Módulo 3. Implementación del modelo relacional en un SGBD

- Lenguaje DDL
- Lenguaje DML
- Generación del script SQL mediante ERD Plus

Copyright 2026

Lenguaje DDL

Para almacenar y tratar la información esquematizada en los modelos E/R y relacional por medio de un Sistema Gestor de Bases de Datos se utiliza el **lenguaje SQL**. SQL son las siglas de Structured Query Language (Lenguaje de Consulta Estructurado).

Es un lenguaje que permite interactuar con los SGBD Relacionales para especificar las operaciones que se desean realizar sobre los datos y su estructura. Es un lenguaje declarativo, lo cual quiere decir que en él se especifica al Sistema qué se quiere obtener pero no cómo conseguirlo.

Lenguaje DDL

Para la creación de las tablas de la base de datos, se utiliza el lenguaje de descripción de datos (DDL, Data Definition Language).

A continuación se explica la sintaxis de cada sentencia para llegar a crear una base de datos SQL.

> Hay que tener en cuenta que en este curso nos centramos en el lenguaje SQL del SGBD MySQL, y cada uno tiene sus particularidades y sus diferencias, pero en lineas generales todos deben seguir el estándar SQL y, por lo tanto, tener todos más o menos la misma sintaxis.

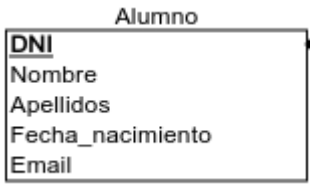
Toda la sintaxis detallada del lenguaje MySQL la podéis encontrar en la web de documentación de MySQL, aquí. A continuación se dan las nociones básicas de las sentencias básicas del lenguaje DDL.

Creación de tablas

```
CREATE TABLE nombreTabla (
    columna1 tipoDato,
    columna2 tipoDato,
    ...
    columnaN tipoDato)
```

Para crear la tabla Alumno:

Copyright 2026



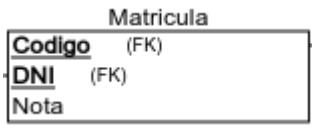

```
CREATE TABLE Alumno (
  DNI CHAR(9),
  Nombre VARCHAR(20),
  Apellidos VARCHAR(50),
  Fecha_nacimiento DATE,
  Email VARCHAR(50)
);
```

Claves primarias

Siguiendo con el ejemplo de la tabla Alumno, la clave primaria está formada por un solo campo, por lo que se puede poner directamente en la columna que es la clave primaria la palabra clave **PRIMARY KEY**:

```
CREATE TABLE Alumno (
  DNI CHAR(9) primary key,
  Nombre VARCHAR(20),
  Apellidos VARCHAR(50),
  Fecha_nacimiento DATE,
  Email VARCHAR(50)
);
```

Si la clave primaria está formada por varias columnas, se debe poner al final, después de la declaración de todas las columnas:




```
CREATE TABLE Matricula (
  Nota FLOAT,
  Codigo CHAR(5),
```

```
  DNI CHAR(9),
  PRIMARY KEY (Codigo, DNI)
);
```

Claves ajenas

La tabla Matricula tiene dos claves ajenas, en la columna Codigo y en la columna DNI.
El diagrama de las tablas Alumno, Matricula y Asignatura:



Se crea con las sentencias SQL siguientes:

```
CREATE TABLE Alumno (
  DNI CHAR(9),
  Nombre VARCHAR(20),
  Apellidos VARCHAR(50),
  Fecha_nacimiento DATE,
  Email VARCHAR(50),
  PRIMARY KEY (DNI)
);

CREATE TABLE Asignatura (
  Codigo CHAR(5),
  Nombre VARCHAR(30),
  Duracion VARCHAR(9),
  Horas_semanales INT,
  Codigo_grado CHAR(5),
  PRIMARY KEY (Codigo)
);

CREATE TABLE Matricula (
```

```
  Nota FLOAT,
  Codigo CHAR(5),
  DNI CHAR(9),
  PRIMARY KEY (Codigo, DNI),
  FOREIGN KEY (Codigo) REFERENCES Asignatura(Codigo),
  FOREIGN KEY (DNI) REFERENCES Alumno(DNI)
);
```

La clave ajena se indica después de todas las columnas, se pone FOREIGN KEY, a continuación entre paréntesis el nombre de la columna o columnas que forman parte de la clave ajena, luego la palabra REFERENCES y el nombre de la tabla con el de la columna a la que hace referencia como clave primaria, entre paréntesis.

Algunas restricciones de integridad

- **Prohibir nulos**. Esto obliga a que la columna tenga que tener obligatoriamente un valor para que sea almacenado el registro. Para esto, basta con añadir en la columna que no se permite dejar vacía en ninguna fila, después del tipo de dato, las palabras claves NOT NULL.

```
CREATE TABLE Alumno(
  DNI CHAR(9) NOT NULL,
  Nombre VARCHAR(20) NOT NULL,
  Apellidos VARCHAR(50) NOT NULL,
  Fecha_nacimiento DATE NOT NULL,
  Email VARCHAR(50) NOT NULL,
  PRIMARY KEY (DNI)
);
```

- **Valores no repetidos.** Esto obliga a que el contenido de una o más columnas no puedan repetir valores en la tabla. Basta con añadir al final de la columna la palabra clave UNIQUE.

  Si no permitimos que dos grados tengan el mismo nombre:

```
CREATE TABLE Grado(
  Codigo CHAR(5),
  Nombre VARCHAR(50) UNIQUE,
  PRIMARY KEY (Codigo)
```

```
);
```

Puede ser que los valores que no se permiten repetir sea la combinación de varios campos. En ese caso se debe poner al final de la declaración de todas las columnas, igual que se indicaba la clave primaria.

Eliminación de tablas

La orden **DROP TABLE** seguida del nombre de una tabla, permite eliminar la tabla en cuestión.

```
DROP TABLE Grado;
```

Normalmente, el borrado de una tabla es irreversible y no hay ninguna petición de confirmación, por lo que conviene ser muy cuidadoso con esta operación.

Modificación de tablas

- Cambiar el nombre a una tabla.

```
ALTER TABLE nombre_viejo RENAME nombre_nuevo;
```

- Añadir columnas.

```
ALTER TABLE Alumno ADD (Direccion VARCHAR(100));
```

- Eliminar columnas.

```
ALTER TABLE Alumno DROP (Direccion);
```

- Modificar una columna. Basta con volver a definirla mediante la siguiente sintaxis:

```
ALTER TABLE Alumno Modify (Email VARCHAR(100));
```

Se ha cambiado el tipo de dato de la columna email y además ahora se permite dejar en blanco.

> Se pueden también añadir y eliminar restricciones, pero esto y más se puede ver en la documentación oficial de SQL.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

Copyright 2026

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
GOBIERNO DE ARAGON

Copyright 2026

Lenguaje DML

El lenguaje de manipulación de datos (Data Manipulation Language) permite indicar al sistema las operaciones que se quieren realizar con los datos almacenados en las estructuras creadas por medio de las sentencias DDL. Por ejemplo son las sentencias que permiten: generar consultas, ordenar, filtrar, añadir, modificar, borrar, etc.

Inserción de datos

La inserción de datos a una tabla se realiza mediante la instrucción **INSERT**. Su sintaxis fundamental es:

INSERT INTO tabla (listaDeColumnas) VALUES (valor1,valor2, ...)

La tabla representa la tabla a la que se quiere añadir el registro y los valores que siguen a VALUES son los valores que se dan a los distintos campos del registro. Si no se especifica la lista de campos, la lista de valores debe seguir el orden de las columnas según fueron creados.

La lista de campos a rellenar se indica si no se quieren rellenar todos los campos. Los campos no rellenados explícitamente con la orden INSERT, se rellenan con su valor por defecto o bien se dejan vacíos (valor nulo) si no se indica valor alguno. Si algún campo tiene restricción de obligatoriedad (NOT NULL), ocurrirá un error si no rellenamos el campo con algún valor.

Inserción de un registro en la tabla Alumno:

```
INSERT INTO ALUMNO VALUES ('25458982A','Juan','Pérez
Gómez','01/06/2003','jperezgomez@gmail.com');
```

Actualización de registros

La modificación de los datos de los registros lo implementa la instrucción UPDATE. Sintaxis:

UPDATE tabla SET columna1=valor1 ,columna2=valor2,... WHERE condición

Se modifican las columnas indicadas en el apartado SET con los valores indicados. La cláusula WHERE permite especificar qué filas serán modificados, aquellas que cumplan la condición indicada.

Copyright 2026

Modificación de la fecha de nacimiento del registro insertado en el ejemplo anterior:

```
UPDATE ALUMNO SET fecha_nacimiento='01/06/1999' where DNI LIKE '25458982A';
```

Borrado de registros

Se realiza mediante la instrucción DELETE:

DELETE FROM tabla WHERE condición

Elimina todas las filas de la tabla que cumplan la condición indicada.

Borrado del registro insertado antes:

```
DELETE FROM ALUMNOS WHERE DNI LIKE '25458982A';
```

> Con la sentencia DELETE hay que tener cuidado. Si se ejecuta la sentencia sin la cláusula WHERE (sin ninguna condición), se elimina todo el contenido de la tabla, es decir, se vacía la tabla.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

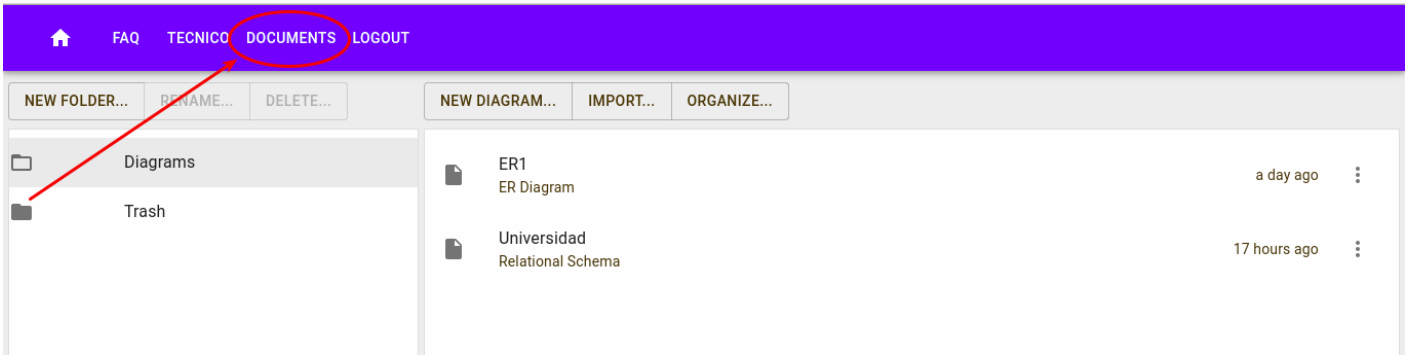


Copyright 2026

Generación del script SQL mediante ERD Plus

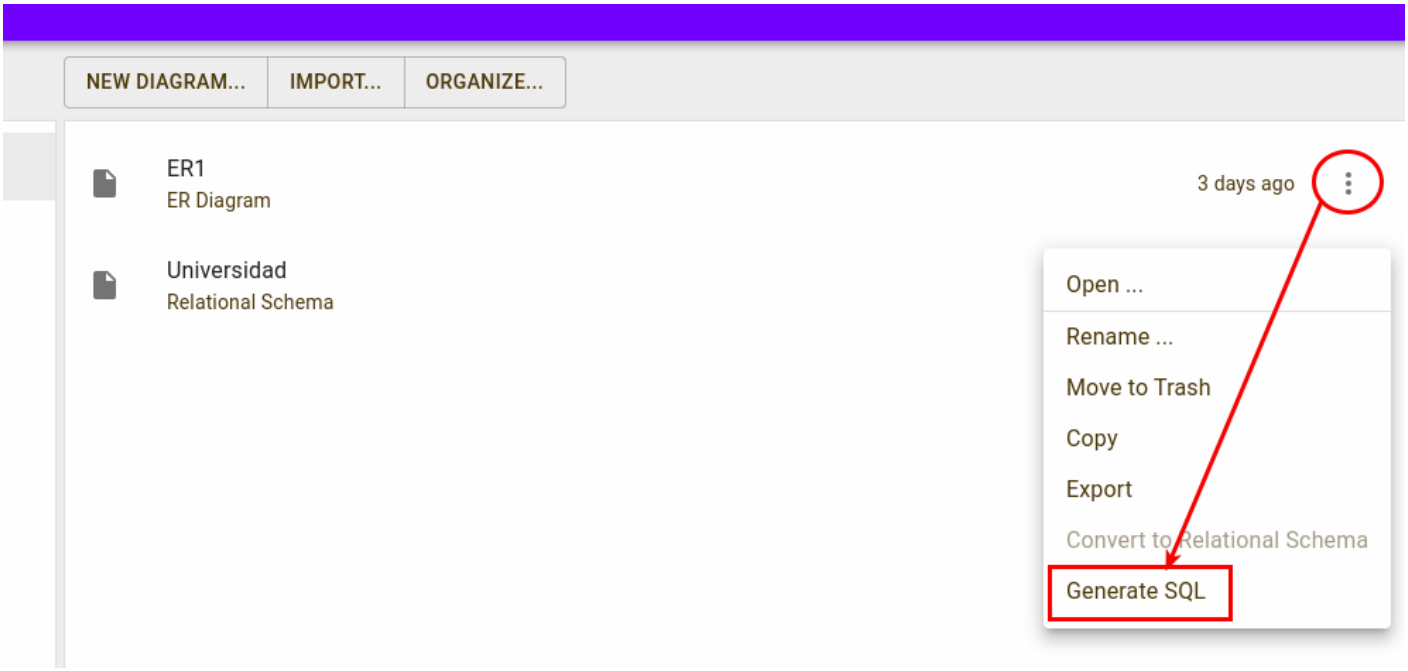
Teniendo el diagrama relacional en la herramienta ERD Plus, se puede generar el SQL de creación de las tablas, con sus restricciones.

Hay que clicar en el elemento “Documents” del menú principal y se abren todos los diseños que tenemos creados en nuestra cuenta:



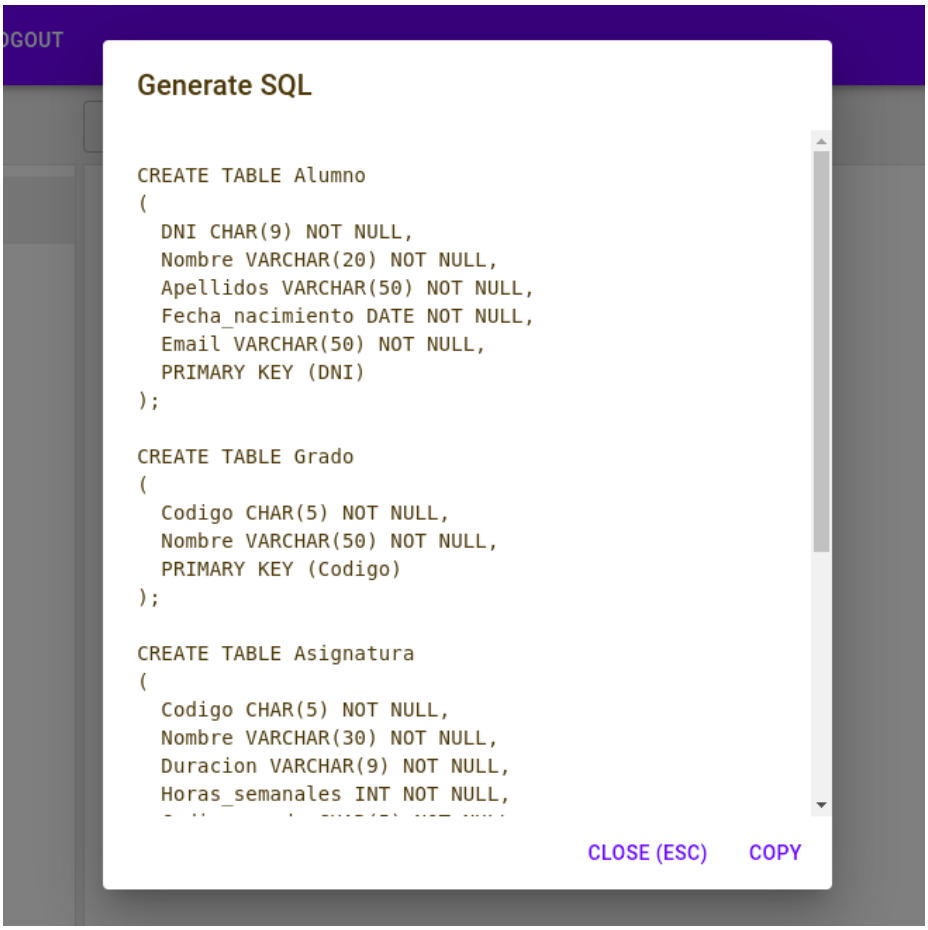
Clicando en los 3 puntos de la derecha del modelo que se desea exportar en SQL, se despliega un menú en el que hay que seleccionar la opción “Generate SQL”.

Copyright 2026



Pinchando en “Generate SQL” se abre una ventana con las sentencias SQL que al ejecutarlas en un sistema gestor de base de datos, creará las tablas con las restricciones del modelo diseñado.

Copyright 2026



Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU