

zenius



Kampus  
Merdeka  
INDONESIA JAYA

# Database : SQL Query III

Hari, Tanggal

Data Analytics

Program Zenius Studi Independen  
Bersertifikat Bersama Kampus Merdeka



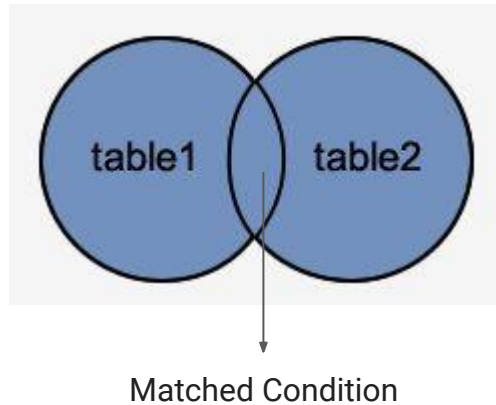
1. Join Operations
2. Union Operations
3. Subquery
4. CTE (Common Table Expressions)

# Join Operations in SQL

# JOIN

**JOIN in SQL is used to combine data from many tables based on a matched condition between them.**

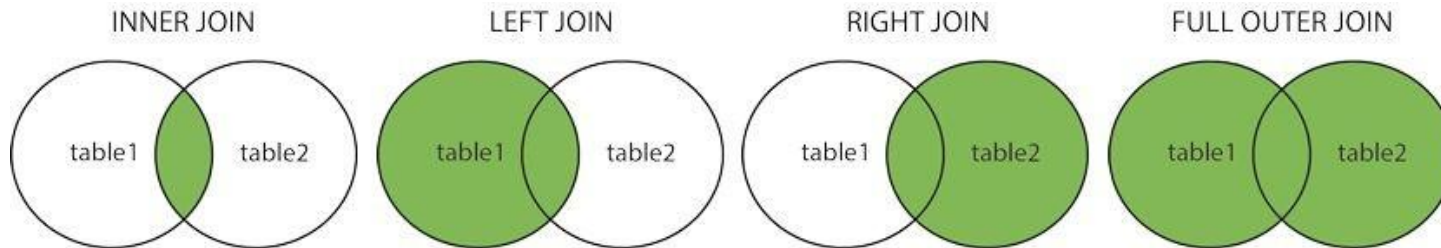
**The data combined using JOIN statement results into new columns.**



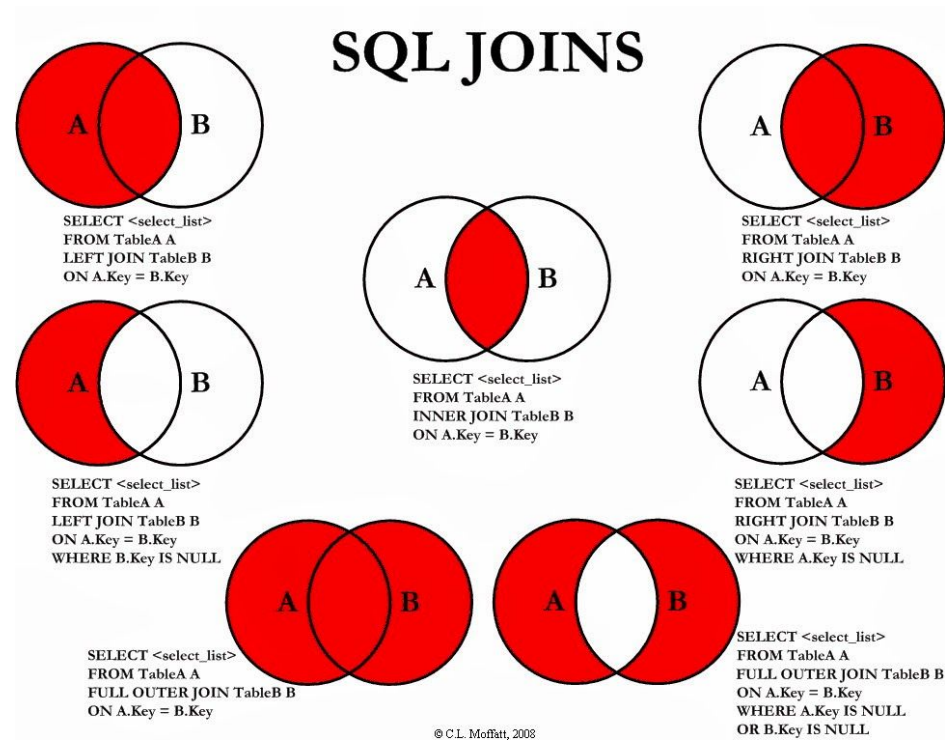
# Types of JOIN

Here are the different types of the JOINS in SQL:

- **(INNER) JOIN:** Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN:** Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN:** Returns all records from left and right table

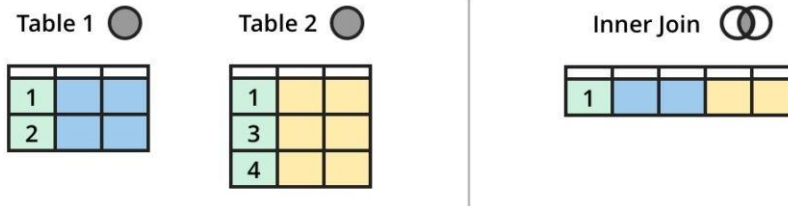


# Types of JOIN



# INNER JOIN/JJOIN

- Returns records that have matching values(in joining key) in both tables



User Table – Table 1

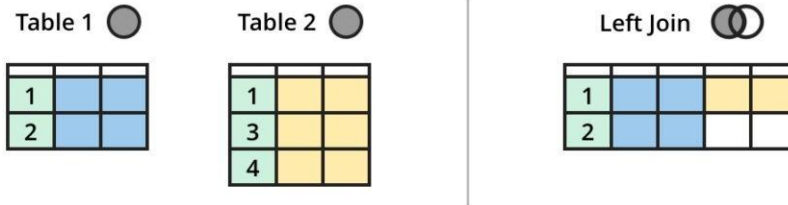
ID (Primary Key)	Name	Address
1	Sally Select	123 Join Dr
2	Frank From	25 Where St

Event Table – Table 2

User ID (Foreign Key)	ID (Primary Key)	Action
1	A	LOGIN
3	B	VIEW PAGE
4	C	LOGIN

# LEFT JOIN

- Returns all records from the left table, and the matched records from the right table



User Table – Table 1

ID (Primary Key)	Name	Address
1	Sally Select	123 Join Dr
2	Frank From	25 Where St

Event Table – Table 2

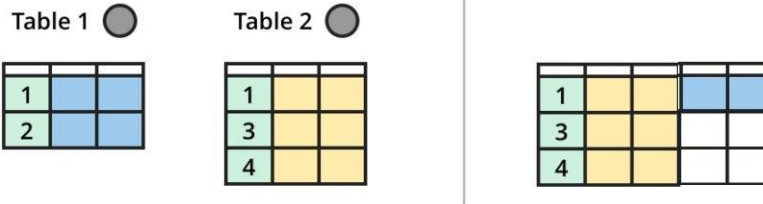
User ID (Foreign Key)	ID (Primary Key)	Action
1	A	LOGIN
3	B	VIEW PAGE
4	C	LOGIN



# RIGHT JOIN

- Returns all records from the right table, and the matched records from the left table

Right join



User Table – Table 1

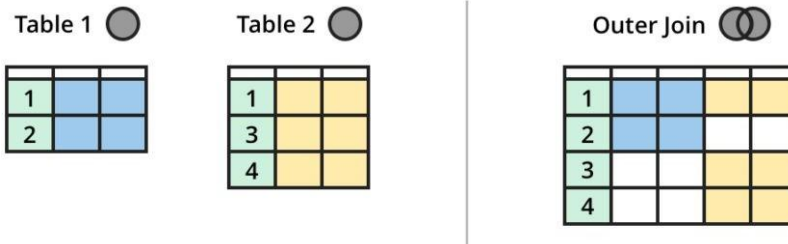
ID (Primary Key)	Name	Address
1	Sally Select	123 Join Dr
2	Frank From	25 Where St

Event Table – Table 2

User ID (Foreign Key)	ID (Primary Key)	Action
1	A	LOGIN
3	B	VIEW PAGE
4	C	LOGIN

# FULL/OUTER JOIN

- Returns all records from left and right table



User Table – Table 1

ID (Primary Key)	Name	Address
1	Sally Select	123 Join Dr
2	Frank From	25 Where St

Event Table – Table 2

User ID (Foreign Key)	ID (Primary Key)	Action
1	A	LOGIN
3	B	VIEW PAGE
4	C	LOGIN

# JOIN Syntax

- Returns all records from left and right table

```
SELECT table_alias1.column_name(s), table_alias2.column_name(s)
```

```
FROM table_name table_alias1
```

Can be different  
type of join

```
JOIN table_name table_alias2
```

```
ON table_alias1.matching_column = table_alias2.matching_column
```

```
[WHERE CLAUSE] – optional      [ORDER BY CLAUSE] – optional
```

# JOIN Example

SQL 1

```
1 SELECT customers.CustomerId, FirstName, Total
2 FROM customers
3 JOIN invoices
4 ON customers.CustomerId = invoices.CustomerId
```

	CustomerId	FirstName	Total
1	1	Luís	3.98
2	1	Luís	3.96
3	1	Luís	5.94
4	1	Luís	0.99
5	1	Luís	1.98

# Mini Practice

[https://www.w3schools.com/sql/sql\\_join.asp](https://www.w3schools.com/sql/sql_join.asp)

## Questions :

Buatlah query yang dapat membuat output data berupa :

1. Semua **CustomerName** dari setiap order yang ada
2. Berapa **jumlah customer** yang order
3. Berapa **jumlah unique customer** yang order
4. Top 3 **CustomerName** dengan order terbanyak

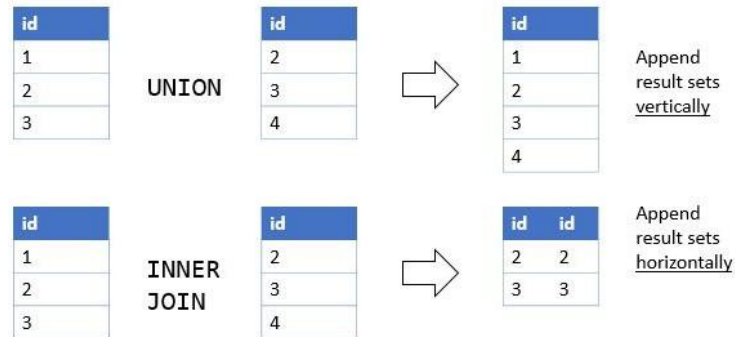
```
SELECT c.CustomerName,  
COUNT(orders.OrderID) AS jumlah_order  
FROM customers c  
JOIN orders ON c.CustomerID =  
orders.CustomerID  
GROUP BY c.CustomerName  
ORDER BY jumlah_order DESC  
LIMIT 3;
```

# Union Operations in SQL

# UNION

**UNION** in SQL is used to combine two or more tables, but will result in additional rows in your output table.

**A Union will stack tables on top of each other resulting in new rows.**





# UNION

Table 1 ●

1		
2		

Table 2 ●

1		
3		
4		

Union ●+●

1		
2		
1		
3		
4		

User Table – Table 1

ID (Primary Key)	Name	Address
1	Sally Select	123 Join Dr
2	Frank From	25 Where St

Event Table – Table 2

User ID (Foreign Key)	ID (Primary Key)	Action
1	A	LOGIN
3	B	VIEW PAGE
4	C	LOGIN

# UNION



The screenshot shows a SQL query editor window titled "SQL 1". The query is as follows:

```
1 SELECT * FROM artists
2 UNION
3 SELECT * FROM genres
```

Below the query editor, the results of the query are displayed in a table with two columns: "ArtistId" and "Name". The results are as follows:

	ArtistId	Name
1	1	AC/DC
2	1	Rock
3	2	Accept
4	2	Jazz
5	3	Aerosmith

# UNION and UNION ALL

If you had the *suppliers* table populated with the following records:

```
SELECT supplier_id  
FROM suppliers  
UNION  
SELECT supplier_id  
FROM orders  
ORDER BY supplier_id;
```

supplier_id	supplier_name
1000	Microsoft
2000	Oracle
3000	Apple
4000	Samsung

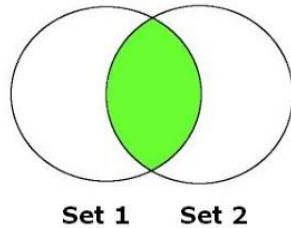
And the *orders* table populated with the following records:

order_id	order_date	supplier_id
1	2015-08-01	2000
2	2015-08-01	6000
3	2015-08-02	7000
4	2015-08-03	8000

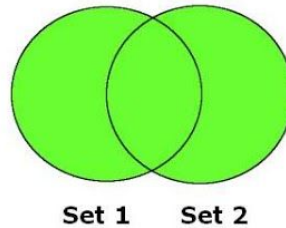
supplier_id
1000
2000
3000
4000
6000
7000
8000

If you **do not wish to remove duplicates**, try using the **UNION ALL** operator

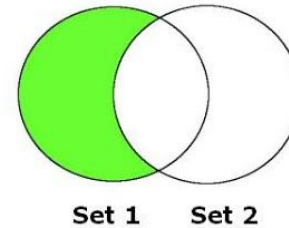
# Types of UNION



Intersect



Union  
Union All



Minus  
Except

# Subquery in SQL

# Subquery

PRODUCT	
Id	
ProductName	
SupplierId	
UnitPrice	
Package	
IsDiscontinued	

ORDERITEM	
Id	
OrderId	
ProductId	
UnitPrice	
Quantity	

**Problem:** List products with order quantities greater than 100.

```
1. SELECT ProductName
2. FROM Product
3. WHERE Id IN (SELECT ProductId
4.              FROM OrderItem
5.              WHERE Quantity > 100)
```

# CTE in SQL

# CTE

PRODUCT	
Id	🔑
ProductName	
SupplierId	
UnitPrice	
Package	
IsDiscontinued	

ORDERITEM	
Id	🔑
OrderId	
ProductId	
UnitPrice	
Quantity	

**Problem:** List products with order quantities greater than 100.

```
WITH filter AS
(SELECT ProductId
FROM OrderItem
WHERE Quantity > 100)

SELECT ProductName
FROM Product
WHERE Id in (SELECT * FROM filter)
```



# Mini Practice

## Questions :

1. Which customers have not rented any movies so far.
2. How many movies offered for rent in store\_id 1 and not offered in store\_id 2
3. Display the movies (film\_id) offered for rent in any of the two stores 1 and 2 **(use UNION)**
4. What is the name of the customer who made the highest total payments
5. Display the customer\_id's for those customers that rented a movie DVD more than once **(use CTE)**

**Thank you!**  
Any Questions?

zenius



Kampus  
Merdeka  
INDONESIA JAYA