

IIC 2413 – Bases de Datos
Interrogación 2

Pregunta 1: Varios

a) [2 ptos] Considere la siguiente relación para almacenar información de vuelos:

```
Vuelos(id_avión, compañía_vuelo, numero_asientos, ciudad_origen,  
       país_origen, ciudad_destino, país_destino, fecha)
```

donde cada atributo representa lo que usted esperaría. Notar que pueden existir dos ciudades con el mismo nombre pero con distintos países (por ejemplo, Santiago de Chile y Santiago de Cuba). Justifique, utilizando dependencias funcionales, por qué este es un mal diseño para almacenar información de vuelos. Luego entregue un esquema que esté en BCNF, y utilizando dependencias funcionales, explique por qué es correcto. Para llegar a su esquema no es necesario usar el algoritmo de descomposición. Debe asegurarse que el esquema resultante tenga al menos los atributos de la tabla **Vuelos**, pero es posible que requiera añadir atributos.

b) [2 ptos] Considere las tablas $R(i \text{ int}, j \text{ int})$ y $S(i \text{ int})$, junto a las siguientes consultas:

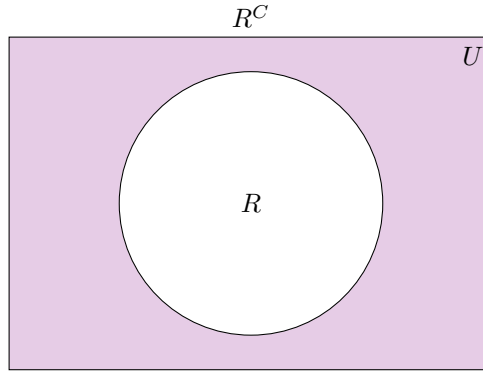
- `SELECT * FROM R WHERE R.i NOT IN (SELECT i FROM S)`
- `SELECT * FROM R WHERE NOT EXISTS (SELECT * FROM S WHERE R.i = S.i)`

Explique por qué cuando hay nulos en la tabla **S**, el output de las consultas es distinto.

c) [2 ptos] Considere las tablas $A(i \text{ int})$ y $B(i \text{ int})$. Escriba un trigger que ante cada inserción de un elemento en la tabla **B**, verifique que ese elemento no está presente en la tabla **A**.

Pregunta 2: Procedimientos Almacenados

En teoría de conjuntos es común querer obtener el complemento de un conjunto R , que corresponde a todos los elementos que están en el universo pero no están en el conjunto R . Esto se muestra en la figura a continuación:



Suponga que usted tiene la relación $R(a \text{ int}, b \text{ int}, c \text{ int})$. Vamos a definir el complemento de R (escrito como R^C) como la siguiente expresión:

$$R^C = \rho(\text{universo}, \pi_a(R) \cup \pi_b(R) \cup \pi_c(R))$$

$$R^C = \text{universo} \times \text{universo} \times \text{universo} - R$$

que intuitivamente es generar todos los valores posible, y luego quitarle los que están presentes en R . Ahora suponga que usted tiene una base de datos que **sólo permite las consultas**:

- `SELECT * FROM R`
- `SELECT a FROM R UNION SELECT b FROM R UNION SELECT c FROM R`

pero tiene acceso a procedimientos almacenados. Entregue un procedimiento almacenado que retorne una tabla cuyo resultado es el complemento de la relación R según lo definido anteriormente.

Pregunta 3: Recursión

Suponga que usted tiene una tabla `Camino`(`ciudad_origen` `varchar(100)`, `costo` `int`, `ciudad_destino` `varchar(100)`, `Primary Key`(`ciudad_origen`, `ciudad_destino`)). Esta tabla representa a las ciudades que están conectadas por un camino desde `ciudad_origen` hasta `ciudad_destino` con un costo asociado (por peajes, combustible, entre otros) equivalente al valor de `costo`.

Entregue una consulta en `SQL` que para un número C dado, entregue todas las ciudades alcanzables entre si cuyo costo acumulado sea menor o igual que C . Por simplicidad asuma que el grafo generado por las ciudades es dirigido, acíclico y que no hay dos formas de llegar a una ciudad.

Pregunta 4: Programación y SQL

En clases vimos el operador `Full Outer Join`. Este operador hace el join de dos tablas A y B . Si alguna tupla de A no hace match con ninguna tupla de B o una tupla de B no hace match con ninguna de A , de todas formas se incluyen en el output, llenando con nulos cuando sea necesario.

Considere dos tablas:

- `A(id int, a_nombre varchar(20))`

■ B(id int, b.nombre varchar(20))

En el siguiente ejemplo se muestra el resultado al hacer el Full Outer Join para una instancia del esquema:

A	id	a_nombre
	1	A1
	2	A2
	3	A3

B	id	b_nombre
	1	B1
	2	B2
	2	B2.2
	4	B4

$A \bowtie B$	A.id	A.a_nombre	B.id	B.b_nombre
	1	A1	1	B1
	2	A2	2	B2
	2	A2	2	B2.2
	3	A3	null	null
	null	null	4	B4

Sobre la base de datos solamente se pueden hacer las consultas `SELECT * FROM A`, `SELECT * FROM B`. Además **las tablas A y B no caben en memoria**. Se pide que haga un programa en Java o Python que imprima en consola el resultado del Full Outer Join entre las dos tablas. Para representar un nulo puede utilizar el *string* vacío.