
PRÁCTICA: 3. LENGUAJE DE MANIPULACIÓN DE DATOS Y CONSULTAS – I

<u>1. SOBRE LA PRÁCTICA.....</u>	<u>2</u>
1.1. OBJETIVOS	2
1.2. MATERIAL	2
1.3. CONTENIDOS	2
1.4. BIBLIOGRAFÍA:	2
<u>2. LENGUAJE DE MANIPULACIÓN DE DATOS.....</u>	<u>3</u>
2.1. INSERCIÓN DE FILAS EN UNA TABLA	3
2.1.1. INSERTAR UNA FILA EN UNA TABLA	3
2.1.2. INSERTAR EN UNA TABLA EL RESULTADO DE UNA CONSULTA (INSERCIÓN MULTIFILA)	3
2.2. ELIMINACIÓN DE FILAS DE UNA TABLA.....	3
2.3. MODIFICACIÓN DE LOS DATOS DE UNA TABLA	4
<u>3. LENGUAJE DE CONSULTAS - I.....</u>	<u>4</u>
3.1. SINTAXIS.....	4
3.1.1. CLÁUSULA SELECT.	4
3.1.2. CLÁUSULA FROM.....	5
3.1.3. CLÁUSULA WHERE.....	5
3.2. DICCIONARIO DE DATOS	6
<u>4. EJERCICIOS.....</u>	<u>7</u>
4.1. EJERCICIOS SOBRE MANIPULACIÓN DE DATOS	7
4.2. EJERCICIOS SOBRE CONSULTAS.....	8

1. Sobre la práctica

1.1. Objetivos

Conocer las sentencias disponibles en SQL para manipulación de datos, es decir, insertar, modificar o eliminar filas de una tabla.

Conocer la sintaxis básica para la realización de consultas sencilla en SQL.

1.2. Material

ORACLE9 versión para WINDOWS XP

1.3. Contenidos

- Inserciones de Datos
- Borrados de Datos
- Modificaciones de Datos
- Consultas I
 - Select
 - From
 - Where
 - Diccionario de Datos
 - Ejercicios

1.4. Bibliografía:

GUÍA DE SQL
JAMES R. GROFF, PAUL N. WEINBERG
MCGRAW-HILL, 1998

ORACLE 9I: GUIA DE APRENDIZAJE
ABBEY, MICHAEL Y COREY, MIKE Y ABRAMSON, IAN
MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, S.A.

2. Lenguaje de Manipulación de Datos

2.1. Inserción de filas en una tabla

Consiste en añadir una o varias filas a una tabla. La sintaxis SQL es la siguiente:

```
INSERT INTO <nombre tabla> [(<lista-de columnas>)]  
    { VALUES (<lista-de valores>)  
    | <sentencia SELECT> }
```

Se puede emplear para dos objetivos distintos:

2.1.1. Insertar una fila en una tabla

```
INSERT INTO <nombre tabla> [(<lista-de columnas>)]  
    VALUES (<lista-de valores>)
```

2.1.2. Insertar en una tabla el resultado de una consulta (inserción multifila)

```
INSERT INTO <nombre tabla> [(<lista-de columnas>)]  
    <sentencia SELECT>
```

- Los valores de tipo DATE o CHAR se deben encerrar entre comillas simples.
- Si no se indica la lista de columnas, se consideran todas las de la tabla.
- En la opción b) la tabla donde se inserta se puede utilizar en la consulta SELECT.
- En la opción b) la consulta no puede incluir una cláusula ORDER BY.
- Los valores insertados deben ser de un tipo compatible con el de las columnas de la tabla.

2.2. Eliminación de filas de una tabla

Permite eliminar filas de una tabla. La sentencia es:

```
DELETE FROM <nombre tabla> [WHERE <condición>]
```

Si no se pone condición, se borran todas las filas de la tabla.

2.3. Modificación de los datos de una tabla

Se emplea para modificar los datos de las filas existentes en una tabla:

UPDATE <nombre tabla> SET

```
{      <columna> = <expresión> [, ...]
|      {( <lista-de columnas> | * ) = ( <lista-de expresiones> )      }
[WHERE <condición>]
```

- Una expresión puede estar formada por una subconsulta SELECT entre paréntesis cuyo resultado es una única fila de una sola columna (un único valor simple).
- La lista de columnas está formada por aquellas columnas a las cuales se les modificará su valor. Si se omite la cláusula WHERE entonces se actualizan todas las filas de la tabla.

3. Lenguaje de Consultas - I

3.1. Sintaxis.

La sintaxis de la orden SELECT consta básicamente de las cláusulas SELECT y FROM como obligatorias y de otras varias cláusulas opcionales:

```
<cláusula SELECT> <cláusula FROM>
  [ <cláusula WHERE> ]
  [ <cláusula GROUP BY> [ <cláusula HAVING> ] ]
  [ <cláusula ORDER BY>]
```

Las cláusulas ORDER BY, GROUP BY y HAVING se verán en otra práctica. El formato detallado es:

```
SELECT [ALL | DISTINCT | UNIQUE ] <lista-de selecciones>
  FROM <nombre de tabla> [alias de tabla] [... ]
  [WHERE <condición>]
  [GROUP BY <lista-de columnas> [HAVING <condición>] ]
  [ORDER BY <nombre de columna> [ASC | DESC] [... ] ]
```

3.1.1. Cláusula SELECT.

Permite indicar los datos (columnas o expresiones) que queremos obtener.

```
SELECT [ALL | DISTINCT | UNIQUE ] <lista-de selecciones>
```

DISTINCT: Elimina las filas duplicadas en el resultado de la consulta.

UNIQUE es igual que **DISTINCT**.

<lista-de selecciones>: Lista de nombre de columnas o expresiones separadas por comas.

NOTAS:

- Las columnas ambiguas se preceden del nombre de la tabla: <tabla>.<columna>
- * refiere a todas las columnas de todas las tablas.

3.1.2. Cláusula FROM.

Permite indicar las tablas que contienen los datos.

FROM { <nombre de tabla> [alias de tabla] } [...]

<alias de tabla>: Es un sinónimo activo sólo en el ámbito de la sentencia SELECT.

3.1.3. Cláusula WHERE.

Sirve para indicar la condición que deben cumplir las filas resultantes.

WHERE <condición>

Una condición está formada por una o varias expresiones condicionales conectadas por los operadores lógicos AND, OR y NOT. Una expresión condicional tiene una de las formas siguientes:

<expresión1> <operador relacional> <expresión2>

Verifica si las dos expresiones satisfacen la comparación.

<expresión1> [NOT] BETWEEN <expresión2> AND <expresión3>

Verifica si la expresión1 tiene un valor comprendido entre los valores de la expresión2 y la expresión3.

<expresión> [NOT] IN (<lista-de valores>)

Verifica si la expresión tiene un valor de los indicados en la lista de valores.

<nombre de columna> [NOT] LIKE "<string>" [ESCAPE "<carácter de escape>"]

Verifica si el valor de la columna se adapta al patrón de búsqueda (string). Se admiten caracteres comodín ("%" representa cero o más caracteres, "_" representa un único carácter). El carácter de escape '/' permite referirse a los caracteres comodín como caracteres y no como comodines.

<nombre de columna> IS [NOT] NULL

Verifica si el valor de la columna es nulo.

3.2. Diccionario de Datos

Conjunto de tablas que contiene las definiciones de los elementos de las bases de datos creadas por el usuario. Estas tablas son creadas y actualizadas directamente por el SGBD. Las tablas del diccionario de datos se pueden consultar con SELECTs

SELECT TABLE_NAME FROM USER_TABLES

Obtenemos el nombre de las tablas existente en la base de datos.

DESCRIBE nombre_tabla

Ofrece la descripción completa de una tabla existente en la base de datos.

4. Ejercicios.

Considerando el siguiente esquema relacional:

DEPARTAMENTOS (codigo, nombre)

AREAS (codigo, nombre, departamento)

PROFESORES (codigo, apellido1, apellido2, nombre_pila, activo, categoria, dedicacion, area)

ASIGNATURAS (siglas, nombre, creditos, curso, anualidad, clase, horas_teoría, horas_practica, grupos_teoría , grupos_practica, alumnos)

LOCALES (codigo, nombre, docente, capacidad, edificio, situacion)

GRUPOS (curso, clase, codigo, nombre)

DOCENCIA (id, curso, clase, grupo, siglas, profesor, local, día, hora, periodicidad)

Y considerando las siguientes claves ajenas:

areas.departamento → departamentos

profesores.area → areas

docencia.curso, clase, grupo → grupos

docencia.profesor → profesores

docencia.local → locales

docencia.siglas → asignaturas

Realizar los siguientes ejercicios:

4.1. Ejercicios Sobre Manipulación de Datos

1. Se incorpora una nueva área de conocimiento al departamento de Informática:
código: ISA
nombre: Ingeniería de Sistemas y Automática
2. Eliminar de la BD los profesores que no están en activo y que no son asociados.
3. Aumentar la capacidad de las aulas de clase y laboratorios en un 10%.
4. El área de conocimiento EIO (Estadística e Investigación Operativa) pasa del Departamento de Informática al Departamento de Matemáticas, y por tanto, todos sus profesores también cambian de departamento.
5. Crear la tabla BAJAS con el código, nombre y categoría (atributos iguales a los de la tabla de profesores). Añadirle los datos de los profesores (con una subconsulta) que no están en activo en el centro y quitarlos de la tabla de Profesores.

4.2. Ejercicios Sobre Consultas

1. Listar todos los datos disponibles de todos los departamentos.
2. Listar el código y nombre de las áreas de conocimiento.
3. Listar los tipos de categorías que tienen los profesores.
4. Listar, código y nombre (nombre de pila y apellidos) de los profesores en activo (1=sí, 0=no).
5. Listar código y nombre de los profesores con código mayor de 20 y menor de 40.
6. Listar los profesores que son numerarios (categorías 'TU', 'CEU' o 'TEU').
7. Listar los profesores que contienen 'Ruiz' en su primer o segundo apellidos.