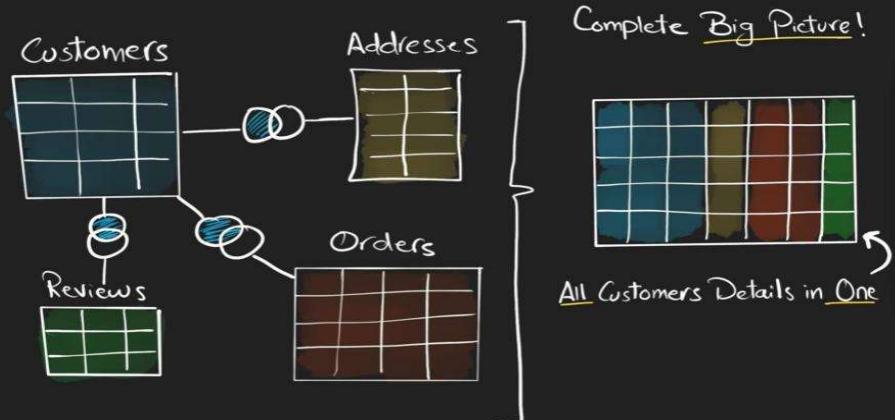
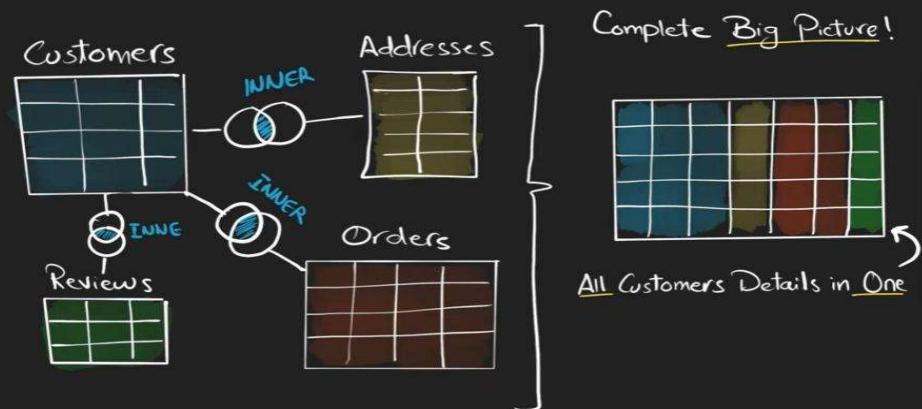


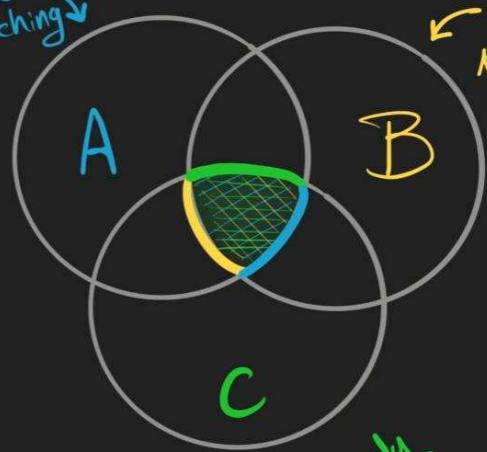
1] Recombine Data



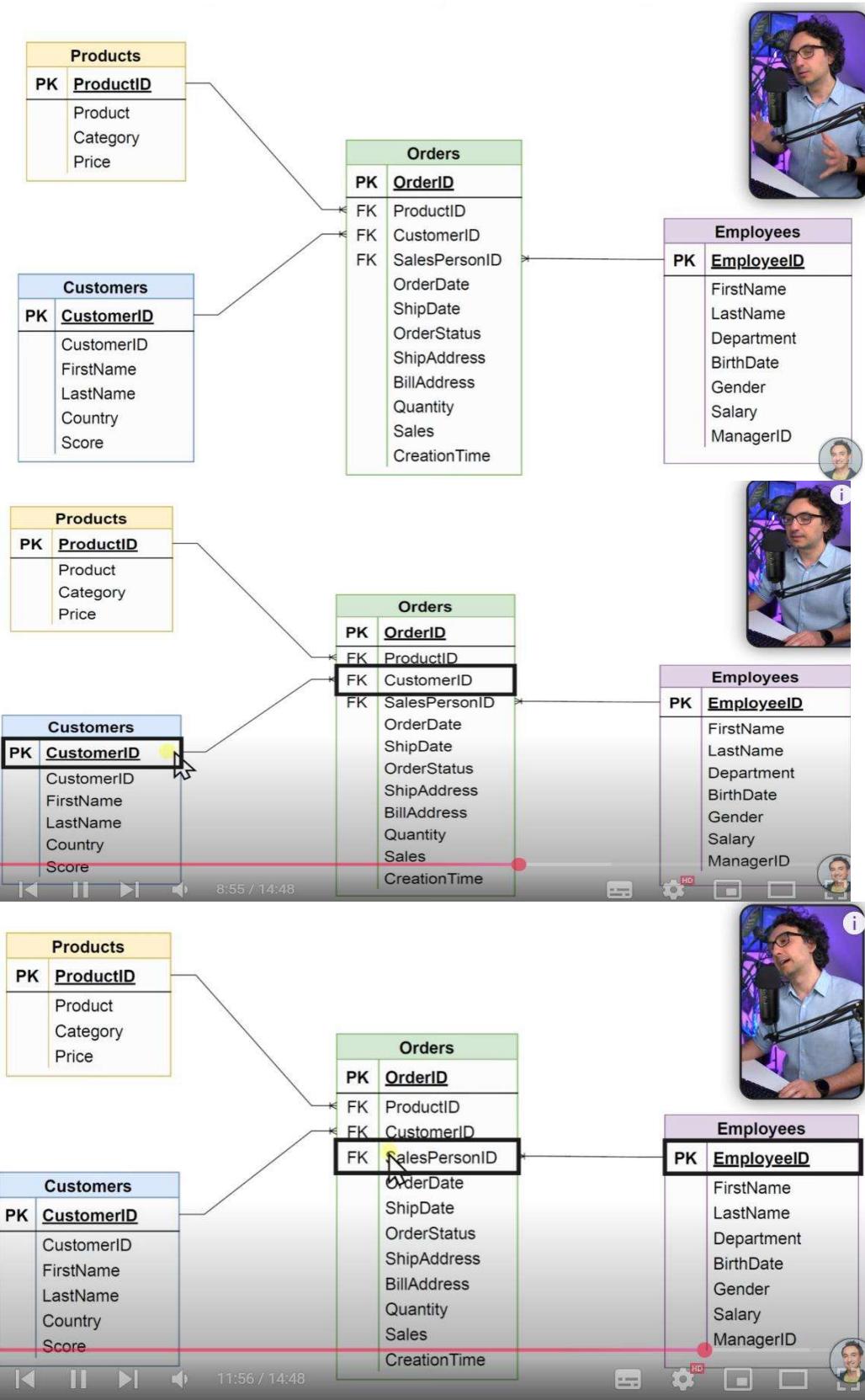
1] Recombine Data



SELECT *
FROM A
INNER B ON...
INNER C ON...
⋮



Only Matching

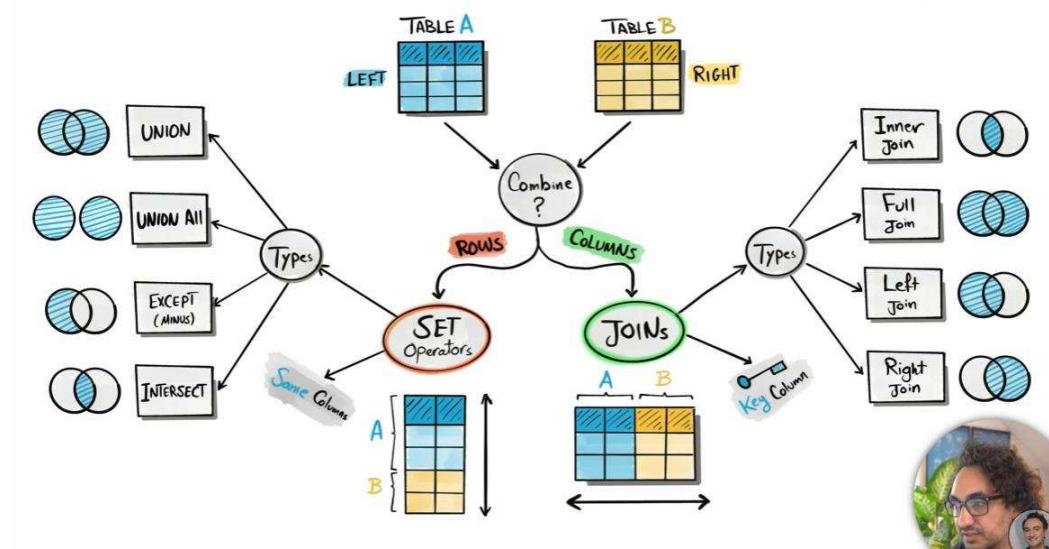
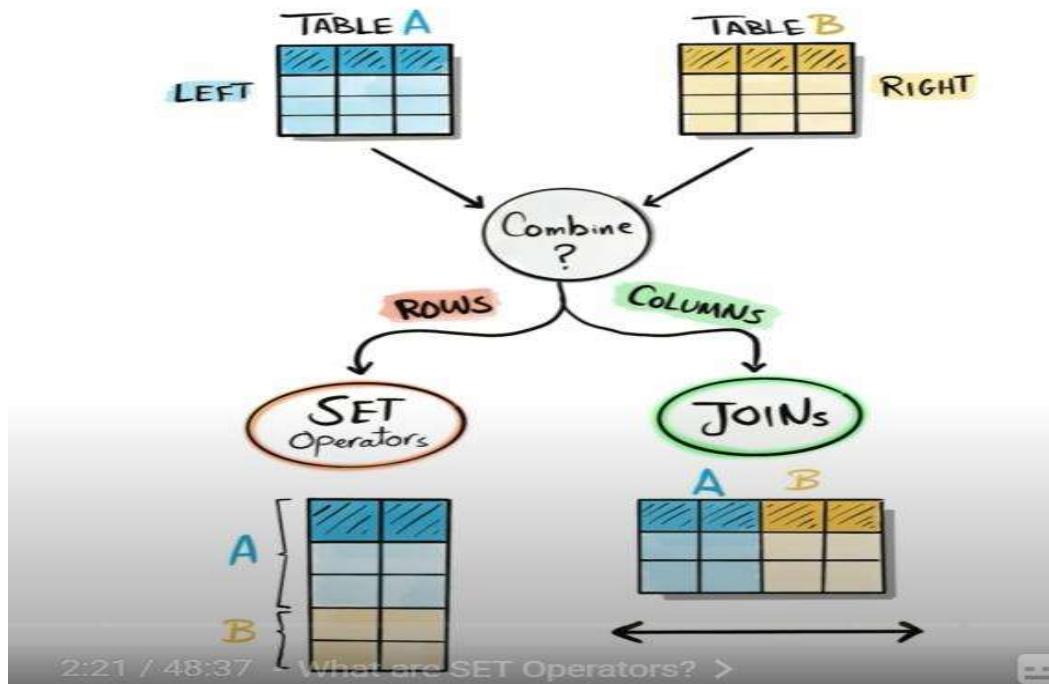


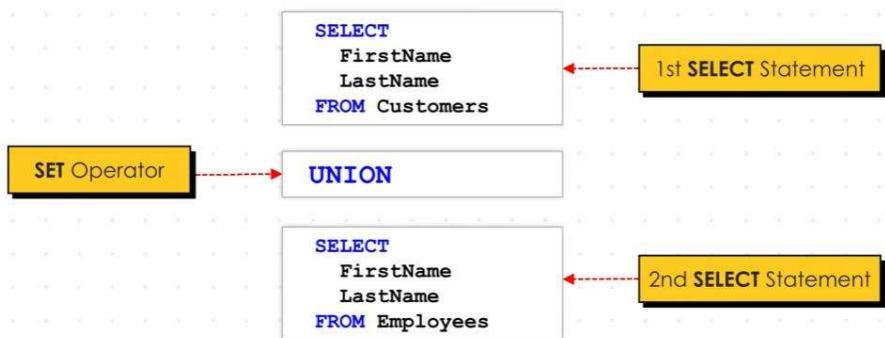
```
SELECT
    o.OrderID,
    o.Sales,
    c.FirstName AS CustomerFirstName,
    c.LastName AS CustomerLastName,
    p.Product AS ProductName,
    p.Price,
    e.FirstName AS EmployeeFirstName,
    e.LastName AS EmployeeLastName
FROM Sales.Orders AS o
LEFT JOIN Sales.Customers AS c
ON o.CustomerID = c.CustomerID
LEFT JOIN Sales.Products AS p
ON o.ProductID = p.ProductID
LEFT JOIN Sales.Employees AS e
ON o.SalesPersonID = e.EmployeeID
```

	OrderID	Sales	CustomerFirstName	CustomerLastName	ProductName	Price	EmployeeFirstName	EmployeeLastName
2	2	15	Mary	NULL	Tire	15	Mary	NULL
3	3	20	Jossef	Goldberg	Bottle	10	Carol	Baker
4	4	60	Jossef	Goldberg	Gloves	30	Mary	NULL
5	5	25	Kevin	Brown	Caps	25	Carol	Baker
6	6	50	Mary	NULL	Caps	25	Carol	Baker
7	7	30	Jossef	Goldberg	Tire	15	Frank	Lee
8	8	90	Mark	Schwarz	Bottle	10	Mary	NULL
9	9	20	Kevin	Brown	Bottle	10	Mary	NULL
10	10	60	Mary	NULL	Tire	15	Carol	Baker

SET OPERATORS

Set operations in SQL combine the results of multiple queries into a single result set.



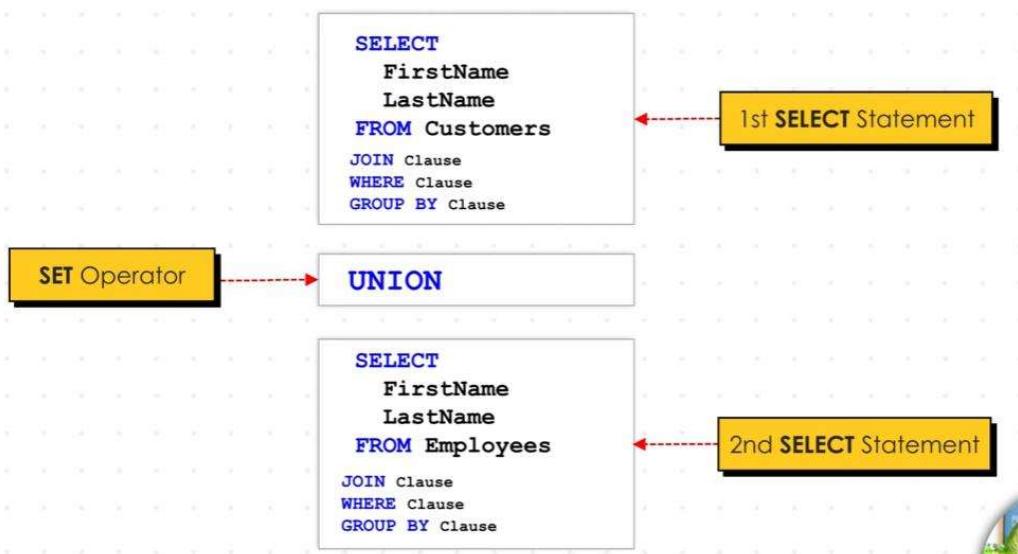


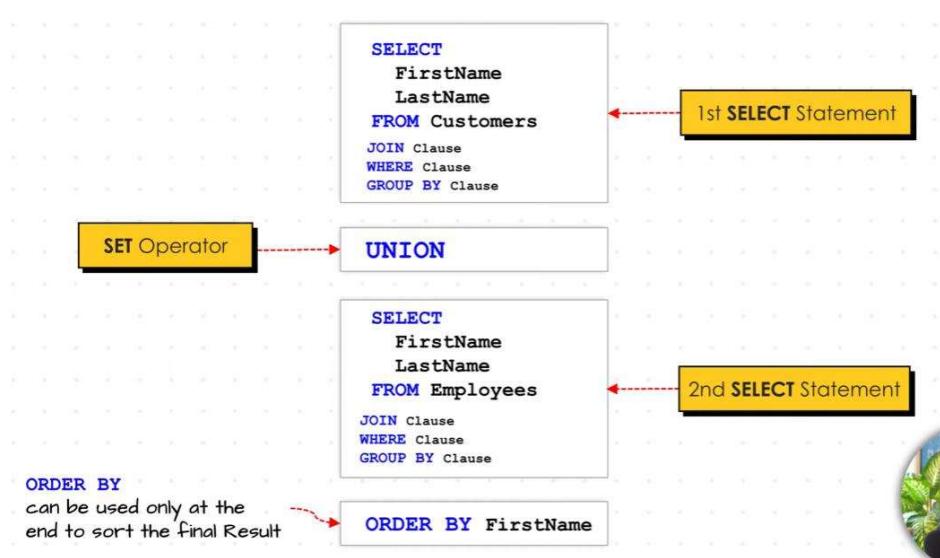
#1 RULE | SQL CLAUSES

- SET Operator can be used almost in all clauses

WHERE | JOIN | GROUP BY | HAVING

- ORDER BY is allowed only once at the end of query





#2 RULE | NUMBER OF COLUMNS

The number of columns in each query must be the same

```

SQLQuery6.sql - D:\SQLBU\Youtube (53) 1 X SQLQuery5.sql - D...\SQLBU\Yout
SELECT FirstName,
       LastName
  FROM Sales.Customers
UNION
SELECT FirstName, ?,
       LastName
  FROM Sales.Employees
    
```

	FirstName	LastName
1	Anna	Adams
2	Carol	Baker
3	Frank	Lee
4	Jossef	Goldberg
5	Kevin	Brown
6	Mark	Schwarz
7	Mary	NULL
8	Michael	Ray

Ex-

Error because set operators need equal number of columns



```
SELECT
CustomerID
FirstName,
LastName
FROM Sales.Customers
UNION
SELECT
FirstName,
LastName
FROM Sales.Employees
```

Msg 205, Level 16, State 1, Line 1
All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of columns. The query failed.
Completion time: 2024-05-17T18:07:29.4008277+02:00

#3 RULE | DATA TYPES

Data types of columns in each query must be compatible

Object Explorer:

- System Tables
- FileTables
- External Tables
- Graph Tables
- Sales.Customers
- Columns
 - CustomerID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Country (varchar(50), null)
 - Score (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

Sales.Employees

- Columns
 - EmployeeID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Department (varchar(50), null)
 - BirthDate (date, null)
 - Gender (char(1), null)
 - Salary (int, null)
 - ManagerID (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

SQL Query Editor:

```

SELECT
    FirstName,
    LastName
FROM Sales.Customers

UNION

SELECT
    FirstName
    LastName
FROM Sales.Employees
  
```

Results Grid:

	FirstName	LastName
1	Anna	Adams
2	Carol	Baker
3	Frank	Lee
4	Jossef	Goldberg
5	Kevin	Brown
6	Mark	Schwarz
7	Mary	NULL
8	Michael	Ray

Message Bar:

Query executed successfully.

Mismatched Data Type:

Tables

- System Tables
- FileTables
- External Tables
- Graph Tables
- Sales.Customers
- Columns
 - CustomerID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Country (varchar(50), null)
 - Score (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

Sales.Employees

- Columns
 - EmployeeID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Department (varchar(50), null)
 - BirthDate (date, null)
 - Gender (char(1), null)
 - Salary (int, null)
 - ManagerID (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

```

SELECT
    CustomerID,
    LastName
FROM Sales.Customers

UNION

SELECT
    FirstName,
    LastName
FROM Sales.Employees
  
```

Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'Frank' to data type int.

Completion time: 2024-05-17T18:11:09.0348376+02:00

Query completed with errors.

Tables

- System Tables
- FileTables
- External Tables
- Graph Tables
- Sales.Customers
- Columns
 - CustomerID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Country (varchar(50), null)
 - Score (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

Sales.Employees

- Columns
 - EmployeeID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Department (varchar(50), null)
 - BirthDate (date, null)
 - Gender (char(1), null)
 - Salary (int, null)
 - ManagerID (int, null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

```

SELECT
    CustomerID,
    LastName
FROM Sales.Customers

UNION

SELECT
    EmployeeID,
    LastName
FROM Sales.Employees
  
```

	CustomerID	LastName
1	1	Goldberg
2	1	Lee
3	2	Brown
4	3	NULL
5	4	Ray
6	4	Schwarz
7	5	Adams
8	5	Baker

Query executed successfully.

#4 RULE | ORDER OF COLUMNS

The order of the columns in each query must be the same

The screenshot shows the Object Explorer on the left, displaying the database structure of AdventureworksDW2022, specifically the SalesDB schema. Two queries are open in the center pane:

```
SELECT CustomerID, LastName
FROM Sales.Customers
UNION
SELECT EmployeeID, LastName
FROM Sales.Employees
```

The results pane on the right shows the output of the UNION query:

	CustomerID	LastName
1	1	Goldberg
2	1	Lee
3	2	Brown
4	3	NULL
5	4	Ray
6	4	Schwarz
7	5	Adams
8	5	Baker

The status bar at the bottom indicates the video is at 9:13 / 48:37 and the set is #4 Rule.

The screenshot shows two separate sessions in SQL Server Management Studio. On the left, the Object Explorer displays the database structure for 'DESKTOP-B4B8QBU\SQLEXPRESS' (SQL Server 16.0.1c). Two queries are run in the main pane:

```

SELECT
    LastName,
    CustomerID
FROM Sales.Customers

UNION

SELECT
    EmployeeID,
    LastName
FROM Sales.Employees

```

The results show an error message: "Msg 245, Level 16, State 1, Line 1 Conversion failed when converting the varchar value 'Goldberg' to data type int." Below the results, the completion time is listed as 2024-05-17T18:20:02.4078093+02:00.

In the second session, the same queries are run, but the column names 'LastName' and 'EmployeeID' are highlighted in red boxes, indicating they are being mapped or aliased. The results show the same error message and completion time.

SQL match or map columns with its order.

#5 RULE | COLUMN ALIASES

The column names in the result set are determined by the column names specified in the first query.

First Query Controls the naming of Output Here naming of second query is totally Ignored

Object Explorer

Connect ▾

DESKTOP-B4B8QBU\SQLEXPRESS (SQL Server 16.0.1C)

- Databases
 - System Databases
 - AdventureWorks2022
 - AdventureWorksDW2022
 - SalesDB
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Sales.Customers
 - Columns
 - CustomerID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Country (varchar(50), null)
 - Score (int, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Sales.Employees
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Sales.Employees
 - Columns
 - EmployeeID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Department (varchar(50), null)
 - BirthDate (date, null)
 - Gender (char(1), null)
 - Salary (int, null)
 - ManagerID (int, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Sales.Orders

SQLQuery6.sql - D:\BQB\Youtube (53)* SQLQuery5.sql - D:\BQB\Youtube (62)*

```

SELECT
CustomerID,
LastName
FROM Sales.Customers

UNION

SELECT
EmployeeID,
LastName
FROM Sales.Employees
  
```

203 %

Results Messages

	CustomerID	LastName
1	1	Goldberg
2	1	Lee
3	2	Brown
4	3	NULL
5	4	Ray
6	4	Schwarz
7	5	Adams
8	5	Baker

11:12 / 48:37 • SET: #5 Rule >

Object Explorer

Connect ▾

ESKTOP-B4B8QBU\SQLEXPRESS (SQL Server 16.0.1C)

- Databases
 - System Databases
 - AdventureWorks2022
 - AdventureWorksDW2022
 - SalesDB
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Sales.Customers
 - Columns
 - CustomerID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Country (varchar(50), null)
 - Score (int, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Sales.Employees
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Sales.Employees
 - Columns
 - EmployeeID (PK, int, not null)
 - FirstName (varchar(50), null)
 - LastName (varchar(50), null)
 - Department (varchar(50), null)
 - BirthDate (date, null)
 - Gender (char(1), null)
 - Salary (int, null)
 - ManagerID (int, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Sales.Orders

SQLQuery6.sql - D:\BQB\Youtube (53)* SQLQuery5.sql - D:\BQB\Youtube (62)*

1st query controls column names

```

SELECT
CustomerID,
LastName
FROM Sales.Customers

UNION

SELECT
EmployeeID,
LastName
FROM Sales.Employees
  
```

203 %

Results Messages

	CustomerID	LastName
1	1	Goldberg
2	1	Lee
3	2	Brown
4	3	NULL
5	4	Ray
6	4	Schwarz
7	5	Adams
8	5	Baker

Object Explorer

SQLQuery6.sql - D:\QBQU\Youtube (53)* SQLQuery5.sql - D:\QBQU\Youtube (52)*

```

SELECT CustomerID AS ID,
      LastName
   FROM Sales.Customers
UNION
SELECT EmployeeID,
      LastName
   FROM Sales.Employees
  
```

Results

ID	LastName
1	Goldberg
2	Lee
3	Brown
4	NULL
5	Ray
6	Schwarz
7	Adams
8	Baker

Query executed successfully.

All Alias do in the first query not in second.

Object Explorer

SQLQuery6.sql - D:\QBQU\Youtube (53)* SQLQuery5.sql - D:\QBQU\Youtube (52)*

```

SELECT CustomerID AS ID,
      LastName
   FROM Sales.Customers
UNION
SELECT EmployeeID,
      LastName AS Last_Name
   FROM Sales.Employees
  
```

Results

ID	LastName
1	Goldberg
2	Lee
3	Brown
4	NULL
5	Ray
6	Schwarz
7	Adams
8	Baker

#6 RULE | CORRECT COLUMNS

- Even if all rules are met and SQL shows no errors, the result may be incorrect.
- Incorrect column selection leads to inaccurate results.

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure of 'SalesDB'. In the center, the 'SQLQuery6.sql' window contains the following code:

```
SELECT FirstName, LastName
FROM Sales.Customers
UNION
SELECT FirstName, LastName
FROM Sales.Employees
```

The 'Results' tab on the right shows the output of the query:

	FirstName	LastName
1	Anna	Adams
2	Carol	Baker
3	Frank	Lee
4	Jossef	Goldberg
5	Kevin	Brown
6	Mark	Schwarz
7	Mary	NULL
8	Michael	Ray

A message at the bottom of the results pane says "Query executed successfully."

In accurate result

The screenshot shows the SSMS interface. On the left, the Object Explorer displays the database structure of 'DESKTOP-B84B8QBU\SQLEXPRESS' (SQL Server 16.0.10). It lists databases like System Databases, AdventureWorks2022, AdventureWorksDW2022, and SalesDB. Under SalesDB, it shows Tables (Sales.Customers, Sales.Employees), Columns, Keys, Constraints, Triggers, Indexes, and Statistics. The Sales.Employees table has columns: EmployeeID (PK, int, not null), FirstName (varchar(50), null), LastName (varchar(50), null), Department (varchar(50), null), BirthDate (date, null), Gender (char(1), null), Salary (int, null), and ManagerID (int, null).

The main window shows two queries in the 'SQLQuery6.sql' tab:

```

SELECT
    FirstName,
    LastName
FROM Sales.Customers

UNION

SELECT
    LastName,
    FirstName
FROM Sales.Employees

```

The results grid displays 10 rows of data from the UNION query:

	FirstName	LastName
1	NULL	Mary
2	Anna	Adams
3	Baker	Carol
4	Brown	Kevi
5	Jossef	Goldberg
6	Kevin	Brown
7	Lee	Frank
8	Mark	Schwarz
9	Mary	NULL
10	Ray	Michael

Same information should be mapped between two queries that we have to take care otherwise we will get inaccurate result.

RULES OF SET OPERATORS

#1 RULE | ORDER BY can be used only once

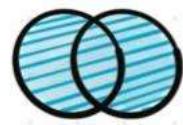
#2 RULE | Same Number of Columns

#3 RULE | Matching Data Types

#4 RULE | Same Order of Columns

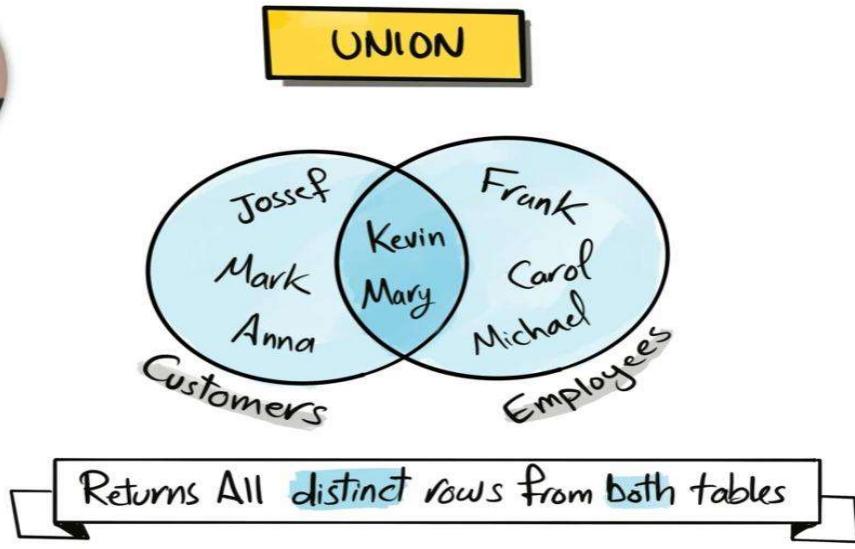
#5 RULE | First Query Controls Aliases

#6 RULE | Mapping Correct Columns



UNION

- Returns all district rows from both queries.
- Removes duplicate rows from the result.



Object Explorer

Connect to SQL Server

Databases: SalesDB, AdventureWorksDW2022, AdventureWorks2022, System Databases, Database Snapshots, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Security, Server Objects, Replication, Management, XEvent Profiler

SQLQuery3.sql - D:\BQBU\Youtube (75)* SQLQuery2.sql - D:\BQBU\Youtube (68)*

```
-- Combine the data from employees and customers into one table
SELECT *
FROM Sales.Customers;
```

1st Query

```
SELECT *
FROM Sales.Employees;
```

2nd Query

Results Messages

CustomerID FirstName LastName Country Score

1	Jossef	Goldberg	Germany	350
2	Kevin	Brown	USA	900
3	Mary	NULL	USA	750
4	Mark	Schwarz	Germany	500
5	Anna	Adams	USA	NULL

EmployeeID FirstName LastName Department BirthDate Gender Salary ManagerID

1	Frank	Lee	Marketing	1988-12-05	M	55000	NULL
2	Kevin	Brown	Marketing	1972-11-25	M	65000	1
3	Mary	NULL	Sales	1986-01-05	F	75000	1
4	Michael	Ray	Sales	1977-02-10	M	90000	2
5	Carol	Baker	Sales	1982-02-11	F	55000	3

17:31 / 48:37 UNION > HD DESKTOP-B4B8QBU\SQLEXPRESS DESKTOP-KABBY

1st Result

2nd Result

Object Explorer

Connect to SQL Server

Databases: SalesDB, AdventureWorksDW2022, AdventureWorks2022, System Databases, Database Snapshots, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Security, Server Objects, Replication, Management, XEvent Profiler

SQLQuery3.sql - D:\BQBU\Youtube (75)* SQLQuery2.sql - D:\BQBU\Youtube (68)*

```
-- Combine the data from employees and customers into one table
SELECT
FirstName,
LastName
FROM Sales.Employees
UNION
SELECT
FirstName,
LastName
FROM Sales.Customers
```

Results Messages

FirstName LastName

1	Anna	Adams
2	Carol	Baker
3	Frank	Lee
4	Jossef	Goldberg
5	Kevin	Brown
6	Mark	Schwarz
7	Mary	NULL
8	Michael	Ray

Object Explorer

Connect to SQL Server

Databases: SalesDB, AdventureWorksDW2022, AdventureWorks2022, System Databases, Database Snapshots, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Security, Server Objects, Replication, Management, XEvent Profiler

SQLQuery3.sql - D:\BQBU\Youtube (75)* SQLQuery2.sql - D:\BQBU\Youtube (68)*

```
-- Combine the data from employees and customers into one table
SELECT
FirstName,
LastName
FROM Sales.Customers
UNION
SELECT
FirstName,
LastName
FROM Sales.Employees
```

Results Messages

FirstName LastName

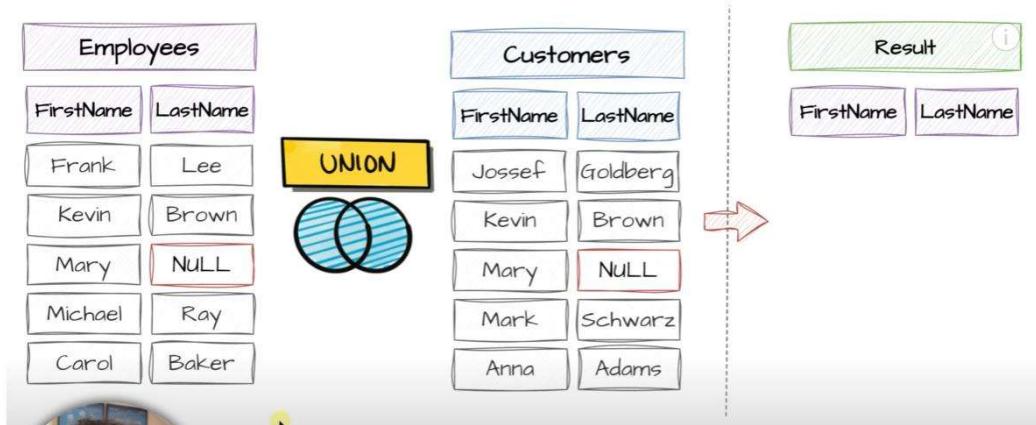
1	Anna	Adams
2	Carol	Baker
3	Frank	Lee
4	Jossef	Goldberg
5	Kevin	Brown
6	Mark	Schwarz
7	Mary	NULL
8	Michael	Ray

19:08 / 48:37 UNION > HD

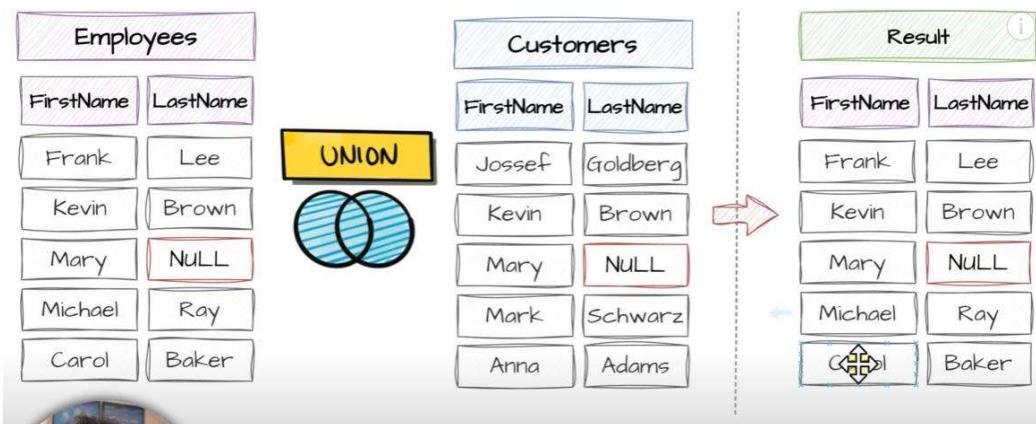
ORDER OF QUERIES

The order of queries in a UNION operation does not affect the result

How Union Works - Execution of union



All all data from first table with no duplicates values



Combines all data from second table by checking it become duplicate from first table or not if it same in first table then SQL ignores that and check for next one

UNION ALL

Returns all rows from both queries, including duplicates.

It combine the data from both query into one table along with
duplicated it keep whole data
Not removed duplicated

UNION ALL vs UNION

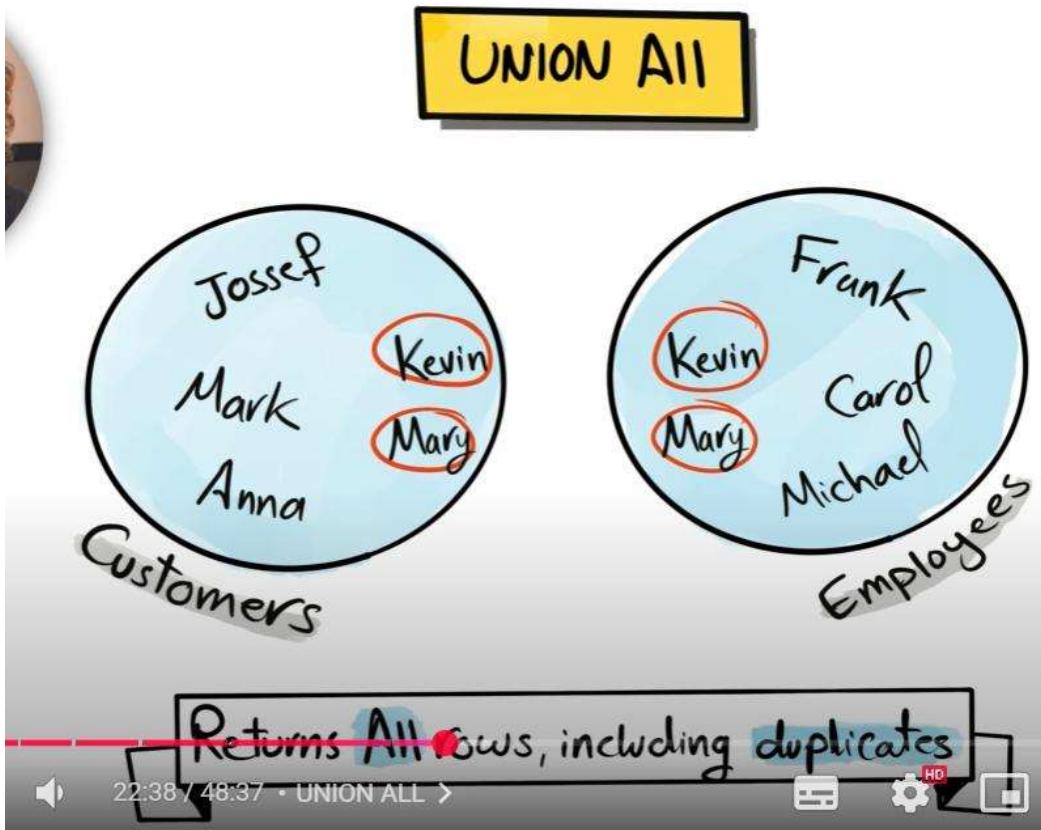
UNION ALL is generally faster than **UNION**

UNION ALL vs UNION

If you're confident there are no duplicates, use **UNION ALL**

UNION ALL vs UNION

Use **UNION ALL** to find duplicates and quality issues



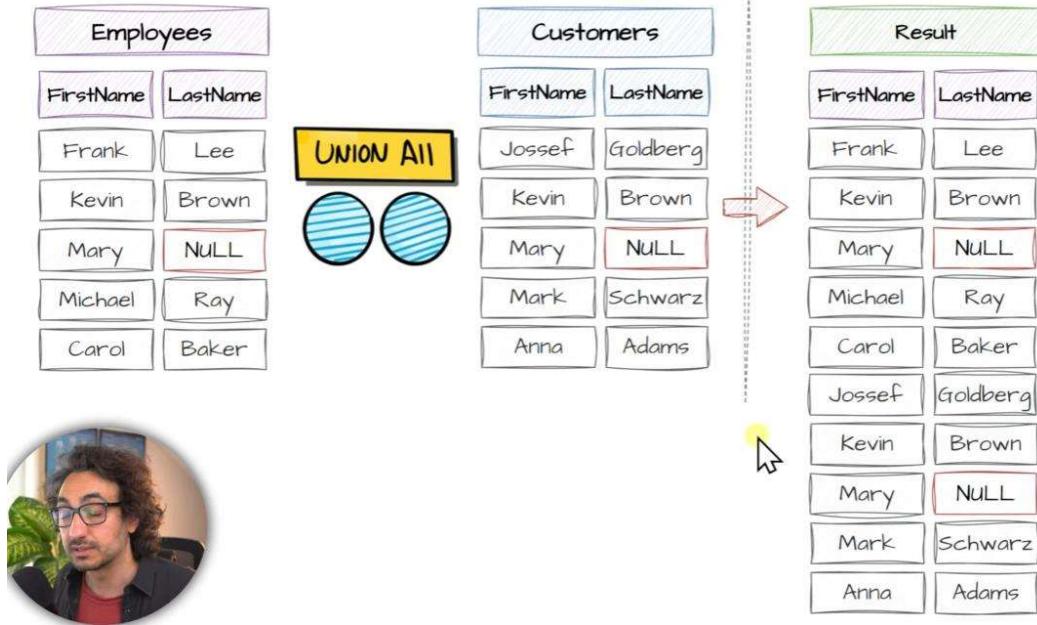
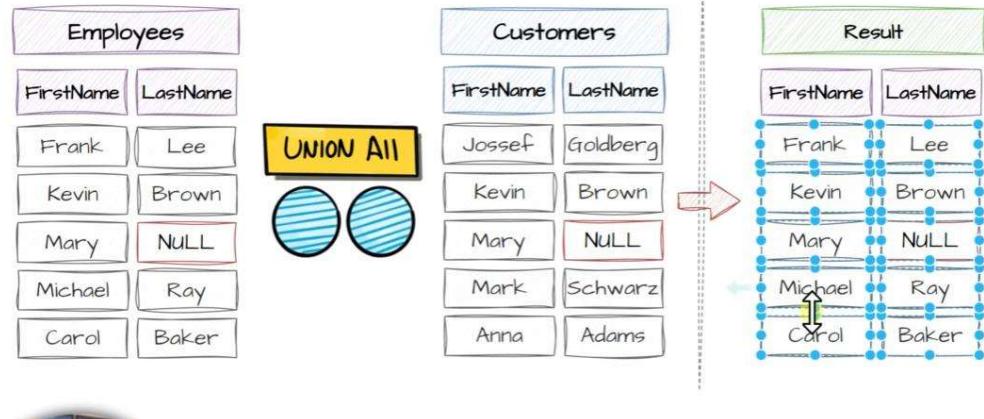
-- Combine the data from employees and customers into one table, including duplicates

```

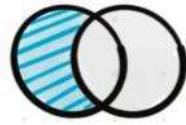
SELECT
    FirstName,
    LastName
FROM Sales.Employees
UNION ALL
SELECT
    FirstName,
    LastName
FROM Sales.Customers
  
```

Results

	FirstName	LastName
1	Frank	Lee
2	Kevin	Brown
3	Mary	NULL
4	Michael	Ray
5	Carol	Baker
6	Jossef	Goldberg
7	Kevin	Brown
8	Mary	NULL
9	Mark	Schwarz
10	Anna	Adams



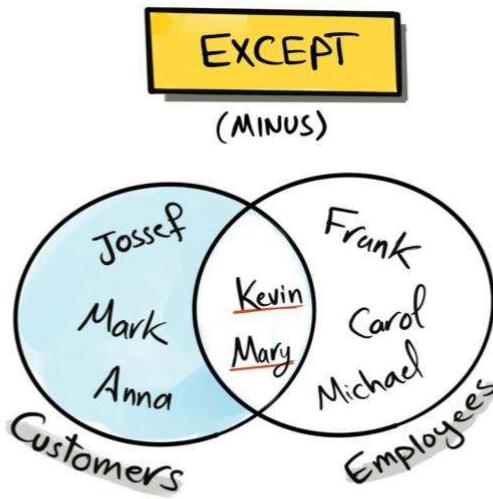
Except - Minus in Other databases but in sql we call it except



EXCEPT

- Returns all distinct rows from the first query that are not found in the second query.
- It is the only one where the order of queries affects the final result.

it's like the others it's going to go I remove the duplicates from the result



Returns unique rows in 1st Table that are not in 2nd Table



Employees who are not customers

-- Find the employees who are not customers at the same time

```

SELECT
    FirstName,
    LastName
FROM Sales.Employees
EXCEPT
SELECT
    FirstName,
    LastName
FROM Sales.Customers

```

	FirstName	LastName
1	Carol	Baker
2	Frank	Lee
3	Michael	Ray

Customers --- who are not employees

-- Find the employees who are not customers at the same time

```

SELECT
    FirstName,
    LastName
FROM Sales.Customers
EXCEPT
SELECT
    FirstName,
    LastName
FROM Sales.Employees

```

	FirstName	LastName
1	Anna	Adams
2	Jossef	Goldberg
3	Mark	Schwarz

ORDER OF QUERIES

The order of queries in a EXCEPT does affect the result !!

Data of one table check into another wheather employee exist in customer table or not if exist then it would not added in result if not exist it add into final result.

Order of query is very important

Employees

FirstName	LastName
Frank	Lee
Kevin	Brown
Mary	NULL
Michael	Ray
Carol	Baker

Customers

FirstName	LastName
Jossef	Goldberg
Kevin	Brown
Mary	NULL
Mark	Schwarz
Anna	Adams

Result

FirstName	LastName
Frank	Lee
Michael	Ray
Carol	Baker



Employees

FirstName	LastName
Frank	Lee
Kevin	Brown
Mary	NULL
Michael	Ray
Carol	Baker

Customers

FirstName	LastName
Jossef	Goldberg
Kevin	Brown
Mary	NULL
Mark	Schwarz
Anna	Adams

Result

FirstName	LastName
Frank	Lee
Michael	Ray
Carol	Baker

Customers

FirstName	LastName
Jossef	Goldberg
Kevin	Brown
Mary	NULL
Mark	Schwarz
Anna	Adams

Employees

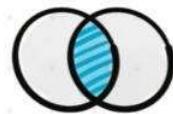
FirstName	LastName
Frank	Lee
Kevin	Brown
Mary	NULL
Michael	Ray
Carol	Baker

Result

FirstName	LastName
Jossef	Goldberg
Mark	Schwarz
Anna	Adams

Customer is main and employee table is for check in second scenerao as we changed order

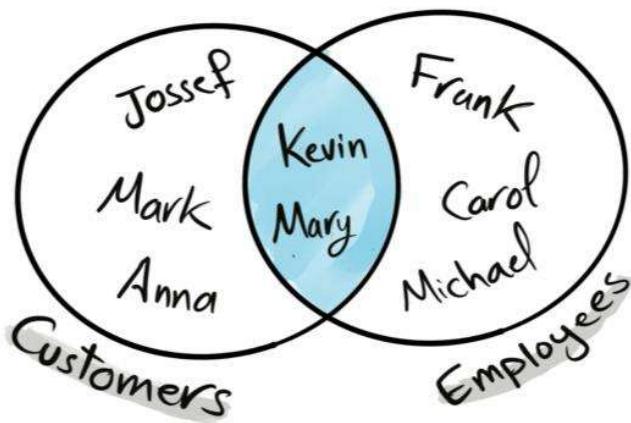
Intersect returns the duplicates



INTERSECT

Returns only the rows that are common in both queries

INTERSECT



Returns Common rows between two Tables

The screenshot shows the Object Explorer on the left with the SalesDB database selected. The main pane displays a T-SQL query:

```
-- Find the Employees, who are also customers.  
SELECT FirstName, LastName  
FROM Sales.Employees  
INTERSECT  
SELECT FirstName, LastName  
FROM Sales.Customers
```

The results pane shows the output:

	FirstName	LastName
1	Kevin	Brown
2	Mary	NULL

Order not matters here

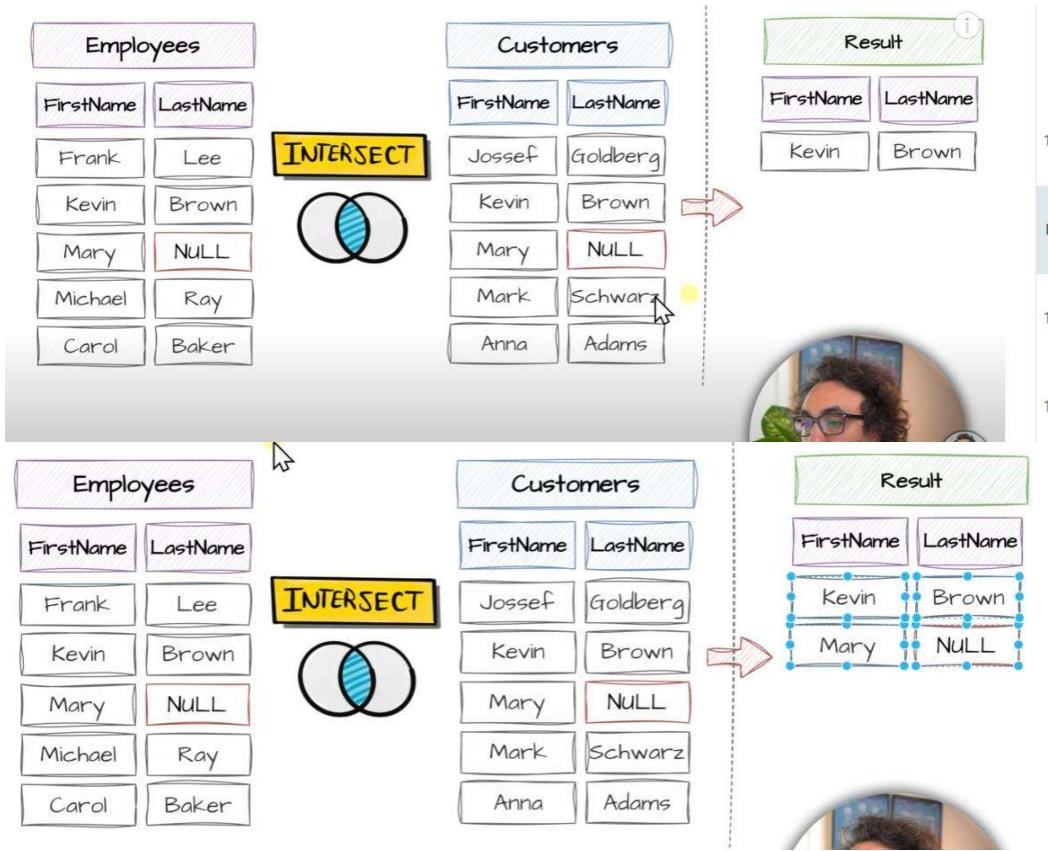
The screenshot shows the Object Explorer on the left with the SalesDB database selected. The main pane displays a T-SQL query:

```
-- Find the Employees, who are also customers.  
SELECT FirstName, LastName  
FROM Sales.Customers  
INTERSECT  
SELECT FirstName, LastName  
FROM Sales.Employees
```

The results pane shows the output:

	FirstName	LastName
1	Kevin	Brown
2	Mary	NULL

Row by row check going on from main table to lookup table



How to used sql for report generation

We have four tables contains info about people now we need to generate report of those

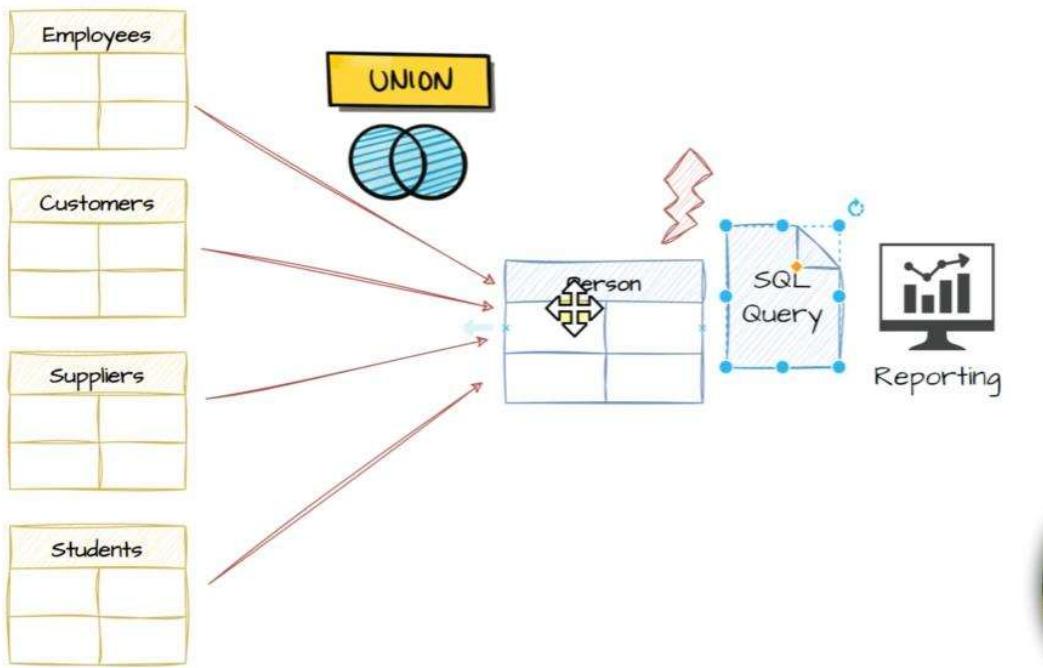
So we will write sql query for each and generate report

What if we forget logic what we write in one query although all queries are of same patterns logic because all tables contains persons info. So this things becomes very hectic and if we write separate logic for each query then there may be a inconsistency in report

Generated Report would not be a clear.

Now we will used set operators inorder to combine all tables into one table

Example for data Analyst-



One query so report would be consistent we can change logic at any time

Orders 2025

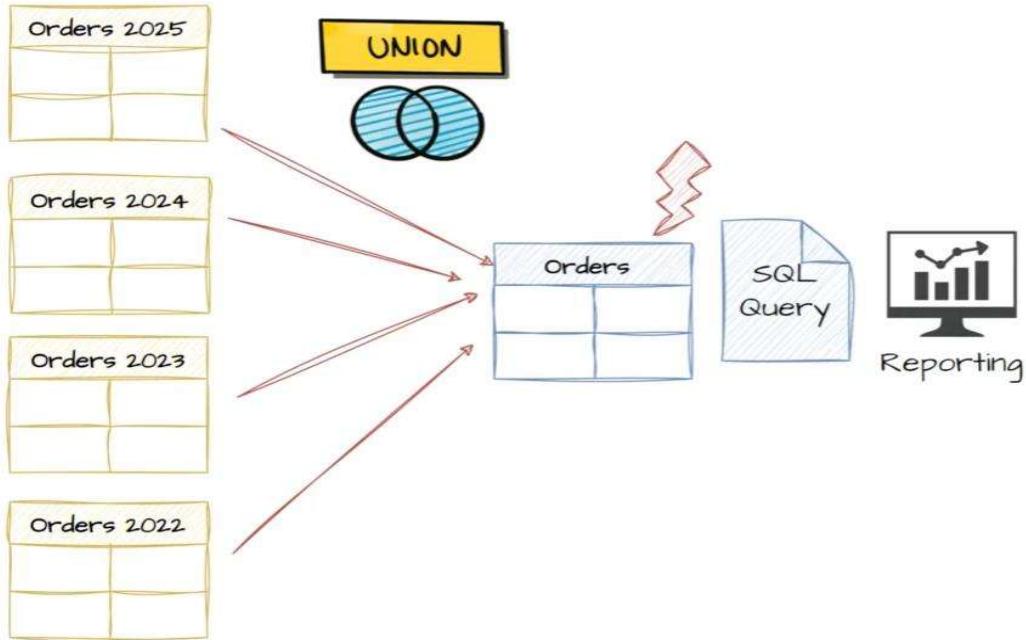
Orders 2024

Orders 2023

Orders 2022

Database developers divide the data into multiple tables to optimize performance and archive old data





SalesDB

```
-- Orders data are stored in separate tables (Orders)
-- Combine all orders data into one report without
SELECT *
FROM Sales.Orders
UNION
SELECT *
FROM Sales.OrdersArchive
```

OrderID	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate
10	5	104	2	5	2024-05-01
11	5	104	2	5	2025-02-01
12	6	101	3	5	2024-05-05
13	6	104	3	5	2024-05-05
14	6	104	3	5	2024-05-05
15	6	104	3	5	2025-02-05
16	7	102	1	1	2025-02-15
17	7	102	3	5	2024-06-15
18	8	101	4	3	2025-02-18
19	9	101	2	3	2025-03-10
20	10	102	5	5	2025-03-15

Query executed successfully.

BEST PRACTICES

Never use an asterisk (*) to combine tables; list needed columns instead

```
-- Orders data are stored in separate tables (Orders and OrdersArchive).
-- Combine all orders data into one report without duplicates.

SELECT *
FROM Sales.Orders
UNION
SELECT *
FROM Sales.OrdersArchive
```

ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddress
2	5	2024-05-01	2024-05-05	Delivered	345 Oak St	678	NULL
14	6	104	2024-05-05	2024-05-10	Delivered	543 Belmont Rd.	376
15	6	104	2025-02-05	2025-02-10	Delivered	1792 Belmont Rd.	NULL
16	7	102	2025-02-15	2025-02-27	Delivered	136 Balboa Court	NULL
17	7	102	2024-06-15	2024-06-20	Shipped	111 Main St	222
18	8	101	2025-02-18	2025-02-27	Shipped	2947 Vine Lane	4311
19	9	101	2025-03-10	2025-03-15	Shipped	3768 Door Way	NULL
20	10	102	2025-03-15	2025-03-20	Shipped	NULL	NULL

```
-- Orders data are stored in separate tables (Orders and OrdersArchive).
-- Combine all orders data into one report without duplicates.

SELECT
    [OrderID],
    [ProductID],
    [CustomerID],
    [SalesPersonID],
    [OrderDate],
    [ShipDate],
    [OrderStatus],
    [ShipAddress],
    [BillAddress],
    [Quantity],
    [Sales],
    [CreationTime]
FROM Sales.Orders
UNION
SELECT
    [OrderID],
    [CustomerID],
    [SalesPersonID],
    [OrderDate],
    [ShipDate],
    [OrderStatus],
    [ShipAddress]
```

OrderID	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddress
1	101	2	3	2024-04-01	2024-04-05	Shipped	123 Main St	456 Blv
2	101	2	3	2025-01-01	2025-01-05	Delivered	9833 Mt. Dias Blv.	1226
3	2	102	3	2024-04-05	2024-04-10	Shipped	456 Elm St	789 Blv
4	2	102	3	2025-01-05	2025-01-10	Shipped	250 Race Court	NULL
5	3	101	1	2024-04-10	2024-04-25	Shipped	789 Maple St	29 M
6	3	101	1	2025-01-10	2025-01-25	Delivered	8157 W. Book	81571

Incorrect data by changing orders of columns so column order matters
and we have to write output column name manually it's a good practice

SalesDB

Object Explorer

```
-- Orders data are stored in separate tables (Orders and OrdersArchive).
-- Combine all orders data into one report without duplicates.

SELECT
    [ProductID]
    ,[OrderID]
    ,[CustomerID]
    ,[SalesPersonID]
    ,[OrderDate]
    ,[ShipDate]
    ,[OrderStatus]
    ,[ShipAddress]
    ,[BillAddress]
    ,[Quantity]
    ,[Sales]
    ,[CreationTime]
    FROM Sales.Orders
UNION
SELECT
    [OrderID]
```

Results

	ProductID	OrderID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddr
10	7	102	3	5	2024-06-15	2024-06-20	Shipped	111 Main St	222 Bill
11	101	1	2	3	2025-01-01	2025-01-05	Delivered	9833 Mt. Dias ...	1226 Si
12	101	3	1	5	2025-01-10	2025-01-25	Delivered	8157 W. Book	8157 W
13	101	8	4	3	2025-02-18	2025-02-27	Shipped	2947 Vine Lane	4311 C
14	101	9	2	3	2025-03-10	2025-03-15	Shipped	3768 Door Way	
15	102	2	3	3	2025-01-05	2025-01-10	Shipped	250 Race Court	NULL
16	102	7	1	1	2025-02-15	2025-02-27	Delivered	136 Balboa Co...	
17	102	10	3	5	2025-03-15	2025-03-20	Shipped	NULL	NULL
18	104	5	2	5	2025-02-01	2025-02-05	Delivered	NULL	NULL
19	104	6	3	5	2025-02-05	2025-02-10	Delivered	1792 Belmont ...	NULL
20	105	4	1	3	2025-01-20	2025-01-25	Shipped	5724 Victory L...	



SalesDB

Object Explorer

```
-- Orders data are stored in separate tables (Orders and OrdersArchive).
-- Combine all orders data into one report without duplicates.

SELECT
    [ProductID]
    ,[OrderID]
    ,[CustomerID]
    ,[SalesPersonID]
    ,[OrderDate]
    ,[ShipDate]
    ,[OrderStatus]
    ,[ShipAddress]
    ,[BillAddress]
    ,[Quantity]
    ,[Sales]
    ,[CreationTime]
    FROM Sales.Orders
UNION
SELECT
    [OrderID]
    ,[ProductID]
    ,[CustomerID]
    ,[SalesPersonID]
    ,[OrderDate]
    ,[ShipDate]
    ,[OrderStatus]
    ,[ShipAddress]
```

Results

	ProductID	OrderID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddr
1	1	101	2	3	2024-04-01	2024-04-05	Shipped	123 Main St	456 Bill
2	2	102	3	3	2024-04-10	2024-04-10	Shipped	456 Elm St	789 Bill
3	3	101	1	4	2024-04-10	2024-04-25	Shipped	789 Maple St	789 Ma
4	4	105	1	3	2024-04-20	2024-04-25	Delivered	987 Victory Lane	
5	4	105	1	3	2024-04-20	2024-04-25	Shipped	987 Victory Lane	
6	5	104	2	5	2024-05-01	2024-05-05	Shipped	345 Oak St	678 Pin



Another way

SalesDB

Object Explorer

```
-- Orders data are stored in separate tables (Orders and OrdersArchive).
-- Combine all orders data into one report without duplicates.

SELECT
    [OrderID]
    ,[ProductID]
    ,[CustomerID]
    ,[SalesPersonID]
    ,[OrderDate]
    ,[ShipDate]
    ,[OrderStatus]
    ,[ShipAddress]
    ,[BillAddress]
    ,[Quantity]
    ,[Sales]
    ,[CreationTime]
    FROM Sales.Orders
UNION
SELECT
    [OrderID]
```

Results

	OrderID	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddr
	101	2	3	4	2024-04-01	2024-04-25	Shipped	123 Main St	456 I
	101	2	2	5	2025-01-01	2025-01-25	Delivered	9833 Mt. Dias Blv.	1226
	101	2	2	5	2025-01-01	2025-01-25	Delivered	456 Elm St	789 I
	101	2	2	5	2025-01-01	2025-01-25	Delivered	789 Maple St	789 I
	101	2	2	5	2025-01-01	2025-01-25	Delivered	987 Victory Lane	
	101	2	2	5	2025-01-01	2025-01-25	Delivered	5724 Victory Lane	
	101	2	2	5	2025-01-01	2025-01-25	Delivered	345 Oak St	678 I
	101	2	2	5	2025-01-01	2025-01-25	Delivered	8157 W. Book	8157
	101	2	2	5	2025-01-01	2025-01-25	Delivered	513 Belmont Rd	513 Belmont Rd



SOURCE FLAG
Include additional column to indicate the source of each row

	5	3	101	1	4	2024-04-10	2024-04-25	Shipped	123 Main St	456 I
	6	3	101	1	5	2025-01-10	2025-01-25	Delivered	9833 Mt. Dias Blv.	1226
	7	4	105	1	3	2024-04-20	2024-04-25	Delivered	456 Elm St	789 I
	8	4	105	1	3	2024-04-20	2024-04-25	Shipped	789 Maple St	789 I
	9	4	105	1	3	2025-01-20	2025-01-25	Shipped	987 Victory Lane	
	10	5	104	2	5	2024-05-01	2024-05-05	Shipped	5724 Victory Lane	
	11	5	104	2	5	2025-02-01	2025-02-05	Delivered	345 Oak St	678 I
	12	6	101	3	5	2025-01-05	2025-01-10	Delivered	8157 W. Book	8157
									513 Belmont Rd	513 Belmont Rd

Query executed successfully.



Object Explorer

```

SELECT
    [CreationTime]
  , [OrderID]
  , [CustomerID]
  , [SalesPersonID]
  , [OrderDate]
  , [ShipDate]
  , [OrderStatus]
  , [ShipAddress]
  , [BillAddress]
  , [Quantity]
  , [Sales]
  , [CreationTime]
FROM Sales.OrdersArchive
ORDER BY OrderID
  
```

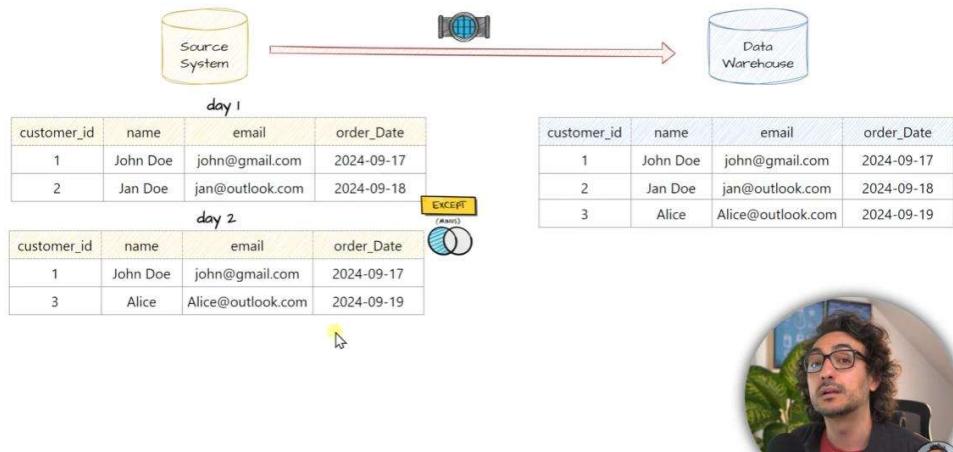
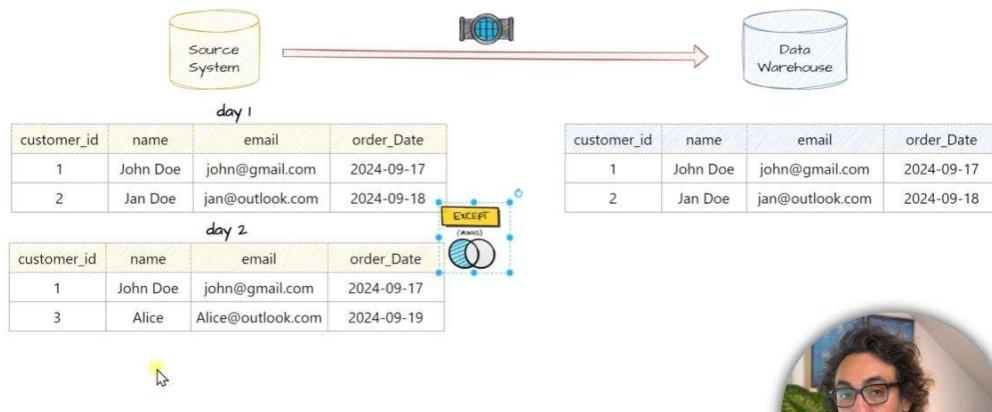
SQLQuery15.sql - 8QBU\Youtube (77) SQLQuery14.sql - 8QBU\Youtube (69) SQLQuery5.sql - D:\8QBU\Youtube (63)

	SourceTable	OrderID	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress
1	Orders	1	101	2	3	2025-01-01	2025-01-05	Delivered	9833 Mt. I...
2	OrdersArchive	1	101	2	3	2024-04-01	2024-04-05	Shipped	123 Main...
3	Orders	2	102	3	3	2025-01-05	2025-01-10	Shipped	250 Race...
4	OrdersArchive	2	102	3	3	2024-04-05	2024-04-10	Shipped	456 Elm S...
5	Orders	3	101	1	5	2025-01-10	2025-01-25	Delivered	8157 W. E...
6	OrdersArchive	3	101	1	4	2024-04-10	2024-04-25	Shipped	789 Maple...
7	Orders	4	105	1	3	2025-01-20	2025-01-25	Shipped	5724 Vick...
8	OrdersArchive	4	105	1	3	2024-04-20	2024-04-25	Delivered	987 Victor...
9	OrdersArchive	4	105	1	3	2024-04-20	2024-04-25	Shipped	987 Victor...
10	Orders	5	104	2	5	2025-02-01	2025-02-05	Delivered	NULL
11	OrdersArchive	5	104	2	5	2024-05-01	2024-05-05	Shipped	1729 Berlin...
12	Orders	6	104	3	5	2025-02-05	2025-02-10	Delivered	1729 Berlin...

Example for data engineer -

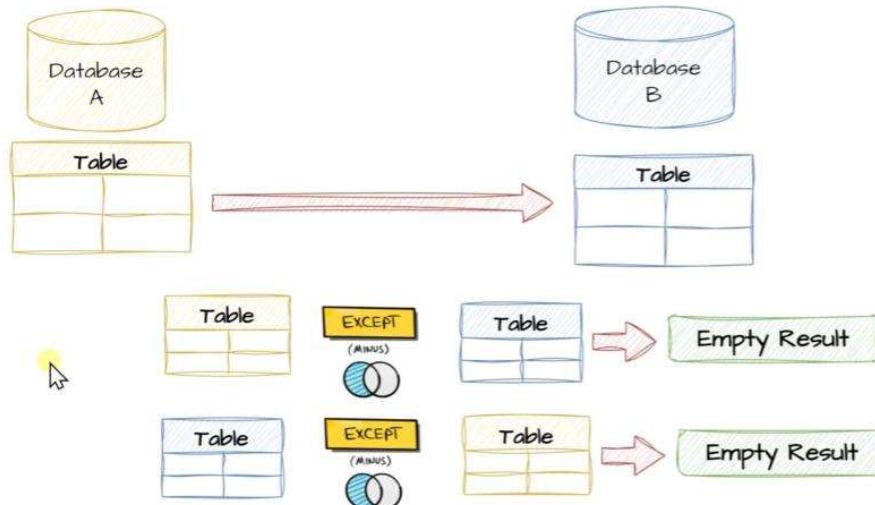
DELTA DETECTION

Identifying the differences or changes (delta)
between two batches of data.



DATA COMPLETENESS CHECK

EXCEPT operator can be used to compare tables to detect discrepancies between databases.



Improve data Migration quality