04

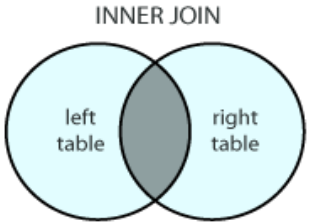
Joining tables & Unions

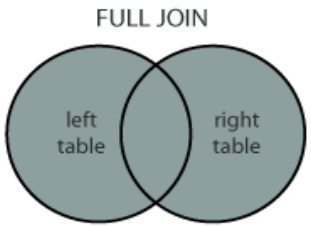
Joining two tables
Join with multible tables

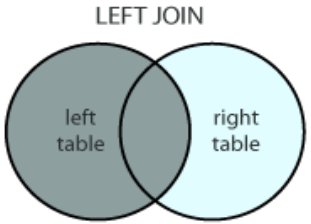
Joins

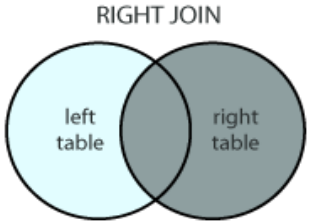
JOIN clause is used to combine rows from two or more tables, based on a related column between them

- **INNER JOIN**: Returns records that have matching values in both tables
- **LEFT JOIN**: Return all records from the left table, and the matched records from the right table
- **RIGHT JOIN**: Return all records from the right table, and the matched records from the left table
- **FULL JOIN**: Return all records when there is a match in either left or right table





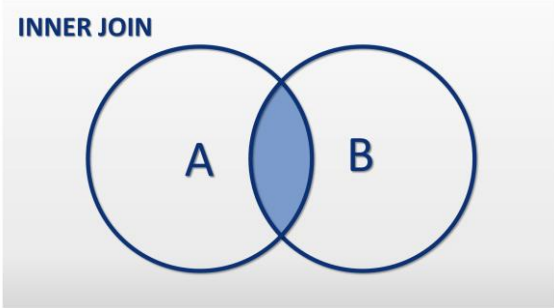




INNER JOIN

Only the rows from the tables that match on the joining columns will show up in the results

```
SELECT <select list>
FROM <table1> AS 'new_name1'
[INNER] JOIN <table2> AS 'new_name_2'
ON new_name1.key_column = new_name_2.key_column
```



Class	Rank
A	2
B	3
C	4
D	1

INNER JOIN

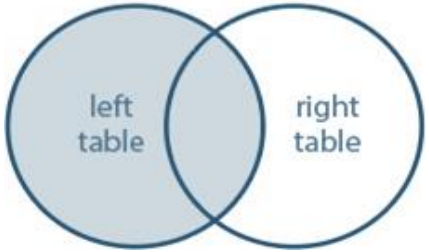
Group	Member
C	6
D	7
E	8
F	9

Group	Rank	Member
C	4	6
D	1	7

LEFT JOIN

If the main table, the table you want to see all the rows from even if there is not a match, is on the left side of the join, you will specify LEFT

```
SELECT <select list>
FROM <table1> AS 'new_name1'
LEFT JOIN <table2> AS 'new_name_2'
ON new_name1.key_column = new_name_2.key_column
```



Class	Rank
A	2
B	3
C	4
D	1

LEFT JOIN

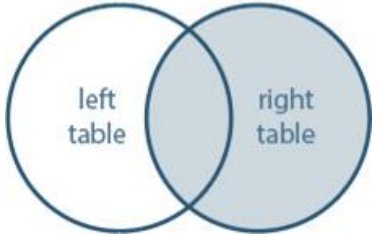
Group	Member
C	6
D	7
E	8
F	9

Group	Rank	Member
A	2	NULL
B	3	NULL
C	4	6
D	1	7

RIGHT JOIN

Differs from LEFT JOIN in just the location of the tables. If the main table, the table in which you want to see all the rows, even if there is not a match, is on the right side of the join, you will specify RIGHT

```
SELECT <select list>
FROM <table1> AS 'new_name1'
RIGHT JOIN <table2> AS 'new_name_2'
ON new_name1.key_column = new_name_2.key_column
```



Class	Rank
A	2
B	3
C	4
D	1

RIGHT JOIN

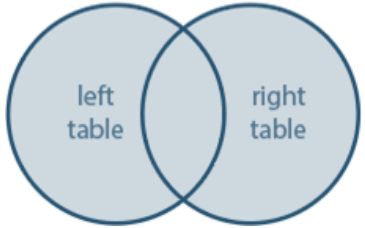
Group	Member
C	6
D	7
E	8
F	9

Group	Rank	Member
C	4	6
D	1	7
E	NULL	8
F	NULL	9

FULL JOIN

FULL OUTER JOIN is similar to LEFT OUTER JOIN and RIGHT OUTER JOIN, but in this case, all the rows from each side of the join are returned

```
SELECT <select list>
FROM <table1> AS 'new_name1'
FULL JOIN <table2> AS 'new_name_2'
ON new_name1.key_column = new_name_2.key_column
```



Class	Rank
A	2
B	3
C	4
D	1

FULL JOIN

Group	Member
C	6
D	7
E	8
F	9

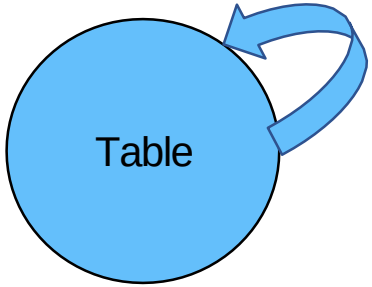
Group	Rank	Member
A	2	NULL
B	3	NULL
C	4	6
D	1	7
E	NULL	8
F	NULL	9

SELF JOIN

Is a regular join, but the table is joined with itself.

The self join can be viewed as a join of two copies of the same table. The table is not actually copied, but SQL performs the command as though it were.

```
SELECT <select list>
FROM <table1> AS 'new_name1'
FULL JOIN <table2> AS 'new_name_2'
ON new_name1.key_column = new_name_2.key_column
```



id	Staff_name	Manager_id
1	Maz	3
2	Lucas	1
3	Hayden	2

SELF JOIN

id	Staff_name	Manager_id
1	Maz	3
2	Lucas	1
3	Hayden	2

id	Staff_name	Manager_name
1	Maz	Hayden
2	Lucas	Maz
3	Hayden	Lucas

Time for practices

Exercise 6: **Generate invoice reports**

Adventure Works Cycles sells directly to retailers, who must be invoiced for their orders. You have been tasked with writing a query to generate a list of invoices to be sent to customers.

Retrieve customer orders:

As an initial step towards generating the invoice report, write a query that returns the company name from the **SalesLT.Customer** table, and the sales order ID and total due from the **SalesLT.SalesOrderHeader** table.

Time for practices

Exercise 7: **Retrieve customer orders with addresses**

Extend your customer orders query to include the **Main Office** address for each customer, including the full street address, city, state or province, postal code, and country or region

Writing UNION Queries

A **UNION** query combines two queries, and the results are returned in one result set

```
SELECT  <col1>, <col2>, <col3>
FROM    <table1>
UNION [ALL]
SELECT  <col4>, <col5>, <col6>
FROM    <table2>
```