

CONTROL III: Bases de Datos

Prof. Andrea Rodríguez, Ayudante: Daniel Campos

13 de Diciembre del 2012

1. El Ministerio de Cultura de Chile desea mantener información acerca de las pinturas que se encuentran en los museos Chilenos, y toda la información relacionada al respecto. Para ello le ha encargado el diseño de una base de datos y le solicita el Modelo Entidad Relación y su traspaso al modelo relacional. De cada pinacoteca se desea saber el nombre, la ciudad en la que se encuentra, la dirección y los metros cuadrados que tiene. Cada pinacoteca tiene una serie de cuadros de los que se quiere guardar su código, nombre, medidas, fecha en que fue pintado y técnica utilizada para pintarlo. Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de fallecimiento). Los cuadros pueden tener reproducciones identificadas por el cuadro original y un número correlativo además de tener una fecha de creación. Los pintores pueden tener uno o varios mecenas que los protegen (nombre, país y ciudad de nacimiento, fecha de su fallecimiento, y fechas en que inicia y termina el mecenazgo). Se quiere consultar acerca de los cuadros en un museo, acerca de pintores, acerca de los mecenas de un pintor y el periodo (1 o varios) de su mecenazgo.(1.4)

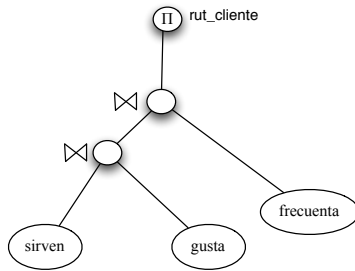
2. Sea el siguiente esquema de modelo relacional:

```
CLIENTE(rut_cliente(10), nombre_cliente(40), direccion_cliente (50))
BAR(rut_bar(10), nombre_bar (40), direccion_bar (50))
CERVEZA(id_cerveza (5), nombre_cerveza (20), caracteristica (25))
FRECUENTA(rut_cliente, rut_bar)
    FK rut_cliente FROM CLIENTE(rut_cliente)
    FK rut_bar FROM BAR(rut_bar)
SIRVEN(rut_bar, id_cerveza)
    FK rut_bar FROM BAR(rut_bar)
    FK id_cerveza FROM CERVEZA(id_cerveza)
GUSTA(rut_cliente, id_cerveza)
    FK rut_cliente FROM CLIENTE(rut_cliente)
    FK id_cerveza FROM CERVEZA(id_cerveza)
```

***Los valores entre () indican largo del campo.

Responda usando SQL las siguientes consultas (1):

- a) Encontrar el rut de las personas que van a bares que sirven alguna cerveza que les guste. (Asumir que cada persona frecuenta al menos un bar y le gusta al menos una cerveza).
- b) Encontrar los bares que ofrecen todas las cervezas.
3. Para la consulta (a) de la pregunta 2, determine el costo del siguiente plan de consulta (1):



Para ello considere la siguiente información:

- Tamaño página: 512
- Números de tuplas en CLIENTE, BAR, CERVEZA, FRECUENTA, SIRVEN y GUSTA es de 1000, 30, 20, 5000, 300 y 3000, respectivamente.
- Largo de registro de CLIENTE, BAR, CERVEZA, FRECUENTA, SIRVEN y GUSTA es de 100, 100, 50, 20, 15 y 15, respectivamente.
- Las tablas están indexadas por su clave primaria usando hash.
- En promedio, a las personas le gustan 3 cervezas y frecuentan 5 bares, los bares sirven 10 tipos de cervezas y las cervezas le gustan a 150.

Se adjunta la siguiente información por si les sirve.

	Barrido	Igualdad	Rango
Pila	BD	$0,5BD$	BD
Ordenado	BD	$D \log_2 B$	$D(\log_2 B + \# \text{ match pages})$
Arbol agrup.	$BD(R + 0,15)$	$D(1 + \log_F 0,15B)$	$D(\log_F 0,15B + \# \text{ match pages})$
Hash no agrup.	$BD(R + 0,125)$	$2D$	BD

	Insertar	Borrar
Pila	$2D$	$Search + D$
Ordenado	$Search + BD$	$Search + BD$
Arbol agrup.	$Search + 2D$	$Search + 2D$
Hash no agrup.	$Search + 2D$	$Search + 2D$

Los join más eficiente usan $3(N + M)$ (N el número de páginas en una de las relaciones y M el número en la otra) usando hash join y sort-merge join, asumiendo una cantidad mínima de memoria.

4. Suponga la existencia de un administrador de bases de datos (DBMS) que utiliza threads para procesar transacciones concurrentes que pueden ejecutar lecturas y escrituras en las tuplas Q de las tablas de la base de datos. El DBMS asigna a cada transacción T_i un valor único de timestamp $Ts(i)$ que permite establecer un orden entre las transacciones. La CPU se asigna de manera circular a las transacciones de manera que a cada transacción se le permite ejecutar dos operaciones read/write y las que sean necesarias de locks/unlocks (bloqueo/desbloqueo). El protocolo de timestamps hace lo siguiente. A cada tupla Q siendo requerida por las transacciones se asocian dos valores: $WTS(Q)$: Máximo de los timestamps de las transacciones que han hecho write(Q). $RTS(Q)$: Máximo de los timestamps de las transacciones que han hecho read(Q). El protocolo que asegura la correcta ejecución de las transacciones es el siguiente. Si T_i intenta ejecutar read(Q):
 - Si $Ts(i) < WTS(Q)$ no se ejecuta la operación y T_i se aborta.
 - Si $Ts(i) > WTS(Q)$ se ejecuta la operación y $RTS(Q) = \max\{RTS(Q), Ts(i)\}$.

Si T_i intenta un write(Q):

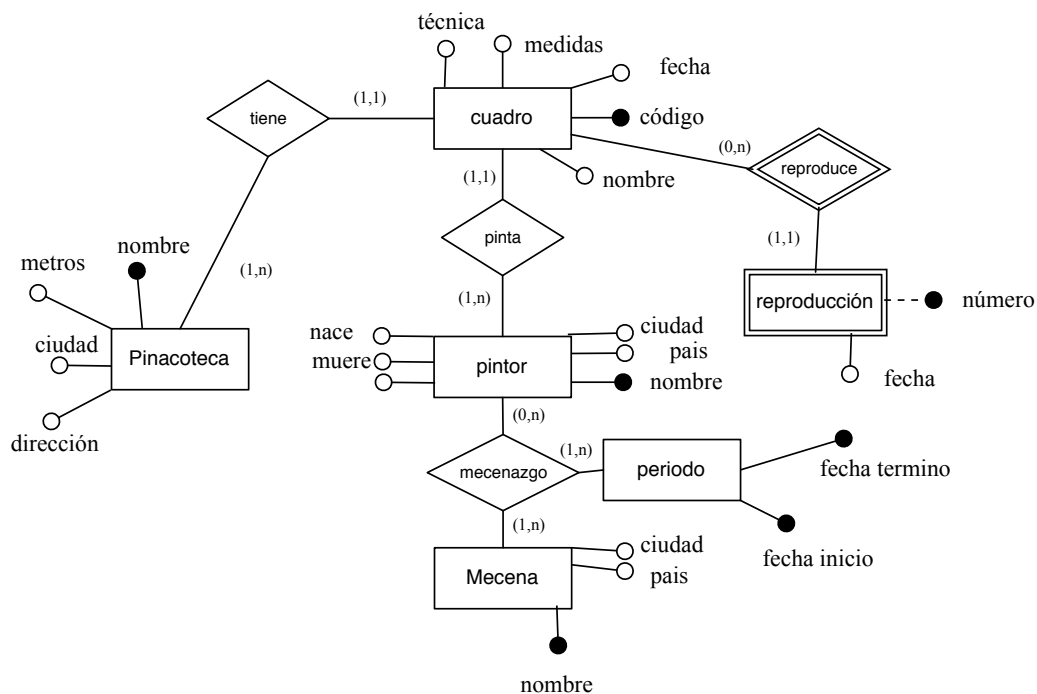
- Si $Ts(i) < RTS(Q)$ no se ejecuta la operación y T_i se aborta.
- Si $Ts(i) < WTS(Q)$ no se ejecuta la operación y T_i continúa.
- Si no, se ejecuta la operación y $WTS(Q) = Ts(i)$.

A partir del esquema de ejecución para las transacciones T_1 , T_2 y T_3 indicadas a continuación se pide llenar la tabla de valores para $WTS(Q)$ y $RTS(Q)$ e indicar cuáles transacciones abortan. Cuando una transacción se re-ejecuta se le asigna un nuevo timestamp. También se pide aplicar el protocolo de 2 fases estricto indicando en cada línea la operación ejecutada lock/unlock/read/write. (1)

T1	T2	T3
W(A) W(B)	W(C) R(A)	R(B) W(A)
R(C) W(C)	R(B) W(B)	

5. Conteste las siguientes preguntas justificando su respuesta: (1.6)
- Dada una instancia de una relación $R(A, B, C, D)$ con dependencia funcional $A \rightarrow B$, ¿Existe una instancia sin valores nulos con una tupla en esta relación que no satisfaga esta dependencia funcional?
 - Entregue un conjunto de dependencias multivaluadas para las cuales el esquema $R(\underline{A}, B, C, D)$ con claves primarias AB está en 3FN pero no en 4FN y luego descomponga.
 - ¿Por qué el procesamiento de consultas mapea una consulta SQL a una expresión del álgebra relacional?
 - Asuma que el sistema tiene un log con n registros de actualización y una tabla de transacciones con m transacciones. Indique el número máximo y mínimo de re escrituras en el REHACER y el número máximo y mínimo de operaciones de DESHACER del algoritmo ARIES para que el sistema se recupere. Indique los casos en que ocurren estos valores.

1. Un modelo entidad relación es:



El modelo relacional correspondiente es:

```

PINACOTECA(nombre:string,ciudad: string,dirección: string,metros: int)
CUADRO(código: int,nombre: string, técnica: string, fecha: date, ancho: int, alto: int, pintor: string
,pinacoteca: string)
    FK pintor FROM PINTOR(nombre)
    FK pinacoteca FROM PINACOTECA(nombre)
REPRODUCCION(código:int,numero:integer,fecha: date)
    FK código FROM CUADRO(código)
PINTOR(nombre:string,ciudad: string,pais: string,nace:date,muere:date)
MECENAnombre:string,ciudad: string,pais: string)
MECENAZGO(pintor:string,mecena:string, inicio:date, término:date)
    FK pintor FROM PINTOR(nombre)
    FK mecena FROM MECENA(nombre)
    
```

2. Las respuesta en SQL son:

- a) Encontrar las personas que van solamente a bares que sirven alguna cerveza que les guste.(Asumir que cada persona frecuenta al menos un bar y le gusta al menos una cerveza):

```
SELECT g.rut_cliente
FROM FROM SIRVEN as s, GUSTA as g, FRECUENTA as f
WHERE g.id_cerveza = s.id_cerveza AND s.ru_bar = f.rut_bar AND
      g.rut_cliente = f.rut_cliente
```

- b) Encontrar los bares que ofrecen todas las cervezas:

```
SELECT b.nombre
FROM BAR as b
WHERE not exists (
    SELECT id_cerveza FROM CERVEZA
    EXCEPT
    SELECT distinct id_cerveza FROM SIRVEN WHERE rut_bar = b.rut_bar
)
```

3. De los datos se tiene que el número de páginas de CLIENTE, BAR, CERVEZA, FRECUENTA, SIRVEN y GUSTA es 196, 6,2,196,9,88 páginas, respectivamente.

Nota: En el certamen se decía originalmente que el costo de join $3(N + M)$ donde N y M eran tuplas, pero son páginas. Se consideró respuesta por tuplas pero la pauta está por página.

Entonces, tenemos los siguientes costos

	Proceso	Output
(1): SIRVE <i>Join</i> GUSTA	$3(9+88)$	$2198 = \lceil 300 * 150 * 25 / 512 \rceil^*$
(2): (1) <i>Join</i> FRECUENTA	$3(2197+196)$	$4395 = \lceil 45000 * 5 * 10 / 512 \rceil^{**}$

* Por cada bar que sirve una cerveza (SIRVE) hay 150 aprox. personas que le gusta esa cerveza, por lo que hay $300 * 150 = 45000$ tuplas combinadas que ocupan 25 de largo (10 rut cliente, 10 rut de bar y 5 id de cerveza).

** Se considera la proyección como parte de la operación de salida del join.

4. Para timestamping:

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17
WTS(A)	0	1	1	1	1	1	3	3	4	4	4	4	4	4	6	6	6	6
RTS(A)	0	0	0	0	2	2	2	2	2	2	2	2	5	5	5	5	5	5
WTS(B)	0	0	1	1	1	1	1	1	1	4	4	4	4	5	5	6	6	6
RTS(B)	0	0	0	0	0	3	3	3	3	3	3	3	5	5	5	5	5	5
WTS(C)	0	0	2	2	2	2	2	2	2	5	5	5	5	5	6	6	6	6
RTS(C)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6

Después de t7, se reinicia T1, después de t8 se reinicia T2 y después de t11, vuelve a abortar t1.

El plan es el siguiente:

$$P = W_1(A), W_1(B), W_2(C), R_2(A), R_3(B), W_3(A), \text{commit } T3, \text{aborta } T1, R_2(B), \text{Aborta } T2, \\ W_1(A), W_1(B), W_2(C)R_2(A)\text{aborta } T1, R_2(B), W_2(B), W_1(A), W_1(B)R_1(C), W_1(C)$$

Para el protocolo de 2 fases estricto un plan es el siguiente:

$$P = X_2(C)W_2(C)S_3(B)R_3(B)X_3(A)W_3(A) \text{ commit } T3(\text{incluye desbloques}) \\ S_2(A)R_2(A)X_2(B)R_2(B)W_2(B) \text{ commit } T2 \\ X_1(A)W_1(A)X_1(B)W_1(B)X_1(C)R_1(C)W_1(C) \text{ commit } T1$$

5.
 - a) Todo instancia con una tupla la satisface.
 - b) Se puede considerar $C \rightarrow D$, lo que implica que $C \rightarrow \rightarrow AB$, no estando en 4FN. Una descomposición factibles es $R_1(\underline{A}, B, C)$ y $R_1(\underline{A}, B, D)$.
 - c) Porque el álgebra relacional indica cómo resolver la consulta. El SQL indica qué se quiere pero no cómo se quiere obtener.
 - d) En el rehacer hay como mínimo 0 (todas las operaciones son de transacciones que han sido confirmadas) y máximo n (todas son operaciones de transacciones sin confirmar), y en el deshacer mínimo 0 (todas las transacciones están confirmadas) y máximo n (todas las transacciones están activas y ejecutan las n operaciones del log).