

Base de datos del Hospital

Introducción:

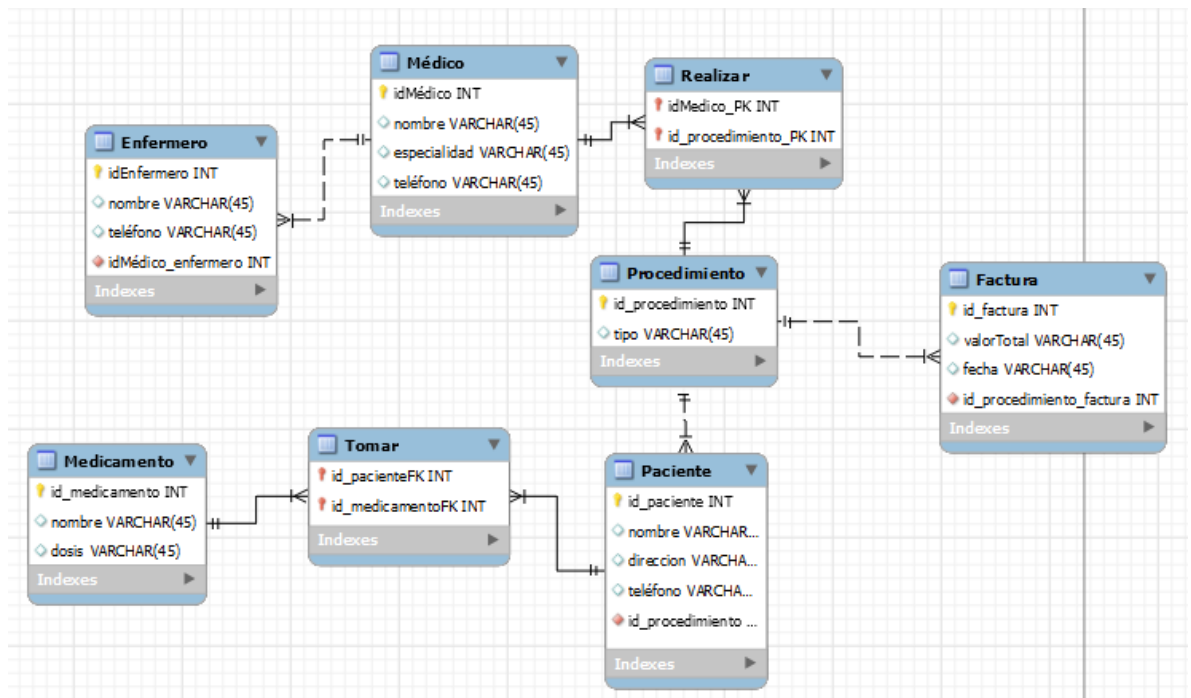
En esta actividad, se trabajará con una base de datos creada a partir del Modelo Relacional (MR) de un hospital, diseñado por los compañeros de clase. El objetivo principal es convertir el MR en una base de datos MySQL, completar la información de las tablas y realizar consultas y vistas para obtener información relevante del sistema.

Desarrollo:

Creación de la base de datos:

a. Creación de la estructura de la base de datos utilizando el diagrama EER.

- Diagrama en Workbench



b. Creación de las tablas utilizando sentencias SQL.

- Consulta sql

```
CREATE SCHEMA IF NOT EXISTS `consultorio_doctor` DEFAULT CHARACTER SET utf8;

USE `consultorio_doctor`;

CREATE TABLE IF NOT EXISTS `Médico` (
  `idMédico` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `especialidad` VARCHAR(45) NULL,
  `teléfono` VARCHAR(45) NULL,
  PRIMARY KEY (`idMédico`)
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Enfermero` (
  `idEnfermero` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `teléfono` VARCHAR(45) NULL,
  `idMédico_enfermero` INT NOT NULL,
  PRIMARY KEY (`idEnfermero`),
  INDEX `idMedico_idx` (`idMédico_enfermero` ASC) VISIBLE,
  CONSTRAINT `FKidMedico` FOREIGN KEY (`idMédico_enfermero`) REFERENCES `Médico` (`idMédico`)
  ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Procedimiento` (
  `id_procedimiento` INT NOT NULL,
  `tipo` VARCHAR(45) NULL,
  PRIMARY KEY (`id_procedimiento`)
) ENGINE=InnoDB;
```

```

CREATE TABLE IF NOT EXISTS `Paciente` (
  `id_paciente` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `direccion` VARCHAR(45) NULL,
  `teléfono` VARCHAR(45) NULL,
  `id_procedimiento` INT NOT NULL,
  PRIMARY KEY (`id_paciente`),
  INDEX `id_procedimiento_idx` (`id_procedimiento` ASC) VISIBLE,
  CONSTRAINT `PKid_procedimiento` FOREIGN KEY (`id_procedimiento`)
REFERENCES `Procedimiento` (`id_procedimiento`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB;

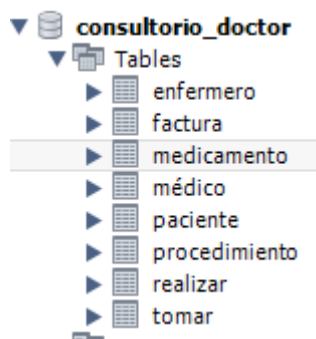
CREATE TABLE IF NOT EXISTS `Medicamento` (
  `id_medicamento` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `dosis` VARCHAR(45) NULL,
  PRIMARY KEY (`id_medicamento`)
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Factura` (
  `id_factura` INT NOT NULL,
  `valorTotal` VARCHAR(45) NULL,
  `fecha` VARCHAR(45) NULL,
  `id_procedimiento_factura` INT NOT NULL,
  PRIMARY KEY (`id_factura`),
  INDEX `id_procedimiento_idx` (`id_procedimiento_factura` ASC) VISIBLE,
  CONSTRAINT `pkid_procedimiento_factura` FOREIGN KEY (`id_procedimiento_factura`)
REFERENCES `Procedimiento` (`id_procedimiento`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `Realizar` (
  `idMedico_PK` INT NOT NULL,
  `id_procedimiento_PK` INT NOT NULL,
  PRIMARY KEY (`idMedico_PK`, `id_procedimiento_PK`),
  INDEX `id_procedimiento_idx` (`id_procedimiento_PK` ASC) VISIBLE,
  CONSTRAINT `idMedico_PK` FOREIGN KEY (`idMedico_PK`) REFERENCES `Médico` (`idMédico`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `id_procedimiento_PK` FOREIGN KEY (`id_procedimiento_PK`)
REFERENCES `Procedimiento` (`id_procedimiento`) ON DELETE NO ACTION ON UPDATE NO ACTION
)

```

- Resultado



Completar la información de las tablas:

a. Inserción de al menos 5 registros en cada una de las tablas.

1. Tabla Enfermero

- Consulta

```
INSERT INTO `consultorio_doctor`.`Enfermero` (`idEnfermero`, `nombre`, `teléfono`, `idMédico_enfermero`)
VALUES
(1, 'Enf. Juan Pérez', '555-555-5555', 1),
(2, 'Enf. Maria Sánchez', '555-555-5556', 2),
(3, 'Enf. José García', '555-555-5557', 3),
(4, 'Enf. Ana Rodriguez', '555-555-5558', 4),
(5, 'Enf. Pedro Martínez', '555-555-5559', 5);
```

- Resultado

idEnfermero	nombre	teléfono	idMédico_enfermero
1	Enf. Juan Pérez	555-555-5555	1
2	Enf. Maria Sánchez	555-555-5556	2
3	Enf. José García	555-555-5557	3
4	Enf. Ana Rodriguez	555-555-5558	4
5	Enf. Pedro Martínez	555-555-5559	5

2. Tabla Medico

- Consulta

```
INSERT INTO `consultorio_doctor`.`Médico` (`idMédico`, `nombre`, `especialidad`, `teléfono`)
VALUES
(1, 'Dr. Juan Pérez', 'Pediatría', '555-555-5555'),
(2, 'Dra. Maria Sánchez', 'Cardiología', '555-555-5556'),
(3, 'Dr. José García', 'Dermatología', '555-555-5557'),
(4, 'Dra. Ana Rodriguez', 'Oftalmología', '555-555-5558'),
(5, 'Dr. Pedro Martínez', 'Ortopedia', '555-555-5559');
```

- Resultado

idMédico	nombre	especialidad	teléfono
1	Dr. Juan Pérez	Pediatría	555-555-5555
2	Dra. Maria Sánchez	Cardiología	555-555-5556
3	Dr. José García	Dermatología	555-555-5557
4	Dra. Ana Rodriguez	Oftalmología	555-555-5558
5	Dr. Pedro Martínez	Ortopedia	555-555-5559

3. Tabla Procedimiento

- Consulta

```
INSERT INTO `consultorio_doctor`.`Procedimiento` (`id_procedimiento`, `tipo`)
VALUES (1, 'Tipo 1'), (2, 'Tipo 2'), (3, 'Tipo 3'), (4, 'Tipo 4'), (5, 'Tipo 5');
```

- Resultado

id_procedimiento	tipo
1	Tipo 1
2	Tipo 2
3	Tipo 3
4	Tipo 4
5	Tipo 5

4. Enfermero Paciente

- Consulta

```
INSERT INTO consultorio_doctor.Paciente (id_paciente, nombre, direccion, teléfono, id_procedimiento)
VALUES
(1, 'Juan', 'Calle 1', '555-555-5555', 1),
(2, 'Maria', 'Calle 2', '555-555-5556', 1),
(3, 'Pedro', 'Calle 3', '555-555-5557', 2),
(4, 'Ana', 'Calle 4', '555-555-5558', 2),
(5, 'Jorge', 'Calle 5', '555-555-5559', 3);
```

- Resultado

id_paciente	nombre	direccion	teléfono	id_procedimiento
1	Juan	Calle 1	555-555-5555	1
2	Maria	Calle 2	555-555-5556	1
3	Pedro	Calle 3	555-555-5557	2
4	Ana	Calle 4	555-555-5558	2
5	Jorge	Calle 5	555-555-5559	3

5. Tabla Factura

- Consulta

```
INSERT INTO Factura (id_factura, valorTotal, fecha, id_procedimiento_factura)
VALUES (1, '100', '2023-02-14', 1),
(2, '50', '2023-02-14', 2),
(3, '75', '2023-02-14', 1),
(4, '200', '2023-02-14', 3),
(5, '80', '2023-02-14', 2);
```

- Resultado

id_factura	valorTotal	fecha	id_procedimiento_factura
1	100	2023-02-14	1
2	50	2023-02-14	2
3	75	2023-02-14	1
4	200	2023-02-14	3
5	80	2023-02-14	2

6. Tabla Medicamento

- Consulta

```
INSERT INTO Medicamento (id_medimento, nombre, dosis)
VALUES (1, 'Paracetamol', '500mg'),
(2, 'Ibuprofeno', '400mg'),
(3, 'Omeprazol', '20mg'),
(4, 'Amoxicilina', '500mg'),
(5, 'Loratadina', '10mg');
```

- Resultado

id_medimento	nombre	dosis
1	Paracetamol	500mg
2	Ibuprofeno	400mg
3	Omeprazol	20mg
4	Amoxicilina	500mg
5	Loratadina	10mg

7. Tabla Realizar

- Consulta

```
INSERT INTO Realizar (idMedico_PK, id_procedimiento_PK)
VALUES (1, 1),
(2, 2),
(3, 1),
(1, 2),
(2, 3);
```

- Resultado

idMedico_PK	id_procedimiento_PK
1	1
3	1
1	2
2	2
2	3

8. Tabla Tomar

- Consulta

```
INSERT INTO Tomar (id_pacienteFK, id_medimentoFK)
VALUES (1, 1),
(2, 2),
(3, 1),
(1, 2),
(2, 3);
```

- Resultado

id_pacienteFK	id_medimentoFK
1	1
3	1
1	2
2	2
2	3

Consultas:

- Consulta para conocer los medicamentos tomados por cada paciente y la dosis suministrada.

- Consulta

```
SELECT p.nombre, m.nombre AS medicamento, dosis
FROM Paciente p
JOIN Tomar t ON p.id_paciente = t.id_pacienteFK
JOIN Medicamento m ON t.id_medimentoFK = m.id_medimento
```

- Resultado

nombre	medicamento	dosis
Juan	Paracetamol	500mg
Juan	Ibuprofeno	Paracetamo
Maria	Ibuprofeno	400mg
Maria	Omeprazol	20mg
Pedro	Paracetamol	500mg

- b. Consulta para conocer los enfermeros que participaron en los procedimientos de los pacientes.

- Consulta

```
select hospitalqa.enfermero.nombre, procedimiento.tipo
from hospitalqa.enfermero
join hospitalqa.medico_procedimiento
on medico_procedimiento.id_medico_ = enfermero.id_enfermero
join hospitalqa.procedimiento
on procedimiento.id_procedimiento = medico_procedimiento.id_procedimiento_;
```

- Resultado

nombre	tipo
Enf. Juan Pérez	Tipo 1
Enf. José García	Tipo 1
Enf. Juan Pérez	Tipo 2
Enf. Maria Sánchez	Tipo 2
Enf. Maria Sánchez	Tipo 3

Vistas:

1. Vista de pacientes con detalles de procedimientos:

- Esta vista permite a los usuarios del sistema obtener una lista de todos los pacientes y los procedimientos que han recibido.

- **Consulta**

```
CREATE VIEW `vista_pacientes_procedimientos` AS
SELECT p.id_paciente, p.nombre AS paciente_nombre, p.direccion, p.teléfono, pr.tipo AS tipo_procedimiento
FROM `Paciente` p
JOIN `Procedimiento` pr ON p.id_procedimiento = pr.id_procedimiento;
```

- **Resultado**

id_paciente	paciente_nombre	direccion	teléfono	tipo_procedimiento
1	Juan	Calle 1	555-555-5555	Tipo 1
2	Maria	Calle 2	555-555-5556	Tipo 1
3	Pedro	Calle 3	555-555-5557	Tipo 2
4	Ana	Calle 4	555-555-5558	Tipo 2
5	Jorge	Calle 5	555-555-5559	Tipo 3

2. Vista de medicamentos recetados a pacientes:

- Esta vista muestra una lista de medicamentos que han sido recetados a cada paciente

- **Consulta**

```
CREATE VIEW `vista_medicamentos_recetados` AS
SELECT p.nombre AS paciente_nombre, m.nombre AS medicamento, m.dosis
FROM `Paciente` p
JOIN `Tomar` t ON p.id_paciente = t.id_pacienteFK
JOIN `Medicamento` m ON t.id_medimentoFK = m.id_medimento;
```

- **Resultado**

paciente_nombre	medicamento	dosis
Juan	Paracetamol	500mg
Juan	Ibuprofeno	400mg
Maria	Ibuprofeno	400mg
Maria	Omeprazol	20mg
Pedro	Paracetamol	500mg

3. Vista de facturas de procedimientos:

- Esta vista permite a los usuarios obtener una lista de facturas y los procedimientos que se realizaron para cada factura.

- Consulta

```
CREATE VIEW `vista_facturas_procedimientos` AS  
SELECT f.id_factura, f.fecha, f.valorTotal, pr.tipo AS tipo_procedimiento  
FROM `Factura` f  
JOIN `Procedimiento` pr ON f.id_procedimiento_factura = pr.id_procedimiento;
```

- Resultado

id_factura	fecha	valorTotal	tipo_procedimiento
1	2023-02-14	100	Tipo 1
3	2023-02-14	75	Tipo 1
2	2023-02-14	50	Tipo 2
5	2023-02-14	80	Tipo 2
4	2023-02-14	200	Tipo 3

¿Qué le agregaría al modelo para dar mas información y esa información cuál sería?

Se le podría agregar las siguientes tablas para tener más información:

Tabla de Citas: Esta tabla podría incluir información sobre las citas programadas para cada paciente, como la fecha y hora de la cita, el médico que atenderá la cita y el motivo de la visita.

Tabla de Historiales Médicos: Esta tabla podría incluir información sobre el historial médico de cada paciente, como las condiciones médicas previas, las alergias, los medicamentos recetados y cualquier otra información relevante.

Conclusión:

En esta actividad se logró convertir el MR en una base de datos funcional en MySQL, se completó la información de las tablas y se realizaron consultas y vistas para obtener información relevante del sistema. El uso de consultas y vistas permite obtener información precisa y relevante de manera más eficiente que realizando múltiples consultas. Además, la creación de una base de datos bien estructurada y completa es fundamental para garantizar la integridad y la seguridad de los datos almacenados.

Anexos:

Archivos con el script SQL utilizado en el trabajo

1. consultas.sql
2. creación_Tablas.sql
3. registro_Tablas.sql
4. vistas.sql