

# INSTITUTO TECNOLOGICO DE MEXICALI



## Base De Datos

Tarea 2 BD

ALUMNOS: FRANCISCO RAMOS VAZQUEZ

MAESTRA: JOSE RAMON BOGARIN VALENZUELA

CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES

SEMESTRE: 4

Un hospital necesita gestionar información de pacientes, médicos y citas médicas.

Identificar entidades clave: Paciente, Médico, Cita, Tratamiento.

Diseñar el modelo E-R con sus relaciones y atributos principales.

Transformar el modelo en un esquema relacional con claves primarias y foráneas.

Implementar la base de datos en SQL mediante sentencias LDD.

```
m paciente
      ⊞ medico
□ Apellido
                                                             Apellido VARCHAR(50),
                           ☐ FechaNaci
☐ Especialidad
  ID_Medico:ID
                        ID_Paciente:ID
                                                            ID INT AUTO_INCREMENT PRIMARY KEY.
            ☐ Fecha
         □ ID_Paciente
□ ID_Medico
                                                        CREATE TABLE Cita (
         ID CITA:ID
        m tratamiento
                                                             ID_Paciente INT,
                                                             ID_Medico INT,
                                                             FOREIGN KEY (ID_Medico) REFERENCES Medico(ID) ON DELETE CASCADE
```

```
CREATE TABLE Tratamiento (
ID INT AUTO_INCREMENT PRIMARY KEY,

Descripción TEXT,

Duración INT,

ID_Cita INT,

FOREIGN KEY (ID_Cita) REFERENCES Cita(ID) ON DELETE CASCADE
);
```

```
SELECT Cita.ID, Cita.Fecha, Cita.Hora, Medico.Nombre AS Medico, Medico.Especialidad FROM Cita

JOIN Paciente 1..n<->1: ON Cita.ID_Paciente = Paciente.ID

JOIN Medico 1..n<->1: ON Cita.ID_Medico = Medico.ID

WHERE Paciente.Nombre = 'Juan' AND Paciente.Apellido = 'Pérez';
-- Consultar las citas de un paciente específico, que este caso es Juan Perez
```

#### 2. Tienda en Línea

Una empresa quiere mejorar la administración de sus pedidos en línea.

Definir entidades: Cliente, Producto, Pedido, Detalle Pedido.

Crear el diagrama E-R que refleje las relaciones entre las entidades.

Convertir el modelo en un esquema de tablas relacionales.

Implementar la base de datos en SQL con restricciones de integridad.

Consultar los productos comprados por un cliente específico usando SQL.

```
CREATE TABLE Pedido (
    ID_Pedido INT AUTO_INCREMENT PRIMARY KEY,
    ID_Cliente INT,
    FechaPedido DATE,
    Estado VARCHAR(50),
    FOREIGN KEY (ID_Cliente) REFERENCES Cliente(ID_Cliente)
);

CREATE TABLE DetallePedido (
    ID_Detalle INT AUTO_INCREMENT PRIMARY KEY,
    ID_Pedido INT,
    ID_Producto INT,
    Cantidad INT,
    Precio DECIMAL(10,2),
    FOREIGN KEY (ID_Pedido) REFERENCES Pedido(ID_Pedido),
    FOREIGN KEY (ID_Producto) REFERENCES Producto(ID_Producto)
```

```
-- Consultar los productos comprados por un cliente específico usando SQL

SELECT

Producto.Nombre AS Producto,

DetallePedido.Cantidad,

DetallePedido.Precio,

(DetallePedido.Cantidad * DetallePedido.Precio) AS Total

FROM Cliente

JOIN Pedido 1<->1..n: ON Cliente.ID_Cliente = Pedido.ID_Cliente

JOIN DetallePedido 1<->1..n: ON Pedido.ID_Pedido = DetallePedido.ID_Pedido

JOIN Producto 1..n<->1: ON DetallePedido.ID_Producto = Producto.ID_Producto

WHERE Cliente.Nombre = 'Juan' AND Cliente.Apellido = 'Pérez';
```

## 3. Biblioteca Digital

Se requiere un sistema para administrar préstamos de libros en una biblioteca digital.

Identificar entidades clave: Usuario, Libro, Préstamo.

Diseñar el modelo E-R que represente los préstamos y relaciones.

Transformar el modelo en un conjunto de tablas relacionales.

Implementar la base de datos en un DBMS.

Realizar consultas SQL para obtener los préstamos activos de un usuario.

```
CREATE TABLE Usuario (
                                   ⊞ libro
                                                                                                                                                              ⊞ usuario

    ∏ Titulo

                                                                                                                                        ■ Nombre varchar(100)

    □ Autor

                                                                                                                                        Correo varchar(100)
                                                                                                                                                                                                                                                                      Fecha_Registro DATE

    □ Fecha_Publicacion

                                                                                                                                         □ Fecha_Registro date

□ ID_Libro
□ ID_L
                                                                                                                                          📭 ID_Usuario
                                                                                                                                                                                                                                              CREATE TABLE Libro (
                                                                                                                                                                                                                                                                      Fecha_Publicacion DATE
                                                                                                                              ID Usuario
                                              m prestamo
                                                                                                                                                                                                                                               CREATE TABLE Prestamo (
                                                                                                                                                                                                                                                                      ID_Prestamo INT AUTO_INCREMENT PRIMARY KEY,
                         ID_Usuario
                                                                                                                                                                                                                                                                      ID_Usuario INT,
                       ID_Libro

    □ Fecha_Prestamo

                                                                                                                                                                                                                                                                       Fecha_Prestamo DATE,
                                                                                                                                                                                                                                                                       Fecha_Devolucion DATE NULL,

    □ Fecha_Devolucion

                        □ ID_Prestamo
                                                                                                                                                                                                                                                                       FOREIGN KEY (ID_Libro) REFERENCES Libro(ID_Libro)
```

```
Usuario.Nombre AS Usuario,
Libro.Titulo AS Libro,
Prestamo.Fecha_Prestamo

FROM Prestamo

JOIN Usuario 1..n<->1: ON Prestamo.ID_Usuario = Usuario.ID_Usuario

JOIN Libro 1..n<->1: ON Prestamo.ID_Libro = Libro.ID_Libro

WHERE Prestamo.Fecha_Devolucion IS NULL

AND Usuario.Nombre = 'Juan Pérez';
```

### 4. Sistema de Recursos Humanos

Una empresa necesita gestionar sus empleados y departamentos.

Definir entidades: Empleado, Departamento, Empresa.

Elaborar el diagrama E-R que represente la estructura organizacional.

Transformar el modelo en un esquema relacional.

Implementar la base de datos en SQL, asegurando la integridad referencial.

Consultar empleados por departamento mediante sentencias SQL.

```
m empresa
Nombre varchar(100)
📭 ID_Empresa
 ID Empresa
 m departamento
                           CREATE TABLE Empresa (
Nombre varchar(100)
                              ID_Empresa INT AUTO_INCREMENT PRIMARY KEY,
ID_Empresa

∏ ID_Departamento

                           CREATE TABLE Departamento (
Departamento
                              FOREIGN KEY (ID_Empresa) REFERENCES Empresa(ID_Empresa) ON DELETE CASCADE
    m empleado
                           CREATE TABLE Empleado (
Nombre varchar(100)
Apellido varchar(100)
□ Puesto
                              ID_Departamento INT,
ID_Departamento
                              FOREIGN KEY (ID_Departamento) REFERENCES Departamento(ID_Departamento) ON DELETE SET NULL
📭 ID_Empleado
```

```
SELECT

Empleado.Nombre,

Empleado.Apellido,

Empleado.Puesto,

Departamento.Nombre AS Departamento

FROM Empleado

JOIN Departamento 1..n<->1: ON Empleado.ID_Departamento = Departamento.ID_Departamento

WHERE Departamento.Nombre = 'Desarrollo';
```

Solo cambias el nombre del departamento si quieres ver otro

### 5. Plataforma de Cursos en Línea

Se requiere un sistema para gestionar la inscripción de usuarios en cursos en línea.

Identificar entidades clave: Usuario, Curso, Inscripción.

Diseñar un modelo E-R para representar la relación entre usuarios y cursos.

Convertir el modelo en un conjunto de tablas con relaciones adecuadas.

Implementar la base de datos en SQL con restricciones adecuadas.

Consultar los cursos inscritos por un usuario en la base de datos.

```
ID_Usuario INT AUTO_INCREMENT PRIMARY KEY,

    □ curso
    □

                              ■ usuario
                                                 Nombre VARCHAR(100) NOT NULL,
□ Nombre
                           Nombre varchar(100)
                                                 Correo VARCHAR(100) UNIQUE NOT NULL
■ Descripcion
                           ☐ Apellido varchar(100) ☐ );
Duracion /* Duración en horas */ in
                           Correo
                                             CREATE TABLE Curso (
📭 ID_Curso
                           🖳 ID_Usuario
                                                 ID_Curso INT AUTO_INCREMENT PRIMARY KEY,
                        ID Usuario
     ID Curso
                                             CREATE TABLE Inscripcion (
                                                 ID_Inscripcion INT AUTO_INCREMENT PRIMARY KEY,
          ■ inscripcion
      ID_Usuario
                                                 ID_Curso INT NOT NULL,
                                                 Fecha_Inscripcion DATE DEFAULT (CURDATE()),
                                                 FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario) ON DELETE CASCADE,

□ Fecha_Inscripcion

                                                 FOREIGN KEY (ID_Curso) REFERENCES Curso(ID_Curso) ON DELETE CASCADE
       □ ID_Inscripcion
```

```
Usuario.Nombre AS Usuario,
Curso.Nombre AS Curso,

☐ Inscripcion.Fecha_Inscripcion

FROM Inscripcion

JOIN Usuario 1..n<->1: ON Inscripcion.ID_Usuario = Usuario.ID_Usuario

JOIN Curso 1..n<->1: ON Inscripcion.ID_Curso = Curso.ID_Curso

WHERE Usuario.Nombre = 'Juan' AND Usuario.Apellido = 'Martínez';
```