

**IIC 2413 – Bases de Datos**  
**Interrogación 2 Rúbrica**

## **Pregunta 1: Varios**

- a) La relación no está en BCNF porque se mezcla información de alumnos con información de cursos. Suponiendo que cada sección de un curso tiene una sala y un profesor, un esquema posible puede ser el siguiente:

```
Cursos(id_curso, nombre_curso, sección, id_profesor, sala, horario)
Alumnos(rut_alumno, nombre)
Toma_curso(id_curso, rut_alumno)
```

En que claramente la única dependencia no trivial de cada relación es la que indica cada llave primaria (`id_curso` y `rut_alumno`).

- b) Si hay nulos en ambas tablas, la consulta de producto cruz con igualdad, no considera a los nulos como respuesta, porque `null = null` no es `true`. En cambio en la intersección, los nulos sí son parte de la respuesta.
- c) La consulta muestra todas las ciudades alcanzables entre sí mediante la misma ruta/camino. La evaluación es de la siguiente forma:
  - Se instancia la tabla recursiva (`Alcanzo`), inicialmente vacío. En la primera iteración se llena con los valores de `SELECT * FROM Caminos`.
  - En cada iteración, se computan las ciudades a distancia 2, después 3, y así. Se debe cumplir el requisito de que sea mediante el mismo camino. Los resultados se van guardando en la tabla `Alcanzo`.
  - Si en una iteración no se agrega ningún valor nuevo, el procedimiento para.
  - Se realiza la consulta `SELECT * FROM Alcanzo`.

## Pregunta 2: Procedimientos Almacenados

El código se presenta a continuación:

```
CREATE OR REPLACE FUNCTION dupl()
RETURNS TABLE (a integer, b integer) as $$
DECLARE
    r1 record;
    r2 record;
    r3 record;
    r4 record;
    r5 record;
    r6 record;
BEGIN
    CREATE TABLE join_table(a integer, b integer, c integer);

    FOR r1 IN SELECT * FROM R LOOP
        FOR r2 IN SELECT * FROM S LOOP
            IF r1.b = r2.b THEN
                INSERT INTO join_table(r1.a, r1.b, r2.c);
            END IF;
        END LOOP;
    END LOOP;

    CREATE TABLE join_filter(a integer, b integer, c integer);

    FOR r3 IN SELECT * FROM join_filter LOOP
        FOR r4 IN SELECT * FROM T LOOP
            IF r1.a = r2.a AND r1.c = r2.c THEN
                INSERT INTO join_filter(r1.a, r1.b, r1.c);
            END IF;
        END LOOP;
    END LOOP;

    CREATE TABLE dupl(a integer, b integer, c integer);

    r2 = NULL;
    FOR r1 IN SELECT * FROM join_filter ORDER BY R.a, R.b, R.c LOOP
        IF r1 <> r2 THEN
            INSERT INTO dupl VALUES(r1.a, r1.b, r1.c);
        END IF;
        r2 = r1;
    END LOOP;

    RETURN QUERY SELECT * FROM dupl;

    DROP TABLE join_table;
    DROP TABLE join_filter;
    DROP TABLE dupl;
    RETURN;
END;
$$ language plpgsql;
```

El puntaje se asigna de la siguiente forma:

- 1 pt por hacer el join inicial entre R y S. 2 pts por filtrar según los que estén en T.
- 3 pts por la eliminación de duplicados.

## Pregunta 3: Programación y SQL

El código se presenta a continuación:

```
1 import psycopg2
2
3 conn = psycopg2.connect("dbname=dbname user=dbuser")
4
5 cur_r = conn.cursor()
6 cur_s = conn.cursor()
7
8 cur_r.execute("SELECT * FROM R ORDER BY b")
9 cur_s.execute("SELECT * FROM S ORDER BY b")
10
11 t_r = cur_r.fetchone()
12 t_s = cur_s.fetchone()
13
14 while True:
15     if t_r == None or t_s == None:
16         break
17     if t_r[1] < t_s[0]:
18         t_r = cur_r.fetchone()
19     elif t_r[1] > t_s[0]:
20         t_s = cur_s.fetchone()
21     else:
22         b_r = t_r[1]
23         b_s = t_s[0]
24         l_r = []
25         l_s = []
26         while t_r != None and b_r == t_r[1]:
27             l_r.append(t_r)
28             t_r = cur_r.fetchone()
29         while t_s != None and b_s == t_s[0]:
30             l_s.append(t_s)
31             t_s = cur_s.fetchone()
32
33         for e_r in l_r:
34             for e_s in l_s:
35                 print('{} {}'.format(e_r, e_s))
36
37 cur_r.close()
38 cur_s.close()
```

El puntaje se asigna de la siguiente forma:

- Máximo 2.5 pts en caso de usar `fetchall()`.
- Máximo 4.5 pts en caso de pasar dos veces por la misma tupla, o por utilizar la consulta `SELECT * FROM R, S WHERE R.b = S.b AND R.b = i`.
- 6 pts por la respuesta correcta.