

Programación para Dispositivos Móviles(T.E.)

Ciclo 1 - 2019

Clase:09

Coordinador:Ing. César González



OBJECTIVE-C

Unidad III

Bases de Datos

Agenda

- **SQL.**
 - Comandos de DDL.
 - Comandos de DML.

SQL

Structured Query Language

SQL

Los comandos del SQL pueden dividirse en tres grupos:

- Comandos de definición de datos (DDL = *Data Definition Language*), que permiten crear y definir nuevas bases de datos, campos, ...
- Comandos de manipulación de datos (DML = *Data Manipulation Language*), que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
- Comandos de control y seguridad de datos, que gobiernan los privilegios de los usuarios, los controles de acceso, ...

Comandos de DDL

Comando	Descripción
CREATE	Encargado de crear nuevas tablas, campos, ...
DROP	Encargado de eliminar tablas
ALTER	Encargado de modificar las tablas, agregando campos o cambiando la definición de los campos

Comandos de DDL

Ejemplos(create)

```
CREATE TABLE alumno (  
    carnet VARCHAR(7) NOT NULL PRIMARY KEY,  
    nombre VARCHAR(30) NOT NULL ,  
    apellido VARCHAR(30) NOT NULL ,  
    sexo VARCHAR(1) NOT NULL ,  
    matganadas INTEGER) NOT NULL ;
```

```
CREATE TABLE materia (  
    codmateria VARCHAR(6) NOT NULL PRIMARY KEY,  
    nommateria VARCHAR(30),  
    unidadesval VARCHAR(1));
```

Comandos de DDL

Ejemplos(create)

```
CREATE TABLE nota (  
    carnet VARCHAR(7) NOT NULL ,  
    codmateria VARCHAR(6) NOT NULL ,  
    ciclo VARCHAR(5) ,  
    notafinal FLOAT ,  
    PRIMARY KEY(carnet,codmateria,ciclo)  
    CONSTRAINT fk_nota_materia FOREIGN KEY (codmateria)  
REFERENCES materia(codmateria) ON DELETE RESTRICT,  
    CONSTRAINT fk_nota_alumno FOREIGN KEY (carnet)  
REFERENCES alumno(carnet) ON DELETE RESTRICT)
```


Comandos de DDL

Ejemplos(Alter)

```
ALTER TABLE alumno add column  
direccion VARCHAR(30);
```

```
Insert into alumno values  
( 'AA01002', 'juan', 'Perez', 'M', 0, 'direccion1' );
```

```
Select * from alumno  
Delete from alumno where carnet="AA01002"
```

Comandos de DDL

Ejemplos(drop)

```
BEGIN TRANSACTION;  
CREATE TEMPORARY TABLE alumno_backup(carnet VARCHAR(7)  
NOT NULL PRIMARY KEY,  
    nombre VARCHAR(30),  
    apellido VARCHAR(30),  
    sexo VARCHAR(1),  
    matganadas INTEGER);  
INSERT INTO alumno_backup SELECT carnet , nombre,apellido, sexo,  
matganadas FROM alumno;  
DROP TABLE alumno;  
CREATE TABLE alumno (  
    carnet VARCHAR(7) NOT NULL PRIMARY KEY,  
    nombre VARCHAR(30),  
    apellido VARCHAR(30),  
    sexo VARCHAR(1),  
    matganadas INTEGER);  
INSERT INTO alumno SELECT carnet , nombre, apellido, sexo,  
matganadas FROM alumno_backup;  
DROP TABLE alumno_backup;  
COMMIT;
```

Comandos de DML

Comando	Descripción
SELECT	Encargado de consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Encargado de cargar lotes de datos en la base de datos en una única operación
UPDATE	Encargado de modificar los valores de los campos y registros especificados
DELETE	Encargado de eliminar registros de una tabla

Comandos de DML

Ejemplos

INSERTAR REGISTROS

Ahora que tenemos nuestra base de datos creada con todas las tablas, empezaremos a llenarlas con datos, para ello, usaremos la siguiente sintaxis básica

INSERT INTO *nombreTabla*(*campo1*, *campo2*,...) **VALUES**
(*dato_campo1*,*dato_campo2*,...)

Donde los valores (**VALUES**) están dispuestos en el mismo orden que los campos definidos después de **INSERT INTO**.

```
INSERT INTO alumno(carnet,nombre,apellido,sexo,matganadas)  
VALUES ('PP12001','Juan','Perez','M',0);
```

Condiciones o criterios

Por medio de ciertos modificadores, llamados **cláusulas**, se consigue generar criterios con el fin de definir los datos que se desea seleccionar o manipular.

Cláusula	Descripción
FROM	Sirve para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Sirve para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Sirve para especificar un criterio adicional por el que agrupar los registros seleccionados
HAVING	Sirve para expresar la condición que debe satisfacer cada grupo anterior
ORDER BY	Sirve para ordenar los registros seleccionados de acuerdo con el orden especificado

Operadores Lógicos

Operador	Descripción
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Operadores Relacionales

Operador	Descripción
<	Menor que
>	Mayor que
≠	Distinto de
≤	Menor ó igual que
≥	Mayor ó igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
IN	Utilizado para especificar registros de una base de datos

Funciones de Agregación

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

Elementos de la Base de Datos

Tabla en las bases de datos, se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa

Una tabla es utilizada para organizar y presentar información. Las tablas se componen de filas y columnas de celdas que se pueden rellenar con textos y gráficos

Las tablas se componen de dos estructuras:

Registro: es cada una de las filas en que se divide la tabla. Cada registro contiene datos de los mismos tipos que los demás registros. Ejemplo: en una tabla de nombres y direcciones, cada fila contendrá un nombre y una dirección.

Elementos de la Base de Datos

Campo: es cada una de las columnas que forman la tabla. Contienen datos de tipo diferente a los de otros campos. En el ejemplo anterior, un campo contendrá un tipo de datos único, como una dirección, o un número de teléfono, un nombre, etc. A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como *índice* o *autoincrementable*, lo cual permite que los datos de ese campo cambien solos o sean el principal indicador a la hora de ordenar los datos contenidos.

Elementos de la Base de Datos

Cada tabla creada debe tener un nombre único en la cada Base de Datos, haciéndola accesible mediante su nombre o su seudónimo (Alias) (dependiendo del tipo de base de datos elegida) La estructura de las tablas viene dado por la forma de un archivo plano, los cuales en un inicio se componían de un modo similar.

Tablas(otra definicion):

Son los objetos principales de bases de datos que se utilizan para guardar datos.

Elementos de la Base de Datos

Claves foráneas:

Las claves foráneas son definidas bajo el estándar SQL a través de la restricción FOREIGN KEY. La sintaxis para agregar tal restricción a una tabla existente es definida en SQL:2003 según se muestra debajo. Si se omite la lista de columnas en la cláusula REFERENCES implica que la clave foránea tendrá que referenciar o relacionarse con la clave primaria de la tabla referenciada.

Disparadores(Triggers)

Definición:

Un ***trigger*** (o disparador) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos

Disparadores(Triggers)

Usos:

Son usados para mejorar la administración de la Base de datos, sin necesidad de contar con que el usuario ejecute la sentencia de SQL. Además, pueden generar valores de columnas, previene errores de datos, sincroniza tablas, modifica valores de una vista, etc.

Disparadores(Triggers)

La estructura básica de un *trigger* es:

Llamada de activación: es la sentencia que permite "disparar" el código a ejecutar.

Restricción: es la condición necesaria para realizar el código. Esta restricción puede ser de tipo condicional o de tipo nulidad.

Acción a ejecutar: es la secuencia de instrucciones a ejecutar una vez que se han cumplido las condiciones iniciales

Disparadores(Triggers)

Ejemplo (en SQLite) integridad referencial

```
CREATE TRIGGER fk_nota_alumno  
BEFORE INSERT ON nota  
FOR EACH ROW  
BEGIN  
    SELECT CASE  
WHEN ((SELECT carnet FROM alumno WHERE  
carnet = NEW.carnet) IS NULL)  
    THEN RAISE(ABORT, 'No existe alumno')  
END;  
END;
```


Disparadores(Triggers)

Ejemplo (en SQLite) actualización en cascada

```
CREATE TRIGGER update_matganadas  
AFTER INSERT ON nota  
WHEN new.notafinal>=6  
BEGIN
```

```
UPDATE alumno SET matganadas=matganadas+1  
WHERE alumno.carnet=new.carnet;
```

```
END
```

Pruebas

Probar los códigos en SQLite Administrator

Complemento: Guía 03