**FUNCIONES BÁSICAS**

**INSERT:** Añade nuevas filas a una tabla ya creada.

**UPDATE:** Modifica los datos insertados en las filas de una tabla.

**DELETE:** Elimina filas de una tabla.

**SELECT:** Consulta la información almacenada en las tablas.

**INSERT INTO:**

Sintaxis: INSERT INTO nombre\_tabla[(columna1, [columna2,….])] VALUES (valor1 [, valor 2…..]);

Ejemplo 1, especificando columnas: INSERT INTO dept( deptno, dname, loc) VALUES (10, ‘Ventas’,’Córdoba’);

Ejemplo 2, sin especificar columnas: INSERT INTO dept VALUES (10, ‘Ventas’,’Córdoba’);

Ejemplo 3, valores nulos especificando columnas: INSERT INTO dept (deptno,dname) VALUES (10, ‘Ventas’);

Ejemplo 4, valores nulos sin especificar columnas: INSERT INTO dept VALUES (10, ‘Ventas’, NULL);

Ejemplo 5, varias filas a la vez: INSERT ALL

INTO mytable (column1, column2, column3)

VALUES ('val1.1', 'val1.2', 'val1.3')

INTO mytable (column1, column2, column3)

VALUES ('val2.1', 'val2.2', 'val2.3')

INTO mytable (column1, column2, column3)

VALUES ('val3.1', 'val3.2', 'val3.3')

SELECT \* FROM dual;

Ejemplo 6, copiar filas entre tablas: INSERT

INTO administrativos(id, nombre, salario)

SELECT id, nombre, salario

FROM empleados

WHERE trabajo=‘ADMINISTRATIVO’;

**UPDATE:**

Sintaxis: UPDATE table SET column=value [, column=value] [WHERE condition]

Ejemplo 1, actualizar un campo: UPDATE Alumnos SET edad=22 WHERE dni=‘76765434T’;

Ejemplo 2, actualizar filas desde otra tabla: UPDATE alumnos

SET edad = (SELECT MAX(edad)

FROM alumnos)

WHERE peso = (SELECT peso

FROM alumnos

WHERE dni LIKE ‘76878767G’));

**DELETE:**

Sintaxis: DELETE [FROM] table [WHERE condition];

Ejemplo 1, eliminar una fila: DELETE FROM departamento WHERE

nombre=’DEVELOPMENT’;

DELETE FROM departamento;

Ejemplo 2, eliminar una fila basada en otra tabla: DELETE FROM empleados

WHERE departamento = (SELECT num FROM departamentos

WHERE nombre LIKE ‘VENTAS’);

**FUNCIONES AVANZADAS**

**CONCAT:** Concatena strings, pueden ser arbitrarias o seleccionadas desde consultas.

**GROUP BY (+HAVING):** Agrupa una columna por el dato indicado. Si se hace la AVG de un GROUP BY, se hará la AVG del grupo en sí, no de la columna. Se puede agrupar por varios campos.

**ORDER BY:** Ordenar una selección por un campo concreto.

**VIEW:** Crear y modificar vistas.

**CONCAT:**

Sintaxis: SELECT CONCAT(nombre, “ ”, apellido)

AS nombre\_completo;

**GROUP BY (+HAVING):**

Sintaxis: SELECT column FROM table

GROUP BY [campo1, campo2…]

HAVING campo2

LIKE “string”;

**ORDER BY:**

Sintaxis: SELECT column1, column2 FORM table

ORDER BY column2;

**VIEW:**

Sintaxis: CREATE VIEW viewName AS “View Name”

SELECT column

FROM table

WHERE campo1 = 44

WITH CHECK OPTION;