

FUNDAMENTOS DE BASES DE DATOS

Examen Julio 2008

La duración del examen es de 3 horas y 30 minutos.

Presentar la resolución del examen:

- Con las hojas numeradas y escritas de un solo lado.
- Con la cantidad de hojas entregadas en la primer hoja.
- Con cédula de identidad y nombre en cada hoja.
- Escrita a lápiz y en forma prolija.
- Comenzando cada ejercicio en una nueva hoja

Ejercicio 1 (25 puntos).

Una librería de una facultad desea implementar un sistema de ventas de apuntes de asignaturas, por internet.

De cada apunte se conoce un título, la fecha de edición y el costo en pesos. Además se conocen los autores del documento y una única asignatura de la cual proviene. Se asume que el título del apunte es único para cada asignatura. De cada autor, se conoce el nombre, su email (que es único), el instituto en donde trabaja y un resumen de su currículum.

De cada asignatura, se conoce un código que la identifica, un nombre, y un conjunto de temas que se tratan en el programa. De cada tema se conoce un nombre y un área temática. Hay que tener en cuenta que pueden haber temas con igual nombre en distintas áreas temáticas (por ejemplo el nombre de tema "bridge" puede referirse al área temática "computación" o al área temática "ingeniería civil" y son cosas diferentes). Los temas constituyen una jerarquía en la que hay temas subordinados a temas principales, en donde cada tema subordinado puede tener un único tema principal. Además cada tema puede ser visto en diferentes asignaturas.

Un apunte también tiene asociado un conjunto de temas de los cuales, por lo menos uno tiene que ser un tema de la asignatura asociada al apunte.

Tanto los apuntes como las asignaturas tratan de por lo menos un tema.

Cada cliente debe registrarse antes de comprar y se le solicitan los siguientes datos:

- nombre
- dirección
- email (que es único)
- cedula de identidad
- Tarjetas de crédito del cliente. De cada tarjeta se conoce su identificador formado por marca y número, y la fecha de vencimiento de la misma.
- teléfono

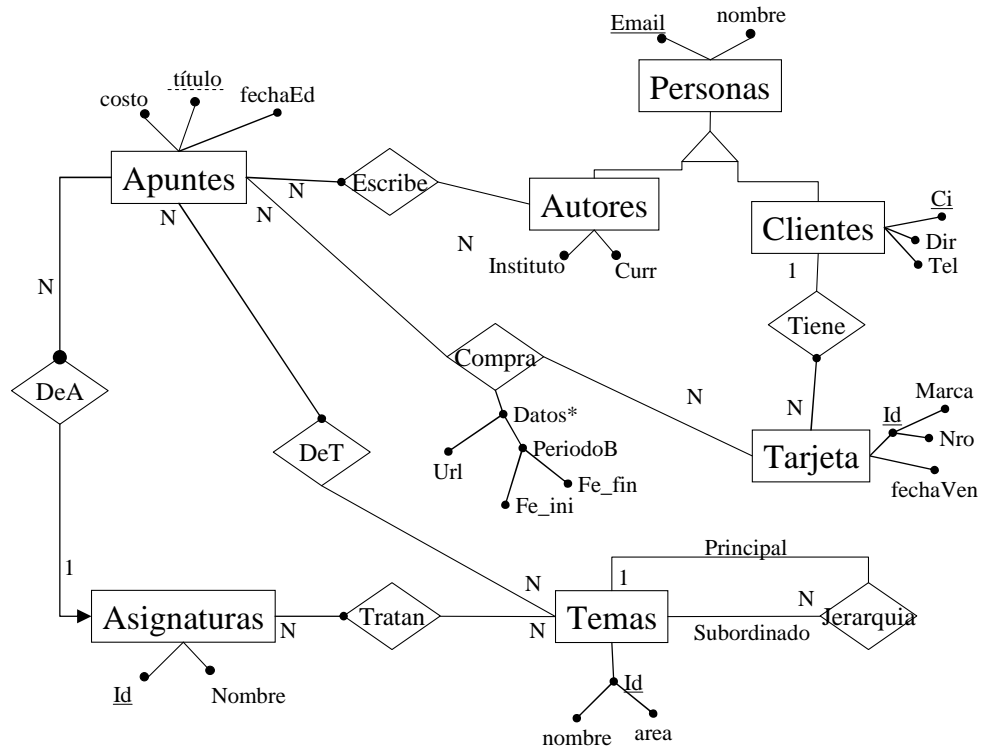
Una tarjeta de crédito tiene un único cliente asociado.

Para cada compra que se realiza se genera una url que depende del cliente y del apunte. Interesa registrar esa url y un "período de bajada" (fecha de inicio y fecha de fin) en que ese cliente puede acceder a ese apunte por esa url. Además interesa registrar con qué tarjeta se paga esa compra. Un cliente podría comprar un apunte varias veces.

Hay que considerar que algunos autores pueden ser clientes.

SE PIDE: Modelo Entidad Relación completo.

SOLUCION:



RNE:

$\forall a \in \text{Apuntes.}$

$\exists a' \in \text{Asignaturas.} (\langle a, a' \rangle \in \text{DeA} \wedge$

$\exists t \in \text{Temas.} (\langle a', t \rangle \in \text{Tratan} \wedge \langle a, t \rangle \in \text{DeT})$

)

Ejercicio 2 (30 puntos).

Una hemeroteca desea automatizar su mecanismo de clasificación de publicaciones. Para eso se definieron las siguientes tablas:

Ediciones-Publicacion(NomPub, Fecha, Tipo, nroEdicion).

Registro de todos los números de cada publicación con que cuenta la hemeroteca. El **tipo** puede ser: Diaria, Semanal, Quincenal, Mensual, Semestral, Anual, Aperiódica. **NroEdición** es el número de esa edición y **Fecha** la fecha en que salió ese número.

Artículos(IdArt, Pag, Título, NomPub, Fecha)

Registro de los artículos que salieron en cada número de cada publicación. **Pág** es la página de la edición en la cual salió el artículo, y (**NomPub**, **Fecha**) referencian a dicha edición.

Temas(IdTema, NomTema)

Definición de los temas que se usan para clasificar los artículos.

TemaArtículo(IdTema, IdArt)

Relaciona cada artículo con los temas que trata.

JerarquiaTemas(IdTPrinc, IdTSub)

Establece una jerarquía entre los temas que puede tratar un artículo.

En este esquema no hay tablas vacías y se cumplen las siguientes dependencias de inclusión.

- i. $\Pi_{\text{nomPub, fecha}}(\text{Artículos}) \subseteq \Pi_{\text{nomPub, fecha}}(\text{Ediciones-Publicacion})$
- ii. $\Pi_{\text{IdArt}}(\text{TemaArtículo}) \subseteq \Pi_{\text{IdArt}}(\text{Artículos})$
- iii. $\Pi_{\text{IdTema}}(\text{TemaArtículo}) \subseteq \Pi_{\text{IdTema}}(\text{Temas})$
- iv. $\Pi_{\text{IdTPrinc}}(\text{JerarquíaTemas}) \subseteq \Pi_{\text{IdTema}}(\text{Temas})$
- v. $\Pi_{\text{IdSub}}(\text{JerarquíaTemas}) \subseteq \Pi_{\text{IdTema}}(\text{Temas})$

Se pide resolver las siguientes consultas.

Algebra Relacional

1. Devolver el título y nombre de publicación de los artículos más viejos (con fecha mínima).

Solución:

$$FxF = \Pi_{\$5}(\text{Artículos}) \mid > < \mid_{\$1 < \$2} \Pi_{\$5}(\text{Artículos})$$

$$Fmin = FxF \ \% \ \Pi_{\$2}(FxF)$$

(En Fmin debería quedar la fecha mínima... Otra forma)

$$Fmin = \Pi_{\$1}(FxF) - \Pi_{\$2}(FxF)$$

(Ahora se hace el join con articulos asumiendo el nombre "fecha" para el atributo).

$$Res = \text{Artículos} * Fmin$$

2. Devolver el nombre de los temas que sólo se publicaron después de 1/1/2007.

Solución:

$$TConF = \Pi_{\text{IdTema, NomTema, Fecha}}(\text{Temas} * \text{TemaArtículo} * \text{Artículos})$$

$$Res = \Pi_{\text{NomTema}} \left(\begin{aligned} & \Pi_{\text{IdTema, NomTema}}(\sigma_{\text{fecha} > 1/1/2007}(TConF)) - \\ & \Pi_{\text{IdTema, NomTema}}(\sigma_{\text{fecha} \leq 1/1/2007}(TConF)) \end{aligned} \right)$$

Solución:

```
Select NomTema
from Temas
where not exists (
    select *
    from publicaciones p
    where p.nomPub="Linux Journal" ^
        not exist (
            select *
            from Articulos A, TemaArticulo Ta
            where A.idTema = Ta.idTema ^
                A.NomPub="Linux Journal" ^
                p.fecha=A.fecha
        )
    )
```

Ejercicio 3 (25 puntos).

Sea el esquema relación $R(A B C D E G H)$ con el conjunto de dependencias funcionales $F = \{ G \rightarrow AD, E \rightarrow ABG, A \rightarrow EG, G \rightarrow AB, BD \rightarrow GH, EG \rightarrow AH \}$.

Se pide:

- Hallar todas las claves, justificar la respuesta.

Solución:

C pertenece a toda clave pues no se genera a partir de otros (no aparece en ninguna dependencia funcional)

H no pertenece a ninguna clave, no aparece a la izquierda de ninguna df por lo cual no genera a nadie

$C^+ = \{ C \} \Rightarrow C$ no es clave

Calculo de claves con 2 atributos

$(CA)^+ = \{ ABCDEGH \} \Rightarrow CA$ es clave

Como tenemos las dependencias $E \rightarrow A$ y $G \rightarrow A$ sabemos que entonces que CE y CG tienen que ser claves ya que al incluir E (o G) en la clausura debemos incluir A, con lo que estaríamos calculando una clausura que incluye la de CA.

Esto nos deja para analizar sólo CB y CD

$(CB)^+ = \{ BC \}$

$(CD)^+ = \{ CD \}$

De donde vemos que ninguno de ellos es clave.

Calculo de claves con 3 atributos

$(CBD)^+ = \{ ABCDEGH \} \Rightarrow CBD$ es clave

No hay mas claves cualquier otra combinación de 3 o mas atributos debería contener a algunas de claves encontradas por lo cual sería superclave no minimal.

Todas las claves son: CA, CE, CG y CBD

- Hallar un cubrimiento minimal de F en R. Mostrar los pasos seguidos.

Paso 1: llevar todas las dfs a la forma $X \rightarrow A_i$

$F_1 = \{ G \rightarrow A, G \rightarrow D, E \rightarrow A, E \rightarrow B, E \rightarrow G, A \rightarrow E, A \rightarrow G, G \rightarrow B, BD \rightarrow G, BD \rightarrow H, EG \rightarrow A, EG \rightarrow H \}$

Paso 2: Eliminar atributos redundantes

$EG \rightarrow A$ tiene atributos redundantes

$EG \rightarrow H$ tiene atributos redundantes

$E_{+F_1} = \{ E, A, B, G, H, B, D \}$

$G_{+F_1} = \{ G, A, D, E, B, D, H \}$

$F_2 = \{ G \rightarrow A, G \rightarrow D, E \rightarrow A, E \rightarrow B, E \rightarrow G, A \rightarrow E, A \rightarrow G, G \rightarrow B, BD \rightarrow G, BD \rightarrow H, E \rightarrow H, G \rightarrow H \}$

Paso 3: Eliminar dfs redundantes

$E \rightarrow A$ es redundante

$E \rightarrow B$ es redundante

$A \rightarrow G$ es redundante

$BD \rightarrow H$ es redundante

$E \rightarrow H$ es redundante

$F_{min} = \{ G \rightarrow A, G \rightarrow D, E \rightarrow G, A \rightarrow E, G \rightarrow B, BD \rightarrow G, G \rightarrow H \}$

c) Dé una descomposición de R en dos tablas que tenga JSP. Justifique.

Solución:

El teorema del JSP nos garantiza que hay JSP si $R_1 \cap R_2 \rightarrow R_1 - R_2$ o $R_1 \cap R_2 \rightarrow R_2 - R_1$. Esto significa que hay que elegir R_1 y R_2 tal que alguna de las dependencias que se cumplen se pueda identificar como una de las dependencias del teorema.

Si tomamos la primer dependencia del minimal calculado anteriormente, hay que tomar

$R_1 \cap R_2 = \{G\}$ y por ejemplo, $R_1 - R_2 = \{A\}$.

Esto nos deja que una posible descomposición es:

$R_1(A G)$ y $R_2(B C D E G H)$

d) Llevar R a 3NF con JSP y preservación de dependencias.

Solución:

Partiendo del conjunto minimal se crea un único esquema relación con todas las dependencias de la forma $X \rightarrow A_i$ formando $R_i (X, A_i.A_j)$

$F_{min} = \{ G \rightarrow A, G \rightarrow D, E \rightarrow G, A \rightarrow E, G \rightarrow B, BD \rightarrow G, G \rightarrow H \}$

=> $R_1 (ABDGH)$

$R_2 (EG)$

$R_3 (AE)$

$R_4 (CA)$ se incluye porque ningún sub esquema tiene una clave

=> resumiendo la descomposición $\rho_{3NF} = (R_1, R_2, R_3, R_4)$

$F_{R_1} = \{ G \rightarrow A, G \rightarrow D, G \rightarrow B, BD \rightarrow G, G \rightarrow H \}$

claves G y BD

$F_{R_2} = \{ E \rightarrow G \}$

clave E

$F_{R_3} = \{ A \rightarrow E \}$

clave A

$F_{R_4} = \{ \}$

clave CA

e) Llevar la descomposición de la parte d a BCNF con JSP.

Solución:

La descomposición de la parte d) ya se encuentra en BCNF ya que para cada dependencia funcional $X \rightarrow A$ de cada subesquema X es una superclave.

Ejercicio 4 (20 puntos).

El siguiente es parte de un esquema de gestión de ventas de una empresa:

CLIENTES (idCliente, nomCliente, rubro)

Representa información de los clientes, su identificador, nombre y rubro en el que trabaja.

PRODUCTOS (idProducto, nomProducto, categProducto)

Representa información de los productos, su identificador, nombre y su categoría.

VENTAS (idCliente, idProducto, fecha, cantidad)

Registra las ventas de determinado producto a cada cliente en determinada fecha, así como la cantidad que se vendió.

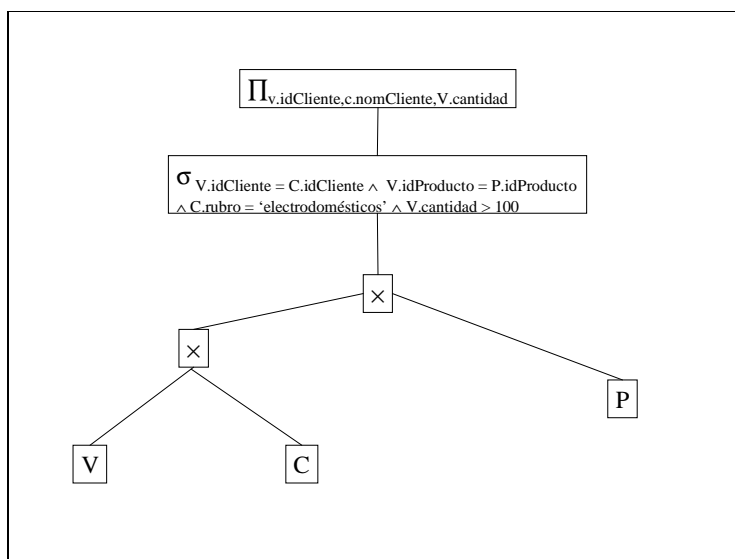
Considere la siguiente consulta sobre el esquema dado:

```
SELECT V.idCliente, C.nomCliente, V.cantidad
FROM Ventas V, Clientes C, Productos P
WHERE V.idCliente = C.idCliente AND
      V.idProducto = P.idProducto AND
      C.rubro = 'electrodomesticos' AND
      V.cantidad > 100
```

Se pide:

- a) Dar el árbol canónico del plan lógico para la consulta

Solución:



- b) Aplicar las heurísticas para optimización llegando al plan optimizado. (Explicar pasos aplicados).

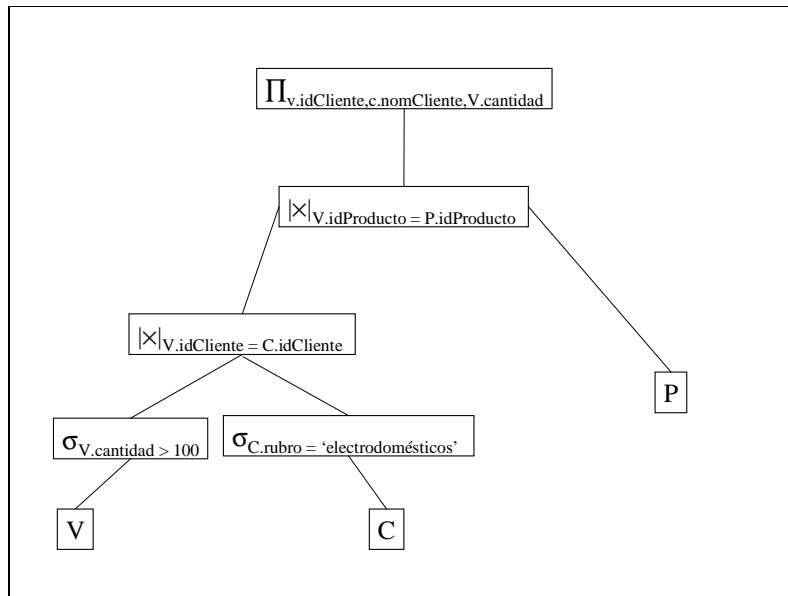
Solución:

Paso 1: Se separan las selecciones en una cascada.

Paso 2: se distribuyen lo más abajo posible.

Paso 3: se colapsan las secuencias de selecciones seguidas de productos cartesianos en joins.

De esta forma se obtiene el siguiente árbol optimizado:



- c) Teniendo en cuenta esta consulta y que son habituales las consultas por las claves, qué índices le parece útil tener para este esquema? Para cada uno, decir que tipo de índice sería y sobre que atributo/s se definiría.

Solución:

Dado que son habituales las consultas por las claves, es razonable suponer que existe un índice primario por la clave primaria en cada tabla. Esto inhibe el uso de índices de cluster ya que los dos exigen que los datos estén físicamente ordenados por los atributos involucrados en el índice.

Por otro lado, el join que aparece más abajo en el árbol, no tiene índices que se puedan explotar dado que se pierden en las selecciones. De esta forma, los índices que convienen tienen que ver con acelerar las selecciones: Un índice secundario como árbol-B+ sobre la cantidad en Ventas y un índice secundario como árbol-B+ o mejor aún, como hash sobre rubro en el cliente. La ventaja del árbol-B+ es que habilita realizar más tipos de consultas sobre el atributo que el hash.

- d) Considerando las respuestas de las partes a), b) y c), dar un plan físico que le parezca adecuado.

Solución:

Dado que se asumen los índices mencionados en la parte C, las selecciones se pueden realizar aplicando los índices.

El join más bajo (más cercano a las hojas) en el árbol sólo se puede resolver por métodos que no involucren índices, por ejemplo, loop anidado. Sin embargo, el join más cercano a la raíz puede explotar el índice de productos y resolverse mediante index join.