DATA WITH BARAA

SQL COMPONENTS

Baraa Khatib Salkini
YouTube | **DATA WITH BARAA**
SQL Course | SELECT Query

```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)

FROM customers

WHERE country = 'Italy'
```

DATA WITH BARAA

SQL STATEMENT

```sql
-- Retrieve Customers Data

SELECT

    name,

    LOWER(country)

FROM customers

WHERE country = 'Italy'
```

Comment

```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)
FROM customers
WHERE country = 'Italy'
```

```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)
FROM customers
WHERE country = 'Italy'
```

Clauses

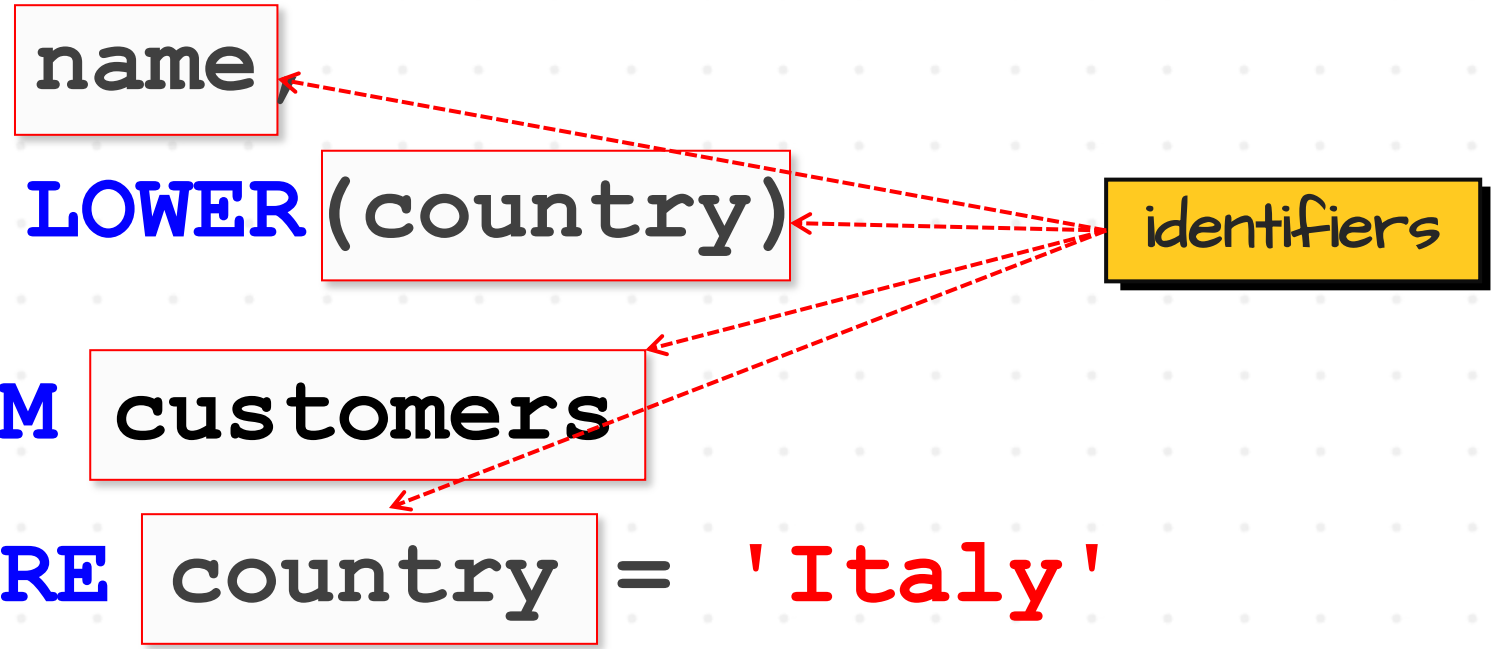
```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)
FROM customers
WHERE country = 'Italy'
```

Keywords

```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)
FROM customers
WHERE country = 'Italy'
```

Function

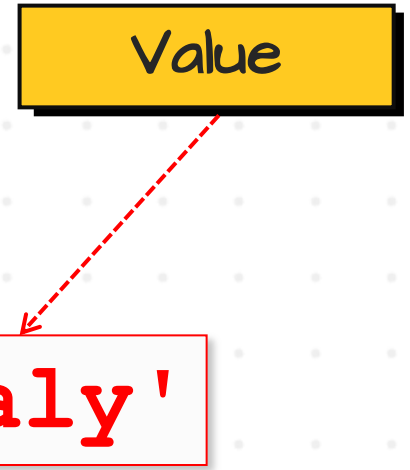
```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)
FROM customers
WHERE country = 'Italy'
```



```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)

FROM customers

WHERE country = 'Italy'
```

Operator

```sql
-- Retrieve Customers Data
SELECT
    name,
    LOWER(country)

FROM customers

WHERE country = 'Italy'
```

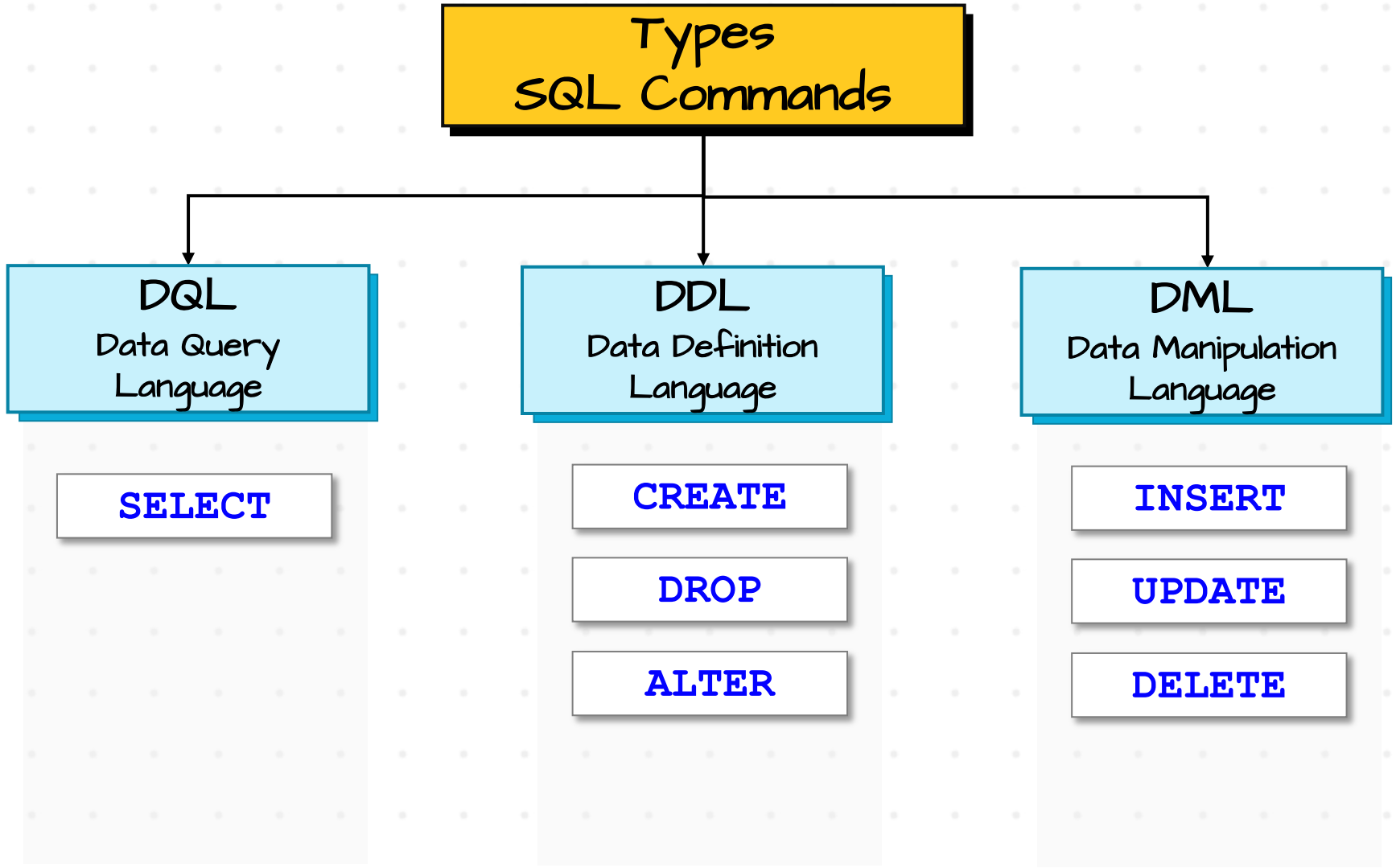


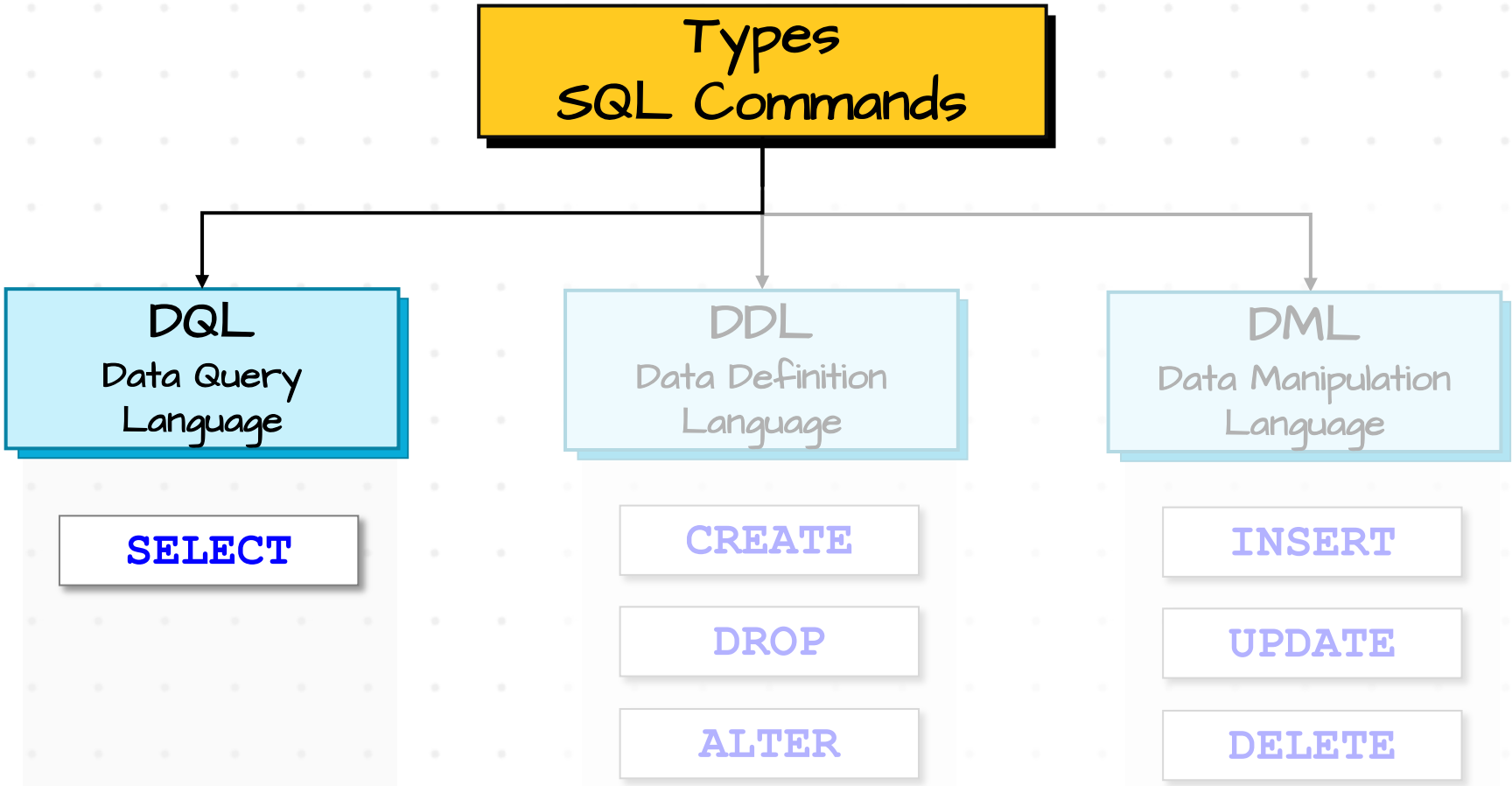
DATA WITH BARAA

QUERY DATA

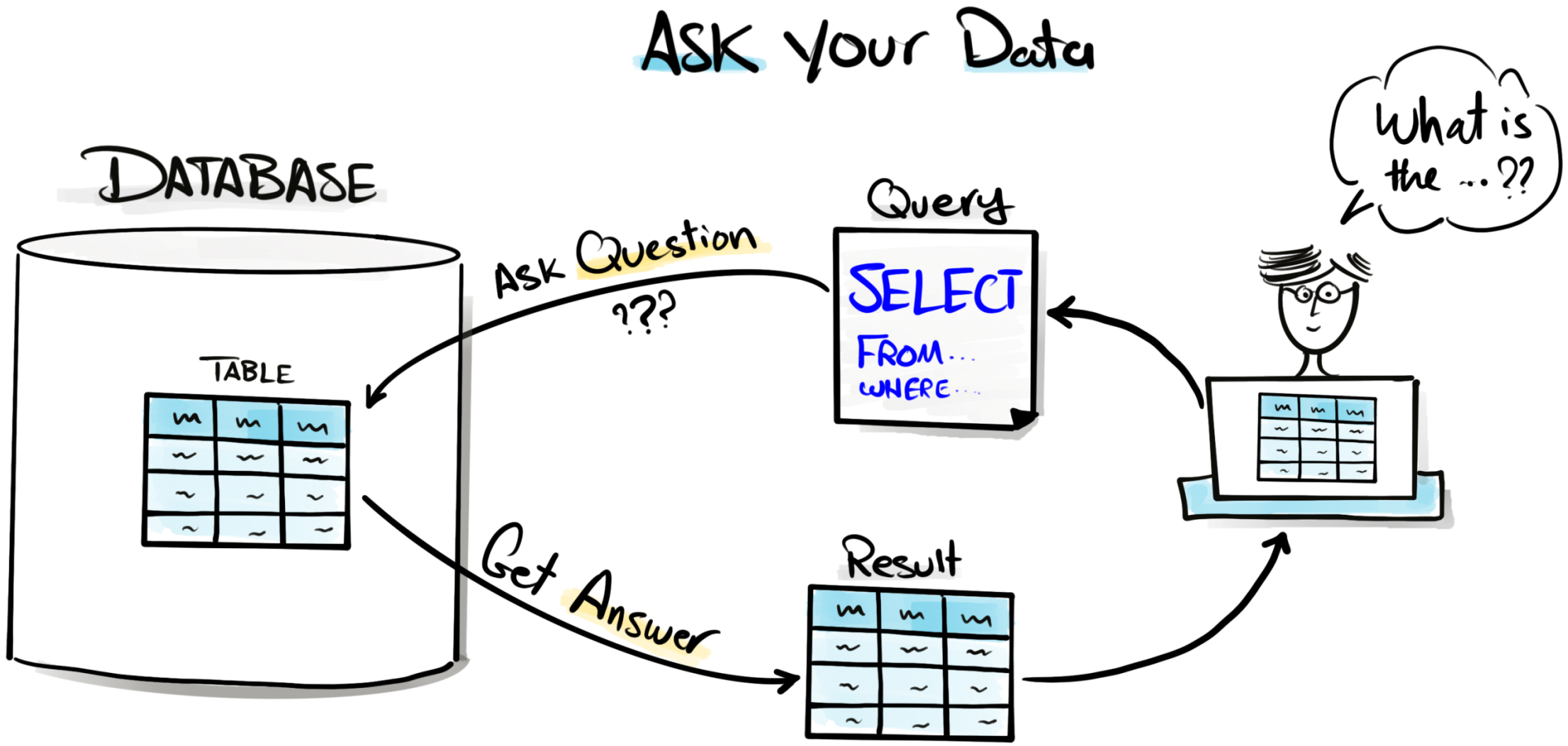
SELECT STATEMENT

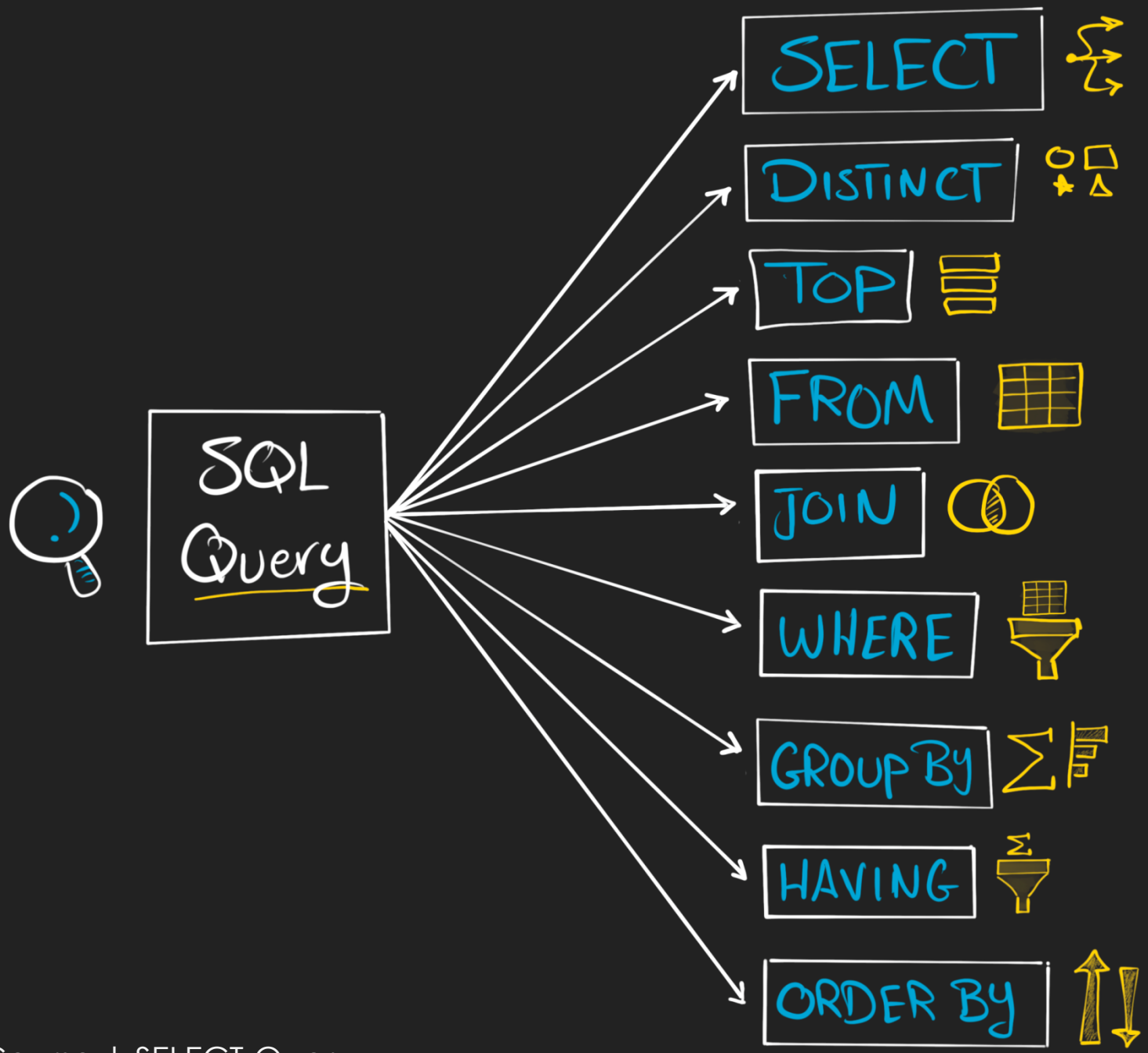
Baraa Khatib Salkini
YouTube | **DATA WITH BARAA**
SQL Course | SELECT Query


Types
SQL Commands
DQL
Data Query Language
SELECT
DDL
Data Definition Language
CREATE
DROP
ALTER
DML
Data Manipulation Language
INSERT
UPDATE
DELETE



Before executing SQL always make sure you are in correct databse.
to chnage database run SQL "USE DB_Name" it will chnage databse in use.


Ask your Data
DATABASE
TABLE
Query
SELECT
FROM...
WHERE...
Ask Question
???
What is the ...??
Get Answer
Result



DATA WITH BARAA

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

DATA WITH BARAA

SELECT *

FROM Table

Select * (All)

Retrieves All Columns (Everything)

From

Tells SQL Where to find Your Data

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

2 SELECT *

1 FROM Table

Keep All Columns!!

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

DATA WITH BARAA

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

```
SELECT
Col1,
Col2
FROM Table
```

Select Few Columns

Pick only the Columns You Need

instead of All.

there is no comma at last element of list.
and result table have columns in order as written in query

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Keeps only Needed Columns

name	Country
Maria	Germany
John	USA
Georg	UK
Martin	Germany
Peter	USA

② SELECT
Col 1,
Col 2

① FROM Table

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

DATA WITH BARAA

```
SELECT *
FROM Table
WHERE Condition
```

Where

Filters Your Data based on a Condition

Score Higher than 500

Data with Baraa

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

3 SELECT *

1 FROM Table

2 WHERE Condition

Score >500

id	name	Country	Score	
1	Maria	Germany	350	✗
2	John	USA	900	✓
3	Georg	UK	750	✓
4	Martin	Germany	500	✗
5	Peter	USA	0	✗

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

```
SELECT *
FROM Table
ORDER BY Score DESC
```

Order By

Sort Your Data

- ASC
  - Lowest
  - Highest
- DESC
  - Highest
  - Lowest

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

id	name	Country	Score
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
1	Maria	Germany	350
5	Peter	USA	0

Highest

Lowest

③ SELECT *

① FROM Table

② ORDER BY Score DESC

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Lowest

Highest

id	name	Country	Score
4	Martin	Germany	500
1	Maria	Germany	350
3	Georg	UK	750
2	John	USA	900
5	Peter	USA	0

Highest

Lowest

Highest

Lowest

③ SELECT *

① FROM Table

② ORDER BY

Country ASC,

Score DESC

order by runs sequentially and for same value of last condition column.

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Every selected column will be only eighter in GROUP By and aggregator function column

Group By

Combines rows with the Same Value

Aggregates a Column By another Column

Total Score By Country

```
SELECT
    Country,
    SUM(score)
FROM Table
GROUP By Country
```

Category

Aggregation

DATA WITH BARAA

Data with Baraa

Database

	id	name	Country	Score
1	1	Maria	Germany	350
2	2	John	USA	900
3	3	Georg	UK	750
4	4	Martin	Germany	500
5	5	Peter	USA	0

Σ

1	Germany	850
2	USA	900
3	UK	750

③ SELECT
Country,
SUM(score)

① FROM Table

② GROUP BY Country

we can group by multiple column combination also.
result will have grouped rows only

Clause execution orderFROMWHEREGROUP BYAGGREGATEHAVINGRANKSELECTDISTINCTORDER BYTOP/LIM

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Having

Filters Data After Aggregation

Can be used only with Group By

```
SELECT
    Country,
    SUM(score)
FROM Table
GROUP BY Country
HAVING SUM(score) > 800
```

DATA WITH BARAA

Data with Baraa

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Germany	850
USA	900

Total Score > 800

4. SELECT
   Country,
   SUM(score)
1. FROM Table
2. GROUP BY Country
3. HAVING SUM(score) > 800

Data with Baraa

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

```
SELECT
    Country,
    SUM(score)
FROM Table
WHERE Score > 400
GROUP BY Country
HAVING SUM(score) > 800
```

Filter Your Data

1. Before Aggregation
2. After Aggregation

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Distinct

Removes Duplicates (Repeated Values)

each Value appears only Once

SELECT DISTINCT

Col

FROM Table

DISTINCT removes duplicate rows.it applied after SELECT and before ORDER BY. it is applies to hall selected row tuple not single row.

TOP n returns top n rows.
LIMIT 10 OFFSET 20;
it skips first 20 roes and then take 10 rows.they used together always.
TOP,LIMIT/FETCH are applied after ORDER BY

SELECT TOP 5 *FROM Employees;

can use this also for top 10% -> SELECT TOP 10 PERCENT *

SELECT *FROM EmployeesLIMIT 5;

LIMIT with OFFSET -> LIMIT 5 OFFSET 10;

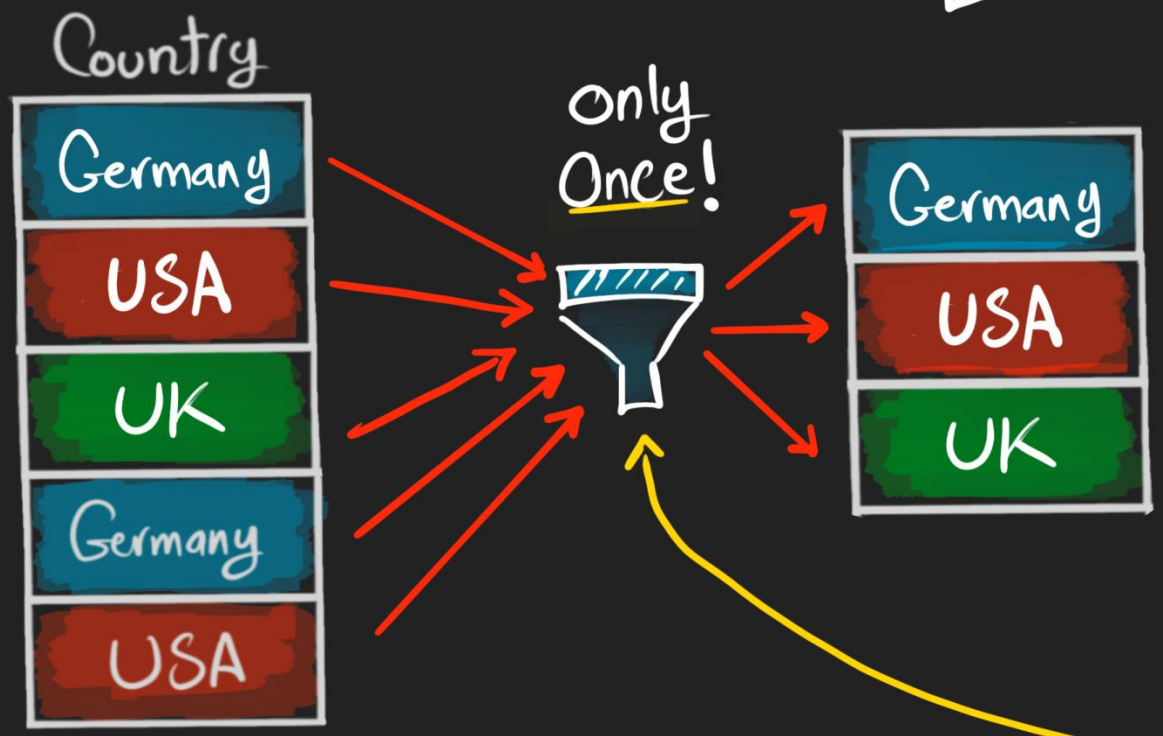
id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Database

③

② SELECT DISTINCT Country

① FROM Table



Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

DATA WITH BARAA

SELECT Top 3 *

FROM Table

Top (Limit)

Restrict the Number of Rows Returned

Data with Baraa

Database

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

③

② SELECT Top 3

+

① FROM Table

	id	name	Country	Score
Row 1 →	1	Maria	Germany	350
Row 2 →	2	John	USA	900
Row 3 →	3	Georg	UK	750

Execute Order

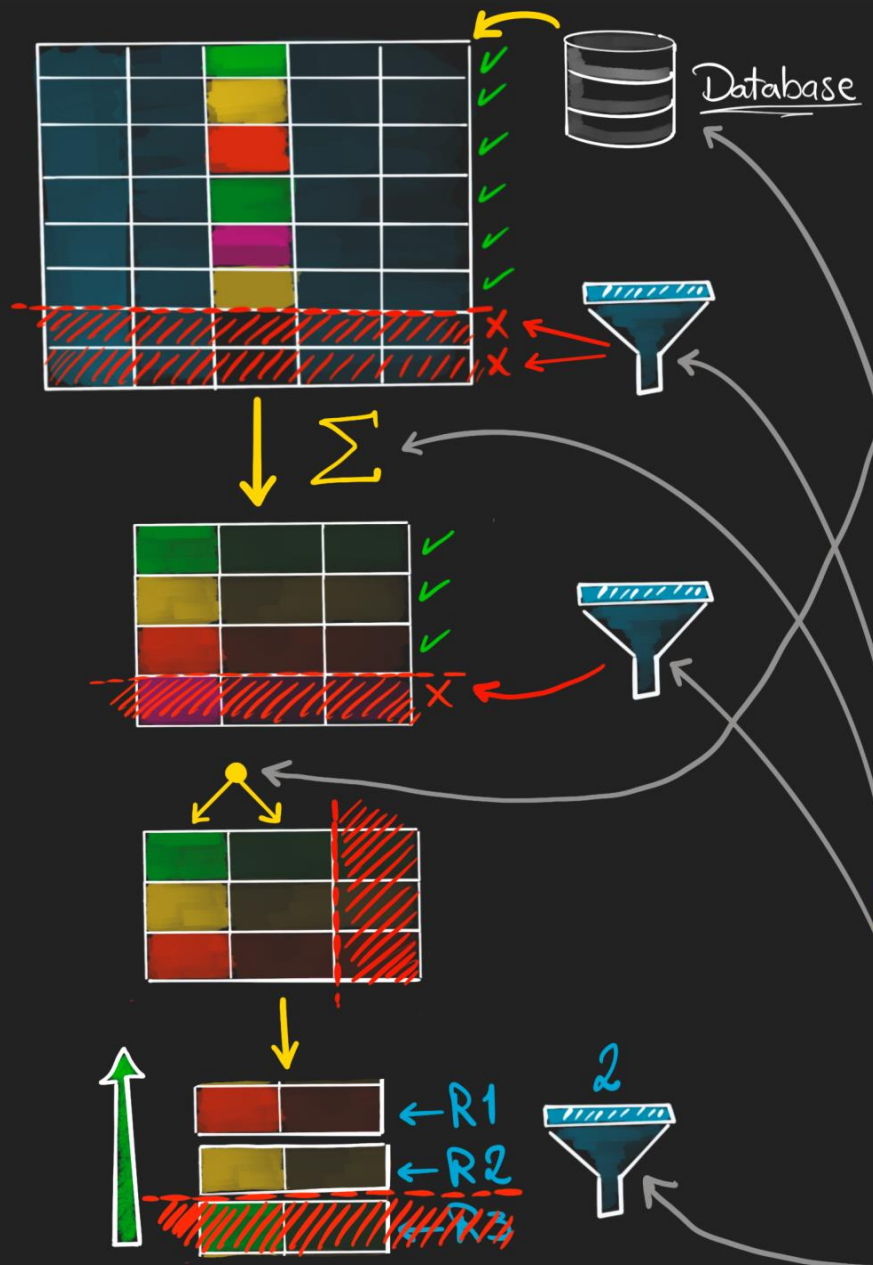
- Filter Columns
- Filter Duplicates
- Filter Result Rows
- Filter Rows Before Aggregations
- Filter Rows After Aggregations

Coding Order

```
SELECT DISTINCT TOP 2
    Col1,
    SUM(Col2)
FROM Table
WHERE Col = 10
GROUP BY Col1
HAVING SUM(Col2) >30
ORDER BY Col1 ASC
```

Execute Order

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT DISTINCT
6. ORDER BY
7. TOP



Coding Order

```
5 SELECT DISTINCT Col1,
      SUM(Col2)
  TOP 2
7
1 FROM Table
2 WHERE Col = 10
3 GROUP BY Col1
4 HAVING SUM(Col2) >30
6 ORDER BY Col1 ASC
```

DATA WITH BARAA

DATA WITH BARAA

BONUS

Sketches

To execute particular part of SQL we can select and run.
we can execute multiple SQL queries also at a time just separate that queries by ;.

Never write =NULL or !=NULL its always false.write IS NULL and IS NOT NULL.

Baraa Khatib Salkini
YouTube | **DATA WITH BARAA**
SQL Course | SELECT Query

Database

Customers

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

① FROM

Result

② SELECT ★

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

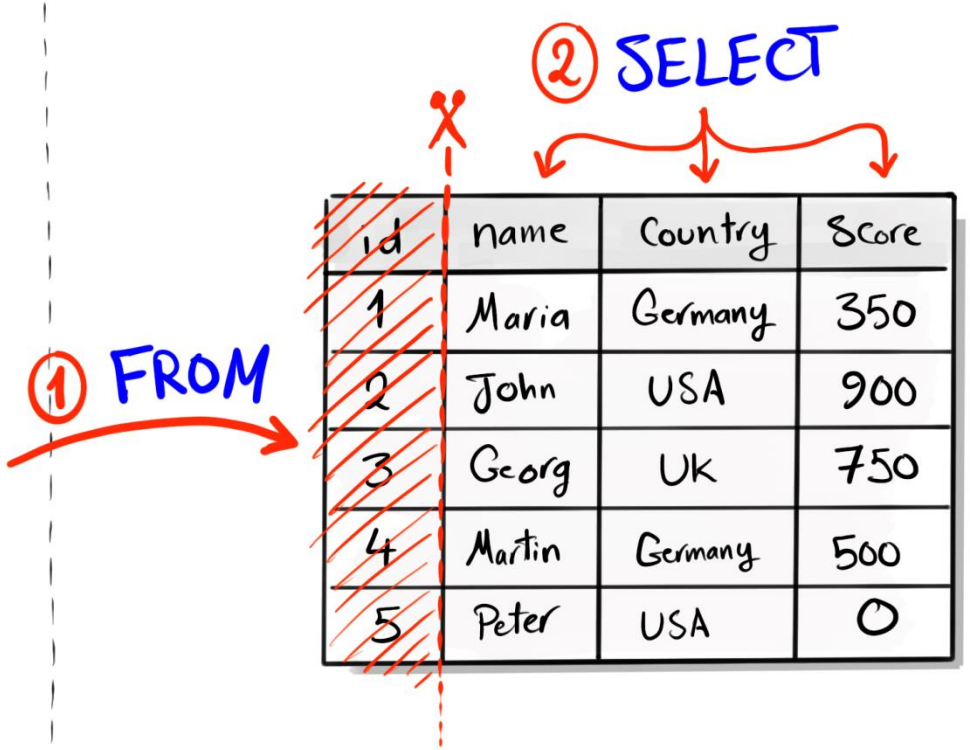
Execute Order

① FROM

Database

Customers

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0



Result

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0

Execute Order

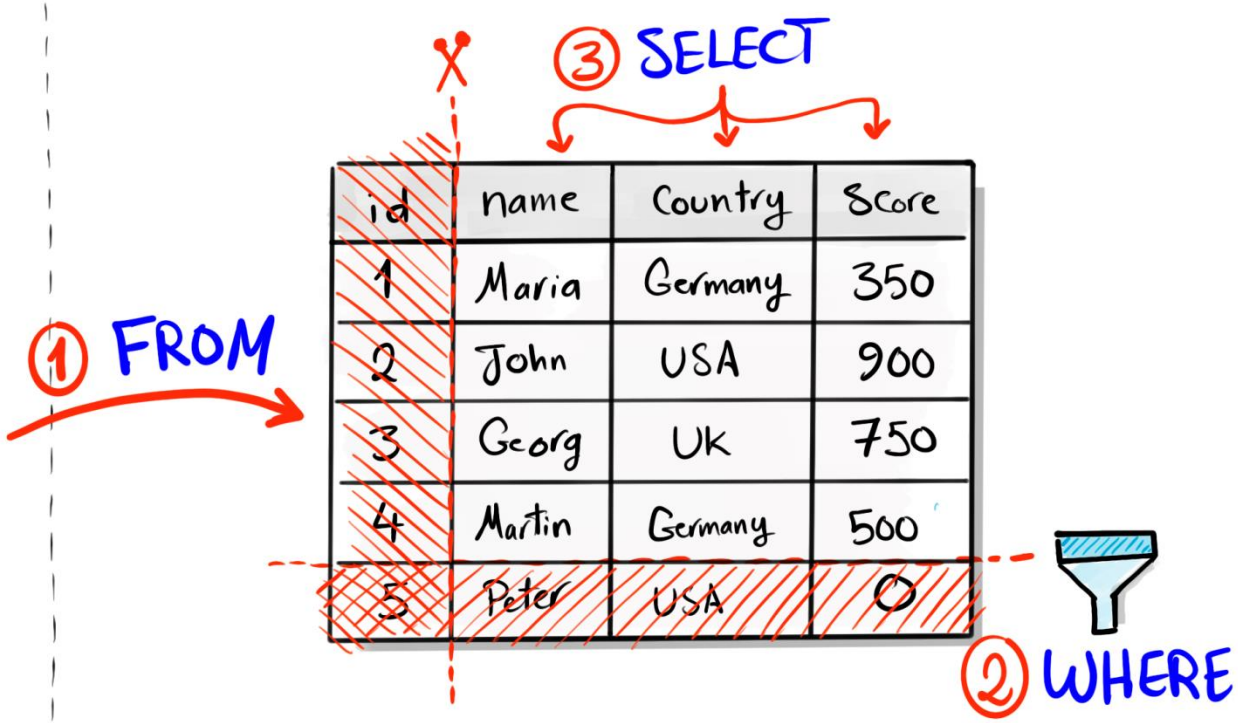
1 FROM

2 SELECT

Database

Customers

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0

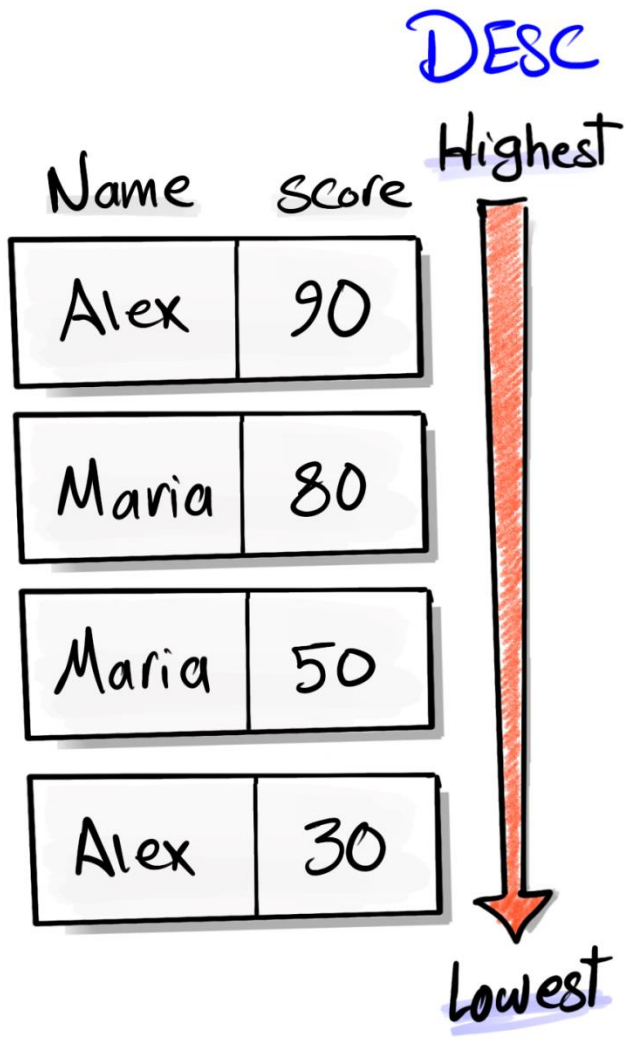


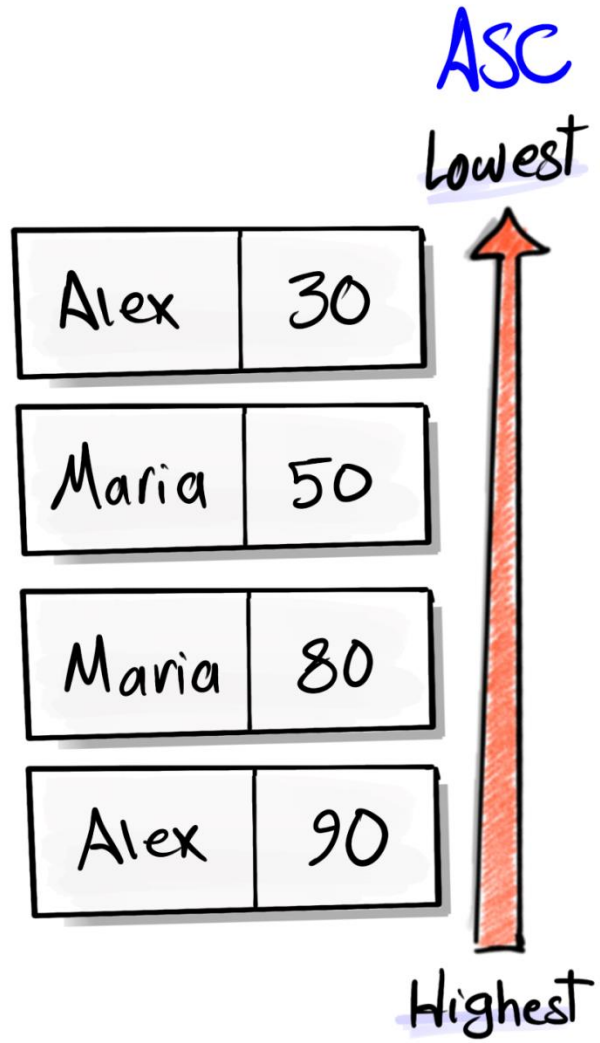
Result

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500

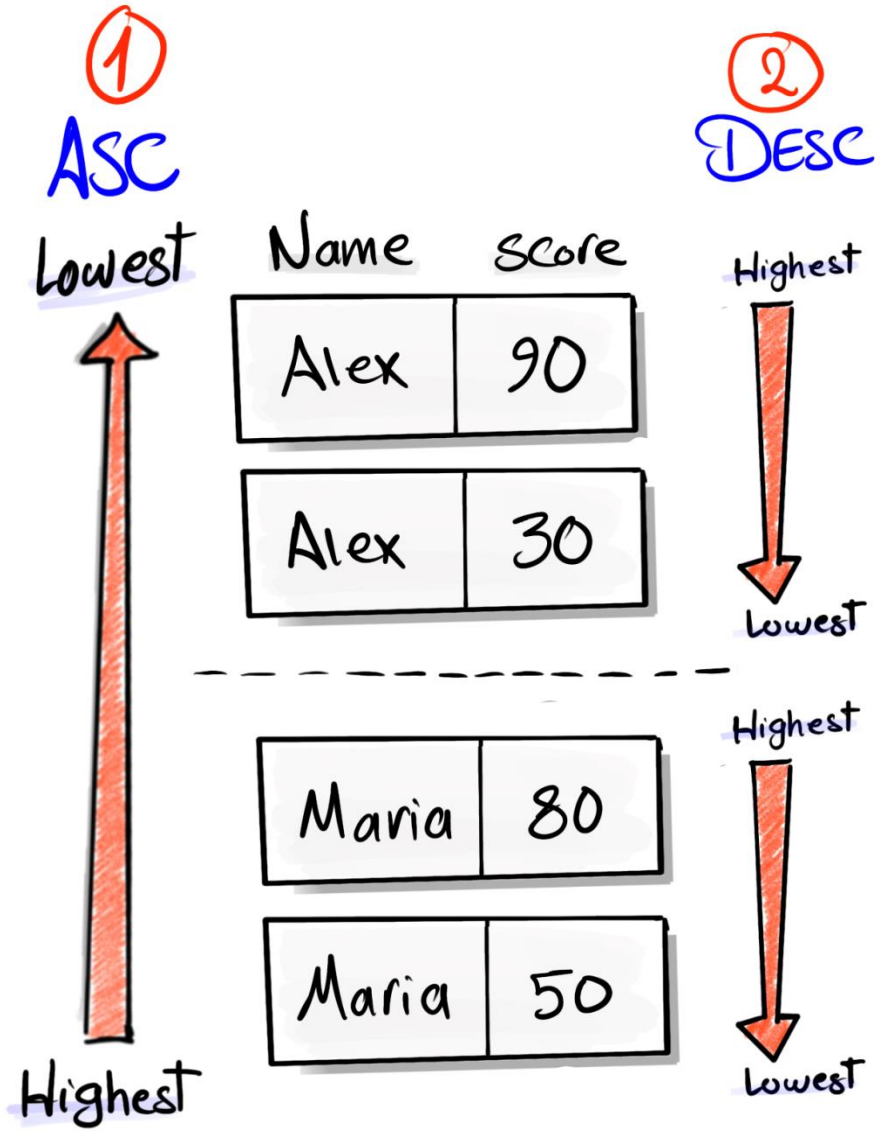
Execute Order

1. FROM
2. WHERE
3. SELECT





Name	score
Alex	90
Maria	50
Alex	30
Maria	80



Database

Customers

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0

① FROM

Result

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0

② WHERE

Σ ③ GROUP BY

Country	Avg
Germany	425
USA	900
Uk	500

④ SELECT

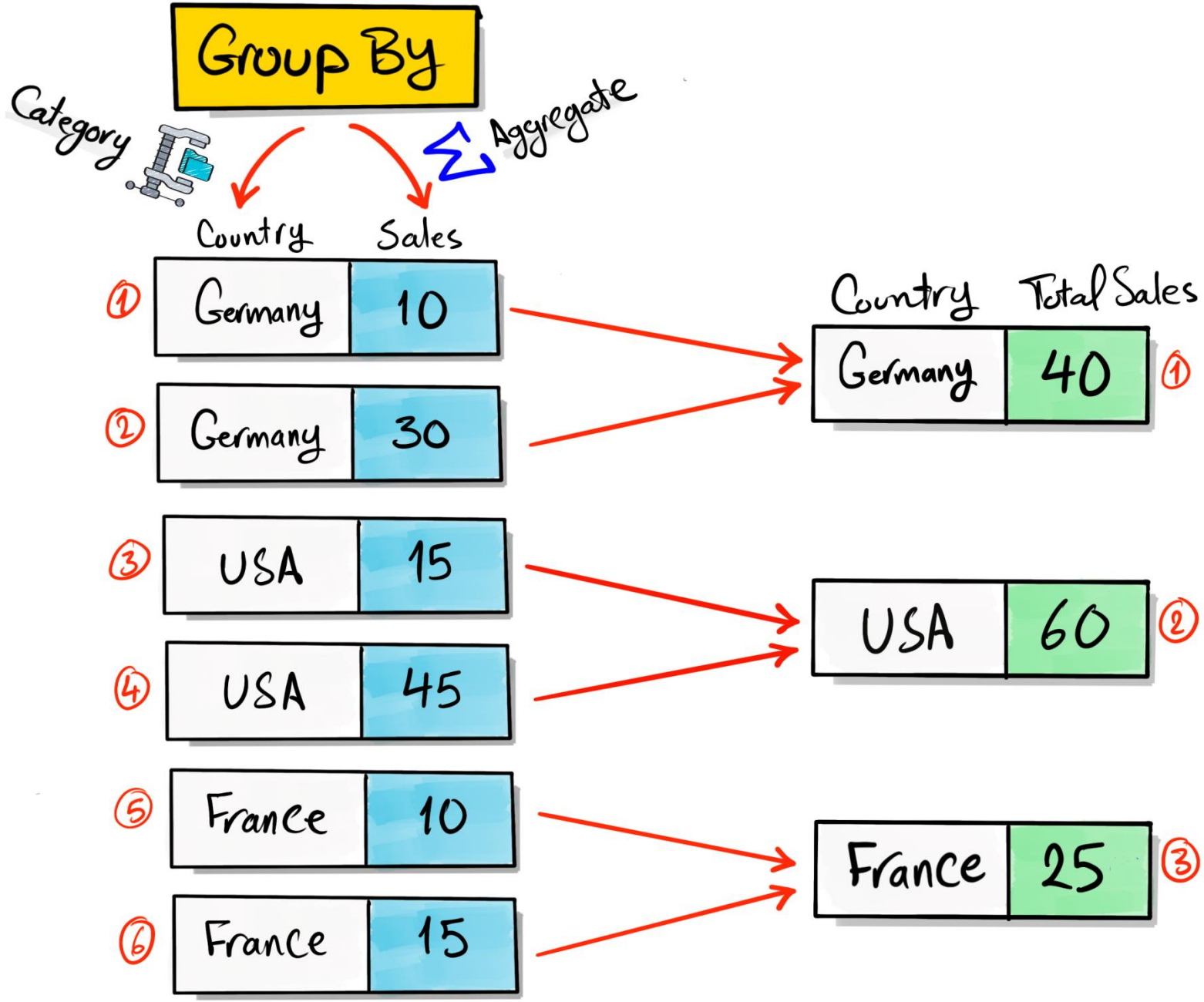
Execute Order

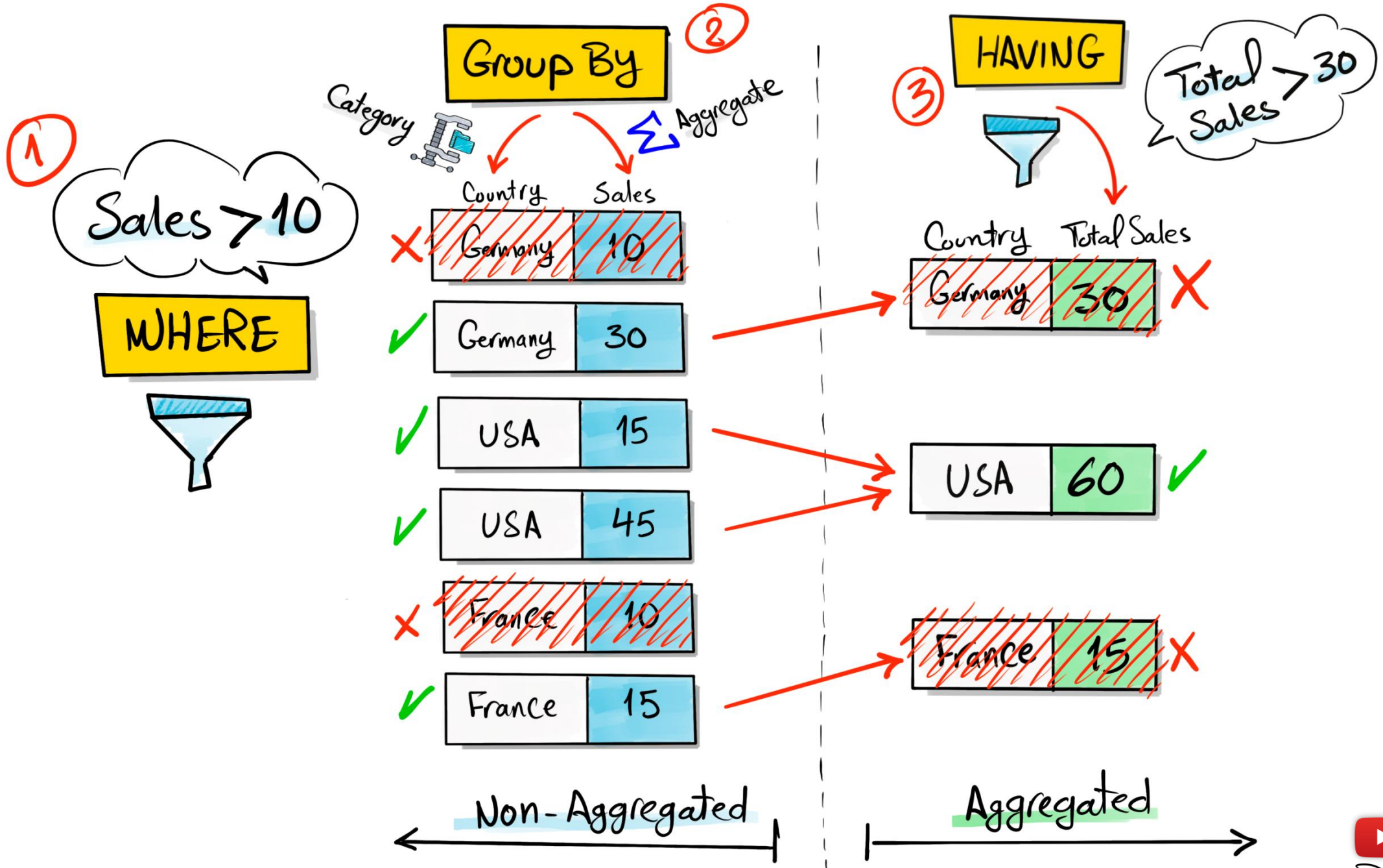
① FROM

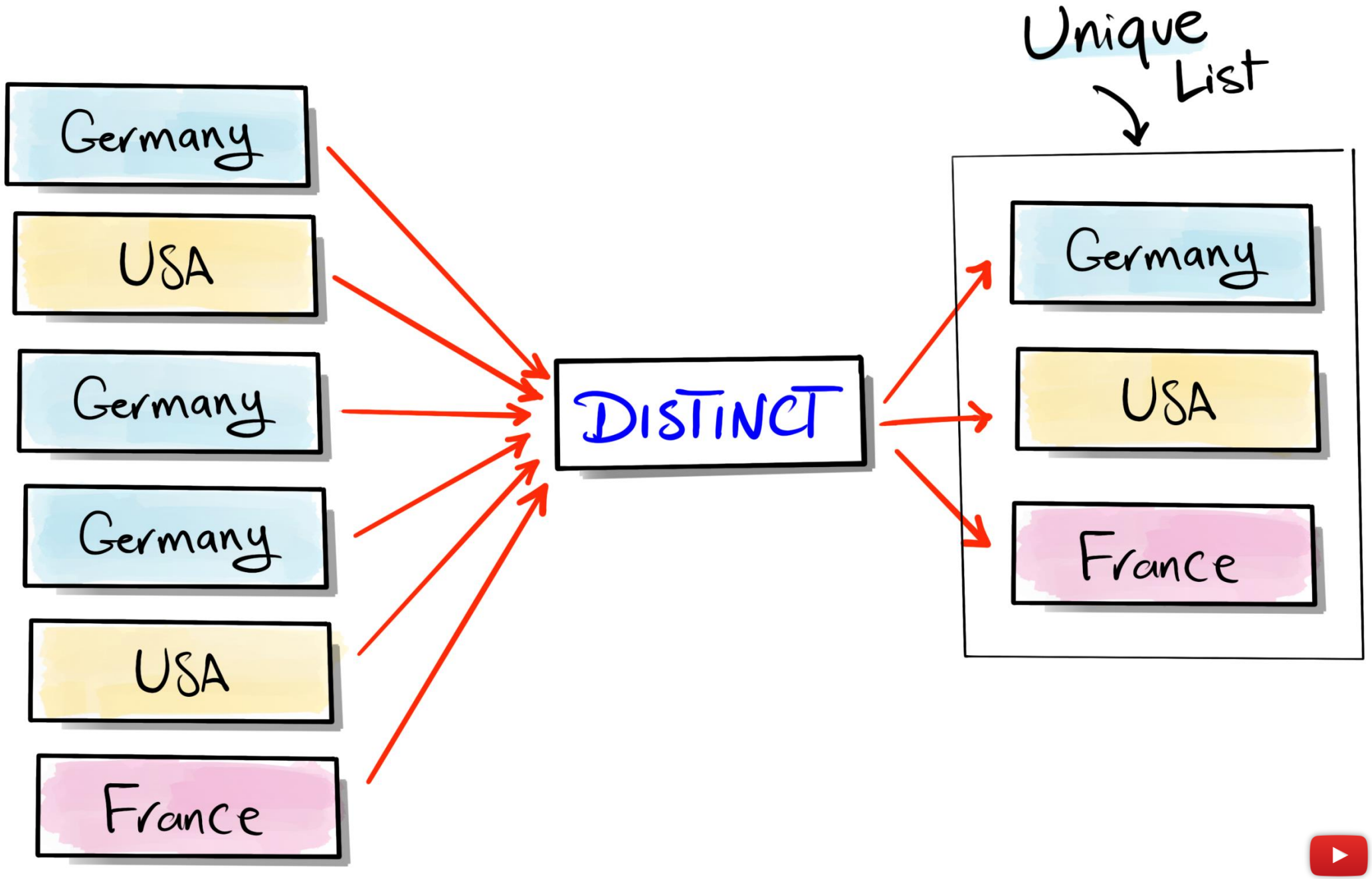
② WHERE

③ GROUP BY

④ SELECT





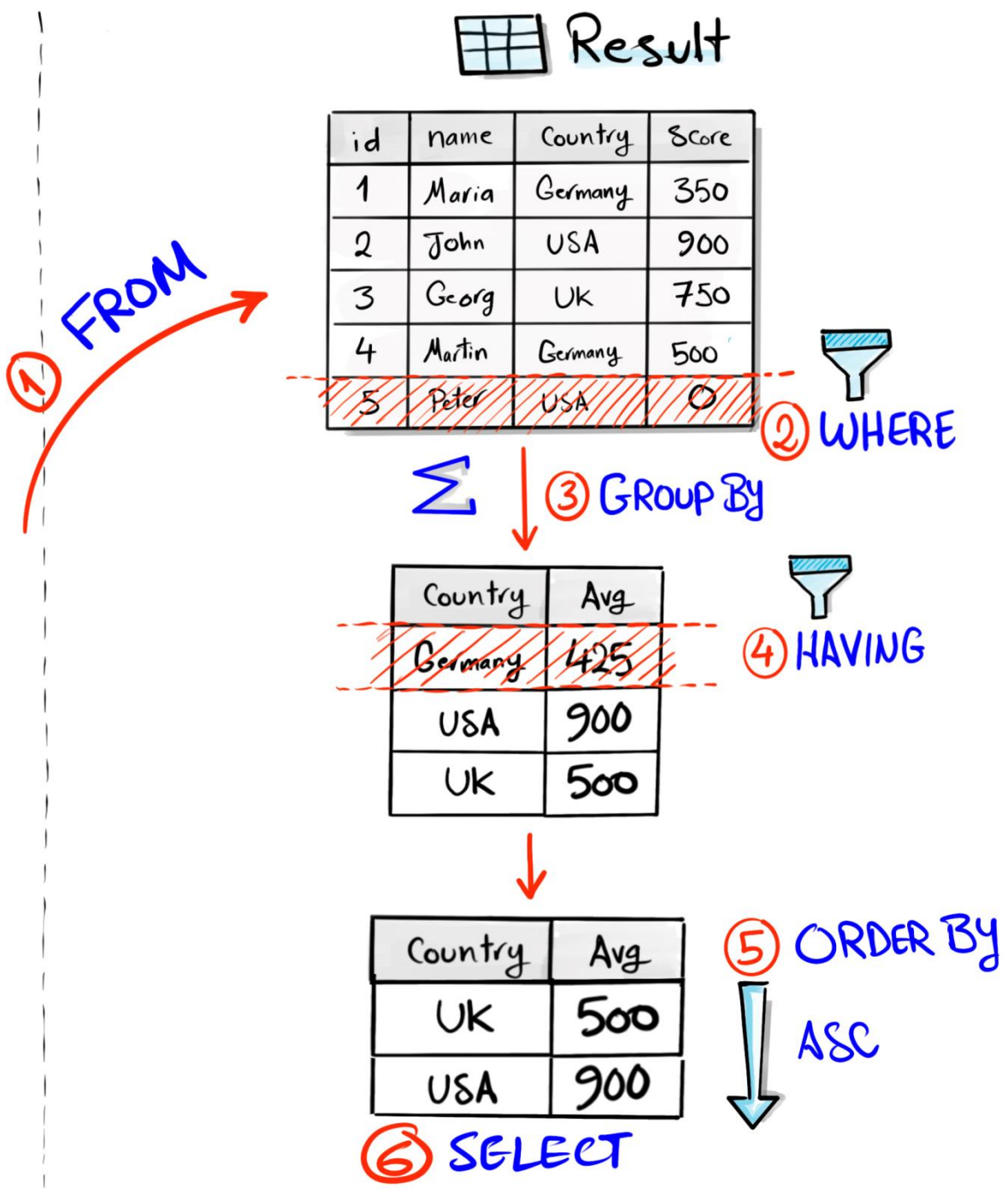


Database

Customers

id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	Uk	750
4	Martin	Germany	500
5	Peter	USA	0

Result

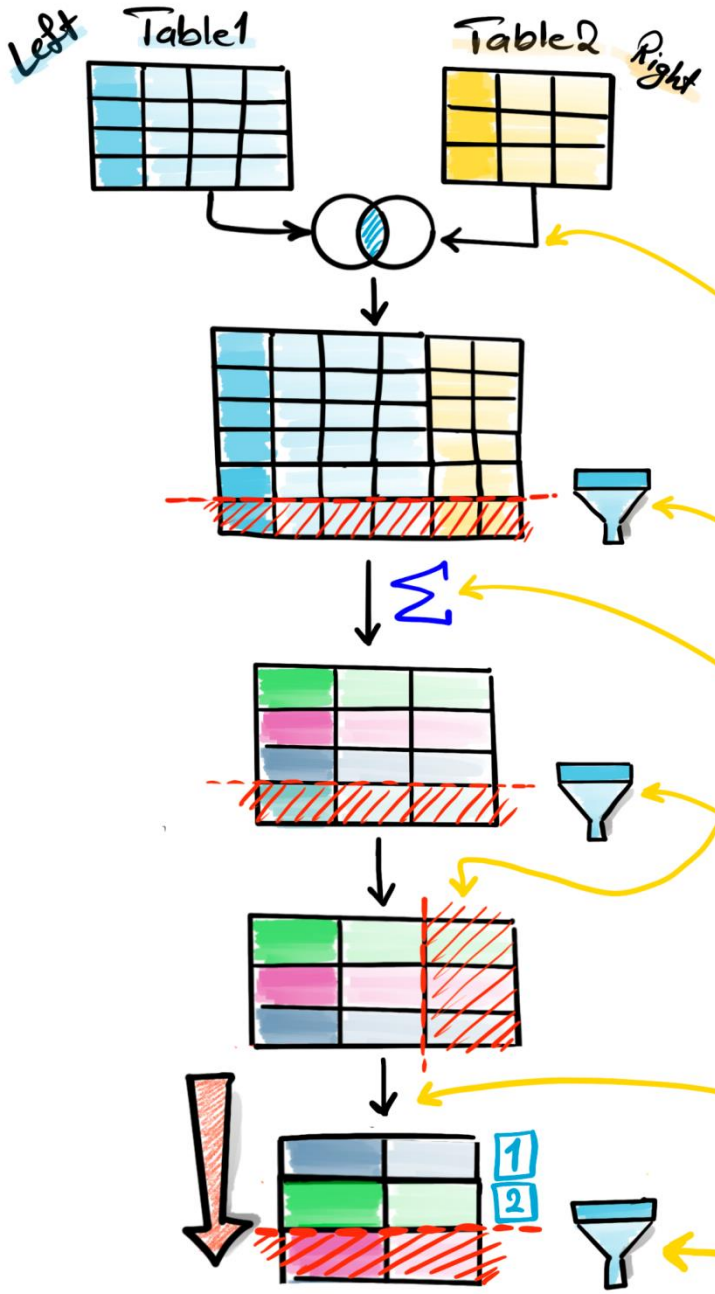


Execute Order

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. ORDER BY
6. SELECT

Execution Order

1. FROM JOIN
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT DISTINCT
6. ORDER BY
7. TOP



Coding Order

5. SELECT DISTINCT
   a.Col,
   SUM(b.Col)

Top 2

7

1. FROM Table1 AS a
   JOIN Table2 AS b
   ON a.id = b.id
2. WHERE a.Col = 10
3. GROUP BY a.Col
4. HAVING SUM(b.Col)
6. ORDER BY a.Col