

Tema 5:

Lenguaje de Consulta de Datos. SQL.



**Bases
De
Datos**

Índice

1. Sentencia SELECT
2. Consultas multitable JOIN
3. Consultas Resumen



1.- Sentencia SELECT

- Cláusula SELECT. Determina las columnas que se mostrarán.
- Cláusula INTO. Determina si el resultado de la consulta se escribe en un fichero.
- Cláusula FROM. Determina las tablas origen para la consulta.
- Cláusula WHERE. Determina qué registros se seleccionan.
- Cláusula GROUP BY. Determina cómo se agrupan los registros.
- Cláusula HAVING. Determina que registros se van a mostrar.
- Cláusula ORDER BY. Determina el orden en el que se muestran los registros resultantes.
- Cláusula LIMIT. Permite seleccionar una sección de filas del resultado

1.- Sentencia SELECT

EJEMPLOS DE USO

1. Uso de alias para columnas

SELECT Nombre as NOMBRE_EMPLEADO FROM Empleado

2. Uso de alias obligatorio para funciones de agregado o nombres ambiguos/duplicados

SELECT AVG(Edad) AS EDAD_MEDIA FROM Empleado

3. Obtener todos los campos

SELECT * FROM Empleado

4. Obtener algunos campos

SELECT Nombre, Edad FROM Empleado

1.- Sentencia SELECT

5. Obtener los datos ordenados

SELECT Nombre, Edad , Alojamiento FROM Empleado ORDER BY Edad
SELECT Nombre, Edad , Alojamiento FROM Empleado ORDER BY Edad
ASC

SELECT Nombre, Edad , Alojamiento FROM Empleado ORDER BY Edad
DESC

6. Obtener los datos agrupados

SELECT Nombre, Edad , Alojamiento FROM Empleado GROUP BY Edad,
Alojamiento

7. Eliminar filas duplicadas

SELECT DISTINCT Alojamiento FROM Empleado

1.- Sentencia SELECT

8. Condición de búsqueda

SELECT * FROM Curso WHERE ctarifa<150

SELECT * FROM Curso WHERE cnombre >="RACIONALISMO"

SELECT ciudad, ventas, objetivo FROM Oficinas WHERE ventas>objetivo

SELECT nombre, ventas, cuota FROM repventas WHERE numempl= 105

9. Eliminar filas duplicadas

SELECT dir FROM oficinas

SELECT DISTINCT dir FROM oficinas

SELECT DISTINCT ctarifa FROM curso

10. Ordenación de resultados

SELECT * FROM personal ORDER BY enombre

**SELECT dept, cargo, sueldo FROM personal ORDER BY dept, sueldo
DESC**

SELECT ciudad, region, ventas FROM oficinas ORDER BY region, ciudad

1.- Sentencia SELECT

SELECT ciudad, region, ventas-objetivo AS RENDIMIENTO FROM oficinas ORDER BY 3 DESC

SELECT ciudad, region, ventas-objetivo AS RENDIMIENTO FROM oficinas ORDER BY region ASC, 3 DESC

11. Columnas calculadas

SELECT nombre, esueldo*100 AS NUEVO_SUELDO FROM personal

**SELECT nombre, esueldo*100 AS NUEVO_SUELDO FROM personal
WHERE esueldo-100 <25000**

**SELECT idfab, idproducto, descripcion , existencias*precio AS
INVENTARIO FROM productos**

**SELECT nombre , cuota, cuota+ 0.003*ventas AS CUOTA_ELEVADA
FROM repventas**

1.- Sentencia SELECT

```
SELECT * FROM clientes WHERE empresa LIKE '_e%'
```

```
SELECT FNOMBRE, FNUMDEP, FDEPT FROM CLAUSTRO WHERE  
fnumdep IS NOT NULL
```

CONTRASTES COMPUESTOS: AND, OR, NOT

```
SELECT * FROM curso WHERE (CTARIFA >=100) AND (ctarifa<=500)
```

SELECT esueldo FROM personal UNION SELECT fsueldo FROM claustro

```
SELECT idfab, idproducto FROM productos WHERE precio >2000 UNION  
select DISTINCT fab, producto FROM PEDIDOS where IMPORTE >3000
```

1.- Sentencia SELECT

```
SELECT idfab, idproducto FROM productos WHERE precio >2000 UNION  
ALL SELECT fab, producto FROM PEDIDOS where IMPORTE >3000
```

```
SELECT idfab, idproducto FROM productos WHERE precio >2000 UNION  
ALL SELECT DISTINCT fab, producto FROM PEDIDOS where IMPORTE  
>3000 ORDER BY 1,2
```

2.- Consultas multitabla JOIN

Consultas multitabla (JOIN)

Las consultas multitabla o JOINS permiten recuperar datos de 2 tablas o más según las relaciones lógicas entre ellas.

Una condición de consulta multitabla (combinación o composición) define la forma de relacionarse las tablas en la consulta:

- Especificar que columna en cada tabla debe usarse para la combinación. P.ej. La clave externa y su clave asociada en la otra tabla.
- Especificar un operador lógico para usarlo en los valores de comparación de las columnas.

2.- Consultas multitabla JOIN

EJEMPLOS DE USO

```
SELECT p.numpedido, p.importe, c.empresa, c.limitecredito FROM  
pedidos p INNER JOIN clientes c ON p.clie=c.numclie
```

(sintáxis ANSI)

Para evitar ambigüedades se puede calificar las columnas con los nombres de las tablas.

```
SELECT pedidos.numpedido, pedidos.importe, clientes.empresa,  
clientes.limitecredito FROM pedidos, clientes WHERE clie=numclie
```

(sintáxis SQL92)

Mejor sintáxis ANSI. Más compatibilidad y separación entre cláusulas ON y WHERE

2.- Consultas multitable JOIN

JOIN DE 2 TABLAS

```
SELECT * FROM curso INNER JOIN departamento ON  
curso.cdept=departamento.dept
```

JOIN DE 2 TABLAS CON CRITERIOS DE SELECCIÓN

```
SELECT c.nombre, ctarifa, dchfno FROM curso c INNER JOIN  
departamento d ON c.cdept=d.dept WHERE ctarifa>100
```

```
SELECT o.ciudad, r.nombre, r.titulo FROM repventas r INNER JOIN  
oficinas o ON o.dir=r.numempl WHERE o.objetivo<600000
```

JOIN DE 2 TABLAS CON MÚLTIPLES COLUMNAS DE EMPAREJAMIENTO

```
SELECT numpedido, importe, descripcion FROM pedidos INNER JOIN  
productos ON fab=idfab AND producto=idproducto;
```

2.- Consultas multitable JOIN

JOIN DE 3 o MÁS TABLAS

```
SELECT numpedido, importe, empresa, nombre FROM pedidos INNER  
JOIN clientes INNER JOIN repventas ON clie=numclie AND rep=numempl  
WHERE importe >25000
```

Para realizar una JOIN no es necesario que las columnas de emparejamiento tengan relación clave primaria/clave foránea

```
SELECT numpedido, importe, fechapedido, nombre FROM pedidos INNER  
JOIN repventas ON fechapedido=contrato
```

Tampoco es necesario que la relación entre las columnas sea la de igualdad

```
SELECT nombre, cuota, ciudad, objetivo FROM repventas INNER JOIN  
oficinas ON cuota > objetivo
```


2.- Consultas multitabla JOIN

JOIN INTERNO

SELECT campos FROM tabla1 INNER JOIN tabla2 ON tabla1.campo1
operadorComparación tabla2.campo2

```
SELECT nombre, titulo, objetivo, oficina, ciudad FROM repventas INNER  
JOIN oficinas ON oficinarep=oficina
```

Se pueden anidar instrucciones JOIN:

```
SELECT campos FROM tabla1 INNER JOIN (tabla2 INNER JOIN [tabla3
[INNER JOIN [ ( ] tablux [INNER JOIN....)]
ON tabla3.campo3 operadorComparación tablaX.campoX )]
ON tabla2.campo2 operadorComparación tabla3.campo3 )
ON tabla1.campo1 operadorComparación tabla2.campo2 )
```

```
SELECT nombre, titulo, objetivo, oficina, ciudad FROM repventas OUTER  
JOIN oficinas ON oficinarep=oficina
```

2.- Consultas multitabla JOIN

JOIN IZQUIERDO Y DERECHO

El LEFT JOIN se utiliza para crear una combinación (JOIN) externa izquierda. Incluirá todos los registros de la primera (parte izquierda) de dos tablas, aunque no haya valores coincidentes para los registros de la segunda tabla(parte derecha)

El RIGHT JOIN se utiliza para crear una combinación (JOIN) externa derecha. Incluirá todos los registros de la segunda (parte derecha) de dos tablas, aunque no haya valores coincidentes para los registros de la primera tabla(parte izquierda)

SELECT campos FROM tabla1 [LEFT | RIGHT] JOIN tabla2 ON tabla1.campo1
operadorComparación tabla2.campo2

**SELECT nombre, titulo, objetivo, oficina, ciudad FROM repventas LEFT
JOIN oficinas ON numempl=dir**

2.- Consultas multitabla JOIN

Una operación LEFT JOIN o RIGHT JOIN se puede anidar en una operación INNER JOIN, pero INNER JOIN no se puede anidar en otra operación LEFT JOIN o RIGHT JOIN.

Puede vincularse múltiples cláusulas ON.

**SELECT empresa, numpedido, importe FROM clientes INNER JOIN pedidos
ON numclie=clie AND numclie='2103'**

**SELECT empresa, numpedido, importe FROM clientes LEFT JOIN pedidos
ON numclie=clie AND numclie='2103'**

**SELECT empresa, numpedido, importe FROM clientes RIGHT JOIN
pedidos ON numclie=clie AND numclie='2103'**

3.- Consultas Resumen

Funciones de agregado en la lista de selección

Las funciones de agregado (SUM, AVG, STD, COUNT, COUNT(*), MAX, MIN, etc..) generan valores de resumen en los conjuntos de resultados de las consultas.

Pueden aplicarse a todas las filas de la tabla, a un subconjunto especificado en WHERE o a uno o varios grupos de filas con GROUP BY.

DISTINCT puede usarse para eliminar duplicados antes de realizar la operación de agregado.

EJEMPLOS:

```
SELECT AVG(ctarifa) AS MEDIA , MAX(ctarifa) AS MAXIMO,  
MIN(ctarifa) AS MINIMO FROM curso WHERE cdept="CIS" AND  
ctarifa>0
```

```
SELECT AVG(cuota) AS CUOTA_PROMEDIO , AVG(ventas) AS  
VENTAS_PROMEDIO, AVG(100*(ventas/cuota)) AS  
RENDIMIENTO_CUOTA_PROMEDIO FROM repventas
```

3.- Consultas Resumen

```
SELECT AVG(ctarifa) AS MEDIA , MAX(ctarifa) AS MAXIMO,  
MIN(ctarifa) AS MINIMO FROM curso WHERE cdept="CIS" AND  
ctarifa>0
```

```
SELECT AVG(cuota) AS CUOTA_PROMEDIO , AVG(ventas) AS  
VENTAS_PROMEDIO, AVG(100*(ventas/cuota)) AS  
RENDIMIENTO_CUOTA_PROMEDIO FROM repventas
```

```
SELECT SUM(cuota) "TOTAL" , SUM(ventas) "VENTAS TOTALES",  
MIN(cuota) "CUOTA MINIMA", MAX(cuota) "CUOTA MAXIMA" FROM  
repventas
```


3.- Consultas Resumen

Valores NULL en funciones de agregado

Tratamiento de filas duplicadas con DISTINCT

Agrupar filas con GROUP BY

La cláusula GROUP BY permite producir valores de agregado para cada conjunto de filas de resultados

**SELECT cdept AS DEPARTAMENTO, SUM(cred) AS CREDITOS FROM
curso GROUP BY cdept**

**SELECT rep AS VENDEDOR, clie AS CLIENTE, SUM(importe) AS
PEDIDO_TOTAL FROM pedidos GROUP BY rep, clie**

