Dibimbing Data Science 33B

Case Study Basic SQL Queries – Day 18

By : Rizky Febri Ibra Habibie

1. **Membuat Database dan Tabel**

```
create database dibimbing;

create table student (
     id INT primary KEY,
     nama varchar,
     institute varchar,
     berat_badan float,
     tinggi_badan float
);

insert into student (id, nama, institute, berat_badan,
tinggi_badan)
values
     (110, 'Astuti', 'ITB', 56, 163),
     (111, 'Bastomi', 'UGM', 70, 174),
     (112, 'Charlie', 'NUS', 63, 166),
     (113, 'Antony', 'Betis', 69, 177),
     (114, 'Yamal', 'Barca', 70, 180);
```

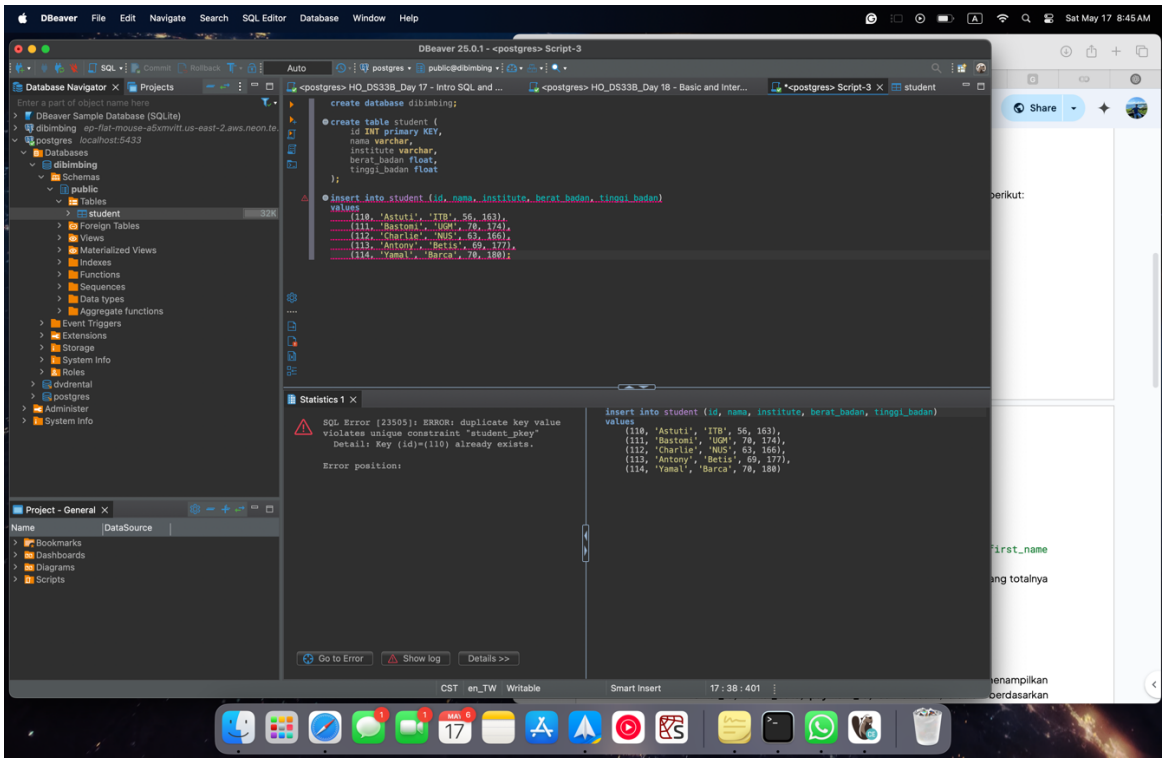

Figure 1. Membuat Database students Dengan 5 Data

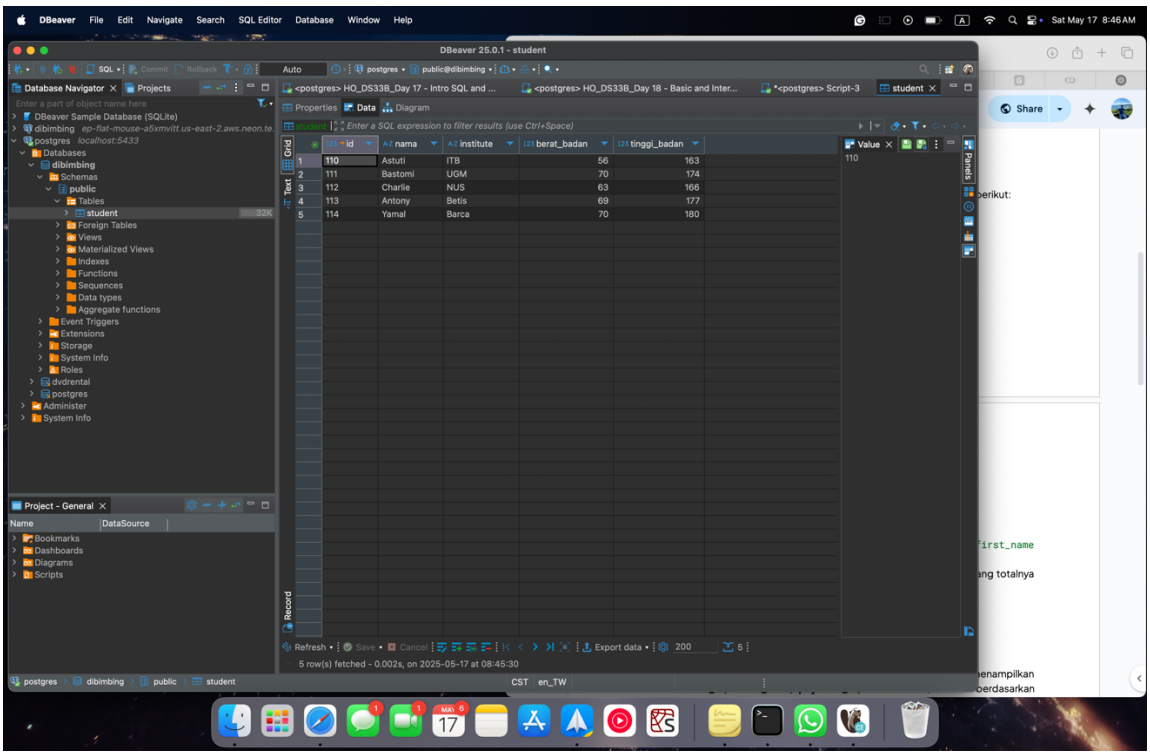

Figure 2. Hasil Tabel Students dengan 5 Data

2. Query Data pada Skema dvdrental
    a. **Tampilkan `first_name` dan `last_name`** dari aktor yang memiliki `first_name` "Jennifer", "Nick", atau "Ed".

```
select *
from actor
where first_name in ('Jennifer', 'Nick', 'Ed');
```

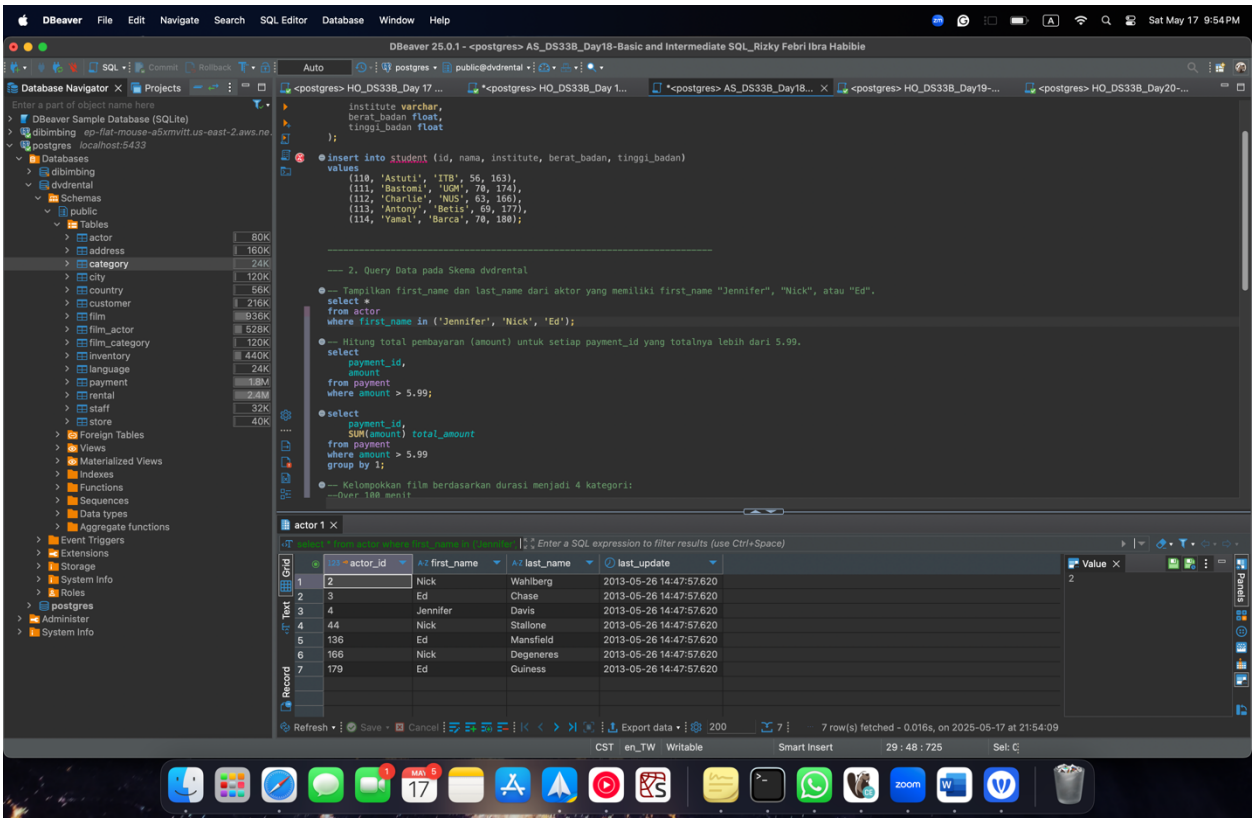

Figure 3. Hasil first_name dan last_name

    b. Hitung **total pembayaran** (amount) untuk setiap payment_id yang totalnya lebih dari **5.99**.

```
select
     payment_id,
     amount
from payment
where amount > 5.99;

select
     payment_id,
     SUM(amount) total_amount
from payment
where amount > 5.99
```

```
group by 1;
```

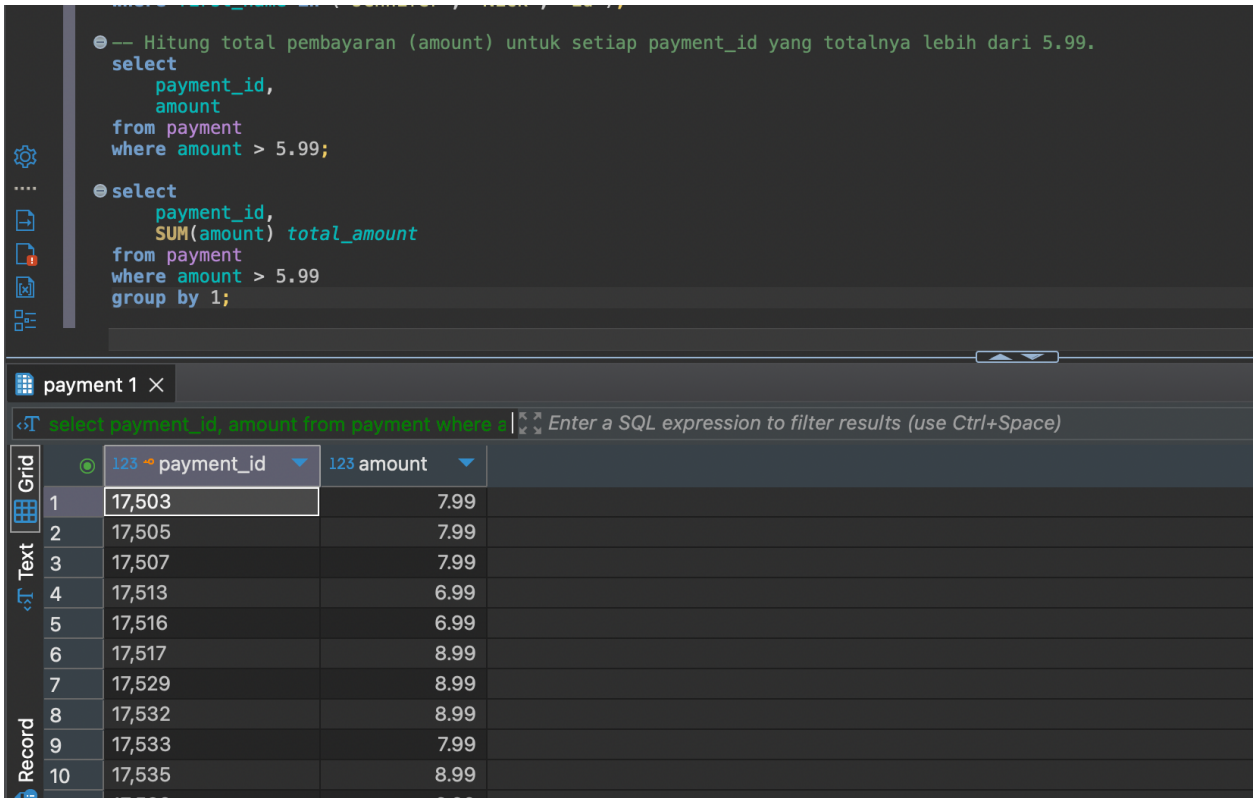

Figure 4. Hasil hitung total pembayaran (amount) untuk setiap payment_id

c. Kelompokkan **film berdasarkan durasi** menjadi 4 kategori:
   i. Over 100 menit
   ii. 87-100 menit
   iii. 72-86 menit
   iv. Under 72 menit

```
select
	film_id,
	title,
	case
		when length > 100 then 'Over 100 menit'
		when length >= 87 then '87-100 menit'
		when length >= 72 then '72-86 menit'
		when length < 72 then 'Under 72 menit'
	end as length_category
from film;
```

```sql
-- Kelompokkan film berdasarkan durasi menjadi 4 kategori:
--Over 100 menit
--87-100 menit
--72-86 menit
--Under 72 menit

select
    film_id,
    title,
    case
        when length > 100 then 'Over 100 menit'
        when length >= 87 then '87-100 menit'
        when length >= 72 then '72-86 menit'
        when length < 72 then 'Under 72 menit'
    end as length_category
from film;
```

film 1

select film_id, title, case when length > 100 then '(Enter a SQL expression to filter results (use Ctrl+Space)

	film_id	title	length_category
1	133	Chamber Italian	Over 100 menit
2	384	Grosse Wonderful	Under 72 menit
3	8	Airport Pollock	Under 72 menit
4	98	Bright Encounters	72-86 menit
5	1	Academy Dinosaur	72-86 menit
6	2	Ace Goldfinger	Under 72 menit
7	3	Adaptation Holes	Under 72 menit
8	4	Affair Prejudice	Over 100 menit
9	5	African Egg	Over 100 menit
10	6	Agent Truman	Over 100 menit
11	7	Airplane Sierra	Under 72 menit

Figure 5. Pengelompokkan film berdasarkan durasi

d. Gabungkan data dari tabel rental dan payment untuk menampilkan **rental_id, rental_date, payment_id, dan amount**, urutkan berdasarkan amount secara **ascending**.

```sql
select *
from rental;

select *
from payment;

select
    r.rental_id,
    r.rental_date,
    p.payment_id,
    p.amount
from rental r
inner join payment p
```

```
    on r.rental_id = p.rental_id
order by amount asc;
```

```
-- Gabungkan data dari tabel rental dan payment untuk menampilkan rental_id, rental_date, payment_id, dan amount, urutkan berdasarkan amount secara ascending.
select *
from rental;

select *
from payment;

select
    r.rental_id,
    r.rental_date,
    p.payment_id,
    p.amount
from rental r
inner join payment p
    on r.rental_id = p.rental_id
order by amount asc;
```

rental(+) 1

select r.rental_id, r.rental_date, p.payment_id, p.ar | Enter a SQL expression to filter results (use Ctrl+Space)

	rental_id	rental_date	payment_id	amount
1	14,516	2006-02-14 15:16:03.000	31,970	0
2	13,161	2006-02-14 15:16:03.000	32,066	0
3	12,610	2006-02-14 15:16:03.000	31,920	0
4	15,191	2006-02-14 15:16:03.000	32,040	0
5	14,425	2006-02-14 15:16:03.000	31,996	0
6	12,959	2006-02-14 15:16:03.000	31,925	0
7	14,769	2006-02-14 15:16:03.000	31,946	0
8	12,915	2006-02-14 15:16:03.000	31,983	0
9	15,282	2006-02-14 15:16:03.000	32,063	0
10	13,968	2006-02-14 15:16:03.000	32,015	0
11	15,497	2006-02-14 15:16:03.000	32,050	0

Value
14516

Figure 6. Penggabungan Data dari Tabel rental dan payment

e. Gunakan **UNION** untuk menggabungkan alamat (address) yang memiliki **city_id = 42** dengan **city_id = 300**.

```
select * from city;
select * from address;

select
    *
from address
where city_id = 300
union
select
    *
from address
where city_id = 42;
```

```sql
-- Gunakan UNION untuk menggabungkan alamat (address) yang memiliki city_id = 42 dengan city_id = 300.
select * from city;
select * from address;

select
    *
from address
where city_id = 300
union
select
    *
from address
where city_id = 42;
```

Results 1

select * from address where city_id = 300 union se | Enter a SQL expression to filter results (use Ctrl+Space)

	address_id	address	address2	district	city_id	postal_code	phone	last_update
1	1	47 MySakila Drive	[NULL]	Alberta	300			2006-02-15 09:45:30.000
2	3	23 Workhaven Lane	[NULL]	Alberta	300		14033335568	2006-02-15 09:45:30.000
3	335	587 Benguela Manor		Illinois	42	91590	165450987037	2006-02-15 09:45:30.000
4	543	43 Vilnius Manor		Colorado	42	79814	484500282381	2006-02-15 09:45:30.000

Figure 7. Penggunaan UNION untuk Menggabungkan address

SQL Link :

https://github.com/rfih/Dibimbing---DSDA/blob/3e4921d6bb7288e5be8de1cdbf8a58dd11083f59/Assignment%20Day%2018%20-%20Basic%20SQL%20Queries/AS_DS33B_Day18-Basic%20and%20Intermediate%20SQL_Rizky%20Febri%20Ibra%20Habibie.sql