

Ejercicio 2 – Estadías

ESTADIA<dni_cliente, cod_hotel, cantidad_habitaciones,direccion_hotel, ciudad_hotel, dni_gerente, nombre_gerente,nombre_cliente, ciudad_cliente, fecha_inicio_hospedaje,cant_dias_hospedaje, habitacion>

Paso 1: Dependencias Funcionales:

cod_hotel → cantidad_habitaciones, direccion_hotel, ciudad_hotel, dni_gerente
(Cada hotel tiene una dirección, ciudad, gerente y cantidad total de habitaciones).

dni_gerente → nombre_gerente
(Cada gerente tiene un único nombre).

dni_cliente → nombre_cliente, ciudad_cliente
(Cada cliente tiene un único nombre y ciudad).

cod_hotel, habitacion, fecha_inicio_hospedaje, dni_cliente → cant_dias_hospedaje
(Un cliente puede alojarse en varias habitaciones en el mismo hotel y fecha, pero la cantidad de días depende de esa combinación).

cod_hotel, habitacion → (identifica una habitación dentro de un hotel, ya que el número de habitación se repite entre hoteles).

Paso 2: Atributos primos y no primos

Atributos primos (forman parte de alguna clave candidata):
dni_cliente, cod_hotel, habitacion, fecha_inicio_hospedaje

Atributos no primos:
cantidad_habitaciones, direccion_hotel, ciudad_hotel, dni_gerente, nombre_gerente, nombre_cliente, ciudad_cliente, cant_dias_hospedaje

Paso 3: Claves Candidatas

Cada estadía se identifica de forma única por:
(dni_cliente, cod_hotel, habitacion, fecha_inicio_hospedaje)

Un cliente (dni_cliente) puede estar en varios hoteles o habitaciones, pero la combinación con cod_hotel, habitacion y la fecha de inicio (fecha_inicio_hospedaje) define de manera única esa estadía.

Clave candidata final:

(dni_cliente, cod_hotel, habitacion, fecha_inicio_hospedaje)

Paso 4: Normalización (1FN-2FN-3FN)

Primera Forma Normal (1FN)

Ya está en 1FN, porque todos los atributos son atómicos (no hay valores repetidos o multivaluados).

Segunda Forma Normal (2FN): Eliminamos dependencias parciales (atributos que dependen solo de parte de la clave primaria compuesta).

Dependencias parciales:

- $\text{cod_hotel} \rightarrow \text{cantidad_habitaciones}, \text{direccion_hotel}, \text{ciudad_hotel}, \text{dni_gerente}$
- $\text{dni_gerente} \rightarrow \text{nombre_gerente}$
- $\text{dni_cliente} \rightarrow \text{nombre_cliente}, \text{ciudad_cliente}$

R1 (Hotel):

HOTEL($\text{cod_hotel}, \text{cantidad_habitaciones}, \text{direccion_hotel}, \text{ciudad_hotel}, \text{dni_gerente}$)

R2 (Gerente):

GERENTE($\text{dni_gerente}, \text{nombre_gerente}$)

R3 (Cliente):

CLIENTE($\text{dni_cliente}, \text{nombre_cliente}, \text{ciudad_cliente}$)

R4 (Estadia):

ESTADIA($\text{dni_cliente}, \text{cod_hotel}, \text{habitacion}, \text{fecha_inicio_hospedaje}, \text{cant_dias_hospedaje}$)

Tercera Forma Normal (3FN)

Eliminamos dependencias **transitivas**, como:

- $\text{cod_hotel} \rightarrow \text{dni_gerente}$ y $\text{dni_gerente} \rightarrow \text{nombre_gerente}$ (ya separadas en HOTEL y GERENTE)
- $\text{dni_cliente} \rightarrow \text{nombre_cliente}, \text{ciudad_cliente}$ (ya separadas en CLIENTE)

Paso 5: Tablas finales en 3FN

HOTEL

- **cod_hotel** (PK)
- cantidad_habitaciones
- direccion_hotel
- ciudad_hotel
- dni_gerente (FK → GERENTE)

GERENTE

- **dni_gerente** (PK)
- nombre_gerente

CLIENTE

- **dni_cliente** (PK)
- nombre_cliente
- ciudad_cliente

ESTADIA

- **dni_cliente** (PK, FK → CLIENTE)
- **cod_hotel** (PK, FK → HOTEL)
- **habitacion** (PK)
- **fecha_inicio_hospedaje** (PK)
- **cant_dias_hospedaje**

Ejercicio 3: Programas de Radio:

PROGRAMA<radio, anio, programa, conductor, gerente, frecuencia_radio>

Paso 1:

- a. Una radio se transmite por una única frecuencia en un año determinado, y puede cambiarla en años diferentes: **radio, anio → frecuencia_radio**
- b. Cada radio tiene un único gerente por año, pero el mismo gerente puede repetirse en la misma radio en diferentes años: **radio, anio → gerente**
- c. Un mismo programa puede transmitirse por varias radios y en diferentes años. (No genera dependencia funcional, solo una relación de muchos a muchos.)
- d. Un programa transmitido en una radio en un año determinado tiene un solo conductor: **radio, anio, programa → conductor**

Paso 2:

Atributos primos (forman parte de la clave candidata): radio, anio, programa

Atributos no primos: frecuencia_radio, gerente, conductor

Paso 3:

(radio, anio, programa) Una radio puede tener varios programas y cada año puede variar la frecuencia o el gerente, pero la combinación **radio + año + programa** identifica un programa específico emitido por esa radio en ese año.

Clave candidata (y primaria) final: (radio, anio, programa)

Paso 4:

1FN: La tabla ya está en 1FN porque todos los atributos son atómicos (no hay valores compuestos ni repetidos).

2FN: Debemos eliminar **dependencias parciales**, es decir, atributos que dependen de una parte de la clave compuesta: **radio, anio → frecuencia_radio, gerente**
(estos atributos dependen solo de parte de la clave: “programa” no interviene)

R1 (RadioPorAño):

RADIO_ANIO(radio, anio, frecuencia_radio, gerente)

R2 (ProgramaPorAño):

PROGRAMA(radio, anio, programa, conductor)

3FN: gerente depende directamente de (radio, anio) y no de ningún otro atributo, por lo que no es transitiva.

conductor depende directamente de (radio, anio, programa) y no de ningún otro atributo.

Paso 5:Tablas finales 3FN

RADIO_ANIO (Representa una radio en un año determinado, con su frecuencia y gerente.)

- **radio** (PK compuesta con anio)
- **anio**
- **frecuencia_radio**
- **gerente**

PROGRAMA (Representa un programa transmitido por una radio en un año específico, con su conductor.)

- **radio** (PK, FK → RADIO_ANIO)
- **anio** (PK, FK → RADIO_ANIO)
- **programa** (PK)
- **conductor**

Ejercicio 4: Talleres de autos

TALLER<codigo_sucursal, domicilio_sucursal, telefono_sucursal, codigo_fosa, largo_fosa, ancho_fosa, patente_auto, marca_auto, modelo_auto, dni_cliente, nombre_cliente, celular_cliente, dni_mecanico, nombre_mecanico, email_mecanico>

Paso 1:

. El código_sucursal identifica una sucursal puntual, con su domicilio, teléfono, fosas y mecánicos: **codigo_sucursal → domicilio_sucursal, telefono_sucursal**

b. De las fosas conocemos el código, largo y ancho. El código de fosa es secuencial *dentro de cada sucursal*: **codigo_sucursal, codigo_fosa → largo_fosa, ancho_fosa**

c. En una fosa se arreglan autos.

Cada fosa puede tener varios autos reparados, pero cada auto se asocia con un cliente: **patente_auto → marca_auto, modelo_auto, dni_cliente**

d. Cada cliente tiene un solo DNI, nombre y celular: **dni_cliente → nombre_cliente, celular_cliente**

e. Cada mecánico tiene un solo DNI, nombre y email: **dni_mecanico → nombre_mecanico, email_mecanico**

f. En cada reparación (fosa–auto–mecánico–sucursal) se pueden registrar los vínculos: **codigo_sucursal, codigo_fosa, patente_auto, dni_mecanico** identifican una reparación concreta.

Paso 2:

Atributos primos (forman parte de alguna clave candidata): codigo_sucursal, codigo_fosa, patente_auto, dni_mecanico

Atributos no primos: domicilio_sucursal, telefono_sucursal, largo_fosa, ancho_fosa, marca_auto, modelo_auto, dni_cliente, nombre_cliente, celular_cliente, nombre_mecanico, email_mecanico

Paso 3:

Cada **reparación** queda identificada de forma única por:
(codigo_sucursal, codigo_fosa, patente_auto, dni_mecanico)

Una sucursal puede tener varias fosas.

En cada fosa se pueden reparar varios autos.

En una reparación puede intervenir un mecánico.

Clave candidata (y primaria): (codigo_sucursal, codigo_fosa, patente_auto, dni_mecanico)

Paso 4:

1FN: La tabla original ya está en 1FN: todos los atributos son atómicos.

2FN: Eliminamos dependencias parciales:

codigo_sucursal → domicilio_sucursal, telefono_sucursal

codigo_sucursal, codigo_fosa → largo_fosa, ancho_fosa

patente_auto → marca_auto, modelo_auto, dni_cliente

dni_cliente → nombre_cliente, celular_cliente

dni_mecanico → nombre_mecanico, email_mecanico

R1 (Sucursal):

SUCURSAL(codigo_sucursal, domicilio_sucursal, telefono_sucursal)

R2 (Fosa):

FOSA(codigo_sucursal, codigo_fosa, largo_fosa, ancho_fosa)

(PK compuesta: (codigo_sucursal, codigo_fosa))

R3 (Cliente):

CLIENTE(dni_cliente, nombre_cliente, celular_cliente)

R4 (Auto):

AUTO(patente_auto, marca_auto, modelo_auto, dni_cliente FK → CLIENTE)

R5 (Mecanico):

MECANICO(dni_mecanico, nombre_mecanico, email_mecanico)

R6 (Reparacion):

REPARACION(codigo_sucursal FK → SUCURSAL, codigo_fosa FK → FOSA, patente_auto FK → AUTO, dni_mecanico FK → MECANICO)

(PK compuesta: (codigo_sucursal, codigo_fosa, patente_auto, dni_mecanico))

3FN: Eliminamos dependencias transitivas:

patente_auto → dni_cliente y dni_cliente → nombre_cliente (ya separadas en AUTO y CLIENTE).

codigo_sucursal → domicilio_sucursal y codigo_sucursal, codigo_fosa → largo_fosa (ya separadas).

Paso 5:

SUCURSAL

- codigo_sucursal (PK)
- domicilio_sucursal
- telefono_sucursal

FOSA

- codigo_sucursal (PK, FK → SUCURSAL)
- codigo_fosa (PK)
- largo_fosa
- ancho_fosa

AUTO

- patente_auto (PK)
- marca_auto
- modelo_auto
- dni_cliente (FK → CLIENTE)

CLIENTE

- dni_cliente (PK)
- nombre_cliente
- celular_cliente

MECANICO

- dni_mecanico (PK)
- nombre_mecanico
- email_mecanico

REPARACION

- codigo_sucursal (PK, FK → SUCURSAL)
- codigo_fosa (PK, FK → FOSA)
- patente_auto (PK, FK → AUTO)

- **dni_mecanico (PK, FK → MECANICO)**

Ejercicio 5: Torneo de Ciclismo

TORNEO<cod_torneo, nombre_torneo, cod_corredor, cod_bicicleta, marca_bicicleta, nyap_corredor, sponsor, dni_presidente_sponsor, dni_medico>

Paso 1:

- a. El código del torneo es único y no se repite entre torneos, pero el nombre puede repetirse:
cod_torneo → nombre_torneo
- b. Un corredor corre varios torneos. Tiene un código único por torneo, pero puede variar entre torneos: **cod_torneo, cod_corredor → nyap_corredor**
- c. Cada corredor tiene varias bicicletas asignadas para un torneo: **cod_torneo, cod_corredor, cod_bicicleta → marca_bicicleta**
- d. Los códigos de bicicletas son únicos dentro de un torneo, pero pueden repetirse entre torneos: **cod_torneo, cod_bicicleta → marca_bicicleta**
- e. Cada corredor tiene varios sponsors en un torneo, y un sponsor puede representar a varios corredores: **cod_torneo, cod_corredor, sponsor → dni_presidente_sponsor, dni_medico**
- f. Cada sponsor tiene un único presidente y un único médico: **sponsor → dni_presidente_sponsor, dni_medico(transitiva a través de sponsor)**

Paso 2:

Atributos primos (forman parte de la clave candidata): cod_torneo, cod_corredor, cod_bicicleta, sponsor

Atributos no primos: nombre_torneo, marca_bicicleta, nyap_corredor, dni_presidente_sponsor, dni_medico

Paso 3:

Cada registro representa la participación de un corredor con una bicicleta y sponsor en un torneo: **(cod_torneo, cod_corredor, cod_bicicleta, sponsor)**

Clave candidata (y primaria):

(cod_torneo, cod_corredor, cod_bicicleta, sponsor)

Paso 4:

1FN: La relación original está en 1FN: no hay grupos repetidos ni atributos multivaluados.

2FN: Eliminamos **dependencias parciales** (atributos que dependen de una parte de la clave compuesta).

Dependencias parciales:

- cod_torneo → nombre_torneo
- cod_torneo, cod_corredor → nyap_corredor
- cod_torneo, cod_bicicleta → marca_bicicleta

- sponsor → dni_presidente_sponsor, dni_medico

R1 (Torneo):

TORNEO(cod_torneo, nombre_torneo)

R2 (Corredor):

CORREDOR(cod_torneo, cod_corredor, nyap_corredor)

(PK compuesta: (cod_torneo, cod_corredor))

R3 (Bicicleta):

BICICLETA(cod_torneo, cod_bicicleta, marca_bicicleta)

(PK compuesta: (cod_torneo, cod_bicicleta))

R4 (Sponsor):

SPONSOR(sponsor, dni_presidente_sponsor, dni_medico)

(PK: sponsor)

R5 (Participacion):

PARTICIPACION(cod_torneo FK→TORNEO, cod_corredor FK→CORREDOR, cod_bicicleta FK→BICICLETA, sponsor FK→SPONSOR)

(PK compuesta: (cod_torneo, cod_corredor, cod_bicicleta, sponsor))

3FN

Ver si hay **dependencias transitivas**:

- En PARTICIPACION, los atributos no clave (dni_presidente_sponsor, dni_medico) ya fueron separados en SPONSOR.
- En CORREDOR, nyap_corredor depende solo de (cod_torneo, cod_corredor).
- En BICICLETA, marca_bicicleta depende solo de (cod_torneo, cod_bicicleta).
- En TORNEO, nombre_torneo depende solo de cod_torneo.

Paso 5:

TORNEO

- **cod_torneo (PK)**
- **nombre_torneo**

CORREDOR

- **cod_torneo (PK, FK → TORNEO)**
- **cod_corredor (PK)**
- **nyap_corredor**

BICICLETA

- **cod_torneo (PK, FK → TORNEO)**
- **cod_bicicleta (PK)**
- **marca_bicicleta**

SPONSOR

- **sponsor (PK)**
- **dni_presidente_sponsor**
- **dni_medico**

PARTICIPACION

- **cod_torneo (PK, FK → TORNEO)**
- **cod_corredor (PK, FK → CORREDOR)**
- **cod_bicicleta (PK, FK → BICICLETA)**
- **sponsor (PK, FK → SPONSOR)**

Ejercicio 6: Juegos Olímpicos

JUEGO<anio.olimpiada, pais.olimpiada, nombre.deportista, pais.deportista, nombre.disciplina, asistente>

Paso 1:

- a. pais.olimpiada es el país donde se realizó el juego olímpico del año correspondiente:
anio.olimpiada → pais.olimpiada
- b. pais.deportista es el país que representa el deportista, y es siempre el mismo para ese deportista: **nombre.deportista → pais.deportista**
- c. Por un país, participan varios deportistas en cada juego: (No genera dependencia nueva; describe multiplicidad).
- d. En un año determinado, los juegos olímpicos se hacen en un solo país, pero un país puede haber sido sede en varios años: **anio.olimpiada → pais.olimpiada** (ya incluida)
- e. Cada deportista puede participar en varios juegos olímpicos y en varias disciplinas en diferentes años. Pero en un mismo juego (año), solo participa en una disciplina:
anio.olimpiada, nombre.deportista → nombre.disciplina
- f. Un deportista tiene un asistente en cada juego olímpico, pero puede variar en diferentes años: **anio.olimpiada, nombre.deportista → asistente**

Paso 2:

Atributos primos (forman parte de alguna clave candidata): anio.olimpiada, nombre.deportista

Atributos no primos: pais.olimpiada, pais.deportista, nombre.disciplina, asistente

Paso 3:

Cada participación de un deportista en los juegos olímpicos está determinada por:
(anio.olimpiada, nombre.deportista)

- Un deportista puede participar en varios años distintos.
- En un mismo año, solo puede competir en una disciplina, con un solo asistente.

Clave candidata (y primaria):

(anio.olimpiada, nombre.deportista)

Paso 4:

1FN: La tabla original está en 1FN (todos los atributos son atómicos).

2FN: Eliminamos dependencias parciales, es decir, aquellas que dependen de solo parte de la clave primaria compuesta.

- . **anio.olimpiada → pais.olimpiada**
- . **nombre.deportista → pais.deportista**

R1 (Olimpiada):

OLIMPIADA(anio.olimpiada, pais.olimpiada)

R2 (Deportista):

DEPORTISTA(nombre.deportista, pais.deportista)

R3 (Participacion):

PARTICIPACION(anio.olimpiada FK → OLIMPIADA, nombre.deportista FK → DEPORTISTA, nombre.disciplina, asistente)
(PK compuesta: (anio.olimpiada, nombre.deportista))

3FN: Verificamos dependencias transitivas:

En OLIMPIADA, pais.olimpiada depende directamente de la clave (anio.olimpiada).

En DEPORTISTA, pais.deportista depende directamente de la clave (nombre.deportista).

En PARTICIPACION, nombre.disciplina y asistente dependen solo de la clave compuesta (anio.olimpiada, nombre.deportista).

Paso 5:

OLIMPIADA

- **anio.olimpiada (PK)**
- **pais.olimpiada**

DEPORTISTA

- **nombre.deportista (PK)**
- **pais.deportista**

PARTICIPACION

- **anio_olimpiada** (PK, FK → OLIMPIADA)
- **nombre_deportista** (PK, FK → DEPORTISTA)
- **nombre_disciplina**
- **asistente**