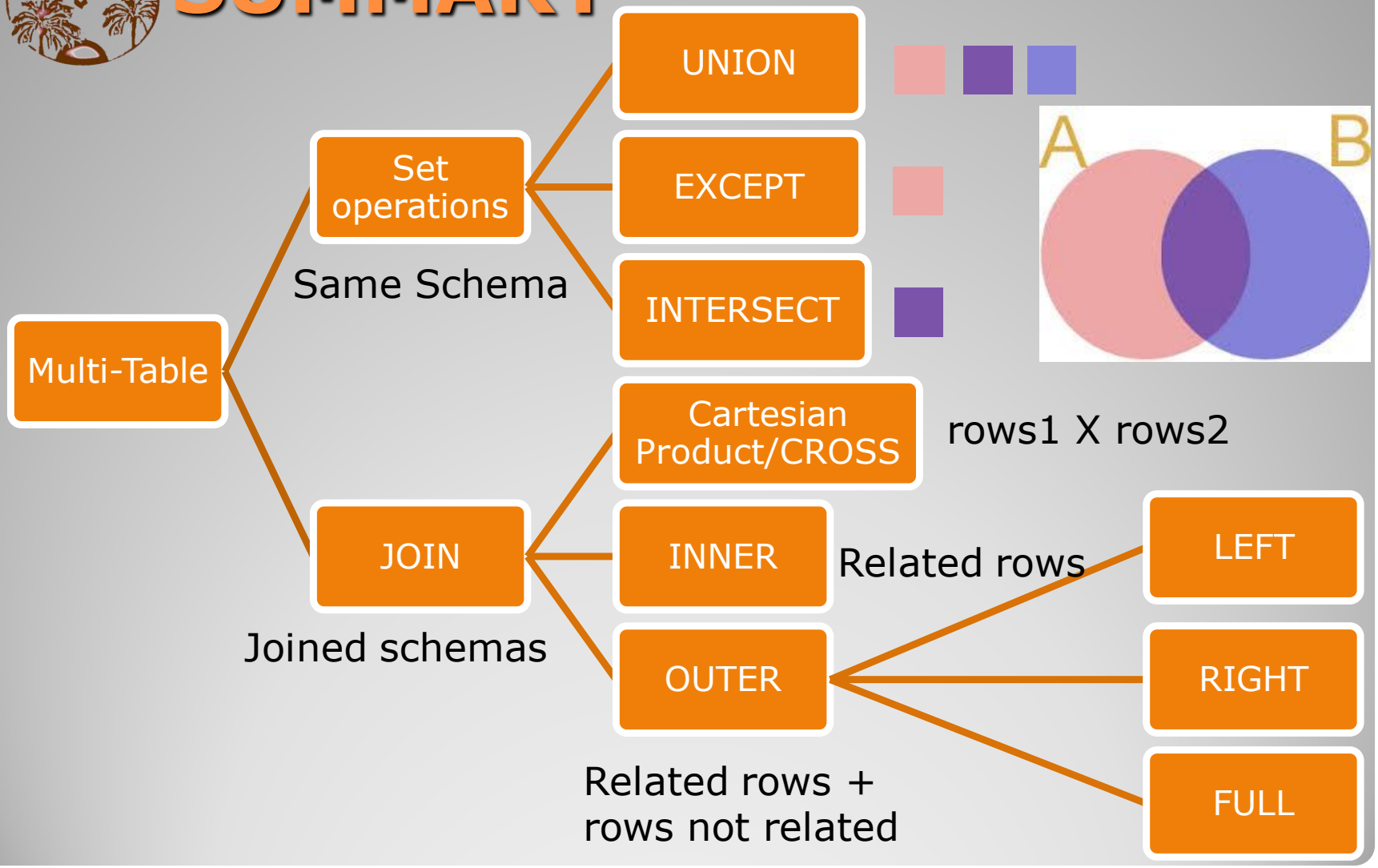
MULTI-TABLE QUERIES

UNIT 5. Part 2

SUMMARY


Multi-Table
Set operations
Same Schema
UNION
EXCEPT
INTERSECT
A
B
JOIN
Joined schemas
Cartesian Product/CROSS
rows1 X rows2
INNER
Related rows
OUTER
Related rows + rows not related
LEFT
RIGHT
FULL

SUMMARY

SET OPERATIONS

Complete SELECT
UNION
Complete SELECT

JOINS

SELECT …
FROM **Table1** CROSS JOIN **Table2**

SELECT …
FROM **Table1**, **Table2**

SELECT …
FROM **Table1** INNER JOIN **Table2** ON condition

SELECT ---
FROM **Table1** LEFT|RIGHT|FULL OUTER JOIN **Table2** ON condition

JOINS - Summary

- Combine each row1 with each row2 → CROSS JOIN.
- With a combining condition → INNER JOIN (at first)
- If it is possible to have rows from table1 that do not comply with the condition AND they must appear in the result → LEFT JOIN.
- If it is possible to have rows from table2 that do not comply with the condition AND they must appear in the result → RIGHT JOIN.
- If we have a LEFT and RIGHT → FULL JOIN.

Combine several operations

- Within ( ) write a complete JOIN:

  (Table …JOIN Table ON condition)

- () has to be considered as a Table:
  (Table … condition)...JOIN Table ON condition

- In *condition* use columns from the tables related with JOIN.

- When several (), they are calculated like a mathematical operation.

Combine several operations

I do recommend the use of ( ).

```
SELECT  *
FROM (oficinas RIGHT JOIN empleados
                        ON empleados.oficina=oficinas.oficina)
                    INNER JOIN pedidos ON rep=numemp;
```

Or:

```
SELECT  *
FROM oficinas RIGHT JOIN (empleados INNER JOIN pedidos ON
rep=numemp)
                    ON empleados.oficina=oficinas.oficina;
```

Derived tables

- A derived table is a complete SELECT placed in ( ) in a FROM clause and that represents a source table. It must have an table-alias.

```
SELECT numemp,nombre,empleados.oficina, ciudad
FROM empleados INNER JOIN oficinas
        ON empleados.oficina=oficinas.oficina
WHERE region='Este';
```

- and

```
SELECT numemp,nombre,empleados.oficina, ciudad
FROM empleados INNER JOIN (SELECT * FROM oficinas WHERE
  region='Este') AS Ofi
        ON empleados.oficina=Ofi.oficina;
```

- Both obtain the same result, but in different ways.

Reflexive queries

- A reflexive query refers to joining a table to itself.
- Any kind of JOINs are supported.
- It works as a normal query and it is quite useful to imagine the table twice.
- The use of a table-alias in at least one table is compulsory.
- Example:

```
SELECT *
FROM Empleados INNER JOIN Empleados AS Jefes
        ON Empleados.jefe=Jefes.numemp;
```

Reflexive queries

```
SELECT *
FROM Empleados INNER JOIN Empleados AS Jefes
        ON Empleados.jefe=Jefes.numemp;
```

Empleados

Empleados AS jefes

Numemp	Nombre	...	jefe	Numemp	Nombre	...	jefe
102	Pepe	...	104	102	Pepe	...	104
103	Luis	...	110	103	Luis	...	110
104	Ana	...		104	Ana	...	
...	...	...	...	...	...	...	...