

1. Base de datos para una tienda en línea

✦ Incluye productos, clientes, pedidos y pagos.

```
CREATE DATABASE tienda_online;
USE tienda_online;

CREATE TABLE clientes (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  telefono VARCHAR(20),
  direccion TEXT,
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE productos (
  id_producto INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(150) NOT NULL,
  descripcion TEXT,
  precio DECIMAL(10,2) NOT NULL,
  stock INT NOT NULL,
  categoria VARCHAR(50),
  fecha_agregado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE pedidos (
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT,
  fecha_pedido TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  total DECIMAL(10,2) NOT NULL,
  estado ENUM('Pendiente', 'Enviado', 'Entregado', 'Cancelado') DEFAULT
  'Pendiente',
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
);

CREATE TABLE detalles_pedido (
  id_detalle INT AUTO_INCREMENT PRIMARY KEY,
  id_pedido INT,
  id_producto INT,
  cantidad INT NOT NULL,
  precio_unitario DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido),
  FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
);
```

2. Base de datos para un sistema de gestión escolar

✦ Incluye alumnos, profesores, cursos y calificaciones.

```
CREATE DATABASE gestion_escolar;
USE gestion_escolar;

CREATE TABLE alumnos (
  id_alumno INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  fecha_nacimiento DATE,
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE profesores (
  id_profesor INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  especialidad VARCHAR(100)
);

CREATE TABLE cursos (
  id_curso INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(150) NOT NULL,
  descripcion TEXT,
  id_profesor INT,
  FOREIGN KEY (id_profesor) REFERENCES profesores(id_profesor)
);

CREATE TABLE matriculas (
  id_matricula INT AUTO_INCREMENT PRIMARY KEY,
  id_alumno INT,
  id_curso INT,
  fecha_matricula TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (id_alumno) REFERENCES alumnos(id_alumno),
  FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)
);

CREATE TABLE calificaciones (
  id_calificacion INT AUTO_INCREMENT PRIMARY KEY,
  id_matricula INT,
  nota DECIMAL(5,2),
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (id_matricula) REFERENCES matriculas(id_matricula)
);
```

3. Base de datos para un sistema de reservas de hotel

✦ Incluye clientes, habitaciones, reservas y pagos.

```
CREATE DATABASE hotel_reservas;
USE hotel_reservas;

CREATE TABLE clientes (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  telefono VARCHAR(20),
  direccion TEXT
);

CREATE TABLE habitaciones (
  id_habitacion INT AUTO_INCREMENT PRIMARY KEY,
  tipo VARCHAR(50) NOT NULL,
  descripcion TEXT,
  precio DECIMAL(10,2) NOT NULL,
  estado ENUM('Disponible', 'Ocupada', 'Mantenimiento') DEFAULT
'Disponible'
);

CREATE TABLE reservas (
  id_reserva INT AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT,
  id_habitacion INT,
  fecha_entrada DATE NOT NULL,
  fecha_salida DATE NOT NULL,
  estado ENUM('Pendiente', 'Confirmada', 'Cancelada') DEFAULT
'Pendiente',
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),
  FOREIGN KEY (id_habitacion) REFERENCES habitaciones(id_habitacion)
);

CREATE TABLE pagos (
  id_pago INT AUTO_INCREMENT PRIMARY KEY,
  id_reserva INT,
  monto DECIMAL(10,2) NOT NULL,
  fecha_pago TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  metodo_pago ENUM('Tarjeta', 'Efectivo', 'Transferencia') NOT NULL,
  FOREIGN KEY (id_reserva) REFERENCES reservas(id_reserva)
);
```

4. Base de datos para una red social

✦ Incluye usuarios, publicaciones, comentarios y likes.

```
CREATE DATABASE red_social;
USE red_social;

CREATE TABLE usuarios (
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  contraseña VARCHAR(255) NOT NULL,
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE publicaciones (
  id_publicacion INT AUTO_INCREMENT PRIMARY KEY,
  id_usuario INT,
  contenido TEXT NOT NULL,
  fecha_publicacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);

CREATE TABLE comentarios (
  id_comentario INT AUTO_INCREMENT PRIMARY KEY,
  id_publicacion INT,
  id_usuario INT,
  contenido TEXT NOT NULL,
  fecha_comentario TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (id_publicacion) REFERENCES
publicaciones(id_publicacion),
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);

CREATE TABLE likes (
  id_like INT AUTO_INCREMENT PRIMARY KEY,
  id_publicacion INT,
  id_usuario INT,
  fecha_like TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (id_publicacion) REFERENCES
publicaciones(id_publicacion),
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
);
```

5. Base de datos para un sistema de gestión de empleados

✦ Incluye empleados, departamentos, salarios y asistencia.

```
CREATE DATABASE gestion_empleados;
USE gestion_empleados;

CREATE TABLE departamentos (
  id_departamento INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);

CREATE TABLE empleados (
  id_empleado INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  telefono VARCHAR(20),
  fecha_contratacion DATE NOT NULL,
  id_departamento INT,
  FOREIGN KEY (id_departamento) REFERENCES
departamentos(id_departamento)
);

CREATE TABLE salarios (
  id_salario INT AUTO_INCREMENT PRIMARY KEY,
  id_empleado INT,
  salario DECIMAL(10,2) NOT NULL,
  fecha_pago DATE NOT NULL,
  FOREIGN KEY (id_empleado) REFERENCES empleados(id_empleado)
);

CREATE TABLE asistencia (
  id_asistencia INT AUTO_INCREMENT PRIMARY KEY,
  id_empleado INT,
  fecha DATE NOT NULL,
  estado ENUM('Presente', 'Ausente', 'Permiso') NOT NULL,
  FOREIGN KEY (id_empleado) REFERENCES empleados(id_empleado)
);
```
