

UT6. SQL Manipulación de Datos**INSERCIÓN, BORRADO Y ACTUALIZACIÓN DE DATOS**

- **Inserción: INSERT**
- **Actualización o modificación: UPDATE**
- **Borrado o eliminación: DELETE**
- **COMMIT y ROLLBACK**

INSERCIÓN (INSERT)

Podemos añadir (insertar) filas en una tabla mediante la instrucción **INSERT** de dos modos:

- **Con la cláusula VALUES**

```
INSERT INTO nombretabla VALUES (valorcol1, valorcol2,...,valorcoln)
```

```
INSERT INTO DEPART VALUES (50,'PUBLICIDAD','MADRID');
```

Para cada columna de la tabla habrá de especificarse su valor.

Las columnas han de ir ordenadas tal y como están definidas en la tabla.

Los valores de cada columna habrán de ser del tipo y tamaño de la columna en cuestión.

Los valores no numéricos (fechas y textos) tendrán que ir entre comillas simples.

Si no se va a dar valor a todas las columnas hay que especificar que columnas van a recibir un valor de la siguiente manera:

```
INSERT INTO nombretabla (columna1, columna2) VALUES (valorcol1, valorcol2);
```

```
INSERT INTO DEPART (Dept_no, Dnombre) VALUES (60,'GASTOS');
```

Las columnas no especificadas se quedaran con valor NULL.

Las columnas que no pueden contener valores nulos (se especifica al crear la tabla) han de recibir siempre un valor

```
INSERT INTO DEPART (Dnombre, Loc) VALUES ('JUERGAS VARIAS','Marbella');
```

ERROR en línea 1:

ORA-01400: no se puede realizar una inserción NULL en
("USUDAW"."DEPART"."DEPT_NO")

```
INSERT INTO DEPART (Dept_no,Dnombre, Loc) VALUES (500,'JUERGAS VARIAS','Marbella')
```

ERROR en línea 1:

ORA-01438: valor mayor que el que permite la precisión especificada para esta columna

- **Mediante una SELECT**

Se insertarán en la tabla las filas resultantes de una instrucción select

```
INSERT INTO nombretabla1 [(columna1, columna2,...)]
```

```
SELECT col1, col2...
```

```
FROM nombretabla2 [Cláusulas select];
```

Ejemplos:

```
INSERT INTO DEPART
```

```
SELECT MAX(DEPT_NO)+10, 'NUEVO', 'TOLEDO'
```

```
FROM DEPART;
```

Ejemplos:

Insertar en ASIGNATURAS una nueva asignatura con código 5 y nombre FOL

```
INSERT INTO ASIGNATURAS VALUES (5,'FOL');
```

Insertar en NOTAS para cada alumno la nota 8 en la asignatura 5

```
INSERT INTO NOTAS
```

```
SELECT DNI,5,8 FROM ALUMNOS;
```

Insertar en ALUMNOS a la hermana de Alcalde García, Elena con dni 898988G, que vive en la misma dirección y con el mismo teléfono y su nombre es Almudena.

```
INSERT INTO ALUMNOS
```

```
SELECT '898988G', 'Alcalde García, Almudena', DIREC, POBLA, TELEF
```

```
FROM ALUMNOS WHERE APENOM = 'Alcalde García, Elena';
```

Insertar en NOTAS las mismas notas en Programación y Marcas que su hermana Elena

```
INSERT INTO NOTAS  
  
SELECT '898988G', N.COD, NOTA  
  
FROM NOTAS N, ALUMNOS A, ASIGNATURAS ASI  
  
WHERE N.DNI = A.DNI  
  
AND ASI.COD = N.COD  
  
AND APENOM = 'Alcalde García, Elena'  
  
AND ASI.NOMBRE IN ('Programación','Marcas');
```

Insertar en NOTAS las mismas notas en el resto de asignaturas la nota más 1 de su hermana.

```
INSERT INTO NOTAS  
  
SELECT '8989888G', N.COD, NOTA + 1  
  
FROM NOTAS N, ALUMNOS A, ASIGNATURAS ASI  
  
WHERE N.DNI = A.DNI  
  
AND ASI.COD = N.COD  
  
AND APENOM = 'Alcalde García, Elena'  
  
AND ASI.NOMBRE NOT IN ('Programación','Marcas');
```

Insertar en EMPLE uno que se apellide 'QUIROGA' con número 1112, la fecha de alta mañana y el resto de datos los de 'GIL'

```
INSERT INTO EMPLE  
  
SELECT 1112, 'QUIROGA', OFICIO,  
  
RESPONSABLE, SYSDATE+1,SALARIO, COMISION,DEPT_NO  
  
FROM EMPLE  
  
WHERE APELLIDO = 'GIL';
```

Insertar en EMPLE uno que se apellide 'GARCÍA' con número 1111 en el departamento con mayor número de empleados. La fecha de alta será la del sistema, la comisión y director nulos, oficio 'GERENTE' y salario 1200.

INSERT INTO EMPLE

SELECT 1111, 'GARCÍA', 'GERENTE', NULL, SYSDATE, 1200, NULL, DEPT_NO

FROM EMPLE

GROUP BY DEPT_NO

HAVING COUNT(*) = (SELECT MAX(COUNT(*))

FROM EMPLE

GROUP BY DEPT_NO);

ACTUALIZACIÓN (UPDATE)

Podemos modificar los datos de una o varias filas de una tabla con la instrucción **UPDATE** .

UPDATE nombretabla

SET columna1 = valor1, columna2 = valor2,..., columna3 = valor3

WHERE condicion;

SET especifica los valores nuevos de las columnas que se desean modificar

WHERE especifica las filas que se van a modificar

El nuevo valor de una columna ha de ser del mismo tipo que la columna

Solo es posible cambiar los datos de una tabla cada vez

UPDATE EMPLE

SET SALARIO = SALARIO + 100, COMISION = 20

WHERE APELLIDO = 'SALA';

- UPDATE CON SELECT

Podemos dar a una columna el valor que nos devuelva una select

UPDATE EMPLE

```
SET SALARIO = (SELECT SALARIO  
FROM EMPLE WHERE APELLIDO = 'SALA')  
WHERE APELLIDO = 'ARROYO';
```

Podemos utilizar en el WHERE una subconsulta

UPDATE EMPLE

```
SET SALARIO = SALARIO + 100  
WHERE DEPT_NO IN (SELECT DEPT_NO FROM DEPART  
WHERE DNOMBRE = 'CONTABILIDAD');
```

Subir 100 euros a los empleados que ganan menos que la media de su departamento.

UPDATE EMPLE E

```
SET SALARIO = SALARIO + 100  
WHERE SALARIO < (SELECT AVG(SALARIO)  
FROM EMPLE  
WHERE DEPT_NO = E.DEPT_NO);
```

Modificar en ASIGNATURAS el nombre de la asignatura de código 5 poniéndole F.O.L.

UPDATE ASIGNATURAS

```
SET NOMBRE = 'F.O.L.'  
WHERE COD = 5;
```

Modificar en NOTAS la nota de BBDD subiéndoles un punto a los alumnos con nota mayor que 6

UPDATE NOTAS

SET NOTA = NOTA +1

WHERE NOTA > 6

AND COD IN (SELECT COD FROM ASIGNATURAS

WHERE NOMBRE = 'BBDD');

BORRADO (DELETE)

Podemos borrar una o varias filas de una tabla con la instrucción **DELETE** .

- **DELETE nombretabla**
WHERE condicion;

Si no ponemos ninguna condición se borrarán todas las filas de la tabla y si ponemos una condición se borrarán las filas de la tabla que cumplan esa condición.

Borrar los empleados de Emple cuyo oficio es VENDEDOR

DELETE EMPL

WHERE OFICIO = 'VENDEDOR';

Borrar los empleados de Emple que trabajan en el departamento de CONTABILIDAD

DELETE EMPL

WHERE DEPT_NO = (SELECT DEPT_NO FROM DEPART

WHERE DNOMBRE= 'CONTABILIDAD');

Borrar los empleados que ganan más de la media del salario de la empresa sin borrar al PRESIDENTE ni tenerlo en cuenta para la media

DELETE EMPL

```
WHERE SALARIO > (SELECT AVG(SALARIO)
FROM EMPL
WHERE OFICIO <> 'PRESIDENTE')
AND OFICIO <> 'PRESIDENTE';
```

Borrar a los empleados que ganan el salario mínimo de su departamento

DELETE EMPL E

```
WHERE SALARIO = (SELECT MIN(SALARIO)
FROM EMPL
WHERE E.DEPT_NO = DEPT_NO);
```

Borrar los departamentos que no tienen empleados

DELETE DEPART

```
WHERE DEPT_NO NOT IN (SELECT DEPT_NO FROM EMPL);
```

COMMIT y ROLLBACK

Una transacción es una secuencia de una o más sentencias SQL

Cuando hacemos transacciones en la base de datos, es decir, cuando insertamos, actualizamos y eliminamos datos en las tablas, los cambios no se aplicarán a la base de datos hasta que no realicemos la orden **COMMIT**

Si antes de hacer COMMIT hacemos **ROLLBACK**, la transacción queda anulada volviendo a la situación de las tablas de la base de datos desde el último COMMIT

Determinadas órdenes como CREATE, llevan a cabo un COMMIT implícito.