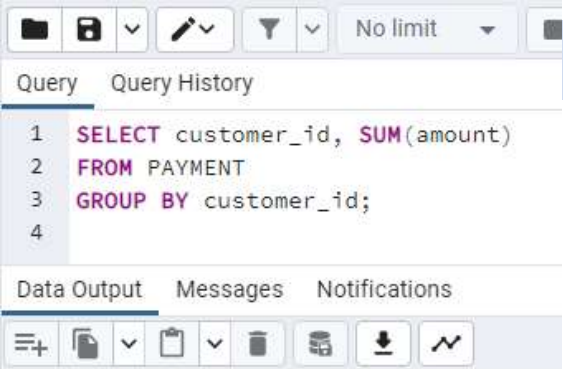


PostgreSQL - Grouping Data and Set Operation

- Tessy Badriyah, PhD.

Pengelompokan Data

- Pengelompokan data dengan GROUP BY
- Biasanya digunakan Bersama dengan fungsi agregasi
- Berikut ini menggunakan data dari database dvdrental
- Tampilkan kode customer (customer_id) beserta total jumlah pembayaran dari customer (diambil dari tabel PAYMENT)
- SELECT customer_id, SUM(amount)
- FROM PAYMENT
- GROUP BY customer_id;

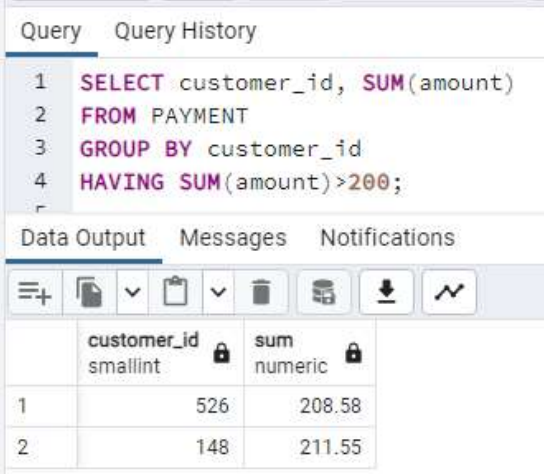


The screenshot shows a SQL query editor interface. At the top, there are icons for file operations and a 'No limit' dropdown. Below the icons, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query with line numbers 1 through 4. The query is:
1 SELECT customer_id, SUM(amount)
2 FROM PAYMENT
3 GROUP BY customer_id;
4
Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 8 rows of data. The table has two columns: 'customer_id' (smallint) and 'sum' (numeric). The data is as follows:

	customer_id smallint	sum numeric
1	184	80.80
2	87	137.72
3	477	106.79
4	273	130.72
5	550	151.69
6	51	123.70
7	394	77.80
8	272	65.87

Pengelompokan Data

- HAVING digunakan untuk menyaring data yang sudah dikelompokkan.
- Misal: tampilkan data customer_id dan jumlah total pembayarannya, hanya untuk yang total pembayarannya lebih dari 200.
- SELECT customer_id, SUM(amount)
- FROM PAYMENT
- GROUP BY customer_id
- HAVING SUM(amount)>200;



The screenshot shows a SQL query editor with a 'Query' tab selected. The query text is as follows:

```
1 SELECT customer_id, SUM(amount)
2 FROM PAYMENT
3 GROUP BY customer_id
4 HAVING SUM(amount)>200;
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with the results of the query. The table has two columns: 'customer_id' (smallint) and 'sum' (numeric). The results are as follows:

	customer_id smallint	sum numeric
1	526	208.58
2	148	211.55

Set operation : UNION

UNION menggabung dua tabel yang memiliki kolom/atribut yang sama.

Buat dua tabel: top Rated_film dan most Popular_film

Perintah pembuatan tabel:

```
CREATE TABLE top Rated_film(  
    title VARCHAR NOT NULL,  
    release_year SMALLINT  
);  
  
CREATE TABLE most Popular_film(  
    title VARCHAR NOT NULL,  
    release_year SMALLINT  
);
```

Perintah pengisian tabel:

```
INSERT INTO  
    top Rated_film(title,release_year)  
VALUES  
    ('The Shawshank Redemption',1994),  
    ('The Godfather',1972),  
    ('12 Angry Men',1957);  
  
INSERT INTO  
    most Popular_film(title,release_year)  
VALUES  
    ('An American Pickle',2020),  
    ('The Godfather',1972),  
    ('Greyhound',2020);
```

Set operation : UNION

Contoh penggunaan UNION

```
SELECT * FROM topRated_films
```

```
UNION
```

```
SELECT * FROM mostPopular_films;
```

Query		Query History	
1	SELECT	*	FROM topRated_films
2	UNION		
3	SELECT	*	FROM mostPopular_films;
4			
Data Output		Messages	
		Notifications	
	title	release_year	
	character varying	smallint	
1	An American Pickle	2020	
2	Greyhound	2020	
3	The Shawshank Redemption	1994	
4	The Godfather	1972	
5	12 Angry Men	1957	

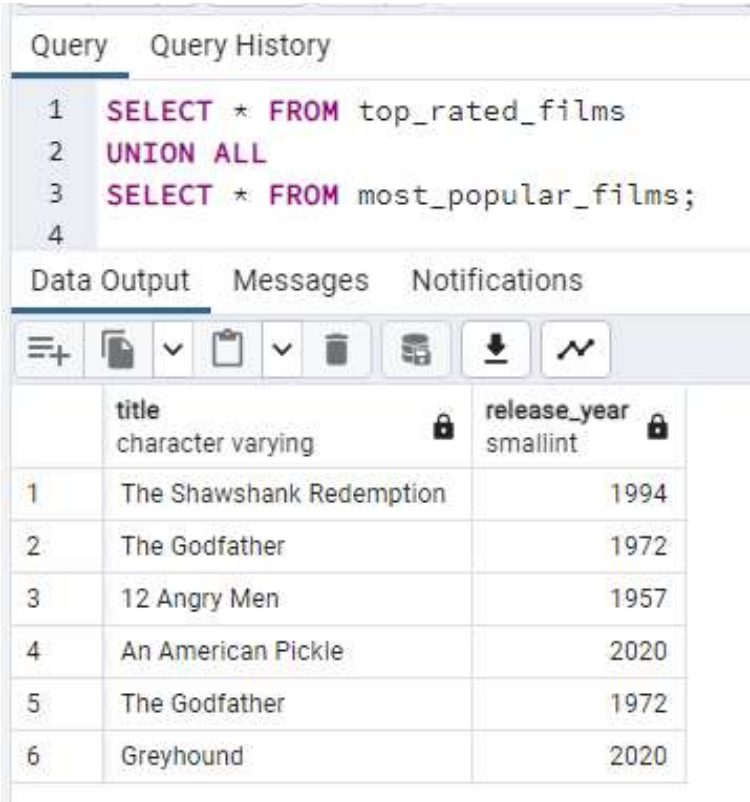
Perbedaan UNION dengan UNION ALL

- UNION ALL mencantumkan semuanya termasuk yang sama isi datanya.

```
SELECT * FROM topRatedFilms
```

```
UNION ALL
```

```
SELECT * FROM mostPopularFilms;
```



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

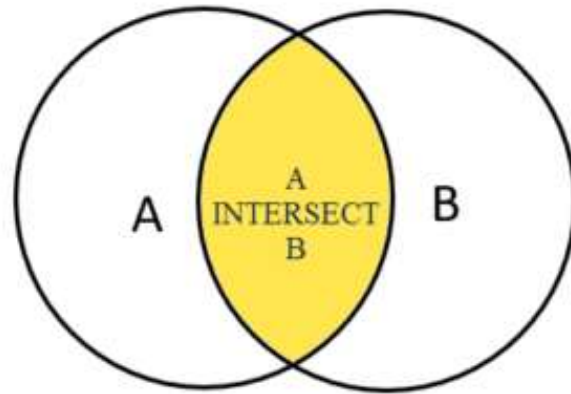
```
1 SELECT * FROM topRatedFilms
2 UNION ALL
3 SELECT * FROM mostPopularFilms;
4
```

The results window displays the output of the query, showing a table with two columns: **title** (character varying) and **release_year** (smallint). The table contains six rows of data, including duplicates from the `mostPopularFilms` table.

	title	release_year
1	The Shawshank Redemption	1994
2	The Godfather	1972
3	12 Angry Men	1957
4	An American Pickle	2020
5	The Godfather	1972
6	Greyhound	2020

INTERSECT Operator

- Ilustrasi :



- Penggunaan:

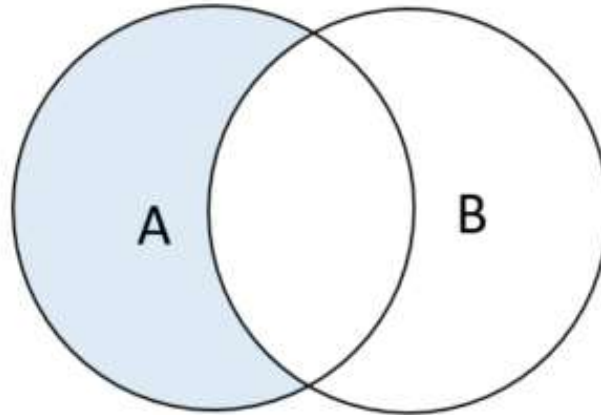
```
SELECT *  
FROM most_popular_films  
INTERSECT  
SELECT *  
FROM top_rated_films;
```

Query		Query History	
1	SELECT *		
2	FROM most_popular_films		
3	INTERSECT		
4	SELECT *		
5	FROM top_rated_films;		
6			
7			

Data Output		Messages	Notifications
	title	release_year	
	character varying	smallint	
1	The Godfather	1972	

EXCEPT Operator

- Ilustrasi :



- Penggunaan:

```
SELECT *  
FROM topRated_films  
EXCEPT  
SELECT *  
FROM most_popular_films;
```

Query		Query History	
1	SELECT *		
2	FROM topRated_films		
3	EXCEPT		
4	SELECT *		
5	FROM most_popular_films;		
6			
Data Output		Messages	Notifications
	title		release_year
	character varying		smallint
1	The Shawshank Redemption		1994
2	12 Angry Men		1957

Materi tambahan

Pelajari juga materi pengelompokan data dari :

https://tessy.lecturer.pens.ac.id/tutorial/introtoSQL/OAI_SQL6.pdf

Summary

- Telah dipelajari GROUP BY, HAVING, dan SET OPERATION

