

Diseño físico relacional

Tema 3

Objetivos

- Definir las estructuras físicas de almacenamiento.
- Crear tablas.
- Seleccionar tipos de datos adecuados.
- Definir campo claves en las tablas.
- Implantar las restricciones establecidas en el diseño lógico.
- Verificación mediante conjuntos de pruebas.
- Uso de asistentes y herramientas gráficas.
- Uso del lenguaje de definición de datos (DDL).

1. El lenguaje de definición de datos

- El sublenguaje de SQL que permite la definición de datos es el DDL (Data Definition Language). Sus funciones son:
 - Crear tablas, índices y otros objetos de la base de datos.
 - Definir las estructuras físicas donde se almacenarán los objetos de las bases de datos.
- Instrucciones básicas de DDL:
 - **CREATE tipo_objeto Nombre Definición.** Crea un objeto de un determinado tipo con un nombre y una definición.
 - **DROP tipo_objeto Nombre.** Elimina un tipo de objeto especificado mediante nombre.
 - **ALTER tipo_objeto Nombre Modificación.** Modifica la definición de un objeto.

2. Operaciones sobre bases de datos

- Creación de bases de datos
- Modificación de una base de datos
- Borrado de bases de datos

2.1. Creación de bases de datos

CREATE DATABASE NombreBD [especificación_create];

- ***Especificación_create:***

[DEFAULT] CHARACTER SET juego_caracteres

[DEFAULT] COLLATE nombre_colación

- ***juego_caracteres:*** permite elegir los caracteres que va a usar la base de datos (utf8, latin1, latin2, etc.)

- ***nombre_colación:*** especifica cómo tratar el alfabeto del juego de caracteres.

- ***Ejemplo:***

CREATE DATABASE Startrek CHARACTER SET Latin1 COLLATE latin1_spanish.ci;

- ***SHOW DATABASES:*** para ver las bases de datos que está controlando el gestor

2.2. Modificación de una base de datos

ALTER DATABASE nombre_bd modificación;

Permite cambiar las características de funcionamiento de la base de datos. En MySQL solo se puede cambiar el juego de caracteres y su colación.

- ***Ejemplo:***

```
ALTER DATABASE Startrek COLLATE latin1_spanish_ci;
```

2.3. Borrado de bases de datos

DROP DATABASESE nombre_bd

Elimina la base de datos indicada.

- ***Ejemplo:***

DROP DATABASE Startrek;

3. Operaciones sobre tablas

- Creación de tablas
- Modificación de tablas
- Borrado de tablas
- Renombrado de tablas

3.1. Creación de tablas

```
CREATE TABLE nombre_tabla  
    [definición_create,...]  
    [opciones_tabla]
```

- ***definición_create***: especifica la definición de los campos que va a contener la tabla y sus restricciones. Por cada campo de la tabla hay que indicar un "definición_create".

```
definición_columna  
| [CONSTRAINT [símbolo]] PRIMARY KEY (nombre_col,...)  
| [CONSTRAINT [símbolo]] FOREIGN KEY (nombre_col,...)  
[definición_referencia]
```

3.1. Creación de tablas

- ***definición_columna***: especifica la definición de un campo o columna de la tabla.

nombre_columna tipo_datos [NOT NULL | NULL] [DEFAULT valor]
[UNIQUE [KEY] | [PRIMARY] KEY]
[definición_referencia]

- ***definición_referencia***: sirve para crear una clave externa.

REFERENCES nombre_tabla [(nombre_columna, ...)]
[ON DELETE {CASCADE | SET NULL | NO ACTION}]
[ON UPDATE {CASCADE | SET NULL | NO ACTION}]

3.1. Creación de tablas

- **Ejemplo1:** creación de la tabla Clientes

```
CREATE TABLE Clientes(  
    dni VARCHAR(9) PRIMARY KEY,  
    nombre VARCHAR(50),  
    direccion VARCHAR(60)  
);
```

- **Ejemplo2:** permite crear claves primarias compuestas

```
CREATE TABLE Clientes(  
    dni VARCHAR(9),  
    nombre VARCHAR(50),  
    direccion VARCHAR(60),  
    PRIMARY KEY (dni)  
);
```

3.1. Creación de tablas

- **Ejemplo3:** definición de clave externa

```
CREATE TABLE Mascotas(  
    codigo INTEGER PRIMARY KEY,  
    nombre VARCHAR(50),  
    raza VARCHAR(50),  
    cliente VARCHAR(9) REFERENCES Clientes(dni)  
);
```

- **Ejemplo4:** definición de clave externa

```
CREATE TABLE Mascotas(  
    codigo INTEGER PRIMARY KEY,  
    nombre VARCHAR(50),  
    raza VARCHAR(50),  
    cliente VARCHAR(9),  
    FOREIGN KEY (cliente) REFERENCES Clientes(dni)  
);
```

3.1.1. Implementación de restricciones

- Las opciones **ON DELETE** Y **ON UPDATE** establecen el comportamiento del gestor en caso de que las filas de la tabla referenciada se borren o actualicen. Comportamientos:
 - **NO ACTION:** si se intenta un borrado o actualización sobre la tabla, la operación se impide.
 - **CASCADE:** la operación se propaga en cascada a la tabla hija (si se actualizan registros de la tabla padre, también lo hacen aquellos registros de la tabla hija que están referenciados).
 - **SET NULL:** se establece a NULL la clave externa afectada.
- Acción por defecto NO ACTION.

3.1.1. Implementación de restricciones

- ***Ejemplo:***

```
CREATE TABLE Clientes(  
    dni VARCHAR(9) PRIMARY KEY,  
    nombre VARCHAR(50),  
    direccion VARCHAR(60)  
    );
```

```
CREATE TABLE Mascotas(  
    codigo INTEGER PRIMARY KEY,  
    nombre VARCHAR(50),  
    raza VARCHAR(50),  
    cliente VARCHAR(9) REFERENCES Clientes(dni)  
        ON DELETE CASCADE ON UPDATE SET NULL  
    );
```

3.1.2. Tipos de datos

Tipo de dato	Naturaleza	Tamaño/Formato
TINYINT [UNSIGNED]	Entero	1 byte
SMALLINT [UNSIGNED]	Entero	2 bytes
MEDIUMINT [UNSIGNED]	Entero	3 bytes
INT [UNSIGNED]	Entero	4 bytes
BIGINT [UNSIGNED]	Entero	8 bytes
INTEGER [UNSIGNED]	Entero	4 bytes
DOUBLE [UNSIGNED]	Real aproximado	8 bytes
FLOAT [UNSIGNED]	Real aproximado	4 bytes
DECIMAL(longitud, decimales)	Real exacto	Variable
NUMERIC(longitud, decimales)	Real exacto	Variable
DATE	Fecha	'aaaa-mm-dd'
TIME	Hora	'hh:mm:ss'
TIMESTAMP	Fecha y hora	'aaaa-mm-dd hh:mm:ss'
DATETIME	Fecha y hora	'aaaa-mm-dd hh:mm:ss'

3.1.2. Tipos de datos

Tipo de dato	Naturaleza	Tamaño/Formato
CHAR(longitud)	Caracteres	Longitud fija
VARCHAR(longitud)	Caracteres	Longitud fija
BLOB	Objetos binarios	Longitud variable
TEXT	Campos Memo	Longitud variable
ENUM(valor1,valor2,valor3...)	Enumeraciones	Lista de valores
SET(valor1,valor2,valor3...)	Conjuntos	Conjuntos de valores

3.1.3. Características de la creación de tablas

Opciones de tabla para MySQL:

Opciones_tabla: **opción_tabla** [**opción_tabla**] ...

- ***Opción_tabla:***

- ENGINE = nombre_motor
 - | AUTO_INCREMENT = valor
 - | [DEFAULT] CHARACTER SET juego_caracteres [COLLATE colación]
 - | CHECKSUM = {0 | 1}
 - | COMMENT = 'string'
 - | MAX_ROWS = valor
 - | MIN_ROWS = valor

3.1.3. Características de la creación de tablas

- ***ENGINE = nombre_motor***

Se indica el motor de almacenamiento para la tabla.

El almacenamiento físico de una tabla en MySQL está controlada por un software especial denominado Motor de almacenamiento.

Puede ser: innodb, mylsam (por defecto), Memory

- ***AUTO_INCREMENT***: permite indicar el valor inicial para campos de tipo AUTO_INCREMENT. Indica que este campo es autoincrementado después de cada inserción.
- ***[DEFAULT] CHARACTER SET***: especifica el conjunto de caracteres para la tabla.
- ***COLLATE***: define la colación por defecto de la tabla.

3.1.3. Características de la creación de tablas

- **CHECKSUM:** mantiene una suma de verificación para todos los registros.
- **COMMENT:** es un comentario para la tabla (máximo 60 caracteres).
- **MAX_ROWS:** es el máximo número de registros que se quiere almacenar en la tabla. No es un límite, sólo un indicador de la al menos se pueden almacenar estos registros.
- **MIN_ROWS:** es el mínimo número de registros que se planea almacenar en la tabla.

3.1.4. Consulta de las tablas de una BD

SHOW TABLES

Para consultar las tablas disponibles de una base de datos en MySQL.

3.1.5. Consulta de la estructura de una tabla

- ***DESCRIBE [esquema.]nombre_tabla***

Para conocer la estructura de una tabla ya creada. Muestra un listado con las columnas de la tabla, aportando información sobre los tipos de datos, restricciones, etc.

- ***Ejemplo:***

```
DESCRIBE nba.equipos;
```

3.2. Modificación de tablas

**ALTER TABLE nombre_tabla
especificación_alter [, especificación_alter]**

Especificación_alter:

- ADD definición_columna [FIRST | AFTER nombre_columna]
- | ADD (definición_columna,...)
- | ADD [CONSTRAINT [símbolo]]
 PRIMARY KEY (nombre_columna,...)
- | ADD [CONSTRAINT [símbolo]]
 UNIQUE (nombre_columna,...)
- | ADD [CONSTRAINT [símbolo]]
 FOREIGN KEY (nombre_columna,...)
 [definición_referencia]
- | CHANGE [COLUMN] anterior_nombre_columna definición_columna
 [FIRST | AFTER nombre_columna]
- | RENAME COLUMN anterior_nombre_col TO nuevo_nombre_col
- | MODIFY definición_columna [FIRST | AFTER nombre_columna]
- | DROP COLUMN nombre_columna
- | DROP PRIMARY KEY
- | DROP FOREIGN KEY fk_símbolo
- | opciones_tabla

3.2. Modificación de tablas

- **ADD:** permite añadir una columna, se puede especificar dónde se va a insertar.
- **MODIFY:** se cambia el tipo de datos de una columna y se añaden restricciones.
- **DROP:** se pueden eliminar restricciones de claves externas y primarias, dejando el tipo de dato y su contenido intacto.
- **CHANGE:** para cambiar el nombre de una tabla.
- **Ejemplo:** Añadir un campo

```
ALTER TABLE Mascotas ADD Especie VARCHAR(10) AFTER Raza;
```

3.3. Borrado de tablas

DROP TABLE nombre_tabla [, nombre_tabla]...

- ***Ejemplo:***

DROP TABLE Mascotas;

DROP TABLE Clientas, Empleados;

3.4. Renombrado de tablas

**RENAME TABLE nombre_tabla TO nuevo_nombre_tabla
[, nombre_tabla TO nuevo_nombre_tabla]...**

- ***Ejemplo:***

RENAME TABLE Mascotas TO Animales;