SQL Basic Operations

LIKE & ILIKE Statement

In order to match a string against a general pattern we use **LIKE** and **ILIKE** for example:
All emails ending with '@gmail.com'

LIKE & ILIKE allows us to perform pattern matching against string data with the use of **wildcard** characters

- **Percent %** - Matches any sequence of characters
- **Underscore _** - Matches any single character

LIKE is case-sensitive whereas **ILIKE** is case-insensitive

LIKE & ILIKE Statment

All names that begin with ‘A’
WHERE name **LIKE** ‘A%’

All names that end with ‘a’
WHERE name **LIKE** ‘%a’

Get all Mission Impossible Films
WHERE name **LIKE** ‘Mission Impossible _ _’

Combination of Wildcards
WHERE name LIKE ‘_ her%’

- Cheryl
- Theresa
- Sherri

Aggregate Functions

The main idea behind aggregate function is to take multiple inputs and return a single output

- **AVG()** - Returns floating point values. **ROUND()** can be used to specify precision after the decimal
- **COUNT()**
- **MAX()**
- **MIN()**
- **SUM()**

GROUP BY Statement

GROUP BY allows us to aggregate data and apply functions to better understand how data is distributed per category

Category	Data
A	10
A	5
B	2
B	4
C	12
C	6

A	10
A	5

B	2
B	4

C	12
C	6

AVG()

Category	Result
A	7.5
B	3
C	9

HAVING BY Statement

HAVING allows us to filter after an aggregation has already taken place.

We can use it along with a **GROUP BY**

AS Statement

AS allows us to create an Alias for a column or a result

AS operator gets executed at the very end of the query meaning we can not use the Alias inside a **WHERE** or **HAVING** operator

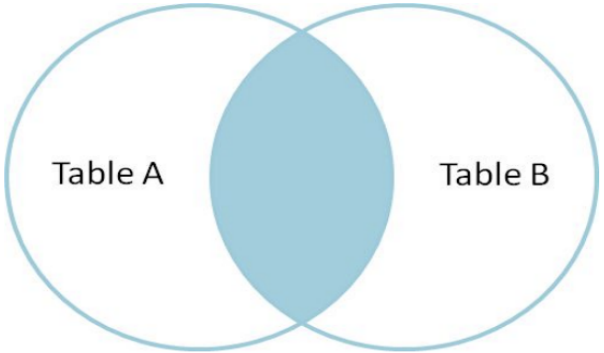
INNER JOIN Statement

Reg_ID	Name
1	Andrew
2	Bob
3	Charlie
4	David

Log_ID	Name
1	Xavier
2	Andrew
3	Yauren
4	Bob

→

Reg_ID	Name	Log_ID	Name
1	Andrew	2	Andrew
2	Bob	4	Bob



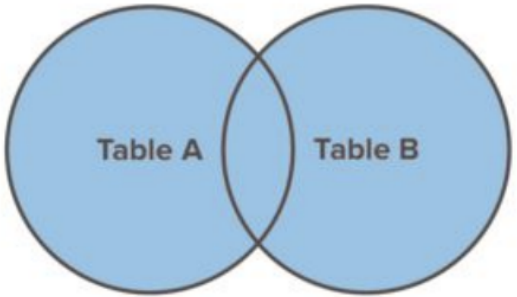
FULL OUTER JOIN Statement

Reg_ID	Name
1	Andrew
2	Bob
3	Charlie
4	David

Log_ID	Name
1	Xavier
2	Andrew
3	Yauren
4	Bob

→

Red_ID	Name	Log_ID	Name
1	Andrew	2	Andrew
2	Bob	4	Bob
3	Charlie	null	null
4	David	null	null
null	null	1	Xavier
null	null	3	Yauren



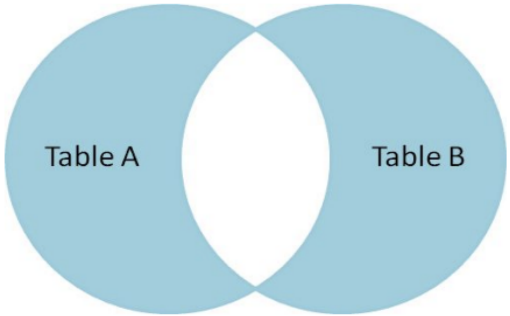
FULL OUTER JOIN with WHERE

Reg_ID	Name
1	Andrew
2	Bob
3	Charlie
4	David

Log_ID	Name
1	Xavier
2	Andrew
3	Yauren
4	Bob

→

Red_ID	Name	Log_ID	Name
3	Charlie	null	null
4	David	null	null
null	null	1	Xavier
null	null	3	Yauren



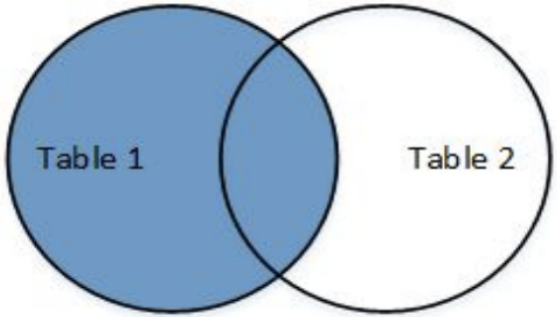
LEFT OUTER JOIN Statement

Reg_ID	Name
1	Andrew
2	Bob
3	Charlie
4	David

Log_ID	Name
1	Xavier
2	Andrew
3	Yauren
4	Bob

→

Red_ID	Name	Log_ID	Name
1	Andrew	2	Andrew
2	Bob	4	Bob
3	Charlie	null	null
4	David	null	null



LEFT OUTER JOIN with WHERE

Reg_ID	Name
1	Andrew
2	Bob
3	Charlie
4	David

Log_ID	Name
1	Xavier
2	Andrew
3	Yauren
4	Bob

→

Red_ID	Name	Log_ID	Name
3	Charlie	null	null
4	David	null	null