

MATERIAL DE ESTUDIO EVALUACIÓN

```
Create table Empleado(  
    id_empleado int Primary Key,  
    nombre varchar(50),  
    id_departamento int,  
    FOREIGN KEY (id_departamento) references Departamento (id_departamento)  
);
```

```
Create table Departamento(  
    id_departamento int Primary Key,  
    nombre varchar(50)  
);
```

```
Create table PPlan_contingencia(  
    id_plan int PRIMARY KEY,  
    descripcion text,  
    id_departamento int,  
    FOREIGN KEY (id_departamento) references Departamento (id_departamento)  
);
```

```
Create table Evaluacion_plan(  
    id_evaluacion int PRIMARY KEY,  
    fecha date,  
    resultado varchar(20),  
    id_plan int,  
    FOREIGN KEY (id_plan) references Plan_contingencia (id_plan)  
);
```

```
-- Consulta que selecciona los planes de contingencia asociados a un departamento  
específico  
select Plan_contingencia.descripcion  
from Plan_contingencia  
join Departamento on plan_contingencia.id_departamento =  
Departamento.id_departamento  
where Departamento.nombre = 'TI';
```

EJERCICIO

--Sistema de gestión de emergencias en hospitales.

```
create table Paciente(  
    id_paciente int PRIMARY KEY,  
    nombre varchar(50),  
    edad int  
);
```

```
create table Medico(  
    id_medico int PRIMARY KEY,  
    nombre varchar(50),  
    especialidad varchar(50)  
);
```

```
create table Emergencia(  
    id_emergencia int PRIMARY KEY,  
    descripcion varchar(50),  
    fecha date,  
    id_paciente int,  
    id_medico int,  
    FOREIGN KEY (id_paciente) references Paciente (id_paciente),  
    FOREIGN KEY (id_medico) references Medico(id_medico)  
);
```

--Sentencias o consulta emergencias atendidas por un medico en particular

```
select Emergencia.fecha, Emergencia.descripcion  
from Emergencia  
join Medico on Emergencia.id_medico = Medico.id_medico  
where Medico.nombre = 'Dr. Perez';
```

--Consulta cuenta cuantas emergencias ha tenido cada paciente y selecciona a los que han tenido más de 3 emergencias

```
select Paciente.nombre, count(Emergencia.id_emergencia) as num_emergencia  
from Paciente  
join Emergencia on Paciente.id_paciente = Emergencia.id_emergencia  
group by Paciente.nombre  
having count(Emergencia.id_emergencia > 3);
```

EJERCICIO

--Sistema de gestión de emergencias en hospitales.

```
create table Paciente(  
    id_paciente int PRIMARY KEY,  
    nombre varchar(50),  
    edad int  
);
```

```
create table Medico(  
    id_medico int PRIMARY KEY,  
    nombre varchar(50),  
    especialidad varchar(50)  
);
```

```
create table Emergencia(  
    id_emergencia int PRIMARY KEY,  
    descripcion varchar(50),  
    fecha date,  
    id_paciente int,  
    id_medico int,  
    FOREIGN KEY (id_paciente) references Paciente (id_paciente),  
    FOREIGN KEY (id_medico) references Medico(id_medico)  
);
```

--Sentencias o consulta emergencias atendidas por un medico en particular

```
select Emergencia.fecha, Emergencia.descripcion  
from Emergencia  
join Medico on Emergencia.id_medico = Medico.id_medico  
where Medico.nombre = 'Dr. Perez';
```

--Consulta cuenta cuantas emergencias ha tenido cada paciente y selecciona a los que han tenido más de 3 emergencias

```
select Paciente.nombre, count(Emergencia.id_emergencia) as num_emergencia  
from Paciente  
join Emergencia on Paciente.id_paciente = Emergencia.id_emergencia  
group by Paciente.nombre  
having count(Emergencia.id_emergencia > 3);
```

EJERCICIO

```
--creando tabla
--create database ejemplo;
use ejemplo;
create table producto
(id_producto int primary key,
 nombreproducto varchar(25) not null,
 precio int null,
 descripcionproducto text null
);

insert into producto(id_producto, nombreproducto, precio, descripcionproducto)
values(1, 'Esponja', 990, 'Esponja para loza'),
      (2, 'Desinfectante', 1590, 'Mata virus y bacterias'),
      (3, 'Arroz', 2000, 'Alimento no perecible'),
      (4, 'Notmayo', 3600, 'Mayonesa artificial');

--pedir dos cosas
SELECT nombreproducto, descripcionproducto FROM Producto;
SELECT * FROM Producto WHERE descripcionproducto = 'Mayonesa Artificial';
SELECT * FROM Producto WHERE precio > 1580;
SELECT * FROM Producto WHERE nombreproducto NOT IN ('Arroz', 'Notmayo');
SELECT DISTINCT nombreproducto, precio FROM Producto WHERE precio < 2000;
SELECT * FROM Producto ORDER BY precio DESC;
SELECT COUNT(*) FROM Producto;
SELECT COUNT(*) FROM Producto WHERE nombreproducto = 'Arroz';
SELECT SUM(precio) FROM Producto;
SELECT COUNT(*), nombreproducto FROM Producto GROUP BY nombreproducto;
UPDATE Producto set precio = 1000 where id_producto = '3';
delete from Producto where id_producto = 4;
```