

# Practica 1 de Bases de Datos Avanzadas

## 1. Teniendo en cuenta el conjunto de dependencias funcionales (L), contestar a las siguientes preguntas

- ¿Puede atender un centro de salud a pacientes que sean de una localidad o provincia diferente a la del centro?
  - Si, si puede tenemos un paciente de Villazacillo, Albacete y la clínica que le atiende es en Metrópolis, Madrid.
- ¿Puede un mismo paciente acudir a centros diferentes?
  - Sí, si puede ya que no existe una dependencia funcional entre los centros y los pacientes.
- ¿Puede una especialidad pertenecer a dos áreas diferentes?
  - No ya que existe una dependencia funcional entre especialidad y área.
- ¿Podría haber dos médicos atendiendo consultas del mismo área en un mismo centro?
  - Si, ya que no hay dependencia funcional.

## 2. Determinar el nivel de normalización de la relación TablaUnica, calculando TODAS las claves existentes y sin eliminar la tabla previamente creada

### Cálculo de claves

Primero comprobamos si la clave se puede constituir con los atributos esenciales

$$(HF)^+ = H, F \neq T$$

Como no se cubre todos los atributos de T probamos con combinaciones de los atributos posibles

$$(HFM)^+ = H, F, M, P, O, S, C, A, D, J, K, B, I, G, E, L, N = T$$

$$(HFL)^+ = H, F, L, P, I, G, E, L, N, O, S, C, M, B, J, K, D, A = T$$

$$(HFL)^+ H, F, L, N \neq T$$

$$(HFD)^+ = H, F, D, C, J, K \neq T$$

$$(HFC)^+ = H, F, C, D, J, K \neq T$$

$$(HFJ)^+ = H, F, J, K \neq T$$

$$(HFS)^+ = H, F, S, A \neq T$$

Con lo cual, la clave principal será: ( H F M ) = ( teléfonos fecha\_hora mID )

La clave secundaria será: ( H F P ) = ( teléfonos fecha\_hora pID )

Atributos Esenciales	
H	F
I, G, E, N, K, O, A, B	P, L, C, D, J, S, M
Atributos no posibles	Atributos posibles

Como no se pueden tener atributos en la clave que sean nulos, quitamos el atributo "teléfono" de las claves

Se establecen las claves en SQL tal que

```
ALTER TABLE tablaunica ADD PRIMARY KEY (fechahora,mID); -- Para la clave primaria
ALTER TABLE tablaunica ADD UNIQUE (fechahora,pID); -- Para la clave secundaria
```

Para finalizar los atributos quedan tal que:

- **Atributos principales (P)** : (H, F, M, P)
- **Atributos no principales (Q)** : (O,S,C,A,D,J,K,B,I,G,E,L,N)

Nivel de normalización

Habiendo calculado las claves, podemos determinar el nivel de normalización.

El esquema está en 1FN ya que hay atributos no principales que dependen de un subconjunto de la clave. Como por ejemplo  $P \rightarrow I$ .

---

## Resolver en SQL las siguientes consultas

a. Obtener el número de pacientes que han sido atendidos en centros de salud de una provincia diferente a la suya en el área Quirúrgica.

El código en SQL será:

```
SELECT count(distinct pID)
FROM tablaunica as t
WHERE t.provinciaC <> t.provinciaP AND t.area = 'Quirúrgica';
```

El resultado es:

	numPacientesDesplazados
▶	1

b. Obtener el centro y su provincia para aquellos centros que hayan atendido tanto a pacientes de Sagrillas como de Peñafría a partir de las 12:00.

El código en SQL será:

```
SELECT cID, nomcentro,provinciaC, fechahora
FROM tablaunica as t
WHERE TIME(fechahora)> '12:00' AND pID IN ( SELECT distinct pID
```

```

FROM tablaunica
as t
WHERE t.localidadP = 'Sagrillas' OR
t.localidadP = 'Peñafría');

```

	cID	nomcentro	provinciaC	fechahora
▶	8	Parquesol	Segovia	2024-01-10 12:30
	8	Parquesol	Segovia	2024-01-15 12:30
	8	Parquesol	Segovia	2024-01-18 16:00
	5	Vallesol	Madrid	2024-01-18 16:00

El resultado es:

c. Obtener para cada centro de salud, el número de asistencias atendidas en viernes a pacientes de género femenino (valor "M").

La consulta en SQL será:

```

-- Solo salen los centros que recibieron visitas
SELECT nomcentro, count(*) as atenciones
FROM tablaunica as t
WHERE dayofweek(t.fechahora)='6' AND t.genero = 'M'
GROUP BY (nomcentro);

-- Para que salgan los valores nulos
SELECT C.nomcentro AS Centro_Salud, COALESCE(Total_Asistencias_Femeninas_Viernes,
0) AS Asistencias_Femeninas_Viernes
FROM (SELECT nomcentro, COUNT(*) AS Total_Asistencias_Femeninas_Viernes
FROM tablaunica as t
WHERE dayofweek(t.fechahora)='6' AND genero = 'M'
GROUP BY nomcentro) AS AsistenciasPorCentro
RIGHT JOIN (SELECT DISTINCT nomcentro FROM tablaunica) AS C ON
AsistenciasPorCentro.nomcentro = C.nomcentro;

```

Explicación: En la segunda consulta lo que se hace es coger la consulta anterior y transformarla como si fuera una tabla, seguidamente, se hace un right join con una tabla con todos los nombres de centros para que aunque haya valores null en nuestra primera consulta, se conserven y finalmente con la función COALESCE() hacemos que los valores NULL se muestre 0.

Resultados:

	Centro_Salud	Asistencias_Femeninas_Viernes
▶	Vallesol	1
	Parquesol	0
	Centro 3	0