**Bases de Datos I**
**12/10/2020**
**Apuntes de la clase 12/10**
IS-501
José Inestroza

Continuación **Ejemplo → DDS_DMS.sql**

**-- RESPUESTA CORRECTA --**
-- Listar todos los computadores que hay por cantidad de RAM, mostrando los grupos donde hay 2 o más dispositivos mostrando solo 3 registros y que estén ordenados de mayor a menor–
**SELECT sma_ram AS “RAM”, COUNT(*) AS “Cantidad” FROM PCInvetory GROUP BY sma_ram HAVING COUNT(*)>=2 ORDER BY sma_ram DESC LIMIT 3;**

-- Liste las computadoras que pertenecen a los tres grupos mayores de RAM. Si una computadora pertenece a la 4ta mayor agrupación de RAM, dicho computador no debe aparecer en la consulta –
-- PCGroup es el nombre de la subcocnsulta usada luego del JOIN –
-- Se usa tilde inversa para aceptar cadenas con valores compuestos --
**SELECT tex_name AS “Nombre del computador” FROM PCInventory JOIN (SELECT sma_ram AS “RAM”, COUNT(*) AS “Cantidad” FROM PCInvetory GROUP BY sma_ram HAVING COUNT(*)>=2 ORDER BY sma_ram DESC LIMIT 3) PCGroup ON PCInventory.sma_ram = PCGroup. `RAM`;**

**--** Usando LEFT JOIN --
**SELECT tex_name AS “Nombre del computador” FROM PCInventory LEFT JOIN (SELECT sma_ram AS “RAM”, COUNT(*) AS “Cantidad” FROM PCInvetory GROUP BY sma_ram HAVING COUNT(*)>=2 ORDER BY sma_ram DESC LIMIT 3) PCGroup ON PCInventory.sma_ram = PCGroup. `RAM`;**

---

## Algebra Relacional

**Table 8.1** Operations of Relational Algebra

| OPERATION | PURPOSE | NOTATION |
|---|---|---|
| SELECT | Selects all tuples that satisfy the selection condition from a relation $R$. | $\sigma_{<\text{selection condition}>}(R)$ |
| PROJECT | Produces a new relation with only some of the attributes of $R$, and removes duplicate tuples. | $\pi_{<\text{attribute list}>}(R)$ |
| THETA JOIN | Produces all combinations of tuples from $R_1$ and $R_2$ that satisfy the join condition. | $R_1 \bowtie_{<\text{join condition}>} R_2$ |
| EQUIJOIN | Produces all the combinations of tuples from $R_1$ and $R_2$ that satisfy a join condition with only equality comparisons. | $R_1 \bowtie_{<\text{join condition}>} R_2$, OR $R_1 \bowtie_{(<\text{join attributes 1}>), (<\text{join attributes 2}>)} R_2$ |
| NATURAL JOIN | Same as EQUIJOIN except that the join attributes of $R_2$ are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all. | $R_1 *_{<\text{join condition}>} R_2$, OR $R_1 *_{(<\text{join attributes 1}>), (<\text{join attributes 2}>)} R_2$ OR $R_1 * R_2$ |
| UNION | Produces a relation that includes all the tuples in $R_1$ or $R_2$ or both $R_1$ and $R_2$; $R_1$ and $R_2$ must be union compatible. | $R_1 \cup R_2$ |
| INTERSECTION | Produces a relation that includes all the tuples in both $R_1$ and $R_2$; $R_1$ and $R_2$ must be union compatible. | $R_1 \cap R_2$ |
| DIFFERENCE | Produces a relation that includes all the tuples in $R_1$ that are not in $R_2$; $R_1$ and $R_2$ must be union compatible. | $R_1 - R_2$ |
| CARTESIAN PRODUCT | Produces a relation that has the attributes of $R_1$ and $R_2$ and includes as tuples all possible combinations of tuples from $R_1$ and $R_2$. | $R_1 \times R_2$ |
| DIVISION | Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in $R_1$ in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$. | $R_1(Z) \div R_2(Y)$ |

- **JOIN →** Retorna los registros que tienen una condición de igualdad ambas tablas
- **LEFT JOIN →** Devuelve todos los registros de la tabla izquierda (la que está en el FROM) y los registros coincidentes de la tabla derecha (la que está en el JOIN), el resultado es NULL desde el lado derecho sino hay coincidencia.
- **RIGTH JOIN →** Devuelve todos los registros de la tabla derecha.