

# PRÁCTICA 1

## Sistemas de Gestión de Bases de Datos



**22 OCTUBRE**

**PRÁCTICA 1 – EJERCICIO 12**

**Mario Ventura Burgos 43223476-J**

**Grado en Ingeniería Informática (GIN 3)**

**CURSO 2023-2024**

# 1. CONSULTA Y TRADUCCIÓN A ÁLGEBRA

La consulta 1 del examen se desarrolló en el documento mvb135\_2.sql y es la siguiente:

## CONSULTA:

```
SELECT ALI.referencia, ALI.nom
FROM Aliment ALI
  JOIN Linia_Albara LA
  ON LA.referencia = ALI.referencia
  JOIN Albara ALB
  ON ALB.codi = LA.codi_alb
  JOIN proveedor PRO
  ON PRO.nif = ALB.nif_pro
  AND PRO.nom = 'UIBFruita'
ORDER BY ALI.referencia ASC;
```

Ahora haremos la transformación a álgebra relacional. Para ello, se debe poder entender y establecer todas las relaciones de la consulta en SQL, por tanto, no podemos empezar desde la parte más “externa” de esta consulta. Debemos adentrarnos en la consulta e ir analizando las relaciones siguiendo un orden que vaya desde la parte más “interna” (se podría entender como los nodos hoja de los árboles que se usaron como representación de estas consultas en sus respectivos ejercicios de plan de ejecución) de la consulta hasta la parte externa o final (operaciones de ordenación de resultados o similares). De esta manera, obtenemos las siguientes relaciones:

## TRADUCCIÓN A ÁLGEBRA:

$R1 \rightarrow (ALB \bowtie PRO)_{ALB.nif\_pro=PRO.nif \text{ AND } PRO.nom='UIBFruita'}$

$R2 \rightarrow (LA \bowtie R1)_{LA.codi\_alb=ALB.codi}$

$R3 \rightarrow (ALI \bowtie R2)_{LA.referencia=ALI.referencia}$

$R4 \rightarrow \Pi_{ALI.referencia, ALI.nom} (R3)$

$R5 \rightarrow \theta_{ALI.referencia} (R4)$

Una vez hecho esto, podemos crear una única expresión algebraica combinando todas las anteriores. Partiendo desde el final, sustituiremos en R5 el valor de R4, en R4 el valor de R3, y así sucesivamente hasta llegar a obtener solo una expresión algebraica. El resultado que obtendríamos es el siguiente:

$\theta_{ALI.referencia} (\pi_{ALI.referencia, ALI.nom} (ALI \bowtie LA.referencia = ALI.referencia (LA \bowtie LA.codi\_alb = ALB.codi (ALB \bowtie ALB.nif\_pro = PRO.nif \text{ AND } PRO.nom = 'UIBFruita')))))$

---

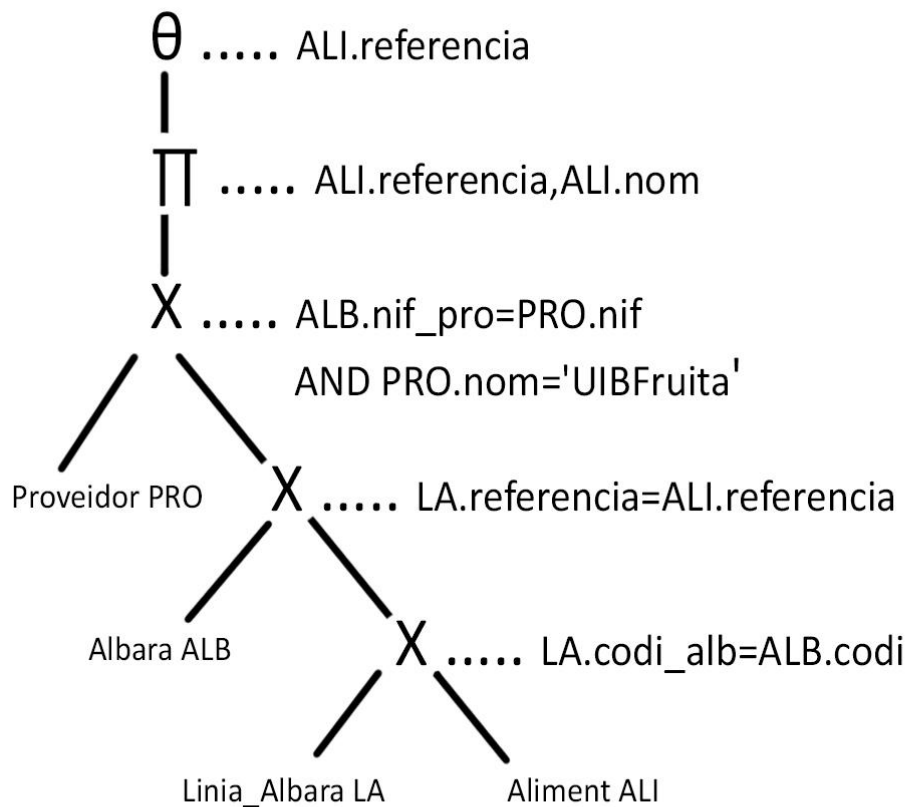
## 2. ANÁLISIS DE LAS RELACIONES

En el paso anterior hemos obtenido una serie de expresiones algebraicas. Estas representan una secuencia de operaciones en álgebra relacional que se utilizan para obtener un conjunto de datos específico a partir de múltiples tablas, aplicando, en cada paso, joins, proyecciones, selecciones... Vamos a analizar cada una de las relaciones obtenidas con el objetivo de entender qué operación algebraica representan, sobre qué tabla se aplica esta operación y con qué finalidad:

- **R1:** Realiza una operación de JOIN entre las tablas Albara (ALB) y Proveedor (PRO) bajo la condición de unión siguiente:  $ALB.nif\_pro=PRO.nif \text{ AND } PRO.nom='UIBFruita'$ .  
La unión viene determinada por el operador  $\bowtie$ , y la condición de unión es la que se puede ver en el ON de la consulta original en SQL. Esta condición asegura que solo se incluyan filas donde el nif\_pro en la tabla ALB sea igual al nif en la tabla PRO y donde el nombre (columna nom) en PRO sea igual a 'UIBFruita'. El resultado contendrá las columnas de ambas tablas.  
Cabe destacar que la segunda parte de la condición ( $AND PRO.nom = 'UIBFruita'$ ) podría haberse quitado del ON y añadirse en un WHERE en la consulta original, pero en ese caso, sería necesario una expresión algebraica más y el árbol de expresiones sería diferente como consecuencia.
- **R2:** Representa otra operación de JOIN, esta vez entre las tablas Linia\_Albara (LA) y las dos tablas anteriores descritas en R1. Dado que se une LA con el resultado obtenido de la expresión R1 se usa el operador  $\bowtie$  con LA y R1.  
En este caso la condición de unión es más simple ya que se basa solamente en la igualdad de la columna codi\_alb en LA con la columna codi en ALB.
- **R3:** Realiza una operación de unión entre las tablas Aliment (ALI) y el resultado de la expresión R2, que, a su vez, contiene el resultado de R1. De esta forma se consigue unir todas estas tablas y esta es la razón por la cual es importante empezar a desarrollar las expresiones algebraicas desde la parte más “interna” de la consulta. Una vez más se usa el operador  $\bowtie$  entre las tablas ALI y R2 bajo la condición de que la columna referencia en ALI sea igual que la columna referencia en R2.
- **R4:** Representa una operación de proyección mediante el SELECT sobre la tabla R3 obtenida anteriormente. Las columnas proyectadas son ALI.referencia y ALI.nom, de la tabla Aliment (ALI), y el objetivo de esto es seleccionar solo las columnas referencia y nom de la tabla resultante de la expresión R3.
- **R5:** Esta expresión algebraica representa la ordenación según la referencia del alimento (ORDER BY en SQL). Se realiza mediante el operador Theta sobre la proyección mostrada

### 3. ÁRBOL DE EXPRESIONES

Ahora que tenemos todas las relaciones establecidas y conocemos que representa cada una, podemos hacer un árbol de expresiones que ilustre de forma gráfica estas operaciones. En este caso, el árbol de expresiones es el siguiente:



Donde X representa la operación de JOIN (no es un producto cartesiano, pero el símbolo de JOIN no estaba disponible en la herramienta utilizada para crear el árbol)