



**Join Table**

## Daftar Isi

<b>A. Join Table</b>	<b>3</b>
<b>B. Mengapa Join Itu Penting?</b>	<b>3</b>
<b>C. Jenis - Jenis Join Table</b>	<b>3</b>
1. Inner Join	3
2. Left [Outer] Join	4
3. Left [Outer] Join without Intersection	4
4. Right [Outer] Join	4
5. Right [Outer] Join without Intersection	5
6. Full [Outer] Join	5
7. Full [Outer] Join without Intersection	6
<b>Case Study</b>	<b>7</b>
<b>References</b>	<b>9</b>

## A. Join Table

**Join table** adalah cara untuk menghubungkan data yang diambil dari tabel-tabel melalui sebuah kolom yang menghubungkan mereka. Misal, pembaca mungkin ingin menghubungkan tabel alamat dengan tabel nomor telepon berdasarkan nama seseorang (contoh: "Berikan saya alamat dan nomor telepon seseorang yang bernama John Smith.").

## B. Mengapa Join Itu Penting?

*Join* memperbolehkan kita untuk mengambil data dari beberapa tabel melalui satu *query*. Hanya menggunakan sebuah tabel artinya kita hanya dapat menyimpan/memperoleh data yang terbatas atau justru menyimpan/memperoleh data yang terlalu banyak sehingga tabelnya menjadi kurang baik. *Join* menghubungkan satu tabel dengan tabel yang lain (inilah yang dimaksud dengan relational dari istilah *relational database*).

## C. Jenis - Jenis Join Table

Berikut ini adalah empat tipe join di SQL (bersama tiga variannya). Sebagai pelengkap dari penjelasan yang ada, kami telah menyediakan contoh kode SQL.

### 1. *Inner Join*

*Inner join* mungkin tipe join yang paling banyak dipakai. *Inner join* mengembalikan baris-baris dari dua tabel atau lebih yang memenuhi syarat.

**SELECT columns**

**FROM TableA**

**INNER JOIN TableB**

**ON A.columnName = B.columnName;**

## 2. *Left [Outer] Join*

*Left outer join* (sering disingkat *left join*) akan mengembalikan seluruh baris dari tabel sebelah kiri yang dikenai kondisi *ON* dan hanya baris dari tabel sebelah kanan yang memenuhi kondisi *join*.

**SELECT columns**

**FROM TableA**

**LEFT OUTER JOIN TableB**

**ON A.columnName = B.columnName**

## 3. *Left [Outer] Join without Intersection*

*Join* ini merupakan variasi dari left outer join. Pada *join* ini kita hanya akan mengambil data dari tabel sebelah kiri yang dikenai kondisi *ON* yang juga memenuhi kondisi *join* tanpa data dari tabel sebelah kanan yang memenuhi kondisi *join*.

**SELECT columns**

**FROM TableA**

**LEFT OUTER JOIN TableB**

**ON A.columnName = B.columnName**

**WHERE B.columnName IS NULL**

## 4. *Right [Outer] Join*

*Right outer join* (sering disingkat *right join*) akan mengembalikan semua baris dari tabel sebelah kanan yang dikenai kondisi *ON* dengan data dari tabel sebelah kiri yang memenuhi kondisi *join*. Teknik ini merupakan kebalikan dari *left outer join*.

**SELECT columns**

```
FROM TableA  
RIGHT OUTER JOIN TableB  
ON A.columnName = B.columnName
```

### 5. *Right [Outer] Join without Intersection*

Teknik ini merupakan variasi dari *right outer join*. Pada *join* ini kita hanya akan mengambil data dari tabel sebelah kanan yang dikenai kondisi *ON* yang juga memenuhi kondisi join tanpa data dari tabel sebelah kanan yang memenuhi kondisi *join*.

```
SELECT columns  
FROM TableA  
RIGHT OUTER JOIN TableB  
ON A.columnName = B.columnName  
WHERE A.columnName IS NULL
```

### 6. *Full [Outer] Join*

*Full outer join* (sering disingkat full join) akan mengembalikan seluruh baris dari kedua tabel yang dikenai *ON* termasuk data-data yang bernilai *NULL*.

```
SELECT columns  
FROM TableA  
FULL JOIN TableB  
ON A.columnName = B.columnName
```

## 7. Full [Outer] Join without Intersection

Variasi lain dari *full outer join* yang akan mengembalikan seluruh data dari kedua tabel yang dikenai *ON* tanpa data yang memiliki nilai *NULL*.

**SELECT columns**

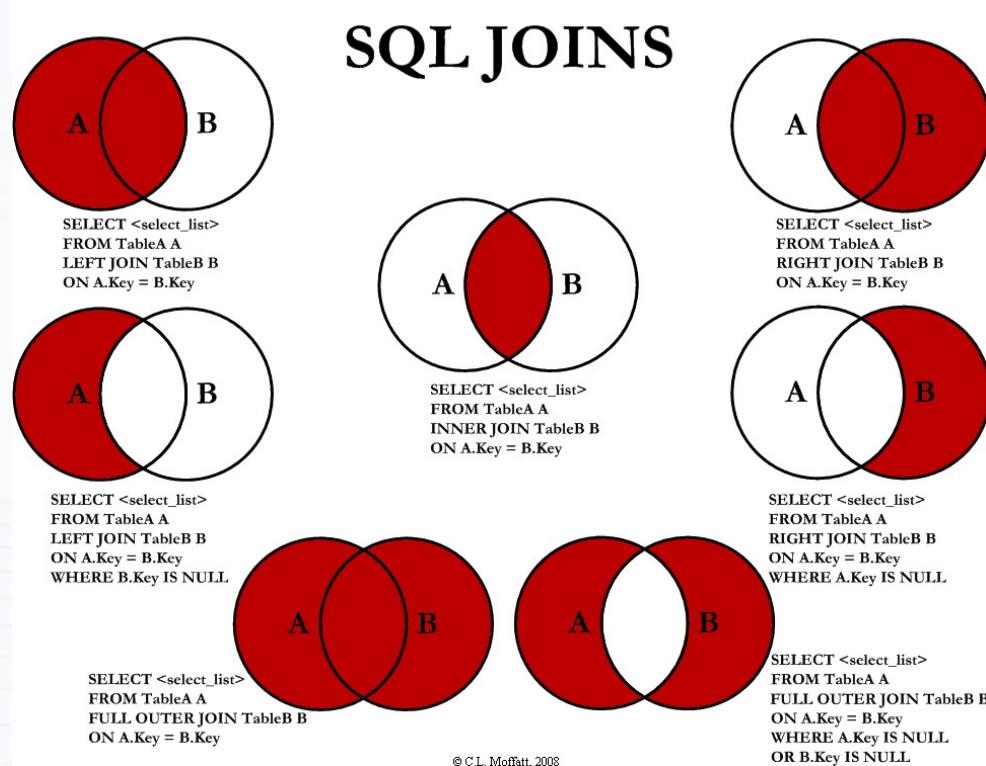
**FROM TableA**

**FULL JOIN TableB**

**ON A.columnName = B.columnName**

**WHERE A.columnName IS NULL**

**OR B.columnName IS NULL**



## Case Study

Anda diminta untuk membuat laporan yang menampilkan total jumlah nasabah dan transaksi dan rata-rata jumlah transaksi per jenis tabungan dan jenis transaksi. Hanya nasabah yang memiliki rata-rata jumlah transaksi di atas Rp. 500,000 yang akan dimasukkan dalam laporan.

### Struktur Tabel:

#### Tabel Nasabah

Kolom: ID\_Transaksi (Primary Key), Nomer\_Rekening (Foreign Key)

Nomer Rekening	ID_Nasabah	Nama_Nasabah	Alamat	Jenis Tabungan
1001	1	John Doe	Jl. Merdeka No. 123	Tabungan Biasa
1002	2	Jane Smith	Jl. Pahlawan No. 456	Tabungan Bisnis
1003	3	Ahmad Abdullah	Jl. Damai No. 789	Tabungan Investasi

#### Tabel ATM\_Transaksi

Kolom: ID\_Transaksi (Primary Key), ID\_Nasabah (Foreign Key)

ID_Transaksi	ID_Nasabah	Jenis_Transaksi	Jumlah_Transaksi	Tanggal_Transaksi	Nomer Rekening
101	1	Penarikan	500000	1/1/2023 8:15	1001
102	2	Transfer	200000	1/2/2023 10:30	1002
103	3	Setoran	1000000	1/2/2023 15:45	1003

### Hasil Query:

```

SELECT
    N.Jenis_Tabungan,
    AT.Jenis_Transaksi,
    COUNT(Distinct N.ID_Nasabah) AS Jumlah_Nasabah,
    COUNT(AT.ID_Transaksi) AS Frekuensi_Transaksi,
    AVG(AT.Jumlah_Transaksi * 1.0) AS Rata_Rata_Jumlah_Transaksi
FROM
    
```

Nasabah N

JOIN

ATM\_Transaksi AT ON N.Nomer\_Rekening = AT.Nomer\_Rekening

GROUP BY

N.Jenis\_Tabungan, AT.Jenis\_Transaksi,

HAVING

AVG(AT.Jumlah\_Transaksi \* 1.0) > 500000;

Contoh hasil query:

Jenis_Tabungan	Jenis_Transaksi	Jumlah_Nasabah	Frekuensi_Transaksi	Rata_Rata_
Tabungan Biasa	Tarik Tunai	2	2	1250000.0
Tabungan Biasa	Transfer	2	2	850000.0
Tabungan Premium	Tarik Tunai	1	1	1500000.0
Tabungan Premium	Transfer	2	2	850000.0

## References

Santoso, B. A. (2017). *Tujuh Teknik Join di SQL*. Codepolitan. Retrieved October 26, 2023, from <https://www.codepolitan.com/tujuh-teknik-join-di-sql-596c537f0deb3/>