Dibimbing

DS33B Day 20

Specialized SQL Techniques

By Rizky Febri Ibra Habibie

Table of Contents

[Table of Figures 3](#_Toc198962517)

[1. Menggunakan Subquery 4](#_Toc198962518)

[a. Tampilkan nama pelanggan yang pernah melakukan transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment. 4](#_Toc198962519)

[b. Ambil daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film. 4](#_Toc198962520)

[c. Buat query untuk menampilkan aktor yang hanya membintangi satu film dalam database. 5](#_Toc198962521)

[2. Menggunakan Window Functions 8](#_Toc198962522)

[a. Gunakan RANK() untuk menentukan peringkat film berdasarkan rental\_rate. 8](#_Toc198962523)

[b. Gunakan DENSE\_RANK() untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan. 8](#_Toc198962524)

[c. Gunakan ROW\_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release\_year. 9](#_Toc198962525)

[3. Menggunakan Common Table Expressions (CTE) 10](#_Toc198962526)

[a. Gunakan CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali. 10](#_Toc198962527)

[b. Gunakan CTE untuk mendapatkan daftar film dengan jumlah rental terbanyak. 11](#_Toc198962528)

[4. Menggunakan CASE WHEN untuk Klasifikasi Data 12](#_Toc198962529)

[a. Buat query yang mengelompokkan film berdasarkan rental\_rate: 12](#_Toc198962530)

[b. Buat query yang mengelompokkan pelanggan berdasarkan total transaksi mereka 13](#_Toc198962531)

[SQL File Link 14](#_Toc198962532)

# **Table of Figures**

[Figure 1. transaksi dengan jumlah lebih dari rata-rata transaksi di tabel payment 4](#_Toc198962071)

[Figure 2. daftar film yang memiliki durasi lebih panjang dibandingkan durasi rata-rata dari semua film dalam tabel film 5](#_Toc198962072)

[Figure 3. query untuk menampilkan aktor yang hanya membintangi satu film dalam database 6](#_Toc198962073)

[Figure 4. Checking 7](#_Toc198962074)

[Figure 5.RANK() untuk menentukan peringkat film berdasarkan rental\_rate 8](#_Toc198962075)

[Figure 6. DENSE\_RANK() untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan 9](#_Toc198962076)

[Figure 7. ROW\_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release\_year 10](#_Toc198962077)

[Figure 8. CTE untuk membuat daftar pelanggan yang melakukan transaksi lebih dari 10 kali 11](#_Toc198962078)

[Figure 9. CTE untuk mendapatkan daftar film dengan jumlah rental terbanyak 12](#_Toc198962079)

[Figure 10. query yang mengelompokkan film berdasarkan rental\_rate 13](#_Toc198962080)

[Figure 11. query yang mengelompokkan pelanggan berdasarkan total transaksi 14](#_Toc198962081)

# **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*

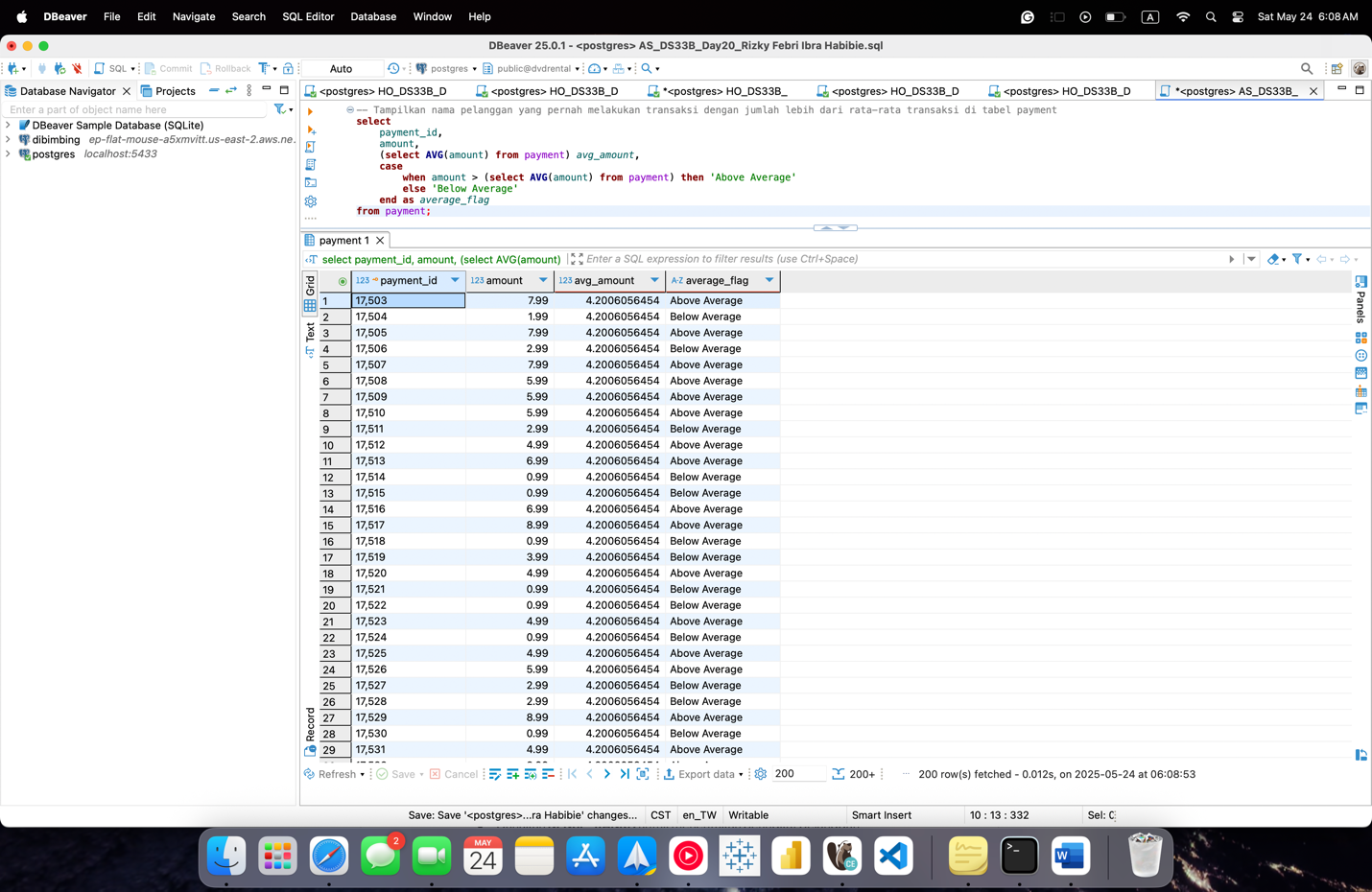
**from** payment;

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

A screenshot of a computer

AI-generated content may be incorrect.**order** **by** 2 **asc**;

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

## 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**;

A screenshot of a computer

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4. Checking

# **Menggunakan Window Functions**

## Gunakan **RANK()** untuk menentukan peringkat film berdasarkan **rental\_rate**.

**select** \*,

**RANK**() **OVER**(**order** **by** rental\_rate **DESC**) *ranking*

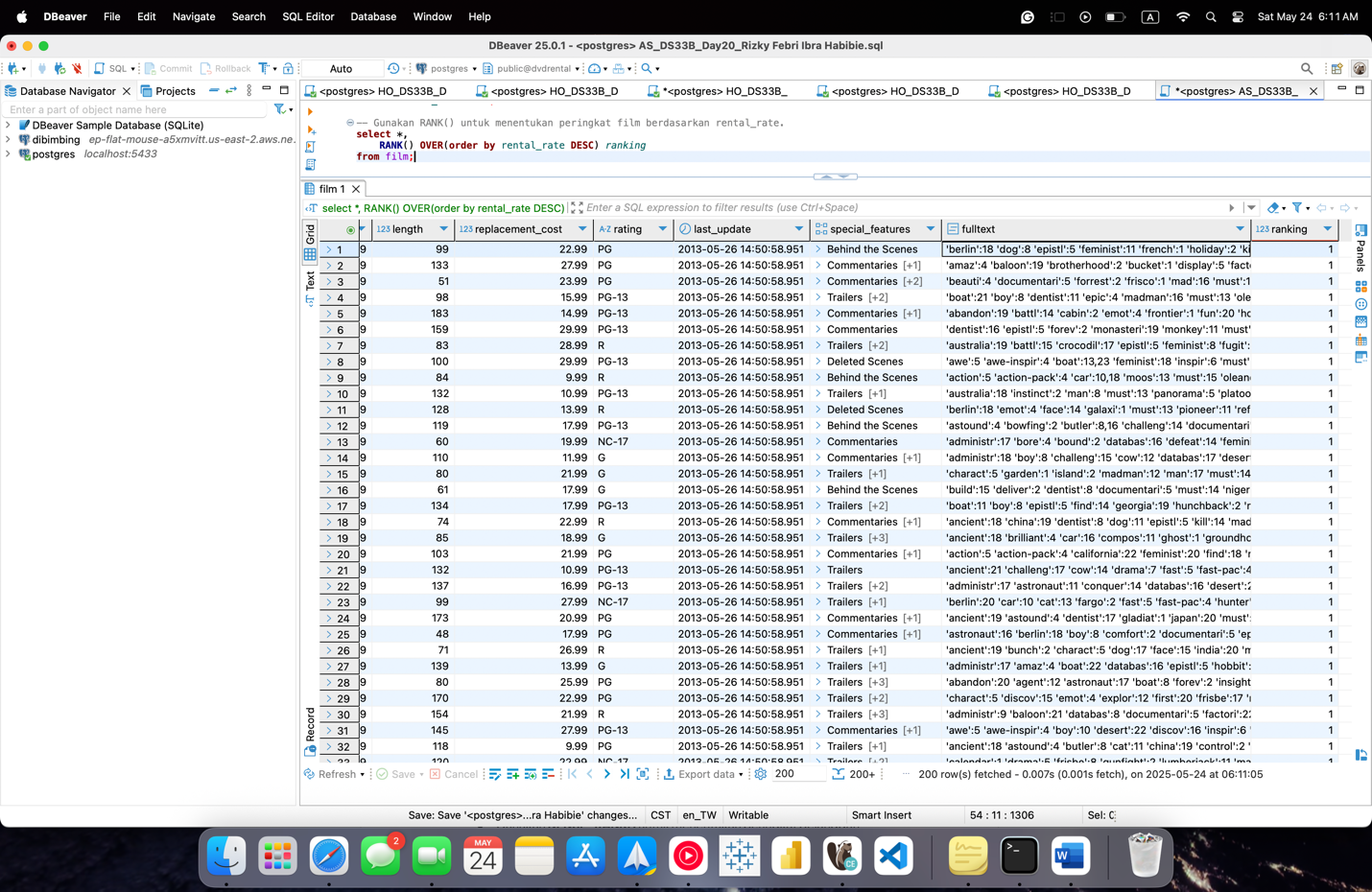
**from** film;

Figure 5.RANK() untuk menentukan peringkat film berdasarkan rental\_rate

## 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*;

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6. DENSE\_RANK() untuk menentukan peringkat pelanggan berdasarkan total transaksi yang mereka lakukan

## 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;

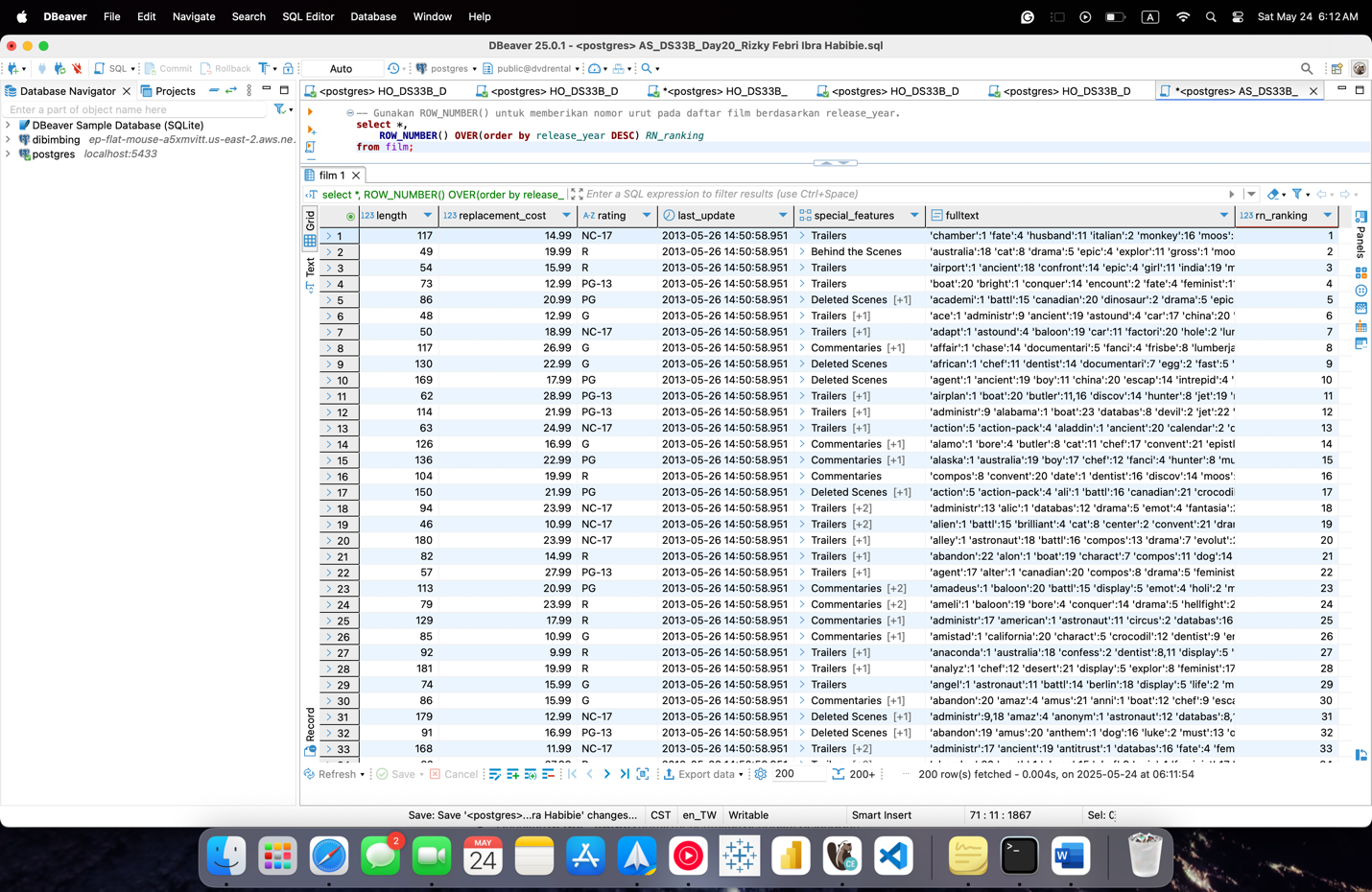


Figure 7. ROW\_NUMBER() untuk memberikan nomor urut pada daftar film berdasarkan release\_year

# **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**;

A screenshot of a computer

AI-generated content may be incorrect.

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

## 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**;

A screenshot of a computer

AI-generated content may be incorrect.

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

# **Menggunakan CASE WHEN untuk Klasifikasi Data**

## Buat query yang mengelompokkan film berdasarkan **rental\_rate**:

* + 1. Jika **rental\_rate** lebih dari 4, kategori "Premium"
    2. Jika **rental\_rate** antara 2 dan 4, kategori "Regular"
    3. 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**;

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10. query yang mengelompokkan film berdasarkan rental\_rate

* 1. Buat query yang mengelompokkan pelanggan berdasarkan total transaksi mereka:
     1. Pelanggan dengan total transaksi lebih dari $100 sebagai "High Value Customer"
     2. Pelanggan dengan transaksi antara $50-$100 sebagai "Medium Value Customer"
     3. 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

A screenshot of a computer

AI-generated content may be incorrect.**ORDER** **BY** *total\_transaction* **DESC**;

Figure 11. query yang mengelompokkan pelanggan berdasarkan total transaksi

# SQL File Link

[Click here](https://github.com/rfih/Dibimbing---DSDA/blob/126e588506ec1d58aae2f0247659f241ff6fa193/Assignment%20Day%2020%20Specialized%20SQL%20Techniques/AS_DS33B_Day20_Rizky%20Febri%20Ibra%20Habibie.sql)