

Introducción al lenguaje de consulta estructurado (SQL)

Structured Query Language

Agenda

- Sentencias avanzadas en SQL.
 - Sentencias INSERT
 - Sentencias UPDATE
 - Sentencias DELETE

INSERT

- Esta sentencia nos permite insertar registros en una tabla de la base de datos.
- Se pueden insertar uno o varios registros a la vez.
- Según la definición de la tabla al momento de su creación, es posible que los registros a insertar generen errores y el motor de base de datos nos impida realizar dicha operación.
- Los errores más comunes son los siguientes
 - Equivocar el tipo de dato.
 - Omitir el valor de un campo requerido.
 - Violar una restricción referencial.

INSERT (cont.)

- Los valores especificados para los campos deben coincidir con los tipos de datos definidos al momento de crear la tabla donde se está insertando el registro.
- Todos los campos requeridos y que no tengan un valor por defecto, deben estar incluidos en la sentencia INSERT.
- Si existe una integridad referencial, el valor otorgado a ese campo, debe ser válido para la entidad que está referenciando.

INSERT (cont.)

- Sintaxis para insertar un registro.

```
INSERT INTO tabla (campo1, campo2, ... , campoN)  
VALUES ('valor1', 'valor2', ... , 'valorN')
```

- INSERT permite especificar los valores a insertar por medio de una sub-consulta

```
INSERT INTO tabla1 (campo1, campo2, ... , campoN)  
SELECT (campo1, campo2, ... , campoN)  
FROM tabla2
```

UPDATE

- Esta sentencia nos permite actualizar registros en una tabla de la base de datos.
- Se pueden modificar uno o varios registros a la vez.
- La sentencia UPDATE se puede utilizar con la cláusula WHERE. Si no se especifica la cláusula WHERE, se actualizan todos los registros de la tabla.
- Se aplican las mismas restricciones que existen para las sentencias INSERT siendo los errores más comunes.
 - Equivocar el tipo de dato.
 - Borrar el valor de un campo requerido.
 - Violar una restricción referencial.

UPDATE (cont.)

- Sintaxis

```
UPDATE tabla
```

```
SET campo1 = expresión1,
```

```
    campo2 = expresión2, ... ,
```

```
    campoN = expresiónN
```

```
WHERE condición1, condición2, ... , condiciónM
```

- Ejemplos:

```
UPDATE Sucursal SET objetivo = 1200000 WHERE id = 4;
```

```
UPDATE Sucursal SET objetivo = 2400000 WHERE nombre =  
'Olivos';
```

```
UPDATE Sucursal SET objetivo = 900000;
```

DELETE

- Esta sentencia nos permite borrar registros en una tabla de la base de datos.
- Se pueden borrar uno o varios registros a la vez.
- La sentencia DELETE se puede utilizar con la cláusula WHERE. Si no se especifica la cláusula WHERE, se borran TODOS los registros de la tabla.
- Si se intenta borrar un registro que está referenciado integralmente, se produce un error en la base de datos.

DELETE (cont.)

- Sintaxis

DELETE

FROM *tabla*

WHERE *condición1, condición2, ... , condiciónN*

- Ejemplo

DELETE FROM Cliente

CUIDADO, BORRA TODOS LOS CLIENTES DE LA
TABLA !!!

Ejercicios

- E46: Insertar un nuevo Cliente llamado “Juan Perez” que nació el 22 de enero de 1977.
- E47: Insertar un empleado cuyo id de sucursal no exista.
- E48: Insertar un cliente sin nombre.
- E49: Insertar un nuevo empleado llamado “Juan Perez” y que trabaje en la sucursal Lanus Oeste, siendo su jefe el actual jefe de dicha sucursal.
- E50: Traspasar a todos los empleados de la sucursal Olivos a la sucursal San Miguel.

Ejercicios (cont.)

- E51: Cambiar la categoría ALTA por SUPER.
- E52: Sumar \$100 a todas las facturas que incluían el producto GPS.
- E53: Borrar los clientes de la sucursal Olivos.
- E54: Borrar una sucursal que tenga un empleado, por ejemplo, la sucursal “Olivos”.

FIN