

U.D. 6. SQL. DML.



BBDD

Nuria Fuentes Pérez

INDICE

1. Consultas simples.
2. Funciones Oracle.
3. Diccionario de datos Oracle.
4. Consultas varias tablas.
 1. Subconsultas.
 2. Combinaciones (Join).
5. Consultas de totales.
6. Consultas de agrupamiento.

1. Consultas simples.

SELECT [ALL | DISTINCT]

{ * |

[usuario.]tabla.* |

expresion [alias] [, expresion [alias]] ...}

FROM [usuario.]tabla [t_alias] [, [usuario.] [tabla [t_alias]]] ...

[**WHERE** condición]

[**ORDER BY** { expresion | posicion } [ASC | DESC]

[, { expresion | posicion } [ASC | DESC]] ... ;

1. Consultas simples.

<Condición> Es una expresión booleana correcta que al evaluarse puede valer TRUE, FALSE o NULL. Además de los operadores relacionales.

Test de pertenencia a un conjunto, busca si un valor esta en una lista o no.

[NOT] IN <ListaValores> La lista de valores entre (,,).

Test de correspondencia con patrón, busca PATRON que puede contener { % | _ }

[NOT] LIKE <Valor1>

Test de valor nulo, Si el contenido del campo es nulo (no cero ni vacío)

IS [NOT] NULL

Test de rango, Entre un rango de valores, dos valores específicos

[NOT] BETWEEN <Valor1> AND <Valor2>

Condiciones compuestas:

NOT , AND , OR

DDL.- Crear tabla desde consulta.

```
CREATE TABLE AS consulta;
```

```
CREATE TABLE VENDEDORES  
AS SELECT EMP_NO, APELLIDO, SALARIO  
FROM EMPL  
WHERE OFICIO='VENDEDOR';
```

2. Funciones de Oracle.

- Aritméticas, de valores simples
- Funciones de grupos de valores
- Funciones de listas
- Funciones de cadenas que devuelven valores carácter
- Funciones de cadenas que devuelven valores numéricos
- Funciones para el manejo de fechas
- Funciones de conversión

3. Diccionario de datos de Oracle.

Diccionario de datos ORACLE.

B.B.D.D.



Nuria Fuentes Pérez

4.1. Consultas varias tablas. Subconsultas.

- **Subconsultas que generan valores simples**

```
SELECT <listaCampos> FROM <tabla>  
WHERE campo = (SELECT ...)
```

- **Subconsultas que generan lista de valores**

```
SELECT <listaCampos> FROM <tabla>  
WHERE campo IN (SELECT ...)
```

```
SELECT <listaCampos> FROM <tabla>  
WHERE EXISTS (SELECT...);
```

- **Referencia externa**

```
SELECT CIUDAD FROM PROVEEDORES  
WHERE TOTAL < (SELECT AVG(CANTIDAD)  
                FROM PEDIDOS  
                WHERE PEDIDOS.COD_S=PROVEEDORES.COD_S);
```

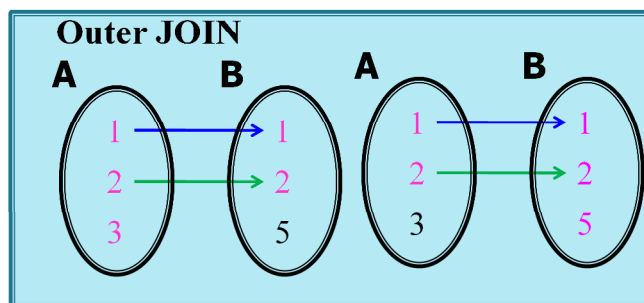
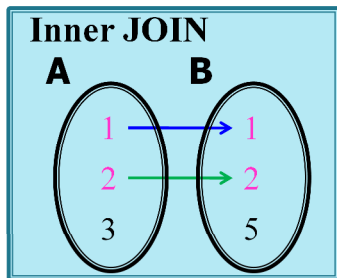
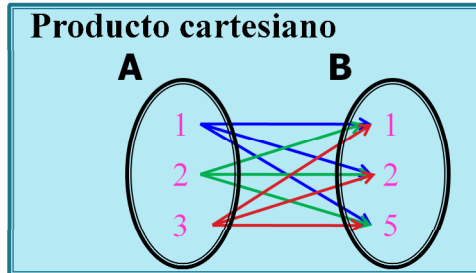
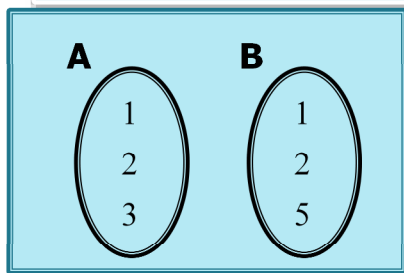

4.1. Consultas varias tablas. Subconsultas.

```
SELECT DNOMBRE FROM DEPART  
WHERE DEPT_NO =(SELECT DEPT_NO FROM EMPL  
WHERE APELLIDO='REY');
```

```
SELECT APELLIDO FROM EMPL  
WHERE DEPT_NO IN (SELECT DEPT_NO FROM DEPART  
WHERE DNOMBRE LIKE '%N');
```

```
SELECT DEPT_NO,DNOMBRE FROM DEPART  
WHERE DEPT_NO NOT IN (SELECT DISTINCT DEPT_NO FROM EMPL);
```

4.2. Consultas varias tablas. JOIN.



B.B.D.D.

 **CENTRO FP
SUPERIOR**
CAMPUS CÁMARA COMERCIO SEVILLA

Nuria Fuentes Pérez

4.2. Consultas varias tablas. JOIN.

PERSONA

num	nombre
1000	Ana
2000	Pepe
4000	Luisa

SECCION

id	nombre	localizacion
1000	Ventas	Sevilla
2000	Marketing	Madrid
3000	Contabilidad	Madrid

Producto cartesiano

Num	Nombre	Id	Nombre	localizacion
1000	Ana	1000	Ventas	Sevilla
1000	Ana	2000	Marketing	Madrid
1000	Ana	3000	Contabilidad	Madrid
2000	Pepe	1000	Ventas	Sevilla
2000	Pepe	2000	Marketing	Madrid
2000	Pepe	3000	Contabilidad	Madrid
4000	Luisa	1000	Ventas	Sevilla
4000	Luisa	2000	Marketing	Madrid
4000	Luisa	3000	Contabilidad	Madrid

SELECT * FROM PERSONA,SECCION;

B.B.D.D.



Nuria Fuentes Pérez

4.2. Consultas varias tablas. JOIN.

PERSONA

num	nombre
1000	Ana
2000	Pepe
4000	Luisa

Inner JOIN

Num	Nombre	Id	Nombre	localizacion
1000	Ana	1000	Ventas	Sevilla
2000	Pepe	2000	Marketing	Madrid

```
SELECT * FROM PERSONA,SECCION  
WHERE NUM=ID;
```

SECCION

id	nombre	localizacion
1000	Ventas	Sevilla
2000	Marketing	Madrid
3000	Contabilidad	Madrid

4.2. Consultas varias tablas. JOIN.

PERSONA

num	nombre
1000	Ana
2000	Pepe
4000	Luisa

Outer JOIN PERSONA a SECCION

Num	Nombre	Id	Nombre	localizacion
1000	Ana	1000	Ventas	Sevilla
2000	Pepe	2000	Marketing	Madrid
4000	Luisa			

SECCION

id	nombre	localizacion
1000	Ventas	Sevilla
2000	Marketing	Madrid
3000	Contabilidad	Madrid

SELECT * FROM PERSONA,SECCION
WHERE NUM=ID(+);

4.2. Consultas varias tablas. JOIN.

PERSONA

num	nombre
1000	Ana
2000	Pepe
4000	Luisa

Outer JOIN SECCION a PERSONA

Num	Nombre	Id	Nombre	localizacion
1000	Ana	1000	Ventas	Sevilla
2000	Pepe	2000	Marketing	Madrid
		3000	Contabilidad	Madrid

SECCION

id	nombre	localizacion
1000	Ventas	Sevilla
2000	Marketing	Madrid
3000	Contabilidad	Madrid

```
SELECT * FROM PERSONA,SECCION
WHERE NUM(+)=ID;
```

B.B.D.D.



Nuria Fuentes Pérez

4.3. Consultas de totales.

FUNCIONES DE GRUPO

AVG (expr)	Valor medio de n, ignorando los valores nulos
COUNT (* col)	Cuenta el numero de filas que hay para una columna ignora nulos. * cuenta filas no nulas
SUM (expr)	Obtiene la suma de los valores.
MAX (expr)	Obtiene el valor máximo.
MIN (expr)	Obtiene el valor mínimo.
STDEV (expr)	Obtiene la desviación típica.

4.3. Consultas de totales.

Las consultas de totales son aquellas consultas que incluyen funciones de grupo pero no utilizan agrupamiento

Estas consultas siempre devuelven una única fila.

```
SELECT COUNT(*) FROM EMPLE;
```

```
SELECT COUNT(EMP_NO),COUNT(COMISION) FROM EMPLE;
```

```
SELECT MAX(SALARIO) FROM EMPLE;
```


4.4. Consultas de agrupamiento.

```
SELECT [ALL | DISTINCT]  
    { * |  
      [usuario.]tabla.* |  
      expresion [alias] [, expresion [alias] ] ...}  
FROM [usuario.]tabla [t_alias] [, [usuario.] [tabla [t_alias]] ] ...  
[WHERE condición]  
[CONNECT BY condicion [START WITH condicion] ]  
[GROUP BY expresion [, expresion] ... ] [HAVING condicion]  
[ORDER BY { expresion | posicion } [ASC | DESC]  
            [, { expresion | posicion } [ASC | DESC] ] ...
```

Orden de evaluación de las cláusulas:

WHERE

GROUP BY

HAVING

ORDER BY

B.B.D.D.



Nuria Fuentes Pérez

4.4. Consultas de agrupamiento.

```
SELECT DEPT_NO,MAX(SALARIO) FROM EMPL  
GROUP BY DEPT_NO;
```

```
SELECT DNOMBRE,COUNT(*) FROM EMPL,DEPART  
WHERE EMPL.DEPT_NO=DEPART.DEPT_NO  
GROUP BY EMPL.DEPT_NO,DNOMBRE;
```

```
SELECT DNOMBRE,COUNT(*) FROM EMPL,DEPART  
WHERE EMPL.DEPT_NO=DEPART.DEPT_NO  
GROUP BY EMPL.DEPT_NO,DNOMBRE  
HAVING COUNT(*)>3;
```

```
SELECT DNOMBRE,COUNT(*) FROM EMPL,DEPART  
WHERE DEPART.DEPT_NO=EMPL.DEPT_NO(+)  
GROUP BY DNOMBRE;
```

```
SELECT DNOMBRE,COUNT(EMP_NO) FROM EMPL,DEPART  
WHERE DEPART.DEPT_NO=EMPL.DEPT_NO(+)  
GROUP BY DNOMBRE;
```

B.B.D.D.



Nuria Fuentes Pérez