

Sentencia CREATE

La sentencia CREATE permite crear la estructura de una base de datos o de una tabla. La sintaxis es la siguiente:

Creación de bases de datos:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_db [opciones_creación]
```



- DATABASE y SCHEMA son sinónimos, por lo que podemos usarlos de forma indistinta.
- Si indicamos [IF NOT EXISTS], nos mostrará una advertencia en caso de que exista una base de datos con el mismo nombre. Si no lo indicamos y existe una base de datos con el mismo nombre se producirá un error.
- nombre_db indica el nombre de la base de datos que queremos crear.
- Las posibles opciones de creación son:
 - [DEFAULT] CHARACTER SET [=] nombre_juego_caracteres
 - [DEFAULT] COLLATE [=] nombre_sistema_ordenación
- CHARACTER SET permite establecer el conjunto de caracteres por defecto para las tablas que se crean en esta base de datos. En caso de no especificarse, se toma el que tiene definido por defecto en la configuración del servidor.
- COLLATE establece el conjunto de reglas para comparar caracteres. En caso de no especificarse, se selecciona el que se corresponde por defecto al juego de caracteres elegido.
- Ejemplo:

```
CREATE DATABASE IF NOT EXISTS prueba DEFAULT CHARACTER SET utf8 DEFAULT  
COLLATE utf8_spanish_ci;
```

Creación de tablas:

```
CREATE [TEMPORARY] TABLE nombre_tabla (nombre_columna tipo_de_dato  
[propiedades_de_la_columna] [, ...]) [opciones_de_tabla]  
[opciones_de_partición]
```

- TEMPORARY es opcional, y permite crear una tabla temporal mientras dure la sesión.
- La definición de cada columna consiste en darle un nombre, asignarle un tipo de dato y asociarle las restricciones necesarias. Las dos primeras partes (nombre y tipo) son obligatorias. MariaDB tiene soporte para muchos tipos diferentes de datos, indicados en el documento “Tipos de datos MariaDB.pdf”.
- Para definir las propiedades de la columna, se pueden usar las cláusulas:
 - NOT NULL (no permite que se tome el valor NULL).
 - NULL (permite que la columna tome el valor NULL).

- UNSIGNED (permite que se tomen únicamente valores positivos en columnas de tipo numérico).
- ZEROFILL (rellena con ceros a la izquierda las columnas de tipo numérico).
- BINARY (diferencia entre mayúsculas y minúsculas en columnas de tipo cadena de caracteres).
- DEFAULT *valor* (indica el valor por defecto para la columna, en caso de no asignarle ningún valor cuando se añade una fila a la tabla. Está permitido utilizar algunos valores proporcionados por funciones del sistema en lugar de valores constantes definidos por el usuario. Por ejemplo, USER se refiere al usuario que establece la conexión con la base de datos, o CURRENT_TIMESTAMP, que se refiere a la fecha y hora actual).
- AUTO_INCREMENT (indica que el valor que toma la columna es calculado por el sistema en caso de no asignarle ningún valor o al ser asignado un valor nulo. El sistema calcula el valor sumándole 1 al valor que toma la columna para la última fila añadida).
- PRIMARY KEY (define la columna como clave primaria. También se puede definir como clave primaria creando una restricción).
- Algunas de las opciones de tabla son:
 - DATA DIRECTORY = 'directorio' (especifica la ruta absoluta en la que se almacenan los datos).
 - INDEX DIRECTORY = 'directorio' (especifica la ruta absoluta en la que se almacenan los índices).
 - ENGINE = {ISAM, MyISAM, InnoDB, ...} (indica el motor de almacenamiento asociado a la tabla).
 - [DEFAULT] CHARACTER SET nombre_juego_caracteres.
 - [DEFAULT] COLLATE nombre_sistema_ordenación.
 - [AUTO_INCREMENT = número] (indica el número de inicio para la columna de tipo autoincremental).
 - COMMENT texto (añade un comentario a la tabla, normalmente para indicar qué es lo que representa).
- Las opciones de particionamiento se explicarán en un apartado posterior.
- Ejemplo:

```
create table test.fotografia
( id integer unsigned not null auto_increment,
  titulo varchar(60) not null default 'sin título',
  autor varchar(60) not null comment 'Apellidos y nombre del autor o autores',
  fecha date null comment 'Fecha en la que fue tomada',
  primary key (id)
) engine = InnoDB;
```

- Ejemplo usando una restricción para crear una clave primaria compuesta:

```
create table test.fotografia
( titulo varchar(60) not null default 'sin título',
  autor varchar(60) not null comment 'Apellidos y nombre del autor o au-
tores',
  fecha date null comment 'Fecha en la que fue tomada',
  primary key (titulo,autor)
) engine = InnoDB;
```