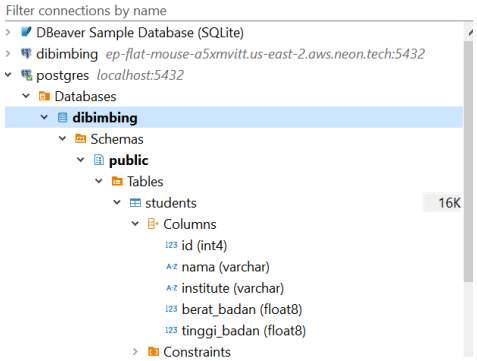
Case Study Basic SQL Queries – Day 18 Assignment

1. **Membuat Database dan Tabel**
   - Buatlah database bernama `dibimbing`.
     Use query: create database dibimbing;



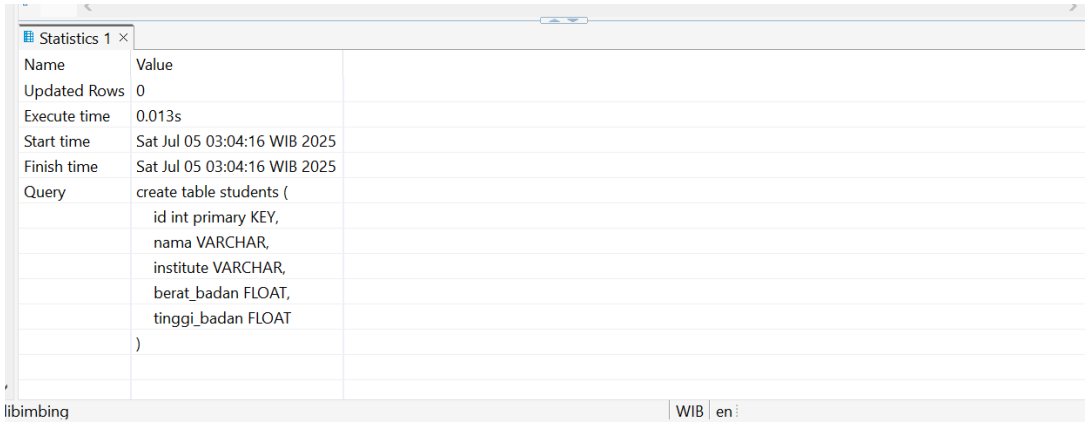
   - Di dalamnya, buat tabel `students` dengan skema berikut:
     - `id` (int, primary key)
     - `nama` (varchar)
     - `institute` (varchar)
     - `berat_badan` (float)
     - `tinggi_badan` (float)

```sql
create table students (
    id int primary KEY,
    nama VARCHAR,
    institute VARCHAR,
    berat_badan FLOAT,
    tinggi_badan FLOAT
);

-- drop table students;

-- Tambahkan setidaknya 5 baris data baru
INSERT INTO students (id, nama, institute, berat_badan, tinggi_badan) VALUES
    (123001, 'Bella',   'Fisika',           55.2, 165),
    (123002, 'Chandra', 'Teknik Elektro',   68.0, 180),
    (123003, 'Devi',    'Matematika',       50.5, 158),
    (123004, 'Eric',    'Biologi',          72.3, 185),
    (123005, 'Farah',   'Teknik Informatika',58.4, 170);
```

Result:

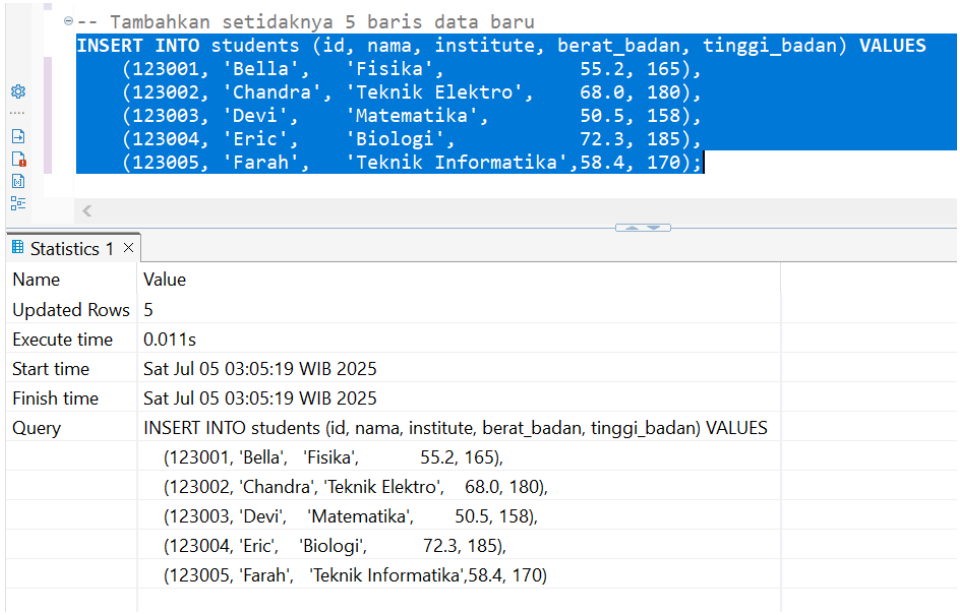


Statistics 1

Name	Value
Updated Rows	0
Execute time	0.013s
Start time	Sat Jul 05 03:04:16 WIB 2025
Finish time	Sat Jul 05 03:04:16 WIB 2025
Query	create table students (id int primary KEY, nama VARCHAR, institute VARCHAR, berat_badan FLOAT, tinggi_badan FLOAT)

libimbing WIB en

- Masukkan minimal 5 data dengan nilai yang berbeda.



```
-- Tambahkan setidaknya 5 baris data baru
INSERT INTO students (id, nama, institute, berat_badan, tinggi_badan) VALUES
    (123001, 'Bella',    'Fisika',            55.2, 165),
    (123002, 'Chandra', 'Teknik Elektro',    68.0, 180),
    (123003, 'Devi',     'Matematika',       50.5, 158),
    (123004, 'Eric',     'Biologi',          72.3, 185),
    (123005, 'Farah',    'Teknik Informatika',58.4, 170);
```

Statistics 1

Name	Value
Updated Rows	5
Execute time	0.011s
Start time	Sat Jul 05 03:05:19 WIB 2025
Finish time	Sat Jul 05 03:05:19 WIB 2025
Query	INSERT INTO students (id, nama, institute, berat_badan, tinggi_badan) VALUES (123001, 'Bella', 'Fisika', 55.2, 165), (123002, 'Chandra', 'Teknik Elektro', 68.0, 180), (123003, 'Devi', 'Matematika', 50.5, 158), (123004, 'Eric', 'Biologi', 72.3, 185), (123005, 'Farah', 'Teknik Informatika',58.4, 170)

2. Query Data pada Skema `dvdrental`

Link Dataset : [dvdrental]

- **Tampilkan `first_name` dan `last_name`** dari aktor yang memiliki `first_name` "Jennifer", "Nick", atau "Ed".

Query and Result:

```sql
-- dvdrental
select
    first_name,
    last_name
from actor
where first_name in ('Jennifer', 'Nick', 'Ed');
```

actor 1

```
select first_name, last_name from ac
```

Enter a SQL expression to filter results (use Ctrl+Sp

Grid | Text | Record

	first_name	last_name
1	Nick	Wahlberg
2	Ed	Chase
3	Jennifer	Davis
4	Nick	Stallone
5	Ed	Mansfield
6	Nick	Degeneres
7	Ed	Guiness

Refresh | Save | Cancel | Export data

- Hitung **total pembayaran** (`amount`) untuk setiap `payment_id` yang lebih besar dari **5.99**.

```sql
-- Hitung total pembayaran (amount) untuk setiap payment_id yang lebih besar dari 5.99.
select
    payment_id,
--  customer_id,
    sum(amount) amount
    from payment p
where amount > 5.99
group by 1
order by 2 desc;

-- Kelompokkan film berdasarkan durasi menjadi 4 kategori:
-- > 100 menit, 87 <= durasi <= 100 menit, 72 <= durasi <= 86 menit, < 72 menit
-- film_id, title, description, release_year
-- select * from film limit 5;
select
    f.film_id,
    f.title
```

ayment 1

```
elect payment_id, sum(amount) amo
```

Enter a SQL expression to filter results (use Ctrl+Space)

	payment_id	amount
1	20,403	11.99
2	24,553	11.99
3	24,866	11.99
4	29,136	11.99
5	22,650	11.99
6	23,757	11.99
7	28,799	11.99
8	28,814	11.99
9	28,449	10.99
10	26,978	10.99

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

```
-- Kelompokkan film berdasarkan durasi menjadi 4 kategori:
-- > 100 menit, 87 <= durasi <= 100 menit, 72 <= durasi <= 86 menit, < 72 menit
-- film_id, title, description, release_year
-- select * from film limit 5;
select
    f.film_id,
    f.title,
    f.length,
    case
        when f.length > 100 then '> 100 minutes'
        when f.length <= 100 and f.length >= 87 then '87-100 minutes'
        when f.length <= 86 and f.length >= 72 then '72-86 minutes'
        else '< 72 minutes'
    end as duration
from film f
order by 2;
```

lm 1 ×

elect f.film_id, f.title, f.length, case wh | Enter a SQL expression to filter results (use Ctrl+Space)

	film_id	title	length	duration
1	1	Academy Dinosaur	86	72-86 minutes
2	2	Ace Goldfinger	48	< 72 minutes
3	3	Adaptation Holes	50	< 72 minutes
4	4	Affair Prejudice	117	> 100 minutes
5	5	African Egg	130	> 100 minutes
6	6	Agent Truman	169	> 100 minutes
7	7	Airplane Sierra	62	< 72 minutes
8	8	Airport Pollock	54	< 72 minutes
9	9	Alabama Devil	114	> 100 minutes
10	10	Aladdin Calendar	63	< 72 minutes

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

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

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

select * from address limit 5;
-- address_id, address, address2, district, city_id, postal_code, phone, last_update
select * from city limit 5;
-- city_id, city, country_id, last_update
```

ental(+) 1 ×

elect r.rental_id, r.rental_date, p.paym 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

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

```sql
-- Gunakan UNION untuk menggabungkan alamat (address)
-- yang memiliki city_id = 42 dengan city_id = 300.

select
    address,
    city_id
from address
where city_id = 42
union all
select
    address,
    city_id
from address
where city_id = 300
order by 1;
```

ults 1 ×

ect address, city_id from address w Enter a SQL expression to filter results (use Ctrl+

address	city_id
23 Workhaven Lane	300
43 Vilnius Manor	42
47 MySakila Drive	300
587 Benguela Manor	42