SELECT title  
FROM titles  
WHERE advance > ANY  
  (SELECT advance  
    FROM publishers INNER JOIN titles  
    ON titles.pub_id = publishers.pub_id  
    AND pub_name = 'New Moon Books')
```

NORTHWIND

23. Listar los identificadores de los pedidos y de los productos de la tabla Order Details donde la cantidad del producto enviado sea mayor que todas las cantidades enviadas de producto de la categoría 2.

```
SELECT OrdD1.OrderID, OrdD1.ProductID  
FROM "Order Details" OrdD1  
WHERE OrdD1.Quantity > ALL  
  (SELECT OrdD2.Quantity  
    FROM "Order Details" OrdD2 JOIN Products Prd  
    ON OrdD2.ProductID = Prd.ProductID  
    WHERE Prd.CategoryID = 2)
```

PUBSS

24. Listar los tipos de libros publicados por más de un editor.

```
USE pubs  
SELECT DISTINCT t1.type  
FROM titles t1  
WHERE t1.type IN  
  (SELECT t2.type  
    FROM titles t2  
    WHERE t1.pub_id <> t2.pub_id)
```