

Dibimbang

DS33B Day 20

Specialized SQL Techniques

By Rizky Febri Ibra Habibie

Table of Contents

<i>Table of Figures.....</i>	3
1. Menggunakan Subquery.....	4
a. Tampilkan nama pelanggan yang pernah melakukan transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment.....	4
b. Ambil daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film.....	4
c. Buat query untuk menampilkan aktor yang hanya membintangi satu film dalam database.	5
2. Menggunakan Window Functions.....	8
a. Gunakan RANK() untuk menentukan peringkat film berdasarkan rental_rate.	8
b. Gunakan DENSE_RANK() untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan.....	8
c. Gunakan ROW_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release_year.	9
3. Menggunakan Common Table Expressions (CTE).....	10
a. Gunakan CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali.	10
b. Gunakan CTE untuk mendapatkan daftar film dengan jumlah rental terbanyak.	11
4. Menggunakan CASE WHEN untuk Klasifikasi Data.....	12
a. Buat query yang mengelompokkan film berdasarkan rental_rate:	12
b. Buat query yang mengelompokkan pelanggan berdasarkan total transaksi mereka	13
SQL File Link	14

Table of Figures

Figure 1. transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment	4
Figure 2. daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film	5
Figure 3. query untuk menampilkan aktor yang hanya membintangi satu film dalam database	6
Figure 4. Checking.....	7
Figure 5.RANK() untuk menentukan peringkat film berdasarkan rental_rate	8
Figure 6. DENSE_RANK() untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan	9
Figure 7. ROW_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release_year	10
Figure 8. CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali	11
Figure 9. CTE untuk mendapatkan daftar film dengan jumlah rental terbanyak	12
Figure 10. query yang mengelompokkan film berdasarkan rental_rate.....	13
Figure 11. query yang mengelompokkan pelanggan berdasarkan total transaksi.....	14

1. Menggunakan Subquery

- Tampilkan nama pelanggan yang pernah melakukan transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment.

```
select
```

```
    payment_id,  
    amount,  
    (select AVG(amount) from payment) avg_amount,  
    case  
        when amount > (select AVG(amount) from payment) then  
            'Above Average'  
        else 'Below Average'  
    end as average_flag  
from payment;
```

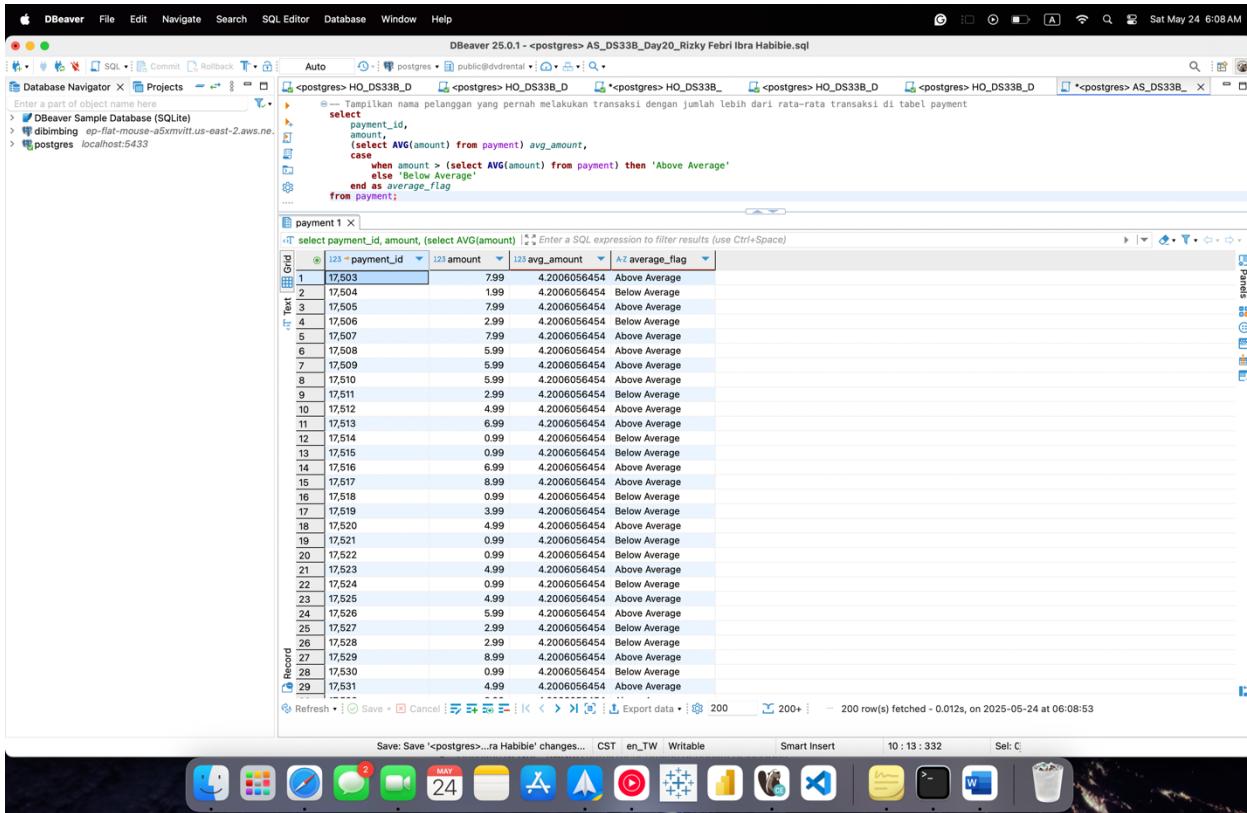


Figure 1. transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment

- Ambil daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film.

```
select
```

```
    film_id,  
    title,  
    length,
```

```
(select AVG(length) from film) avg_length,  
case  
    when length > (select AVG(length) from film) then  
'Above Average'  
        else 'Below Average'  
    end as average_flag  
from film  
order by 2 asc;
```

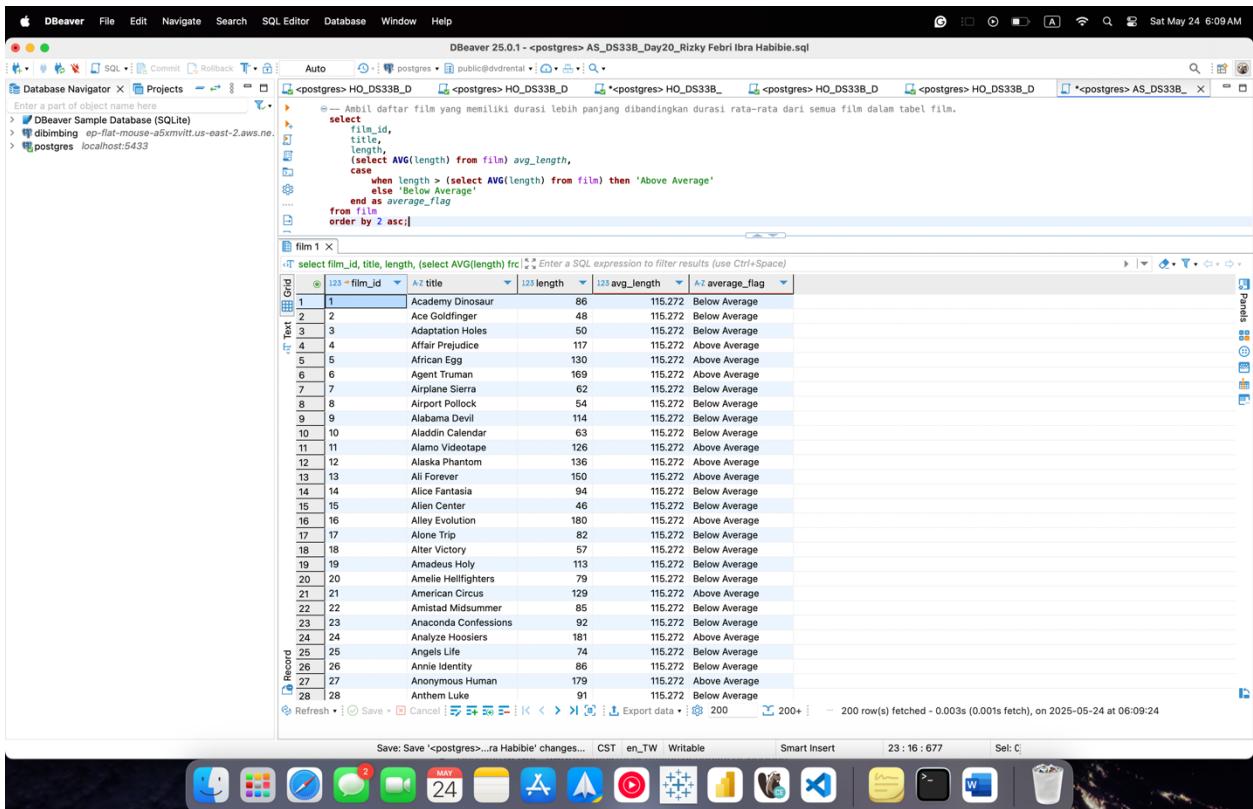


Figure 2, daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film

- c. Buat query untuk menampilkan aktor yang hanya membintangi satu film dalam database.

```
select
    actor_id,
    first_name,
    last_name
from actor
where actor_id in (
    select actor_id
    from film_actor
    group by actor_id
    having count(film_id) = 1)
```

```

-- Check
select
    a.actor_id,
    a.first_name,
    a.last_name,
    COUNT(fa.film_id) film_count
FROM actor a
JOIN film_actor fa
    ON a.actor_id = fa.actor_id
GROUP by
    a.actor_id,
    a.first_name,
    a.last_name
ORDER BY film_count asc;

```

The screenshot shows the DBeaver interface. The SQL Editor tab is active, displaying the SQL query from the previous code block. The Database Navigator tab is also visible, showing a list of databases and projects. The bottom of the screen shows the macOS Dock with various application icons.

Figure 3. query untuk menampilkan aktor yang hanya membintangi satu film dalam database

DBeaver 25.0.1 - <postgres> AS_DS33B_Day20_Rizky Febri Ibra Habibie.sql

```

-- Buat query untuk menampilkan aktor yang hanya membintangi satu film dalam database.
select
    actor_id,
    first_name,
    last_name
from actor
where actor_id in (
    select
        actor_id
    from film_actor
    group by actor_id
    having count(film_id) = 1);

-- Check
select
    a.actor_id,
    a.first_name,
    a.last_name,
    COUNT(fa.film_id) film_count
FROM actor a
JOIN film_actor fa
    ON a.actor_id = fa.actor_id
GROUP by
    a.actor_id,
    a.first_name,
    a.last_name
ORDER BY film_count asc;

```

actor 1 ×

	actor_id	first_name	last_name	film_count
1	148	Emily	Dee	14
2	199	Julia	Fawcett	15
3	35	Judy	Dean	15
4	186	Julia	Zellweger	16
5	71	Adam	Grant	18
6	31	Sissy	Sobieski	18
7	183	Russell	Close	19
8	30	Sandra	Peck	19
9	1	Penelope	Guiness	19
10	63	Cameron	Wray	19
11	91	Christopher	Berry	20
12	153	Minnie	Kilmer	20
13	160	Chris	Depp	20
14	8	Matthew	Johansson	20
15	88	Kenneth	Pesci	20
16	200	Thora	Temple	20
17	55	Fay	Kilmer	20
18	6	Bette	Nicholson	20
19	135	Rita	Reynolds	20

Save: Save '<postgres>...ra Habibie' changes... CST en_TW Writable Smart Insert 49 : 25 : 1161 Sel: C

Figure 4. Checking

2. Menggunakan Window Functions

- a. Gunakan **RANK()** untuk menentukan peringkat film berdasarkan **rental_rate**.

```
select *,
      RANK() OVER(order by rental_rate DESC) ranking
   from film;
```

The screenshot shows the DBBeaver interface with a PostgreSQL connection. The SQL Editor contains the query:

```
select *,
      RANK() OVER(order by rental_rate DESC) ranking
   from film;
```

The results pane displays a table with columns: id, title, length, replacement_cost, rating, last_update, special_features, and ranking. The ranking column shows the rank of each movie based on its rental rate.

id	title	length	replacement_cost	rating	last_update	special_features	ranking
1	9	99	22.99	PG	2013-05-26 14:50:58.951	> Behind the Scenes	1
2	9	133	27.99	PG	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
3	9	51	23.99	PG	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
4	9	98	15.99	PG-13	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
5	9	183	14.99	PG-13	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
6	9	159	29.99	PG-13	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
7	9	83	28.99	R	2013-05-26 14:50:58.951	> Deleted Scenes > Trailers [+2]	1
8	9	100	29.99	PG-13	2013-05-26 14:50:58.951	> Deleted Scenes > Trailers [+2]	1
9	9	84	9.99	R	2013-05-26 14:50:58.951	> Behind the Scenes > Trailers [+2]	1
10	9	132	10.99	PG-13	2013-05-26 14:50:58.951	> Deleted Scenes > Trailers [+2]	1
11	9	128	13.99	R	2013-05-26 14:50:58.951	> Deleted Scenes > Trailers [+2]	1
12	9	119	17.99	PG-13	2013-05-26 14:50:58.951	> Behind the Scenes > Trailers [+2]	1
13	9	60	19.99	NC-17	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
14	9	110	11.99	G	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
15	9	80	21.99	G	2013-05-26 14:50:58.951	> Trailers [+2]	1
16	9	61	17.99	G	2013-05-26 14:50:58.951	> Behind the Scenes > Trailers [+2]	1
17	9	134	17.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+2]	1
18	9	74	22.99	R	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
19	9	85	18.99	G	2013-05-26 14:50:58.951	> Trailers [+2]	1
20	9	103	21.99	PG	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
21	9	132	10.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+2]	1
22	9	137	16.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+2]	1
23	9	99	27.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+2]	1
24	9	173	20.99	PG	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
25	9	48	17.99	PG	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
26	9	71	26.99	R	2013-05-26 14:50:58.951	> Trailers [+2]	1
27	9	139	13.99	G	2013-05-26 14:50:58.951	> Trailers [+2]	1
28	9	80	25.99	PG	2013-05-26 14:50:58.951	> Trailers [+2]	1
29	9	170	22.99	PG	2013-05-26 14:50:58.951	> Trailers [+2]	1
30	9	154	21.99	R	2013-05-26 14:50:58.951	> Trailers [+2]	1
31	9	145	27.99	PG-13	2013-05-26 14:50:58.951	> Commentaries [+] > Trailers [+2]	1
32	9	118	9.99	PG	2013-05-26 14:50:58.951	> Trailers [+2]	1
...
100	9	22.99	NC-17	PG	2013-05-26 14:50:58.951	> Trailers [+2]	1

Figure 5. RANK() untuk menentukan peringkat film berdasarkan rental_rate

- b. Gunakan **DENSE_RANK()** untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan.

```
select
  c.customer_id,
  c.first_name,
  c.last_name,
  COUNT(c.customer_id) as total_transaction,
  DENSE_RANK() OVER(order by COUNT(c.customer_id) DESC)
transaction_rank
  from customer c
  join payment p on c.customer_id = p.customer_id
  group by 1,2,3
  order by transaction_rank;
```

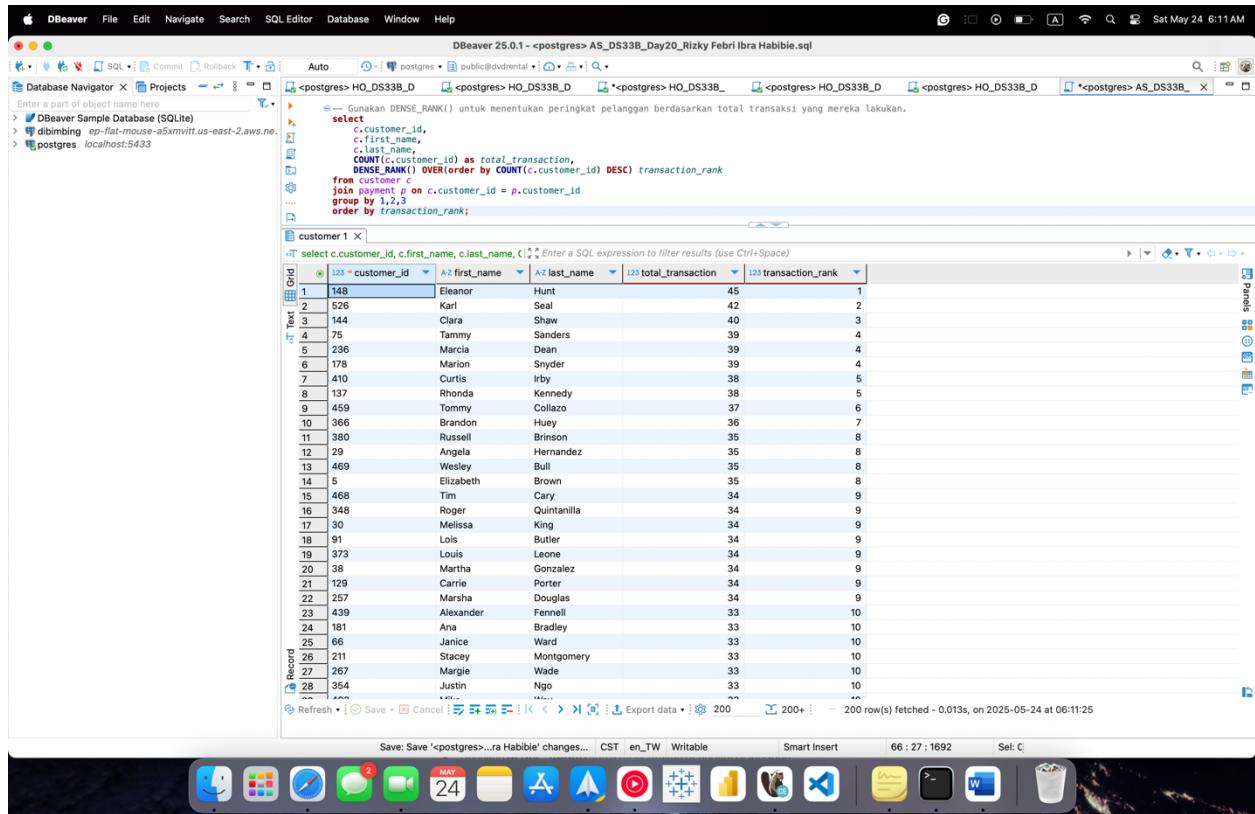


Figure 6. *DENSE_RANK()* untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan

- c. Gunakan **ROW_NUMBER()** untuk memberikan nomor urut pada daftar film berdasarkan **release_year**.

```

select *,
       ROW_NUMBER() OVER(order by release_year DESC) RN_ranking
from film;

```

DBeaver 25.0.1 - <postgres> AS_DS33B_Day20_Rizky Febri Ibra Habibie.sql

Auto Database Navigator Projects

Enter a part of object name here

> DBeaver Sample Database (SQL)

> dblimbing ep-flat-mouse-a5xmvitt.us-east-2.aws.ne...

> postgres localhost:5433

<postgres> HO_DS33B_D <postgres> HO_DS33B_D <postgres> HO_DS33B_D <postgres> HO_DS33B_D <postgres> HO_DS33B_D <postgres> HO_DS33B_D

— Gunakan ROW_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release_year.

```
select *, ROW_NUMBER() OVER(order by release_year DESC) rn_ranking
from film;
```

film 1

select * , ROW_NUMBER() OVER(order by release_... Enter a SQL expression to filter results (use Ctrl+Space)

	length	replacement_cost	rating	last_update	special_features	fulltext	rn_ranking
1	117	14.99	NC-17	2013-05-26 14:50:58.951	> Trailers	'chamber':1 'fate':4 'husband':11 'italian':2 'monkey':16 'moos':	1
2	49	19.99	R	2013-05-26 14:50:58.951	> Behind the Scenes	'australia':18 'cat':8 'drama':5 'epic':4 'explor':11 'gross':1 'moo	2
3	54	15.99	R	2013-05-26 14:50:58.951	> Trailers	'airport':1 'ancient':18 'confront':14 'epic':4 'girl':11 'india':19 'm	3
4	73	12.99	PG-13	2013-05-26 14:50:58.951	> Trailers	'boat':20 'bright':1 'conquer':14 'encount':2 'fate':4 'feminist':1	4
5	86	20.99	PG	2013-05-26 14:50:58.951	> Trailers	'academy':1 'battl':15 'canadian':20 'dinosaur':2 'drama':5 'epic	5
6	48	12.99	G	2013-05-26 14:50:58.951	> Trailers [+1]	'ace':1 'administr':9 'ancient':19 'astound':4 'car':17 'chine':20	6
7	50	18.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+1]	'adapt':1 'astound':4 'baloon':19 'car':11 'factor':20 'hole':2 'fur	7
8	117	26.99	G	2013-05-26 14:50:58.951	> Commentaries [+1]	'airfair':1 'chase':14 'documentar':5 'fanci':4 'frisbe':8 'lumberja	8
9	130	22.99	G	2013-05-26 14:50:58.951	> Deleted Scenes	'african':1 'che':11 'dentist':14 'documentar':7 'egg':2 'fast':5'	9
10	169	17.99	PG	2013-05-26 14:50:58.951	> Deleted Scenes	'agent':1 'ancient':15 'boy':11 'china':20 'despac':14 'intrepid':4 '	10
11	62	28.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+1]	'airplan':1 'boat':20 'bullet':11 'discov':14 'hunter':8 'jet':19 '	11
12	114	21.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+1]	'administr':9 'alabama':1 'boat':23 'databas':8 'devil':2 'jet':22	12
13	63	24.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+1]	'action':5 'action-pack':4 'aladdin':1 'ancient':20 'calendar':2 'c	13
14	126	16.99	G	2013-05-26 14:50:58.951	> Commentaries [+1]	'alamo':1 'bore':4 'butler':1 'cat':11 'chef':17 'convent':21 'epist	14
15	136	22.99	PG	2013-05-26 14:50:58.951	> Commentaries [+1]	'alaska':1 'australia':19 'boy':17 'chef':12 'fanci':4 'hunter':8 'ml	15
16	104	19.99	R	2013-05-26 14:50:58.951	> Commentaries [+1]	'compos':8 'convent':20 'date':1 'dentist':16 'discov':14 'moos':	16
17	150	21.99	PG	2013-05-26 14:50:58.951	> Deleted Scenes [+1]	'action':5 'action-pack':4 'all':1 'battl':16 'canadian':21 'crocodi	17
18	94	23.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+2]	'administr':13 'alic':1 'databas':12 'drama':5 'emot':4 'fantasia':	18
19	46	10.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+2]	'allen':1 'battl':15 'baloon':4 'cat':8 'center':2 'convent':21 'drai	19
20	180	23.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+1]	'alley':1 'astronaut':18 'battl':16 'compos':13 'drama':7 'evoluti	20
21	82	14.99	R	2013-05-26 14:50:58.951	> Trailers [+1]	'abandon':22 'alon':1 'boat':16 'charact':7 'compos':11 'dog':14	21
22	57	27.99	PG-13	2013-05-26 14:50:58.951	> Trailers [+1]	'agent':17 'alter':1 'canadian':20 'compos':8 'drama':5 'feminist	22
23	113	20.99	PG	2013-05-26 14:50:58.951	> Commentaries [+2]	'amadeus':1 'balloon':19 'bore':4 'conquer':14 'drama':5 'hellfight':2	23
24	79	23.99	R	2013-05-26 14:50:58.951	> Commentaries [+2]	'ameli':1 'baloon':19 'bore':4 'conquer':14 'drama':5 'hellfight':2	24
25	129	17.99	R	2013-05-26 14:50:58.951	> Commentaries [+1]	'administr':17 'american':1 'astronaut':11 'circus':2 'databas':16	25
26	85	10.99	G	2013-05-26 14:50:58.951	> Commentaries [+1]	'amistad':1 'california':20 'charact':5 'crocodi':12 'dentist':9 'er	26
27	92	9.99	R	2013-05-26 14:50:58.951	> Trailers [+1]	'anaconda':1 'australia':18 'confess':2 'dentist':2 'display':5	27
28	181	19.99	R	2013-05-26 14:50:58.951	> Trailers [+1]	'analy':1 'chef':12 'desert':21 'display':5 'explor':8 'feminist':17	28
29	74	15.99	G	2013-05-26 14:50:58.951	> Trailers	'angel':1 'astronaut':11 'battl':14 'berlin':18 'display':5 'life':2 'm	29
30	86	15.99	G	2013-05-26 14:50:58.951	> Commentaries [+1]	'abandon':20 'amaz':4 'amus':21 'ann':1 'boat':12 'chef':9 'esci	30
31	179	12.99	NC-17	2013-05-26 14:50:58.951	> Deleted Scenes [+1]	'administr':9.18 'amaz':4 'anonym':3 'astronaut':12 'databas':8,	31
32	91	16.99	PG-13	2013-05-26 14:50:58.951	> Deleted Scenes [+1]	'abandon':19 'amus':20 'anthem':1 'dog':16 'luke':2 'must':13 'c	32
33	168	11.99	NC-17	2013-05-26 14:50:58.951	> Trailers [+2]	'administr':17 'ancient':19 'antitrust':1 'databas':16 'fate':4 'fem	33

200 row(s) fetched - 0.004s, on 2025-05-24 at 06:11:54

Figure 7. ROW_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release_year

3. Menggunakan Common Table Expressions (CTE)

- Gunakan CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali.

```
WITH transaction_count AS (
    SELECT
        customer_id,
        COUNT(payment_id) AS total_transactions
    FROM payment
    GROUP BY customer_id
)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    tc.total_transactions
FROM transaction_count tc
JOIN customer c ON c.customer_id = tc.customer_id
WHERE tc.total_transactions > 10
ORDER BY tc.total_transactions DESC;
```

The screenshot shows the DBeaver interface with a SQL Editor window containing the following code:

```

-- Gunakan CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali.
WITH transaction_count AS (
    SELECT
        customer_id,
        COUNT(payment_id) AS total_transactions
    FROM payment
    GROUP BY customer_id
)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    tc.total_transactions
FROM transaction_count tc
JOIN customer c ON c.customer_id = tc.customer_id
WHERE tc.total_transactions > 10
ORDER BY tc.total_transactions DESC;

```

The results are displayed in a table titled "customer 1" with the following data:

Grid	customer_id	first_name	last_name	total_transactions
1	148	Eleanor	Hunt	45
2	526	Karl	Seal	42
3	144	Clara	Shaw	40
4	75	Tammy	Sanders	39
5	236	Marcia	Dean	39
6	178	Marion	Snyder	39
7	137	Rhonda	Kennedy	38
8	410	Curtis	Irby	38
9	459	Tommy	Collazo	37
10	366	Brandon	Huey	36
11	380	Russell	Brinson	35
12	29	Angela	Hernandez	35
13	469	Wesley	Bull	35
14	5	Elizabeth	Brown	35
15	348	Roger	Quintanilla	34
16	91	Lois	Butler	34
17	38	Martha	Gonzalez	34
18	129	Carrie	Porter	34
19	257	Marsha	Douglas	34
20	373	Louis	Leone	34
21	30	Melissa	King	34
22	468	Tim	Cary	34
23	66	Janice	Ward	33
24	267	Margie	Wade	33

At the bottom of the screen, the Mac OS X dock is visible with various application icons.

Figure 8. CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali

b. Gunakan **CTE** untuk mendapatkan daftar film dengan jumlah rental terbanyak.

```

WITH film_rental_count AS (
    SELECT
        i.film_id,
        COUNT(r.rental_id) AS rental_count
    FROM rental r
    JOIN inventory i ON r.inventory_id = i.inventory_id
    GROUP BY i.film_id
)
SELECT
    f.film_id,
    f.title,
    frc.rental_count
FROM film_rental_count frc
JOIN film f ON f.film_id = frc.film_id
ORDER BY frc.rental_count DESC;

```

```

SELECT
    f.film_id,
    f.title,
    f.rental_count
FROM film f
JOIN film_rental_count frc
ON f.film_id = frc.film_id
ORDER BY frc.rental_count DESC;

```

film_id	title	rental_count
103	Bucket Brotherhood	34
738	Rocketeer Mother	33
767	Scalawag Duck	32
382	Grit Clockwork	32
489	Juggler Hardy	32
730	Ridgemont Submarine	32
753	Rush Goodfelas	31
369	Goodfelas Salute	31
735	Robbers Joon	31
621	Network Peak	31
418	Hobbit Alien	31
31	Apache Divine	31
973	Wife Turn	31
891	Timberland Sky	31
1000	Zorro Ark	31
869	Suspects Quills	30
702	Pulp Beverly	30
609	Muscle Bright	30
979	Witches Panic	30
239	Dogma Family	30
341	Frost Head	30
563	Massacre Usual	30
374	Graffiti Love	30
127	Cat Coneheads	30

Figure 9. CTE untuk mendapatkan daftar film dengan jumlah rental terbanyak

4. Menggunakan CASE WHEN untuk Klasifikasi Data

- Buat query yang mengelompokkan film berdasarkan **rental_rate**:
 - Jika **rental_rate** lebih dari 4, kategori "Premium"
 - Jika **rental_rate** antara 2 dan 4, kategori "Regular"
 - Jika **rental_rate** kurang dari 2, kategori "Budget"

```

SELECT
    film_id,
    title,
    rental_rate,
    CASE
        WHEN rental_rate > 4 THEN 'Premium'
        WHEN rental_rate BETWEEN 2 AND 4 THEN 'Regular'
        WHEN rental_rate < 2 THEN 'Budget'
    END AS rental_category
FROM film
ORDER BY rental_rate DESC;

```

The screenshot shows the DBeaver interface on a Mac OS X desktop. The top menu bar includes Apple, DBeaver, File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The status bar at the bottom right shows the date and time: Sat May 24 6:12 AM.

The SQL Editor tab is active, displaying a query:

```

SELECT film_id, title, rental_rate, CASE WHEN rental_rate > 4 THEN 'Premium'
                                         WHEN rental_rate BETWEEN 2 AND 4 THEN 'Regular'
                                         WHEN rental_rate < 2 THEN 'Budget'
                                     END AS rental_category
FROM film
ORDER BY rental_rate DESC;

```

The results grid below shows the output of the query, listing 600 rows of film data with columns: film_id, title, rental_rate, and rental_category. The rental_rate column has values ranging from 2.99 to 4.99, and the rental_category column shows 'Premium', 'Regular', and 'Budget' categories.

Figure 10. query yang mengelompokkan film berdasarkan rental_rate

- b. Buat query yang mengelompokkan pelanggan berdasarkan total transaksi mereka:
- Pelanggan dengan total transaksi lebih dari \$100 sebagai "High Value Customer"
 - Pelanggan dengan transaksi antara \$50-\$100 sebagai "Medium Value Customer"
 - Pelanggan dengan transaksi di bawah \$50 sebagai "Low Value Customer"

```

SELECT
  c.customer_id,
  c.first_name,
  c.last_name,
  SUM(p.amount) AS total_transaction,
  CASE
    WHEN SUM(p.amount) > 100 THEN 'High Value Customer'
    WHEN SUM(p.amount) BETWEEN 50 AND 100 THEN 'Medium Value Customer'
    WHEN SUM(p.amount) < 50 THEN 'Low Value Customer'
  END AS customer_category
FROM customer c

```

```
JOIN payment p ON c.customer_id = p.customer_id  
GROUP BY 1,2,3  
ORDER BY total_transaction DESC;
```

DBeaver 25.0.1 - <postgres> AS_DS33B_Day20_Rizky Febri Ibra Habille.sql

Auto <postgres> HO_DS33B_D <postgres> AS_DS33B_D

Buat query yang mengekspresikan pelanggan berdasarkan total transaksi mereka:
 - Pelanggan dengan total transaksi lebih dari \$100 sebagai "High Value Customer"
 - Pelanggan dengan transaksi antara \$50-\$100 sebagai "Medium Value Customer"
 - Pelanggan dengan transaksi di bawah \$50 sebagai "Low Value Customer"

```

SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(p.amount) AS total_transaction,
    CASE
        WHEN SUM(p.amount) > 100 THEN 'High Value Customer'
        WHEN SUM(p.amount) BETWEEN 50 AND 100 THEN 'Medium Value Customer'
        WHEN SUM(p.amount) < 50 THEN 'Low Value Customer'
    END AS customer_category
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY 1,2,3
ORDER BY total_transaction DESC;

```

customer x

customer_id first_name last_name total_transaction customer_category

customer_id	first_name	last_name	total_transaction	customer_category
288	100	Robin Hayes	101.77	High Value Customer
289	593	Rene Mcalister	101.76	High Value Customer
290	405	Leonard Schofield	101.7	High Value Customer
291	486	Glen Talbert	100.77	High Value Customer
292	617	Brad McCurdy	100.76	High Value Customer
293	572	Sidney Burleson	100.76	High Value Customer
294	219	Willie Howell	100.75	High Value Customer
295	261	Vickie Brewer	100.75	High Value Customer
296	135	Juanita Mason	100.72	High Value Customer
297	519	Ron Deluca	99.78	Medium Value Customer
298	11	Lisa Anderson	99.77	Medium Value Customer
299	57	Evelyn Morgan	99.74	Medium Value Customer
300	297	Sherri Rhodes	99.74	Medium Value Customer
301	563	Ken Prewitt	99.73	Medium Value Customer
302	603	Angel Barclay	99.72	Medium Value Customer
303	452	Tom Milner	99.71	Medium Value Customer
304	381	Bobby Boudreau	99.68	Medium Value Customer
305	19	Ruth Martinez	98.82	Medium Value Customer
306	329	Frank Waggoner	98.77	Medium Value Customer
307	537	Clinton Buford	98.76	Medium Value Customer
308	175	Annette Olson	98.76	Medium Value Customer
309	264	Gwendolyn May	98.75	Medium Value Customer
310	456	Shane O'neil	98.75	Medium Value Customer

Save: Save '<postgres>...ra Habille' changes... CST en_TW Writable Smart Insert 142 : 33 : 4024 Sel: C

Figure 11. query yang mengelompokkan pelanggan berdasarkan total transaksi

SQL File Link

[Click here](#)