

TP 3 - Normalización

Ejercicio 2)

Dependencias funcionales:

$\text{cod_hotel} \rightarrow \text{dni_gerente}$

$\text{dni_gerente} \rightarrow \text{nombre_gerente}$

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

$\text{cod_hotel} \rightarrow \text{cantidad_habitaciones}$

$\text{cod_hotel} \rightarrow \text{direccion_hotel}, \text{ciudad_hotel}$

$\text{dni_cliente} \rightarrow \text{nombre_cliente}, \text{ciudad_cliente}$

Atributos primos y no primos:

atributos primos: $\text{dni_cliente}, \text{cod_hotel}, \text{fecha_inicio_hospedaje}, \text{habitacion}$

atributos no primos:

$\text{cantidad_habitaciones}, \text{direccion_hotel}, \text{ciudad_hotel}, \text{dni_gerente}, \text{nombre_gerente}, \text{nombre_cliente}, \text{ciudad_cliente}, \text{cant_dias_hospedaje}$

Determinar clave candidata:

La relacion estadia tiene como clave candidata:

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

el cliente (dni_cliente) su nombre y ciudad

el hotel (cod_hotel) su cantidad de habitaciones, direccion, ciudad, gerente y su nombre

Normalizar a 3FN:

CLIENTE (dni_cliente PK, nombre_cliente, ciudad_cliente)

GERENTE (dni_gerente PK, nombre_gerente)

HOTEL (cod_hotel PK, direccion_hotel, ciudad_hotel, cantidad_habitaciones, dni_gerente FK)

ESTADIA (dni_cliente FK, cod_hotel FK, fecha_inicio_hospedaje, habitacion, cant_dias_hospedaje, PK (dni_cliente, cod_hotel, fecha_inicio_hospedaje, habitacion))

Crear tablas en 3FN:

Script adjunto en github.

Ejercicio 3)

A) Dependencias funcionales:

1. radio , anio \rightarrow frecuencia_radio
2. radio , anio \rightarrow gerente
3. radio, anio, programa \rightarrow conductor
4. radio, anio, programa \rightarrow radio, anio, programa, conductor, gerente, frecuencia_radio

B) Atributos primos y no primos:

Atributos primos: radio , anio , programa.

Atributos no primos: conductor , gerente , frecuencia_radio.

C) Clave candidata:

(radio, anio, programa)

D) Normalización:

RADIO

(radio PK, anio, frecuencia_radio, dni_gerente FK)

GERENTE

(dni_gerente PK, nombre_gerente)

PROGRAMA

(radio FK, anio, programa, conductor,
PK (radio, anio, programa))

E) Crear tablas en 3FN:

Script adjunto en github.

Ejercicio 4

- a) $\text{codigo_sucursal} \rightarrow \text{domicilio_sucursal}, \text{telefono_sucursal}$
- b) $\{\text{codigo_sucursal}, \text{codigo_fosa}\} \rightarrow \text{largo_fosa}, \text{ancho_fosa}$
- c) $\text{patente_auto} \rightarrow \text{marca_auto}, \text{modelo_auto}, \text{dni_cliente}$
- d) $\text{dni_cliente} \rightarrow \text{nombre_cliente}, \text{celular_cliente}$
- e) $\text{dni_mecanico} \rightarrow \text{nombre_mecanico}, \text{email_mecanico}$
- f) $\{\text{codigo_sucursal}, \text{codigo_fosa}, \text{patente_auto}, \text{dni_mecanico}\}$
- g)

Atributos primos: $\text{codigo_sucursal}, \text{codigo_fosa}, \text{patente_auto}, \text{dni_mecanico}$

Atributos no primos: $\text{domicilio_sucursal}, \text{telefono_sucursal}, \text{largo_fosa}, \text{ancho_fosa}, \text{marca_auto}, \text{modelo_auto}, \text{dni_cliente}, \text{nombre_cliente}, \text{celular_cliente}, \text{nombre_mecanico}, \text{email_mecanico}$

Claves candidatas: la clave que identifica cada laburo del taller es:
 $\{\text{codigo_sucursal}, \text{codigo_fosa}, \text{patente_auto}, \text{dni_mecanico}\}$

Normalizar a 3FN:

SUCURSAL ($\text{codigo_sucursal}, \text{domicilio_sucursal}, \text{telefono_sucursal}$)

fosa ($\text{codigo_sucursal}, \text{codigo_fosa}, \text{largo_fosa}, \text{ancho_fosa}$)

Cliente ($\text{dni_cliente}, \text{nombre_cliente}, \text{celular_cliente}$)

Auto ($\text{patente_auto}, \text{marca_auto}, \text{modelo_auto}, \text{dni_cliente}$)

Mecanico($\text{dni_mecanico}, \text{nombre_mecanico}, \text{email_mecanico}$)

Trabajo ($\text{codigo_sucursal}, \text{codigo_fosa}, \text{patente_auto}, \text{dni_mecanico}$)

Crear tablas en 3FN:

Script adjunto en github.

Ejercicio 5)

Dependencias funcionales:

$\text{cod_torneo} \rightarrow \text{nombre_torneo}$

$(\text{cod_torneo}, \text{cod_corredor}) \rightarrow \text{nyap_corredor}$

$(\text{cod_torneo}, \text{cod_bicicleta}) \rightarrow \text{marca_bicicleta}$

$\text{sponsor} \rightarrow \text{dni_presidente_sponsor}, \text{dni_medico}$

Atributos primos y no primos:

atributos primos: $\text{cod_torneo}, \text{cod_corredor}, \text{cod_bicicleta}, \text{sponsor}$

atributos no primos:

$\text{nombre_torneo}, \text{marca_bicicleta}, \text{nyap_corredor}, \text{dni_presidente_sponsor}, \text{dni_medico}$

Determinar clave candidata:

La relacion TORNEO tiene como clave candidata:

$(\text{cod_torneo}, \text{cod_corredor}, \text{cod_bicicleta}, \text{sponsor})$

Normalizar a 3FN:

TORNEO ($\text{cod_torneo}, \text{nombre_torneo}$)

CORREDOR_TORNEO ($\text{cod_torneo}, \text{cod_corredor}, \text{nyap_corredor}$)

BICICLETA_TORNEO ($\text{cod_torneo}, \text{cod_bicicleta}, \text{marca_bicicleta}$)

SPONSOR ($\text{sponsor}, \text{dni_presidente_sponsor}, \text{dni_medico}$)

PARTICIPACION ($\text{cod_torneo}, \text{cod_corredor}, \text{cod_bicicleta}, \text{sponsor}$)

Crear tablas en 3FN:

Script adjunto en github.

Ejercicio 6)

dependencias funcionales:

- a) $\text{anio_olimpiada} \rightarrow \text{pais_olimpiada}$
- b) $\text{nombre_deportista} \rightarrow \text{pais_deportista}$ } {
- c) $\{\text{anio_olimpiada}, \text{nombre_deportista}\} \rightarrow \text{asistente}$
- d) $\{\text{anio_olimpiada}, \text{nombre_deportista}\} \rightarrow \text{nombre_disciplina}$

atributos primos:

$\text{anio_olimpiada}, \text{nombre_deportista}$

atributos no primos:

$\text{pais_olimpiada}, \text{pais_deportista}, \text{nombre_disciplina}, \text{asistente}$

clave candidata:

Para identificar cuando un atleta enm las olimpiadas
 $\{\text{anio_olimpiada}, \text{nombre_deportista}\}$

Normalizar a 3FN:

Olimpiada ($\text{anio_olimpiada}, \text{pais_olimpiada}$)

Deportista ($\text{nombre_deportista}, \text{pais_deportista}$)

Participacion ($\text{anio_olimpiada}, \text{nombre_deportista}, \text{nombre_disciplina}, \text{asistente}$)

Crear tablas en 3FN:

Scrip adjunto en github.