Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre de 2016

IIC 2413 – Bases de Datos
Interrogación 2

Pregunta 1: Varios

a) [0.5 pto] Considere una relación **R** con atributos a, b, c, d, e, f, g y las siguientes dependencias funcionales:

$$ad \rightarrow e$$
$$bce \rightarrow f$$
$$b \rightarrow c$$
$$af \rightarrow g$$

¿Cuál(es) es (son) las posibles llaves de **R**? Explique cómo obtuvo su respuesta.

b) [1.5 pto] Considere la relación de la pregunta anterior. Se añade a ella el atributo h y la siguiente dependencia funcional:

$$h \rightarrow abd$$

¿Cuál(es) es (son) las posibles llaves de **R**? Explique cómo obtuvo su respuesta.

c) [2 ptos] Imagina que eres el repartidor de una prestigiosa revista. Los repartos se hacen una vez al mes a las personas. Los domicilios que debías visitar en septiembre están en la tabla `RepartoAlumnoSeptiembre(domicilio, número_revistas)`. Además, cada vez que se entrega el reparto a una casa, se ingresa en la tabla `Reparto(domicilio, alumno, mes, año)`. Dado que este mes has estado muy ocupado con tus cursos y festejando las fiestas patrias, no has podido ir ni a la mitad de tus domicilios asignados en septiembre. Mañana tu jefe ejecutará la consulta:

```
SELECT domicilio
FROM RepartoAlumnoSeptiembre
WHERE domicilio NOT IN
    (SELECT domicilio FROM Reparto
    WHERE alumno='alumno' AND mes=09 AND año=2016)
```

Sin embargo, tienes un contacto que es el encargado de la base de datos de la compañía, quien te dio la oportunidad de insertar una única tupla a la base de datos, para que no quedaras tan mal con tu jefe. Entregue el comando del tipo `INSERT ... INTO Reparto ...` que inserte una tupla en la tabla `Reparto` y que haga que la consulta señalada anteriormente sea vacía (para que tu jefe crea que fuiste a todos los domicilios). Justifica tu respuesta.

d) [1 pto] Cree dos restricciones de tipo `ASSERTION` para la tabla Reparto de la pregunta anterior. La primera es que durante septiembre de este año el no existan más de 100 registros. La segunda es una restricción que impida la inserción de una tupla como la que insertó en la pregunta anterior.

e) [1 pto] Suponga la existencia de la tabla `Persona(id, Nombre)` y de la tabla `Mascota(Mid, Pid, Nombre)`. La tabla mascota contiene su id y el id de su dueño. Haga una consulta en SQL que haga el join entre ambas tablas **pero** que en caso de que una persona no tenga mascotas, la persona sea parte del ouput de la consulta junto a un `null`.

Pregunta 2: Procedimientos Almacenados

Suponga que tiene una base de datos con las siguientes tablas:

- `A(id int, Anombre varchar(20))`
- `B(id int, Bnombre varchar(20))`

Sin embargo, sobre la base de datos solamente se pueden hacer las consultas `SELECT * FROM A` y `SELECT * FROM B`. Cree un procedimiento almacenado llamado `operacion_conjunto(numero integer)`. Este procedimiento hace lo siguiente:

- **[3 ptos]** Si recibe un 1 calcula la intersección entre `A` y `B`
- **[3 pto]** Si recibe un 2 calcula la unión entre `A` y `B`

Pregunta 3: Recursión

Considere las siguientes tablas de una base de datos:

- `Recorrido(Paradero1, Compañía, Paradero2)`.
- `Pertenece(Compaía1, Compañía2)`

La primera tabla contiene los pares de paraderos conectados por un bus que pertenece a una compañía en particular. La segunda tabla indica si una compañía pertenece a otra compañía. Una instancia del esquema es:

Paradero1	Compañía	Paradero2
Vicuña Mackena	Buses Amarillos	Alameda
Pajaritos	Buses Verdes	Teniente Cruz
Bilbao	Buses Naranjos	Manquehue
General Velazquez	Buses Amarillos	Gran Avenida

Cuadro 1: Instancia de Recorrido

Compañía 1	Compañía 2
Buses Amarillos	Compaía A
Buses Verdes	Compañía B
Buses Naranjos	Compañía C
Compañía A	Compañía Grande
Compañía B	Compañía Grande
Compañía C	Compañía no tan Grande

Cuadro 2: Instancia de Pertenece

Así, en el ejemplo anterior, puedo llegar desde la Vicuña Mackena a la Alameda mediante la Compañía "Buses Amarillos", pero también mediante la Compañía "Compañía A" y la Compañía "Compañía Grande". De la misma forma, para ir de Pajaritos a Teniente Cruz, puedo llegar mediante la Compañía "Buses Verdes", pero también mediante la Compañía "Compañía B" y la Compañía "Compañía Grande". Lo mismo para las demás tuplas.

Entregue una consulta que para cada par de paraderos, entregue todas las compañías mediante las que se puede llegar. Para la instancia anterior, el output de la consulta sería:

Paradero1	Compañía	Paradero2
Vicuña Mackena	Buses Amarillos	Alameda
Pajaritos	Buses Verdes	Teniente Cruz
Bilbao	Buses Naranjos	Manquehue
General Velazquez	Buses Amarillos	Gran Avenida
Vicuña Mackena	Compañía A	Alameda
Pajaritos	Compañía B	Teniente Cruz
Bilbao	Compañía C	Manquehue
General Velazquez	Compañía A	Gran Avenida
General Velazquez	Buses Amarillos	Gran Avenida
Vicuña Mackena	Compañía Grande	Alameda
Pajaritos	Compañía Grande	Teniente Cruz
Bilbao	Compañía no tan Grande	Manquehue
General Velazquez	Compañía Grande	Gran Avenida

Cuadro 3: Ejemplo Output

Pregunta 4: SQL + Programación

Definimos a continuacion el operador / sobre el algebra relacional, llamado *división*. Este operador solo es aplicable a un par de relaciones A, B tal que A tiene atributos X e Y, y B solo tiene el atributo Y. Luego A/B es una relación unaria que tiene solo el atributo X, y contiene a todos los valores x tal que para toda tupla y en B existe una tupla (x, y) en A.

Como un ejemplo, considere las siguientes instancias para relaciones $R_1(a_1, a_2)$ y $R_2(a_2)$:

R_1	a_1	a_2
	1	2
	1	3
	2	3
	2	4

R_2	a_2
	2
	3

R_1/R_2	a_1
	1

Considere ahora una base de datos que contiene dos tablas `A(x,y)` y `B(y)`. Además solamente puede realizar las consultas `SELECT * FROM A` y `SELECT * FROM B`. Cree un programa usando Java o Python que imprima en consola la división entre A y B.