

Base de Datos (75.15 / 75.28 / 95.05)**Evaluación Parcial - Primera Oportunidad**

TEMA 2021211	SQL			Fecha: 27 de octubre de 2021 Padrón: 103945 Apellido: Biancardi Nombre: Julian
	AR			
	MOD			
	DR			
Corrigió:				<input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente
Nota:				

Criterio de aprobación: El examen está compuesto por 7 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. El examen se aprueba con nota mayor o igual a 4(cuatro) y la condición de aprobación es desarrollar al menos un ítem bien (B/B-) de entre los dos de SQL, un ítem bien de entre los dos de diseño relacional, y un ítem bien entre los tres que restan en álgebra relacional y mapeo de modelos. Adicionalmente, no deberá haber más de dos ítems mal o no desarrollados.

1. Para cada uno de los siguientes requerimientos, escriba una única consulta SQL que dé cumplimiento al mismo:

- a. Sabiendo que el ganador de la prueba es el que obtuvo el menor tiempo sumados todos los tiempos de todas las etapas, obtenga los datos del ciclista ganador. Debe mostrar el código del ganador, el tiempo total empleado en la prueba y su puntaje total obtenido.

Nota: Observe que para ganar, el ciclista debe haber completado todas las etapas existentes.

- **EQUIPOS**(cod_equipo, nombre_equipo, director)

// ('EOK', 'Eolo Kometa', Jesús Hernández)<

- **CICLISTAS**(cod_ciclista, nombre_ciclista, fecha_nac, nacionalidad, peso, **cod_equipo**)

// (624, 'Alberto Contador', 06/12/1982, 'ESP', 62, 'EOK')

- **ETAPAS**(num_etapa, tipo, fecha, distancia, salida, llegada, altura, pendiente_media)

// (3, 'MONTAÑA', 16/08/2021, 203, 'STO DOMINGO DE SILOS', 'PICÓN BLANCO', 1500, 25)

- **MARCAS**(cod_ciclista, num_etapa, tiempo, puntaje)

// (624, 3, 9h 25' 44", 63)

Respuesta:

```
WITH ciclistas_todas_etapas AS(  
SELECT DISTINCT cod_ciclista
```

```
FROM marcas m1
WHERE NOT EXISTS(
    SELECT num_etapa
    FROM etapas
    EXCEPT
    SELECT num_etapa
    FROM marcas m2
    WHERE m1.cod_ciclista = m2.cod_ciclista
))

WITH tiempos_ciclistas AS(
SELECT cod_ciclista, SUM(tiempo), SUM(puntaje)
FROM marcas INNER JOIN ciclistas_todas_etapas
USING(cod_ciclista)
GROUP BY cod_ciclista)

SELECT cod_ciclista, tiempo, puntaje
FROM tiempos_ciclistas
WHERE tiempo = (SELECT MIN(tiempo)
                FROM tiempos_ciclistas);
```

- b. Teniendo los mismos esquemas que en el ítem a), obtener en una única fila, a los ganadores de las tres primeras etapas. Por ejemplo, si en la tabla **MARCAS** tiene:

cod_ciclista	num_etapa	tiempo	puntaje
10	1	1000	20
10	2	1000	20
20	1	800	60
20	2	1500	15
30	1	1150	20
30	2	1800	40
30	3	1800	20
10	3	2200	15

Lo requerido será:

etapas	Etapa 1	Etapa 2	Etapa 3
ganadores por etapas	20	10	30

- **EQUIPOS**(cod_equipo, nombre_equipo, director)
//('EOK', 'Eolo Kometa', Jesús Hernández)<
- **CICLISTAS**(cod_ciclista, nombre_ciclista, fecha_nac, nacionalidad, peso, **cod_equipo**)
//(624, 'Alberto Contador', 06/12/1982, 'ESP', 62, 'EOK')
- **ETAPAS**(num_etapa, tipo, fecha, distancia, salida, llegada, altura, pendiente_media)
//(3, 'MONTAÑA', 16/08/2021, 203, 'STO DOMINGO DE SILOS', 'PICÓN BLANCO', 1500, 25)
- **MARCAS**(cod_ciclista, num_etapa, tiempo, puntaje)
//(624, 3, 9h 25' 44", 63)

Respuesta:

```
SELECT "ganadores por etapas" as etapas
m1.cod_ciclista as Etapa1, m2.cod_ciclista as Etapa2,
m3.cod_ciclista as Etapa3
FROM marcas m1, marcas m2, marcas m3
WHERE
m1.etapa = 1 AND m1.tiempo = (SELECT MIN(tiempo)FROM
```

```
marcas WHERE etapa = 1)
AND
m2.etapa = 2 AND m1.tiempo = (SELECT MIN(tiempo)FROM
marcas WHERE etapa = 2)
AND
m2.etapa = 3 AND m1.tiempo = (SELECT MIN(tiempo)FROM
marcas WHERE etapa = 3)
```

- c. Obtener aquellos niveles para los que hay al menos una escuela (de dicho nivel) en todas las comunas.

- **comunas**(nro_comuna, población)
- **barrios**(nombre_barrio, superficie, nro_comuna)
- **escuelas**(id, nombre, dirección, nivel, cantidad_alumnos, barrio, es_publica)

Respuesta:

```
SELECT DISTINCT nivel
FROM escuelas e1
WHERE NOT EXIST (
SELECT nro_comuna
FROM comunas

EXECPT

SELECT nro_comuna
FROM escuelas e2 INNER JOIN barrios b ON (e2.barrio =
b.nombre_barrio)
WHERE e1.nivel = e2.nivel
)
```

- b. En la *Copa Davis*, para elegir qué equipo será local en un enfrentamiento es muy importante conocer quién fue el local la última vez que los mismos equipos jugaron entre sí.

Sabiendo que se tiene una tabla “resultados” que indica para cada enfrentamiento los equipos local y visitante y la cantidad de partidos ganados en dicho enfrentamiento por cada equipo:

■ **resultados**(año, local, visitante, ganados_local, ganados_visitante)

Escriba una consulta SQL que devuelva quién fue el equipo local la última vez que se enfrentaron Argentina (ARG) e Italia (ITA).

***Nota:** No es necesario considerar el caso de que no haya enfrentamientos entre ambos equipos.*

A modo de ejemplo (mostrando sólo las últimas 3 participaciones de argentina en la copa), si la tabla de resultados tuviera las siguientes filas, la respuesta debería ser “ARG”.

En cambio si se hubiera pedido el último local para Argentina vs. Croacia, la respuesta correcta hubiera sido “CRO”.

resultados

año	local	visitante	ganados_local	ganados_visitante
2016	ARG	POL	2	3
2016	ITA	ARG	1	3
2016	ING	ARG	2	3
2016	CRO	ARG	2	3
2017	ARG	ITA	2	3
2019	ARG	ESP	1	2

Respuesta:

```
SELECT r.local
FROM resultados as r
WHERE (r.local = 'ARG' AND r.visitante = 'ITA') OR
(r.local = 'ITA' AND r.visitante = 'ARG')
ORDER BY año DESC
OFFSET 0 ROWS FETCH FIRST 1 ROWS ONLY
```

- b. Para las cafeterías que sirven al menos todos los cafés que sirve "Petit Colon" y que abrieron antes que esa cafetería, mostrar su nombre, capacidad y fecha de apertura.

■ **cafeterías**(id_cafetería, nombre, calle_nombre, calle_altura, tamaño, capacidad, fecha_apertura)

■ **cafés_servidos**(id_café, id_cafetería, fecha_introducción)

■ **cafés**(id_café, nombre, tipo_grano, cantidad_agua, cantidad_leche, cantidad_azúcar)

■ **balances_mensuales**(id_cafetería, mes, año, ganancias, gastos, impuestos)

Respuesta:

```
WITH cafeterias_tc AS (  
  SELECT id_cafeteria  
  FROM cafes_servidos cf1  
  WHERE NOT EXIST (  
    SELECT id_cafe  
    FROM cafes_servidos cf2  
    WHERE cf1.id_cafeteria = cf2.id_cafeteria  
  
  EXECPT  
  
  SELECT id_cafe  
  FROM cafes_servidos  
  WHERE id_cafeteria = (SELECT id_cafeteria FROM  
    cafeterias WHERE nombre = 'Petit Colon')  
  ))  
  
SELECT nombre, capacidad, fecha_apertura  
FROM cafeterias INNER JOIN cafeterias_tc ctc  
  USING(id_cafeteria)  
WHERE nombre != 'Petit Colon' AND fecha_apertura <  
  (SELECT fecha_apertura FROM cafeterias WHERE nombre =  
    'Petit Colon')
```

b) Dados los siguientes esquemas de relación que almacenan información sobre distintas reservas naturales en el país, sus balances mensuales, los ejemplares que viven en ellas y su clasificación científica:

- `reserva_natural(id_reserva, nombre, ubicación, año_creación, tamaño, capacidad)`
- `balance(id_reserva, año, mes, ingreso_neto, gastos, subsidios)`
- `animal_en_reserva(id_reserva, nombre_científico, cantidad_ejemplares)`
- `animal(nombre_científico, nombre_común, familia, género, especie)`

Escriba una única consulta SQL que dé cumplimiento al siguiente requerimiento:

- Obtener para cada animal (indicando su nombre científico y nombre común en el resultado) la cantidad total de ejemplares del mismo, contabilizando únicamente las reservas cuya capacidad sea superior a 100 km².

Respuesta:

```
SELECT nombre_cientifico, nombre_comun,
SUM(cantidad_ejemplares)
FROM animal_en_reserva INNER JOIN animal
USING(nombre_cientifico)
WHERE id_reserva IN (SELECT id_reserva FROM
reserva_natural WHERE capacidad > 100km)
GROUP BY (nombre_cientifico, nombre_comun)
```

a) Considere la siguiente instancia de la tabla ONG:

id	nombre	categoría	ubicación	año_creación
1	Actitud Animal	Animales	<i>null</i>	1989
2	AEDIN	Educación	Teodoro García 2948	1964
3	Fundación Perro Comunitario	Animales	Del Tero 25	2010
4	AKIM	Social	Arcos 2319	1992
5	Cascos Verdes	Social	Leandro N. Alem 1026	2007
6	ALPI	Discapacidad	Soler 3945	1946
7	ASDRA	Discapacidad	Uriarte 2011	1988
8	Asociación Conciencia	Educación	Tucumán 731	1999
9	CAII	Educación	<i>null</i>	2001
10	Casa Rafael	Educación	Carlos Calvo 378	2006
11	Enseñá por Argentina	Educación	Teodoro García 2964	1995

Tabla 1: ONG(id, nombre, categoría, ubicación, año_creación)

Nos interesa obtener, para las categorías que no tienen ninguna ONG sin ubicación, el nombre de la categoría, el año promedio de creación y el año más reciente de creación de las ONG de dicha categoría. Muestre los resultados ordenados por la columna correspondiente al año más reciente de creación, en forma descendente. Escriba una única consulta SQL que dé cumplimiento al requerimiento.

Por ejemplo, para una instancia de ONG como la mostrada en la *Tabla 1*, el resultado esperado es el siguiente:

categoría	año_creación_promedio	año_reciente_creación
Social	1999.5	2007
Discapacidad	1967	1988

Tabla 2: resultado(categoría, año_creación_promedio, año_reciente_creación)

Respuesta:

```
SELECT categoria, AVG(año_creacion)
año_creacion_promedio, MAX(año_creacion)
año_reciente_creacion
FROM ong
GROUP BY (categoria)
HAVING (COUNT(*) = COUNT(ubicacion))
```


b) Escriba una única consulta SQL que dé cumplimiento al siguiente requerimiento:

- Obtener el DNI, apellido, nombre y teléfono de los colaboradores que pertenezcan a una ONG de la categoría “Social”, y a al menos otras dos ONG.

- ONG(id, nombre, categoría, ubicación, año_creación)
- colaborador(dni, nombre, apellido, año_nacimiento, teléfono, email)
- colaborador_por_ong(id_ong, dni, puesto, fecha_asociación)
- balance_mensual(id_ong, mes, año, gastos, donaciones, subsidios_gobierno)

Respuesta:

```
SELECT id
FROM ong
WHERE categoria = "Social"
```

b) Dados los siguientes esquemas de relación que almacenan información sobre reservas naturales en el país, sus balances mensuales, los ejemplares que viven en ellas y su clasificación científica:

- reserva_natural(id_reserva, nombre, ubicación, año_creación, tamaño, capacidad)
- balance(id_reserva, año, mes, ingreso_netos, gastos, subsidios)
- animal_en_reserva(id_reserva, nombre_científico, cantidad_ejemplares)
- animal(nombre_científico, nombre_común, familia, género, especie)

Escriba una única consulta SQL que dé cumplimiento al siguiente requerimiento:

- Obtener el nombre, ubicación, id_reserva, año de creación e ingreso neto total anual de las reservas naturales cuyos ingresos netos totales en el año 2018 hayan sido superiores a los 10 millones de pesos.

Respuesta:

```
WITH reservas as (
SELECT id_reserva, SUM(ingreso_netos) as ingreso_netos
FROM balance
WHERE año = 2018
GROUP BY (id_reserva)
HAVING (SUM(ingreso_netos) > 10000000))

SELECT nombre, ubicacion, id_reserva, año_creacion,
ingreso_netos
FROM reserva_natural INNER JOIN reservas USING(id_reserva)
```

- a) Un cliente nos solicitó que a partir de los datos mostrados en la Tabla 1, extraigamos la información que muestra la Tabla 2. Para ello, exprese para cada contribuyente qué porcentaje representan sus facturas respecto al total de facturas. Escriba una única consulta SQL que dé cumplimiento al requerimiento.

num_factura	fecha	CUIT
10	10/07/2018	27123456781
11	10/07/2018	20555555559
13	10/07/2018	27123456781
14	10/07/2018	20444444442
15	10/07/2018	27123456781
17	11/07/2018	20444444442

Tabla 1: Facturas(num_factura, fecha, CUIT)

CUIT	cantidad_facturas	porcentaje_del_total
27123456781	3	50,00
20444444442	2	33,33
20555555559	1	16,66

Tabla 2: Resultado requerido

Respuesta:

```
SELECT CUIT, COUNT(num_factura) as cantidad_facturas,
COUNT(num_factura) / (SELECT COUNT(*) FROM facturas) as
porcentaje_del_total
FROM facturas
GROUP BY (CUIT)
```

- b) Dados los siguientes esquemas de relación que almacenan información sobre los contribuyentes, las actividades económicas que desarrollan y los pagos mensuales que efectúan a la AFIP por cada actividad que realizan:

- contribuyentes(CUIT, razón_social, tipo)
- actividades(cod_actividad, descripción_actividad)
- pagos(CUIT, cod_actividad, año, mes, importe)
- realiza(CUIT, cod_actividad, fecha_inicio)

Escriba una única consulta SQL que dé cumplimiento al siguiente requerimiento:

“Obtener las actividades económicas (código y descripción) para las cuales la recaudación de la AFIP, durante el primer semestre del año actual, haya aumentado en al menos un 31 % con respecto a la recaudación –para esa actividad– en el mismo semestre del año 2017”.

Nota: No es necesario que liste a aquellas actividades que no registraron ningún pago durante el 2017.

Respuesta:

```
WITH pagos_1semestre as (  
SELECT cod_actividad,año, SUM(importe) as importe_total  
FROM pagos  
WHERE (año = 2021 OR año = 2020) AND (mes >= 1 AND mes <= 6)  
GROUP BY (cod_actividad,año)  
)  
  
WITH cod_actividades as (  
SELECT cod_actividad  
FROM pagos_1semestre as p1, pagos_1semestre as p2  
WHERE p1.cod_actividad = p2.cod_actividad AND  
p1.año != p2.año AND  
p1.importe_total >= 1.31 * p2.importe_total  
)  
  
SELECT cod_actividad, descripcion_actividad  
FROM cod_actividades INNER JOIN (actividades)  
USING(cod_actividad)
```

- facturas (nro_factura, fecha, CUIT)
- detalle_facturas (nro_factura , nro_artículo, cantidad)
- articulos (nro_artículo, precio, descripcion, peso, color)
- clientes(CUIT, razón_social, fecha_de_alta)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- a) Obtener número y descripción de aquellos artículos que no fueron vendidos en el año 2016 ni en el 2017.

Respuesta:

```
WITH articulos_result as (  
SELECT nro_articulo  
FROM articulos  
  
EXCEPT  
  
SELECT DISTINCT nro_articulo  
FROM facturas INNER JOIN detalle_facturas USING(nro_factura)  
WHERE (fecha = 2016) OR (fecha = 2017))  
  
SELECT nro_articulo, descripcion
```

```
FROM articulos_result INNER JOIN articulos  
USING(nro_artiuclo)
```

- b) Obtener el número y fecha de aquellas facturas en las que se hayan facturado todos los artículos de color rojo.

Respuesta:

```
SELECT DISTINCT nro_factura, fecha  
FROM detalle_facturas f1 INNER JOIN facturas USING(nro_factura)  
WHERE NOT EXISTS (  
    SELECT nro_articulo  
    FROM articulos  
    WHERE color = 'rojo'  
  
    EXCEPT  
  
    SELECT nro_articulo  
    FROM detalle_facturas f2  
    WHERE f1.nro_factura = f2.nro_factura  
)
```

(SQL) Los siguientes esquemas representan a una tienda virtual que vende notebooks. En cada una de sus ventas la tienda establece la garantía a través de un soporte técnico en distintas localidades.

- usuario (num_usuario, nombre, teléfono, email, dirección, localidad)
- compra (cod_compra, cod_equipo, cod_soporte, num_usuario, fecha, cantidad)
- notebook (cod_equipo, marca, descripción, detalle, precio)
- soporte_técnico (cod_soporte, nombre, teléfono, email, dirección, localidad)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- Mostrar el código y nombre de soporte técnico para aquellos soportes que fueron asignados al menos a una compra de notebook de marca 'HP' y jamás fueron asignados a una compra de notebook de marca 'LENOVO'.
- Mostrar para cada usuario su número, nombre y cantidad total de equipos comprados en 2017. Incluir también a los usuarios que no hayan efectuado ninguna compra en 2017.

Respuesta:

```
WITH soportes as (  
  SELECT DISTINCT cod_soporte  
  FROM compra INNER JOIN notebook USING(cod_equipo)  
  WHERE marca = 'HP'  
  
  EXCEPT  
  
  SELECT DISTINCT cod_soporte  
  FROM compra INNER JOIN notebook USING(cod_equipo)  
  WHERE marca = 'LENOVO')  
  
SELECT cod_soporte, nombre  
FROM soporte s INNER JOIN soporte_tecnico USING(cod_soporte)
```

Respuesta:

```
SELECT num_usuario, nombre, SUM(cantidad) as cantidad_comprados  
FROM compra INNER JOIN usuario USING(num_usuario)  
WHERE fecha = 2017  
GROUP BY (num_usuario)
```

1. (SQL) Los siguientes esquemas representan a una tienda virtual que vende notebooks. En cada una de sus ventas la tienda establece la garantía a través de un soporte técnico en distintas localidades.

- usuario (num_usuario, nombre, teléfono, email, dirección, localidad)
- compra (cod_compra, cod_equipo, cod_soporte, num_usuario, fecha, cantidad)
- notebook (cod_equipo, marca, descripción, detalle, precio)
- soporte_técnico (cod_soporte, nombre, teléfono, email, dirección, localidad)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- Para cada par de soportes técnicos distintos que estén en la misma localidad, devolver el código de soporte y nombre de cada uno de ellos. No se debe devolver cada par más de una vez.
- Listar el nombre de usuario y el teléfono para aquellos usuarios que tengan al menos una compra con cada uno de los soportes técnicos de la localidad de Lanús.

Respuesta:

```
SELECT s1.cod_soporte, s1.nombre, s2.cod_soporte, s2.nombre
FROM soporte_tecnico s1, soporte_tecnico s2
WHERE s1.cod_soporte != s2.cod_soporte AND
s1.localidad = s2.localidad

EXECPT

SELECT s1.cod_soporte, s1.nombre, s2.cod_soporte, s2.nombre
FROM soporte_tecnico s1, soporte_tecnico s2
WHERE s1.cod_soporte != s2.cod_soporte AND
s1.localidad = s2.localidad AND
s1.cod_soporte > s2.cod_soporte
```

Respuesta:

```
SELECT DISTINCT nombre, telefono
FROM compra c1 INNER JOIN usuario USING(num_usuario)
WHERE NOT EXISTS(
  SELECT cod_soporte
  FROM soporte_tecnico
  WHERE localidad = 'Lanus'

EXECPT

SELECT cod_soporte
FROM compra c2
WHERE c1.num_usuario = c2.num_usuario
)
```

1. (SQL) Dados los siguientes esquemas sobre el envío de productos a clientes:

- clientes (cod_cliente, nombre, tipo_cliente)
- productos (nro_prod, descripción, unidad_medida, precio)
- envíos (cod_envio, cod_cliente, dirección, ciudad, provincia)
- detalles_envios (cod_envio, nro_prod, cantidad_enviada)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- a) Para aquellos productos que hayan sido enviados al menos una vez a todos los clientes, devuelva su número de producto (columna nro_prod) y descripción.
- b) Para cada envío hecho al cliente de código 1000 (columna cod_cliente), indique el código de envío, la dirección a la que debe ser enviado, cuántos distintos productos se enviaron en el envío y la cantidad total enviada de todos los productos que conforman el envío.

Respuesta:

```
WITH pallclientes as
SELECT DISTINCT(nro_prod)
FROM envios e1 INNER JOIN detalles_envios d1 USING(cod_envio)
WHERE NOT EXISTS(
    SELECT cod_cliente
    FROM clientes

    EXEPT

    SELECT cod_cliente
    FROM envios e2 INNER JOIN detalles_envios d2 USING(cod_envio)
    WHERE d1.nro_prod = d2.nro_prod
)

SELECT nro_prod, descripcion
FROM pallclientes p1 INNER JOIN productos USING(nro_prod)
```

Respuesta:

```
SELECT cod_envio, direccion, COUNT(DISTINCT nro_prod)
cant_productos, SUM(cantidad_enviada) cantidad_total
FROM detalles_envios INNER JOIN envios USING(cod_envio)
WHERE cod_cliente = 1000
GROUP BY (cod_envio)
```

1. (SQL) Dados los siguientes esquemas sobre el envío de productos a clientes:

- clientes (cod_cliente, nombre, tipo_cliente)
- productos (nro_prod, descripción, unidad_medida, precio)
- envíos (cod_envio, cod_cliente, dirección, ciudad, provincia)
- detalles_envios (cod_envio, nro_prod, cantidad_enviada)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- a) Para aquellos clientes a los que se les haya enviado al menos uno de los productos de mayor precio, devuelva su código de cliente y nombre.
- b) Para cada tipo de cliente, devuelva el tipo de cliente, cuántos clientes son de dicho tipo y cuántos envíos se les hizo a clientes de ese tipo.

Respuesta:

```
WITH producto_max as(
SELECT nro_prod
FROM productos
WHERE precio = (SELECT MAX(precio ) FROM productos))

WITH clientes_result as(
SELECT cod_cliente
FROM detalles_envios INNER JOIN envios USING(cod_envio)
WHERE nro_prod IN producto_max)

SELECT cod_cliente,nombre
FROM clientes_result cr INNER JOIN clientes USING(cod_cliente)
```

Respuesta:

```
SELECT tipo_cliente, COUNT(DISTINCT cod_cliente),
COUNT(cod_envio)
FROM clientes INNER JOIN envios USING(cod_cliente)
GROUP BY(tipo_cliente)
```

1. (SQL) Dados los siguientes esquemas sobre el envío de productos a clientes:

- clientes (cod_cliente, nombre, tipo_cliente)
- productos (nro_prod, descripcion, unidad_medida, precio)
- envios (cod_envio, cod_cliente, direccion, ciudad, provincia)
- detalles_envios (cod_envio, nro_prod, cantidad_enviada)

Resuelva cada una de las siguientes consultas con una única sentencia SQL:

- a) Obtenga el número (columna *nro_prod*) y descripción de aquellos productos que fueron enviados al cliente de código 1000 (columna *cod_cliente*) y no fueron enviados ni al cliente de código 2000 ni al de código 3000.
- b) Para cada cliente al que se le hayan efectuado al menos tres envíos, devuelva su código de cliente, nombre y cantidad de distintos productos que se le enviaron. Ordene el listado por esta última cantidad descendente.

Respuesta:

```
WITH prod_1000 as(
SELECT DISTINCT nro_prod
FROM detalles_envios INNER JOIN envios USING(cod_envio)
WHERE cod_cliente = 1000

EXCEPT

SELECT DISTINCT nro_prod
FROM detalles_envios INNER JOIN envios USING(cod_envio)
WHERE cod_cliente = 2000 or cod_cliente = 3000
)

SELECT nro_prod,descripcion
FROM prod_1000 pr INNER JOIN productos USING(nro_prod)
```

Respuesta:

```
SELECT cod_cliente,nombre, COUNT(DISTINCT nro_prod)
FROM detalles_envios INNER JOIN envios USING(cod_envio)
GROUP BY(cod_cliente)
HAVING COUNT(DISTINCT cod_envio) >= 3
ORDER BY COUNT(DISTINCT nro_prod) DESC
```

2. (SQL) Considere las siguientes tablas que almacenan información sobre los distintos circuitos de Fórmula 1®, las carreras que se corrieron en cada uno de ellos a lo largo de los años, los pilotos que han participado de cada carrera y las escuderías a las que pertenecían, y el tiempo que cada piloto ha marcado en cada vuelta que logró terminar:

- Circuitos(nombre_circuito, ciudad, país)
- Carreras(nombre_circuito, temporada)
- Pilotos(nombre_piloto, fecha_nacimiento, nacionalidad)
- Participación(nombre_piloto, nombre_circuito, temporada)
- EscuderíasPilotos(nombre_piloto, temporada, escudería)
- Timings(nombre_piloto, nombre_circuito, temporada, nro_vuelta, tiempo)

Escriba una consulta SQL que encuentre para cada circuito cuál fue la escudería que logró marcar el tiempo de vuelta histórico más corto, indicando el nombre del circuito y el nombre de la escudería que posee dicho récord de tiempo de vuelta. Si varias escuderías ostentan el mismo récord, devuelva una línea por cada una, pero no devuelva más de una vez a la misma escudería para un mismo circuito.

Respuesta:

```
WITH records as (  
  SELECT nombre_circuito, MIN(tiempo)  
  FROM timings  
  GROUP BY(nombre_circuito))  
  
WITH pilotos_r as (  
  SELECT DISTINCT (nombre_piloto, nombre_ciruito, temporada)  
  FROM timings t INNER JOIN records USING(nombre_circuito)  
  WHERE t.timepo == r.tiempo)  
  
SELECT escuderia, nombre_circuito  
FROM pilotos_r INNER JOIN escuderias  
USING(nombre_piloto,temporada)
```

2. (*Álgebra relacional*) Dados los mismos esquemas del ejercicio 1a, y utilizando la siguiente notación para representar las operaciones del álgebra relacional: π , σ , ρ , \leftarrow , \bowtie , \cup , $-$, \cap , \bowtie , \div , resuelva las siguientes consultas:

a. Obtener el nombre del ciclista que hizo mejor tiempo en la etapa 21.

■ **EQUIPOS**(cod_equipo, nombre_equipo, director)

// ('EOK', 'Eolo Kometa', Jesús Hernández')

■ **CICLISTAS**(cod_ciclista, nombre_ciclista, fecha_nac, nacionalidad, peso, **cod_equipo**)

// (624, 'Alberto Contador', 06/12/1982, 'ESP', 62, 'EOK')

■ **ETAPAS**(num_etapa, tipo, fecha, distancia, salida, llegada, altura, pendiente_media)

// (3, 'MONTAÑA', 16/08/2021, 203, 'STO DOMINGO DE SILOS', 'PICÓN BLANCO', 1500, 25)

■ **MARCAS**(cod_ciclista, num_etapa, tiempo, puntaje)

// (624, 3, 9h 25' 44'', 63)

Respuesta (utilice fuente tamaño 14):

$\text{marcas_etapa21} \leftarrow \sigma_{\text{num_etapa}=21}(\text{marcas})$

$\text{mejor_tiempo} \leftarrow \pi_{\text{tiempo}} \text{marcas_etapa21} - \pi_{\text{m1.tiempo}} \sigma_{\text{m1.tiempo} > \text{m2.tiempo}} (\rho_{\text{m1}}(\text{marcas_etapa21}) \times \rho_{\text{m2}}(\text{marcas_etapa21}))$

$\text{mejor_ciclista} \leftarrow \pi_{\text{cod_ciclista}}(\text{marcas_etapa21} \bowtie_{\text{tiempo}} \text{mejor_tiempo})$

$\pi_{\text{nombre}}(\text{mejor_ciclista} \bowtie_{\text{cod_ciclista}} \text{ciclistas})$

Reservorio de caracteres: π_{xxx} , σ_{xxx} , ρ_{xxx} , \leftarrow , \bowtie , \cup , $-$, \cap , \bowtie_{xxx} , \div , \wedge , \vee

Tip: Presionando <CTRL> + ';' puede convertir texto a subíndice.

- b. Mostrar el nombre de el/los ciclistas que hicieron un tiempo menor a 7 horas en cada una de las etapas de tipo LLANO.

■ **EQUIPOS**(cod_equipo, nombre_equipo, director)

// ('EOK', 'Eolo Kometa', Jesús Hernández)

■ **CICLISTAS**(codi_ciclista, nombre_ciclista, fecha_nac, nacionalidad, peso, **cod_equipo**)

// (624, 'Alberto Contador', 06/12/1982, 'ESP', 62, 'EOK')

■ **ETAPAS**(num_etapa, tipo, fecha, distancia, salida, llegada, altura, pendiente_media)

// (3, 'MONTAÑA', 16/08/2021, 203, 'STO DOMINGO DE SILOS', 'PICÓN BLANCO', 1500, 25)

■ **MARCAS**(cod_ciclista, num_etapa, tiempo, puntaje)

// (624, 3, 9h 25' 44'', 63)

Respuesta (utilice fuente tamaño 14):

$$\text{etapas_llano} \leftarrow \pi_{\text{num_etapa}} \sigma_{\text{tipo}='LLANO'}(\text{etapas})$$

$$\text{marcas_llano_7hs} \leftarrow \pi_{\text{cod_ciclista}, \text{num_etapa}} \sigma_{\text{tiempo} < 7\text{h}} (\text{etapas_llano} \bowtie_{\text{num_etapa}} \text{marcas})$$

$$\text{ciclistas_cod} \leftarrow (\text{marcas_llano_7hs} \div \text{etapas_llano})$$

$$\pi_{\text{nombre}}(\text{ciclistas_cod} \bowtie_{\text{cod_ciclista}} \text{ciclistas})$$

Reservorio de caracteres: π_{xxx} , σ_{xxx} , ρ_{xxx} , \leftarrow , x , \cup , $-$, \cap , \bowtie_{xxx} , \div , \wedge , \vee

Tip: Presionando <CTRL> + ',' puede convertir texto a subíndice.

a) Obtener el nombre y la capacidad de aquellas reservas naturales creadas con posterioridad a 1995 y que contengan ejemplares de todos los animales del género *Felis*.

- $\text{reserva_natural}(\text{id_reserva}, \text{nombre}, \text{ubicación}, \text{año_creación}, \text{tamaño}, \text{capacidad})$
- $\text{balance}(\text{id_reserva}, \text{año}, \text{mes}, \text{ingreso_neto}, \text{gastos}, \text{subsídios})$
- $\text{animal_en_reserva}(\text{id_reserva}, \text{nombre_científico}, \text{cantidad_ejemplares})$
- $\text{animal}(\text{nombre_científico}, \text{nombre_común}, \text{familia}, \text{género}, \text{especie})$

Respuesta (utilice fuente tamaño 14):

$$\begin{aligned} \text{animales_felis} &= \pi_{\text{nombre_científico}} \sigma_{\text{genero}='Felis'}(\text{animal}) \\ \text{id_reservas_felis} &= \pi_{\text{id_reserva}, \text{nombre_científico}} \text{animal_en_reserva} \div \text{animales_felis} \\ \text{reservas_post1995} &= \sigma_{\text{anio_creacion} < 1995}(\text{reserva_natural} \bowtie_{\text{id_reserva}} \text{id_reservas_felis}) \\ \pi_{\text{nombre}, \text{capacidad}}(\text{reservas_post1995}) \end{aligned}$$

b) Obtener todas las reservas naturales que tengan menor cantidad de ejemplares de lechuzones (nombre científico: *Asio clamator*) que la reserva '*Pamakawa reserva natural*', pero que no contengan ejemplares de águilas (nombre científico: *Geranoaetus melano-leucus*).

Respuesta (utilice fuente tamaño 14):

$$\begin{aligned} \text{id_reservaP} &= \pi_{\text{id_reserva}} \sigma_{\text{nombre}='Pamakawa..'}(\text{reserva_natural}) \\ \text{cant_lechuzones_P} &= \pi_{\text{cantidad_ejemplares}} \sigma_{\text{nombre}='Asio cl...'}(\text{animal_en_reserva} \bowtie_{\text{id_reserva}} \text{id_reservaP}) \\ \text{reservas_con_aguilas} &= \pi_{\text{id_reserva}} \sigma_{\text{nombre_c}='Geranoaetus ..'}(\text{animal_en_reserva}) \\ \text{reservas_menor} &= \pi_{\text{id_reserva}} \sigma_{\text{nombre_c}='Asio cl..' \text{ AND } \text{cantidad} < \text{cantidadP}}(\text{animal_en_reserva} \times \text{cant_lechuzones_P}) \\ (\text{reservas_menor} - \text{reservas_con_aguilas}) &\bowtie_{\text{id_reserva}} \text{reserva_natural} \end{aligned}$$

a) Obtener el id y el nombre de la/s ONG que haya/n recibido el mayor importe mensual en donaciones de entre todos los importes mensuales de donaciones de 2018.

- `ONG(id, nombre, categoría, ubicación, año_creación)`
- `colaborador(dni, nombre, apellido, año_nacimiento, teléfono, email)`
- `colaborador_por_ong(id_ong, dni, puesto, fecha_asociación)`
- `balance_mensual(id_ong, mes, año, gastos, donaciones, subsidios_gobierno)`

Respuesta (utilice fuente tamaño 14):

$$b_2018 \leftarrow \pi_{id_ong, donaciones} \sigma_{año=2018}(balance_mensual)$$

$$dmax_2018 \leftarrow \pi_{donaciones} b_2018 - \pi_{b1.donaciones} \sigma_{b1.donaciones < b2.donaciones} (\rho_{b1}(b_2018) \times \rho_{b2}(b_2018))$$

$$ids_ong \leftarrow \pi_{id_ong} (b_2018 \bowtie_{donaciones} dmax_2018)$$

$$\pi_{id_ong, nombre}(ong \bowtie_{id_ong} ids_ong)$$

b) Obtener el DNI, apellido y nombre de aquellos colaboradores que trabajen en dos o más ONG.

Respuesta (utilice fuente tamaño 14):

$$c1 \leftarrow \rho_{c1}(colaborador_por_ong)$$

$$c2 \leftarrow \rho_{c2}(colaborador_por_ong)$$

$$c_dni \leftarrow \pi_{c1.dni} \sigma_{c1.id_ong \neq c2.id_ong} (c1 \bowtie_{c1.dni=c2.dni} c2)$$

$$\pi_{dni, apellido, nombre}(ccolaborador \bowtie_{dni} c_dni)$$

a) Obtener el nombre y el año de inauguración de aquellas reservas naturales de capacidad mayor a 500 personas y que contengan ejemplares de todos los animales del género *Canis*.

- `reserva_natural(id_reserva, nombre, ubicación, año_creación, tamaño, capacidad)`
- `balance(id_reserva, año, mes, ingreso_neto, gastos, subsidios)`
- `animal_en_reserva(id_reserva, nombre_científico, cantidad_ejemplares)`
- `animal(nombre_científico, nombre_común, familia, género, especie)`

Respuesta (utilice fuente tamaño 14):

```
id_canis ← πnombre_científicoσgenero = 'Canis' (animal)

id_reservas ← (πid_reserva, nombre_científico animal_en_reserva) ÷ id_canis

πnombre, año_creacionσcapacidad > 500 (reserva_natural ⋈id_reserva id_reservas)
```

a) Obtener el CUIT y la razón social de el/los cliente/s que efectuó/aron el pago de mayor importe.

- `contribuyentes(CUIT, razón_social, tipo)`
- `actividades(cod_actividad, descripción_actividad)`
- `pagos(CUIT, cod_actividad, año, mes, importe)`
- `realiza(CUIT, cod_actividad, fecha_inicio)`

Respuesta (utilice fuente tamaño 14):

```
max_importe ← πimportepagos - πp1.importeσp1.importe > p2.importe (ρp1 pagos x ρp2 pagos )

cuits ← πCUIT(pagos ⋈importe max_importe)

πCUIT, razon_social(contribuyentes ⋈CUIT cuits)
```

- b) Obtener el código y descripción de las actividades que registran pagos del cliente de CUIT 20777777771 y no registran pagos del cliente de CUIT 27888888881.

Respuesta (utilice fuente tamaño 14):

```
act_207 ← πcod_actividadσCUIT = 20777777771(pagos)
act_278 ← πcod_actividadσCUIT = 27888888881(pagos)

act ← act_207 - act_278

(actividades ⋈cod_actividad act)
```

- a) Obtener el CUIT y razón social de los clientes que compraron los artículos número 10 y 20 (es decir, ambos artículos), pero que no hayan comprado el artículo número 30.

- facturas (nro_factura, fecha, CUIT)
- detalle_facturas (nro_factura , nro_artículo, cantidad)
- articulos (nro_artículo, precio, descripcion, peso, color)
- clientes(CUIT, razón_social, fecha_de_alta)

Respuesta (utilice fuente tamaño 14):

```
facturas_aux ← (detalle_facturas ⋈nro_factura facturas)

cuit_10 ← πcuitσnro_articulo = 10(facturas_aux)
cuit_20 ← πcuitσnro_articulo = 20(facturas_aux)
cuit_30 ← πcuitσnro_articulo = 30(facturas_aux)

cuits ← (cuit_10 ∩ cuit_20) - cuit_30

πcuit,razon_social(clientes ⋈cuit cuits)
```


b) Obtener el número de artículo y descripción del (ó de los) artículo/s más pesado/s.

Respuesta (utilice fuente tamaño 14):

$$\text{max_peso} \leftarrow \pi_{\text{peso}} \text{articulos} - \pi_{a1.\text{peso}} \sigma_{a1.\text{peso} > a2.\text{peso}} (\rho_{a1} \text{articulos} \times \rho_{a2} \text{articulos})$$

$$\pi_{\text{nro_articulo, descripcion}} (\text{articulos} \bowtie_{\text{peso}} \text{max_peso})$$

(Álgebra relacional) Dados los mismos esquemas del ejercicio 1 y utilizando la siguiente notación para representar las operaciones del álgebra relacional: $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$, resuelva las siguientes consultas:

- a) Mostrar los nombres de los usuarios que realizaron más de una compra.
- b) Listar los nombres de los usuarios que compraron notebooks de todas las marcas.
 - usuario (num_usuario, nombre, teléfono, email, dirección, localidad)
 - compra (cod_compra, cod_equipo, cod_soporte, num_usuario, fecha, cantidad)
 - notebook (cod_equipo, marca, descripción, detalle, precio)
 - soporte_técnico (cod_soporte, nombre, teléfono, email, dirección, localidad)

Respuesta (utilice fuente tamaño 14):

$$c1 \leftarrow \pi_{\text{cod_compra, num_usuario}} \text{compra}$$

$$c2 \leftarrow \pi_{\text{cod_compra, num_usuario}} \text{compra}$$

$$\text{id_users} \leftarrow \pi_{c1.\text{num_usuario}} \sigma_{c1.\text{um_usuario} = c2.\text{um_usuario} \wedge c1.\text{cod_compra} \neq c2.\text{cod_compra}} (c1 \times c2)$$

$$\pi_{\text{nombre}} (\text{usuario} \bowtie_{\text{num_usuario}} \text{id_users})$$

Respuesta (utilice fuente tamaño 14):

$$\text{marcas} \leftarrow \pi_{\text{marca}} (\text{notebook})$$

$$\text{users_marcas} \leftarrow \pi_{\text{num_usuario, marca}} (\text{compra} \bowtie_{\text{cod_equipo}} \text{notebook})$$

$$\text{id_users} \leftarrow (\text{user_marcas} \div \text{marcas})$$

$$\pi_{\text{nombre}} (\text{usuario} \bowtie_{\text{num_usuario}} \text{id_users})$$

2. (*Álgebra relacional*) Dados los mismos esquemas del ejercicio 1 y utilizando la siguiente notación para representar las operaciones del álgebra relacional: $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$, resuelva las siguientes consultas:

- a) Listar los nombres de los Soportes Técnicos que deban atender, por la garantía de compra, a usuarios de Lanús.
- b) Mostrar los nombres de los Soportes Técnicos a los que se les haya asignado (en distintas compras) la garantía de equipos de todas las marcas de notebooks registradas.
 - usuario (num_usuario, nombre, teléfono, email, dirección, localidad)
 - compra (cod_compra, cod_equipo, cod_soporte, num_usuario, fecha, cantidad)
 - notebook (cod_equipo, marca, descripción, detalle, precio)
 - soporte_técnico (cod_soporte, nombre, teléfono, email, dirección, localidad)

Respuesta (utilice fuente tamaño 14):

```
users_lanus ←  $\pi_{\text{num\_usuario}} \sigma_{\text{localidad} = \text{'Lanu'}}(\text{usuario})$   
  
cod_soportes ←  $\pi_{\text{cod\_soporte}}(\text{compra} \bowtie_{\text{num\_usuario}} \text{users\_lanus})$   
  
 $\pi_{\text{nombre}}(\text{soporte\_tecnico} \bowtie_{\text{cod\_soporte}} \text{cod\_soportes})$ 
```

Respuesta (utilice fuente tamaño 14):

```
marcas ←  $\pi_{\text{marca}}(\text{notebook})$   
  
compras_marcas ←  $\pi_{\text{cod\_soporte, marca}}(\text{compra} \bowtie_{\text{cod\_equipo}} \text{notebook})$   
  
cod_soportes ←  $(\text{compras\_marcas} \div \text{marcas})$   
  
 $\pi_{\text{nombre}}(\text{soporte\_tecnico} \bowtie_{\text{cod\_soporte}} \text{cod\_soportes})$ 
```

2. (*Álgebra relacional*) Dados los mismos esquemas del ejercicio 1 y utilizando la siguiente notación para representar las operaciones del álgebra relacional: $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$, resuelva la siguiente consulta:

- Obtener los clientes (columnas cod_cliente, nombre y tipo_cliente) que hayan recibido productos cuya unidad de medida sea la tonelada (columna unidad_medida) tanto en la provincia de Santa Fé como en la del Neuquén (en ambas provincias, aunque no necesariamente el mismo producto en las dos).

- clientes (cod_cliente, nombre, tipo_cliente)
- productos (nro_prod, descripción, unidad_medida, precio)
- envíos (cod_envio, cod_cliente, dirección, ciudad, provincia)
- detalles_envios (cod_envio, nro_prod, cantidad_enviada)

Respuesta (utilice fuente tamaño 14):

```
p_tonelada ←  $\pi_{\text{nro\_prod}} \sigma_{\text{unidad\_medida} = \text{'tonelada'}}(\text{productos})$ 

e_tonelada ←  $\pi_{\text{cod\_envio}}(\text{detalles\_envios} \bowtie_{\text{nro\_prod}} p\_tonelada)$ 

c_cod_sf ←  $\pi_{\text{cod\_cliente}} \sigma_{\text{provincia} = \text{'Santa Fe'}}(\text{envíos} \bowtie_{\text{cod\_envio}} e\_tonelada)$ 
c_cod_n ←  $\pi_{\text{cod\_cliente}} \sigma_{\text{provincia} = \text{'Neuquen'}}(\text{envíos} \bowtie_{\text{cod\_envio}} e\_tonelada)$ 

clientes  $\bowtie_{\text{cod\_cliente}} (c\_cod\_sf \cap c\_cod\_n)$ 
```

2. (*Álgebra relacional*) Dados los mismos esquemas del ejercicio 1 y utilizando la siguiente notación para representar las operaciones del álgebra relacional: $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$, resuelva la siguiente consulta:

- Obtener los datos de los clientes (su código, nombre y tipo) a los que se les hayan enviado todos los productos.
- clientes (cod_cliente, nombre, tipo_cliente)
 - productos (nro_prod, descripción, unidad_medida, precio)
 - envíos (cod_envio, cod_cliente, dirección, ciudad, provincia)
 - detalles_envios (cod_envio, nro_prod, cantidad_enviada)

Respuesta (utilice fuente tamaño 14):

```
id_prod ←  $\pi_{\text{nro\_prod}}(\text{productos})$ 

aux ←  $\pi_{\text{nro\_prod, cod\_cliente}}(\text{envíos} \bowtie_{\text{cod\_envio}} \text{detalles\_envios})$ 

id_clientes ←  $(aux \div id\_prod)$ 

clientes  $\bowtie_{\text{cod\_cliente}} id\_clientes$ 
```

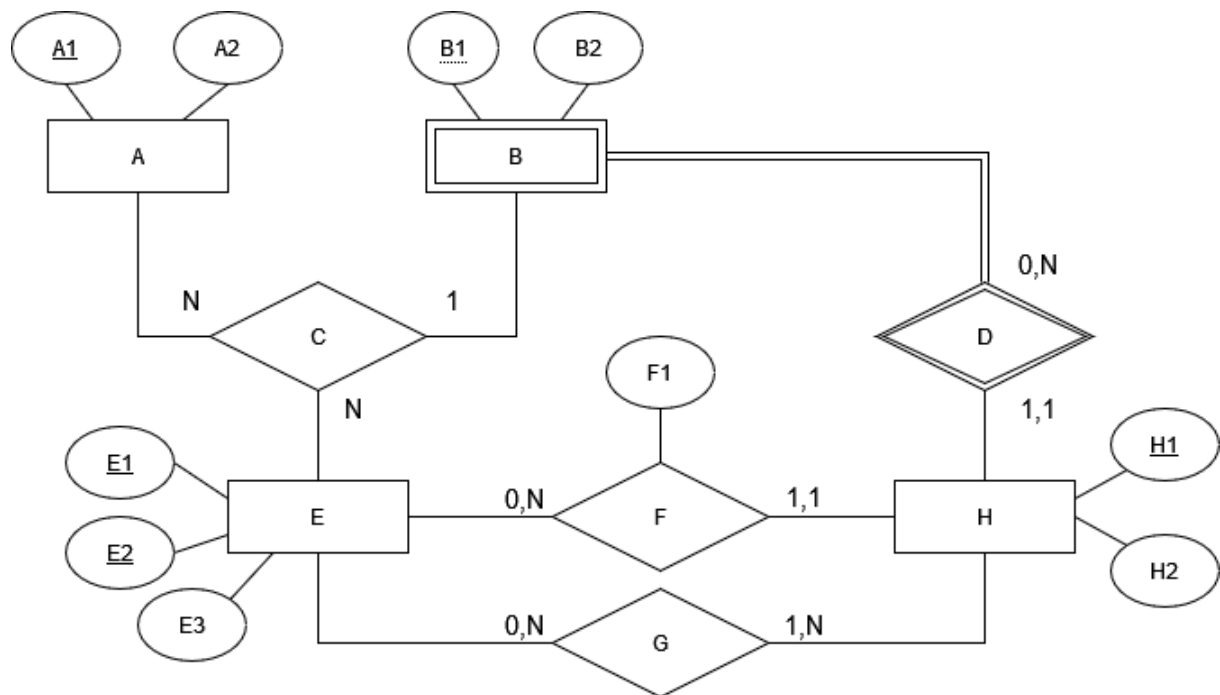
2. (*Álgebra Relacional*) Dados los mismos esquemas del ejercicio 1 y utilizando la siguiente notación para representar las operaciones del álgebra relacional: $\pi, \sigma, \times, \cup, -, \cap, \bowtie, \div$, resuelva la siguiente consulta:

Obtener los clientes, con sus datos completos, que hayan comprado el o los productos más caros.

Respuesta (utilice fuente tamaño 14):

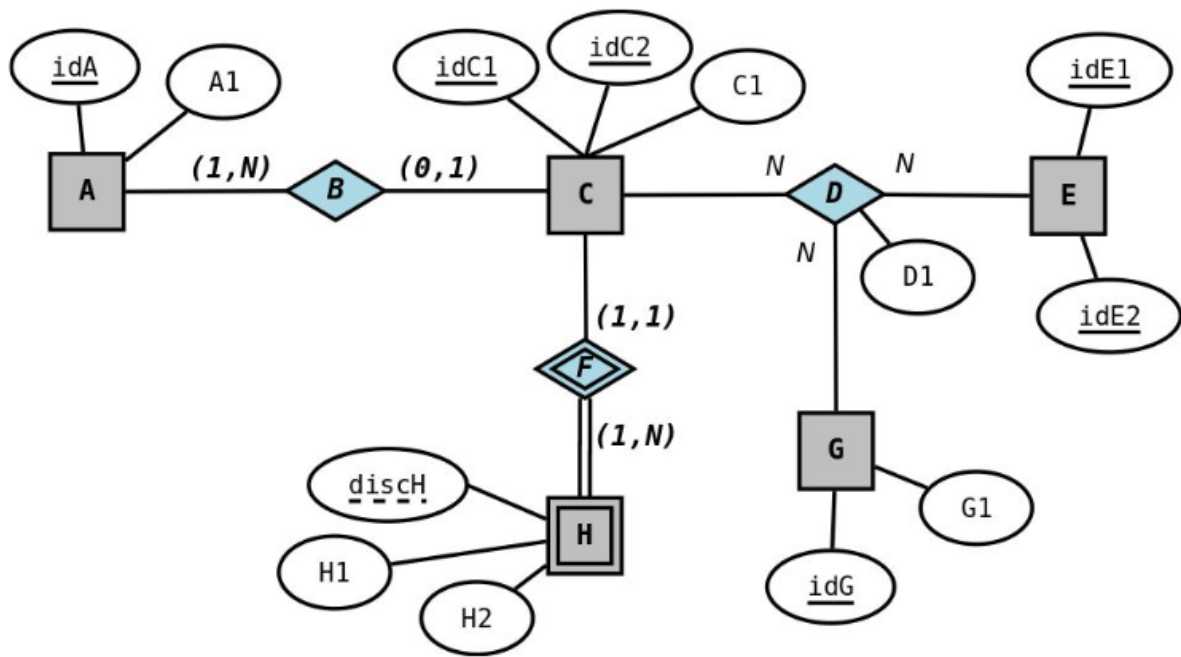
$$\text{max_precio} \leftarrow \pi_{\text{precio}}(\text{producto} - \pi_{p1.\text{precio}} \sigma_{p1.\text{precio} < p2.\text{precio}} (\rho_{p1} \text{producto} \times \rho_{p2} \text{producto}))$$
$$\text{id_prod} \leftarrow \pi_{\text{nro_prod}}(\text{producto} \bowtie_{\text{precio}} \text{max_precio})$$
$$\text{id_envio} \leftarrow \pi_{\text{cod_envio}}(\text{detalles_envios} \bowtie_{\text{nro_prod}} \text{id_prod})$$
$$\text{id_cliente} \leftarrow \pi_{\text{cod_cliente}}(\text{envio} \bowtie_{\text{cod_envio}} \text{id_envio})$$
$$\text{cliente} \bowtie_{\text{cod_cliente}} \text{id_cliente}$$

3. (Modelado)



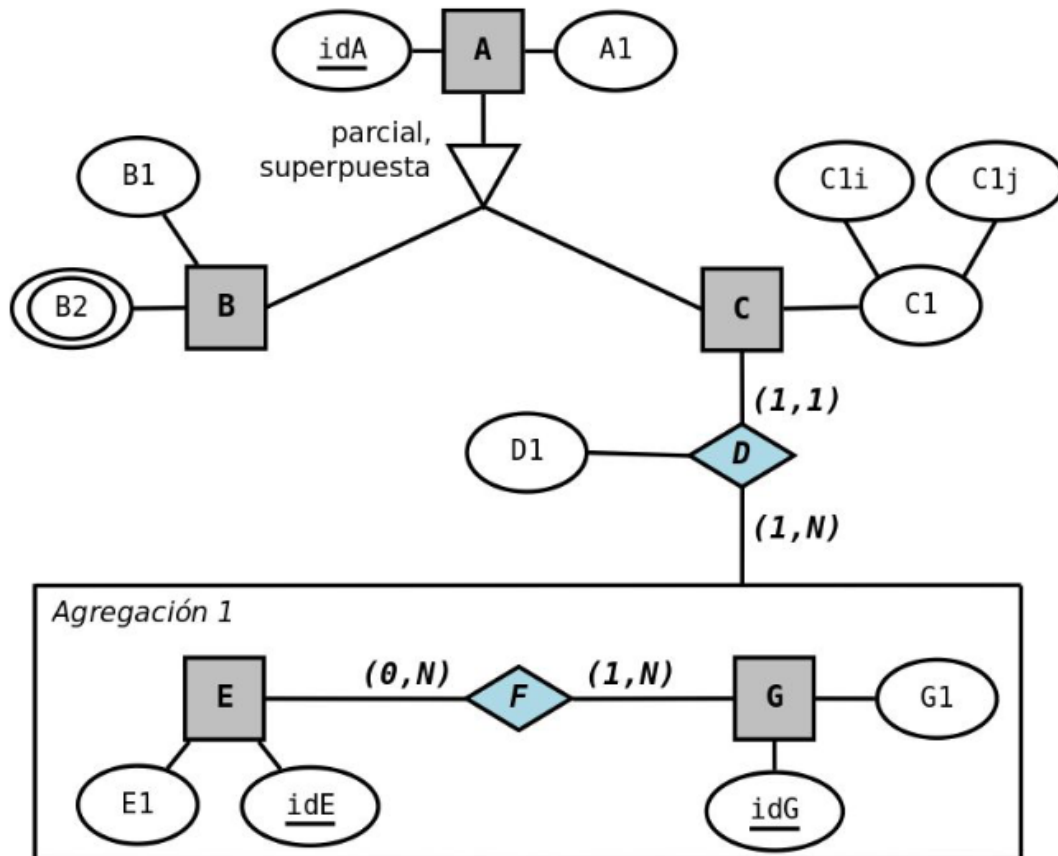
Respuesta:

Relación	CKs	PK	FKs
A(A1,A2)	{A1}	{{A1}}	
H(H1,H2)	{H1}	{{H1}}	
B(H1,B1,B2)	{H1,B1}	{{H1,B1}}	{H1} ref H
E(E1,E2,E3,H1,F1)	{E1,E2}	{{E1,E2}}	{H1} ref H
C(A1,E1,E2,H1,B1)	{A1,E1,E2}	{{A1,E1,E2}}	{A1} ref A {E1,E2} ref E {H1,B1} ref B
G(E1,E2,H1)	{E1,E2,H1}	{{E1,E2,H1}}	{E1,E2} ref E {H1} ref H



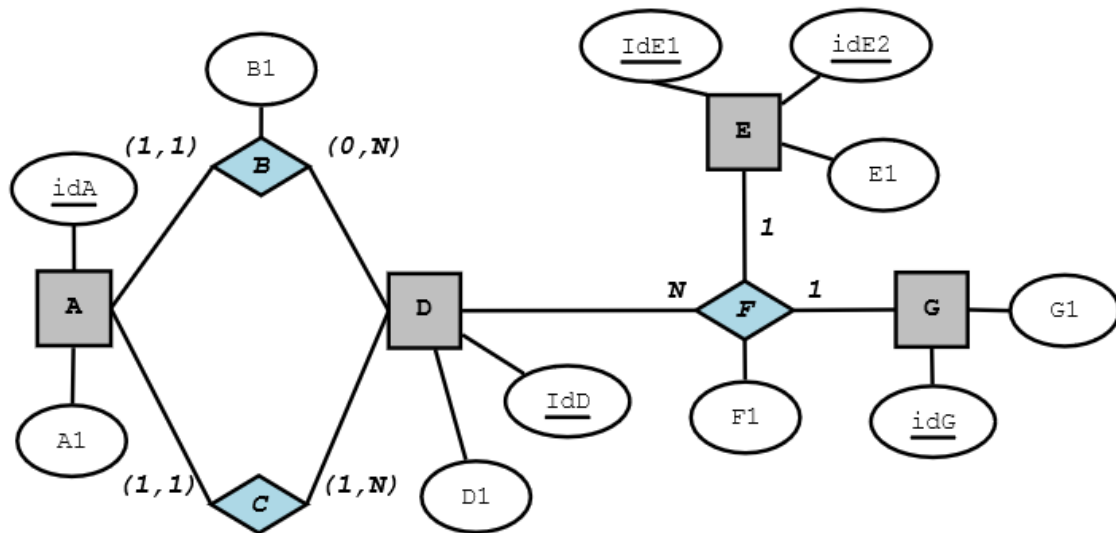
Respuesta:

Relación	CKs	PK	FKs
A(idA,a1)	idA	ida	
B(ida,idc1,idc2)	(idc1, idc2)	(idc1, idc2)	ida ref A (idc1, idc2) ref C
C(idc1,idc2,c1)	(idc1, idc2)	(idc1, idc2)	
D(idc1,idc2,ide1,ide2,idg,d1)	(idc1, idc2,ide1,ide2,idg)	(idc1, idc2,ide1,ide2,idg)	(idc1, idc2) ref C (ide1,ide2) ref E idg ref G
E(ide1,ide2)	(ide1,ide2)	(ide1,ide2)	
G(idg,g1)	idg	idg	
H(idc1,idc2,discH,h1,h2)	(idc1,idc2,discH)	(idc1,idc2,discH)	(idc1, idc2) ref C



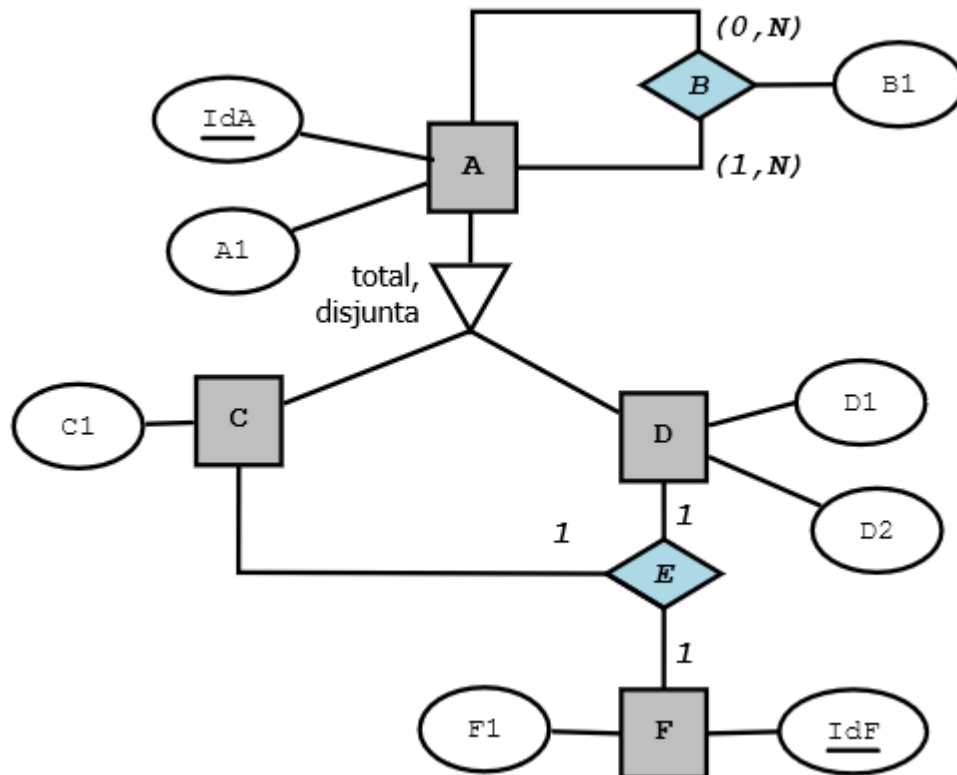
Respuesta:

Relación	CKs	PK	FKs
A(ida,a1)	ida	ida	
B(ida,b1)	ida	ida	ida ref A
B2(ida, b2)	(ida, b2)	(ida, b2)	ida ref A
C(ida,c1i,c1j)	ida	ida	ida ref A
D(ida,ide,idg,d1)	(ide,idg)	(ide,idg)	ida ref A ide ref E idg ref G
E(ide,e1)	ide	ide	
G(idg,g1)	idg	idg	
F(ide,idg)	(ide,idg)	(ide,idg)	ide ref E idg ref G



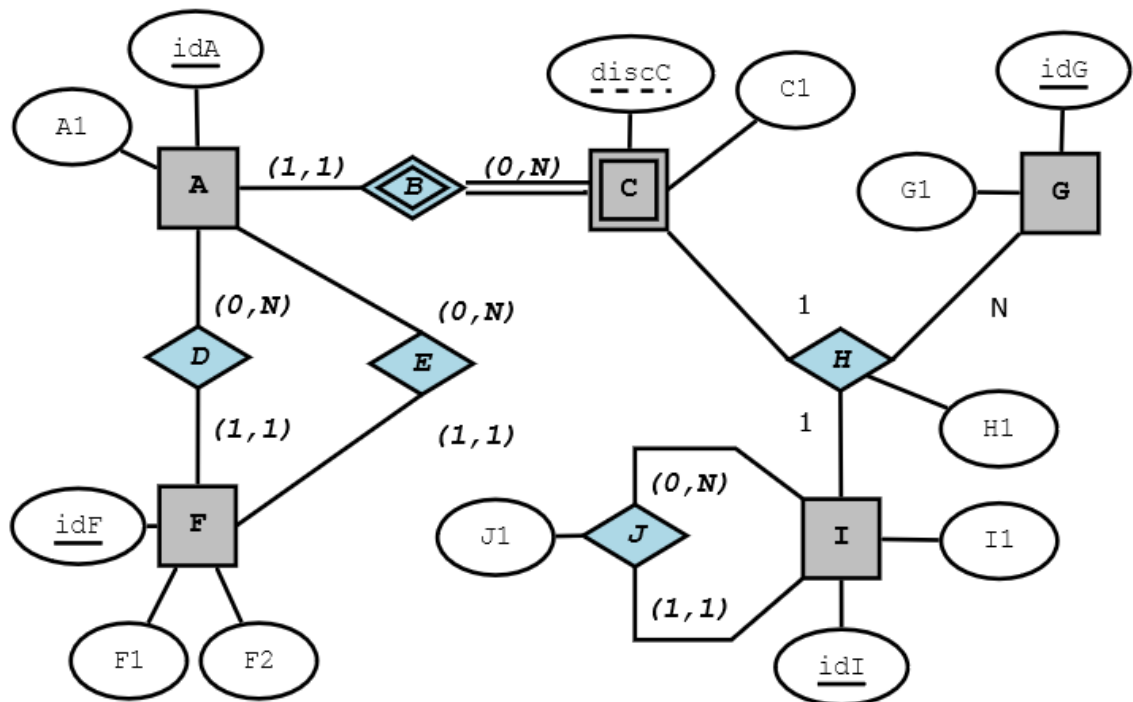
Respuesta:

Relación	CKs	PK	FKs
A(ida,a1)	ida	ida	
D(idd,d1,ida,b1)	idd	idd	ida ref A
E(ide1,ide2,e1)	(ide1,ide2)	(ide1,ide2)	
G(idg,g1)	idg	idg	
F(idd,ide1,ide2,idg,f1)	(idd,ide1,ide2) , (idd,idg)	(idd,ide1,ide2)	idd ref D (ide1,ide2) ref E idg ref G



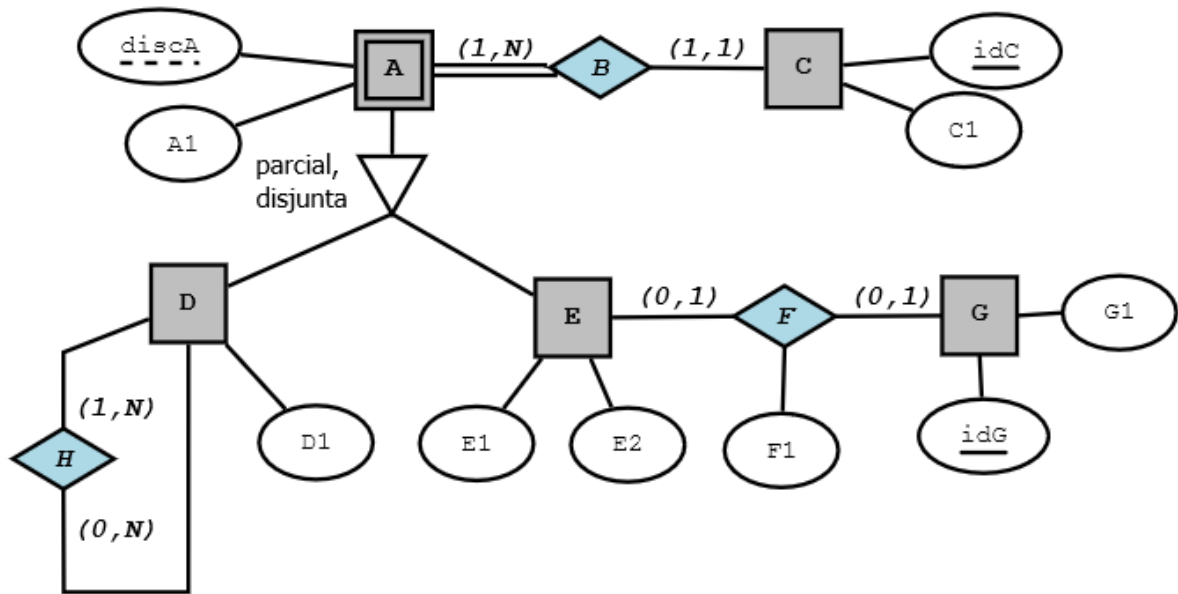
Respuesta:

Relación	CKs	PK	FKs
A(ida,a1)	ida	ida	
B(ida,ida',b1)	(ida,ida')	(ida,ida')	ida ref A ida' ref A
C(ida,c1)	ida	ida	ida ref A
D(ida,d1,d2)	ida	ida	ida ref A
F(idf,f1)	idf	idf	
E(idac,idad,idf)	(idac,idad) , (idac,idf) , (idad,idf)	(idac,idf)	idac ref A idad ref A idf ref F



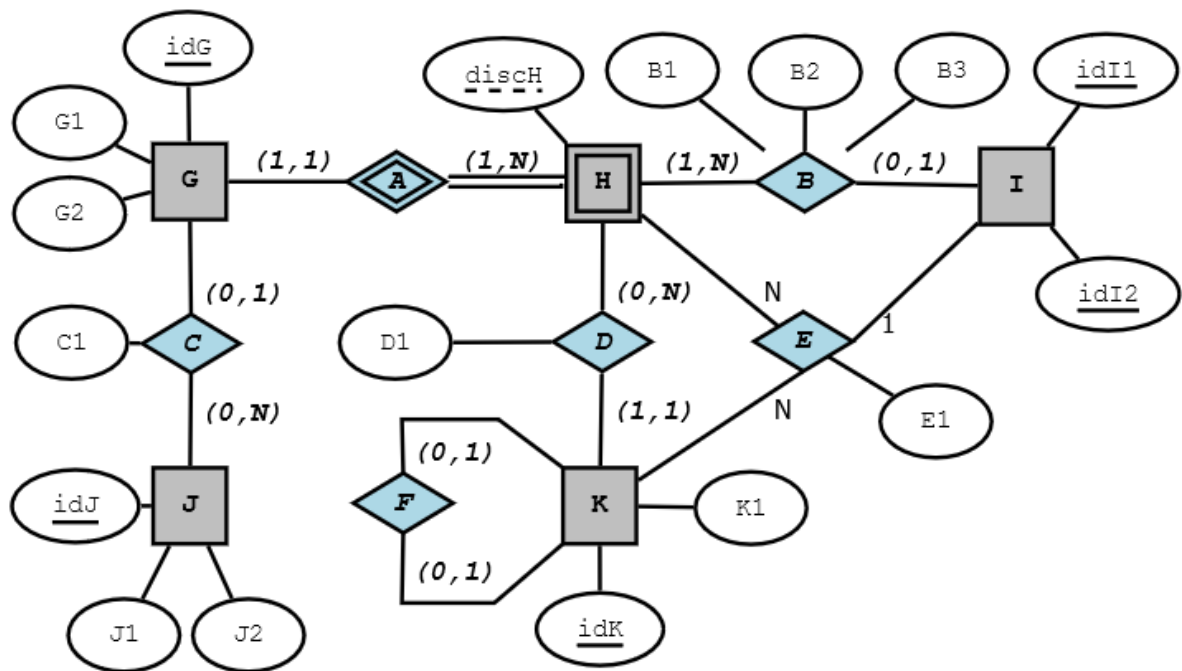
Respuesta:

Relación	CKs	PK	FKs
A(ida,a1,idf,idf')	ida	ida	idf ref F idf' ref F
C(ida,discc,c1)	(ida,discc)	(ida,discc)	ida ref A
F(idf,f1,f2)	idf	idf	
G(idg,g1)	idg	idg	
H(ida,discc,idg,idi,h1)	(ida,discc,idg) , (idg,idi)	(idg,idi)	(ida,discc) ref C ida ref A idg ref G idi ref I
I(idi,i1)	idi	idi	
J(idi,idi')	(idi,idi')	(idi,idi')	idi ref I idi' ref I

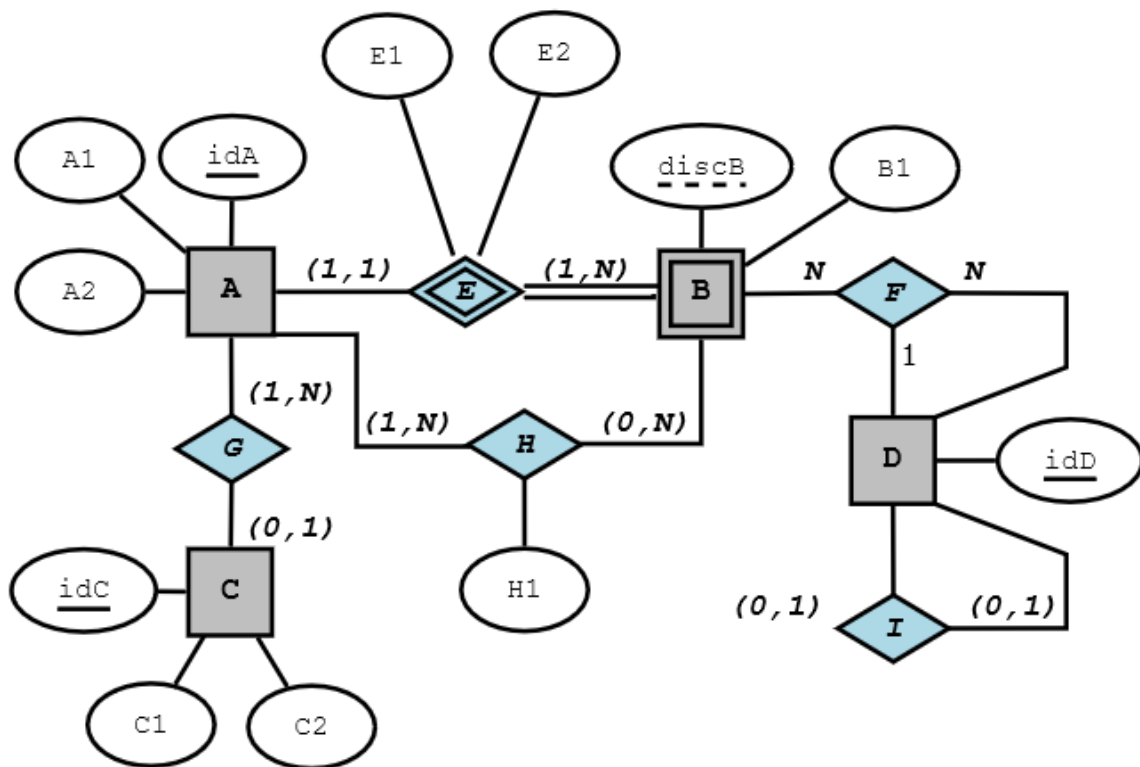


Respuesta:

Relación	CKs	PK	FKs
A(idc, discA, a1)	(idc, discA)	(idc, discA)	idc ref C
C(idc, c1)	idc	idc	
D(idc, discA, d1)	(idc, discA)	(idc, discA)	idc ref C (idc, discA) ref A
E(idc, discA, e1, e2)	(idc, discA)	(idc, discA)	idc ref C (idc, discA) ref A
F(idc, discA, idg, f1)	(idc, discA), (idg)	(idg)	idg ref G idc ref C (idc, discA) ref A
G(idg, g1)	idg	idg	
H(idc, discA, idc', discA')	(idc, discA, idc', discA')	(idc, discA, idc', discA')	(idc, discA) ref A (idc', discA') ref A

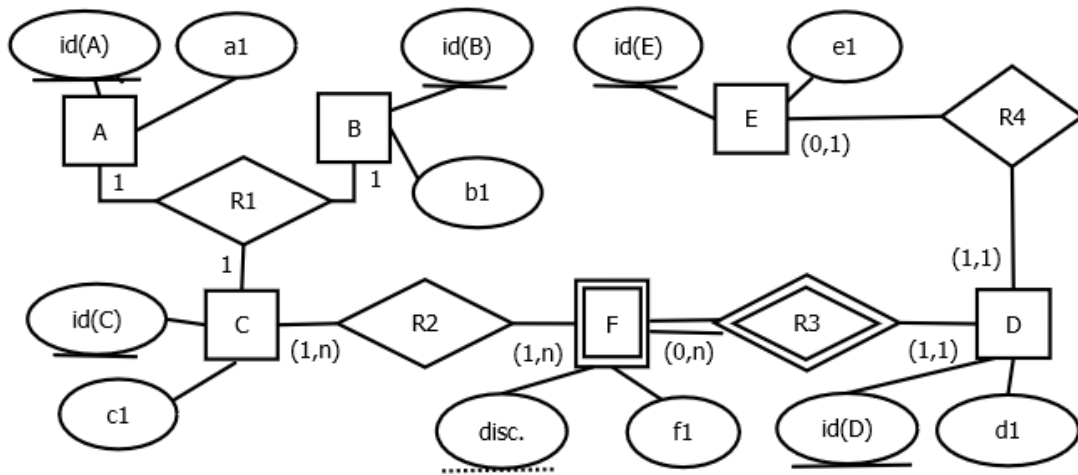
**Respuesta:**

Relación	CKs	PK	FKs
B(idg,disch,idi1,idi2,b1,b2,b3)	(idg,disch)	(idg,disch)	(idg,disch) ref H idg ref G (idi1,idi2) ref I
C(idg,idj,c1)	idj	idj	idg ref G idj ref J
E(idg,disch,idi1,idi2,idk,e1)	(idg,disch,idk)	(idg,disch,idk)	(idg,disch) ref H idg ref G idk ref K
F(idk,idk')	idk o idk'	idk	idk ref K idk' ref K
G(idg,g1,g2)	idg	idg	
H(idg,disch, idk, d1)	(idg,disch)	(idg,disch)	idg ref G idk ref K
I(idi1,idi2)	(idi1,idi2)	(idi1,idi2)	
J(idj,j1,j2)	idj	idj	
K(idk,k1)	idk	idk	

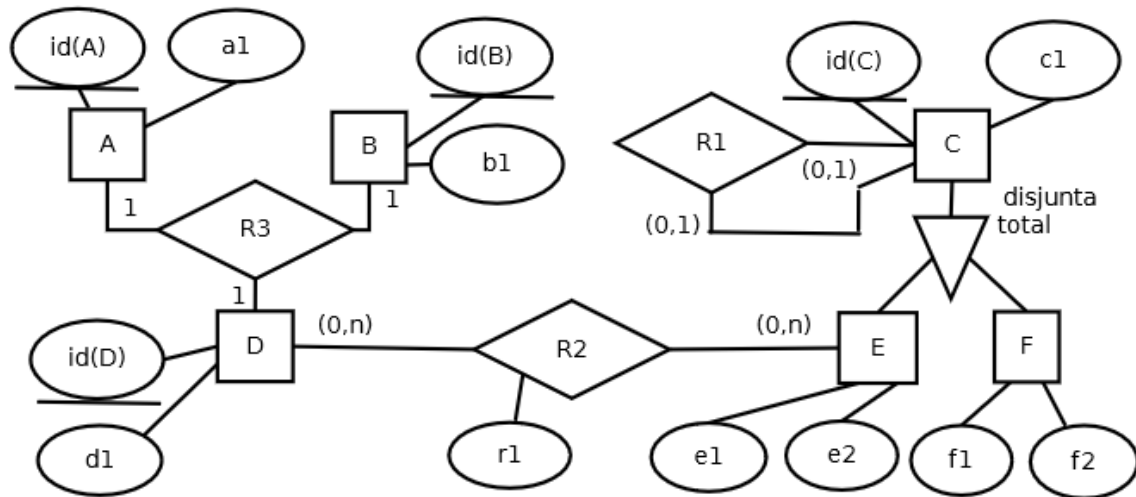


Respuesta:

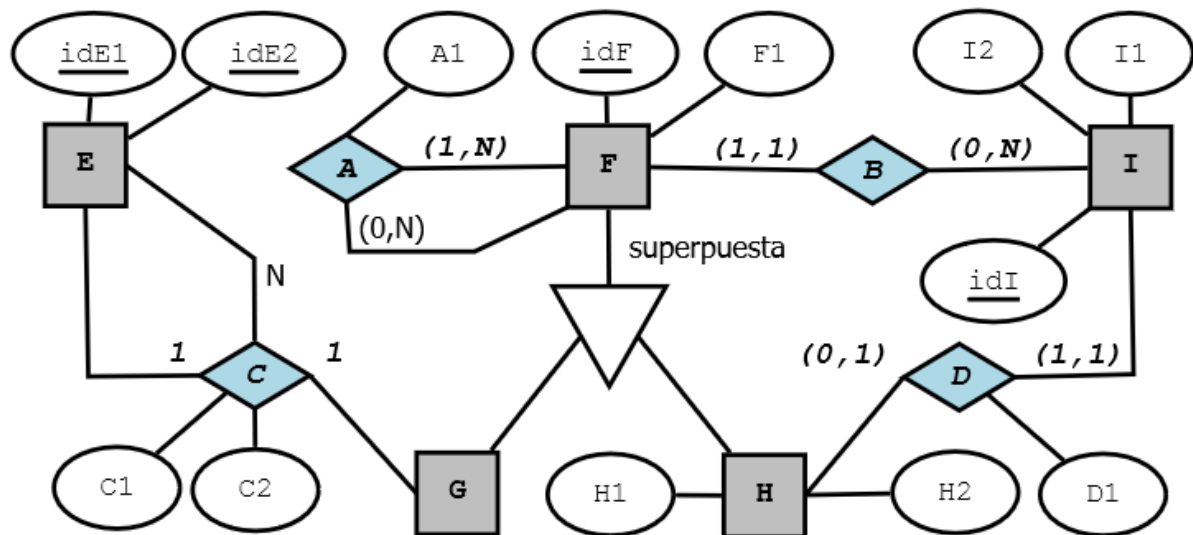
Relación	CKs	PK	FKs
A(ida,a1,a2)	ida	ida	
B(ida,discb,b1,e1,e2)	(ida,discb)	(ida,discb)	ida ref A
C(idc,c1,c2)	idc	idc	
D(idd)	idd	idd	
F(ida,discb,idd,idd')	(ida,discb,idd)	(ida,discb,idd)	(ida,discb) ref B ida ref A idd ref D idd' ref D
G(ida,idc)	ida	ida	ida ref A idc ref C
H(ida,ida',discb,h1)	(ida,ida',discb)	(ida,ida',discb)	ida ref A (ida',discb) ref B ida' ref A
I(idd,idd')	idd, idd'	idd	idd ref D idd' ref D

**Respuesta:**

Relación	CKs	PK	FKs
A(ida,a1)	ida	ida	
B(idb,b1)	idb	idb	
C(idc,c1)	idc	idc	
D(idd,d1)	idd	idd	
E(ide,e1)	ide	ide	
F(idd,discF,f1)	(idd,discF)	(idd,discF)	idd ref D
R1(ida,idb,idc)	(ida,idb), (ida,idc), (idb,idc)	(ida,idb)	ida ref A idb ref B idc ref C
R2(idc,idd,discF)	(idc,idd,discF)	(idc,idd,discF)	idc ref C idd ref D (idd,discF) ref F
R4(idd,ide)	idd	idd	idd ref D ide ref E

**Respuesta:**

Relación	CKs	PK	FKs
A(ida,a1)	ida	ida	
B(idb,b1)	idb	idb	
C(idc,c1)	idc	idc	
D(idd,d1)	idd	idd	
E(idc,e1,e2)	idc	idc	idc ref C
F(idc,f1,f2)	idc	idc	idc ref C
R1(idc,idc')	idc o idc'	idc	idc ref C idc' ref C
R2(idd,idc,r1)	(idd,idc)	(idd,idc)	idd ref D idc ref C
R3(ida,idb,idd)	(ida,idb), (ida,idd), (idb,idd)	(ida,idb)	ida ref A idb ref B idd ref D

**Respuesta:**

Relación	CKs	PK	FKs
A(idf,idf',a1)	(idf,idf')	(idf,idf')	idf ref F idf' ref F
C(ide1,ide2,ide1',ide2',idf,c1,c2)	(ide1,ide2,ide1',ide2'), (ide1,ide2,idf)	(ide1,ide2,idf)	(ide1,ide2) ref E (ide1',ide2') ref E idf ref F
D(idi,idf,d1)	idi o idh	idi	idi ref I idf ref F
E(ide1,ide2)	(ide1,ide2)	(ide1,ide2)	
F(idf,f1)	idf	idf	
G(idf)	idf	idf	idf ref F
H(idf,h1,h2)	idf	idf	idf ref F
I(idi,i1,i2,idf)	idi	idi	

1. (Diseño relacional)

a. Sea la relación $R(A, B, C, D, E, G, H)$ con el siguiente conjunto de dependencias funcionales $F = \{AB \rightarrow G, AC \rightarrow DH, BD \rightarrow AC, CGH \rightarrow E, CD \rightarrow AB\}$.

i. Encuentre todas las claves candidatas, utilizando el algoritmo correspondiente.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow A, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

Luego calculo las clausuras de todos los atributos que luego me será útil:

$$A^+ = \{A\}$$

$$B^+ = \{B\}$$

$$C^+ = \{C\}$$

$$D^+ = \{D\}$$

$$E^+ = \{E\}$$

$$G^+ = \{G\}$$

$$H^+ = \{H\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Se ve claramente que con las clausuras de cada atributo no hay implicantes redundantes.

$$F = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow A, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

Ahora eliminó aquellas dfs que sean redundantes:

$$AB^+ \text{ sin } AB \rightarrow G = \{A, B\} \text{ no es redundante}$$

$$AC^+ \text{ sin } AC \rightarrow D = \{A, C, H\} \text{ no es redundante}$$

$$AC^+ \text{ sin } AC \rightarrow H = \{A, B, C, D, G\} \text{ no es redundante}$$

$$BD^+ \text{ sin } BD \rightarrow A = \{A, B, C, D, G, H, E\} \text{ es redundante}$$

$$F = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

$$BD^+ \text{ sin } BD \rightarrow C = \{B, D\} \text{ no es redundante}$$

$$CGH^+ \text{ sin } CGH \rightarrow E = \{C, G, H\} \text{ no es redundante}$$

$$CD^+ \text{ sin } CD \rightarrow A = \{B, C, D\} \text{ no es redundante}$$

$$CD^+ \text{ sin } CD \rightarrow B = \{A, C, D, H\} \text{ no es redundante}$$

Entonces resulta ser que F_{min} :

$$F = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe

un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E, G, H\}$$

Ahora veo si existe atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$$K = \{\}$$

Veo aquellos atributos que sean implicantes y implicados:

$$A_{id} = \{A, B, C, D, G, H\}$$

Empiezo a probar combinaciones para encontrar claves candidatas:

$$F_{min} = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

$$AB^+ = \{A, B, G\} \neq C_a \text{ no es clave candidata}$$

$$AC^+ = \{A, B, C, D, H, G, E\} = C_a \text{ es clave candidata}$$

....

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$$F_{min} = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$$

$$CK_s = \{AC, BD, CD\}$$

ii. ¿Cuál es la máxima forma normal en que se encuentra R? Justifique.

Respuesta:

Del ejercicio anterior tenemos que:

$F_{min} = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$

$CKs = \{AC, BD, CD\}$

Atributos primos: A,B,C,D

Atributos no primos: G,E,H

Desde ya que se encuentra en 1FN porque se supone que los atributos son monovaluados.

No se encuentra en FNBC puesto que la primera dfs no cumple con que el lado izquierdo es una superclave.

No se encuentra en 3FN ya que la primera dfs el lado izquierdo no es una superclave ni el lado derecho son atributos primos.

No se encuentra en 2FN ya que la primera dfs el atributo G que resulta ser no primo depende parcialmente de una clave candidata.

$F_{min} = \{AB \rightarrow G, AC \rightarrow D, AC \rightarrow H, BD \rightarrow C, CGH \rightarrow E, CD \rightarrow A, CD \rightarrow B\}$

$CKs = \{AC, BD, CD\}$

Agarro la df $AB \rightarrow G$ que evita que R esté en FNBC:

$R1(A,B,G) \quad F1 = \{AB \rightarrow G\} \quad CKs = \{AB\}$ está en FNBC

$R2(A,B,C,D,E,H)$ no terminado

- b. Sea la relación $R(A, B, C, D, E, F, G)$ con el siguiente conjunto de dependencias funcionales $F = \{AB \rightarrow C, A \rightarrow DE, B \rightarrow F, E \rightarrow G\}$.
- i. Encuentre todas las claves candidatas, utilizando el algoritmo correspondiente.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$$

Luego calculo las clausuras de todos los atributos que luego me serán útiles:

$$A^+ = \{A, D, E, G\}$$

$$B^+ = \{B, F\}$$

$$C^+ = \{C\}$$

$$D^+ = \{D\}$$

$$E^+ = \{E, G\}$$

$$F^+ = \{F\}$$

$$G^+ = \{G\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Tenemos una única df que tiene más de un implicante $AB \rightarrow C$, y vemos que C no está en A^+ ni en B^+ por lo que ningún atributo es redundante.

$$F = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$$

Ahora eliminó aquellas dfs que sean redundantes:

$$AB^+ \text{ sin } AB \rightarrow C = \{A, B, D, E, F, G\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow D = \{A, E, G\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow E = \{A, D\} \text{ no es redundante}$$

$$B^+ \text{ sin } B \rightarrow F = \{B\} \text{ no es redundante}$$

$$E^+ \text{ sin } E \rightarrow G = \{E\} \text{ no es redundante}$$

Entonces resulta ser que F_{min} :

$$F_{min} = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E, F, G\}$$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$K = \{A, B\}$

$A+ = \{A, D, E, G\}$ no es clave

$B+ = \{B, F\}$ no es clave

$AB+ = \{A, B, C, D, E, F, G\}$ es clave

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$F_{min} = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$

$CKs = \{AB\}$

ii. ¿Cuál es la máxima forma normal en que se encuentra R? Justifique.

Respuesta:

Del ejercicio anterior tenemos que:

$F_{min} = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$

$CKs = \{AB\}$

Atributos primos: A, B

Atributos no primos: C, D, E, F, G

Desde ya que se encuentra en 1FN porque se supone que los atributos son monovaluados.

No se encuentra en FNBC puesto que la segunda dfs no cumple con que el lado izquierdo es una superclave.

No se encuentra en 3FN ya que la última dfs el lado izquierdo no es una superclave ni el lado derecho son atributos primos.

No se encuentra en 2FN ya que la dfs $B \rightarrow F$ el atributo F que resulta ser no primo depende parcialmente de una clave.

iii. Si R no se encuentra en FNBC, descompóngala en FNBC utilizando el algoritmo adecuado. Al finalizar, si hubo alguna pérdida de dependencia funcional, indíquela.

Respuesta:

Del ejercicio anterior tenemos que:

$F_{min} = \{AB \rightarrow C, A \rightarrow D, A \rightarrow E, B \rightarrow F, E \rightarrow G\}$

$CKs = \{AB\}$

Agarro la df $A \rightarrow D$ que evita que R esté en FNBC:

$A^+ = \{A, D, E, G\}$

$R1(A, D, E, G) \quad F1 = \{A \rightarrow D, A \rightarrow E, E \rightarrow G\} \quad CKs = \{A\}$ no está en FNBC

$R2(A, B, C, F) \quad F2 = \{AB \rightarrow C, B \rightarrow F\} \quad CKs = \{AB\}$ no está en FNBC

Agarro la df $E \rightarrow G$ que evita que R1 esté en FNBC:

$E^+ = \{E, G\}$

$R11(E, G) \quad F11 = \{E \rightarrow G\} \quad CKs = \{E\}$ está en FNBC

$R12(A, D, E) \quad F12 = \{A \rightarrow D, A \rightarrow E\} \quad CKs = \{A\}$ está en FNBC

Agarro la df $B \rightarrow F$ que evita que R2 esté en FNBC:

$B^+ = \{B, F\}$

$R21(B, F) \quad F21 = \{B \rightarrow F\} \quad CKs = \{B\}$ está en FNBC

$R22(A, B, C) \quad F2 = \{AB \rightarrow C\} \quad CKs = \{AB\}$ está en FNBC

Entonces nos queda:

$R1(E, G) \quad F11 = \{E \rightarrow G\} \quad CKs = \{E\}$

$R2(A, D, E) \quad F12 = \{A \rightarrow D, A \rightarrow E\} \quad CKs = \{A\}$

$R3(B, F) \quad F21 = \{B \rightarrow F\} \quad CKs = \{B\}$

$R4(A, B, C) \quad F2 = \{AB \rightarrow C\} \quad CKs = \{AB\}$

- c. Sea la relación $R(A, B, C, D, E)$ con el siguiente conjunto de dependencias funcionales $F = \{CDE \rightarrow A, A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$.

- i. Encuentre todas las claves candidatas, utilizando el algoritmo correspondiente.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{CDE \rightarrow A, A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$$

Luego calculo las clausuras de todos los atributos que luego me serán útiles:

$$A^+ = \{A, B\}$$

$$B^+ = \{B\}$$

$$C^+ = \{C, D\}$$

$$D^+ = \{D\}$$

$$E^+ = \{E\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$CDE \rightarrow A$, ni C ni D ni E contienen al atributo A, por lo tanto ninguno es redundante

$DE \rightarrow A$, ni D ni E contienen al atributo A, por lo tanto ninguno es redundante

$$F = \{CDE \rightarrow A, A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$$

Ahora eliminó aquellas dfs que sean redundantes:

CDE^+ sin $CDE \rightarrow A = \{A, B, C, D, E\}$ es redundante

$$F = \{A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$$

A^+ sin $A \rightarrow B = \{A\}$ no es redundante

C^+ sin $C \rightarrow D = \{C\}$ no es redundante

DE^+ sin $DE \rightarrow A = \{D, E\}$ no es redundante

Entonces resulta ser que F_{min} :

$$F_{min} = \{A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E\}$$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$$K = \{C, E\}$$

$C^+ = \{C, D\}$ no es clave

$E^+ = \{E\}$ no es clave

$CE^+ = \{A, B, C, D, E\}$ es clave

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$F_{min} = \{A \rightarrow B, C \rightarrow D, DE \rightarrow A\}$

$CKs = \{CE\}$

Atributos primos: C,E

Atributos no primos: A,B,D

Desde ya que se encuentra en 1FN suponiendo que los atributos son monovaluados.

No se encuentra en 2FN ya que la segunda df el lado derecho es un atributo no primo y depende parcialmente de una clave candidata.

No se encuentra en 3FN porq no esta en 2FN.

No se encuentra en FNBC ya que la primera df el lado izquierdo no es una superclave

Descompongo F_{min} hasta que se encuentre en FNBC:

Tomo la df $A \rightarrow B$ que impide el FNBC:

$A^+ = \{A,B\}$

$R1(A,B) F1 = \{A \rightarrow B\} CKs = \{A\}$, se encuentra en FNBC

$R2(A,C,D,E) F2 = \{C \rightarrow D, DE \rightarrow A\} CKs = \{CE\}$, no se encuentra en FNBC

Tomo la df $C \rightarrow D$ que impide el FNBC

$C^+ = \{C,D\}$

$R11(C,D) F11 = \{C \rightarrow D\} CKs = \{C\}$, se encuentra en FNBC

$R2(A,C,E) F2 = \{CE \rightarrow A\} CKs = \{CE\}$, se encuentra en FNBC

d. Sea la relación $R(A, B, C, D, E)$ con el siguiente conjunto de dependencias funcionales $F = \{CE \rightarrow A, AB \rightarrow C, A \rightarrow D, A \rightarrow E\}$.

i. Encuentre todas las claves candidatas, utilizando el algoritmo correspondiente.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{CE \rightarrow A, AB \rightarrow C, A \rightarrow D, A \rightarrow E\}$$

Luego calculo las clausuras de todos los atributos que luego me serán útiles:

$$A^+ = \{A, D, E\}$$

$$B^+ = \{B\}$$

$$C^+ = \{C\}$$

$$D^+ = \{D\}$$

$$E^+ = \{E\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$CE \rightarrow A$, ni C ni E contienen al atributo A, por lo tanto ninguno es redundante
 $AB \rightarrow C$, ni A ni B contienen al atributo C, por lo tanto ninguno es redundante

$$F = \{CE \rightarrow A, AB \rightarrow C, A \rightarrow D, A \rightarrow E\}$$

Ahora eliminó aquellas dfs que sean redundantes:

$$CE^+ \text{ sin } CE \rightarrow A = \{C, E\} \text{ no es redundante}$$

$$AB^+ \text{ sin } AB \rightarrow C = \{A, B, D, E\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow D = \{A, E\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow E = \{A, D\} \text{ no es redundante}$$

Entonces resulta ser que F_{min} :

$$F_{min} = \{CE \rightarrow A, AB \rightarrow C, A \rightarrow D, A \rightarrow E\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E\}$$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$K = \{B\}$

$B^+ = \{B\}$ no es clave

Atributos que sean implicantes y implicados:

$Aid = \{A, C, E\}$ prubeo combinaciones con estos atributos

$AB^+ = \{A, B, C, D, E\} = Ca$, es clave candidata

$BC^+ = \{B, C\} \neq Ca$, no es clave

$BE^+ = \{B, E\} \neq Ca$, no es clave

$BCE^+ = \{A, B, C, D, E\} = Ca$ es clave candidata

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$Fmin = \{CE \rightarrow A, AB \rightarrow C, A \rightarrow D, A \rightarrow E\}$

$CKs = \{AB, BCE\}$

Atributos primos: A, B, C, E

Atributos no primos: D

Desde ya que se encuentra en 1FN suponiendo que los atributos son monovaluados.

No se encuentra en 2FN ya que la tercera/cuarta df el lado derecho es un atributo no primo y depende parcialmente de una clave candidata.

No se encuentra en 3FN porq no esta en 2FN.

No se encuentra en FNBC ya que la primera df el lado izquierdo no es una superclave

Descompongo $Fmin$ hasta que se encuentre en FNBC:

Tomo la df $CE \rightarrow A$ que impide el FNBC:

$CE^+ = \{A, C, E, D\}$

$R1(A, C, E, D) F1 = \{CE \rightarrow A, A \rightarrow D, A \rightarrow E\}$ $CKs = \{CE\}$, no esta en FNBC

$R2(B, C, E) F2 = \{\}$ $CKs = \{BCE\}$, se encuentra en FNBC

Tomo la df $A \rightarrow D$ que impide el FNBC

$A^+ = \{A, D, E\}$

$R11(A, D, E) F11 = \{A \rightarrow D, A \rightarrow E\}$ $CKs = \{A\}$, se encuentra en FNBC

$R12(A, C) F12 = \{\}$ $CKs = \{AC\}$, se encuentra en FNBC

- a) Para la información contenida en el siguiente formulario, diseñe un esquema relacional que se encuentre en Tercera Forma Normal (3FN) que la represente. Comience generando la *relación universal* que contiene a todos los atributos, luego identifique las dependencias funcionales y finalmente normalice aplicando el algoritmo correspondiente.

REPORTE DE COSTO DE PRODUCTOS

PROVEEDOR	PRODUCTO	COSTO	MARKUP	PRECIO	DEPT COD
21 – Very Veggie	4108 – tomates redondos	189	5 %	199	PR
32 – Fab Fruits	4081 – bananas	20	75 %	35	PR
32 – Fab Fruits	4027 – uvas	45	100 %	90	PR
32 – Fab Fruits	4851 – apio	100	100 %	200	PR
08 – Meats R Us	331100 – alas de pollo	50	300 %	150	BU
08 – Meats R Us	331105 – bife ancho	60	400 %	240	BU
08 – Meats R Us	332110 – pechugas de pollo	250	100 %	500	BU
10 – Jerry’s Juice	411100 – jugo de naranja	25	400 %	100	FR
10 – Jerry’s Juice	521101 – jugo de manzana	25	400 %	100	FR
45 – Icey Creams	866503 – helado de vainilla	250	100 %	500	FR
45 – Icey Creams	866504 – helado de chocolate	250	100 %	500	FR

Nota: Cada proveedor se identifica con su código (en la misma columna) y cada producto (que incluye su código en la misma columna) es provisto por un único proveedor. A su vez cada producto se almacena en un único departamento (DEPT COD).

Respuesta:

Mediante un instante dado de la tabla que se nos presenta podemos identificar algunas dependencias funcionales tales como:

NPR: número de proveedor

NP: número de producto

C: costo

M: markup

P: precio

DPT: código del departamento

$NP \rightarrow NPR, NP \rightarrow C, NP \rightarrow DPT$

$C, M \rightarrow P$

Entonces nos queda:

$F = \{NP \rightarrow NPR, NP \rightarrow C, NP \rightarrow DPT, CM \rightarrow P\}$ que ya resulta ser mínimo

La claves candidatas para esta F son:

$Cks = \{NPM\}$

Descompongo en 3FN:

$R1(NP, NPR) \quad NP \rightarrow NPR$

$R2(NP, C) \quad NP \rightarrow C$

$R3(NP, DPT) \quad NP \rightarrow DPT$

$R4(CM, P) \quad CM \rightarrow P$

$R5(NP, M)$

Junto $R1$ $R2$ y $R3$

$R1(NP, NPR, C, DPT) \quad NP \rightarrow NPR \quad NP \rightarrow C \quad NP \rightarrow DPT$

$R2(CM, P) \quad CM \rightarrow P$

$R3(NP, M)$

a) Sea la relación $R(A, B, C, D, E, G)$ con el siguiente conjunto de dependencias funcionales $F = \{A \rightarrow BD, C \rightarrow E\}$.

1) ¿Cuál es la máxima forma normal en que se encuentra R ?

2) Especifique con una instancia de R un posible caso de anomalía.

Nota: Una instancia puede ser descripta detallando en una tabla los valores de por lo menos dos filas, por ejemplo:

A	B	C	D	E	G
a1	b1	c1	d1	e1	g1
a2	b2	c2	d2	e2	g2

Tabla 3: Ejemplo de instancia de $R(A, B, C, D, E, G)$.

Explique en qué consiste la anomalía en la instancia de R que propuso.

3) Obtenga una descomposición de R en 3FN.

4) Verifique si la descomposición obtenida se encuentra en FNBC.

Respuesta:

1-

Desde ya se encuentra en 1FN suponiendo que todos los atributos son monovaluados.

Busco las claves candidatas para saber en que maxima forma normal se encuentra:

$F = \{A \rightarrow BD, C \rightarrow E\}$

$F_{min} = \{A \rightarrow B, A \rightarrow D, C \rightarrow E\}$

Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$A_i = \{G\} \quad C_a = \{A, B, C, D, E\}$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$K = \{A, C\}$

$A^+ = \{A, B, D\}$ no es clave

$C^+ = \{C, E\}$ no es clave

$AC^+ = \{A, B, D, C, E\}$ es clave candidata

Entonces las claves candidatas resultan ser:

$F_{min} = \{A \rightarrow B, A \rightarrow D, C \rightarrow E\}$

$CKs = \{ACG\}$

Atributos primos: A, C, G

Atributos no primos: B, D, E

No se encuentra en FNBC ya que la primera df el lado izquierdo no es una superclave

No se encuentra en 2FN ya que la primera df el lado derecho es un atributo no primo y por ende depende de una clave parcial. Entonces tampoco se encuentra en 3FN.

Descompongo R en 3FN:

$R_1(A, B) A \rightarrow B$

$R_2(A, D) A \rightarrow D$

$R_3(C, E) C \rightarrow E$

$R_4(A, C, G)$

Agrupo aquellas relaciones que están incluidas en otras:

$R_1(A, B, D) A \rightarrow B A \rightarrow D Cks = A$ está en FNBC

$R_3(C, E) C \rightarrow E Cks = C$ está en FNBC

$R_4(A, C, G) Cks = ACG$ está en FNBC

b) Considere la relación **PACIENTE-VISITA**(paciente, hospital, doctor), en la que un paciente puede ser atendido en distintos hospitales, pero para un determinado paciente sólo hay un doctor que lo atiende en el mismo. Adicionalmente, cada doctor trabaja en un único hospital.

- 1) ¿Cuál es la máxima forma normal en que se encuentra **PACIENTE-VISITA**?
- 2) ¿Qué anomalías se pueden producir en este esquema?
- 3) ¿Qué descomposición puede hacer para llevarlo a una forma normal superior?
- 4) ¿Qué consecuencias no deseadas tendría esta descomposición?

Respuesta:

P = paciente

H = hospital

D = doctor

Mediante el enunciado se encuentran las siguientes dependencias funcionales:

$PH \rightarrow D$

$D \rightarrow H$

1-

Desde ya se encuentra en 1FN suponiendo que todos los atributos son monovaluados.

Busco las claves candidatas para saber en que maxima forma normal se encuentra:

$F = \{PH \rightarrow D, D \rightarrow H\}$

$P^+ = \{P\}$

$H^+ = \{H\}$

$F_{min} = \{PH \rightarrow D, D \rightarrow H\}$

Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$A_i = \{\}$ $Ca = \{P,H,D\}$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$K = \{P\}$

$P^+ = \{P\}$ no es clave

Atributos que sean implicantes y implicados:

$A_{id} = \{H,D\}$

$PH^+ = \{P,H,D\}$ es clave candidata

$PD+ = \{P,D,H\}$ es clave candidata
(no sigo porque sino encontraría superclaves)

Entonces las claves candidatas resultan ser:

$F_{\min} = \{PH \rightarrow D, D \rightarrow H\}$

$CKs = \{PD, PH\}$

Atributos primos: P,D,H

Atributos no primos:

No se encuentra en FNBC ya que la segunda df el lado izquierdo no es una superclave

La relación se encuentra en 3FN.

Descompongo R en FNBC:

Tomo $D \rightarrow H$ que impide FNBC:

$D+ = \{D,H\}$

$R_1(D,H) F_1 = \{D \rightarrow H\} CKs = \{D\}$, está en FNBC

$R_2(P,D) F_2 = \{\} CKs = \{PD\}$, se encuentra en FNBC

b) Dada la relación $R(A,B,C,D,E)$ y el siguiente cubrimiento minimal de su conjunto de dependencias, $F_{\min} = \{E \rightarrow D, AD \rightarrow C, B \rightarrow A\}$:

- 1) Justifique por qué R se encuentra en Primera Forma Normal (1FN).
- 2) Normalice R hasta obtener una descomposición en Forma Normal Boyce-Codd (FNBC), aplicando el algoritmo correspondiente. Muestre la descomposición en pasos sucesivos en un árbol como el siguiente, indicando para cada subrelación obtenida su clave y la máxima forma normal en que se encuentra.

Respuesta:

Ya el enunciado nos dice que tenemos F_{\min} por ende buscamos las claves candidatas para saber en máxima forma normal se encuentra R:

$R(A,B,C,D,E)$

$F_{\min} = \{E \rightarrow D, AD \rightarrow C, B \rightarrow A\}$

Atributos que no aparezcan en F ya que serán parte de las claves candidatas:

$A = \{\} Ca = \{A,B,C,D,E\}$

Todos se encuentran en F_{\min} . Sigo buscando aquellos atributos equivalentes es decir que $X+ = Y+$ donde X e Y son equivalentes:

$A+ = \{A\}$

$B^+ = \{A, B\}$

$C^+ = \{C\}$

$D^+ = \{D\}$

$E^+ = \{E, D\}$

No hay atributos equivalentes. Sigo buscando aquellos atributos que solo son implicantes:

$Aid = \{B, E\}$

$B^+ = \{A, B\}$ no es clave

$E^+ = \{E, D\}$ no es clave

$BE^+ = \{A, B, C, D, E\}$ es clave

(no sigo probando ya que solo encontraría superclaves)

$R(A, B, C, D, E)$

$F_{min} = \{E \rightarrow D, AD \rightarrow C, B \rightarrow A\}$ $CKs = \{BE\}$

Atributos primos : B, E

Atributos no primos : A, C, D

Desde ya que se encuentra en 1FN debido a que los atributos son monovaluados.

No se encuentra en 2FN ya que en la primera df el lado derecho es un atributo no primo y depende de una clave parcial.

No se encuentra en 3FN ya que no lo está en 2FN.

No se encuentra en FNBC ya que la primera df el lado izquierdo no es una superclave.

Descompongo R para llevarlo a FNBC:

Tomo $E \rightarrow D$ que impide el FNBC

$E^+ = \{E, D\}$

$R_1(E, D)$ $F_1 = \{E \rightarrow D\}$ $CKs = \{E\}$ está en FNBC

$R_2(A, B, C, E)$ $F_2 = \{AE \rightarrow C, B \rightarrow A\}$ no está en FNBC

Tomo $B \rightarrow A$ que impide el FNBC

$B^+ = \{A, B\}$

$R_{21}(B, A)$ $F_{21} = \{B \rightarrow A\}$ $CKs = \{B\}$ está en FNBC

$R_{22}(B, C, E)$ $F_{22} = \{BE \rightarrow C\}$ $CKs = \{BE\}$ está en FNBC

- a) Indique la forma normal del siguiente esquema relacional y lleve el mismo a una forma normal más alta si fuera posible:

VIAJE(idRuta, horaSalida, fecha, matricula, DNichofer, ApellNombre)

Considere que el esquema tiene asociadas las siguientes dos dependencias funcionales:

$DF_1 : \{idRuta, horaSalida\} \rightarrow \{matricula, DNichofer\}$

$DF_2 : \{DNichofer\} \rightarrow \{ApellNombre\}$

Respuesta:

Para saber en máxima forma normal se encuentra la relación debemos buscar las claves candidatas:

$F = \{idruta, horasalida \rightarrow matricula, dnichofer, dnichofer \rightarrow apellnombre\}$

Vemos si tenemos Fmin:

Separo el lado izquierda de las dfs para que quede un único atributo:

$F = \{idruta, horasalida \rightarrow matricula, idruta, horasalida \rightarrow dnichofer, dnichofer \rightarrow apellnombre\}$

Veó si hay algún atributo del lado derecho que sea redundante:

$idruta^+ = \{idruta\}$

$horasalida^+ = \{horasalida\}$

no contiene a matrícula entonces ninguno es redundante, lo mismo ocurre con dnichofer.

Entonces deducimos que ninguna dependencia funcional es redundante (falta hacer la demostración) por lo que:

$Fmin = \{idruta, horasalida \rightarrow matricula, idruta, horasalida \rightarrow dnichofer, dnichofer \rightarrow apellnombre\}$

Ahora busco las claves candidatas:

Aquellos atributos que no aparezcan en ninguna df ya que serán parte de las claves candidatas:

$A = \{fecha\}$ $Ca = \{idruta, horasalida, matricula, dnichofer, apellnombre\}$

Aquellos atributos equivalentes, no hay ya que $X^+ \neq Y^+$ para cualquier par de atributos.

Aquellos atributos que sean solo implicantes:

$Ai = \{idruta, horasalida\}$

$idruta^+ = \{idruta\}$

$horasalida^+ = \{horasalida\}$

$idruta, horasalida^+ = \{idruta, horasalida, matricula, dnichofer, apellnombre\}$ es clave

(no sigo buscando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$Cks = \{idruta, horasalida, fecha\}$

Atributos primos: $idruta, horasalida, fecha$

Atributos no primos: $matricula, dnichofer, apellnombre$

$Fmin = \{idruta, horasalida \rightarrow matricula, idruta, horasalida \rightarrow dnichofer, dnichofer \rightarrow apellnombre\}$

Desde ya se encuentra en 1FN ya que se supone que los atributos son monovaluados.

No se encuentra en 2FN ya que la primera df el lado derecho es un atributo no primo y depende de una clave parcial. Por ende tampoco está en 3FN.

No se encuentra en FNBC ya que la primera df el lado izquierdo no es una superclave.

- b) Sea el esquema de relación $R(A, B, C, D, E, F)$, el conjunto de dependencias funcionales asociado $DF = \{AB \rightarrow C; C \rightarrow B; A \rightarrow D; D \rightarrow A; EF \rightarrow C; C \rightarrow F\}$ y la descomposición $\{D_1(A, B, C), D_2(A, D), D_3(E, F, C)\}$

Se pide:

- 1) Halle todas las claves candidatas de R . Justifique.
- 2) La descomposición $\{D_1, D_2, D_3\}$, ¿preserva la información? Justifique.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{AB \rightarrow C, C \rightarrow B, A \rightarrow D, D \rightarrow A, EF \rightarrow C, C \rightarrow F\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$$A^+ = \{A, D\}$$

$$B^+ = \{B\}$$

$$C^+ = \{B, C, F\}$$

$$D^+ = \{A, D\}$$

$$E^+ = \{E\}$$

$$F^+ = \{F\}$$

$AB \rightarrow C$, ni A ni B contienen al atributo C, por lo tanto ninguno es redundante
 $EF \rightarrow C$, ni E ni F contienen al atributo C, por lo tanto ninguno es redundante

$$F = \{AB \rightarrow C, C \rightarrow B, A \rightarrow D, D \rightarrow A, EF \rightarrow C, C \rightarrow F\}$$

Ahora eliminó aquellas dfs que sean redundantes:

$$AB^+ \text{ sin } AB \rightarrow C = \{A, B, D\} \text{ no es redundante}$$

$$C^+ \text{ sin } C \rightarrow B = \{C, F\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow D = \{A\} \text{ no es redundante}$$

$$D^+ \text{ sin } D \rightarrow A = \{D\} \text{ no es redundante}$$

$$EF^+ \text{ sin } EF \rightarrow C = \{E, F\} \text{ no es redundante}$$

$$C^+ \text{ sin } C \rightarrow F = \{B, C\} \text{ no es redundante}$$

Entonces resulta ser que F_{min} :

$$F_{min} = \{AB \rightarrow C, C \rightarrow B, A \rightarrow D, D \rightarrow A, EF \rightarrow C, C \rightarrow F\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E, F\}$$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente existen atributos equivalentes: A y D por lo que ahora usaremos A y luego lo

reemplazamos por D

$F = \{AB \rightarrow C, C \rightarrow B, EF \rightarrow C, C \rightarrow F\}$

Veo aquellos atributos que sean solo implicantes:

$K = \{A, E\}$

$A^+ = \{A\}$ no es clave

$E^+ = \{E\}$ no es clave

$AE^+ = \{A, E\}$ no es clave

Atributos que sean implicantes y implicados:

$Aid = \{B, C, F\}$ prubeo combinaciones con estos atributos

$ABE^+ = \{A, B, C, E, F\} = Ca$, es clave candidata

$ACE^+ = \{A, B, C, E, F\} = Ca$, es clave candidata

$AEF^+ = \{A, B, C, E, F\} = Ca$, es clave candidata

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$Fmin = \{AB \rightarrow C, C \rightarrow B, A \rightarrow D, D \rightarrow A, EF \rightarrow C, C \rightarrow F\}$

$CKs = \{ABE, ACE, AEF, DBE, DCE, DEF\}$

- a) Halle el conjunto de claves candidatas de $R(A, B, C, D, E)$ dado el conjunto de dependencias funcionales $F = \{ACE \rightarrow B, AB \rightarrow D, BC \rightarrow AD, C \rightarrow ED, E \rightarrow C\}$. Analice en qué forma normal se encuentra R, justificando el análisis.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{ACE \rightarrow B, AB \rightarrow D, BC \rightarrow A, BC \rightarrow D, C \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$$A^+ = \{A\}$$

$$B^+ = \{B\}$$

$$C^+ = \{C, D, E\}$$

$$D^+ = \{D\}$$

$$E^+ = \{C, D, E\}$$

$$AE^+ = \{A, B, C, D, E\}$$

$$CE^+ = \{C, D, E\}$$

$$AC^+ = \{A, B, C, D, E\}$$

$ACE \rightarrow B$, C es redundante o E es redundante

$AB \rightarrow D$, ni A ni B contienen al atributo D, ninguno es redundante

$BC \rightarrow A$, ni B ni C contienen al atributo A, ninguno es redundante

$BC \rightarrow D$, ni B ni C contienen al atributo D, ninguno es redundante

$$F = \{AE \rightarrow B, AB \rightarrow D, BC \rightarrow A, BC \rightarrow D, C \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

Ahora eliminó aquellas dfs que sean redundantes:

AE^+ sin $AE \rightarrow B = \{A, D, E\}$ no es redundante

AB^+ sin $AB \rightarrow D = \{A, B\}$ no es redundante

BC^+ sin $BC \rightarrow A = \{B, C, D, E\}$ no es redundante

BC^+ sin $BC \rightarrow D = \{B, C, D, E\}$ es redundante

$$F = \{AE \rightarrow B, AB \rightarrow D, BC \rightarrow A, C \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

C^+ sin $C \rightarrow E = \{C, D\}$ no es redundante

C^+ sin $C \rightarrow D = \{C, E\}$ no es redundante

E^+ sin $E \rightarrow C = \{E\}$ no es redundante

Entonces resulta ser que F_{min} :

$$F_{min} = \{AE \rightarrow B, AB \rightarrow D, BC \rightarrow A, C \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves

candidatas:

$A_i = \{\}$ $Ca = \{A,B,C,D,E\}$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente existen atributos equivalentes: C y E por lo que ahora usaremos C y luego lo reemplazamos por E

$F_{min} = \{AC \rightarrow B, AB \rightarrow D, BC \rightarrow A, C \rightarrow D\}$

$Ca = \{A,B,C,D\}$

Veo aquellos atributos que sean solo implicantes:

$K = \{C\}$

$C^+ = \{C,D\}$ no es clave

Atributos que sean implicantes y implicados:

$Aid = \{A,B\}$ prubeo combinaciones con estos atributos

$AC^+ = \{A,B,C,D\} = Ca$, es clave candidata

$BC^+ = \{A,B,C,D\} = Ca$, es clave candidata

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$F_{min} = \{AE \rightarrow B, AB \rightarrow D, BC \rightarrow A, C \rightarrow E, C \rightarrow D, E \rightarrow C\}$

$CKs = \{AC, BC, AE, BE\}$

Atributos primos: A,B,C,E

Atributos no primos: D

Desde ya se encuentra en 1FN debido a que los atributos son monovaluados.

No se encuentra en 2FN ya que la segunda df el lado derecho es un atributo no primo y depende de una clave parcial.

Entonces su maxima forma normal es 1FN.

b) Dado el siguiente esquema relacional con sus dependencias funcionales y claves candidatas:

- $R(A, B, C, D, E)$
- $F = \{A \rightarrow C, AD \rightarrow E, B \rightarrow D, C \rightarrow A, E \rightarrow C\}$
- $CC : \{AB\}, \{BC\}, \{BE\}$

Aplicando el algoritmo correspondiente, encuentre una descomposición sin pérdida de información en FNBC. Detalle cada paso del algoritmo utilizado, indicando en cada relación R_i las claves candidatas y el F_i . La descomposición obtenida, ¿es sin pérdida de dependencias funcionales? Justifique.

Respuesta:

Ya el enunciado nos da F_{min} y sus claves candidatas.

Atributos primos: A,B,C,E

Atributos no primos: D

Descompongo R para llevarlo a FNBC:

Tomo $A \rightarrow C$ que impide el FNBC

$A^+ = \{A, C\}$

$R_1(A, C) F_1 = \{A \rightarrow C, C \rightarrow A\}$ CKs = $\{A, C\}$ esta en FNBC

$R_2(A, B, D, E) F_2 = \{AD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ no esta en FNBC

Tomo $AD \rightarrow E$ que impide el FNBC

$AD^+ = \{A, D, E\}$

$R_{21}(A, D, E) F_{21} = \{AD \rightarrow E, E \rightarrow A\}$ no esta en FNBC

$R_{22}(A, B, D) F_{22} = \{B \rightarrow D\}$ no esta en FNBC

Tomo $E \rightarrow A$ que impide el FNBC

$E^+ = \{A, E\}$

$R_{211}(A, E) F_{211} = \{E \rightarrow A\}$ CKs = $\{E\}$ esta en FNBC

$R_{22}(D, E) F_{22} = \{\}$ CKs = $\{AD\}$ esta en FNBC

Tomo $B \rightarrow D$ que impide el FNBC

$B^+ = \{B, D\}$

$R_{211}(B, D) F_{211} = \{B \rightarrow D\}$ CKs = $\{B\}$ esta en FNBC

$R_{22}(A, B) F_{22} = \{\}$ CKs = $\{AB\}$ esta en FNBC

$R_1(A, C)$

$R_2(A, E)$

$R_3(D, E)$

$R_4(B, D)$

$R_5(A, B)$

- a) Dado el esquema de relación $R(A, B, C, D, E, G, H)$ con su conjunto de dependencias funcionales $F = \{AB \rightarrow D, BD \rightarrow E, BE \rightarrow AH, EGH \rightarrow BC\}$, descompóngalo a 3FN si es que no se encuentra en ella. Justifique.

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{AB \rightarrow D, BD \rightarrow E, BE \rightarrow A, BE \rightarrow H, EGH \rightarrow B, EGH \rightarrow C\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$$A^+ = \{A\}$$

$$B^+ = \{B\}$$

$$D^+ = \{D\}$$

$$E^+ = \{E\}$$

$$G^+ = \{G\}$$

$$H^+ = \{H\}$$

$$EG^+ = \{E, G\}$$

$$EH^+ = \{E, H\}$$

$$HG^+ = \{G, H\}$$

$AB \rightarrow D$, ni A ni B contienen al atributo D, ninguno es redundante

$BD \rightarrow E$, ni B ni D contienen al atributo E, ninguno es redundante

$BD \rightarrow A$, ni B ni D contienen al atributo A, ninguno es redundante

$BD \rightarrow H$, ni B ni D contienen al atributo H, ninguno es redundante

$EGH \rightarrow B$, ni E ni G ni H contienen al atributo B, ninguno es redundante

$EGH \rightarrow C$, ni E ni G ni H contienen al atributo C, ninguno es redundante

$$F = \{AB \rightarrow D, BD \rightarrow E, BE \rightarrow A, BE \rightarrow H, EGH \rightarrow B, EGH \rightarrow C\}$$

Ahora eliminó aquellas dfs que sean redundantes:

AB^+ sin $AB \rightarrow D = \{A, B\}$ no es redundante

BD^+ sin $BD \rightarrow E = \{B, D\}$ no es redundante

BE^+ sin $BE \rightarrow A = \{B, E, H\}$ no es redundante

BE^+ sin $BE \rightarrow H = \{A, B, D, E\}$ no es redundante

EGH^+ sin $EGH \rightarrow B = \{C, E, G, H\}$ no es redundante

EGH^+ sin $EGH \rightarrow C = \{A, B, D, E, G, H\}$ no es redundante

Entonces resulta ser que F_{min} :

$$F_{min} = \{AB \rightarrow D, BD \rightarrow E, BE \rightarrow A, BE \rightarrow H, EGH \rightarrow B, EGH \rightarrow C\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E, G, H\}$$

Ahora veo si existen atributos equivalentes. Esto se ve claramente si $X^+ = Y^+$ entonces X e Y son atributos equivalentes. Por lo calculado previamente no existen atributos equivalentes.

Veo aquellos atributos que sean solo implicantes:

$K = \{G\}$

$G^+ = \{G\}$ no es clave

Atributos que sean implicantes y implicados:

$Aid = \{A,B,D,E,H\}$ prubeo combinaciones con estos atributos

$AG^+ = \{A,G\} \neq Ca$, no es clave candidata

$BG^+ = \{B,G\} \neq Ca$, no es clave candidata

...

$ABG^+ = \{A,B,C,D,E,G,H\} = Ca$, es clave candidata

$BDG^+ = \{A,B,C,D,E,G,H\} = Ca$, es clave candidata

$BEG^+ = \{A,B,C,D,E,G,H\} = Ca$, es clave candidata

$EGH^+ = \{A,B,C,D,E,G,H\} = Ca$, es clave candidata

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$Fmin = \{AB \rightarrow D, BD \rightarrow E, BE \rightarrow A, BE \rightarrow H, EGH \rightarrow B, EGH \rightarrow C\}$

$CKs = \{ABG, BDG, BEG, EGH\}$

La descomposicion ya se encuentra en 3FN debdi a que existe un unico atributo no primo C y en la unica dependencia funcional en la que aparece $EGH \rightarrow C$ el lado izquierdo es una superclave.

a) Halle todas las claves candidatas para el siguiente esquema relacional R con el conjunto de dependencias funcionales F :

- $R(A, B, C, D, E)$
- $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow AD\}$

Respuesta:

Primero debo encontrar F_{min} , para esto descompongo las dfs del lado derecho de manera que haya un solo implicado:

$$F = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A, E \rightarrow D\}$$

Ahora reduzco aquellas dependencias funcionales que tengan algún atributo implicante redundante. Por ejemplo :

$$A^+ = \{A, B, C, D, E\}$$

$$B^+ = \{B, D\}$$

$$C^+ = \{C\}$$

$$D^+ = \{D\}$$

$$E^+ = \{A, B, C, D, E\}$$

$CD \rightarrow E$, ni C ni D contienen al atributo E , ninguno es redundante

$$F = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A, E \rightarrow D\}$$

Ahora eliminó aquellas dfs que sean redundantes:

$$A^+ \text{ sin } A \rightarrow B = \{A, C\} \text{ no es redundante}$$

$$A^+ \text{ sin } A \rightarrow C = \{A, B, D\} \text{ no es redundante}$$

$$CD^+ \text{ sin } CD \rightarrow E = \{C, D\} \text{ no es redundante}$$

$$B^+ \text{ sin } B \rightarrow D = \{B\} \text{ no es redundante}$$

$$E^+ \text{ sin } E \rightarrow A = \{D, E\} \text{ no es redundante}$$

$$E^+ \text{ sin } E \rightarrow D = \{A, B, C, D, E\} \text{ es redundante}$$

Entonces resulta ser que F_{min} :

$$F_{min} = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$$

Teniendo F_{min} podemos calcular las claves candidatas. Primero vemos si existe un atributo que no aparezca en ninguna dfs ya que será parte de las claves candidatas:

$$A_i = \{\} \quad C_a = \{A, B, C, D, E\}$$

Veo aquellos atributos que sean solo implicantes:

$$K = \{\}$$

Atributos que sean implicantes y implicados:

$$A_{id} = \{A, B, C, D, E\} \text{ pruebo combinaciones con estos atributos}$$

$A+ = \{A, B, C, D, E\} = Ca$, es clave candidata

$E+ = \{A, B, C, D, E\} = Ca$, es clave candidata

$Aid = \{B, C, D\}$ pruebo combinaciones con estos atributos

$BC+ = \{A, B, C, D, E\} = Ca$, es clave candidata

$BD+ = \{B, D\} \neq Ca$, no es clave candidata

$CD+ = \{A, B, C, D, E\} = Ca$, es clave candidata

(no sigo probando ya que solo encontraría superclaves)

Entonces las claves candidatas resultan ser:

$Fmin = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

$CKs = \{A, E, BC, CD\}$
