

Nombre de la práctica	Creación de bases de datos en MYSQL			No.	01
Asignatura:	Taller de bases de datos	Carrera:	Ingeniería sistemas	en	Duración de la práctica (Hrs)

Nombre del alumno: Jesús Navarrete Martínez

Grupo: 3501

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado:

- Equipo de computo

IV. Desarrollo de la práctica:

• Problemática 1:

Una escuela necesita gestionar los alumnos y los cursos en los que están inscritos. Cada alumno puede inscribirse en varios cursos y cada curso puede tener varios alumnos.

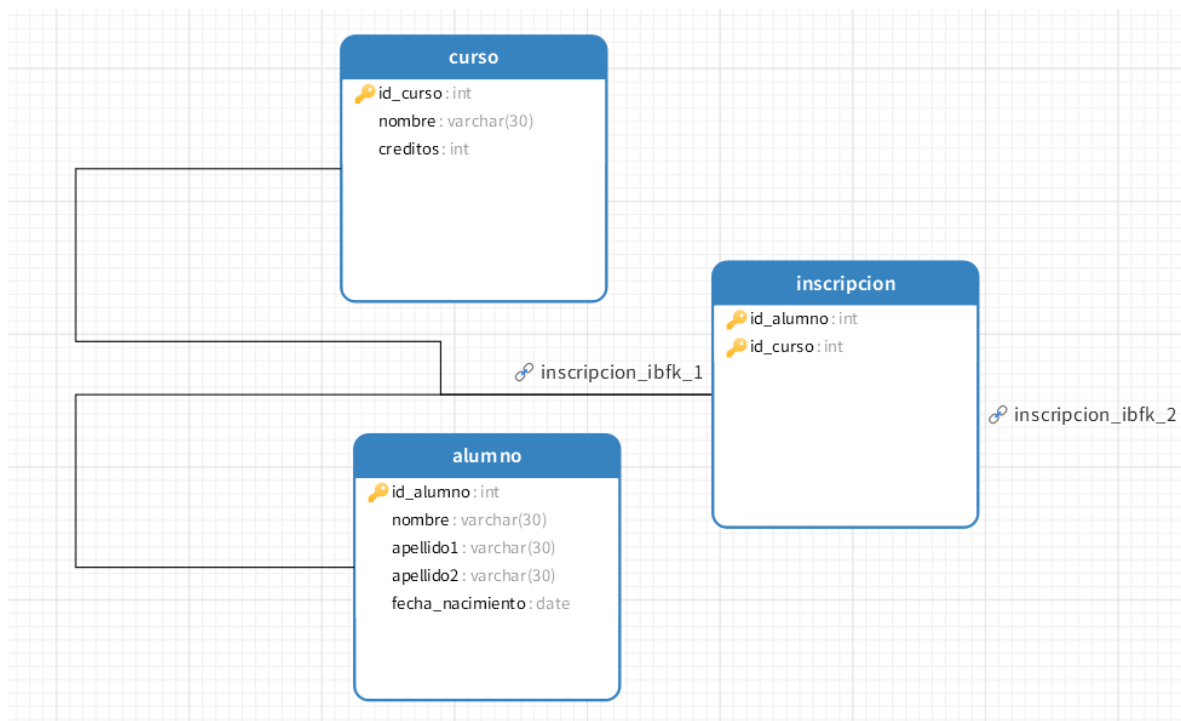
Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
mysql80-localhost escuela Run Stop Explain
1 CREATE TABLE Alumno (
2   id_alumno INT PRIMARY KEY AUTO_INCREMENT,
3   nombre VARCHAR(30) NOT NULL,
4   apellido1 VARCHAR(30) NOT NULL,
5   apellido2 VARCHAR(30) NOT NULL,
6   fecha_nacimiento DATE
7 );
8
9
10 CREATE TABLE Curso (
11   id_curso INT PRIMARY KEY AUTO_INCREMENT,
12   nombre VARCHAR(30) NOT NULL,
13   credits INT CHECK ( credits > 0 )
14 );
15
16 CREATE TABLE inscripcion (
17   id_alumno INT,
18   id_curso INT,
19   PRIMARY KEY (id_alumno, id_curso),
20   FOREIGN KEY ( id_alumno) REFERENCES
21   alumno(id_alumno) ON UPDATE CASCADE ON DELETE RESTRICT,
22   FOREIGN KEY ( id_curso) REFERENCES
23   curso(id_curso) ON UPDATE CASCADE ON DELETE RESTRICT
24 );
25
```

Uso de distintos tipos de restricciones:

- **PRIMARY KEY:** sirve para definir la clave primaria de la tabla, lo que significa que este campo debe ser único para cada registro y no puede contener valores nulos.
- **AUTO_INCREMENT:** Hace que el valor definido se incremente automáticamente con cada nuevo registro, asegurando que cada alumno tenga un identificador único.
- **VARCHAR(30):** Define un campo de texto variable de hasta 30 caracteres.
- **NOT NULL:** Indica que un campo no puede contener valores nulos, es decir, siempre debe tener un valor.
- **DATE:** Indica que este campo almacenará una fecha
- **CHECK (creditos > 0):** Esta restricción asegura que el valor de créditos sea siempre mayor que 0, evitando la posibilidad de crear cursos sin créditos.
- **PRIMARY KEY (id_alumno, id_curso):** Define una clave primaria compuesta formada por los campos id_alumno e id_curso, lo que asegura que no se pueda inscribir a un alumno en el mismo curso más de una vez.
- **FOREIGN KEY:** Establece una relación entre la tabla inscripcion y la tabla Alumno. El campo id_alumno en inscripcion debe coincidir con un id_alumno en la tabla Alumno.
- **ON UPDATE CASCADE:** Si el id_alumno en la tabla Alumno cambia, este cambio se propagará a la tabla inscripcion.
- **ON DELETE RESTRICT:** Evita que un registro en la tabla Alumno sea eliminado si tiene registros relacionados en inscripcion.

Tablas Resultantes:



Problemática 2:

Una tienda necesita gestionar las ventas, los clientes y los productos. Cada venta tiene un cliente asociado y puede incluir varios productos.

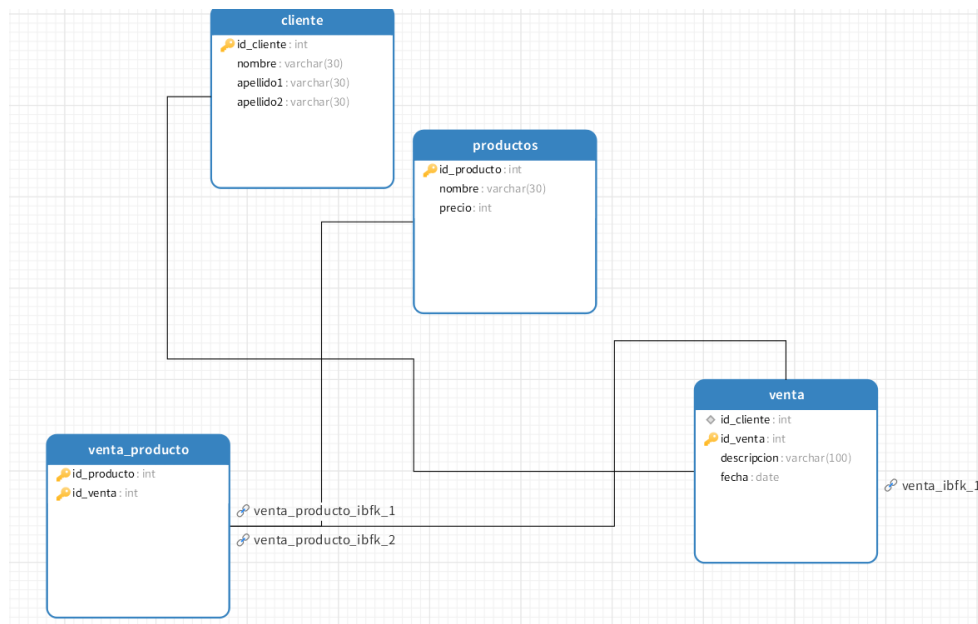
Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
mysql80-localhost tienda Run Stop Explain
1 CREATE TABLE cliente (
2   id_cliente INT PRIMARY KEY AUTO_INCREMENT,
3   nombre VARCHAR (30) NOT NULL,
4   apellido1 VARCHAR (30) NOT NULL,
5   apellido2 VARCHAR (30) NOT NULL
6 );
7
8 CREATE TABLE venta (
9   id_cliente INT,
10  id_venta INT PRIMARY KEY AUTO_INCREMENT,
11  descripcion VARCHAR (100),
12  fecha DATE NOT NULL,
13  FOREIGN KEY ( id_cliente) REFERENCES cliente(id_cliente) ON UPDATE CASCADE ON DELETE RESTRICT
14 );
15
16 CREATE TABLE productos(
17   id_producto INT PRIMARY KEY AUTO_INCREMENT,
18   nombre VARCHAR (30) NOT NULL,
19   precio INT NOT NULL CHECK (precio>0)
20 );
21
22
23 CREATE TABLE venta_producto(
24   id_producto INT,
25   id_venta INT,
26   PRIMARY KEY (id_producto, id_venta),
27   FOREIGN KEY ( id_producto) REFERENCES productos(id_producto) ON UPDATE CASCADE ON DELETE RESTRICT,
28   FOREIGN KEY ( id_venta) REFERENCES venta(id_venta) ON UPDATE CASCADE ON DELETE RESTRICT
29 );
```

Uso de distintos tipos de restricciones:

- **PRIMARY KEY:** sirve para definir la clave primaria de la tabla, lo que significa que este campo debe ser único para cada registro y no puede contener valores nulos.
- **AUTO_INCREMENT:** Hace que el valor definido se incremente automáticamente con cada nuevo registro, asegurando que cada alumno tenga un identificador único.
- **VARCHAR(30):** Define un campo de texto variable de hasta 30 caracteres.
- **NOT NULL:** Indica que un campo no puede contener valores nulos, es decir, siempre debe tener un valor.
- **DATE:** Indica que este campo almacenará una fecha
- **CHECK (precio > 0):** Esta restricción asegura que el valor del producto sea siempre mayor que 0.
- **PRIMARY KEY (id_producto, id_venta):** Define una clave primaria compuesta formada por los campos id_producto e id_venta.
- **FOREIGN KEY:** Establece una relación entre la tabla producto y la tabla venta.

Tablas Resultantes:



Problemática 3:

Una empresa necesita gestionar los proyectos y los empleados que trabajan en ellos. Cada empleado puede trabajar en varios proyectos y cada proyecto puede tener varios empleados.

Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```

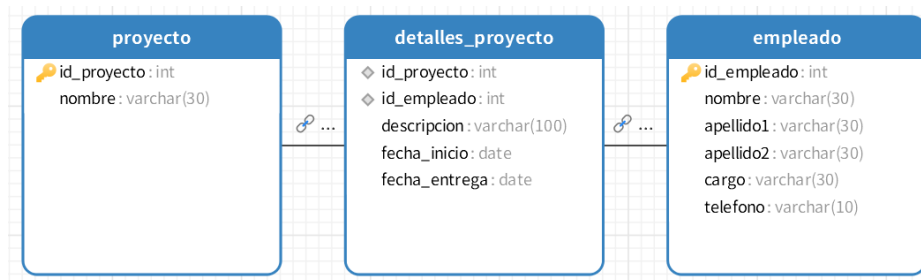
mysql80-localhost empresa Run Stop Explain
1 CREATE TABLE empleado(
2   id_empleado INT PRIMARY KEY AUTO_INCREMENT,
3   nombre VARCHAR (30) NOT NULL,
4   apellido1 VARCHAR (30) NOT NULL,
5   apellido2 VARCHAR (30) NOT NULL,
6   cargo VARCHAR (30) NOT NULL,
7   telefono VARCHAR (10)
8 );
9
10 CREATE TABLE proyecto(
11   id_proyecto INT PRIMARY KEY AUTO_INCREMENT,
12   nombre VARCHAR (30) NOT NULL
13 );
14
15 CREATE TABLE detalles_proyecto(
16   id_proyecto INT,
17   id_empleado INT,
18   descripcion VARCHAR (100),
19   fecha_inicio DATE NOT NULL,
20   fecha_entrega DATE NOT NULL,
21   FOREIGN KEY ( id_empleado ) REFERENCES empleado(id_empleado) ON UPDATE CASCADE ON DELETE RESTRICT,
22   FOREIGN KEY ( id_proyecto ) REFERENCES proyecto(id_proyecto) ON UPDATE CASCADE ON DELETE RESTRICT
23 );
24

```

Uso de distintos tipos de restricciones:

- **PRIMARY KEY:** sirve para definir la clave primaria de la tabla, lo que significa que este campo debe ser único para cada registro y no puede contener valores nulos.
- **AUTO_INCREMENT:** Hace que el valor definido se incremente automáticamente con cada nuevo registro, asegurando que cada alumno tenga un identificador único.
- **VARCHAR(30):** Define un campo de texto variable de hasta 30 caracteres.
- **NOT NULL:** Indica que un campo no puede contener valores nulos, es decir, siempre debe tener un valor.
- **DATE:** Indica que este campo almacenará una fecha
- **FOREIGN KEY:** Establece una relación entre la tabla empleado y la tabla proyecto.

Tablas Resultantes:



Problemática 4: Un hospital necesita gestionar a sus pacientes, médicos y las consultas que se realizan. Cada paciente puede tener múltiples consultas, y cada consulta es atendida por un médico.

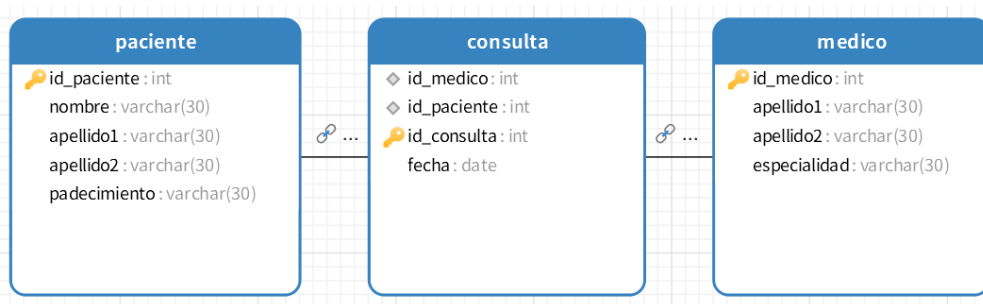
Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
mysql80-localhost hospital Run Stop Explain
1 CREATE TABLE paciente(
2   id_paciente INT PRIMARY KEY AUTO_INCREMENT,
3   nombre VARCHAR (30) NOT NULL,
4   apellido1 VARCHAR (30) NOT NULL,
5   apellido2 VARCHAR (30) NOT NULL,
6   padecimiento VARCHAR (30) NOT NULL
7 );
8
9 CREATE TABLE consulta (
10  id_medico INT,
11  id_paciente INT,
12  id_consulta INT PRIMARY KEY AUTO_INCREMENT,
13  fecha DATE NOT NULL,
14  FOREIGN KEY ( id_paciente) REFERENCES paciente(id_paciente) ON UPDATE CASCADE ON DELETE RESTRICT,
15  FOREIGN KEY ( id_medico) REFERENCES medico(id_medico) ON UPDATE CASCADE ON DELETE RESTRICT
16 );
17
18
19
20 CREATE TABLE medico (
21  id_medico INT PRIMARY KEY AUTO_INCREMENT,
22  apellido1 VARCHAR (30) NOT NULL,
23  apellido2 VARCHAR (30) NOT NULL,
24  especialidad VARCHAR (30) NOT NULL
25 );
```

Uso de distintos tipos de restricciones:

- **PRIMARY KEY:** sirve para definir la clave primaria de la tabla, lo que significa que este campo debe ser único para cada registro y no puede contener valores nulos.
- **AUTO_INCREMENT:** Hace que el valor definido se incremente automáticamente con cada nuevo registro, asegurando que cada alumno tenga un identificador único.
- **VARCHAR(30):** Define un campo de texto variable de hasta 30 caracteres.
- **NOT NULL:** Indica que un campo no puede contener valores nulos, es decir, siempre debe tener un valor.
- **DATE:** Indica que este campo almacenará una fecha
- **FOREIGN KEY:** Establece una relación entre la tabla médico, la tabla paciente y la tabla consulta.

Tablas Resultantes:



V. Conclusiones:

La creación de bases de datos es un proceso fundamental en el desarrollo de aplicaciones y sistemas de información, ya que permite la organización, almacenamiento y manejo eficiente de grandes volúmenes de datos. Durante este proceso, es crucial considerar el diseño adecuado de las tablas y las relaciones entre ellas para garantizar que la base de datos sea tanto funcional como robusta.

Las restricciones, como claves primarias, claves foráneas, y otras como NOT NULL, CHECK, y ON UPDATE CASCADE, juegan un papel esencial en la integridad de los datos. Estas restricciones no solo aseguran que los datos sean consistentes y válidos, sino que también protegen contra posibles errores que podrían surgir durante las operaciones de inserción, actualización o eliminación.