UNIVERSIDAD TECNOLOGICA
IZUCAR DE MATAMOROS
Organismo Público Descentralizado del Estado de Puebla

COMANDOS DDL Y DML

Alumna: Lizeth Ángela Peláez González

Docente: Lic. Carlos González González

Materia: Base de datos

INDICE

Comandos del DDL y del DML

Comandos del DDL

CREATE

DROP

ALTER

TRUNCATE

Commandos DML

SELECT

INSERT

DELETE

UPDATE

Comandos del DDL y del DML

Comandos DLL: El lenguaje de definición de datos

Commando	Description
CREATE	Utilizado para crear nuevas tablas, stored procedures e índices
DROP	Empleado para eliminar tablas, stored procedures e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos
TRUNCATE	Borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

Comandos DML: Un lenguaje de manipulación de datos (*Data Manipulation Language*, o *DML* en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

Commando	Description
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
DELETE	Utilizado para modificar los valores de los campos y registros especificados.
UPDATE	Utilizado para eliminar registros de una tabla de una base de datos.

INDICE

Comandos del DDL

CREATE

Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte.

Ejemplo (crear una tabla)

```
CREATE TABLE 'TABLA_NOMBRE'
  'CAMPO_1' INT,
  'CAMPO_2' STRING
```

DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

Ejemplo

```
ALTER TABLE "TABLA_NOMBRE"
(
   DROP COLUMN "CAMPO_NOMBRE1"
)
```

ALTER

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla)

```
ALTER TABLE 'TABLA_NOMBRE' (
   ADD NUEVO_CAMPO INT UNSIGNED meel
)
```

TRUNCATE

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

Ejemplo

```
TRUNCATE TABLE "TABLA_NOMBRE1"
```

INDICE

Commandos DML

SELECT

La sentencia SELECT nos permite consultar los datos almacenados en una tabla de la base de datos.

Ejemplo:

SELECT "nombre_columna" FROM "nombre_tabla"

Para ilustrar el ejemplo anterior, suponga que tenemos la siguiente tabla:

Tabla ***Store_Information***

store_name	Sales	Date
Los Angeles	1500 €	05-Jan-1999
San Diego	250 €	07-Jan-1999
Los Angeles	300 €	08-Jan-1999
Boston	700 €	08-Jan-1999

INSERT

Una sentencia *INSERT* de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Ejemplo:

Forma básica

INSERT INTO "tabla" ("columna1", ["columna2,... "]) VALUES ("valor1", ["valor2,..."])
Las cantidades de columnas y valores deben ser iguales. Si una columna no se especifica, le será asignado el valor por omisión. Los valores especificados (o implícitos) por la sentencia INSERT deberán satisfacer todas las restricciones aplicables. Si ocurre un error de sintaxis o si alguna de las restricciones es violada, no se agrega la fila y se devuelve un error.
Ejemplo

INSERT INTO agenda_telefonica (nombre, numero) VALUES ('Roberto Jeldrez', 4886850);

Cuando se especifican todos los valores de una tabla, se puede utilizar la sentencia acortada:

INSERT INTO "tabla" VALUES ("valor1", ["valor2,..."])

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas de la tabla 'agenda_telefonica'):

INSERT INTO agenda_telefonica VALUES ('Roberto Jeldrez', 4886850);

INDICE

DELETE

Una sentencia *DELETE* de SQL borra uno o más registros existentes en una tabla,

Forma básica

```
DELETE FROM ''tabla'' WHERE ''columna1'' = ''valor1''
```

Ejemplo

```
DELETE FROM My_ table WHERE field2 = 'N';
```

OTRO EJEMPLO DE DELETE

Delete from

store_name	Sales	Date
Los Angeles	1500 €	05-Jan-1999
San Diego	250 €	07-Jan-1999
Los Angeles	300 €	08-Jan-1999
Boston	700 €	08-Jan-1999

y decidimos no mantener ninguna información sobre Los Ángeles en esta tabla. Para lograrlo, ingresamos el siguiente SQL:

DELETE FROM Store _ Information
WHERE store_ name = "Los Angeles"

Ahora el contenido de la tabla se vería,

Tabla ***Store_Information***

store_name	Sales	Date
San Diego	250 €	07-Jan-1999
Boston	700 €	08-Jan-1999

INDICE

UPDATE

Una sentencia *UPDATE* de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Forma básica

```
 UPDATE ''tabla''
SET ''columna1'' = ''valor1'' ,''columna2'' = ''valor2'',...
 WHERE ''columnaN = ''valorN''
```

Ejemplo:

```
UPDATE My_ table SET field1 = 'updated value' WHERE field2 = 'N';
```

INDICE