

IIC 2413 – Bases de Datos
Interrogación 2

1. Pregunta 1: Varios

a) [2 ptos] Considere la siguiente relación para almacenar información de cursos de una universidad:

```
Cursos(id_curso, nombre_curso, sección, id_profesor,  
       sala, rut_alumno, carrera_alumno, horario)
```

donde cada atributo representa lo que usted esperaría. Notar que, por ejemplo, la sección 1 del curso “Cálculo 1” debería tener un id distinto a la sección 2. También, un profesor puede dictar más de un curso y que una sala puede tener distintos cursos en los distintos horarios del día. Además, los alumnos pueden estudiar una única carrera y el horario se representa como una sigla (ej. V 5, M-J 2). Justifique, utilizando dependencias funcionales, por qué este es un mal diseño para almacenar información de los cursos. 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 **Cursos**, pero es posible que requiera añadir atributos.

b) [1 ptos] Considere las tablas $R(i \text{ int})$ y $S(i \text{ int})$, junto a las siguientes consultas:

- `SELECT * FROM R, S WHERE R.i = S.i`
- `SELECT R.i FROM R INTERSECT SELECT S.i FROM S`

Explique por qué cuando hay nulos en ambas tablas, la cardinalidad del output de las consultas puede ser distinto.

c) [3 ptos] Suponga que usted tiene la siguiente relación:

```
Caminos(  
  ciudad_origen VARCHAR(100), ciudad_destino VARCHAR(100), nombre_camino VARCHAR(100)  
)
```

Que indica los pares de ciudades conectadas entre sí, y el nombre del camino que las conecta. Considere ahora la siguiente consulta:

```
WITH RECURSIVE Alcanzo(co, cd, n_camino) AS (  
  SELECT * FROM Caminos  
  UNION  
  SELECT C.ciudad_origen, A.cd, A.n_camino  
  FROM Caminos C, Alcanzo A  
  WHERE C.ciudad_destino = A.co AND C.nombre_camino = A.n_camino  
)  
SELECT * FROM Alcanzo;
```

a) [0.5 pto] ¿Qué representa el resultado de la consulta anterior?

a) [2.5 pto] Explique la forma en que se computa una consulta recursiva en un DBMS. Sea claro sobre que se computa en cada iteración del `WITH RECURSIVE` y cómo el sistema sabe en qué momento detenerse en el cómputo de la recursión.

Pregunta 2: Procedimientos Almacenados

Para esta pregunta considere el siguiente esquema:

```
R(a int, b int)
S(b int, c int)
T(a int, c int)
```

Tú necesitas hacer la siguiente consulta sobre el esquema:

```
SELECT DISTINCT R.a, S.b, T.c
FROM R, S, T
WHERE R.b = S.b AND S.c = T.c AND R.a = T.a;
```

Que corresponde a una consulta de *join*. Tu usuario no tiene los permisos para realizar esta consulta, pero afortunadamente, tu base de datos soporta el lenguaje PL-PgSQL por lo que puedes hacer un **procedimiento almacenado** que compute el resultado de la consulta. En esta pregunta se pide que hagas un procedimiento almacenado llamado `join_triangulo()` que compute la consulta señalada. Dentro del procedimiento está permitido hacer consultas del tipo:

- `SELECT * FROM <Tabla> ORDER BY ...`
- `CREATE TABLE`
- `DROP TABLE`
- `INSERT INTO`

Pregunta 3: Programación y SQL

Considere las relaciones `R(a int, b int, PRIMARY KEY(a, b))` y `S(b int, c int, PRIMARY KEY(b, c))`. Tienes acceso a Python y `psycopg2`, pero sólo puedes hacer consultas a la base de datos del tipo:

- `SELECT * FROM R ORDER BY ...`
- `SELECT * FROM S ORDER BY ...`
- `CREATE TABLE`
- `DROP TABLE`
- `INSERT INTO`

Entregue un algoritmo en Python que haga uso la librería `psycopg2` que calcule la consulta `SELECT * FROM R, S WHERE R.b = S.b`, pero que lea cada tupla una única vez. Además asuma que los valores no caben en memoria, por lo que no es posible usar la función `fetchall()`. Sin embargo, puede asumir que el resultado de la consulta `SELECT * FROM R, S WHERE R.b = S.b AND R.b = i` sí cabe en memoria, para un valor `i` en particular. Deben notar que desde Python **no pueden** hacer una consulta del tipo `SELECT * FROM R, S WHERE R.b = S.b AND R.b = i`. Tampoco puede usar librerías externas, salvo las necesarias para usar `PSQL`.

Bonus [2 décimas en la I2]: Diga el nombre de las 6 gemas del infinito presentes en el MCU.