Unitat 6: Disseny Físic. DML. Consultes Simples.

Bases de Dades

DML

Llenguatge de manipulació de dades (DML). Fins ara hem vist el llenguatge DDL, per definir les estructures del BD. Ja tenim una realitat dissenyada i creada. Però ara hem de fer els més importants, donar-li vida.

DML és una part fonamental de SQL i ens serveix per:

Afegeix dades (INSERT)

Actualitza les dades (UPDATE)

Suprimeix les dades (DELETE)

Dades de consulta (SELECT)

Insereix dades

- La manera d'inserir dades en una taula és amb la clàusula INSERT INTO, la sintaxi completa és:
-

```
INSERT INTO <NOMBRE_TABLA>
[(<Campos>[{,<Campo>}])]
Values
(<Valores>)
```

- On veiem que els camps de la taula són opcionals, un exemple:
- INSERT INTO ALUMNOS

```
(DNI, NOMBRE, EDAD) -- Opcional
VALUES
('111C', 'PABLO', 30);
```

- És equivalent a, si la taula d'estudiants té 3 camps i es creen en aquest ordre:
-

```
INSERT INTO ALUMNOS
VALUES
('111C', 'PABLO', 30);
```

Actualitza les dades

- Amb INSERT aconseguim introduir dades a la Base de Dades, però si volem actualitzar o modificar les dades ja introduïdes hem d'utilitzar un altre element, l'ACTUALITZACIÓ.
-

UPDATE <NOMBRE TABLA>
SET CAMPO=<VALOR> [{ , CAMPO=<VALOR>}
[WHERE <CONDICIONES>]

UPDATE ALUMNOS
SET EDAD=25, TELEFONO=6111111111
WHERE DNI='111C';

Actualitzar els camps Edat de l'Estudiant i Telèfon 111C.

UPDATES COMPLEJAS

També podem obtenir les dades a actualitzar a partir d'una Consulta:

UPDATE ALUMNOS
SET TELEFONO=(SELECT TELEFONO FROM PERSONA WHERE DNI='222B')
WHERE DNI='111C';

Esborra les dades

- També podem eliminar dades de la BASE DE DADES, amb la clàusula DELETE:

```
DELETE <NOMBRE TABLA>
[WHERE <CONDICIÓN>

DELETE ALUMNOS
WHERE DNI='111C';
```

Consultes

Ja sabem com introduir, modificar i eliminar dades, ara tenim el més important i el motiu de crear una Base de Dades, per poder consultar-la, per a això tenim la clàusula SELECT. La seva expressió més simple és:

```
SELECT <CAMPOS o EXPRESIONES>
 FROM <TABLA O TABLAS DE LAS QUE OBTENER LOS DATOS>
[WHERE <CONDICIONES O RESTRICCIONES EN LA CONSULTA>]
```

El més senzill és obtenir totes les dades possibles d'una sola taula:

```
SELECT *
 FROM ALUMNOS;
```

L'asterisc és la carta salvatge per aconseguir-ho tot, aquesta consulta equival a:

```
SELECT DNI, NOMBRE, EDAD, TELEFONO
 FROM ALUMNOS;
```

Clàusula Where

Símbolo	Significado	Ejemplo
=	Igualdad	1 = 2
!= <> ^=	Desigualdad	1 != 2 1 <> 2 1 ^= 2
>	Mayor que	1 > 2
<	Menor que	1 < 2
>=	Mayor o igual que	1 >= 2
<=	Menor o igual que	1 <= 2
IN (RS)	Igual a algún elemento del result set.	1 IN (1,2) [TRUE]
<op.> ANY <op.> SOME	<op> a algún elemento del result set (derecha). Debe ser estar precedido por =, !=, <, <=, >, >= Hace un OR lógico entre todos los elementos.	10 >= ANY (1,2,3,10) [TRUE]
<op.> ALL	<op> a todos los elementos del result set (derecha), Debe ser estar precedido por =, !=, <, <=, >, >= Hace un AND lógico entre todos los elementos.	10 <= ALL (1,2,3,10) [TRUE]
BEETWEEN x AND y	Operando de la izquierda entre x e y. Equivalente a op >= x AND op <= y	10 BETWEEN 1 AND 100
EXISTS <subconsulta>	Si la <subconsulta> retorna al menos una fila	EXISTS(SELECT 1 FROM DUAL)
LIKE(*)	Es como	'pepe' LIKE 'pe%'
IS NULL	Si es nulo	1 IS NULL
IS NOT NULL	Si es No nulo	1 IS NOT NULL
NOT cond.	Niega la condición posterios	NOT EXISTS... NOT BETWEEN NOT IN NOT =
cond AND cond	Hace un AND lógico entre dos condiciones	1=1 AND 2 IS NULL
Cond OR cond	Hace un OR lógico entre dos condiciones	1=1 OR 2 IS NULL

Consultes

```
SELECT NOMBRE
 FROM ALUMNOS
WHERE EDAD>20;
```

Hi haurà alumnes de més de 20 anys.

Expressions aritmètiques

```
SELECT NOMBRE SUELDO*(1+COMISION/100)
FROM PERSONAL
```

```
SELECT NOMBRE, 2*2 OPER
 FROM ALUMNOS
```

Consultes

Files duplicades

Si volem aconseguir un camp o conjunt d'ells i volem evitar duplicats,
Utilitzarem la clàusula DISTINCT

```
SELECT DISTINCT NOMBRE
 FROM ALUMNOS;
```

Si hi ha dos o més noms iguals, només en sortirà un.

```
SELECT DISTINCT NOMBRE, EDAD
 FROM ALUMNOS;
```

En aquest cas augmentem el camp AGE a la condició, és a dir, el nom sortirà
Repetits si tenen una edat diferent, però si tenen la mateixa edat només sortirà 1.

Consultes

Limitar files mitjançant una selecció. Clàusula WHERE

El operador IN / NOT IN

```
SELECT NOMBRE
FROM ALUMNOS
WHERE EDAD IN (5,10,15,20)
```

Els estudiants que tenen 5 o 10 o 15 o 20 anys, serien equivalents a posar AGE = 5 O EDAT = 10 ...

```
SELECT NOMBRE
FROM ALUMNOS
WHERE EDAD NOT IN (5,10,15,20)
```

Els estudiants que no tinguin 5 o 10 o 15 o 20 anys equivaldrien a posar AGE<>5
AND EDAD<>10...

BETWEEN

Serà una consulta entre 2 dades

```
SELECT NOMBRE
FROM ALUMNOS
WHERE EDAD BETWEEN 15 AND 20;
```

Consultes

Condició LIKE

S'utilitza per a cadenes on en coneixem part, però no del tot, si ho sabéssim hauríem d'utilitzar l'operador =.

Tenim 2 comodins:
% (porcentaje) Indica cap o molts caràcters.
_ (guión bajo) Indica un sol caràcter.

```
SELECT NOMBRE
FROM ALUMNOS
WHERE NOMBER LIKE '%A'
```

Els alumnes acabats en A

```
SELECT NOMBRE
FROM ALUMNOS
WHERE NOMBER LIKE '%_A_%'
```

Estudiants amb lletra A al mig del nom, ni al principi ni al final.

Consultes

Condició EXISTS (Se usa en consultas anidadas.)

```
SELECT *
FROM ALUMNOS A
WHERE EXISTS (SELECT 1 FROM MATRICULAR M WHERE M.DNI=A.DNI)
```

Tenim els alumnes matriculats.

Condicions Lógiques
Es poden utilitzar diverses condicions en un ON, per a elles utilitzarem AND o /i O. Nosaltres tampoc no ho hem fet.

```
SELECT NOMBRE
FROM ALUMNOS
WHERE EDAD=20
AND TELEFONO>111
AND NOMBRE LIKE '%A'
```

Has de complir els 3 requisits perquè puguis presentar-te a la consulta

```
SELECT NOMBRE
FROM ALUMNOS
WHERE EDAD=20
OR TELEFONO>111
OR NOMBRE LIKE '%A'
```

N'hi ha prou que compleixi un dels requisits perquè marxi.

Consultes

Exemples de NOT:
... WHERE COD_TIPE NOT IN ('RES','VEJ');
... WHERE SUELDO_BRUTO_PERS NOT BETWEEN 1000 AND 2000
... WHERE NOMBRE_PERS NOT LIKE 'P%'
... WHERE ID_CLIE IS NOT NULL

Comparaciones con NULL
Con NULL no podemos usar los conectores habituales, NULL es INCOMPARABLE, debemos usar IS o IS NOT

```
SELECT *
FROM ALUMNOS
WHERE TELEFONO = NULL
```
Aquesta consulta no tornarà mai més. Ho hem de fer:

```
SELECT *
FROM ALUMNOS
WHERE TELEFONO IS NULL
```

Clàusula ORDER BY

Si volem que s'ordenin les consultes hem d'utilitzar aquesta clàusula, per a elles després de l'ON o DES la posarem i ordenarem pel camp de la taula que vulguem.

```
SELECT *
FROM ALUMNOS
WHERE EDAD>10
ORDER BY NOMBRE;
```

Això ordenarà alfabèticament els alumnes, si volem que ho faci contràriament a l'ordre alfabètic:

```
SELECT *
FROM ALUMNOS
WHERE EDAD>10
ORDER BY NOMBRE DESC;
```

Podem ordenar per més d'un camp:

```
SELECT *
FROM ALUMNOS
WHERE EDAD>10
ORDER BY NOMBRE, EDAD;
```

En aquest cas s'ordenarà pel seu nom i en els casos en què hi hagi noms repetits els ordenarà per AGE.

Consultes amb diverses taules

És possible que vulguem consultar diverses taules alhora, per a això en el FROM nidifiquem separant per comes totes les taules desitjades. Normalment per a encreuaments entre taules utilitzarem ALIASes, s'utilitzen per a la claredat de la consulta i per als casos en què hi ha taules amb camps que tenen el mateix nom.

```
SELECT ALU.NOMBRE, ASIG.NOMBRE, FECHA

FROM ALUMNOS ALU,  MATRICULAR MAT,  ASIGNATURAS ASIG

WHERE ALU.DNI=MAT.DNI  AND  MAT.COD=ASIG.COD;
```