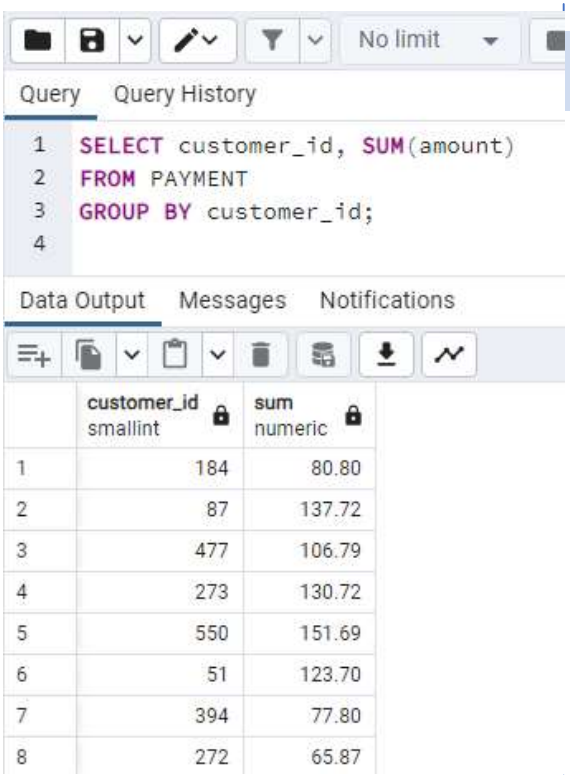
PostgreSQL - Grouping Data and Set Operation

- Tessy Badriyah, PhD.

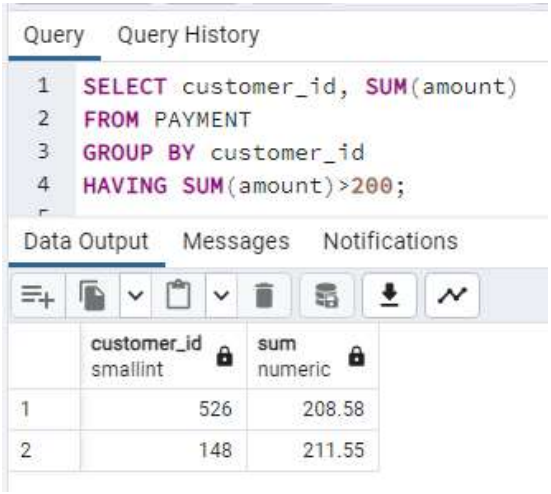
Pengelompokan Data

- Pengelompokan data dengan GROUP BY
- Biasanya digunakan Bersama dengan fungsi aggregasi
- 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;



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;



Set operation : UNION

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

Buat dua tabel: top_rated_films dan most_popular_films

Perintah pembuatan tabel:

```
CREATE TABLE top_rated_films(
        title VARCHAR NOT NULL,
        release_year SMALLINT
);
CREATE TABLE most_popular_films(
        title VARCHAR NOT NULL,
        release_year SMALLINT
);
```

Perintah pengisian tabel:

```
INSERT INTO
   top_rated_films(title,release_year)
VALUES
   ('The Shawshank Redemption',1994),
   ('The Godfather',1972),
   ('12 Angry Men',1957);


INSERT INTO
   most_popular_films(title,release_year)
VALUES
   ('An American Pickle',2020),
   ('The Godfather',1972),
   ('Greyhound',2020);
```

Set operation : UNION

Contoh penggunaan UNION

SELECT * FROM top_rated_films

UNION

SELECT * FROM most_popular_films;

Query | Query History

```
1  SELECT * FROM top_rated_films
2  UNION
3  SELECT * FROM most_popular_films;
4
```

Data Output | Messages | Notifications

	title character varying	release_year 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 top_rated_films

UNION ALL

SELECT * FROM most_popular_films;

Query | Query History

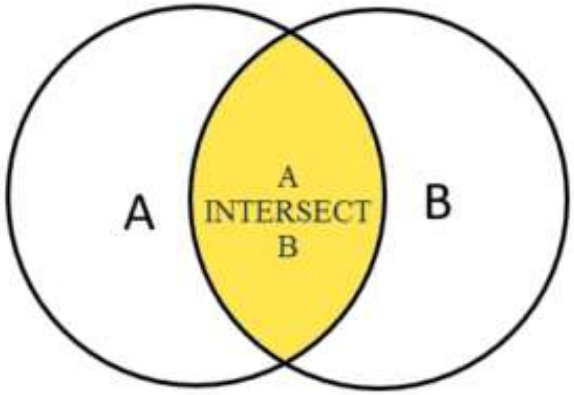
```
1  SELECT * FROM top_rated_films
2  UNION ALL
3  SELECT * FROM most_popular_films;
4
```

Data Output | Messages | Notifications

	title character varying	release_year smallint
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

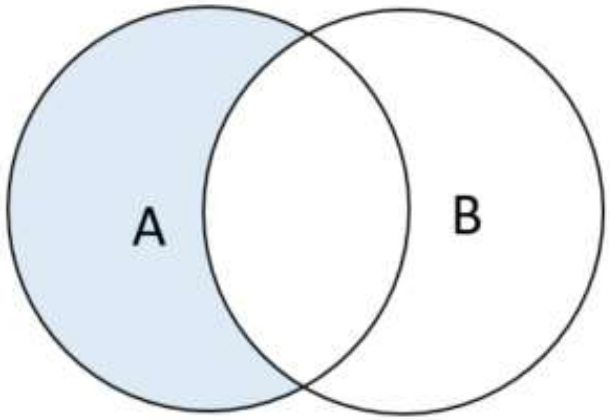
```
1  SELECT *
2  FROM most_popular_films
3  INTERSECT
4  SELECT *
5  FROM top_rated_films;
6
7
```

Data Output | Messages | Notifications

	title character varying	release_year smallint
1	The Godfather	1972

EXCEPT Operator

- Ilustrasi :



- Penggunaan:

```
SELECT *
FROM top_rated_films
EXCEPT
SELECT *
FROM most_popular_films;
```

Query | Query History

```
1  SELECT *
2  FROM top_rated_films
3  EXCEPT
4  SELECT *
5  FROM most_popular_films;
```

Data Output | Messages | Notifications

	title character varying	release_year 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