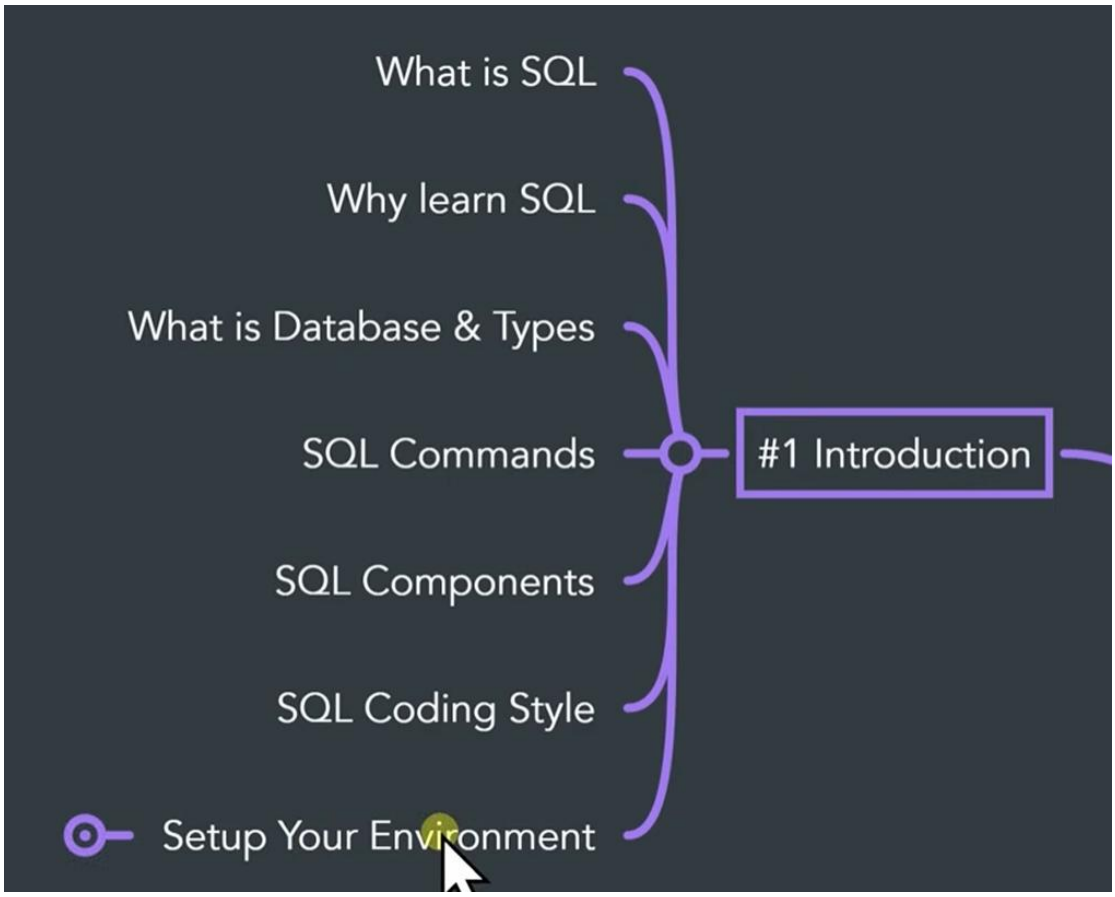
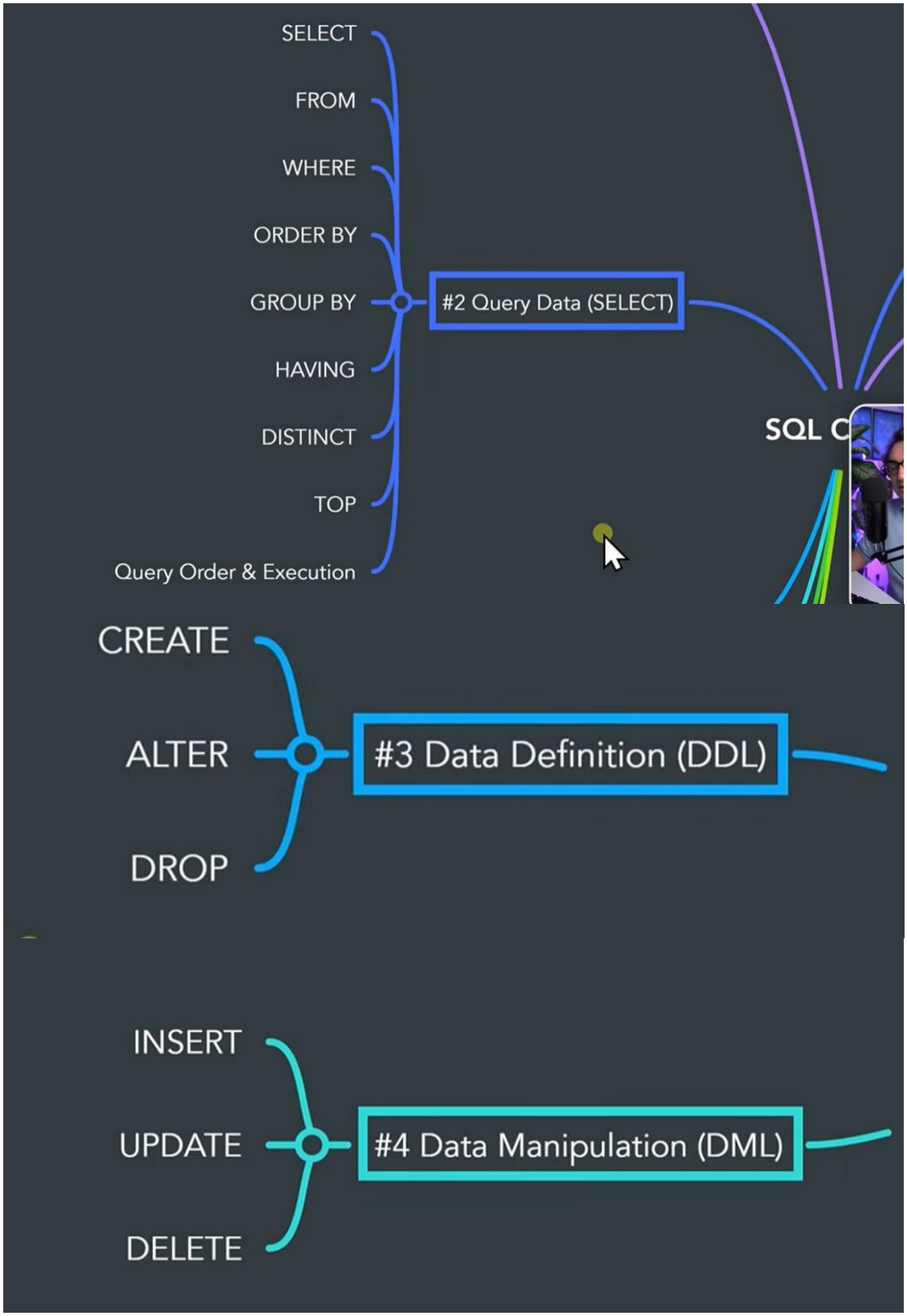
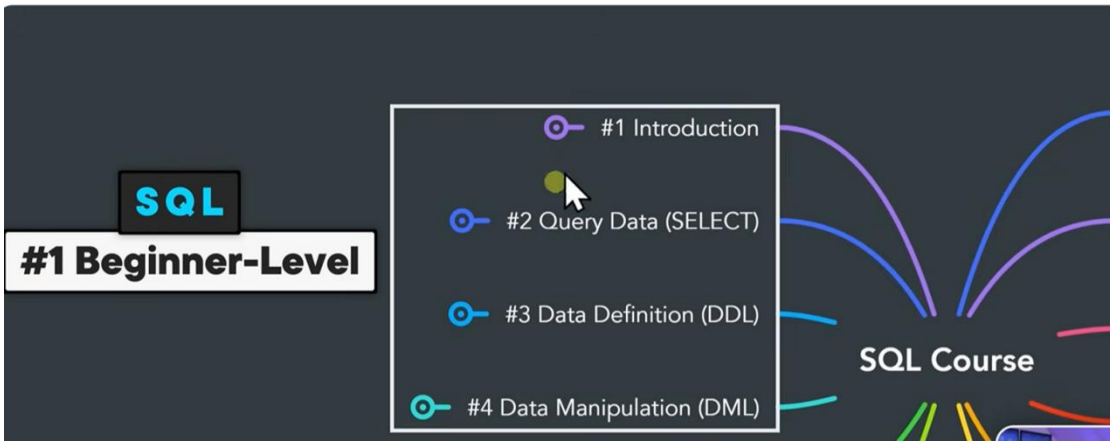
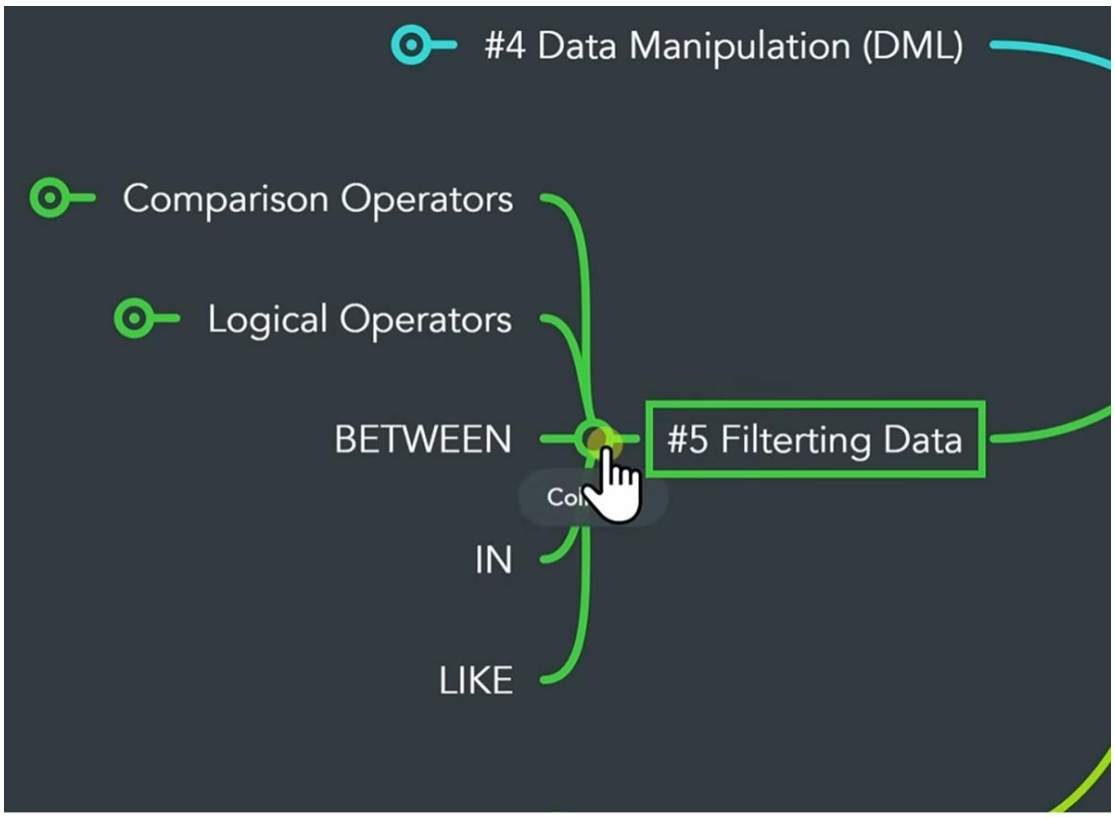
SQL Learning:

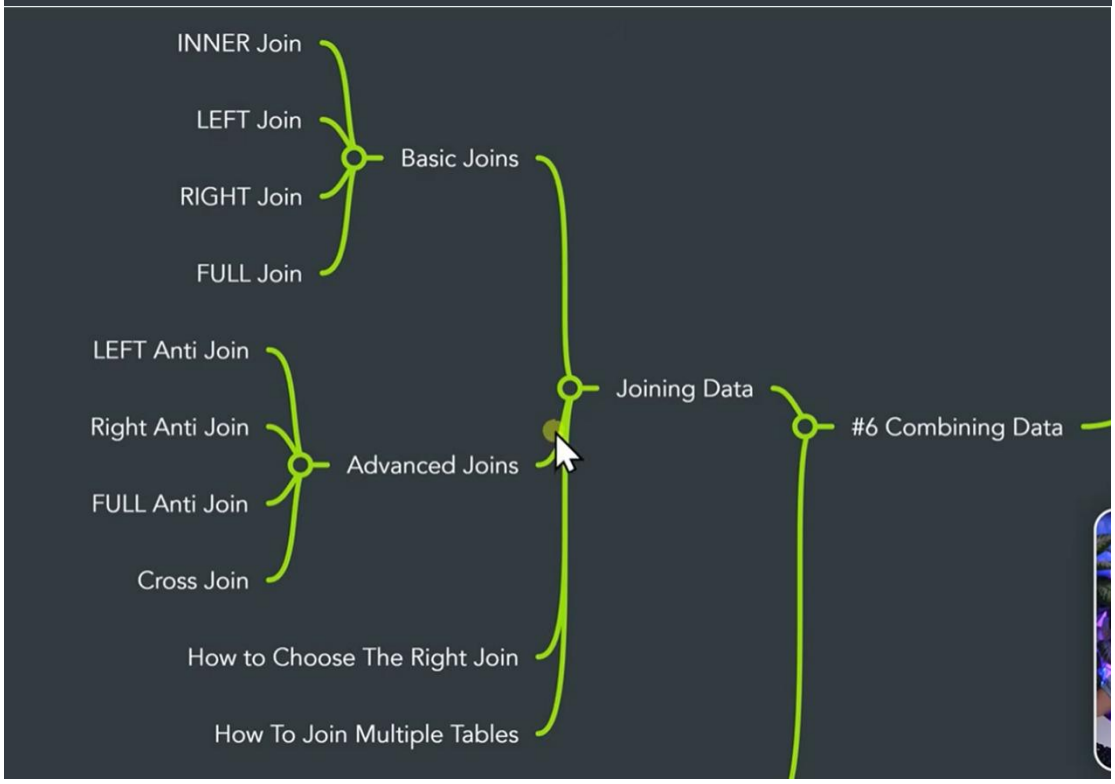


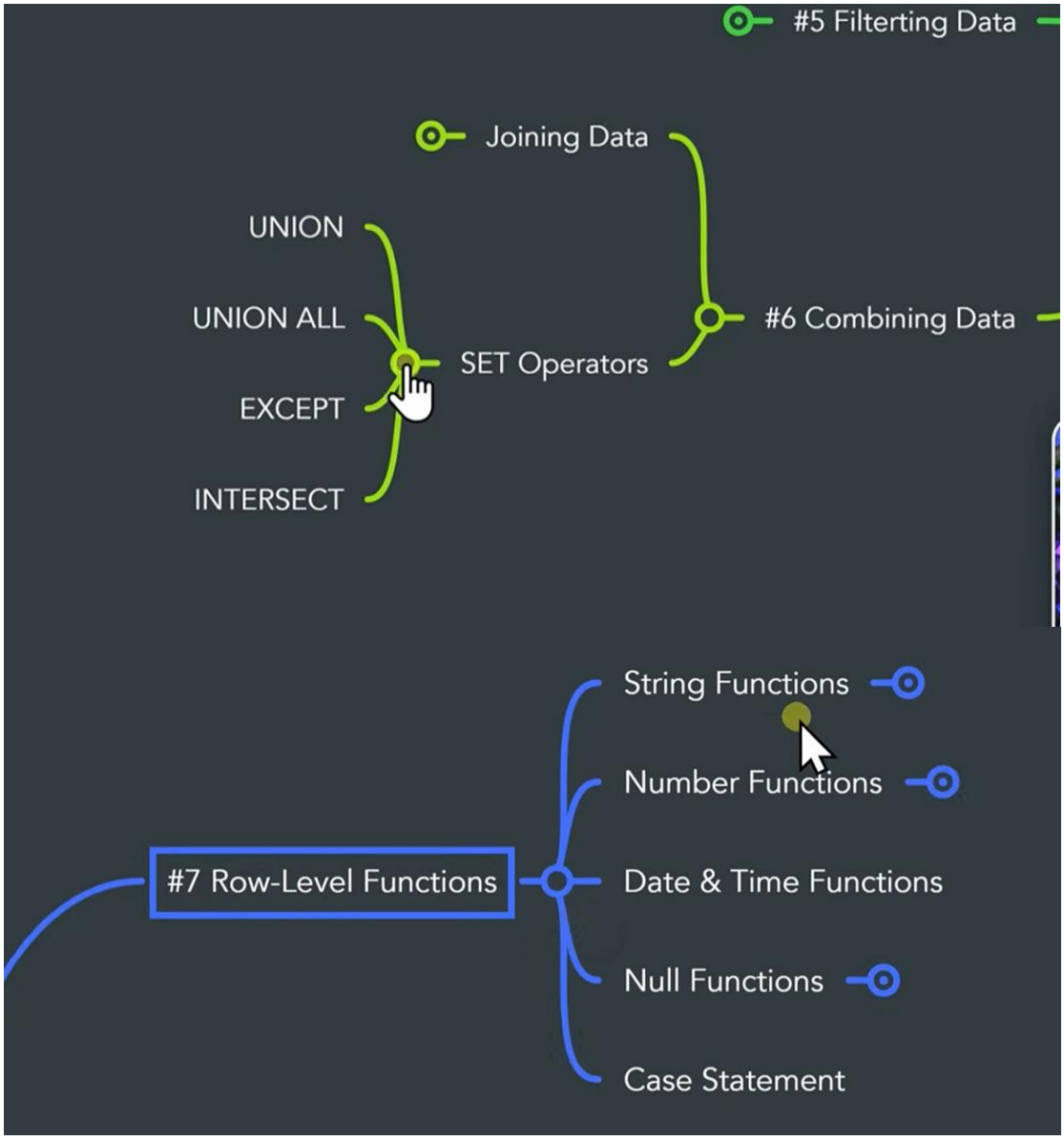

SELECT
FROM
WHERE
ORDER BY
GROUP BY
HAVING
DISTINCT
TOP
Query Order & Execution
#2 Query Data (SELECT)
SQL C
CREATE
ALTER
DROP
#3 Data Definition (DDL)
INSERT
UPDATE
DELETE
#4 Data Manipulation (DML)


SQL
#1 Beginner-Level
#1 Introduction
#2 Query Data (SELECT)
#3 Data Definition (DDL)
#4 Data Manipulation (DML)
SQL Course


#4 Data Manipulation (DML)
Comparison Operators
Logical Operators
BETWEEN
#5 Filterting Data
IN
LIKE

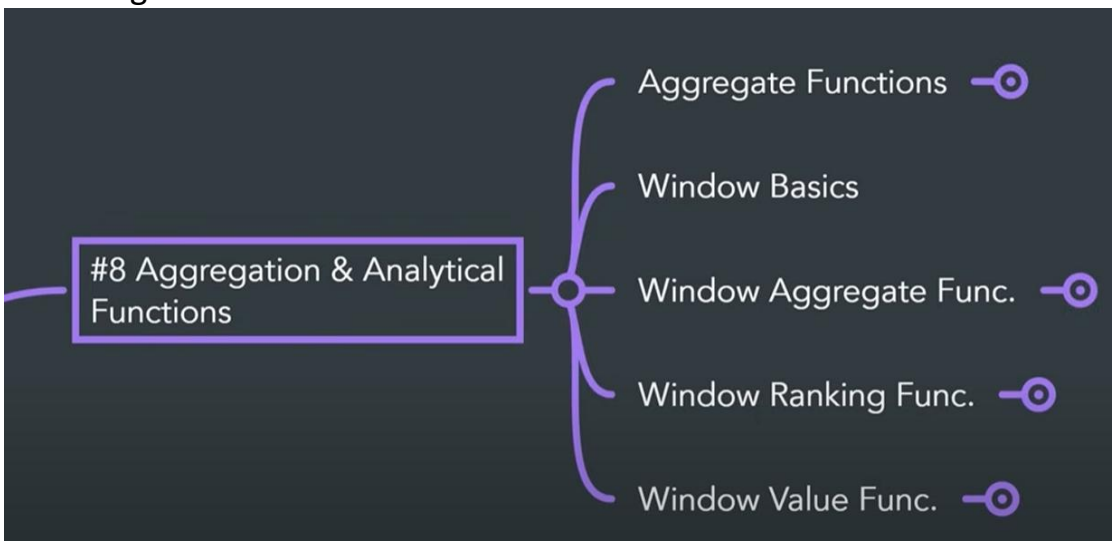

Joining Data
SET Operators
#6 Combining Data


INNER Join
LEFT Join
RIGHT Join
FULL Join
Basic Joins
LEFT Anti Join
Right Anti Join
FULL Anti Join
Cross Join
Advanced Joins
How to Choose The Right Join
How To Join Multiple Tables
Joining Data
#6 Combining Data

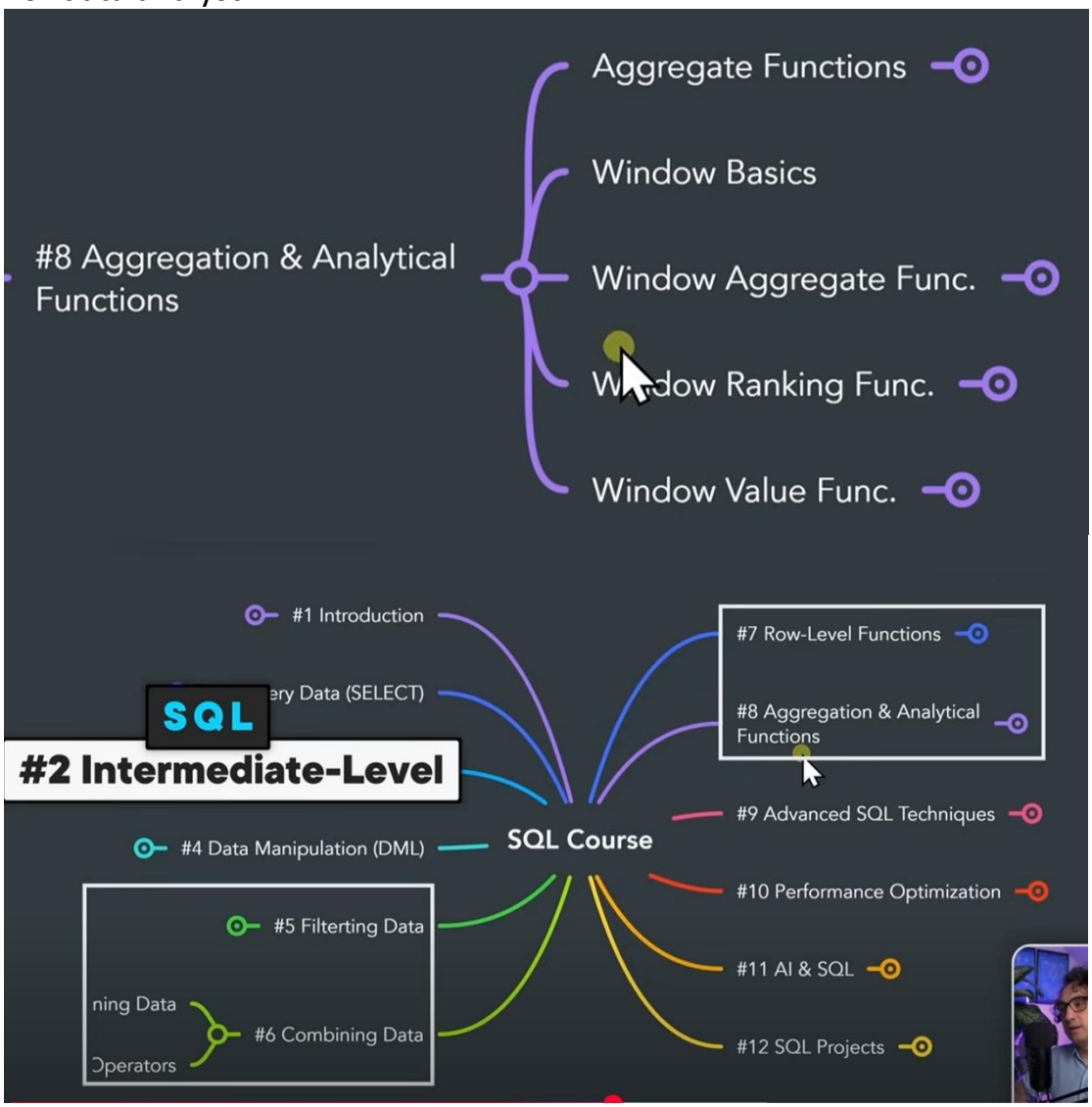


Row level transformation functions in sql transformation for single value

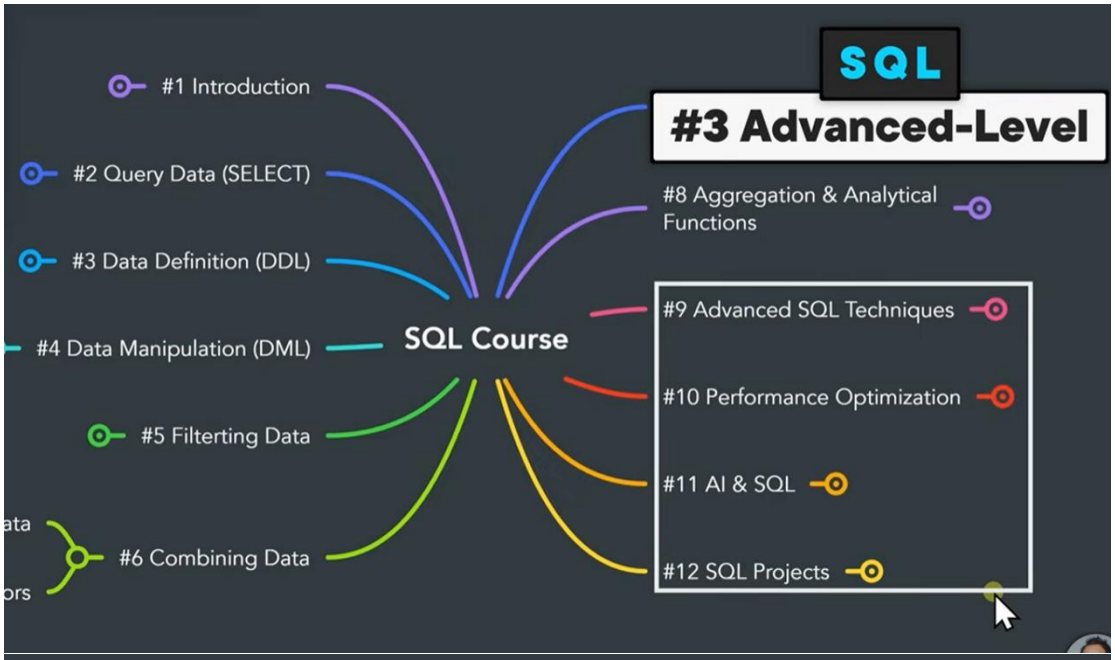
Data engineers

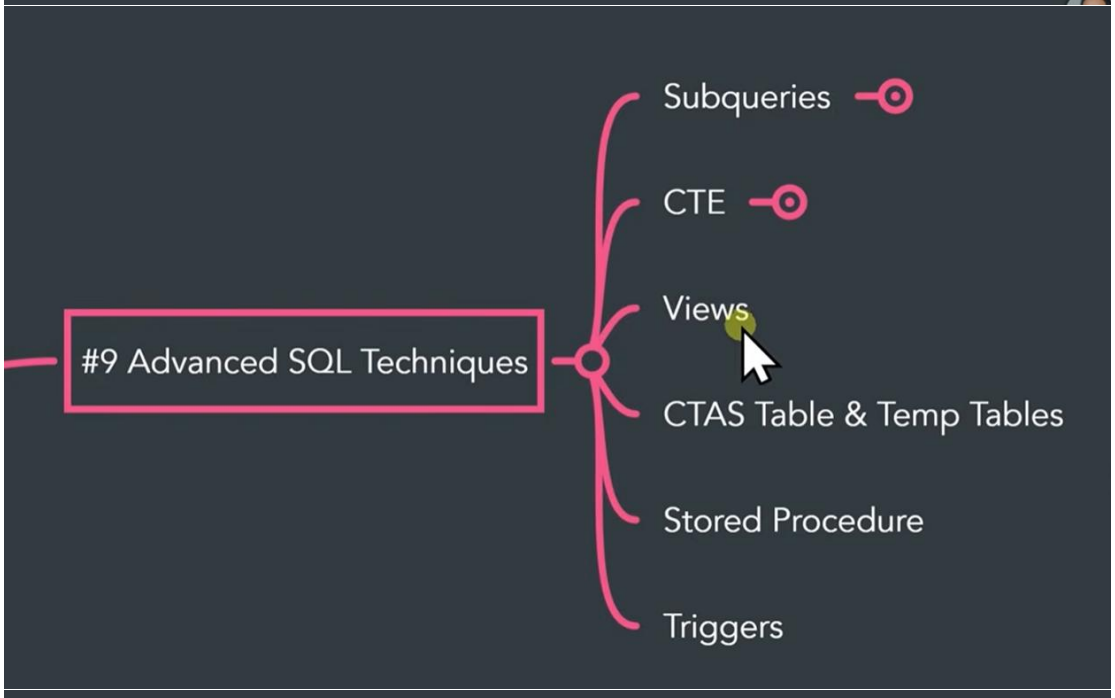


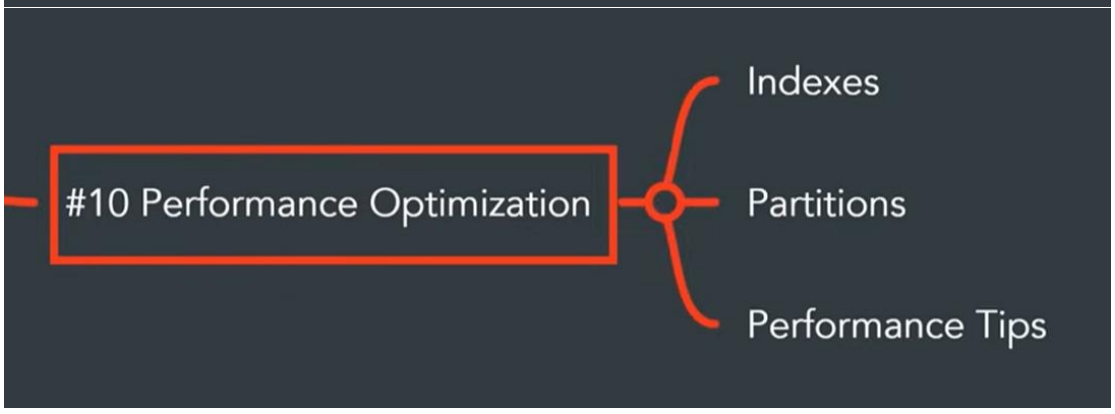
Data analyst and aggregation
For data analyst



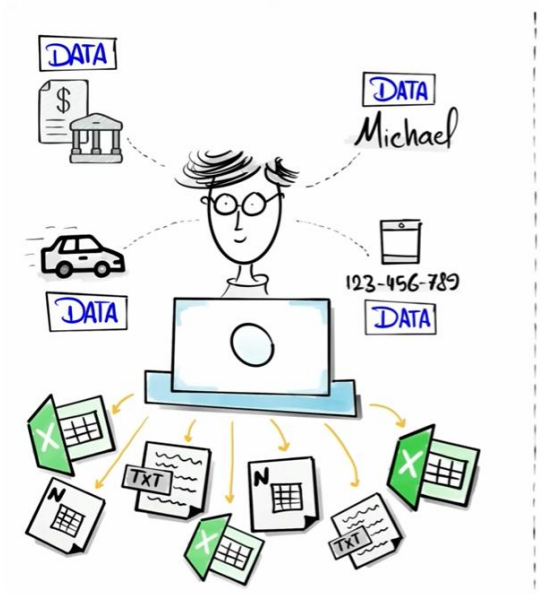
Advanced level-

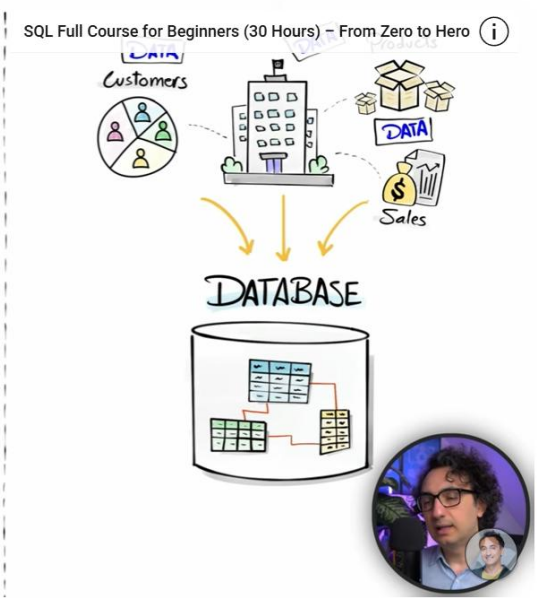

SQL
#3 Advanced-Level
#1 Introduction
#2 Query Data (SELECT)
#3 Data Definition (DDL)
#4 Data Manipulation (DML)
#5 Filterting Data
#6 Combining Data
SQL Course
#8 Aggregation & Analytical Functions
#9 Advanced SQL Techniques
#10 Performance Optimization
#11 AI & SQL
#12 SQL Projects


#9 Advanced SQL Techniques
Subqueries
CTE
Views
CTAS Table & Temp Tables
Stored Procedure
Triggers


#10 Performance Optimization
Indexes
Partitions
Performance Tips

What is sql -
Everything generating data and everything is data



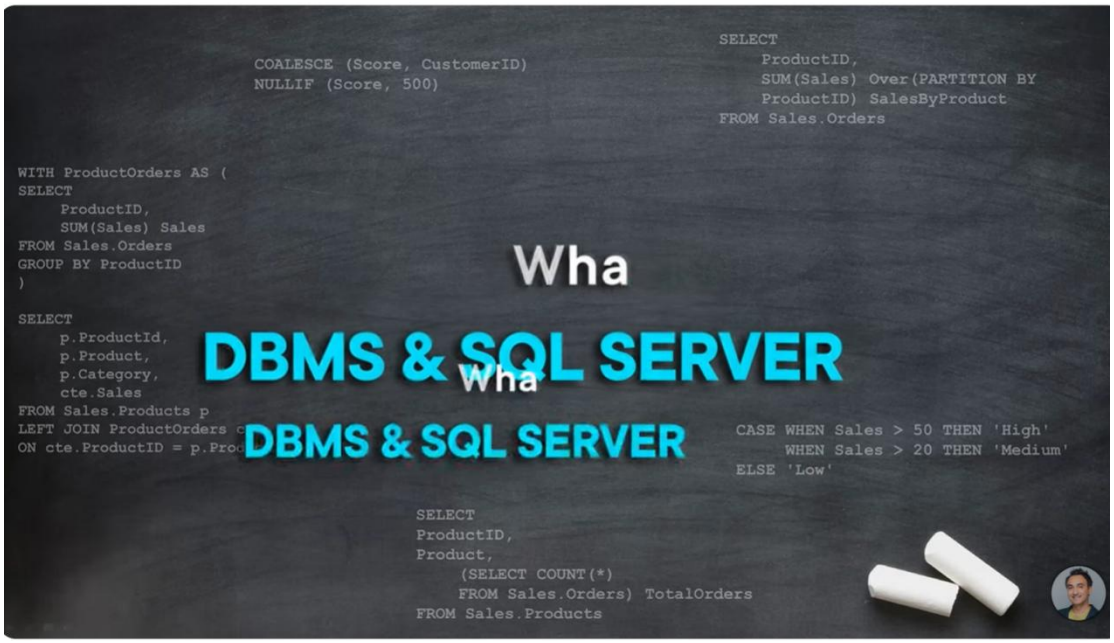


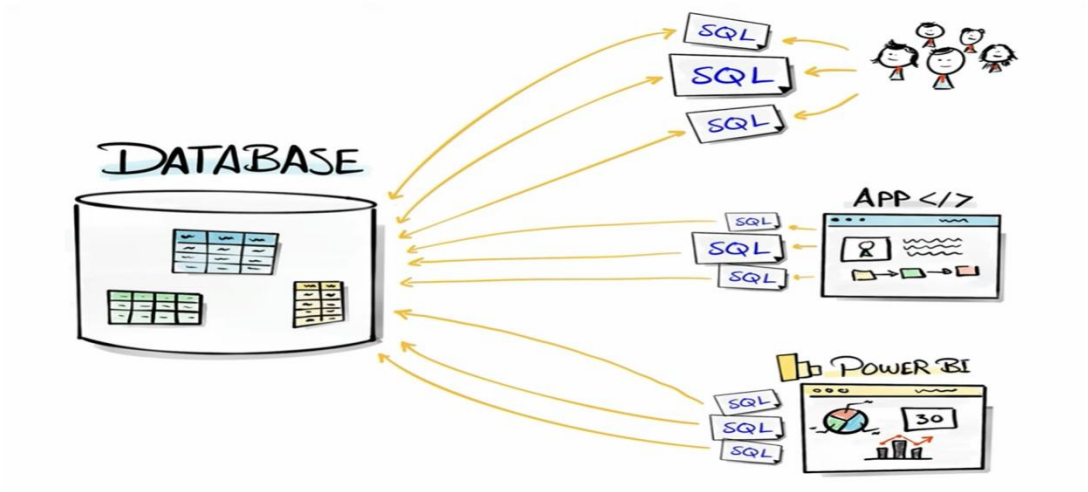
Container to store data - Database

Sql to read database or talk to database

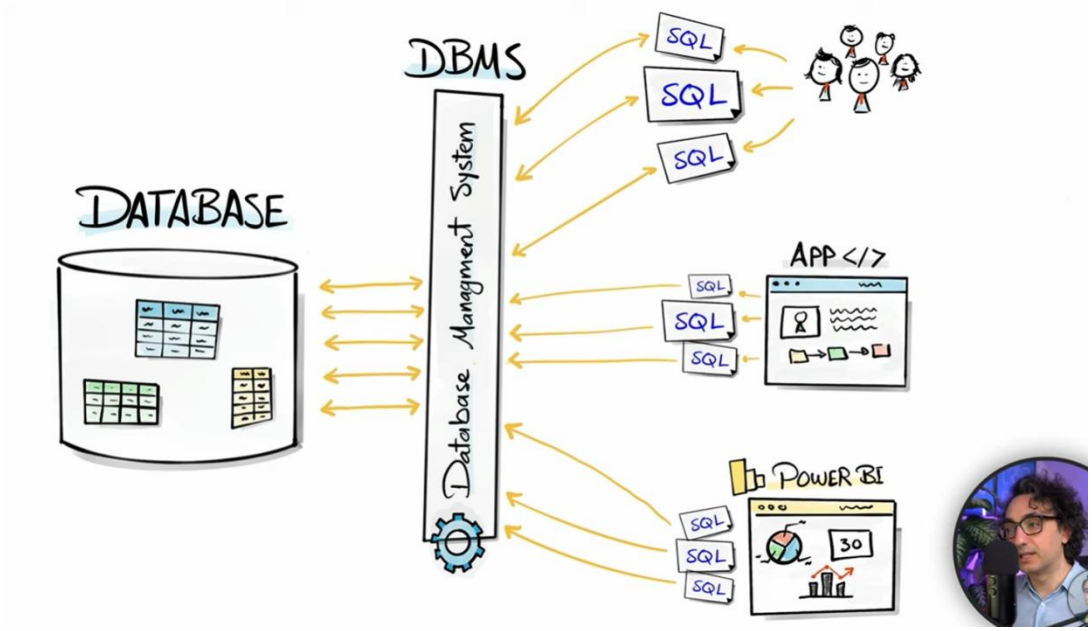
\

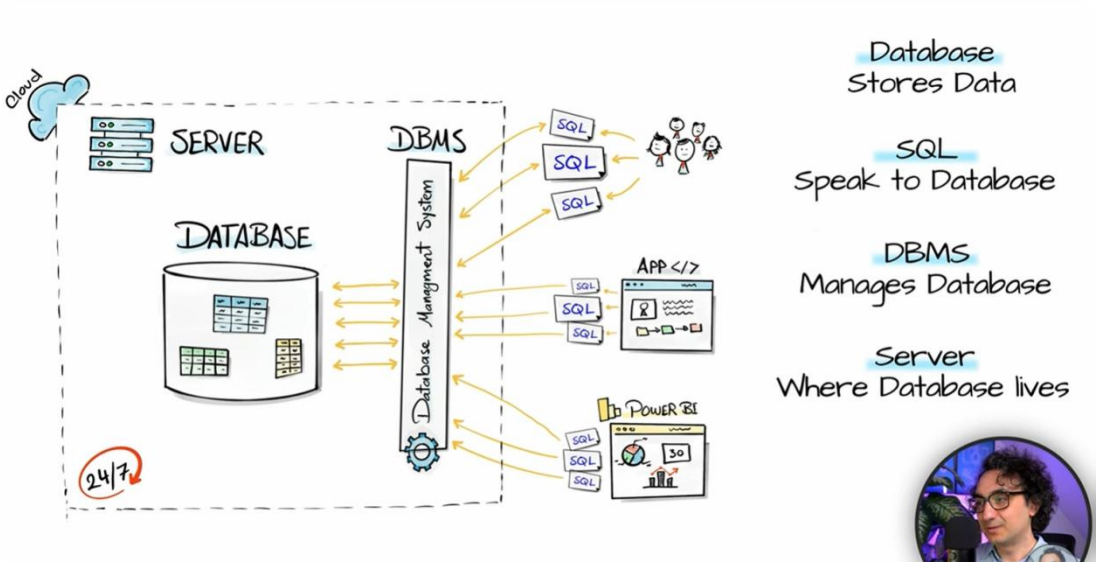
Databse - secure, and access controable.



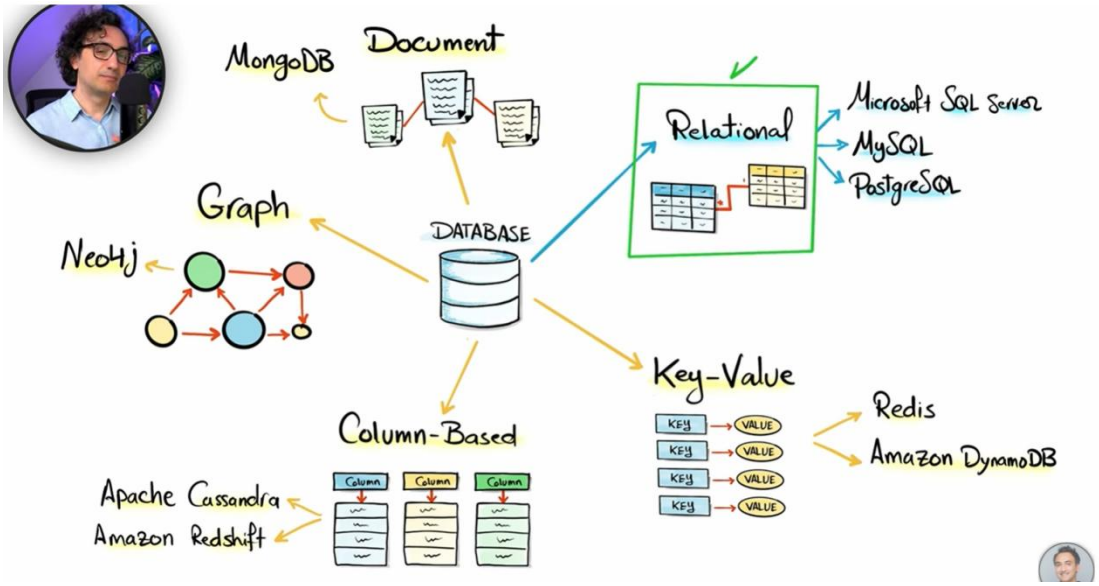


Software to manage data - DBMS

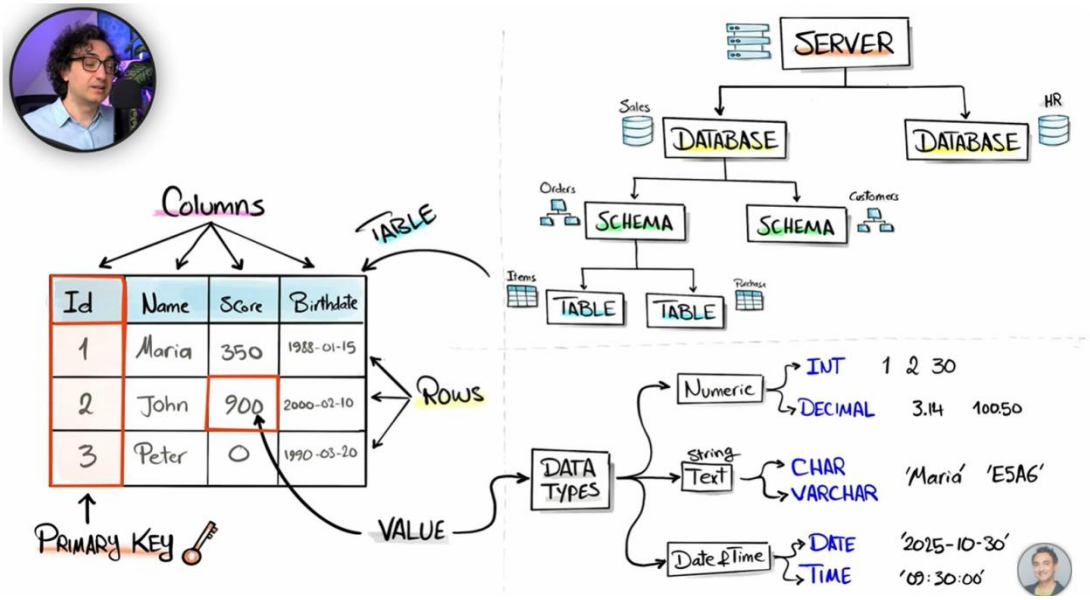


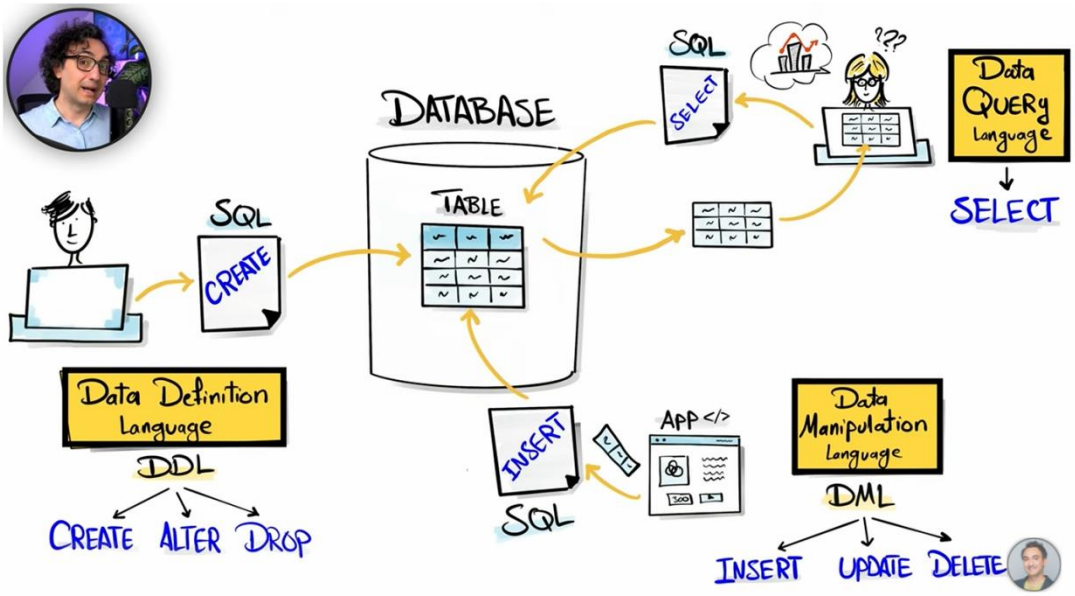


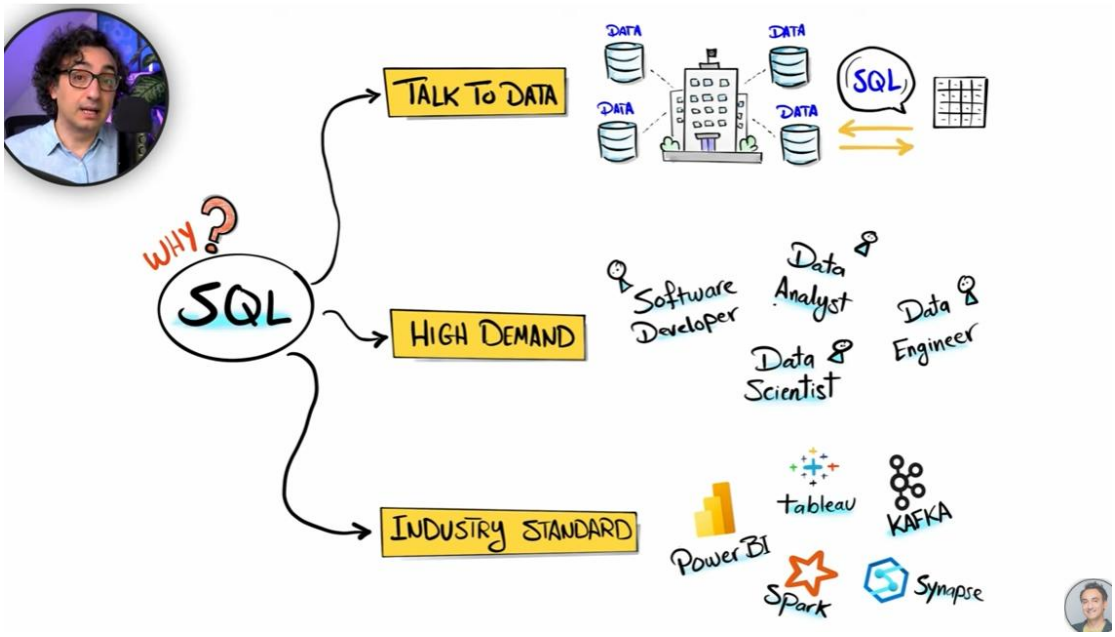
DatabaseTypes:



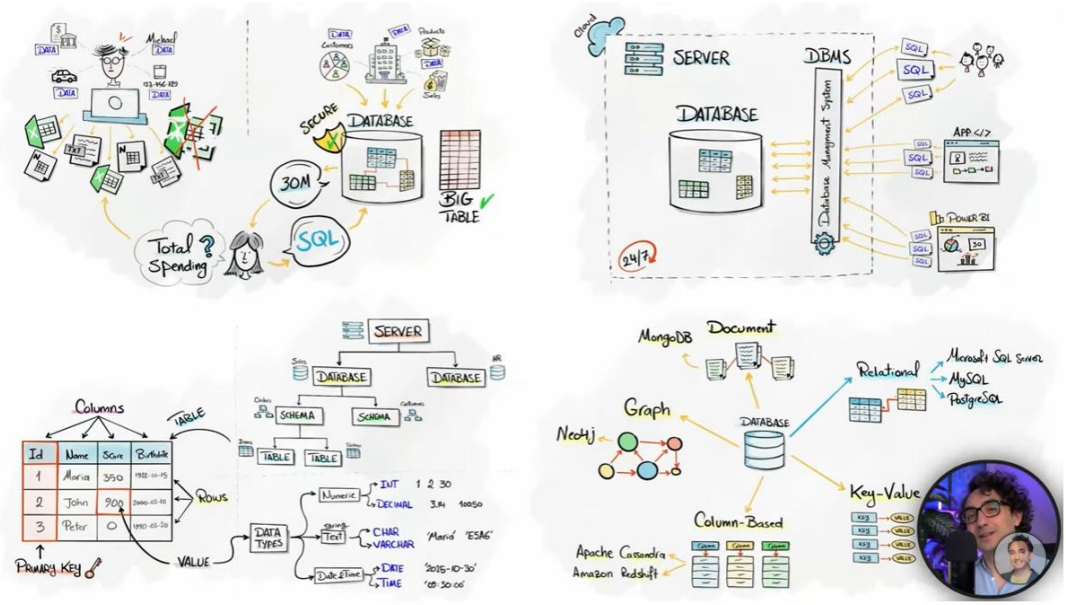
Database structures

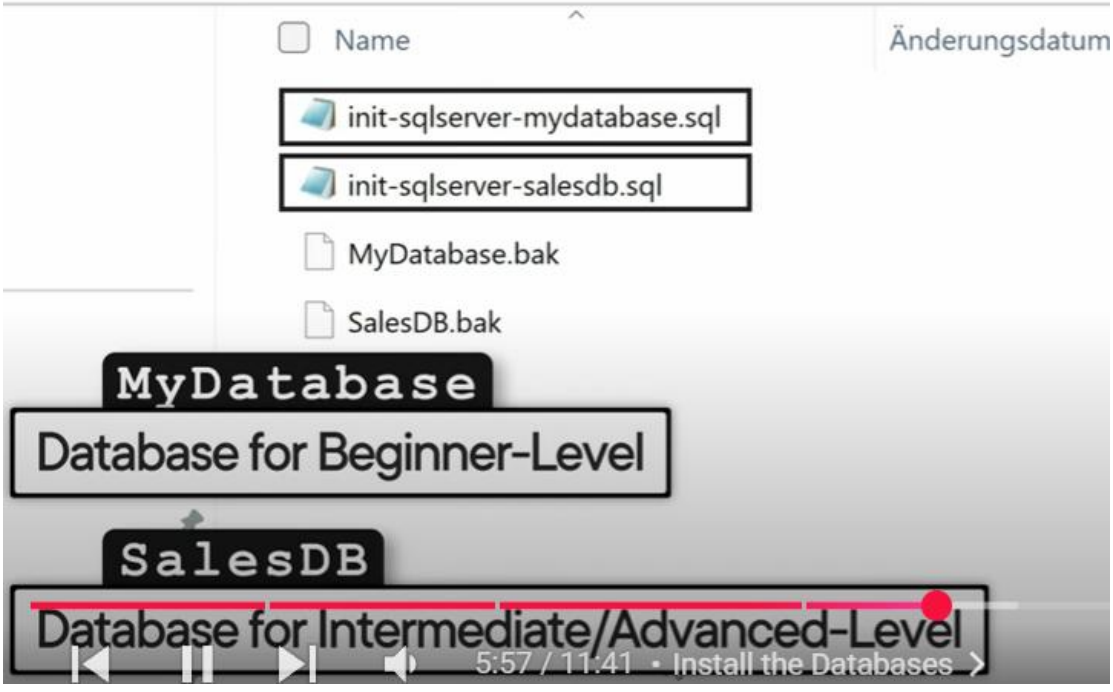

SERVER
Sales
DATABASE
HR
DATABASE
Orders
SCHEMA
SCHEMA
Customers
Items
TABLE
TABLE
Purchase
Columns
TABLE
Id
Name
Score
Birthdate
1
Maria
350
1988-01-15
2
John
900
2000-02-10
3
Peter
0
1990-03-20
ROWS
PRIMARY KEY
VALUE
DATA TYPES
Numeric
INT
1 2 30
DECIMAL
3.14
10.50
String
Text
CHAR
VARCHAR
'Maria'
'E5A6'
Date & Time
DATE
'2025-10-30'
TIME
'09:30:00'

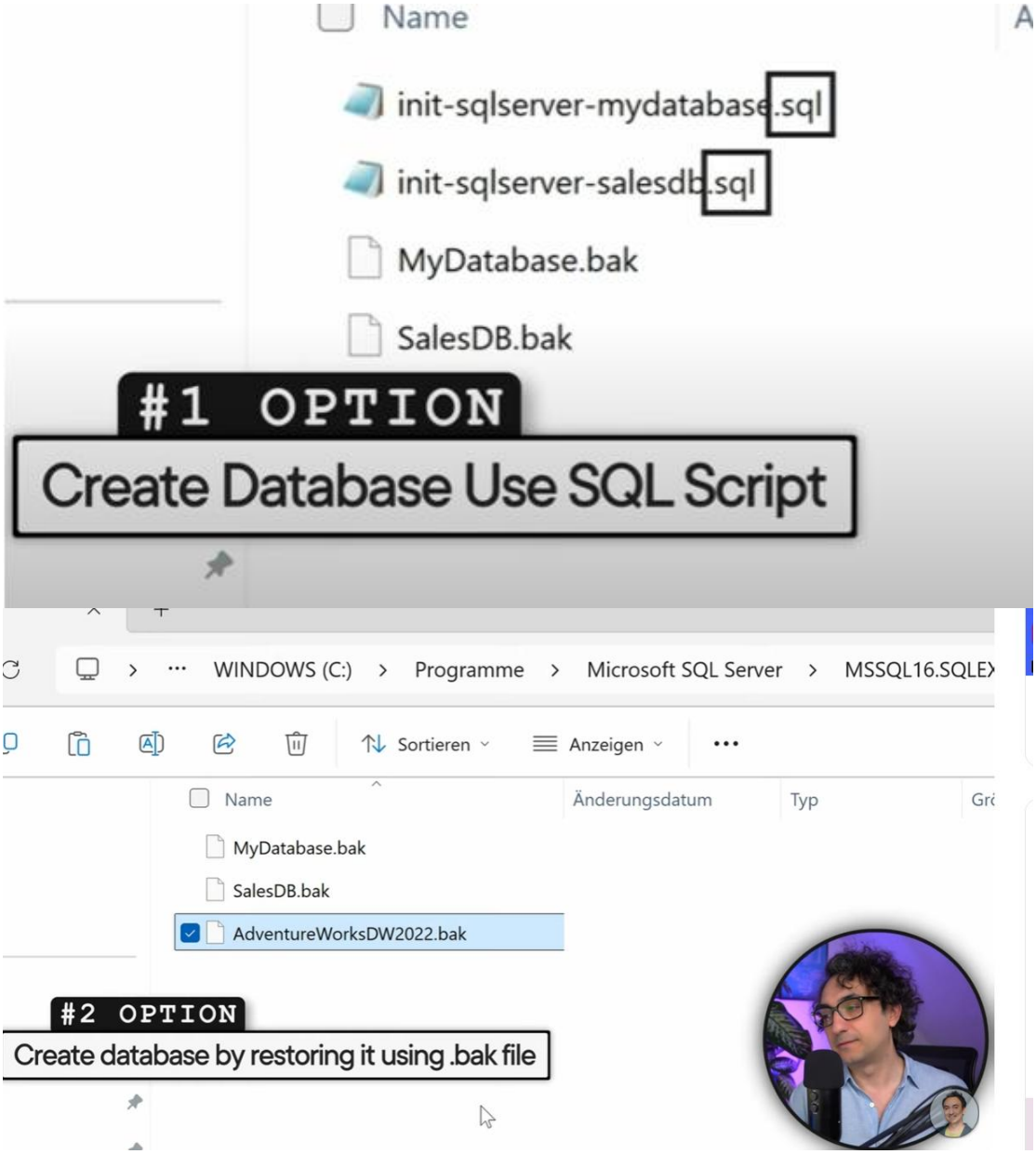

DATABASE
TABLE
SQL
SELECT
???
Data Query Language
SELECT
SQL
CREATE
Data Definition Language
DDL
CREATE
ALTER
DROP
INSERT
SQL
APP </>
Data Manipulation Language
DML
INSERT
UPDATE
DELETE

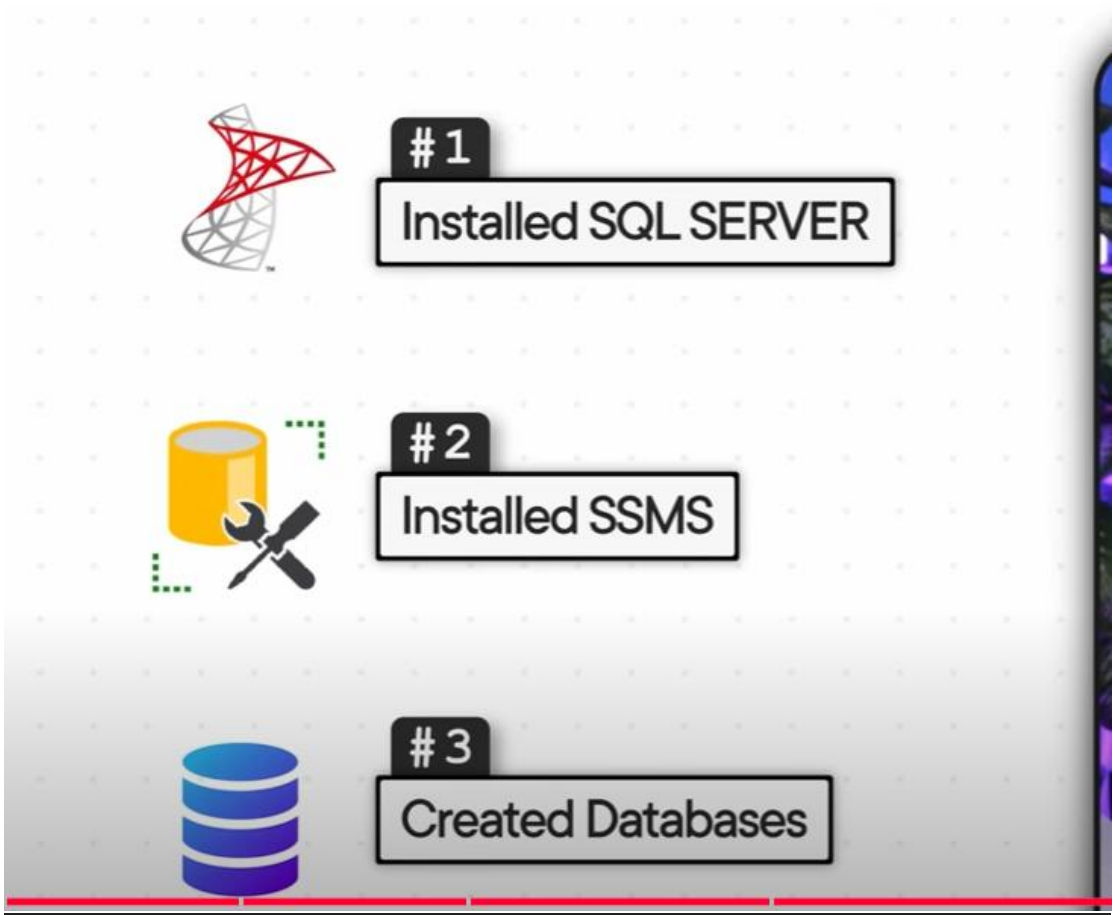


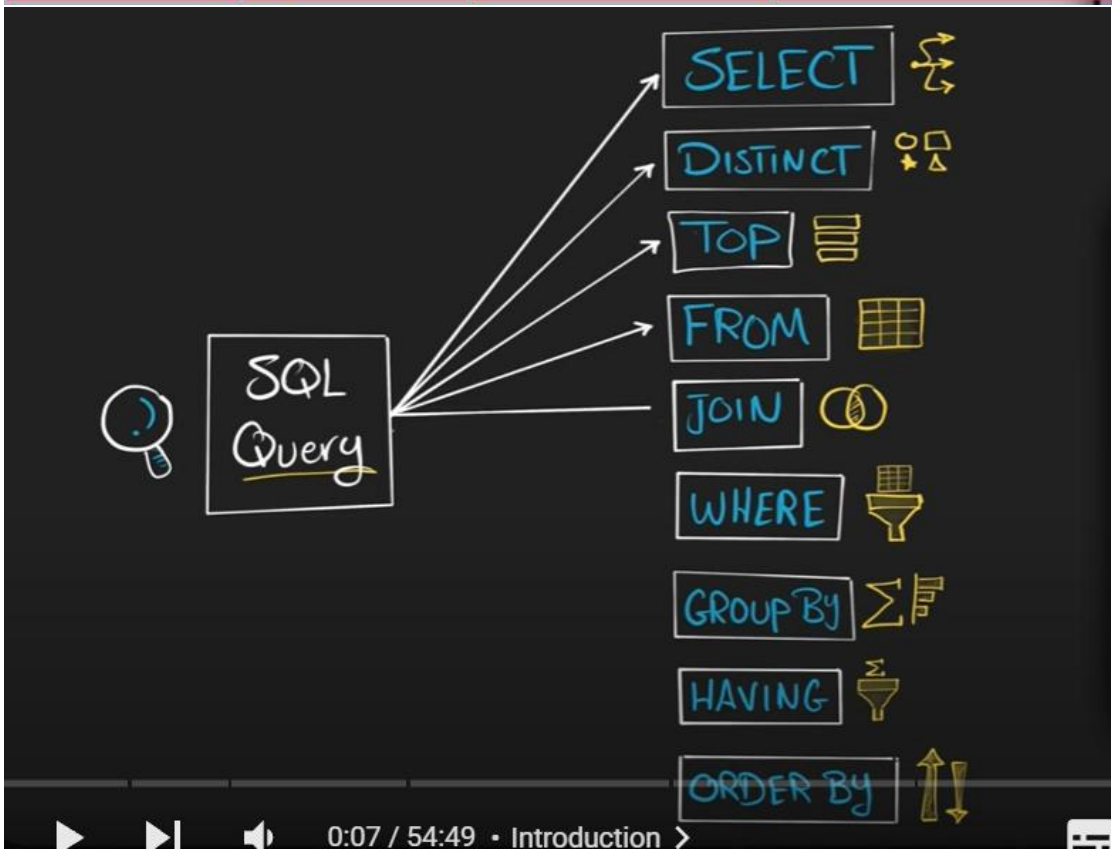
Overall lecture 2:

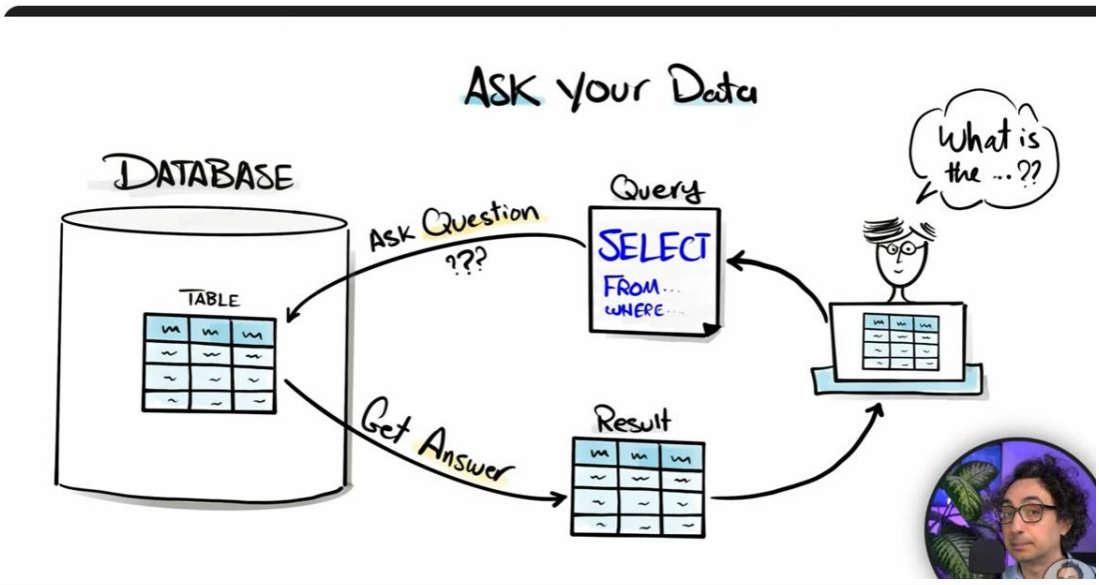

DATABASE
SECURE
30M
BIG TABLE
SQL
Total Spending
SERVER
DBMS
DATABASE
Database Management System
SQL
APP
POWER BI
24/7
Columns
TABLE
Id
Name
Score
Birthdate
ROWS
PRIMARY KEY
VALUE
SERVER
DATABASE
SCHEMA
TABLE
DATA TYPES
Numeric
INT
DECIMAL
Text
CHAR
VARCHAR
Date&Time
DATE
TIME
MongoDB
Document
Relational
Microsoft SQL Server
MySQL
PostgreSQL
Graph
Neo4j
DATABASE
Key-Value
Column-Based
Apache Cassandra
Amazon Redshift


Name
Änderungsdatum
init-sqlserver-mydatabase.sql
init-sqlserver-salesdb.sql
MyDatabase.bak
SalesDB.bak
MyDatabase
Database for Beginner-Level
SalesDB
Database for Intermediate/Advanced-Level
5:57 / 11:41 • Install the Databases


Name
init-sqlserver-mydatabase.sql
init-sqlserver-salesdb.sql
MyDatabase.bak
SalesDB.bak
#1 OPTION
Create Database Use SQL Script
WINDOWS (C:)
Programme
Microsoft SQL Server
Sortieren
Anzeigen
Name
Änderungsdatum
Typ
MyDatabase.bak
SalesDB.bak
AdventureWorksDW2022.bak
#2 OPTION
Create database by restoring it using .bak file


#1
Installed SQL SERVER
#2
Installed SSMS
#3
Created Databases


SQL Query
SELECT
DISTINCT
TOP
FROM
JOIN
WHERE
GROUP BY
HAVING
ORDER BY
0:07 / 54:49 • Introduction



Database

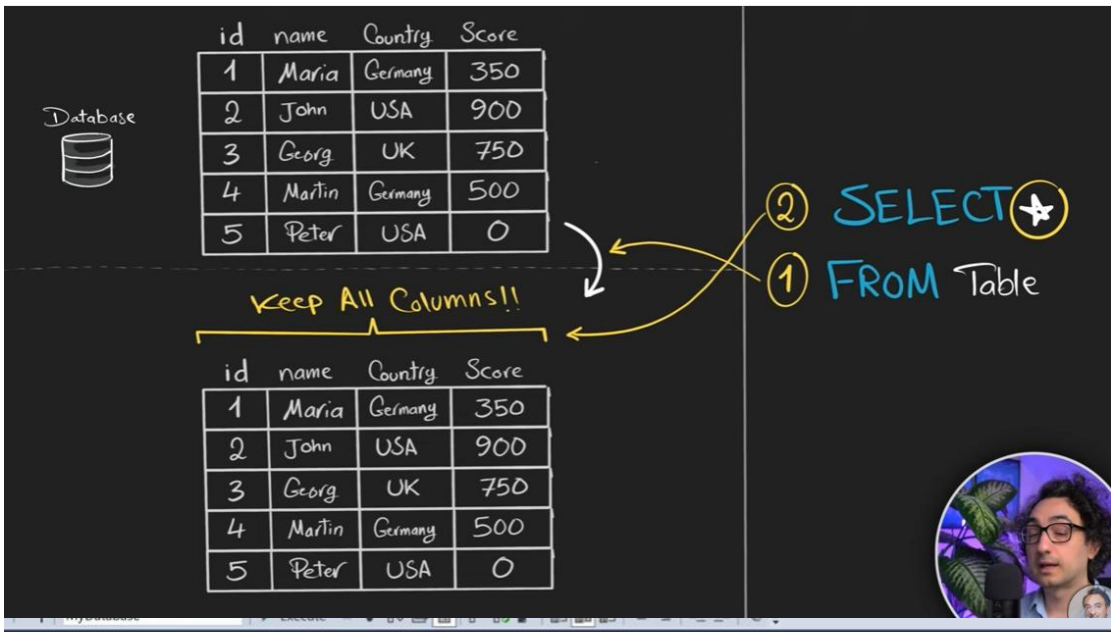
id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

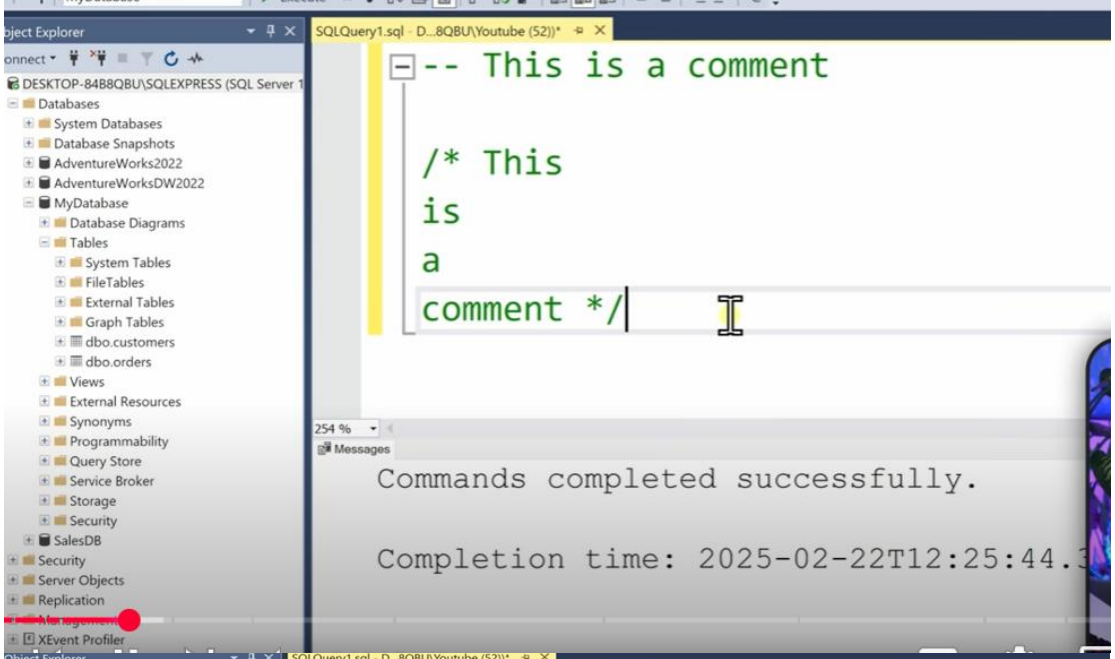
Select * (All)

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
Keep All Columns!!
2 SELECT *
1 FROM Table
id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0


Object Explorer
DESKTOP-84B8QBU\SQLEXPRESS (SQL Server 1
Databases
System Databases
Database Snapshots
AdventureWorks2022
AdventureWorksDW2022
MyDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.customers
dbo.orders
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
SalesDB
Security
Server Objects
Replication
XEvent Profiler
SQLQuery1.sql - D...8QBU\Youtube (52))*
-- This is a comment
/* This
is
a
comment */
254 %
Messages
Commands completed successfully.
Completion time: 2025-02-22T12:25:44.


Object Explorer
Connect
DESKTOP-84B8QBU\SQLEXPRESS (SQL Server 1
Databases
System Databases
Database Snapshots
AdventureWorks2022
AdventureWorksDW2022
MyDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.customers
dbo.orders
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
SalesDB
Security
Server Objects
Replication
Management
XEvent Profiler
SQLQuery1.sql - D...8QBU\Youtube (52))*
-- Retrieve All Customer Data
SELECT *
FROM customers
Results
Messages
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

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 Few Columns

Pick only the Columns You Need

instead of All

Database

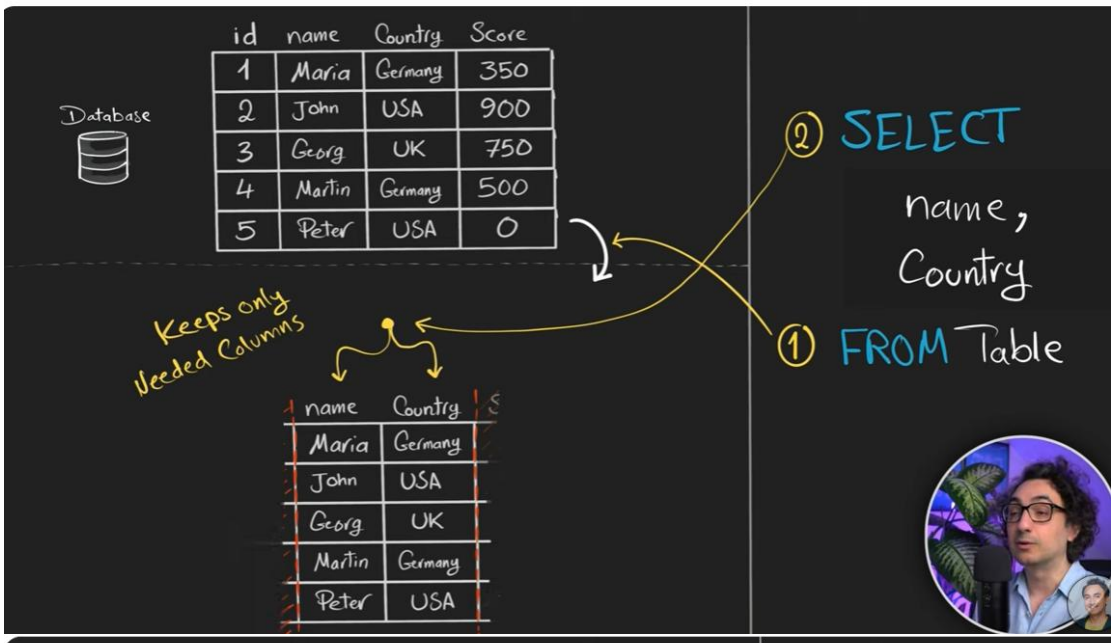
id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

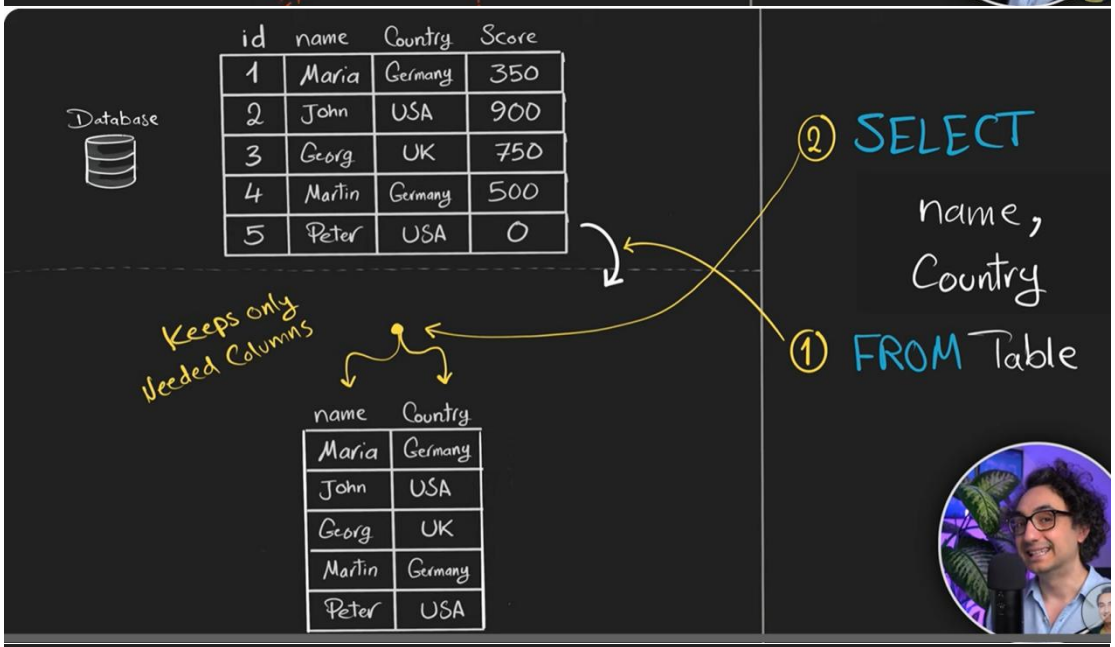
② SELECT
name,
Country

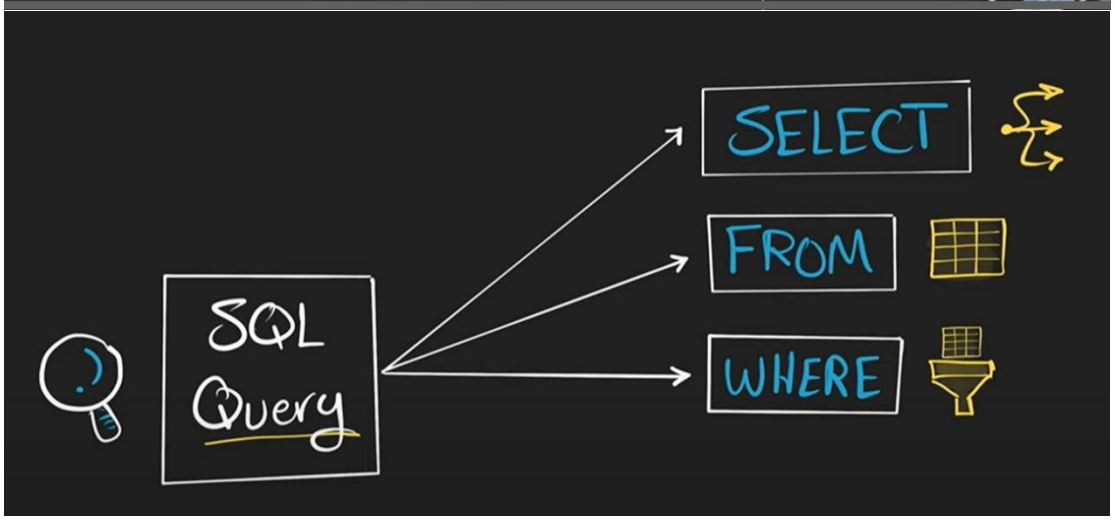
① FROM Table

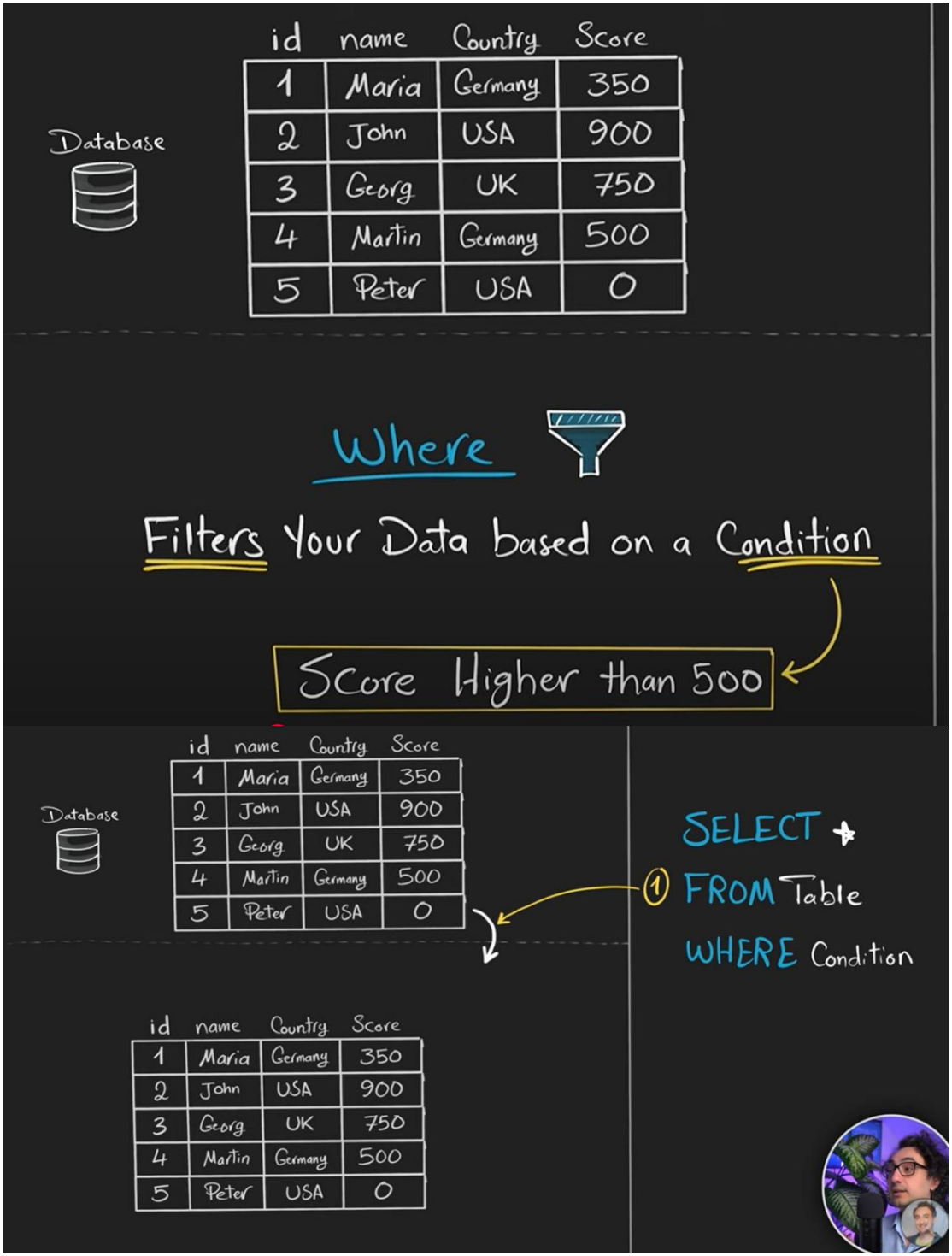
Keeps only
We

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

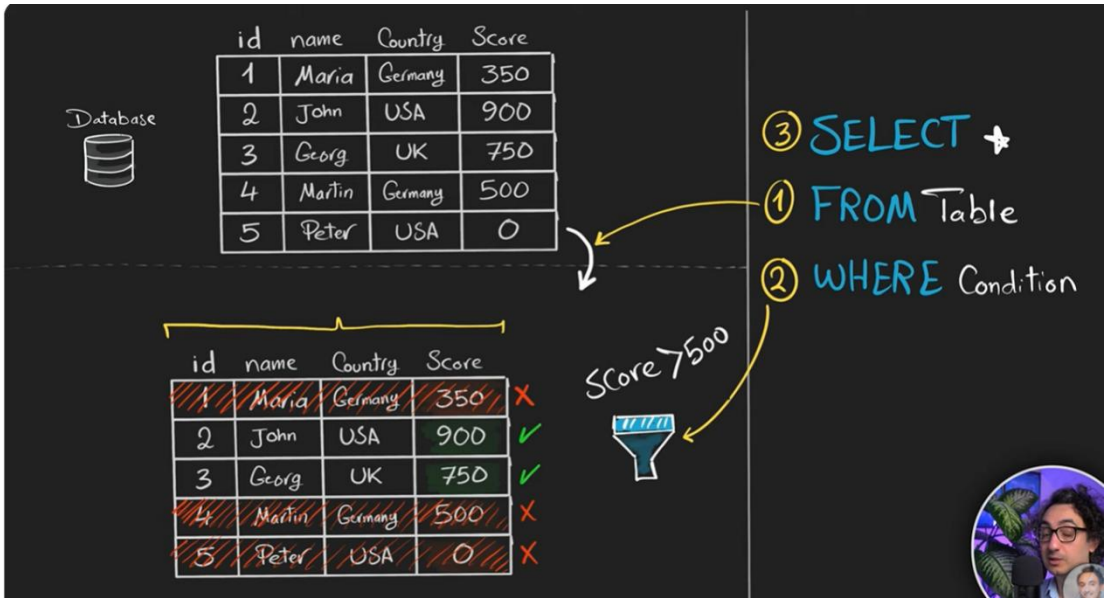

Database
id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
② SELECT
name,
Country
① FROM Table
Keeps only Needed Columns
name Country
Maria Germany
John USA
Georg UK
Martin Germany
Peter USA

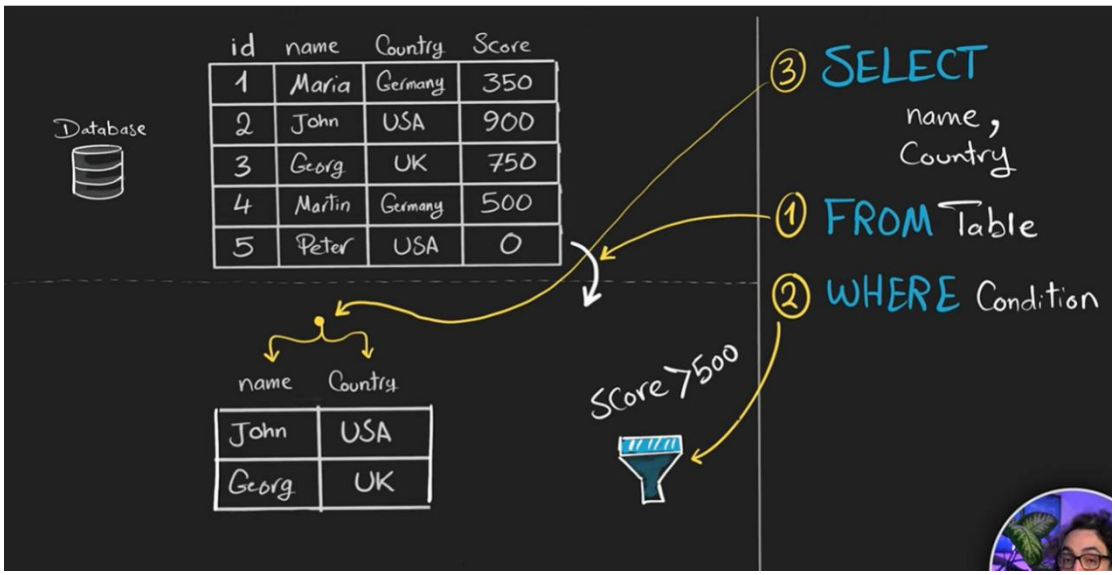

Database
id name Country Score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500
5 Peter USA 0
② SELECT
name,
Country
① FROM Table
Keeps only Needed Columns
name Country
Maria Germany
John USA
Georg UK
Martin Germany
Peter USA

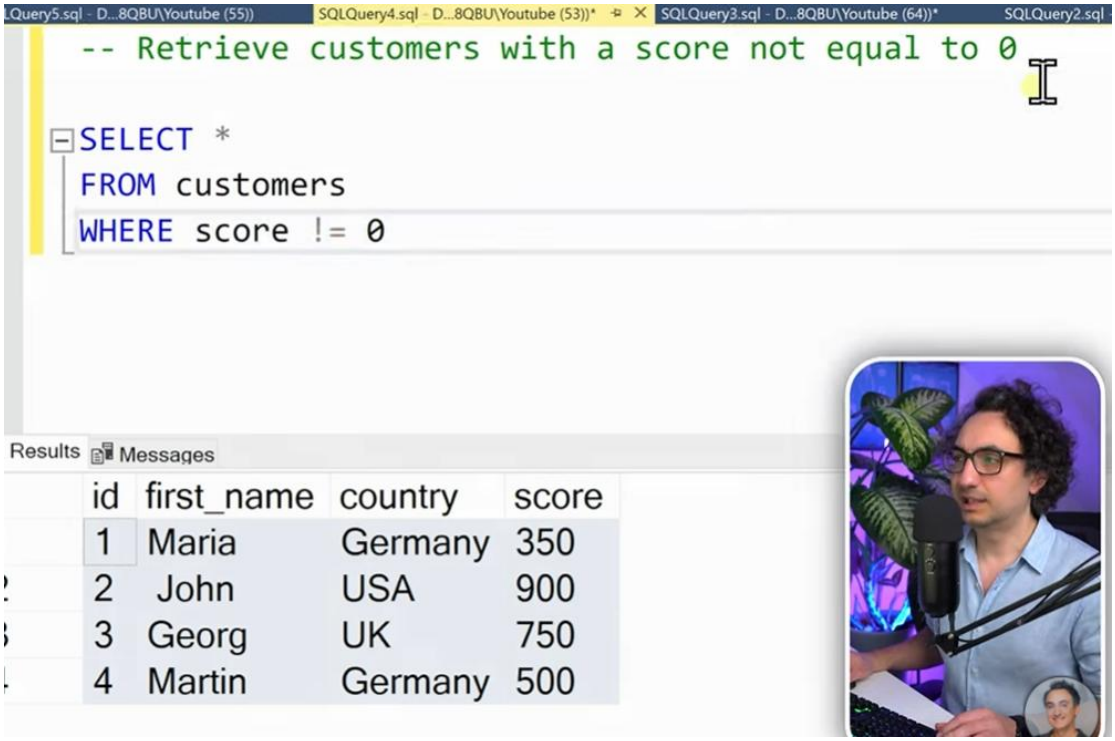

SQL Query
SELECT
FROM
WHERE

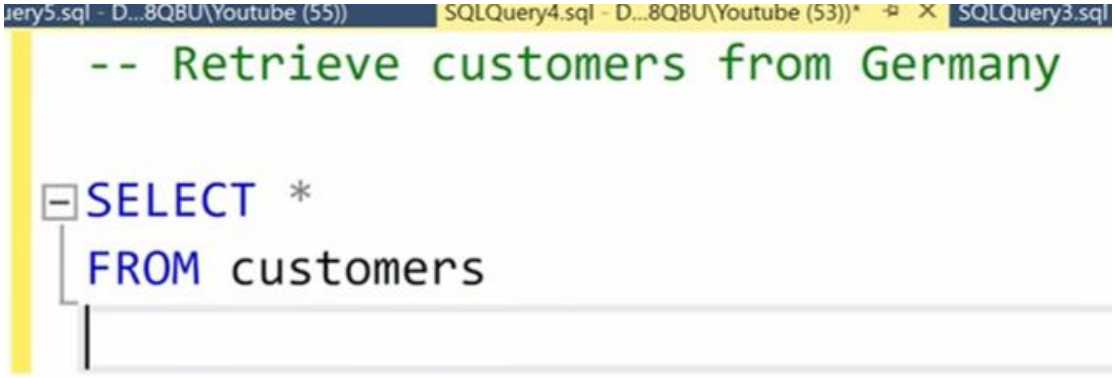


Filtering rows based on conditions


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


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
name, Country
1 FROM Table
2 WHERE Condition
name Country
John USA
Georg UK
Score > 500


Query5.sql - D...8QBU\Youtube (55))
SQLQuery4.sql - D...8QBU\Youtube (53))*
SQLQuery3.sql - D...8QBU\Youtube (64))*
SQLQuery2.sql
-- Retrieve customers with a score not equal to 0
SELECT *
FROM customers
WHERE score != 0
Results
Messages
id first_name country score
1 Maria Germany 350
2 John USA 900
3 Georg UK 750
4 Martin Germany 500



esults Messages

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

```sql
-- Retrieve customers from Germany

SELECT *
FROM customers
WHERE country = 'Germany'
```

Results Messages

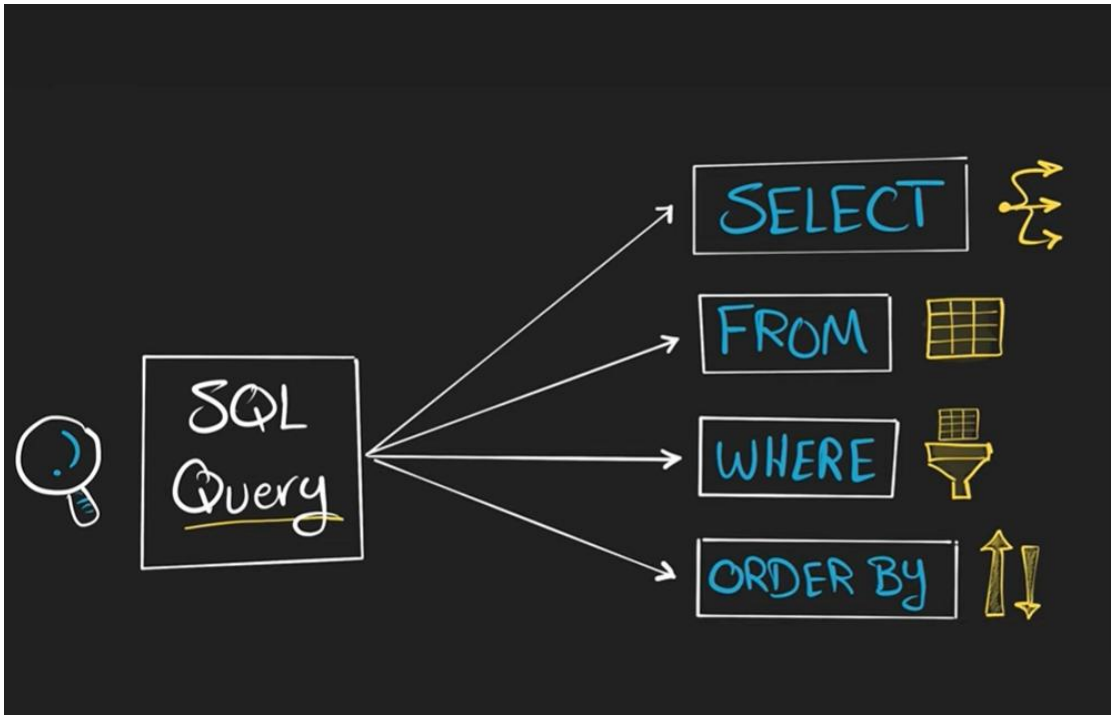
	id	first_name	country	score
1	1	Maria	Germany	350
2	4	Martin	Germany	500

```sql
-- Retrieve customers from Germany

SELECT
    first_name,
    country
FROM customers
WHERE country = 'Germany'
```

Results Messages

	first_name	country
1	Maria	Germany
2	Martin	Germany



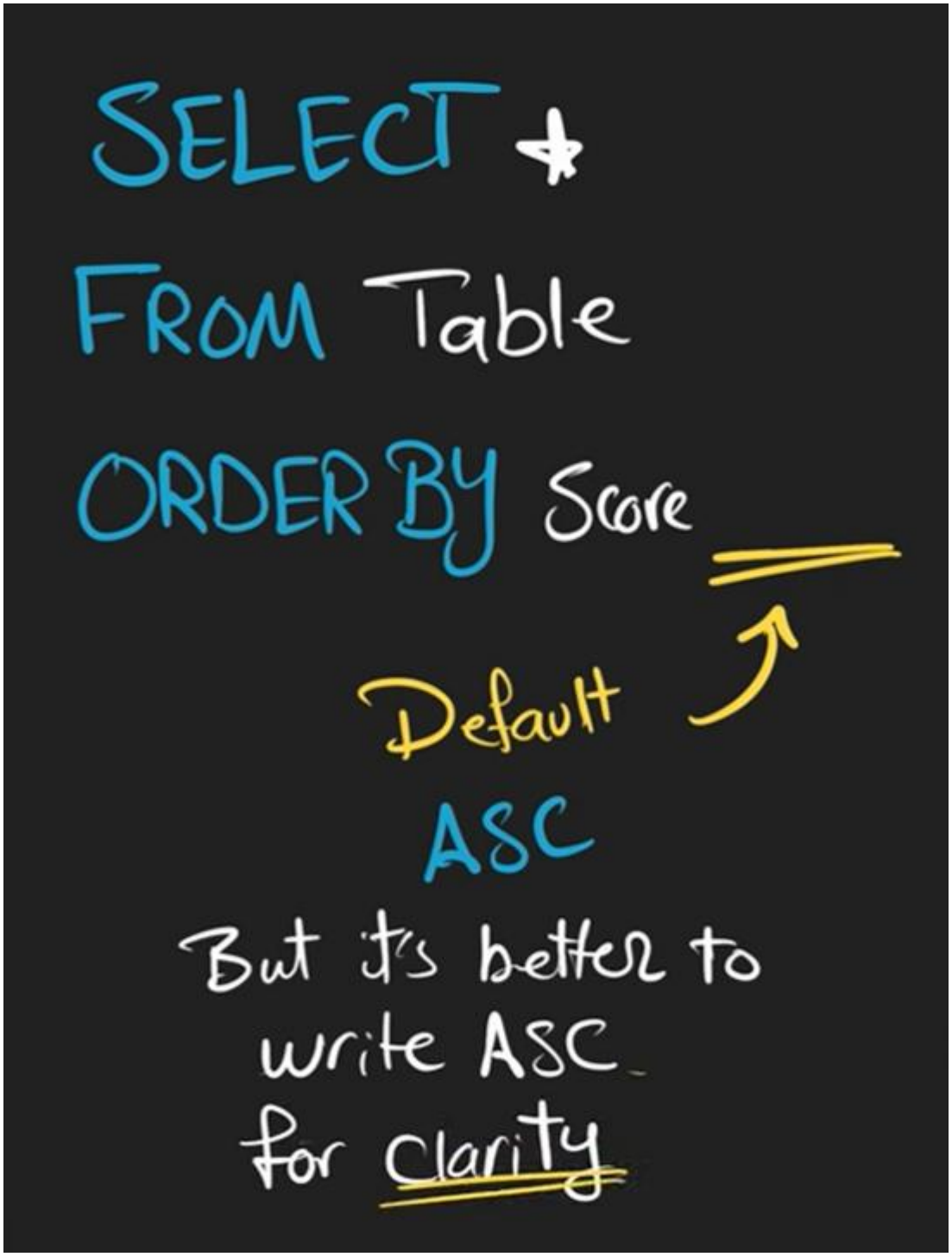
Database

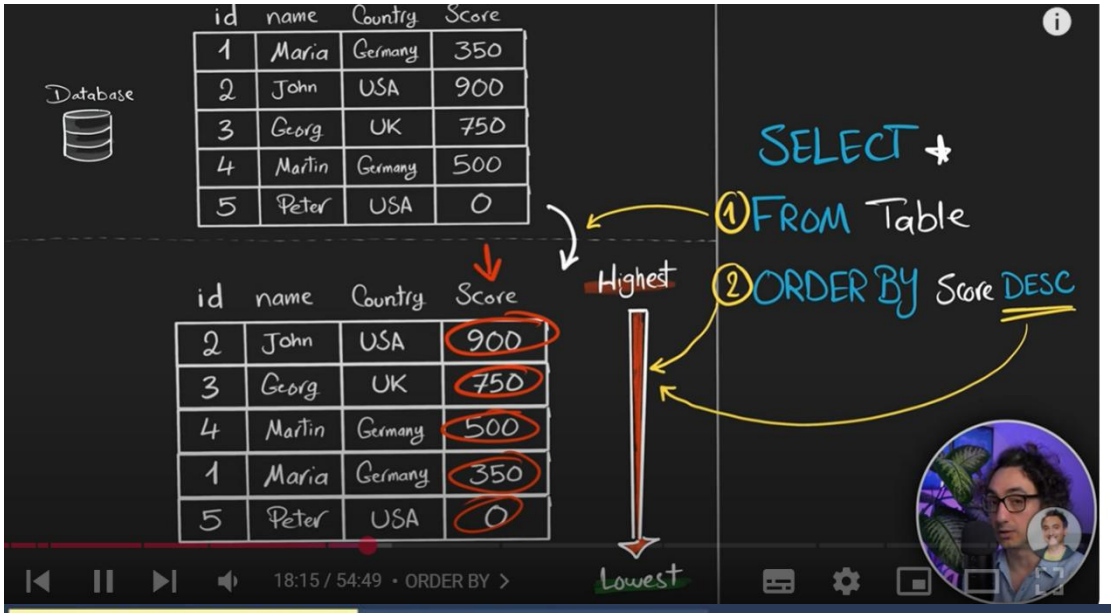
id	name	Country	Score
1	Maria	Germany	350
2	John	USA	900
3	Georg	UK	750
4	Martin	Germany	500
5	Peter	USA	0

Order By

Sort Your Data

- ASC: Lowest → Highest
- DESC: Highest → Lowest


SELECT *
FROM Table
ORDER BY Score
Default
ASC
But it's better to
write ASC
for clarity



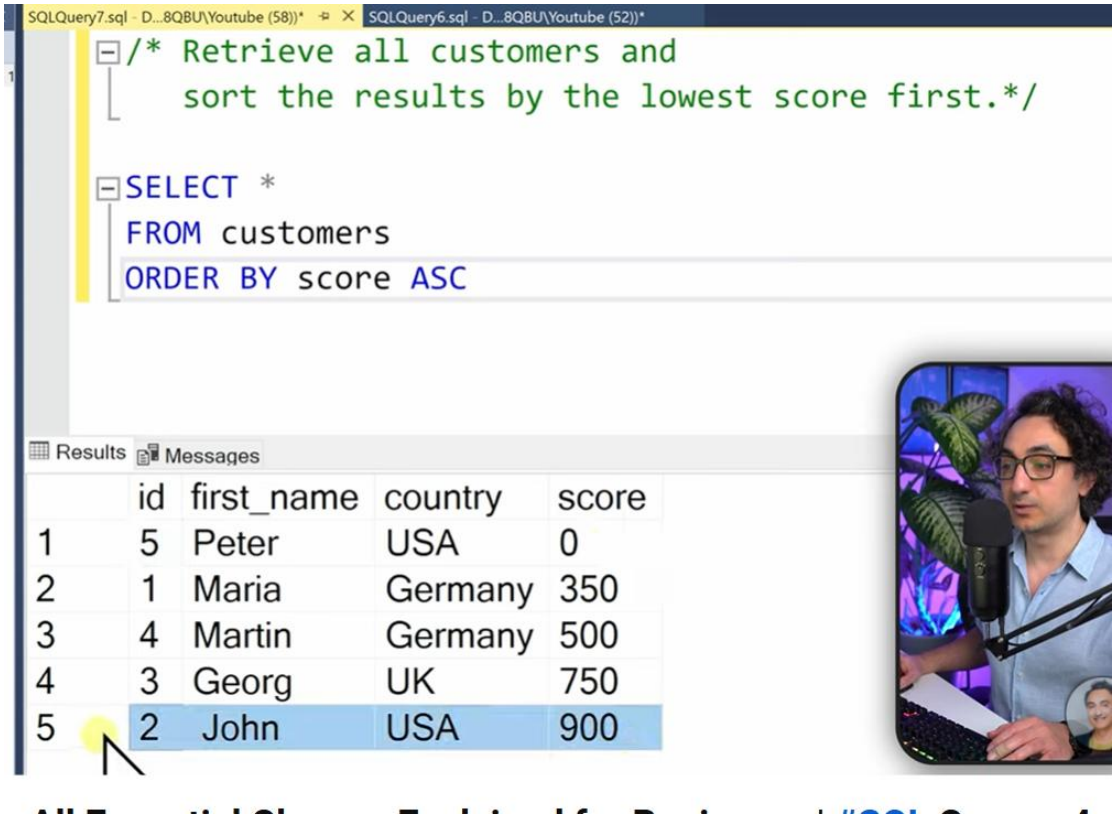
SQLQuery7.sql - D...8QBU\Youtube (58))* SQLQuery6.sql - D...8QBU\Youtube (52))*

```sql
/* Retrieve all customers and
   sort the results by the highest score first. */

SELECT *
FROM customers
ORDER BY score DESC
```

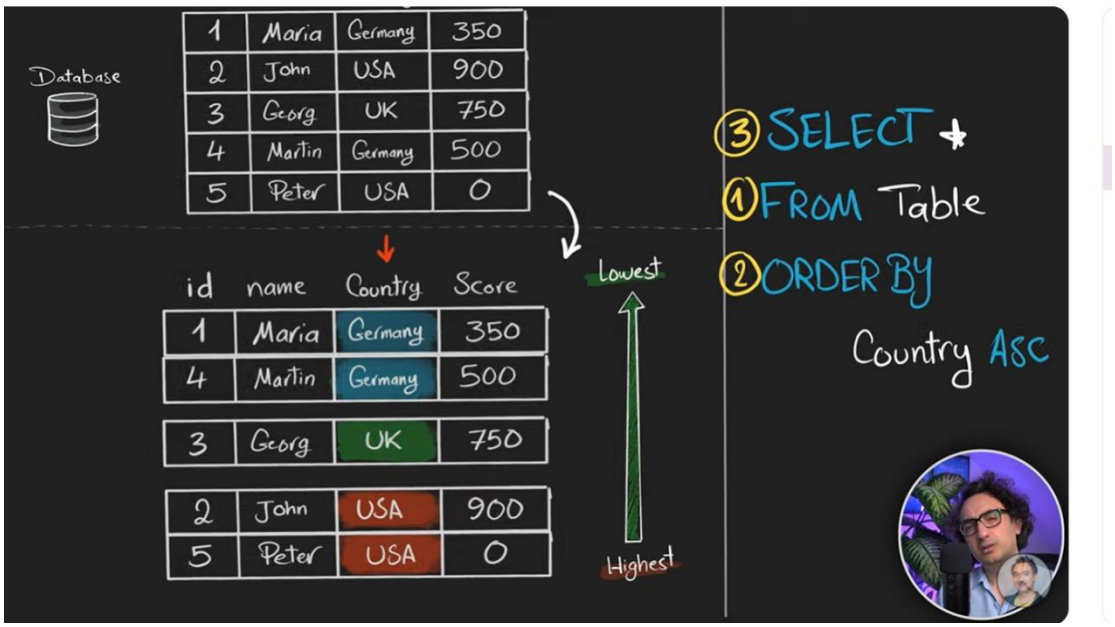
Results Messages

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

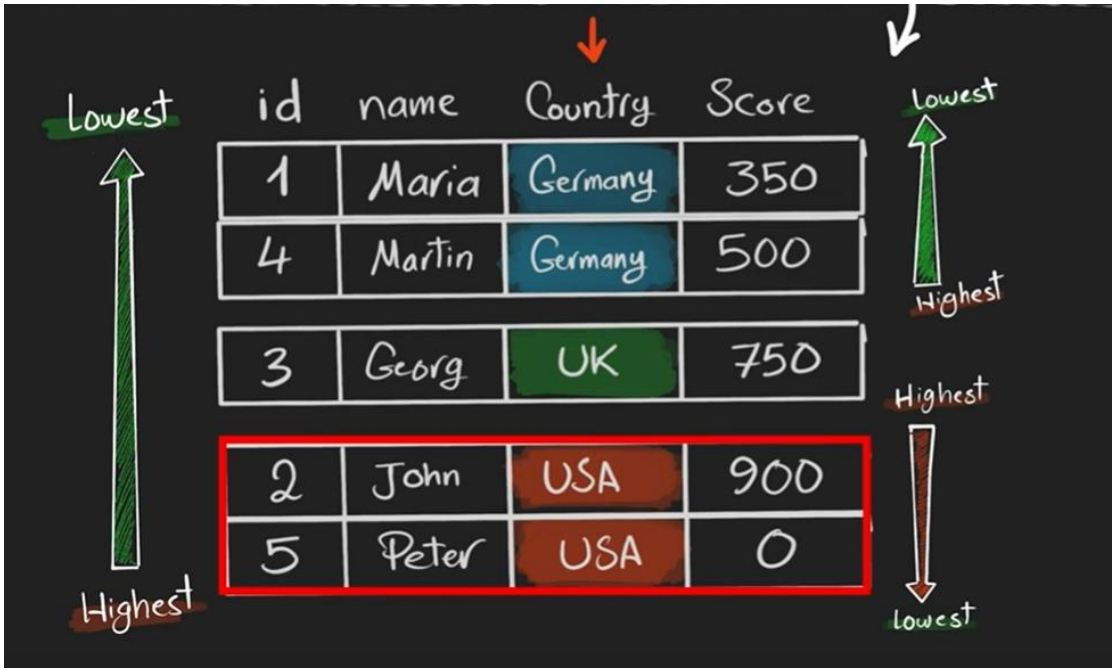


Nested Order by

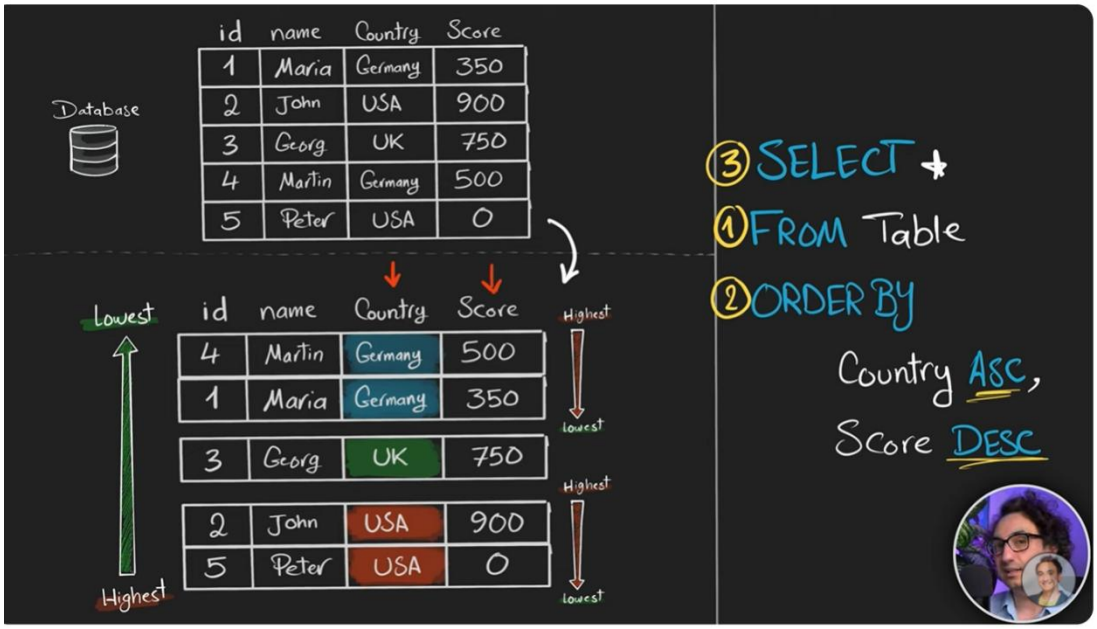
Here we see data is not correctly sorted it is sorted only in alphabetically order

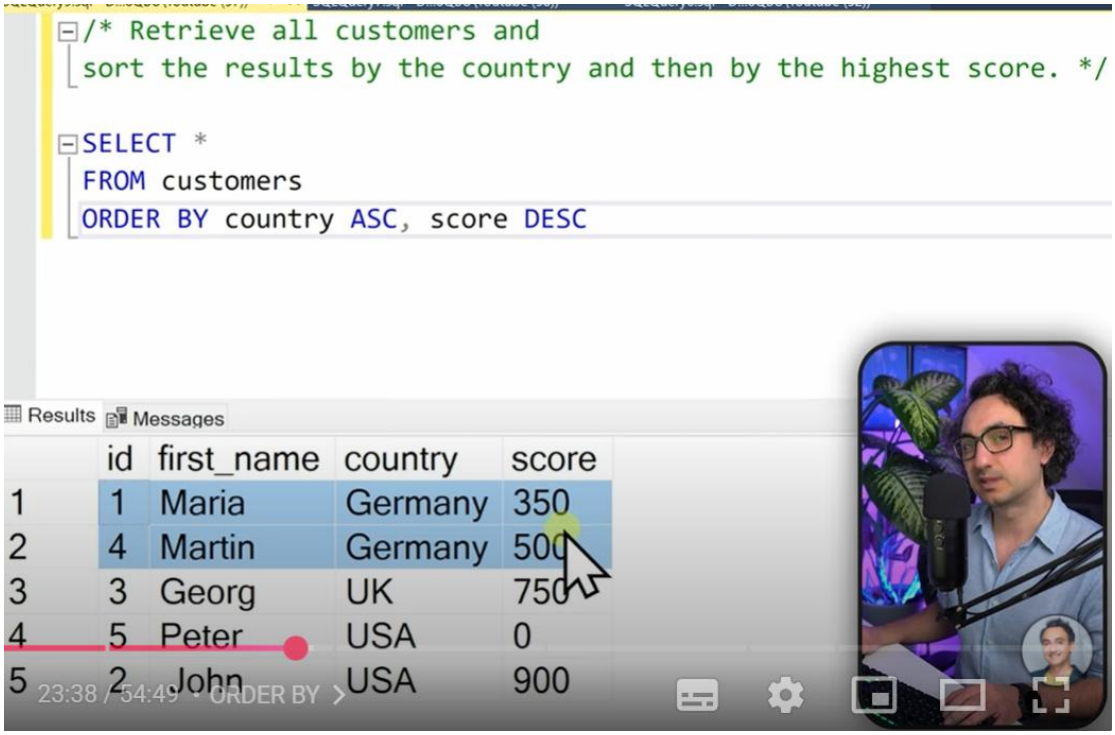


This issue happen on repetition of data columns

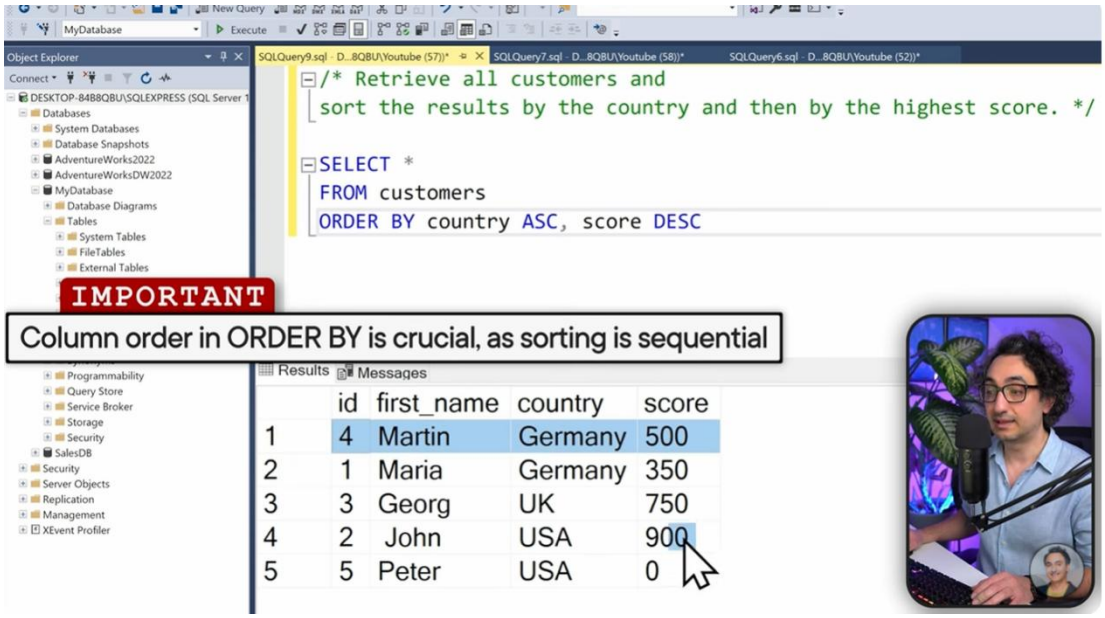

Lowest
id
name
Country
Score
Lowest
1
Maria
Germany
350
4
Martin
Germany
500
Highest
3
Georg
UK
750
Highest
2
John
USA
900
5
Peter
USA
0
Highest
Lowest

correction


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
Highest
Lowest
Highest
Lowest
③ SELECT *
① FROM Table
② ORDER BY
Country ASC,
Score DESC



Scores shorted after country so more priority to country

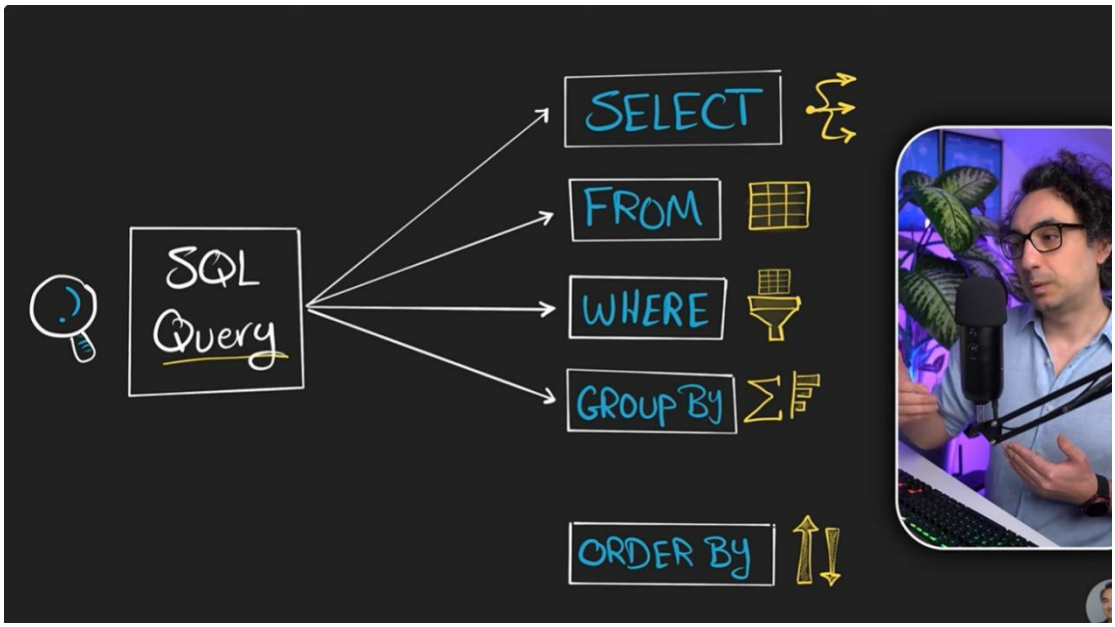


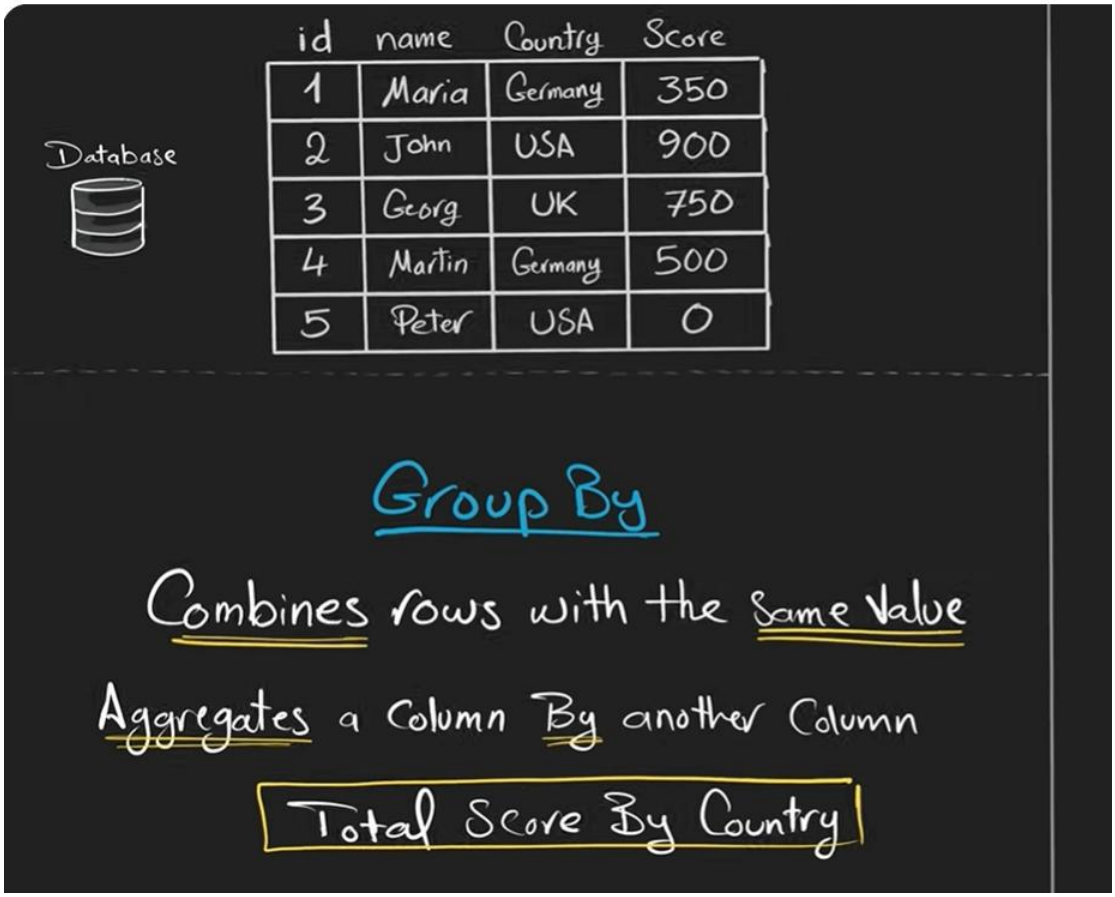
```sql
/* Retrieve all customers and
sort the results by the country and then by the highest score. */

SELECT *
FROM customers
ORDER BY score DESC, country ASC
```

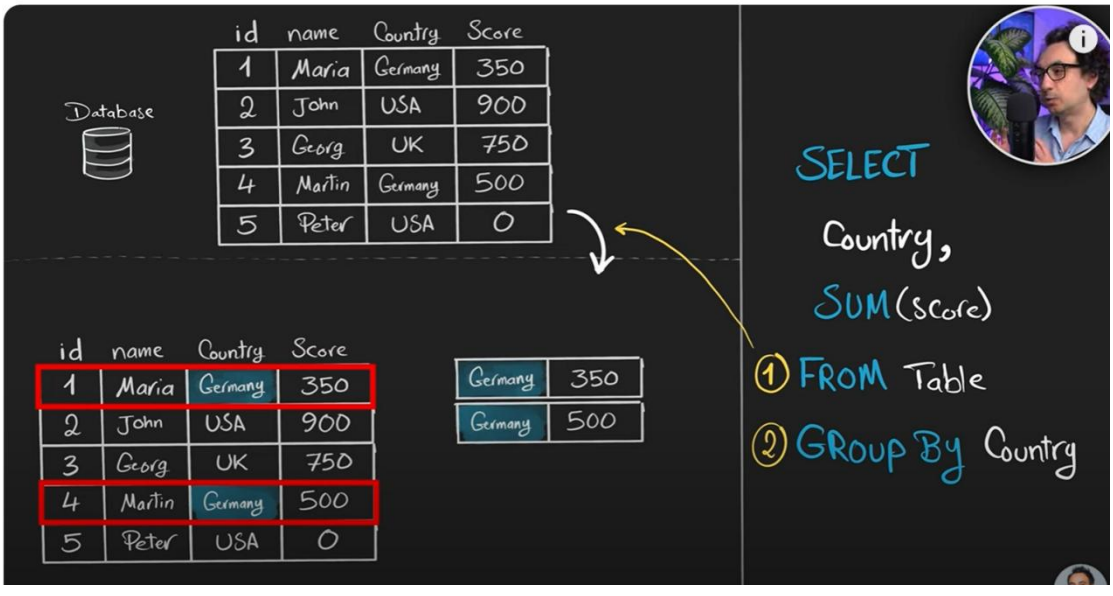
Results | Messages

	id	first_name	country	score
1	2	John	USA	900
2	3	Georg	UK	750
3	4	Martin	Germany	500
4	1	Maria	Germany	350
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
Group By
Combines rows with the Same Value
Aggregates a Column By another Column
Total Score By Country


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