

MODUL PRAKTIKUM BASIS DATA

Pertemuan 10 : Fungsi JOIN dan UNION Basis Data

A. CPMK

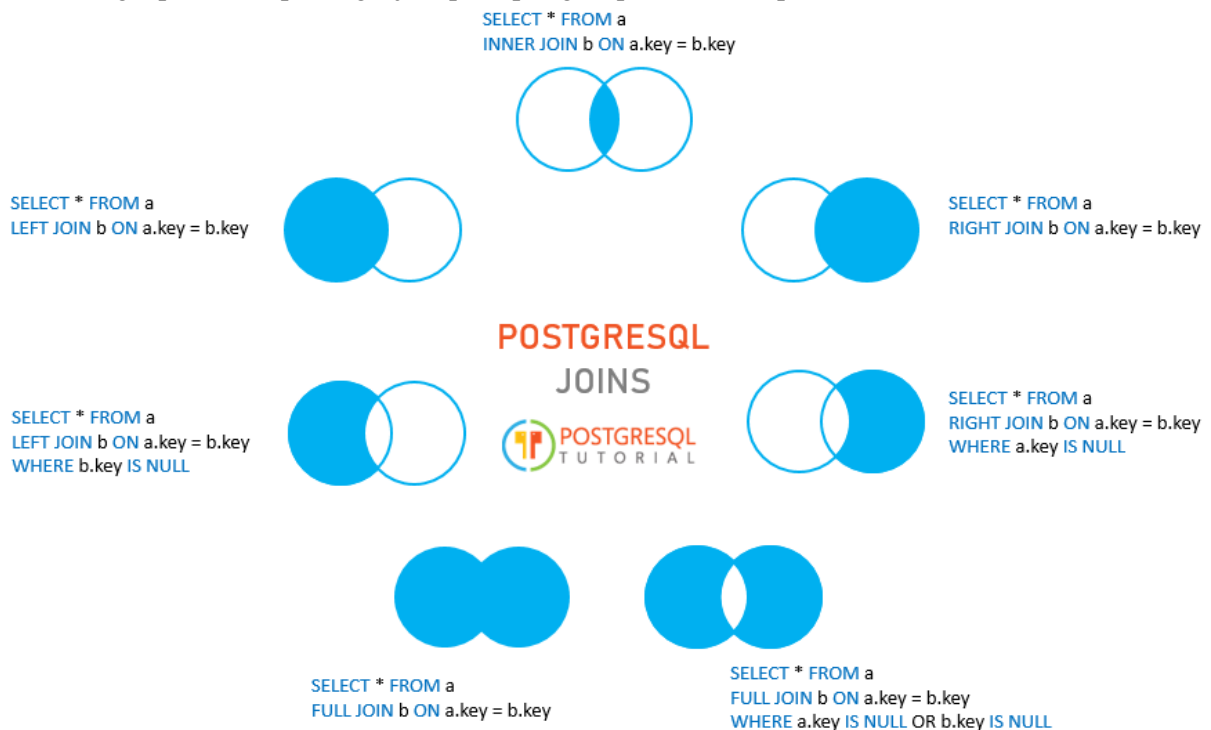
- Mampu mendeskripsikan dan mengimplementasikan teknik perancangan basis data.
- Mampu menjelaskan dan mempraktekkan kelompok pernyataan sql & teknik manipulasi data, modifikasi tabel.
- Mampu menjelaskan dan membuat perintah-perintah dasar sql & perintah query

B. Tujuan Praktikum

- Mampu menggunakan fungsi join pada basis data untuk mempermudah menampilkan data yang diinginkan pada basis data.

C. Dasar Teori

Dalam basis data, fungsi join digunakan untuk menggabungkan baris-baris data dari dua atau lebih tabel berdasarkan kolom yang memiliki nilai yang sama. Fungsi join memungkinkan Anda untuk mengambil data dari beberapa tabel secara bersamaan dan menghasilkan hasil gabungan yang lebih lengkap. Beberapa fungsi join pada postgresql bisa dilihat pada Gambar 1.



Gambar 1 Join pada PostgreSQL

Src(<https://neon.tech/postgresqtutorial/PostgreSQL-Joins.png>)

D. PERCOBAAN

Ada beberapa jenis join yang umum digunakan:

■ Tabel 1: customers

customer_id	name
1	Andi
2	Budi
3	Citra

■ Tabel 2: orders

order_id	customer_id	product
101	1	Laptop
102	1	Mouse
103	2	Keyboard

1. Inner Join

Inner join menggabungkan tabel berdasarkan attribute yang mempunyai kesamaan di kedua tabel.

```
SELECT kolom
FROM tabel1
INNER JOIN tabel2
ON kondisi;
```

Contoh inner join:

```
SELECT customers.name, orders.product
FROM customers
INNER JOIN orders
ON customers.customer_id = orders.customer_id;
```

name	product
Andi	Laptop
Andi	Mouse
Budi	Keyboard

2. Left Join (Left Outer Join)

LEFT JOIN memilih SEMUA records dari tabel “kiri” (Tabel1), dan record yang cocok/sama dari tabel “kanan” (Tabel2).

```
SELECT kolom  
FROM tabel1  
LEFT JOIN tabel2  
ON kondisi;
```

Contoh:

```
SELECT customers.name, orders.product  
FROM customers  
LEFT JOIN orders  
ON customers.customer_id = orders.customer_id;
```

name	product
Andi	Laptop
Andi	Mouse
Budi	Keyboard
Citra	NULL

Dengan menggunakan LEFT JOIN kita akan mendapatkan semua record dari Tabel1, bahkan yang tidak ada kecocokannya dalam Tabel2.

3. Right Join

Right JOIN memilih SEMUA records dari tabel “kanan” (Tabel2), dan record yang cocok/sama dari tabel “kiri” (Tabel1).

```
SELECT kolom  
FROM tabel1  
RIGHT JOIN tabel2  
ON kondisi;
```

Contoh:

```
SELECT customers.name, orders.product  
FROM customers  
RIGHT JOIN orders  
ON customers.customer_id = orders.customer_id;
```

name	product
Andi	Laptop
Andi	Mouse
Budi	Keyboard

Dengan menggunakan Right JOIN kita akan mendapatkan semua record dari Tabel2, bahkan yang tidak ada kecocokannya dalam Tabel1.

Catatan: RIGHT JOIN tidak tersedia di semua database (misalnya SQLite).

4. Full Join

FULL JOIN memilih SEMUA data dari kedua tabel, meskipun tidak ada yang cocok. Untuk baris yang cocok, nilai dari kedua tabel tersedia, jika tidak ada yang cocok, kolom kosong akan mendapatkan nilai NULL.

```
SELECT kolom
FROM tabel1
FULL OUTER JOIN tabel2
ON kondisi;
```

Contoh:

```
SELECT customers.name, orders.product
FROM customers
FULL OUTER JOIN orders
ON customers.customer_id = orders.customer_id;
```

name	product
Andi	Laptop
Andi	Mouse
Budi	Keyboard
Citra	NULL

5. Cross Join

Menggabungkan semua record dari tabel pertama dengan semua record di tabel kedua.
Banyaknya record dari cross join = jumlah record tabel pertama X jumlah record tabel kedua

```
SELECT kolom
FROM tabel1
CROSS JOIN tabel2;
```

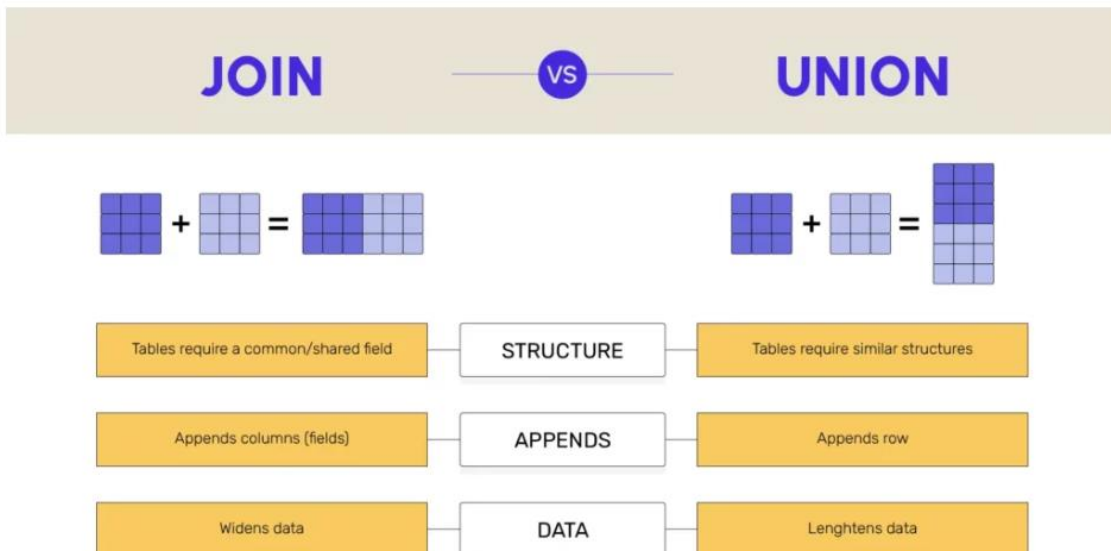
Contoh:

```
SELECT customers.name, orders.product
FROM customers
CROSS JOIN orders;
```

Hasil (3 customers × 3 orders = 9 baris)

name	product
Andi	Laptop
Andi	Mouse
Andi	Keyboard
Budi	Laptop
...	...

Ada join ada union dalam postgresQL, perbedaan union dan join dalam postgresql ditunjukkan pada Gambar 2.



Gambar 2 Perbedaan Join dan Union

Src(https://db0dce98.rocketcdn.me/en/files/2024/11/Schema_JOIN-vs.-UNION-1024x576.webp)

Union dalam sql harus memenuhi 3 syarat :

- Jumlah kolomnya harus sama
- Kolom-kolomnya harus memiliki tipe data yang sama
- Kolom-kolomnya harus dalam urutan yang sama

Berikut adalah contoh query union yang digunakan pada postgresQL.

■ Tabel: customers

name
Andi
Budi
Citra

■ Tabel: employees

name
Budi
Dewi

1. Union

```
SELECT kolom1, kolom2, ...  
FROM tabel1  
UNION  
SELECT kolom1, kolom2, ...  
FROM tabel2;
```

Contoh:

```
SELECT name FROM customers  
UNION  
SELECT name FROM employees;
```

name
Andi
Budi
Citra
Dewi

2. Union all

Dengan operator UNION, jika beberapa baris dalam dua kueri mengembalikan hasil yang sama persis, hanya satu baris yang akan dicantumkan, karena UNION hanya memilih nilai yang berbeda.

Gunakan UNION ALL untuk mengembalikan nilai duplikat.

```
SELECT kolom1, kolom2, ...  
FROM tabel1  
UNION ALL  
SELECT kolom1, kolom2, ...  
FROM tabel2;
```

Contoh:

```
SELECT name FROM customers  
UNION ALL  
SELECT name FROM employees
```

name
Andi
Budi
Citra
Budi
Dewi

E. Latihan Praktikum

STUDI KASUS: SISTEM PENJUALAN TOKO ONLINE

Sebuah toko online memiliki data sebagai berikut:

1 Tabel customers

Menyimpan data pelanggan.

customer_id	customer_name	city
1	Andi	Jakarta
2	Budi	Bandung
3	Citra	Surabaya
4	Dewi	Jakarta

2 Tabel orders

Menyimpan data pesanan pelanggan.

order_id	customer_id	order_date
101	1	2024-01-10
102	1	2024-01-15
103	2	2024-01-20
104	5	2024-01-25

3 Tabel payments

Menyimpan data pembayaran pesanan.

payment_id	order_id	amount
201	101	500000
202	102	150000
203	105	300000

1. Soal 1: INNER JOIN

Tampilkan:

- customer_name
- order_id
- order_date

Hanya untuk pelanggan yang benar-benar memiliki pesanan.

Gunakan INNER JOIN antara: customers dan orders

2. Soal 2: LEFT JOIN

Tampilkan:

- customer_name
- order_id
- order_date

Untuk SEMUA pelanggan, termasuk yang belum pernah order.

3. Soal 3 – RIGHT JOIN

Tampilkan:

- customer_name
- order_id
- order_date

Untuk SEMUA data order, termasuk order yang tidak memiliki customer valid.

4. Soal 4 – FULL JOIN

Tampilkan:

- customer_name
- order_id
- order_date

Gabungkan SEMUA data customer dan SEMUA data order, meskipun tidak saling cocok.

5. Soal 5 – LEFT JOIN + IS NULL

Tampilkan pelanggan yang belum pernah melakukan order.

Kolom yang ditampilkan:

- customer_id
- customer_name

Gunakan LEFT JOIN + WHERE ... IS NULL

6. Soal 6 - JOIN 3 Tabel (INNER JOIN)

Tampilkan:

- customer_name
- order_id
- amount

Hanya untuk:

- Order yang valid
- Customer yang valid
- Payment yang valid

Gunakan INNER JOIN pada:

- customers
- orders
- payments

7. Soal 7 - CROSS JOIN

Tampilkan semua kemungkinan kombinasi antara:

- customer_name
- city

Dengan daftar kota berikut (gunakan subquery / tabel dummy):

city
Jakarta
Bandung
Surabaya

Gunakan CROSS JOIN

8. Soal 8 - UNION

Gabungkan daftar nama orang dari:

- customers.customer_name
- Daftar admin berikut:

admin_name
Admin1
Budi

Admin2

⌋ Hasil tidak boleh duplikat
Gunakan UNION

9. Soal 9 - UNION ALL

Gabungkan kembali soal nomor 8, tetapi:

- SEMUA data harus ditampilkan
- Termasuk data yang sama

Gunakan UNION ALL