

División Relacional Usando SQL

Estimados alumnos, espero poder explicar a través de este medio la forma de resolver la división en SQL.

En el ejercicio g del punto 4 del práctico 7 pide:

Los docentes que dictan alguna materia en todas las carreras a las que dicha materia pertenece.

Y supongamos que tenemos la siguiente relación dicta

Dicta		
Profesor	Materia	Carrera
Nodi	Algebra	ISI
Nodi	Algebra	LOI
Nodi	Algebra	CIV
Lucrecia	Algebra	ISI
Julio	Análisis I	ISI
Julio	Análisis I	LOI
Mario	Análisis I	ISI
Mario	Análisis I	LOI
Mario	Análisis I	CIV

La estrategia para aplicar la división en SQL no es sencilla, pero básicamente se hacen dos subconsultas anidadas una con otra utilizando el NOT EXISTS.

La resolución de este ejercicio es la siguiente:

```
SELECT DISTINCT d1.dni --del dictado obtener los diferentes docentes
FROM dicta d1
WHERE
    NOT EXISTS (SELECT * --tal que no existe carrera para la materia
                FROM dicta d2
                WHERE
                    d2.idmateria = d1.idmateria AND
                    NOT EXISTS (SELECT * --donde ese docente no dicte
                                FROM dicta d3
                                WHERE
                                    d3.idcarrera = d2.idcarrera AND
                                    d3.idmateria = d2.idmateria AND
                                    d3.dni = d1.dni
                                )
                )
ORDER BY dni;
```

Para entender esto pensemos en una resolución por pasos, sin mirar el cuerpo de la consulta, podemos observar que vamos a utilizar la relación dicta. Por lo tanto tomemos la primer tubla:

Profesor	Materia	Carrera
Nodi	Algebra	ISI

Con esta primer tupla comenzaremos validaremos si la misma se tiene que devolver en el resultado, respondiéndolo el siguiente condicional. Devolver la tupla

Profesor	Materia	Carrera
Nodi	Algebra	ISI

si no existe

```

SELECT DISTINCT d1.dni --del dictado obtener los diferentes docentes
FROM dicta d1
WHERE
    NOT EXISTS (SELECT * --tal que no existe carrera para la materia
                FROM dicta d2
                WHERE
                    d2.idmateria = d1.idmateria AND
                    NOT EXISTS (SELECT * --donde ese docente no dicte
                                FROM dicta d3
                                WHERE
                                    d3.idcarrera = d2.idcarrera AND
                                    d3.idmateria = d2.idmateria AND
                                    d3.dni = d1.dni
                                )
                )
ORDER BY dni;

```

Paso 1, Paso 4

Paso 2, Paso 5

Paso 3, Paso 6

Paso 1

Profesor	Materia	Carrera
Nodi	Algebra	ISI

Paso 2

Profesor	Materia	Carrera
Nodi	Algebra	ISI
Nodi	Algebra	LOI
Nodi	Algebra	CIV
Lucrecia	Algebra	ISI

Paso 3

Profesor	Materia	Carrera
Nodi	Algebra	ISI
Nodi	Algebra	LOI
Nodi	Algebra	CIV

Paso 1:

Como describimos en la página anterior, nos imaginamos que tomamos la primer tupla para operar.

Esta primer tupla va a ser devuelta si no existe ninguna tupla en la subconsulta.

Paso 2:

Para este paso, la estrategia es buscar todas las tuplas en **dicta** donde el código de materia coincide con el código de materia de la tupla del **Paso 1** (o sea donde la materia se Algebra).

Paso 3:

En este paso, para simplificar la comprensión, no haremos el procedimiento iterativo. Pero lo que pregunta la sub consulta anidada es lo siguiente “No debe existir ninguna **carrera** en la relación del **Paso 2** donde el docente sea Nodi”. Lo cual es falso porque Nodi está relacionada con las 3 carreras donde se dicta Algebra, por lo que el resultado de la primer subconsulta será vacío y por lo tanto se devolverá la tupla del **Paso 1**.

Aplicando el mismo criterio y continuando con la iteración de tuplas que siguen a la tupla del **Paso 1** en dicta, se devolverán las tuplas:

Profesor	Materia	Carrera
Nodi	Algebra	LOI
Nodi	Algebra	CIV

Pero cuando lleguemos a la tupla siguiente, que a partir de ahora llamaremos tupla del **Paso 4**.

Profesor	Materia	Carrera
Lucrecia	Algebra	CIV

Paso 5:

Para este paso, la estrategia es buscar todas las tuplas en *dicta* donde el código de materia coincide con el código de materia de la tupla del **Paso 4** (o sea donde la materia se Algebra).

Paso 6:

De nuevo, para simplificar la comprensión, no haremos el procedimiento iterativo. Pero lo que pregunta la sub consulta anidada es lo siguiente “No debe existir ninguna *carrera* en la relación del **Paso 5** donde el docente sea Lucrecia”. Lo cual es verdadero porque Lucrecia no está relacionada con las carreras CIV y LOI que también se dicta Algebra, por lo que el resultado de la primer subconsulta no será vacío y por lo tanto se no devolverá la tupla del **Paso 4**.

Así se seguirá iterando por el resto de las tuplas de dicta hasta completar todo, por último se haría el paso del DISTINCT donde se eliminarían los docentes repetidos.

Autores: Pedro Agustín Brau y Matías Molina

Paso 5

Profesor	Materia	Carrera
Nodi	Algebra	ISI
Nodi	Algebra	LOI
Nodi	Algebra	CIV
Lucrecia	Algebra	ISI

Anexo

Una cosa a tener en cuenta para la resolución de este ejercicio es que ninguna de las siguientes consultas es equivalente.

Consulta 1

```
SELECT DISTINCT d1.dni
FROM dicta d1
WHERE
    NOT EXISTS (SELECT *
                FROM dicta d2
                WHERE
                    d2.idmateria = d1.idmateria AND
                    NOT EXISTS (SELECT *
                              FROM dicta d3
                              WHERE
                                  d3.idcarrera = d2.idcarrera AND
                                  d3.idmateria = d1.idmateria AND
                                  d3.dni = d1.dni
                              )
                )
ORDER BY dni;
```

Consulta 2

```
SELECT DISTINCT d1.dni
FROM dicta d1
WHERE
    NOT EXISTS (SELECT *
                FROM dicta d2
                WHERE
                    d2.idmateria = d1.idmateria AND
                    NOT EXISTS (SELECT *
                              FROM dicta d3
                              WHERE
                                  d3.idcarrera = d2.idcarrera AND
                                  d3.dni = d1.dni
                              )
                )
ORDER BY dni;
```

¿Alguien me puede decir por que no son equivalentes?

Acá les dejo una estructura para que prueben con datos. Deberían cargar esta estructura, verificar cuales docentes deberían devolver como resultado y que resultados les dan las distintas consultas. Suman puntos de conceptos los que den una respuesta coherente.

```
CREATE TABLE dicta2(profesor VARCHAR(25) NOT NULL, materia VARCHAR(25) NOT NULL, carrera
VARCHAR(25) NOT NULL, CONSTRAINT pk_dicta2 PRIMARY KEY(profesor, materia, carrera));
```

```
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Nodi', 'Algebra', 'ISI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Lucrecia', 'Algebra', 'ISI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Lucrecia', 'Analisis I', 'LOI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Lucrecia', 'Analisis I', 'CIV');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Nodi', 'Algebra', 'LOI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Nodi', 'Algebra', 'CIV');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Julio', 'Analisis I', 'ISI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Julio', 'Analisis I', 'LOI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Mario', 'Analisis I', 'ISI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Mario', 'Analisis I', 'CIV');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Mario', 'Analisis I', 'LOI');
INSERT INTO dicta2(profesor, materia, carrera) VALUES('Mario', 'Analisis I', 'ELE');
```