

```
mysql -u root -p  
password: root
```

JOINS

PRODUCTO CARTESIANO

```
Select * from t1, t2;
```

```
Select * from t1 join t2;
```

Por **cada fila** de la tabla t_1 genera una nueva tupla en la tabla resultado adjuntándole **cada una de las filas** de la tabla t_2 .

Si la aridad de t_1 es 1 y la aridad de t_2 es 1, una proyección total sobre la tabla resultado de un **PC** arrojará 2 columnas.

¿Cuál es **número** de **TUPLAS** esperado luego de un **PC**?

T1		T2		T1 x T2																					
<table><tr><th>A</th></tr><tr><td>a1</td></tr><tr><td>a2</td></tr></table>	A	a1	a2	x	<table><tr><th>B</th></tr><tr><td>b1</td></tr><tr><td>b2</td></tr><tr><td>b3</td></tr></table>	B	b1	b2	b3	=	<table><tr><th>A</th><th>B</th></tr><tr><td>a1</td><td>b1</td></tr><tr><td>a1</td><td>b2</td></tr><tr><td>a1</td><td>b3</td></tr><tr><td>a2</td><td>b1</td></tr><tr><td>a2</td><td>b2</td></tr><tr><td>a2</td><td>b3</td></tr></table>	A	B	a1	b1	a1	b2	a1	b3	a2	b1	a2	b2	a2	b3
A																									
a1																									
a2																									
B																									
b1																									
b2																									
b3																									
A	B																								
a1	b1																								
a1	b2																								
a1	b3																								
a2	b1																								
a2	b2																								
a2	b3																								

INNER JOIN

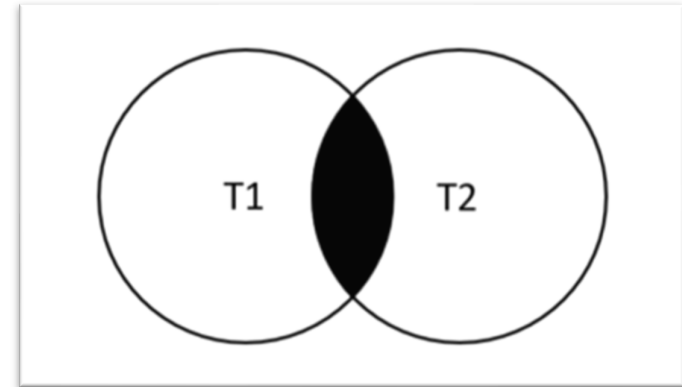
```
Select * from t1 inner join t2 using t1.atr1 = t2.atr2;
```

Compara **cada fila** de la tabla t_1 **con cada fila** de la tabla t_2 buscando una coincidencia entre los atributos atr_1 y atr_2 .

Si en un renglón atr_1 es igual a atr_2 , el resultado es un TRUE con lo que **INNER JOIN** toma los valores de cada una de las columnas de las tablas t_1 y t_2 y los inserta en una nueva fila en la tabla resultado.

En caso de que $atr_1 \neq atr_2$, **INNER JOIN** simplemente ignora las filas.

Si ninguna fila entre las tablas provoca una evaluación de TRUE, **INNER JOIN** devuelve un conjunto de resultados vacío.



INNER JOIN + ON: recuperará en la tabla resultado todas las columnas de las tablas t_1 y t_2 .

E.g.: si la aridad de t_1 es 2 y la de t_2 es 3, una proyección total sobre la tabla resultado arrojará 5 columnas.

INNER JOIN + USING: sólo puede utilizarse si el atributo de condición tiene el mismo nombre en ambas tablas. Recuperará el total de columnas resultante de sumar el número de campos de t_1 con el número de campos de t_2 menos una.

E.g.: si la aridad de t_1 es 2 y la de t_2 es 3, pero ambas tienen un atributo llamado $atri_esp$ y por él se condiciona el join, la proyección total sobre la tabla resultado arrojará 4 columnas.

¿Cuál es el máximo número de TUPLAS que podemos esperar luego de un INNER JOIN?

NATURAL JOIN

```
Select * from t1 natural join t2;
```

Realiza la misma tarea que una **INNER JOIN**, sin la necesidad de especificar la cláusula **ON** o **USING**.

Las columnas de las tablas intervenidas con el mismo nombre aparecerán una sola vez.

¿Cuál es el **máximo** número de **TUPLAS** que podemos esperar luego de aplicar un **NATURAL JOIN**?

¿Cuál es el **mínimo** número de **COLUMNAS** que podemos obtener luego de un **NATURAL JOIN** si la tabla resultado no está vacía?

SQL Query Execution Order

En SQL, el "orden de ejecución" es la jerarquía bajo la cual el lenguaje evalúa las cláusulas dentro de una consulta.

- Suele ser diferente al orden que está escrita la consulta.
- Es útil cuando se busca depurar y optimizar consultas.

FROM: señala la tabla en dónde se almacenan los datos requeridos.

JOIN + ON: obtiene los registros coincidentes basado en columna(s) compartida(s) cuando los datos requeridos viven en varias tablas.

WHERE: filtra los datos de origen solicitados a nivel de fila.
No permite alias de columna o funciones de agregación a aplicar.

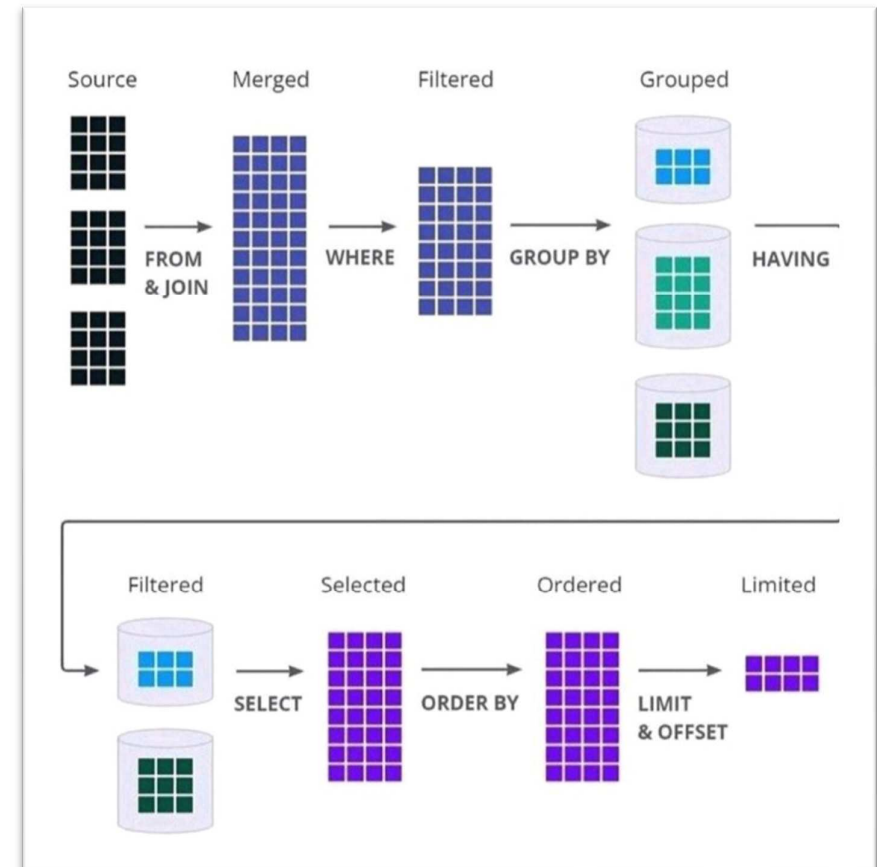
GROUP BY: agrupa filas con el mismo valor juntas. A menudo se usa con funciones agregadas. Permiten alias de columna.

HAVING: aplica condiciones y/o filtros después de que tiene lugar la agregación. No permite alias de columna.

SELECT: proyecta columnas específicas.

ORDER BY: ordena las filas resultantes después de filtrado y agregación. Admite alias de columna.

LIMIT: reduce el número de filas devueltas por la consulta.



Fuente: <https://data-science-infinity.teachable.com/>

Generar un script (archivo .sql) que realice las siguientes instrucciones:

Ejercicio 1 – Base de datos ‘bank’

0. Mandar llamar al archivo ‘base_bank.sql’.

tabla	num_registros
account	24
acc_transaction	21
employee	18
customer	13
individual	9
product	8
branch	4
business	4
department	4
officer	4
product_type	3

11 rows in set (0.00 sec)

1. Mostrar el código de producto, nombre de producto y tipo de producto de todos los productos que hayan sido calificado por su tipo.
2. Mostrar el resultado anterior, pero ahora filtrando solo los productos cuyo código comience con M.

3. Encontrar la fecha, monto y ciudad de las transacciones que se realizaron por clientes de Salem o por los que tienen cuentas con un saldo mayor a los 5,000 pesos.
4. Resolver el ejercicio anterior con **NATURAL JOIN**.
5. Calcular el número de días transcurridos entre la última transacción y la fecha de las transacciones del ejercicio 3.
6. Mostrar los siguientes datos de la(s) empresa(s) que tenga(n) el menor número de cuentas abiertas en este banco.

Empresa	Num cuentas	Ciudad apertura	Empleado encargado
Northeast Cooling Inc.	1	Woburn	Paula Roberts
Superior Auto Body	1	Salem	Theresa Markham
AAA Insurance Inc.	1	Quincy	John Blake

3 rows in set (0.00 sec)

Ejercicio extra - Base de datos 'bank'

Generar una **captura de pantalla** que incluya la consulta y el siguiente resultado:

```
***** 1. row *****
      id: 4
Información: Susan Hawthorne tiene 5 empleados a cargo.
***** 2. row *****
      id: 6
Información: Helen Fleming tiene 3 empleados a cargo.
***** 3. row *****
      id: 1
Información: Michael Smith tiene 2 empleados a cargo.
***** 4. row *****
      id: 10
Información: Paula Roberts tiene 2 empleados a cargo.
***** 5. row *****
      id: 13
Información: John Blake tiene 2 empleados a cargo.
***** 6. row *****
      id: 16
Información: Theresa Markham tiene 2 empleados a cargo.
***** 7. row *****
      id: 3
Información: Robert Tyler tiene 1 empleado a cargo.
7 rows in set (0.00 sec)
```

❖ Recompensa:

5/100 sobre Práctica 06 Update, Delete & Joins
(la máx calif ahora es 115).

❖ Condiciones:

- Modicar el prompt para que muestre la hora y su núm de equipo.
- El resultado de la consulta debe ser idéntico al anterior.
- Mandar por telegram a @pejcovich antes de las 23:59 del 8 de nov.

