

SQL ANSI

Lenguaje de Definición de Datos DDL



Sentencia CREATE TABLE

Crear tablas en MySQL



NombreTabla1	
Columna1	INT
Columna2	VARCHAR(45)
Columna3	VARCHAR(45)
Columna4	DATE
Indexes	

NombreTabla2	
Columna1	INT
Columna2	VARCHAR(45)
Columna3	VARCHAR(45)
Columna4	DATE
Indexes	

Como crear una tabla en MySql workbench

La sentencia básica de MySQL que se utiliza para crear una tabla es **create table** su función es crear una tabla con las características que le indiquemos seguidamente del comando en referencia.

Uno de los aspectos importantes a tomar en cuenta es que nuestra tabla estará compuesta por columnas y a cada columna debemos especificarle un tipo de dato para identificar si son datos numéricos, cadenas (alfanuméricos) o fechas y horas los que estaremos guardando en dicha columna.

Sintaxis para crear tablas en Mysql

La sintaxis de como crear una tabla en MySQL es la siguiente:

```
CREATE TABLE Nombre_Tabla  
(  
Nombre_Columna1 Tipo_de_Dato (longitud),  
Nombre_Columna2 Tipo_de_Dato (longitud),  
Nombre_Columna3 Tipo_de_Dato (longitud),  
....  
);
```

Los parámetros "**Nombre_Columna**" especifican los nombres de las columnas que integran las tablas.
Los parámetros "**Tipo_de_Dato**" especifican que tipo de datos admitirá esa columna (ej. varchar, integer, decimal, date, etc.).

El parámetro "**Longitud**" especifica la longitud máxima de caracteres que admitirá la columna de la tabla.
Cabe aclarar que hay varias propiedades que pueden ser aplicadas a las columnas de la tabla, por ejemplo una de las más comunes es **primary key**, la cual nos permite indicar que columna será llave primaria, también podemos establecer que alguna columna no acepte valores nulos, a través de la propiedad **not null**.

Código para crear tablas en MySql

A continuación presentare un ejemplo de cómo crear una tabla llamada **"Alumnos"** con cinco columnas, el código es el siguiente:

MySQL

```
CREATE DATABASE Tutorial;

use Tutorial;

CREATE TABLE Alumnos
(
  IdAlumno int primary key not null,
  Nombre varchar (25),
  Apellido Varchar (25),
  Edad int,
  Direccion_Residencia varchar (50)
);
```



En el código anterior estamos creando la tabla “**Alumnos**” que tiene 5 columnas, podemos darnos cuenta que una de las novedades es en la columna IdAlumno, ya que tenemos la propiedad **PRIMARY KEY**, la cual indica que es la clave primaria que identifica de manera única cada registro/fila de la tabla, así mismo con la propiedad **not null** estamos indicando que esa columna es indispensable, es decir que siempre debe ser ingresado un IdAlumno.

Cabe aclarar que la propiedad not null puede ser implementada en otras columnas que consideremos que son indispensables, por ejemplo el nombre del alumno es fundamental que sea ingresado, en ese caso deberíamos agregarle la propiedad not null para garantizar que el nombre siempre será ingresado.

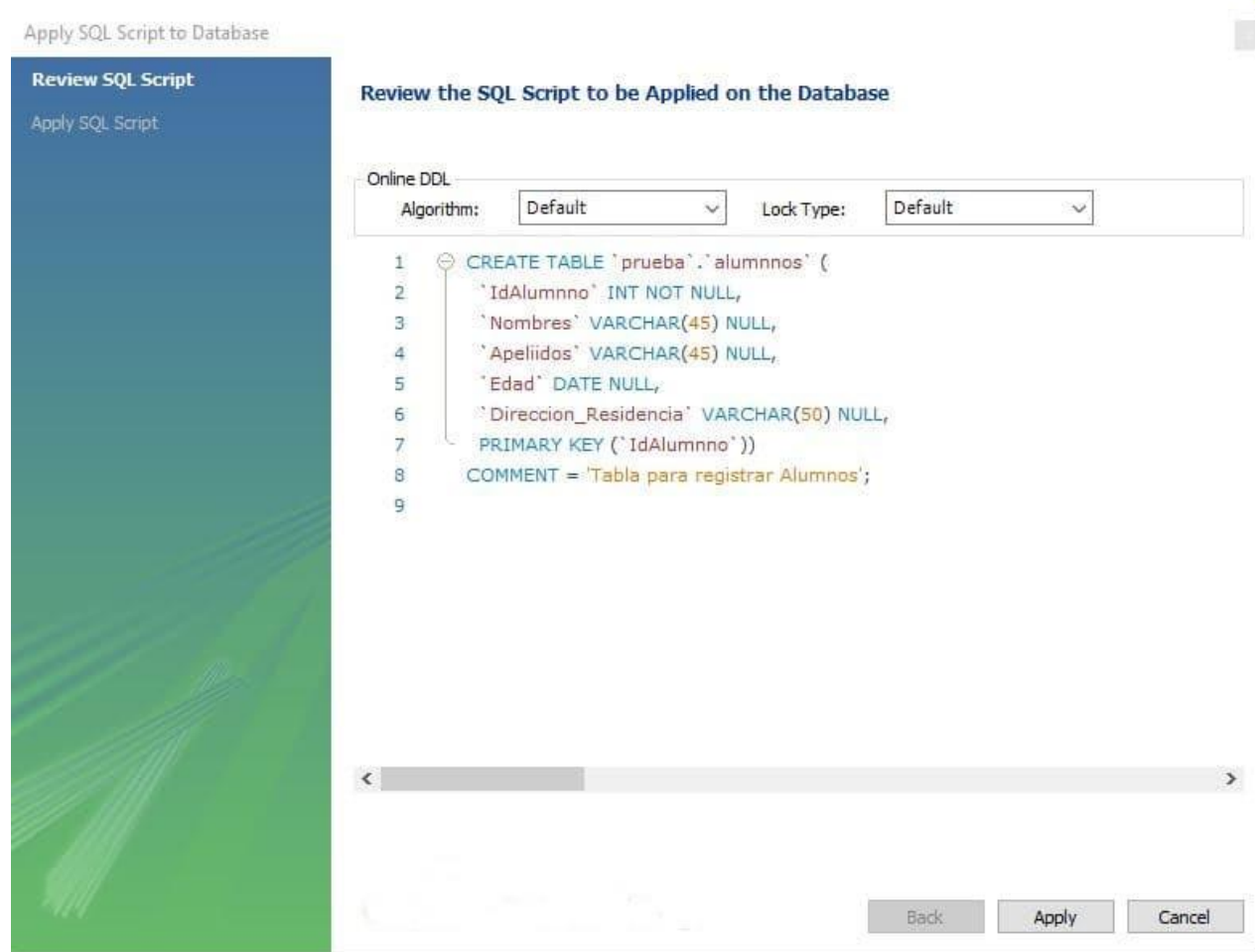
Para efectos de este ejemplo no ha sido aplicado, sin embargo cuando estamos creando unas Base de Datos con una buena estructura, debemos tomar en cuenta todos esos aspectos.

A continuación presentare un ejemplo de cómo crear una tabla llamada **"Cliente"** con seis columnas, el código es el siguiente:

```
-- -----  
-- Table `EMPRESA`.`CLIENTE`  
-- -----  
⊖ CREATE TABLE IF NOT EXISTS `EMPRESA`.`CLIENTE` (  
    `idCLIENTE` INT NOT NULL AUTO_INCREMENT,  
    `NOMBRE` VARCHAR(45) NOT NULL,  
    `APELLIDO` VARCHAR(45) NOT NULL,  
    `CUIT` INT NOT NULL,  
    `TELEFONO` INT NULL,  
    `FECHA` DATE NULL,  
    PRIMARY KEY (`idCLIENTE`))
```


A continuación un ejemplo de cómo crear una tabla llamada "**Ventas**" con cinco columnas, con su **Foreign Key** correspondiente. El código es el siguiente:

```
-- -----  
-- Table `EMPRESA`.`VENTA`  
-- -----  
CREATE TABLE IF NOT EXISTS `EMPRESA`.`VENTA` (  
  `idVENTA` INT NOT NULL,  
  `NUMERO` INT NOT NULL,  
  `FECHA` DATE NOT NULL,  
  `Id_CLIENTE` INT NOT NULL,  
  PRIMARY KEY (`idVENTA`),  
  INDEX `IdCLIENTE_idx` (`Id_CLIENTE` ASC) VISIBLE,  
  CONSTRAINT `IdCLIENTE`  
    FOREIGN KEY (`Id_CLIENTE`)  
    REFERENCES `EMPRESA`.`CLIENTE` (`idCLIENTE`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

Existe otra forma de **crear tablas sin necesidad de usar código o script** de la Base de Datos, esto se realiza a través de MySQL Workbench que a través de sus funciones nos facilita realizar esta tarea.

Es importante mencionar que cuando creamos una tabla de forma gráfica también obtenemos el script o código de lo que hemos realizado, tal como se muestra en la imagen anterior.

Inclusión de restricciones

- Las restricciones aplican reglas a nivel de tabla.
- Las restricciones impiden la supresión de una tabla si hay dependencias.
- Los siguientes tipos de restricciones son válidos:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK



Inclusión de restricciones

Restricción	Descripción
NOT NULL	Especifica que la columna no puede contener un valor nulo.
UNIQUE	Especifica una columna o combinación de columnas cuyos valores deben ser únicos para todas las filas de la tabla.
PRIMARY KEY	Identifica de forma única cada fila de la tabla.
FOREIGN KEY	Establece y aplica la integridad referencial entre la columna y la columna de la tabla a la que se hace referencia; por ejemplo, busca los valores de una tabla que coinciden con los valores de la otra tabla.
CHECK	Especifica una condición que debe ser verdadera.

Definición de restricciones

✓ Ejemplo de una restricción a nivel de columna:

```
CREATE TABLE employees(  
    employee_id  NUMBER(6)  
        CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
    first_name   VARCHAR2(20) ,  
    ...);
```

1

✓ Ejemplo de una restricción a nivel de tabla:

```
CREATE TABLE employees(  
    employee_id  NUMBER(6) ,  
    first_name   VARCHAR2(20) ,  
    ...  
    job_id       VARCHAR2(10) NOT NULL,  
    CONSTRAINT emp_emp_id_pk  
        PRIMARY KEY (EMPLOYEE_ID) );
```

2

Definición de restricciones

- ✓ Las restricciones se crean normalmente al mismo tiempo que la tabla.
- ✓ Las restricciones se pueden agregar a una tabla después de su creación y se pueden desactivar temporalmente.

Ambos ejemplos de la diapositiva crean una restricción de clave primaria en la columna `EMPLOYEE_ID` de la tabla `EMPLOYEES`.

- ✓ En el primer ejemplo se utiliza la sintaxis de nivel de columna para definir la restricción.
- ✓ En el segundo ejemplo se utiliza la sintaxis de nivel de tabla para definir la restricción. Encontrará más información sobre la restricción de clave primaria más adelante en esta lección.

Restricción FOREIGN KEY: Palabras Clave

- FOREIGN KEY: define la columna en la tabla secundaria a nivel de restricción de tabla
- REFERENCES: identifica la tabla y la columna en la tabla principal
- ON DELETE CASCADE: suprime las filas dependientes de la tabla secundaria cuando se suprime una fila de la tabla principal
- ON DELETE SET NULL: convierte los valores de clave ajena dependiente en nulos.