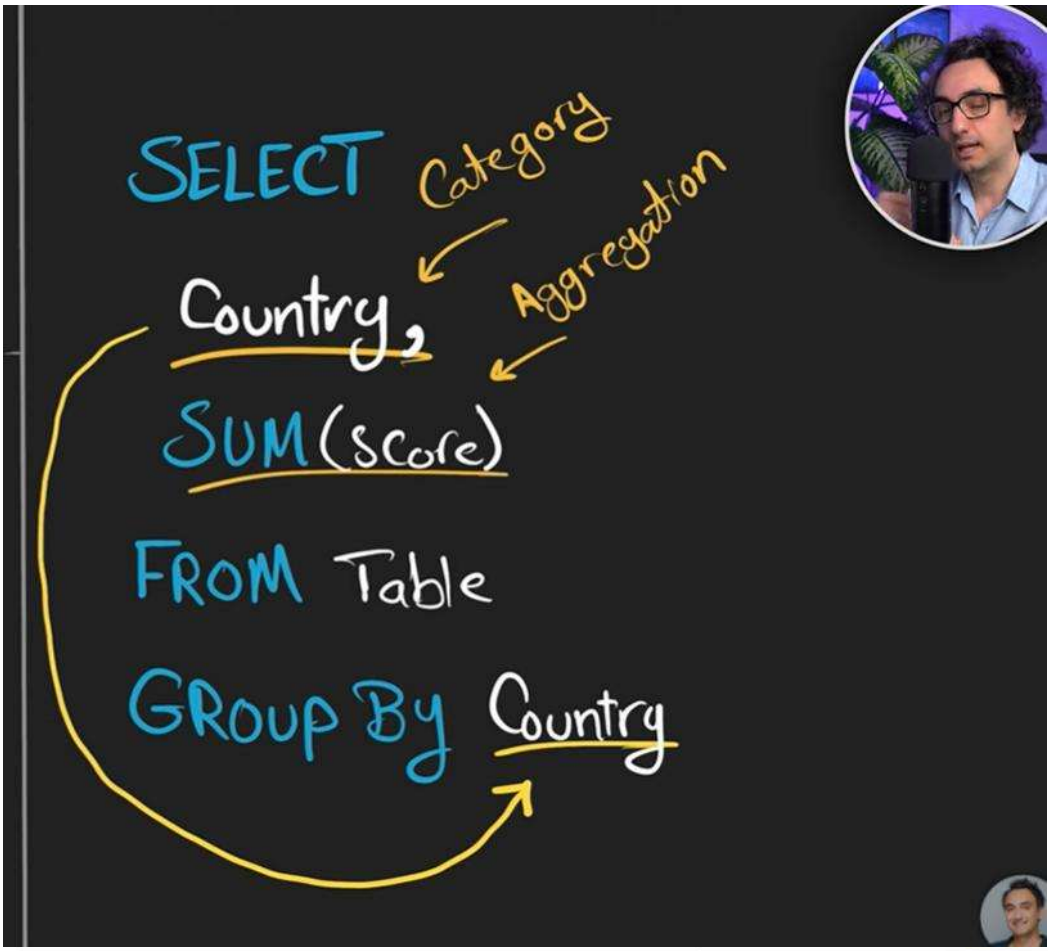
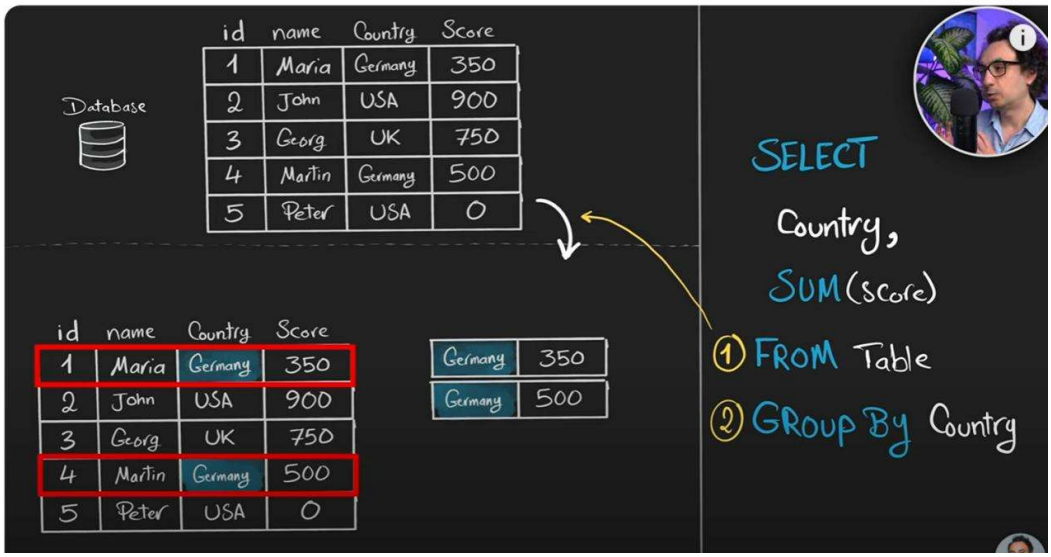
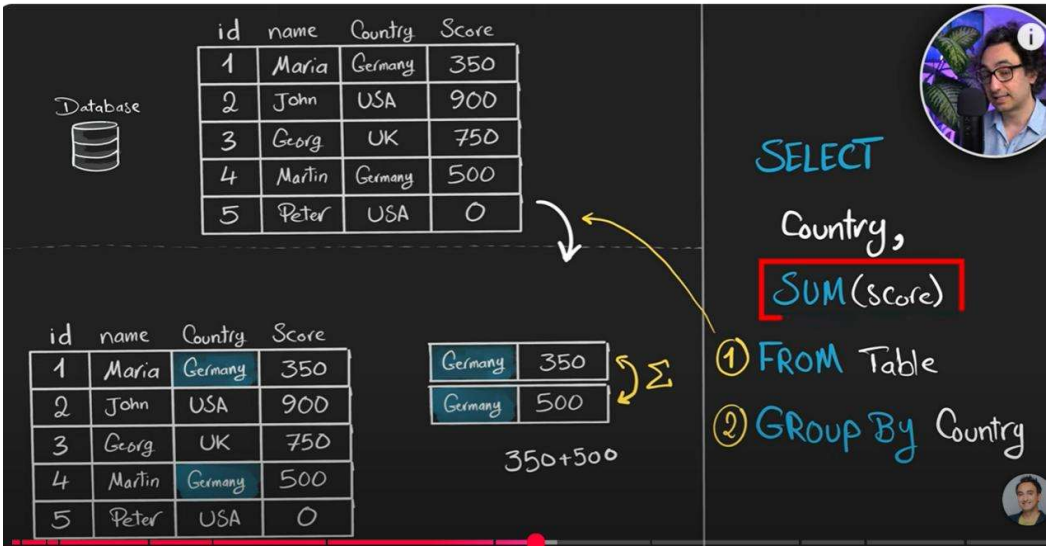
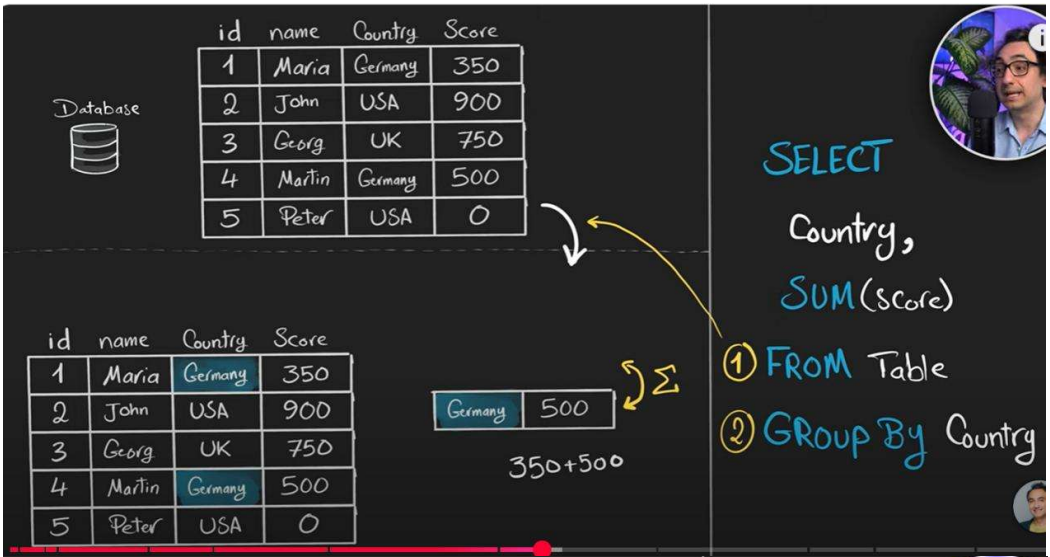
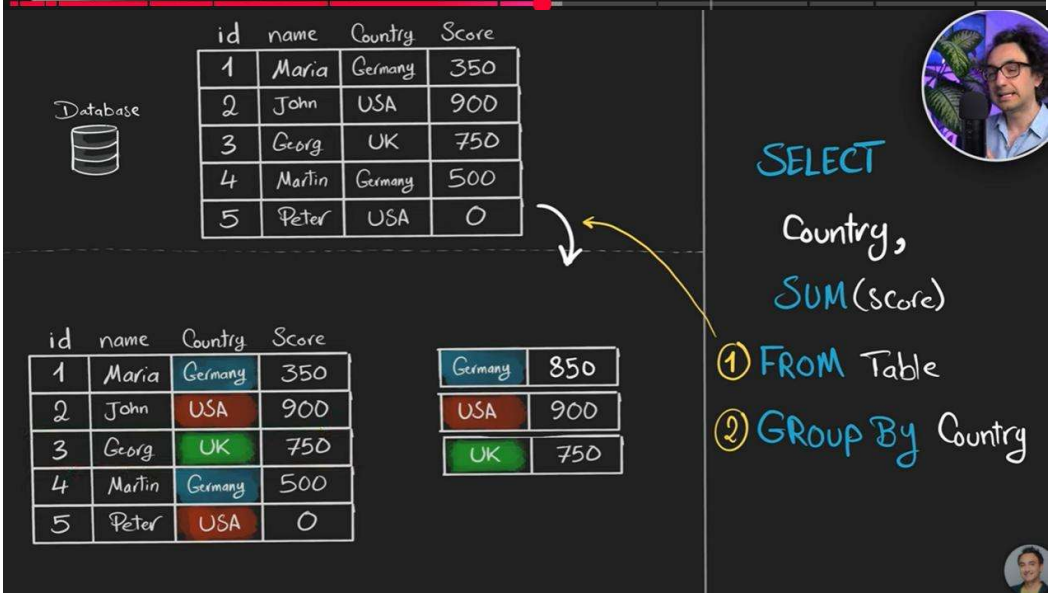
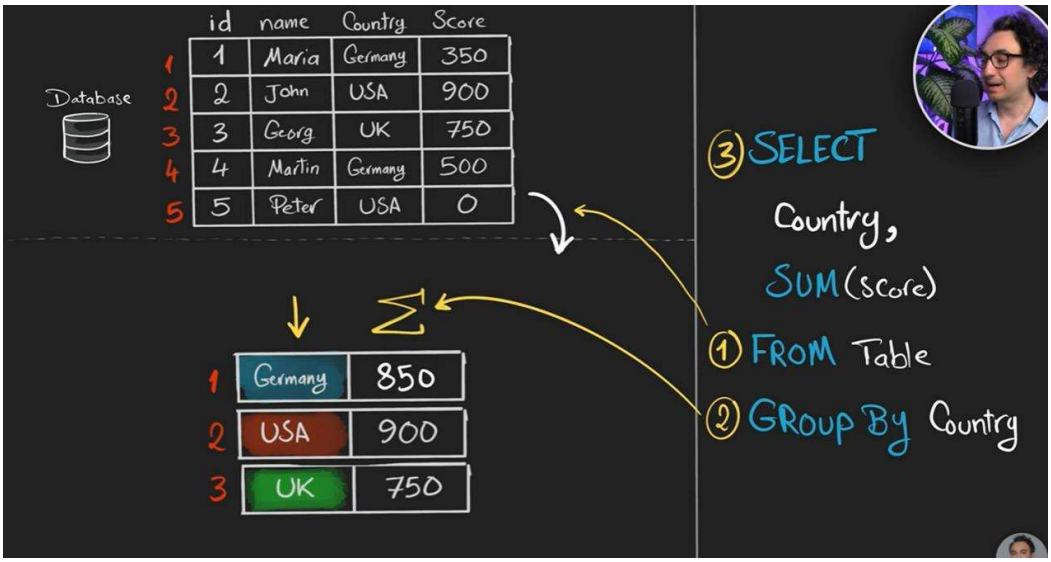

SELECT
Category
Country,
Aggregation
SUM(score)
FROM Table
GROUP BY Country


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
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
Germany 350
Germany 500
SELECT
Country,
SUM(score)
① FROM Table
② GROUP BY Country


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 350
Germany 500
350+500
SELECT
Country,
SUM(score)
1 FROM Table
2 GROUP BY Country


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 500
350+500
SELECT
Country,
SUM(score)
1 FROM Table
2 GROUP By Country

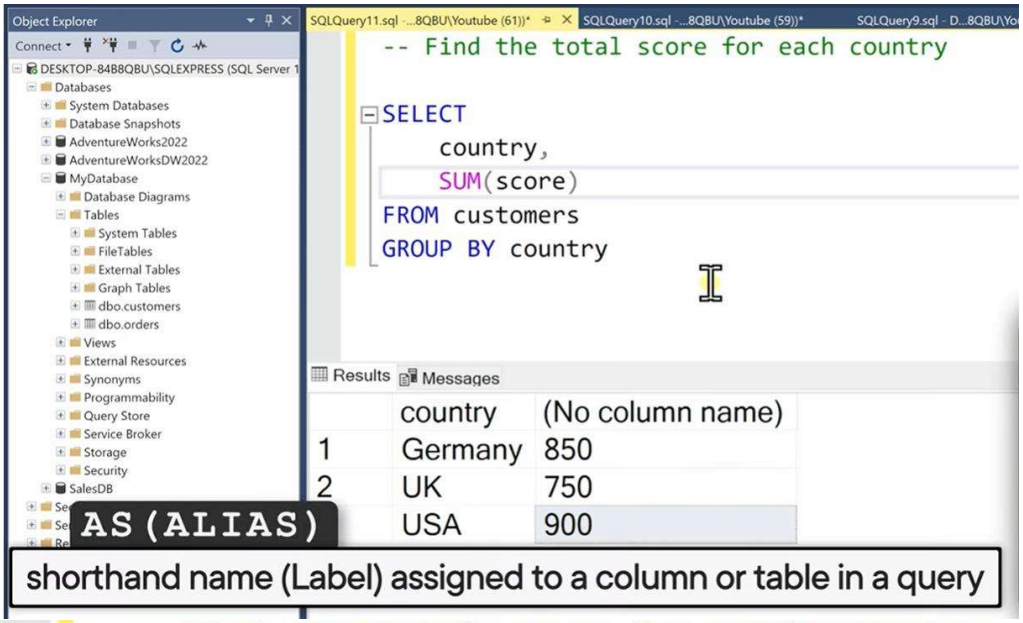

Database
id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
Germany 850
USA 900
UK 750
SELECT
Country,
SUM(score)
1 FROM Table
2 GROUP By Country




```sql
-- Find the total score for each country

SELECT
    country,
    SUM(score)
FROM customers
GROUP BY country
```

Results Messages

	country	(No column name)
1	Germany	850
2	UK	750
3	USA	900



AS (ALIAS)

shorthand name (Label) assigned to a column or table in a query

```sql
-- Find the total score for each country

SELECT
    country,
    SUM(score) AS total_score
FROM customers
GROUP BY country
```

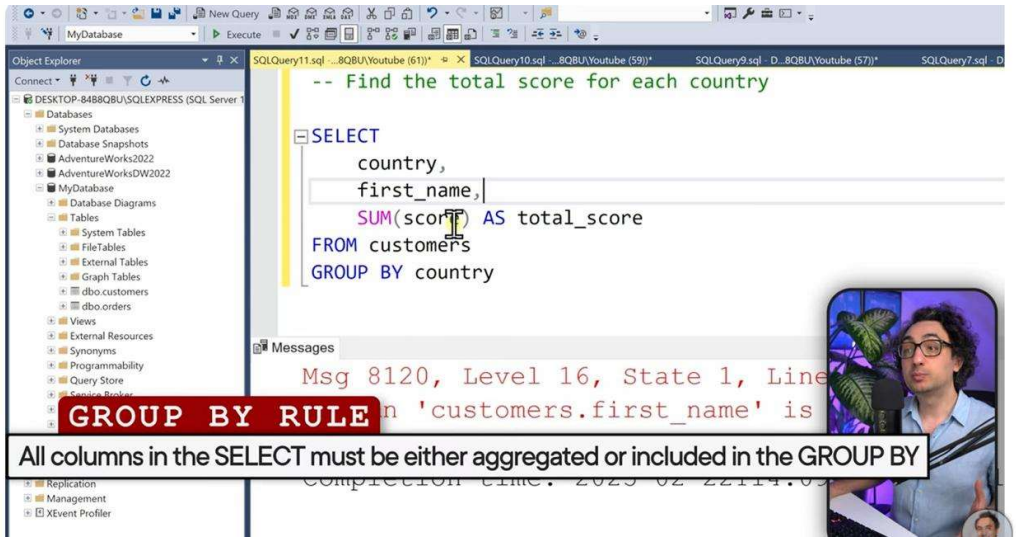
Results | Messages

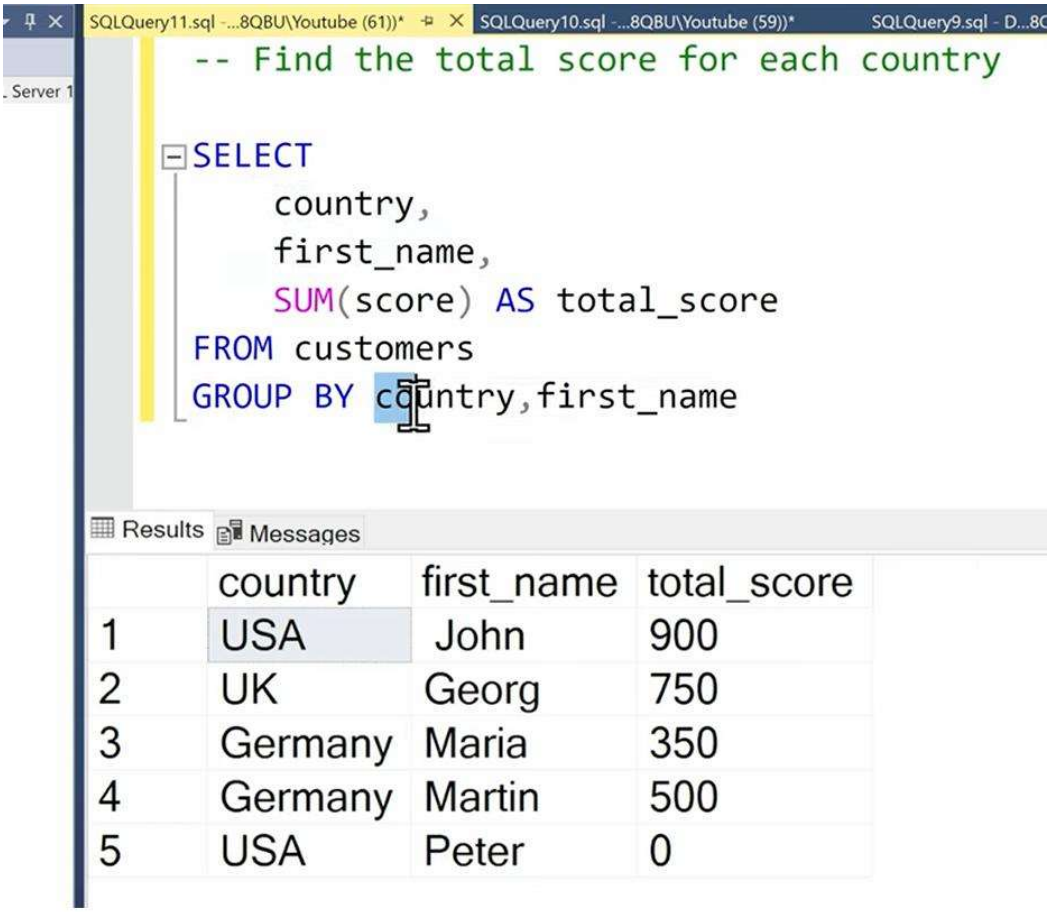
	country	total_score
1	Germany	850
2	UK	750
3	USA	900

```sql
-- Find the total score for each country

SELECT
    country AS customer_country,
    SUM(score) AS total_score
FROM customers
GROUP BY country
```

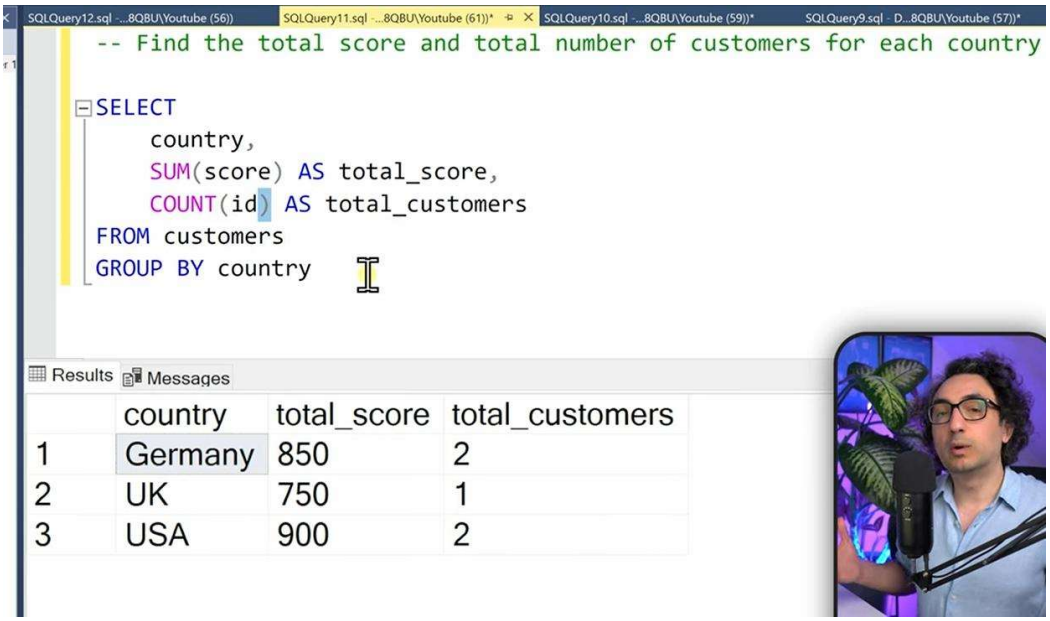
Results | Messages

customer_country	total_score
Germany	850
UK	750
USA	900



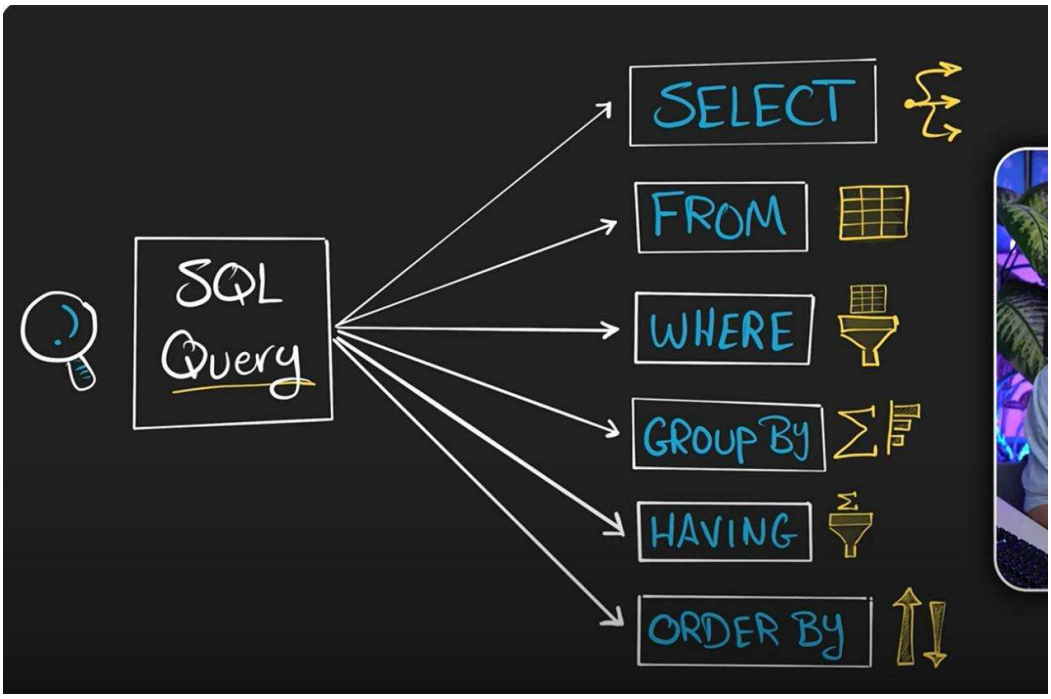


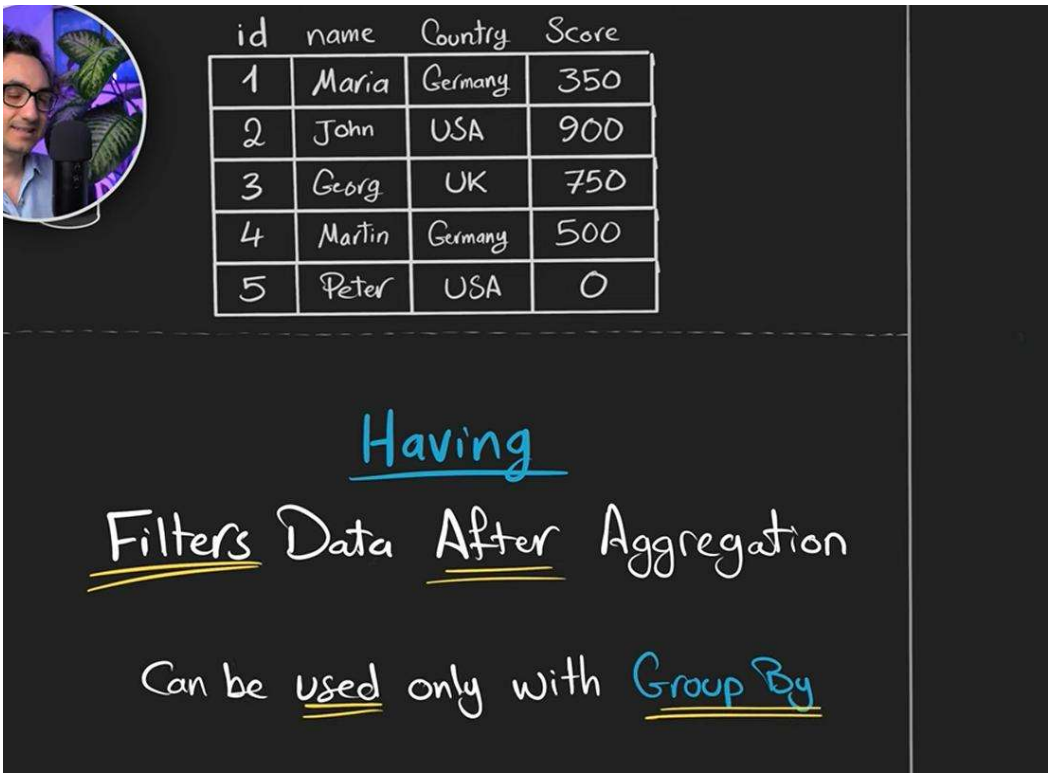
	country	first_name	total_score
1	USA	John	900
2	UK	Georg	750
3	Germany	Maria	350
4	Germany	Martin	500
5	USA	Peter	0

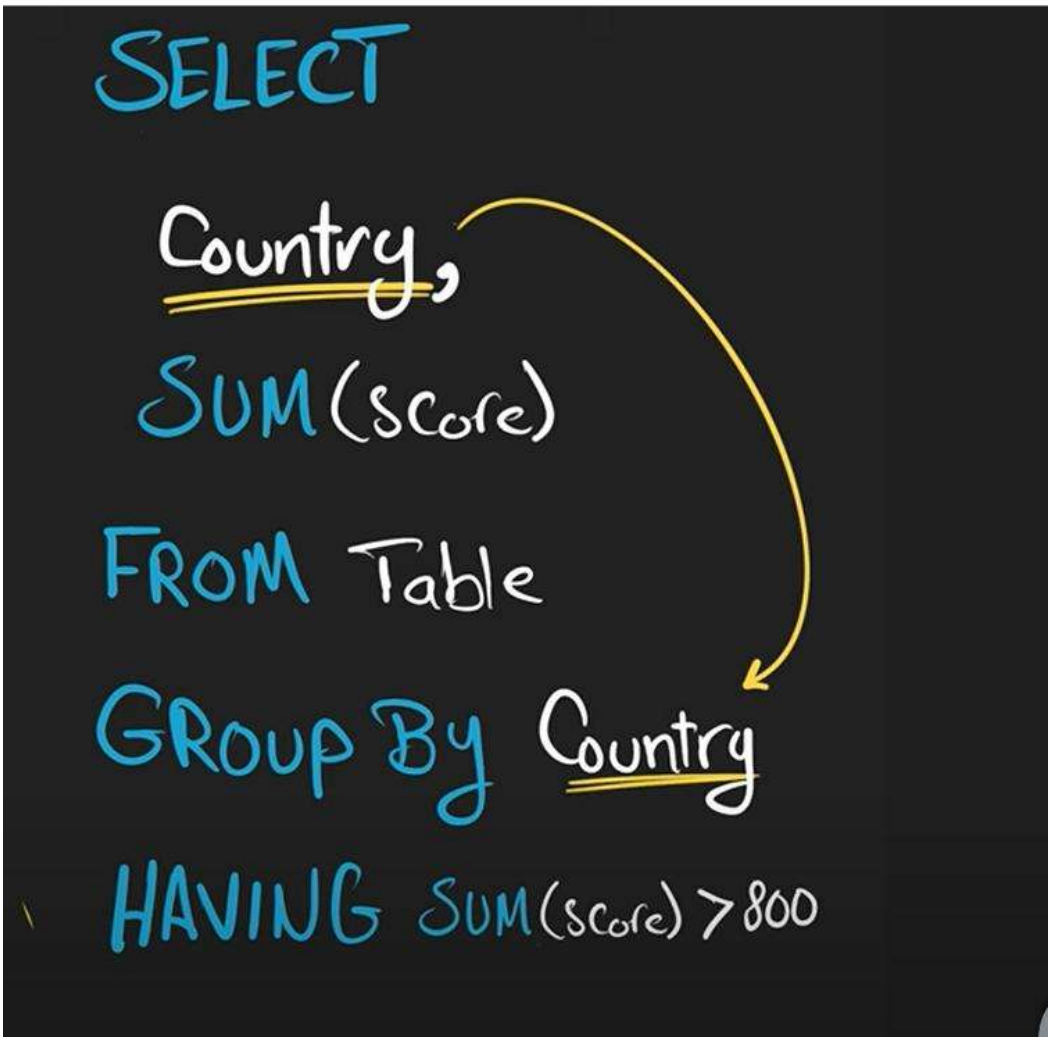


	country	total_score	total_customers
1	Germany	850	2
2	UK	750	1
3	USA	900	2

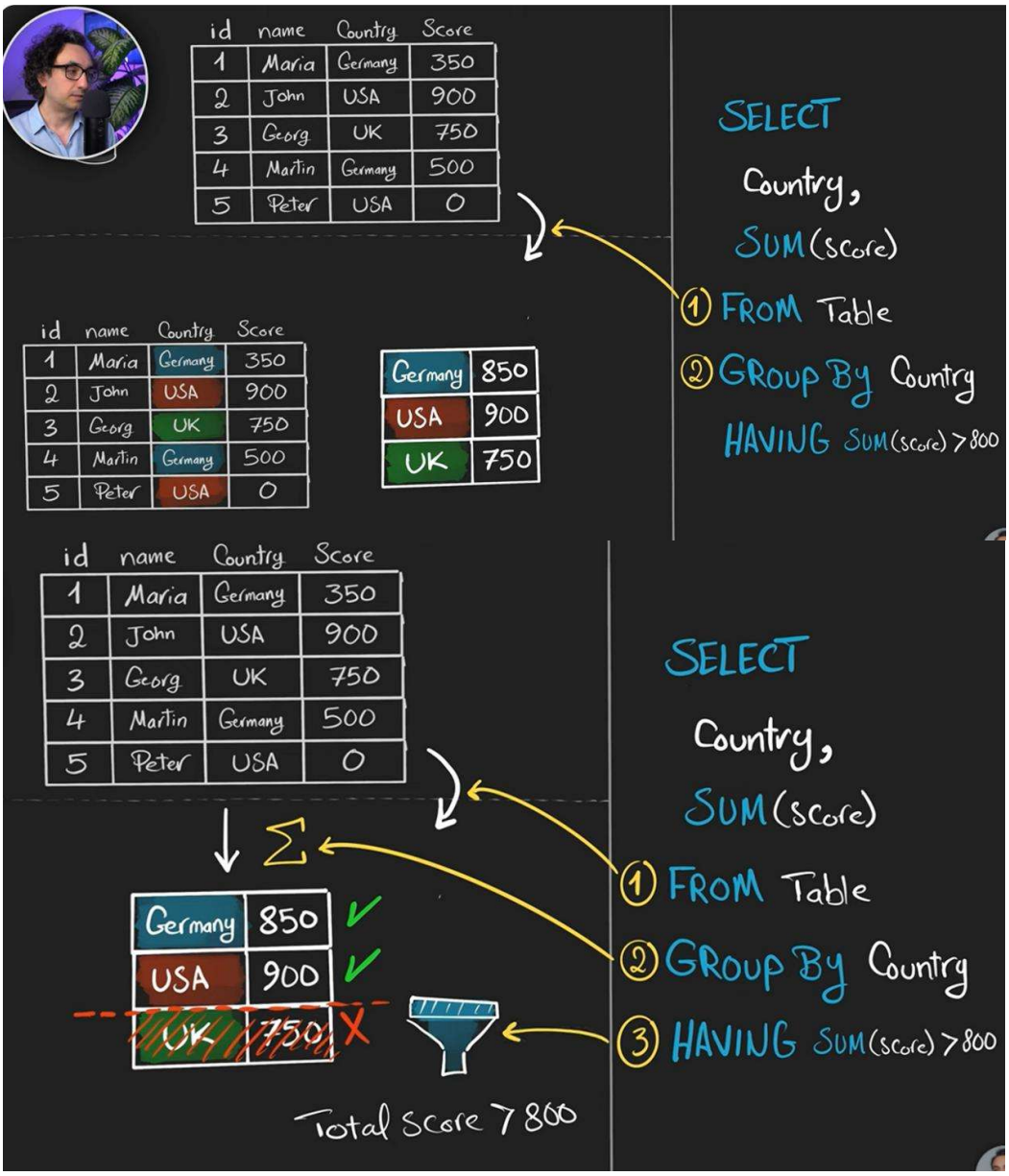
Group by works with aggeregate function without mentioning them into group by clause

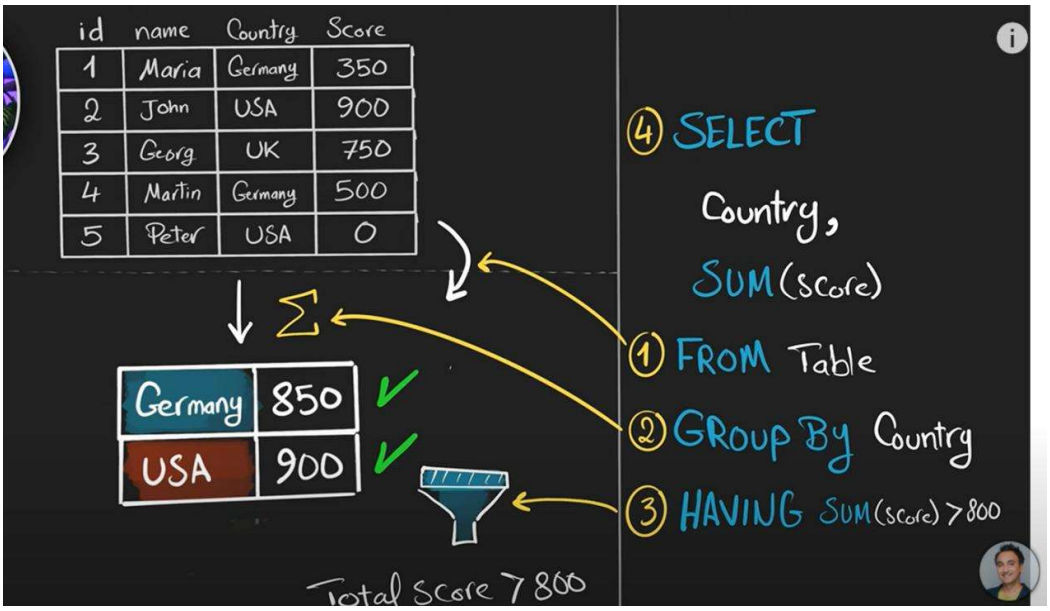

SQL Query
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY


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

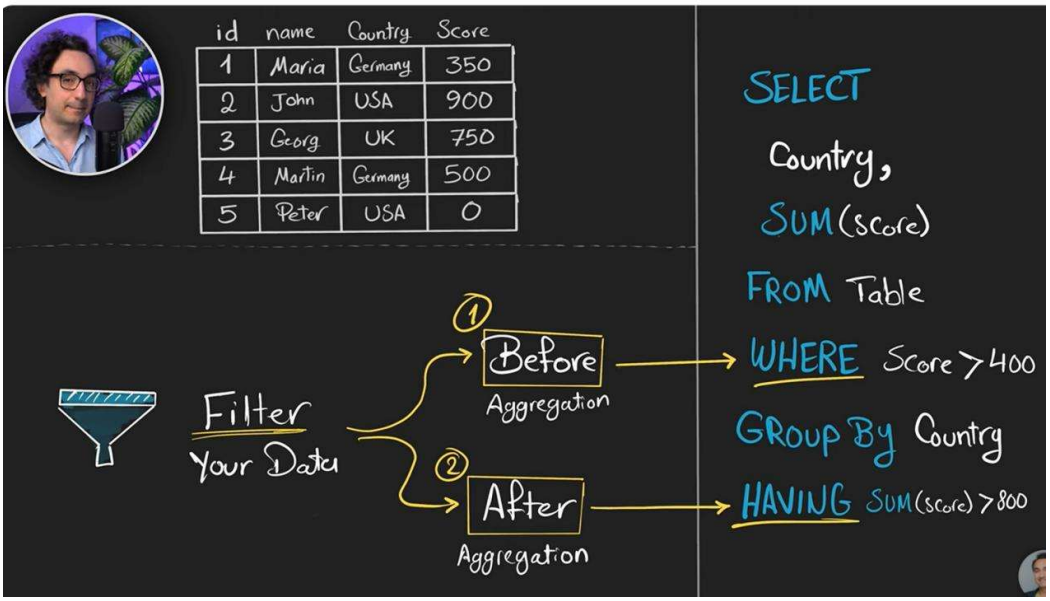

SELECT
Country,
SUM(score)
FROM Table
GROUP BY Country
HAVING SUM(score) > 800

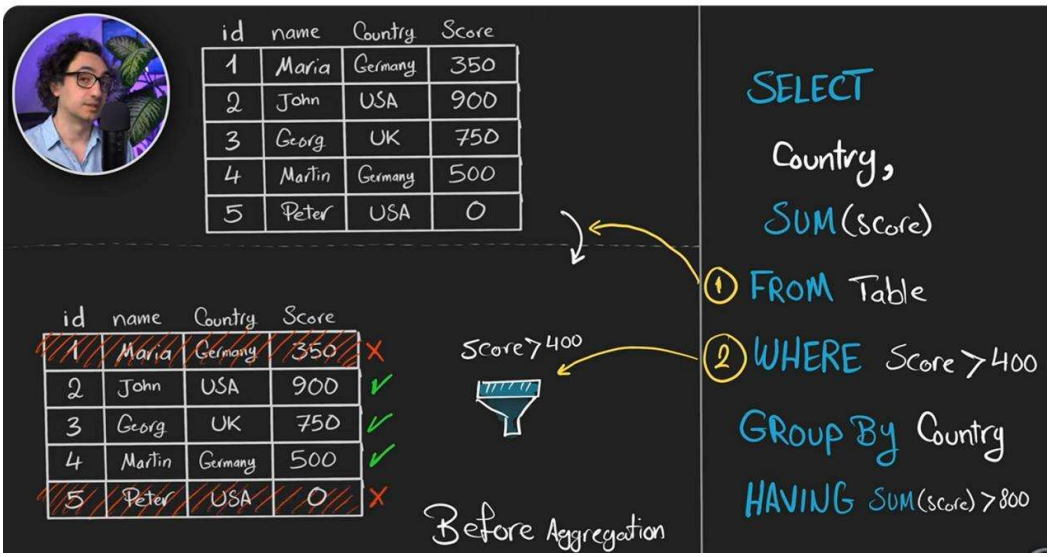

SELECT
Country,
SUM(score)
FROM Table
GROUP BY Country
HAVING Condition

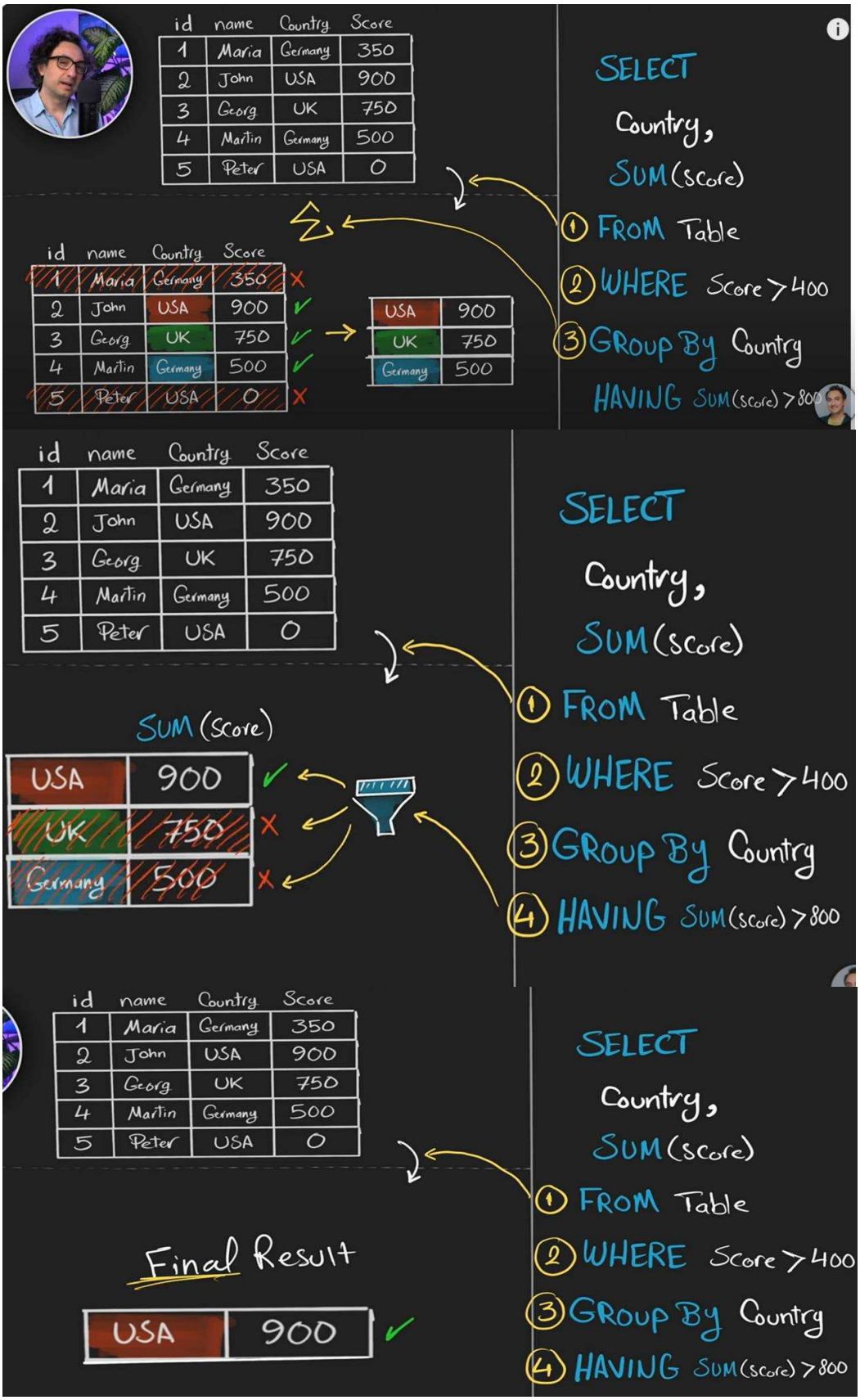

id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
SELECT
Country,
SUM(score)
① FROM Table
② GROUP BY Country
HAVING SUM(score) > 800
Germany 850
USA 900
UK 750
Σ
Total Score > 800


id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
Σ
Germany 850
USA 900
Total Score > 800
4 SELECT
Country,
SUM(score)
1 FROM Table
2 GROUP BY Country
3 HAVING SUM(score) > 800

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

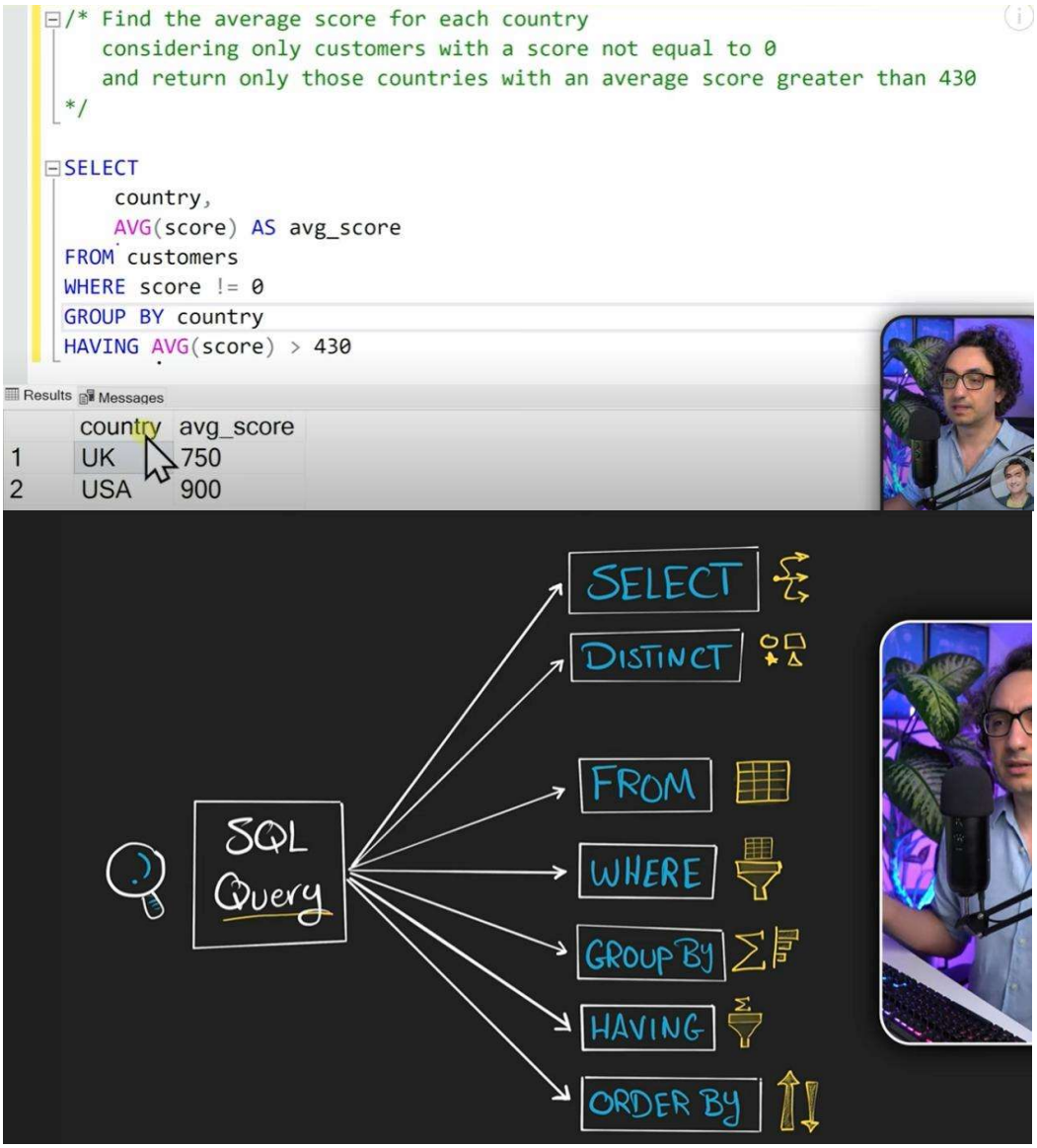

id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
Filter Your Data
1 Before Aggregation
2 After Aggregation
SELECT
Country,
SUM(score)
FROM Table
WHERE Score > 400
GROUP BY Country
HAVING SUM(score) > 800


id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
Score > 400
Before Aggregation
SELECT
Country,
SUM(score)
1 FROM Table
2 WHERE Score > 400
GROUP BY Country
HAVING SUM(score) > 800


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)
1 FROM Table
2 WHERE Score > 400
3 GROUP BY Country
HAVING SUM(score) > 800
USA 900
UK 750
Germany 500
SUM (Score)
Final Result

Having execute after aggregation - works on aggregation functions after group by

Where execute before aggregation



Database

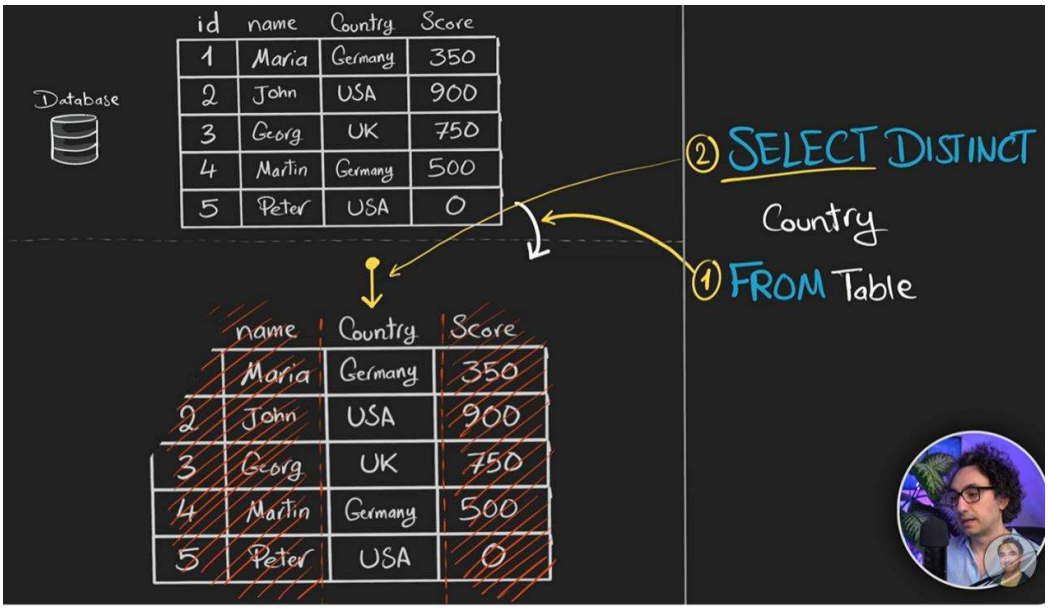
id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

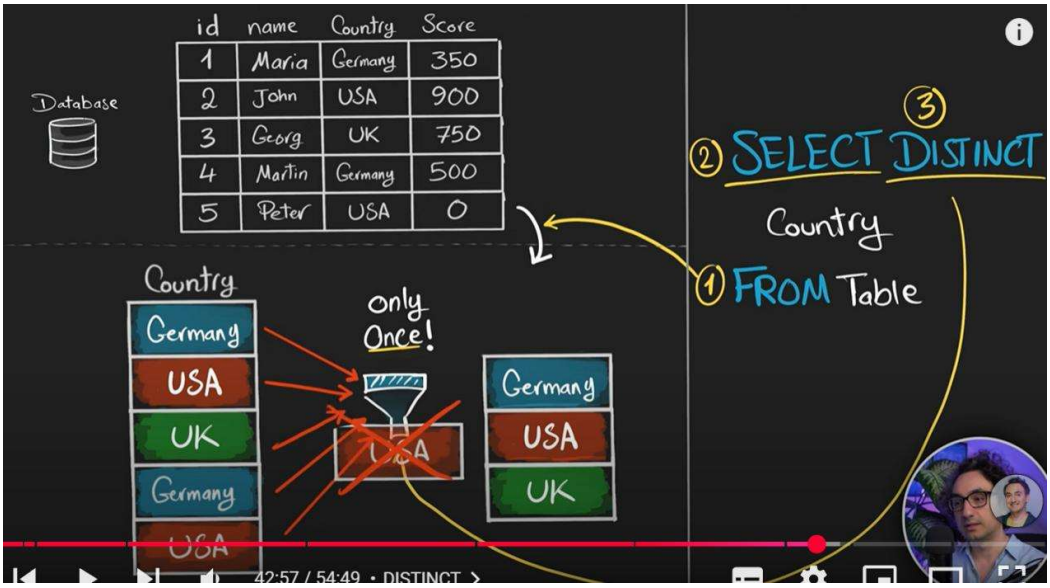
Distinct

Removes Duplicates (Repeated Values)

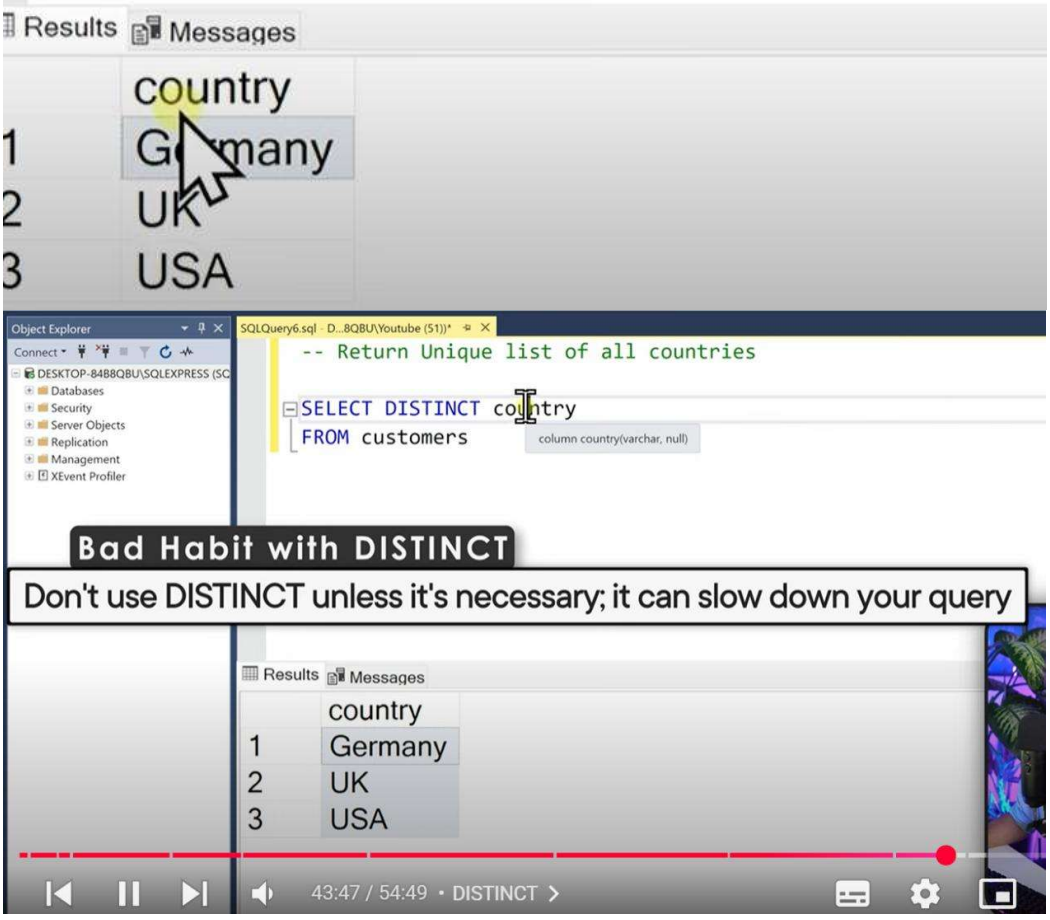
each Value appears only Once

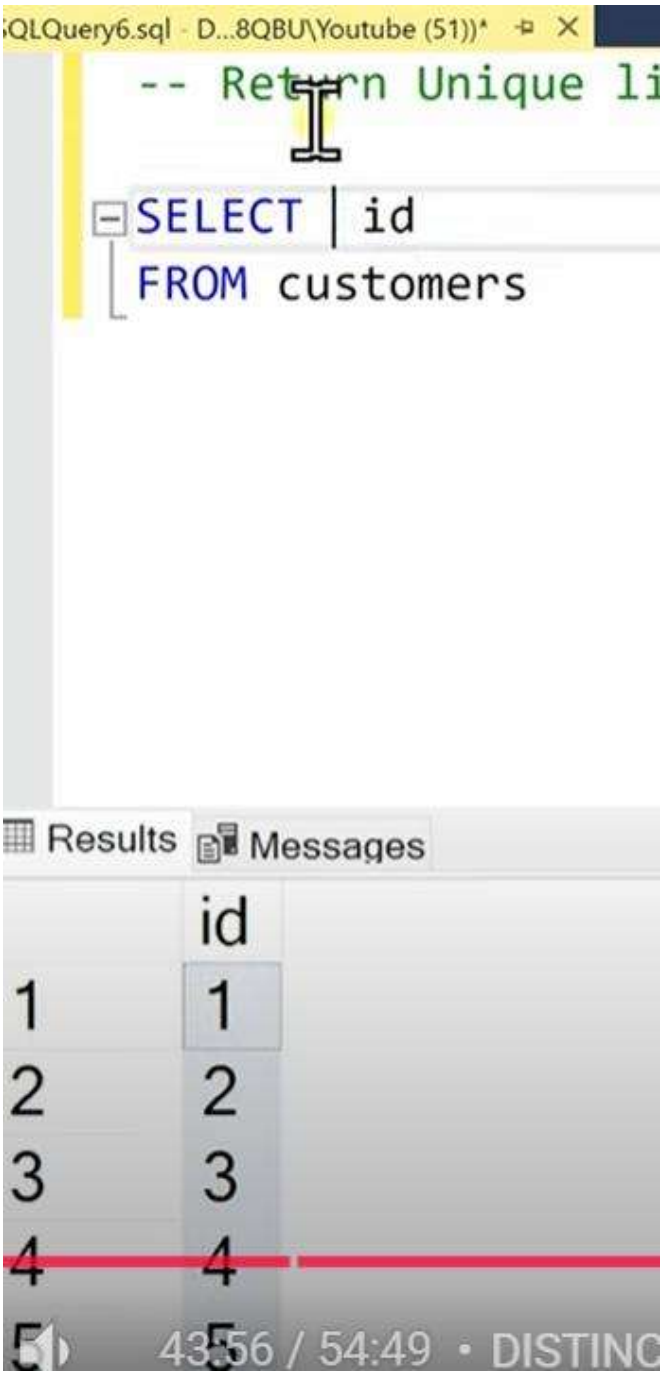
```
SELECT DISTINCT
Col
FROM Table
```

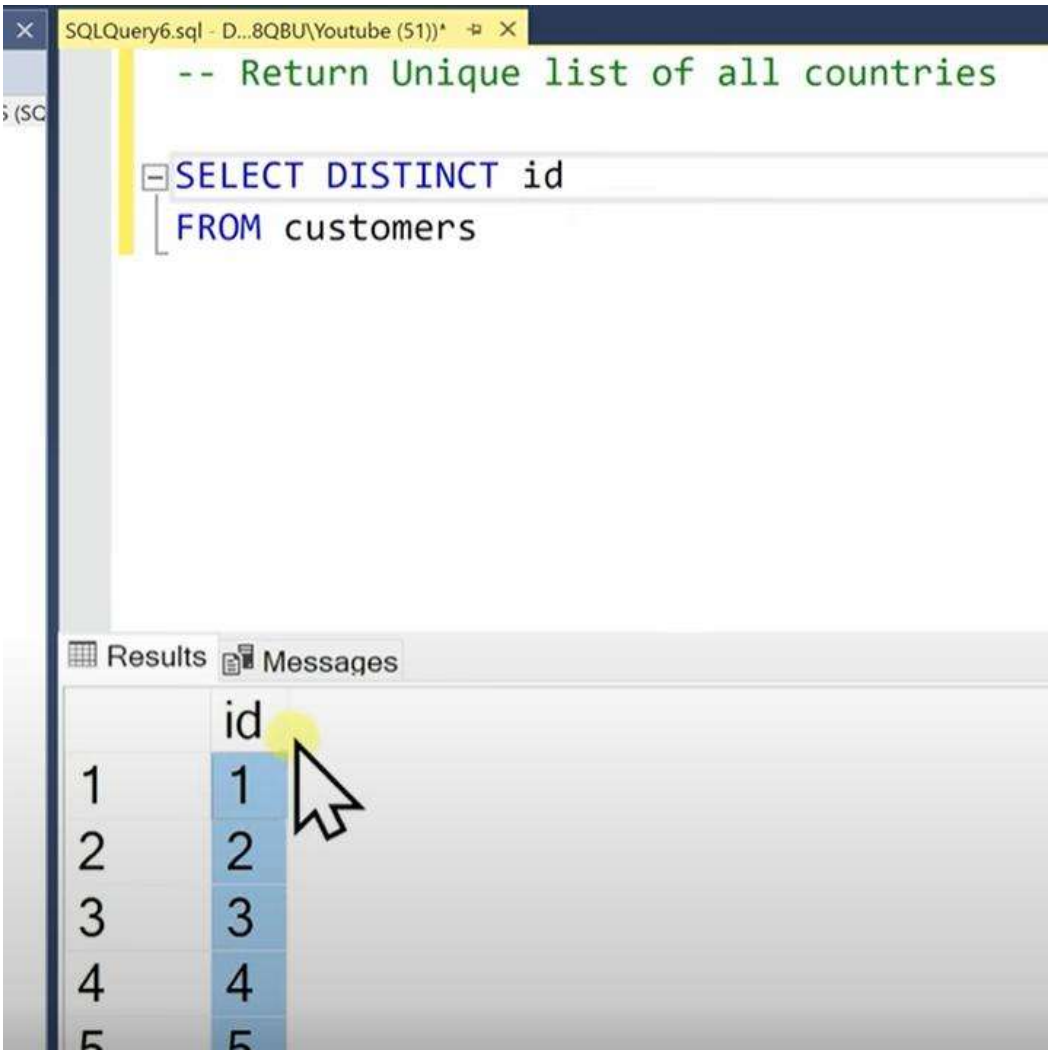

Database
id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
② SELECT DISTINCT
Country
① FROM Table
name Country Score
Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0


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
③
② SELECT DISTINCT
Country
① FROM Table
Country
Germany
USA
UK
Germany
USA
only Once!
Germany
USA
UK
42:57 / 54:49 • DISTINCT

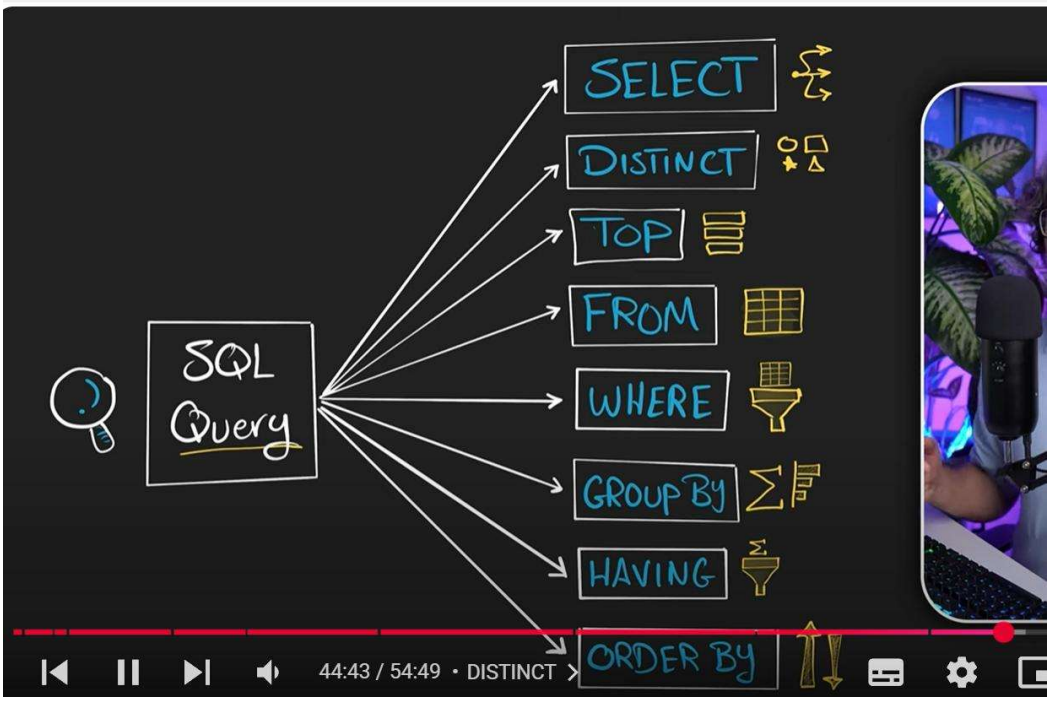
```sql
-- Return Unique list of all countries

SELECT DISTINCT country
FROM customers
```



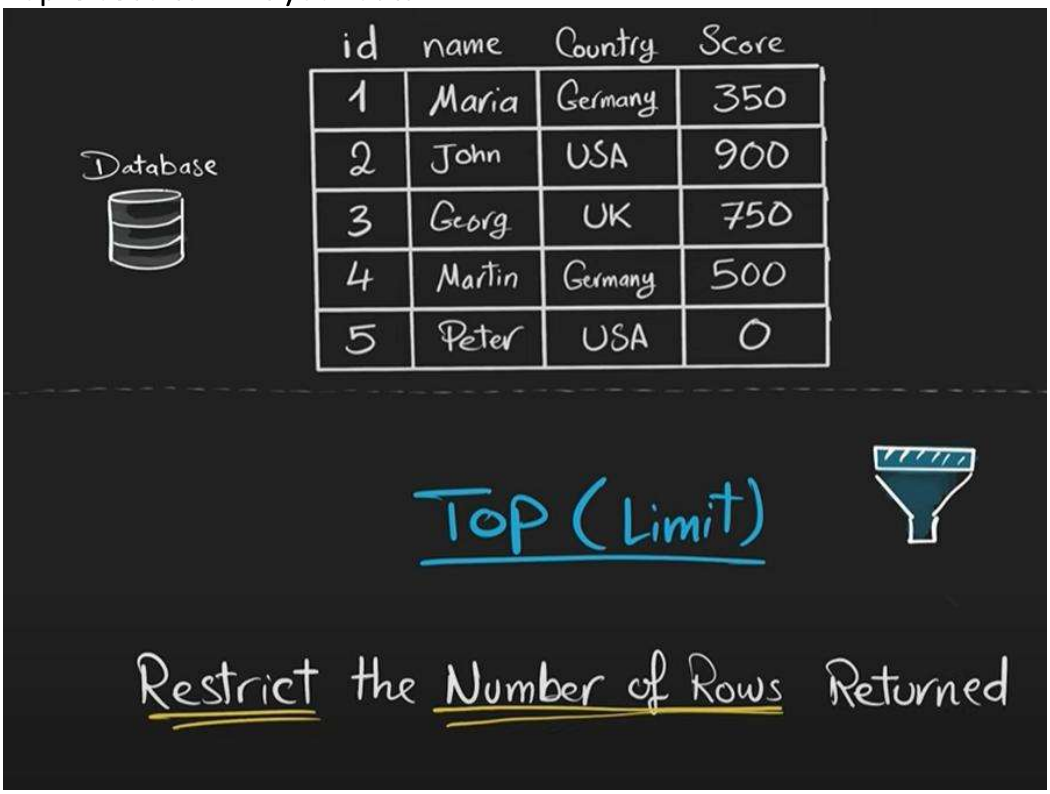

QLQuery6.sql - D...8QBU\Youtube (51))*
-- Return Unique li
SELECT id
FROM customers
Results
Messages
id
1 1
2 2
3 3
4 4
5 5
43:56 / 54:49 • DISTINC



Distinct makes the query too slow so do not used when not required



Top is used to limit your data



SELECT Top 3 *

FROM Table

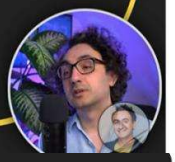
Database

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

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
Row 4 →	4	Martin	Germany	500
Row 5 →	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

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

```sql
-- Retrieve the Top 3 Customers with the Highest Scores

SELECT TOP 3 *
FROM customers
ORDER BY score DESC
```

Results | Messages

	id	first_name	country	score
1	2	John	USA	900
2	3	Georg	UK	750
3	4	Martin	Germany	500

```sql
-- Retrieve the Lowest 2 Customers based on the score

SELECT TOP 2 *
FROM customers
ORDER BY score ASC
```

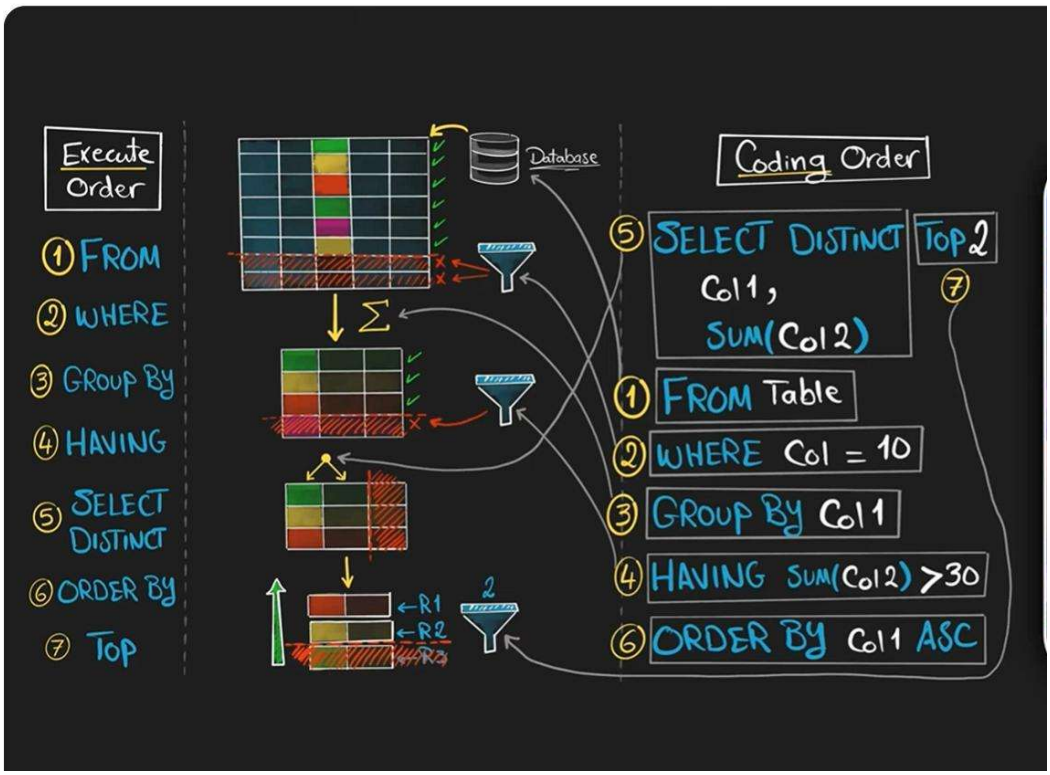
Results | Messages

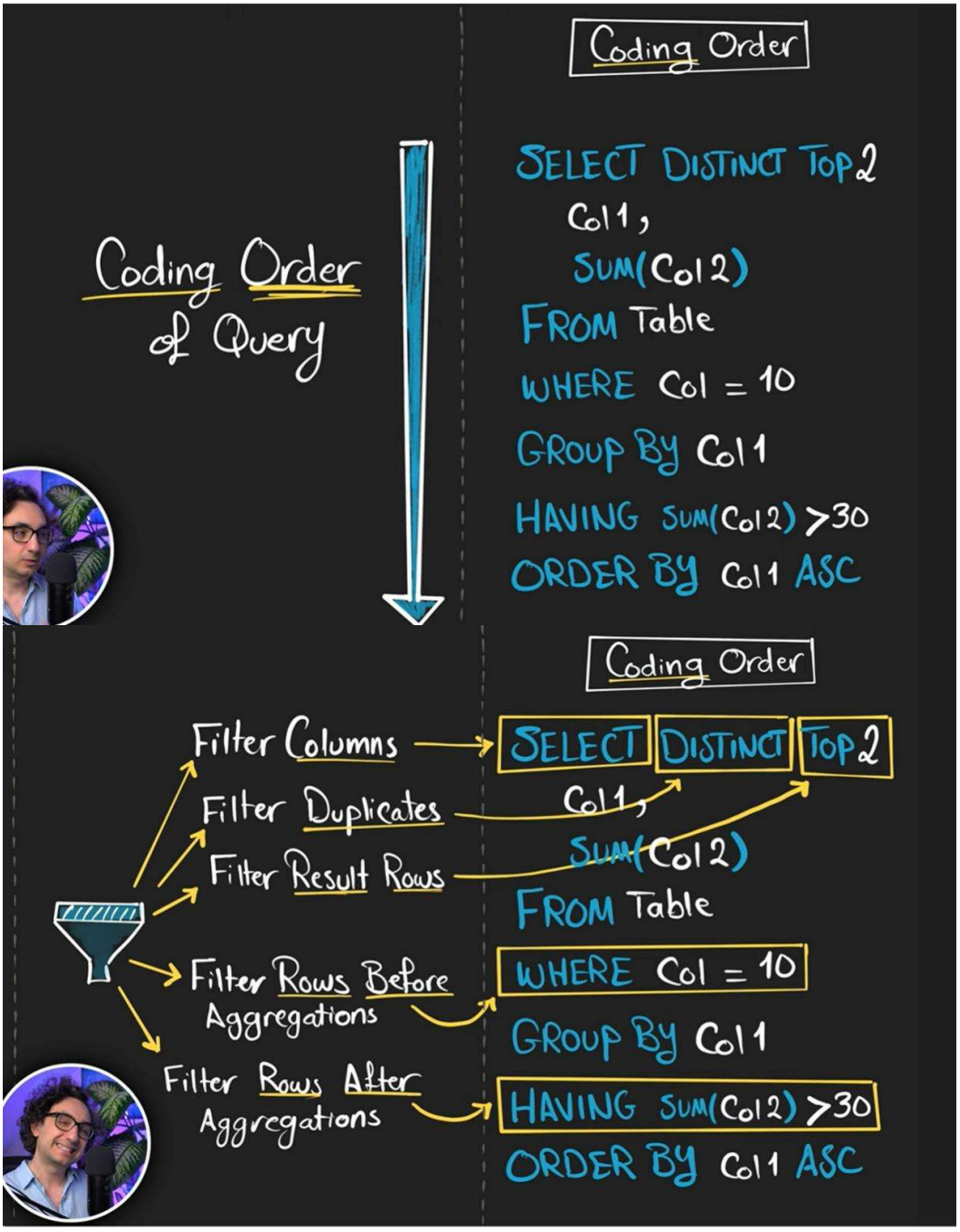
	id	first_name	country	score
1	5	Peter	USA	0
2	1	Maria	Germany	350

```sql
-- Get the Two Most Recent Orders

SELECT TOP 2 *
FROM orders
ORDER BY order_date DESC
```

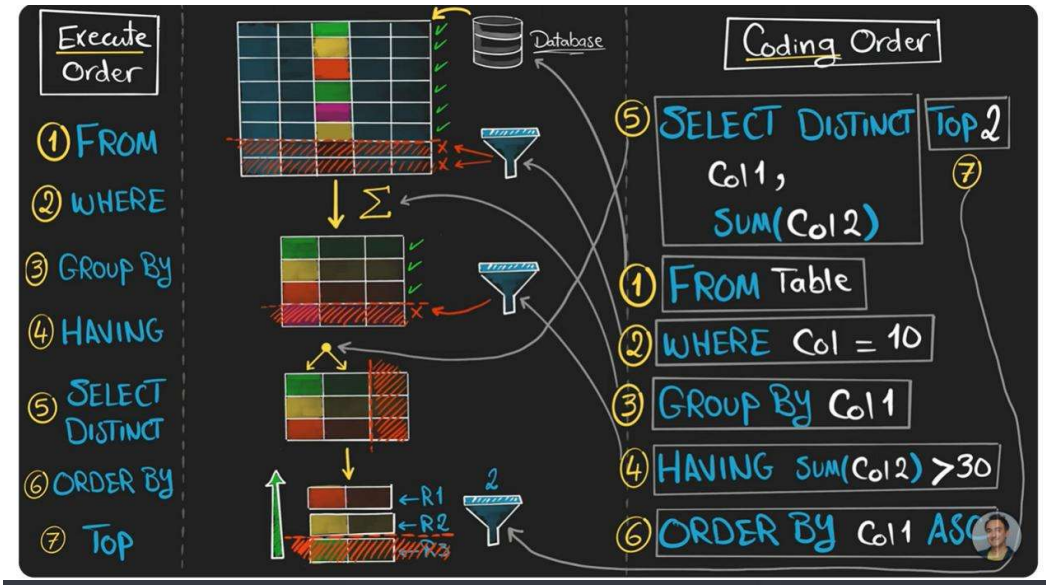
Results | Messages

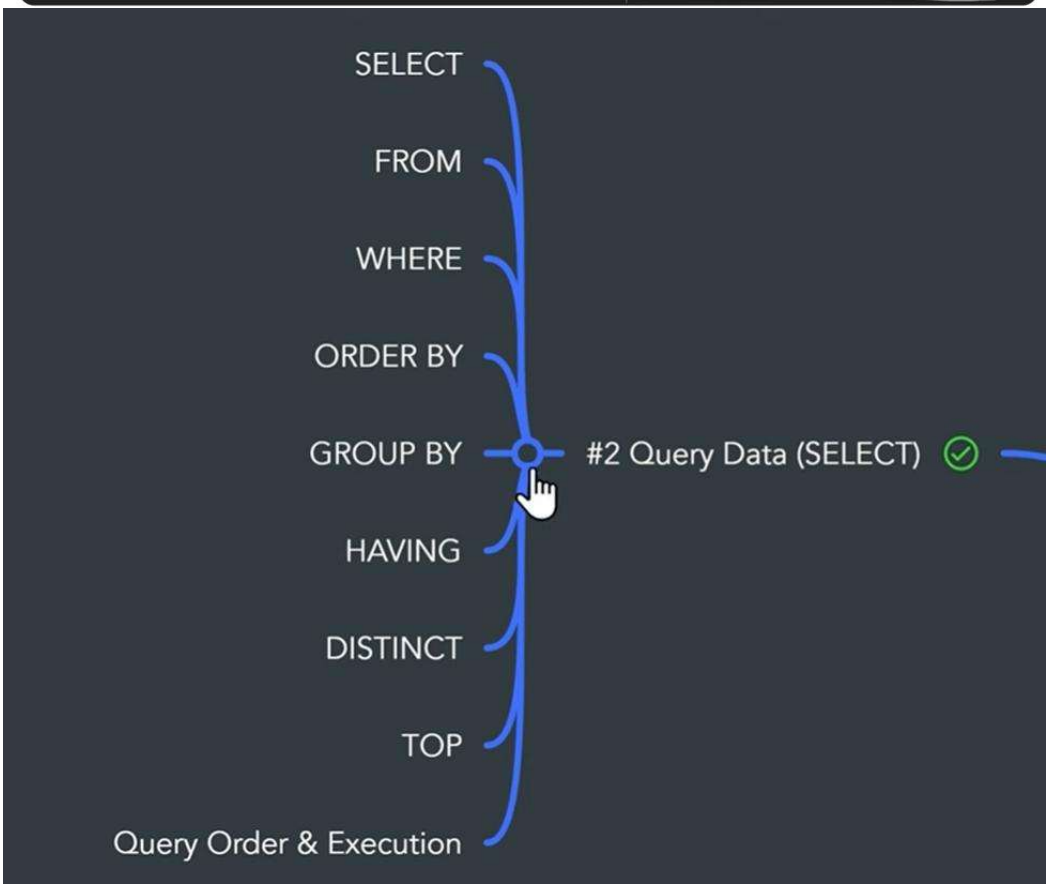
	order_id	customer_id	order_date	sales
1	1004	6	2021-08-31	10
2	1003	3	2021-06-18	20





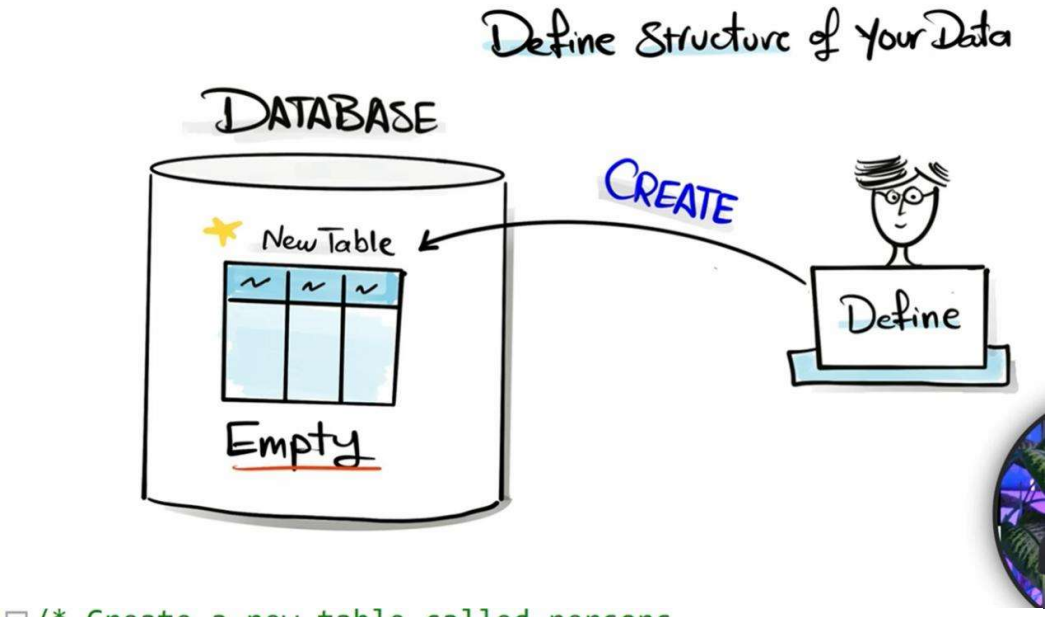
SQL SELECT Queries: All Essential Clauses Explained for Beginners | #SQL Cou





Complete This

DDL Commands




```
/* Create a new table called persons
with columns: id, person_name, birth_date, and phone */

CREATE TABLE persons (
    id INT NOT NULL
)
```

Column Name | Data Type | Constraint

```sql
/* Create a new table called persons
with columns: id, person_name, birth_date, and phone */

CREATE TABLE persons (
    id INT NOT NULL,
    person_name VARCHAR(50) NOT NULL,
    birth_date DATE,
    phone VARCHAR(15) NOT NULL,
    CONSTRAINT pk_persons PRIMARY KEY (id)
)
```

192 %

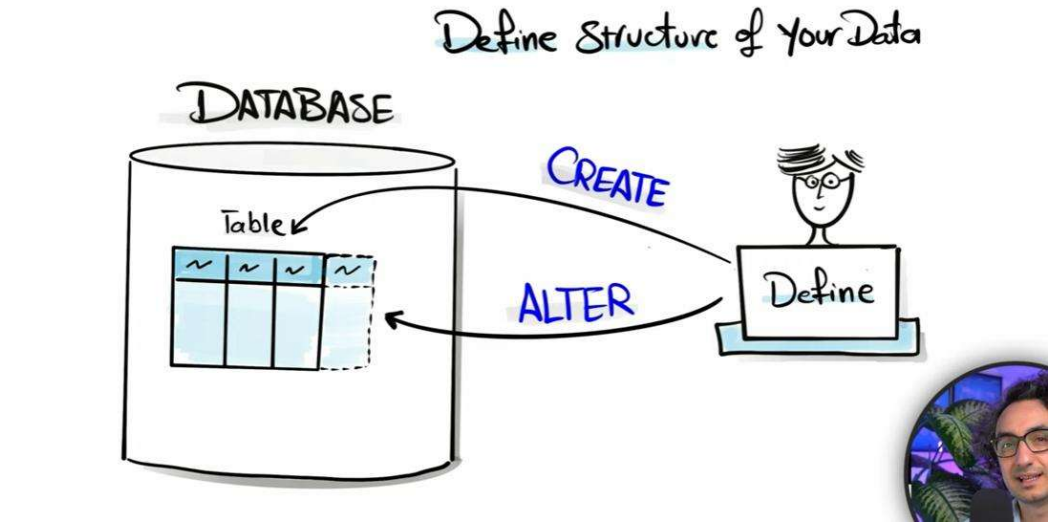
Messages

```
Commands completed successfully.

Completion time: 2025-02-16T14:08:55.4156439+01:00
```

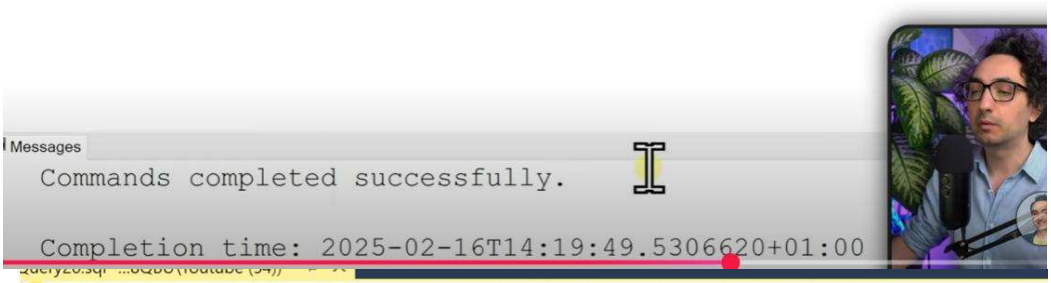
DDL commands never retern data it's changing the structure of your database

BY using direct click of creating table code.

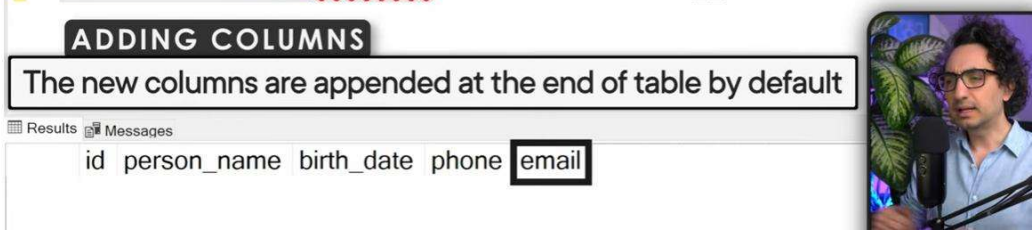
```sql
USE [MyDatabase]
GO

/****** Object:  Table [dbo].[persons]    Script Date: 16/02/2025 14:13:14 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[persons](
    [id] [int] NOT NULL,
    [person_name] [varchar](50) NOT NULL,
    [birth_date] [date] NULL,
    [phone] [varchar](15) NOT NULL,
 CONSTRAINT [pk_persons] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, A
) ON [PRIMARY]
GO
```




```sql
-- Add a new column called email to the persons table

ALTER TABLE persons
ADD email VARCHAR(50) NOT NULL
```




```sql
-- Add a new column called email to the persons table

ALTER TABLE persons
ADD email VARCHAR(50) NOT NULL

SELECT * FROM persons
```



To add column name in middle we have to drop table and create new one that is bad practice may loose our data already existing inside the table.

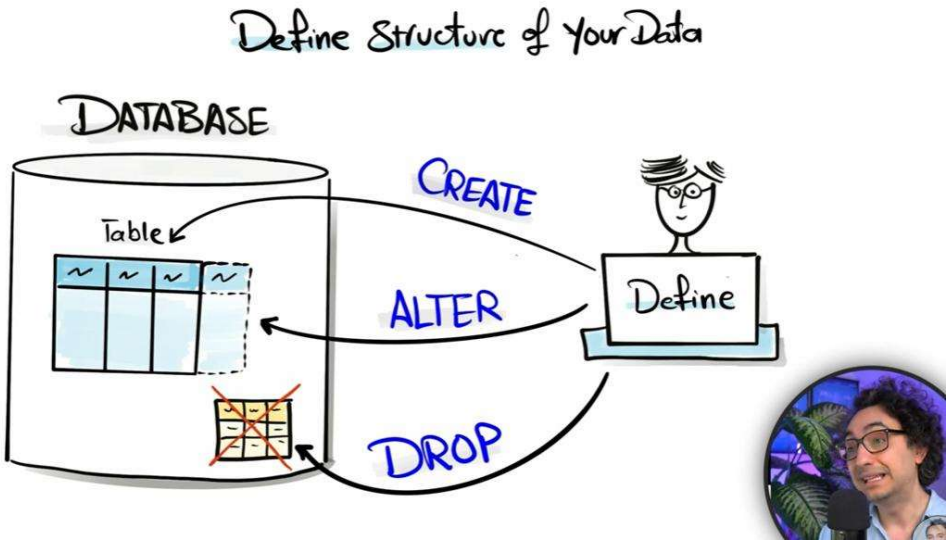
```
-- Remove the column phone from the persons table

ALTER TABLE persons
DROP COLUMN phone

SELECT * FROM persons
```

```
Messages
Commands completed successfully.

Completion time: 2025-02-16T14:27:01.4046064+01:00
```

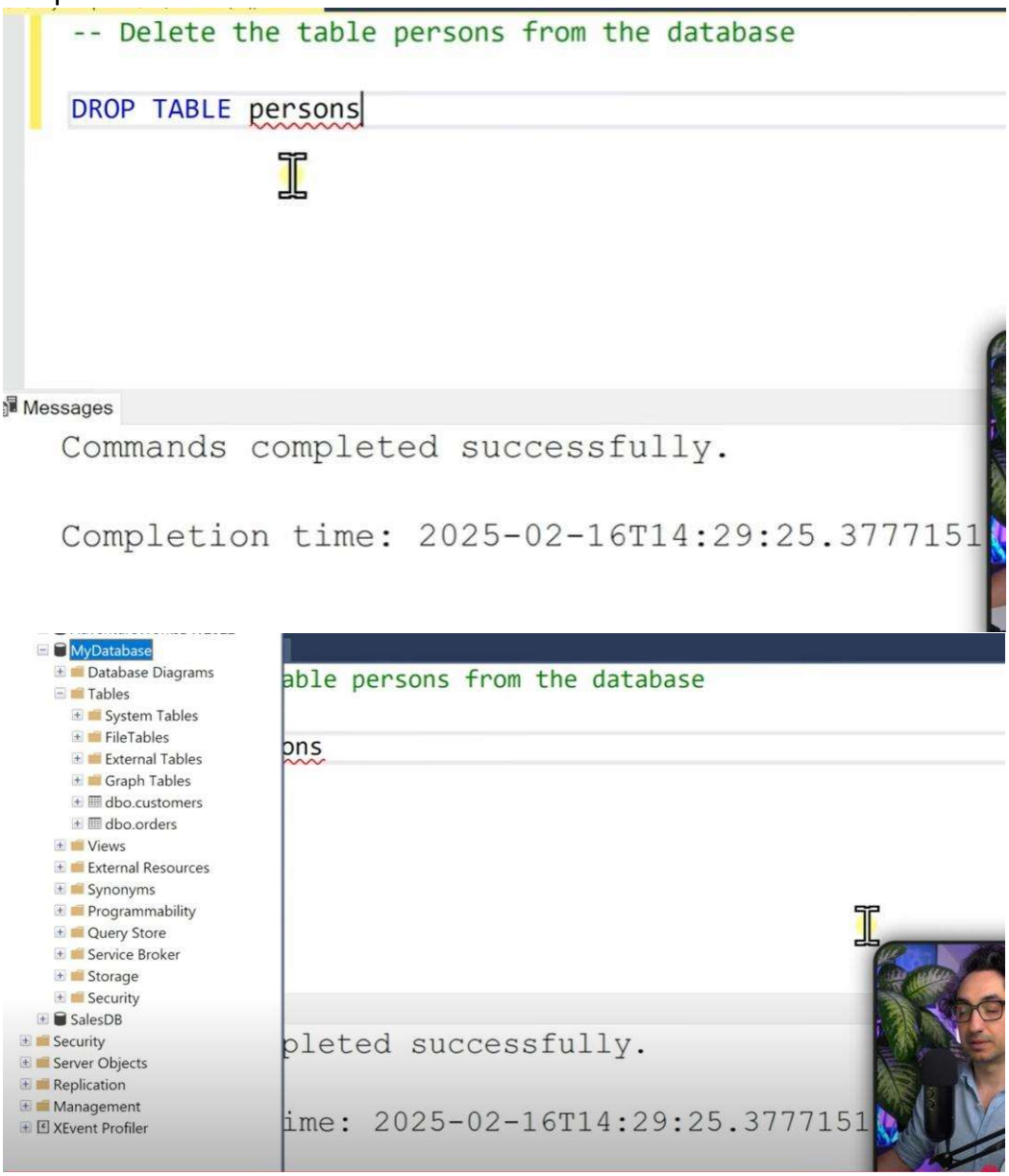
```
-- Remove the column phone from the persons table

ALTER TABLE persons
DROP COLUMN phone

SELECT * FROM persons
```

```
Results  Messages
id  person_name  birth_date  email
```

if we delete column we will lose its entire data

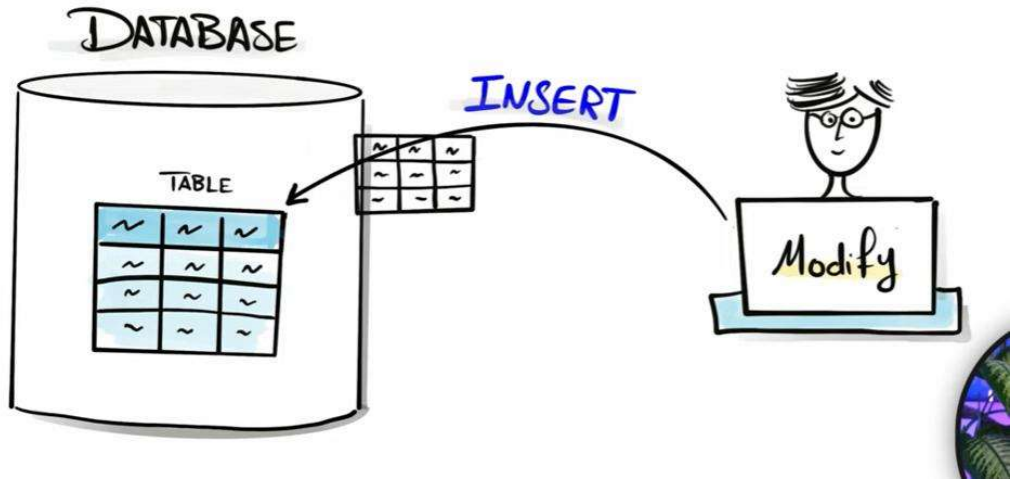


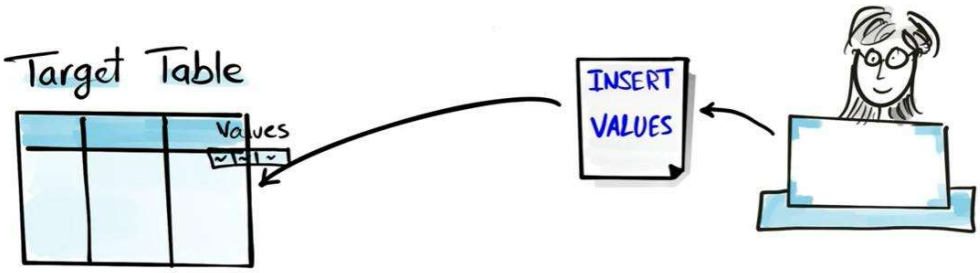
Drop remove data and table both from database

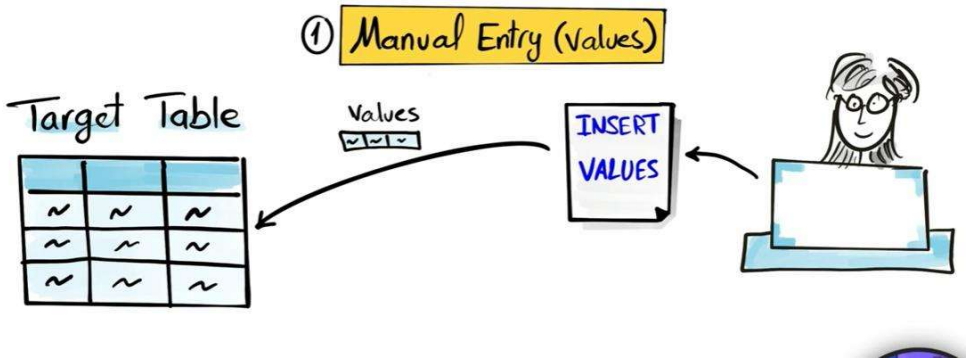


DML commands:

Modify (Manipulate) Your Data







INSERT Syntax

OPTIONAL: If no columns are specified, SQL expects values for all columns

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Rule

- Match the number of columns and Values.

INSERT Syntax

OPTIONAL: If no columns are specified, SQL expects values for all columns

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
      ,(value1, value2, value3,...)
```

Multiple Inserts

Rule

- Match the number of columns and Values.

```
INSERT INTO customers (id, first_name, country, score)
VALUES
    (6, 'Anna', 'USA', NULL),
    (NULL, 'Sam', NULL, 100)
```

```
Messages
Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'i
The statement has been terminated.

Completion time: 2025-02-16T14:51:03.912631
```

Primary key errror

```
INSERT INTO customers (id, first_name, country, score)
VALUES
    (6, 'Anna', 'USA', NULL),
    (7, 'Sam', NULL, 100)
```

```
Messages
(2 rows affected)

Completion time: 2025-02-16T14:51:24.339695
```

```
INSERT INTO customers (id, first_name, country, score)
VALUES
    (6, 'Anna', 'USA', NULL),
    (7, 'Sam', NULL, 100)
```

CAUTION

Columns and values must be in the same order

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
6	6	Anna	USA	NULL
7	7	Sam	NULL	100

Accepted although there is country name in place of name and name in place of country name this is because both are in varchar format

```sql
INSERT INTO customers (id, first_name, country, score)
VALUES
    (8, 'USA', 'Max', NULL)

SELECT * FROM customers
```

Messages

```
(1 row affected)

Completion time: 2025-02-16T14:54:39.507605
```

Due to data type rules and its constraints

```sql
INSERT INTO customers (id, first_name, country, score)
VALUES
    (8, 'USA', 'Max', NULL)

RULES FROM customers
```

Matching Data Types, Column Count & Constraints

	id	first_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
6	6	Anna	USA	NULL
7	7	Sam	NULL	100
8	8	USA	Max	NULL

```sql
INSERT INTO customers (id, first_name, country, score)
VALUES
    ('Max', 9, 'Max', NULL)

RULES FROM customers
```

Matching Data Types, Column Count & Constraints

```
Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varch

Completion time: 2025-02-16T14:55:38.832552
```