

# Practica 4 BD1

## Esquema final

La clave primaria está en negrita

**PATIENT** (**patient\_id** , patient\_name, patient\_address, patient\_city, primary\_phone, secondary\_phone)

**DOCTOR** (**doctor\_id**, doctor\_name, doctor\_address, doctor\_city, doctor\_speciality)

**APPOINTMENT** (**patient\_id**, **appointment\_date**, appointment\_duration, contact\_phone, observations, payment\_card)

**MEDICAL\_REVIEW** (**patient\_id**, **appointment\_date**, **doctor\_id**)

**PRESCRIBED\_MEDICATION** (**patient\_id**, **appointment\_date**, **medication\_name**)

## Ejercicio 1

1. Crea un usuario para las bases de datos usando el nombre '*appointments\_user*'. Asigne a estos todos los permisos sobre sus respectivas tablas. Habiendo creado este usuario evitaremos el uso de '*root*' para el resto del trabajo práctico.

Adicionalmente, con respecto a esta base de datos:

1. Cree un usuario sólo con permisos para realizar consultas de selección, es decir que no puedan realizar cambios en la base. Use el nombre '*appointments\_select*'.
2. Cree un usuario que pueda realizar consultas de selección, inserción, actualización y eliminación a nivel de filas, pero que no puedan modificar el esquema. Use el nombre '*appointments\_update*'.
3. Cree un usuario que tenga los permisos de los anteriores, pero que además pueda modificar el esquema de la base de datos. Use el nombre '*appointments\_schema*'.

## Resolución

### Inciso 1

```
CREATE USER 'appointments_user'@'localhost' IDENTIFIED BY 'bd1_2025';
GRANT SELECT, INSERT, UPDATE, DELETE ON appointments.* TO
'appointments_user'@'localhost';
```

### Inciso 1.1

```
CREATE USER 'appointments_select'@'localhost' IDENTIFIED BY 'bd1_2025';
```

```
GRANT SELECT ON appointments.* TO 'appointments_select'@'localhost';
```

## Inciso 1.2

```
CREATE USER 'appointments_update'@'localhost' IDENTIFIED BY 'bd1_2025';  
GRANT SELECT, INSERT, UPDATE, DELETE ON appointments.* TO  
'appointments_update'@'localhost';
```

## Inciso 1.3

```
CREATE USER 'appointments_schema'@'localhost' IDENTIFIED BY 'bd1_2025';  
GRANT ALL PRIVILEGES ON appointments.* TO  
'appointments_schema'@'localhost';
```

## Ejercicio 2

Hallar aquellos pacientes que para todas sus consultas médicas siempre hayan dejado su número de teléfono primario (nunca el teléfono secundario).

### Solución 1

```
SELECT p.patient_id, p.patient_name  
FROM patient p INNER JOIN appointment a ON p.patient_id = a.patient_id  
GROUP BY p.patient_id, p.primary_phone, p.patient_name  
HAVING COUNT(*) = SUM(a.contact_phone = p.primary_phone);
```

### Solución 2

```
SELECT p.patient_id, p.patient_name  
FROM PATIENT p  
WHERE NOT EXISTS (  
    SELECT *  
    FROM APPOINTMENT a  
    WHERE a.patient_id = p.patient_id  
        AND a.contact_phone <> p.primary_phone  
);
```

## Ejercicio 3

Crear una vista llamada 'doctors\_per\_patients' que muestre los id de los pacientes y los id de doctores de la ciudad donde vive el paciente.

### Resolución

```
CREATE VIEW appointments.doctors_per_patients AS
SELECT p.patient_id, d.doctor_id
FROM appointments.patient p, appointments.doctor d
ON p.patient_city = d.doctor_city;
```

## Ejercicio 4

Utiliza la vista generada en el ejercicio anterior para resolver las siguientes consultas:

1. Obtener la cantidad de doctores por cada paciente que tiene disponible en su ciudad
2. Obtener los nombres de los pacientes sin doctores en su ciudad
3. Obtener los doctores que comparten ciudad con más de cinco pacientes.

## Resolución

### Inciso 1

```
SELECT patient_id, COUNT(doctor_id) AS cantidad_doctores
FROM appointments.doctors_per_patients
GROUP BY patient_id;
```

### Inciso 2

```
SELECT p.patient_name
FROM patient p
LEFT JOIN appointments.doctors_per_patients dp
  ON p.patient_id = dp.patient_id
WHERE dp.doctor_id IS NULL;
```

### Inciso 3

```
SELECT doctor_id
FROM appointments.doctors_per_patients
GROUP BY doctor_id
HAVING COUNT(patient_id) > 5;
```

## Ejercicio 5

### Resolución

```
CREATE TABLE appointments_per_patient (
  idApP INT(11) NOT NULL AUTO_INCREMENT,
  id_patient INT(11),
  count_appointments INT(11),
```

```
last_update DATETIME,  
user VARCHAR(16),  
PRIMARY KEY (idApP)  
);
```