

Bases de Datos

Unidad 3: Diseño Físico de Bases de Datos

UNIDAD 3: DISEÑO FÍSICO DE BASES DE DATOS

1. Características físicas del almacenamiento de la base de datos
2. Herramientas gráficas para la implementación de la base de datos.
3. El lenguaje de definición de datos.
4. Creación modificación y eliminación de bases de datos.
5. Tipos de datos. Valores. Operadores
6. Administración de tablas
 - 6.1 Sintaxis de la instrucción CREATE TABLE
 - 6.2 Propiedades de tablas
 - 6.3 Modificación de tablas
 - 6.3.1 ALTER TABLE
 - 6.3.2 CREATE INDEX
 - 6.3.3 DROP INDEX
 - 6.3.4 RENAME TABLE
 - 6.4 Eliminación de tablas
7. Vistas
8. El lenguaje de control de datos.
9. Usuarios y privilegios

6.3 Modificación de tablas

Para modificar la estructura o el nombre de una tabla, se dispone de las instrucciones:

☐ **ALTER TABLE**

☐ **CREATE INDEX**

☐ **DROP INDEX**

☐ **RENAME TABLE**

La sintaxis de ALTER TABLE es:

ALTER TABLE tabla Especificacion_alter [,Especificacion_alter] ...

Con las especificaciones ALTER podemos establecer cada una de las modificaciones de estructura en una tabla. A continuación vamos a ver cada una de esas especificaciones

6.3.1 ALTER TABLE

Especificaciones de ALTER TABLE I:

ADD definicion_columna [AFTER col_name | FIRST]

Permite añadir una columna a la tabla. Se indicará el nombre de la columna, el tipo y, si fuera necesario, las restricciones de columna. AFTER indica que la columna se añada después de la columna que se indique. FIRST indica que la columna que se añade será la primera de la tabla.

ADD INDEX [nombre_indice] (columna_indice,...)

Permite añadir un índice sobre las columnas indicadas entre paréntesis. Opcionalmente se puede dar un nombre al índice.

ADD FULLTEXT [nombre_indice] (columna_indice,...)

Permite añadir un índice de tipo búsqueda de texto.

ADD UNIQUE [nombre_indice] (columna_indice,...)

Permite añadir un índice de tipo clave alternativa.

Aclaración: ¿Qué es un INDEX?

- Es una estructura de datos que sirve para mejorar la velocidad de las consultas.
- Se crean con una o varias columnas.
- Se deben crear con aquellas columnas que se van a utilizar para realizar queries.
- Físicamente son tablas con el índice y un puntero al elemento. Estas tablas tienen un orden adecuado para el índice además de ser más pequeñas. Por eso es más rápido realizar las queries.

6.3.1 ALTER TABLE

Especificaciones de ALTER TABLE II:

ADD PRIMARY KEY (columna1,...)

Permite crear una clave primaria, en la tabla, formada por las columnas indicadas.

ADD [CONSTRAINT [nombre_constraint]]

FOREIGN KEY [nombre_foreign] (col_clave_ajena,...)

REFERENCES tabla_origen (col1,..) [condiciones referencia]

Permite añadir una clave ajena. La especificación de clave ajena, como puede verse, es idéntica a la usada con este mismo objetivo en la creación de una tabla.

ALTER column {SET DEFAULT literal | DROP DEFAULT}

Permite hacer que una columna reciba un valor por defecto (SET DEFAULT) o, al contrario, que se elimine la condición de que una columna tome valor por defecto (DROP DEFAULT).

6.3.1 ALTER TABLE

Especificaciones de ALTER TABLE III:

CHANGE column definicion_nueva_columna [FIRST | AFTER col]

Permite modificar el nombre de columna, su tipo y las restricciones mediante lo indicado en definicion_nueva_columna.

MODIFY definicion_columna [FIRST | AFTER columna]

Permite modificar el tipo y las restricciones de la columna indicada en definicion_columna por lo indicado en esa definicion.

DROP column

Permite eliminar una columna.

DROP PRIMARY KEY

Permite eliminar una clave primaria (la restriccion de clave primaria y no las columnas que forman la clave primaria).

6.3.1 ALTER TABLE

Especificaciones de ALTER TABLE IV:

DROP INDEX nombre_índice

Permite eliminar un índice. Sirve para eliminar los índices INDEX, UNIQUE y FULLTEXT.

DROP FOREIGN KEY nombre_constraint

Permite eliminar una restricción de clave ajena. Cuando se elimine una clave ajena, puede necesitarse eliminar el índice correspondiente.

RENAME nuevo_nombre_tabla

Permite renombrar la tabla.

AUTO_INCREMENT=valor

Hace que en la columna autoincremental de esa tabla se inserte la próxima vez el valor indicado.

6.3.1 ALTER TABLE

Ejemplos de uso de ALTER TABLE:

Ejemplo 1: Añadir a una tabla automoviles una columna que indique el concesionario donde se compró el coche. Esta columna no admitirá nulos y será un índice:

```
ALTER TABLE automoviles ADD concesionario VARCHAR(25) NOT NULL INDEX;
```

Ejemplo 2: Establecer que la columna localidad de la tabla clientes sea un índice con nombre IND_LOC:

```
ALTER TABLE clientes ADD INDEX IND_LOC (localidad);
```

Ejemplo 3: Establecer que la columna matrícula de la tabla contratos es clave ajena (relacionada con la columna matrícula de la tabla automoviles) con borrado restringido y modificación en cascada. Hay que dar un nombre a la restricción:

```
ALTER TABLE contratos ADD CONSTRAINT fk_matri FOREIGN KEY(matricula)  
REFERENCES automoviles (matricula) ON DELETE RESTRICT ON UPDATE CASCADE;
```

Ejemplos de uso de ALTER TABLE:

Ejemplo 4: Eliminar la clave ajena establecida antes sobre la matricula en contratos:

ALTER TABLE contratos DROP FOREIGN KEY fk_matri;

Ejemplo 5: Eliminar la clave alternativa o índice UNIQUE correspondiente al dni de la tabla ALUMNOS al que se supone que se dio el nombre IND_DNI_ALU.

ALTER TABLE alumnos DROP INDEX IND_DNI_ALU;

6.3.2 CREATE INDEX

CREATE INDEX permite crear o añadir índices. Hemos visto que mediante una especificación de ALTER TABLE también se podían añadir índices.

La sintaxis de CREATE INDEX es:

CREATE [UNIQUE | FULLTEXT] INDEX *nombre_indice* ON *nombre_tabla* (*columna*,...)

Si no se especifica UNIQUE o FULLTEXT, el índice es un índice normal.

6.3.2 CREATE INDEX

Ejemplo 1: Crear un índice sin repeticiones para las columnas apellidos y nombre de la tabla alumnos:

CREATE UNIQUE INDEX indNomApe ON alumnos (apellidos,nombre);

Ejemplo 2: Crear un índice para los 10 primeros caracteres de la columna apellidos de la tabla profesores:

CREATE INDEX indApe10e ON profesores (apellidos(10));

6.3.2 DROP INDEX

DROP INDEX permite eliminar un índice normal, UNIQUE o FULLTEXT.

Sintaxis:

DROP INDEX *nombre_indice* ON *nombre_tabla*;

Para mostrar las características de los índices de una tabla se usa la sentencia:

SHOW INDEX FROM *nombre_tabla*;

6.3.2 RENAME TABLE

RENAME TABLE permite renombrar una tabla.

Sintaxis:

RENAME TABLE *nombre_actual* TO *nombre_nuevo*;

Ejemplo: Renombrar la tabla **Alumn** para que se llame **Alumnos**.

RENAME TABLE *alumn* TO *alumnos*;

Recuerda que esto se podría hacer también con **ALTER TABLE**

ALTER TABLE *alumn* RENAME *alumnos*;

6.4 Eliminación de tablas

La sentencia **SQL** para eliminar tablas es **DROP TABLE** y tiene la sintaxis:

DROP TABLE [IF EXISTS] tabla1 [, tabla2,...];

La cláusula **IF EXISTS** hace que si la tabla no existe, la sentencia no devuelva un mensaje y código de error(que en un procedimiento o función puede ser muy importante como se verá más adelante). Cuando se realiza un **DROP TABLE** se produce automáticamente un **COMMIT** con lo cual la tabla no es recuperable y, además, si estábamos realizando una transacción, dicha transacción habrá quedado confirmada (ya veremos también más adelante de que va esto).

Si se va a eliminar una tabla que tiene tablas relacionadas por clave ajena, se tienen en cuenta las restricciones de integridad referencial establecidas.

Si hay alguna tabla relacionada por clave ajena con restricción **NO ACTION o sin establecer, se rechaza la eliminación de la tabla.**