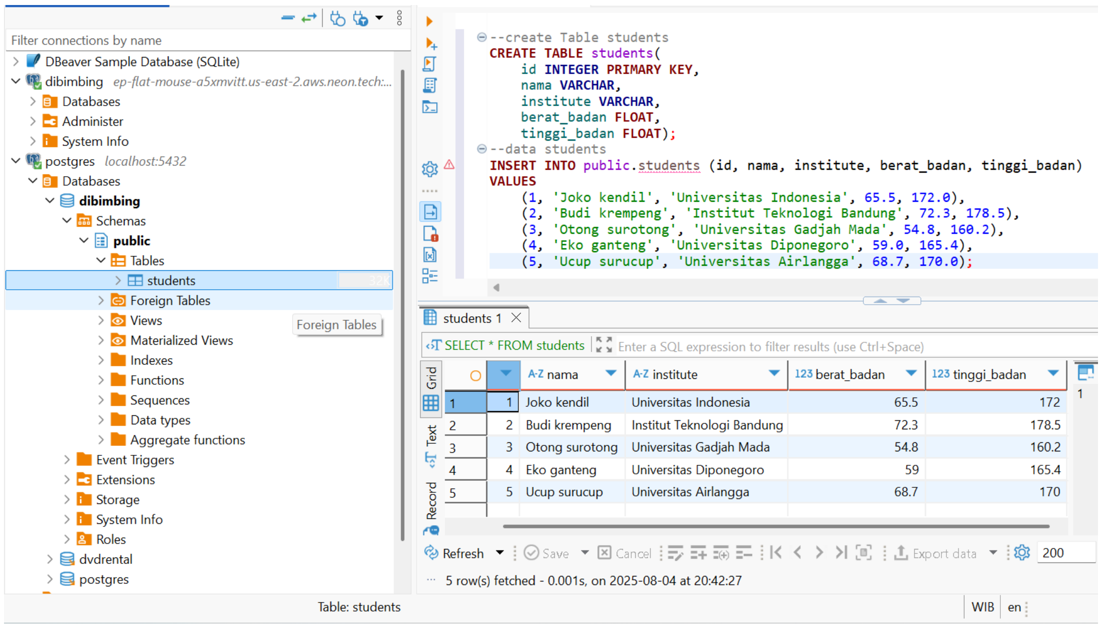
1. Buatlah sebuah database dengan nama **`dibimbing`**.
2. Di dalam database tersebut, buat sebuah tabel bernama **`students`** dalam skema **`public`**, dengan struktur kolom sebagai berikut:
   - `id` (integer)
   - `nama` (varchar)
   - `institute` (varchar)
   - `berat_badan` (float)
   - `tinggi_badan` (float)

Isi tabel tersebut dengan minimal **5 baris data**, dan pastikan setiap baris memiliki nilai yang **berbeda-beda**. Nilai data dapat ditentukan secara bebas.

```sql
--create Table students
CREATE TABLE students(
    id INTEGER PRIMARY KEY,
    nama VARCHAR,
    institute VARCHAR,
    berat_badan FLOAT,
    tinggi_badan FLOAT
);

--data students
INSERT INTO public.students (id, nama, institute, berat_badan, tinggi_badan)
VALUES
    (1, 'Joko kendil', 'Universitas Indonesia', 65.5, 172.0),
    (2, 'Budi krempeng', 'Institut Teknologi Bandung', 72.3, 178.5),
    (3, 'Otong surotong', 'Universitas Gadjah Mada', 54.8, 160.2),
    (4, 'Eko ganteng', 'Universitas Diponegoro', 59.0, 165.4),
    (5, 'Ucup surucup', 'Universitas Airlangga', 68.7, 170.0);
```



```sql
--create Table students
CREATE TABLE students(
    id INTEGER PRIMARY KEY,
    nama VARCHAR,
    institute VARCHAR,
    berat_badan FLOAT,
    tinggi_badan FLOAT);
--data students
INSERT INTO public.students (id, nama, institute, berat_badan, tinggi_badan)
VALUES
    (1, 'Joko kendil', 'Universitas Indonesia', 65.5, 172.0),
    (2, 'Budi krempeng', 'Institut Teknologi Bandung', 72.3, 178.5),
    (3, 'Otong surotong', 'Universitas Gadjah Mada', 54.8, 160.2),
    (4, 'Eko ganteng', 'Universitas Diponegoro', 59.0, 165.4),
    (5, 'Ucup surucup', 'Universitas Airlangga', 68.7, 170.0);
```

SELECT * FROM students | Enter a SQL expression to filter results (use Ctrl+Space)

		nama	institute	berat_badan	tinggi_badan
1	1	Joko kendil	Universitas Indonesia	65.5	172
2	2	Budi krempeng	Institut Teknologi Bandung	72.3	178.5
3	3	Otong surotong	Universitas Gadjah Mada	54.8	160.2
4	4	Eko ganteng	Universitas Diponegoro	59	165.4
5	5	Ucup surucup	Universitas Airlangga	68.7	170

5 row(s) fetched - 0.001s, on 2025-08-04 at 20:42:27

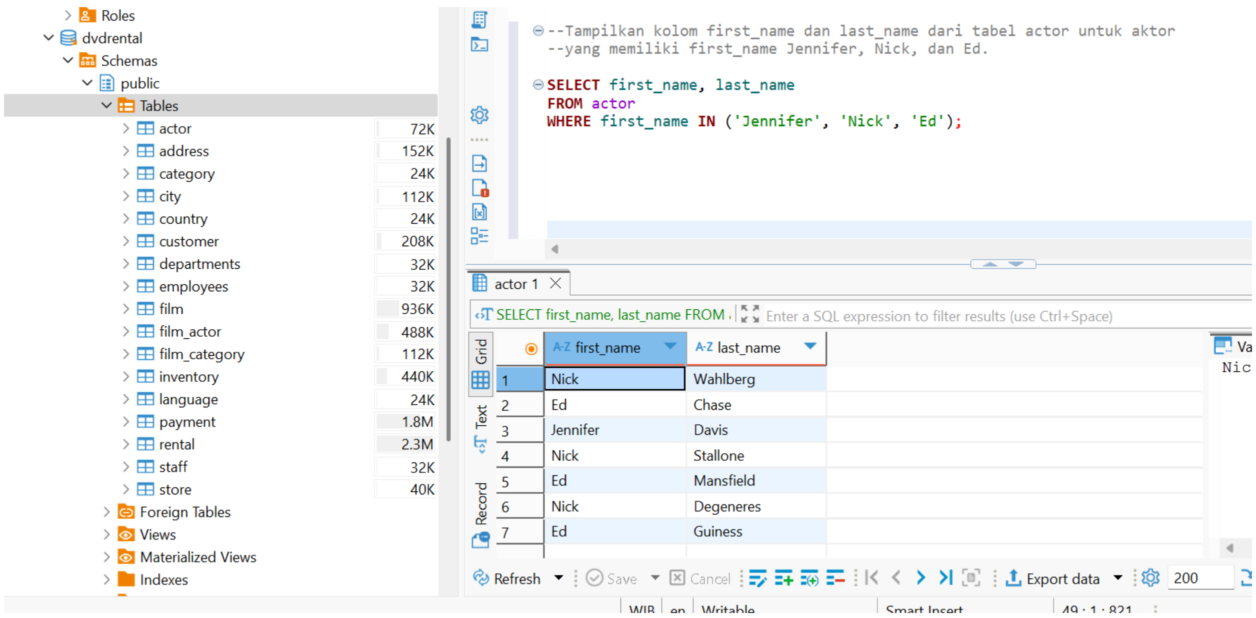
Part 2: Query on Existing Schema `dvdrental`

Gunakan skema **`dvdrental`** untuk menjawab pertanyaan berikut:

Tampilkan kolom `first_name` dan `last_name` dari tabel `actor` untuk aktor yang memiliki `first_name` **Jennifer**, **Nick**, dan **Ed**.

```sql
--Tampilkan kolom first_name dan last_name dari tabel actor untuk aktor
--yang memiliki first_name Jennifer, Nick, dan Ed.

SELECT first_name, last_name
FROM actor
WHERE first_name IN ('Jennifer', 'Nick', 'Ed');
```



```
Roles
dvdrental
  Schemas
    public
      Tables
        actor          72K
        address        152K
        category       24K
        city           112K
        country        24K
        customer       208K
        departments    32K
        employees      32K
        film           936K
        film_actor     488K
        film_category  112K
        inventory      440K
        language       24K
        payment        1.8M
        rental         2.3M
        staff          32K
        store          40K
      Foreign Tables
      Views
      Materialized Views
      Indexes
```

```sql
--Tampilkan kolom first_name dan last_name dari tabel actor untuk aktor
--yang memiliki first_name Jennifer, Nick, dan Ed.

SELECT first_name, last_name
FROM actor
WHERE first_name IN ('Jennifer', 'Nick', 'Ed');
```

actor 1

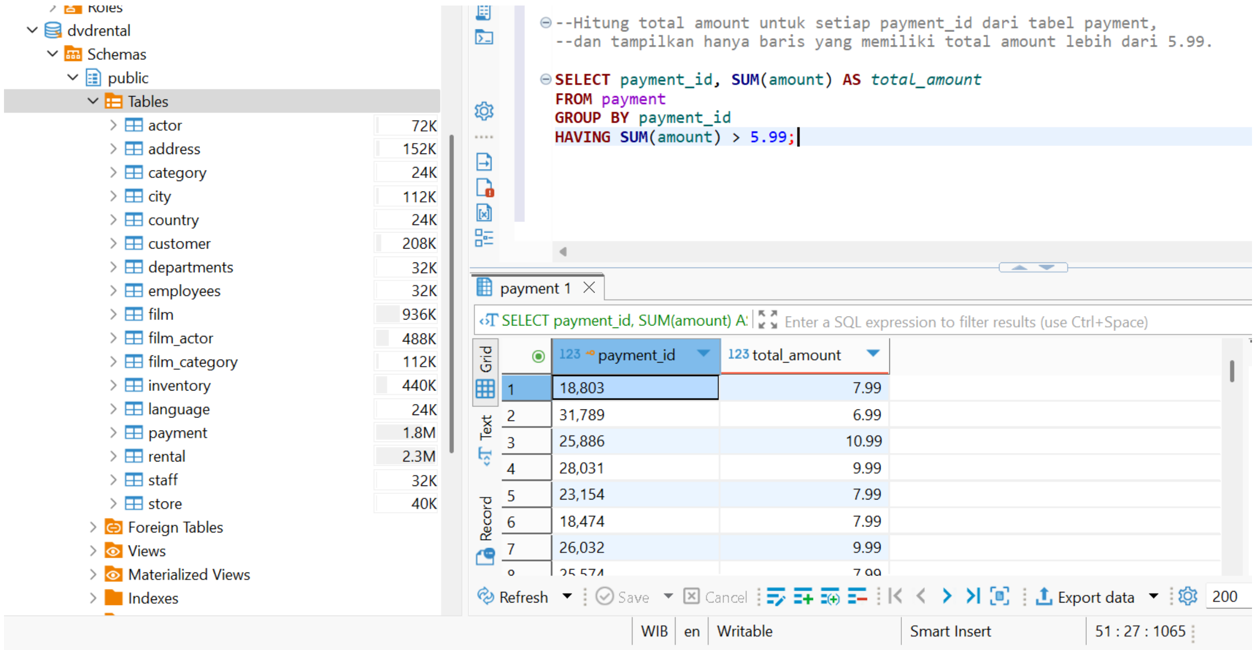
SELECT first_name, last_name FROM

	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

Hitung total `amount` untuk setiap `payment_id` dari tabel `payment`, dan **tampilkan hanya** baris yang memiliki total `amount` **lebih dari 5.99**.
Petunjuk: Gunakan klausa `HAVING`.

```sql
--Hitung total amount untuk setiap payment_id dari tabel payment,
--dan tampilkan hanya baris yang memiliki total amount lebih dari 5.99.

SELECT payment_id, SUM(amount) AS total_amount
FROM payment
GROUP BY payment_id
HAVING SUM(amount) > 5.99;
```



```
--Hitung total amount untuk setiap payment_id dari tabel payment,
--dan tampilkan hanya baris yang memiliki total amount lebih dari 5.99.

SELECT payment_id, SUM(amount) AS total_amount
FROM payment
GROUP BY payment_id
HAVING SUM(amount) > 5.99;
```

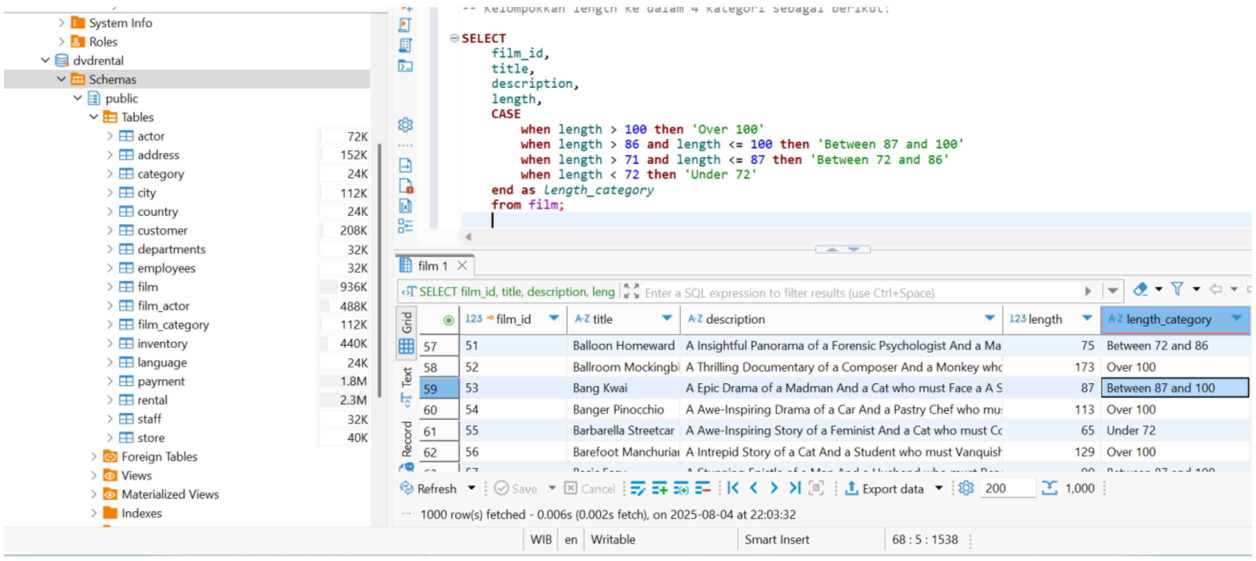
	payment_id	total_amount
1	18,803	7.99
2	31,789	6.99
3	25,886	10.99
4	28,031	9.99
5	23,154	7.99
6	18,474	7.99
7	26,032	9.99

Dari tabel `film`, tampilkan `film_id`, `title`, `description`, dan `length`.
Kelompokkan `length` ke dalam **4 kategori** sebagai berikut:

- Over 100
- Between 87 and 100
- Between 72 and 86
- Under 72

Penamaan kategori dapat ditentukan sendiri. Gunakan klausa `CASE WHEN` atau `BETWEEN`.

```
SELECT
    film_id,
    title,
    description,
    length,
    CASE
       when length > 100 then 'Over 100'
       when length > 86 and length <= 100 then 'Between 87 and 100'
       when length > 71 and length <= 87 then 'Between 72 and 86'
       when length < 72 then 'Under 72'
    end as length_category
    from film;
```



```
-- Kelompokkan length ke dalam 4 kategori sebagai berikut:

SELECT
    film_id,
    title,
    description,
    length,
    CASE
        when length > 100 then 'Over 100'
        when length > 86 and length <= 100 then 'Between 87 and 100'
        when length > 71 and length <= 87 then 'Between 72 and 86'
        when length < 72 then 'Under 72'
    end as length_category
    from film;
```

SELECT film_id, title, description, leng

	film_id	title	description	length	length_category
57	51	Balloon Homeward	A Insightful Panorama of a Forensic Psychologist And a Ma	75	Between 72 and 86
58	52	Ballroom Mockingbi	A Thrilling Documentary of a Composer And a Monkey whc	173	Over 100
59	53	Bang Kwai	A Epic Drama of a Madman And a Cat who must Face a A S	87	Between 87 and 100
60	54	Banger Pinocchio	A Awe-Inspiring Drama of a Car And a Pastry Chef who mu:	113	Over 100
61	55	Barbarella Streetcar	A Awe-Inspiring Story of a Feminist And a Cat who must Cc	65	Under 72
62	56	Barefoot Manchuriai	A Intrepid Story of a Cat And a Student who must Vanquish	129	Over 100

1000 row(s) fetched - 0.006s (0.002s fetch), on 2025-08-04 at 22:03:32

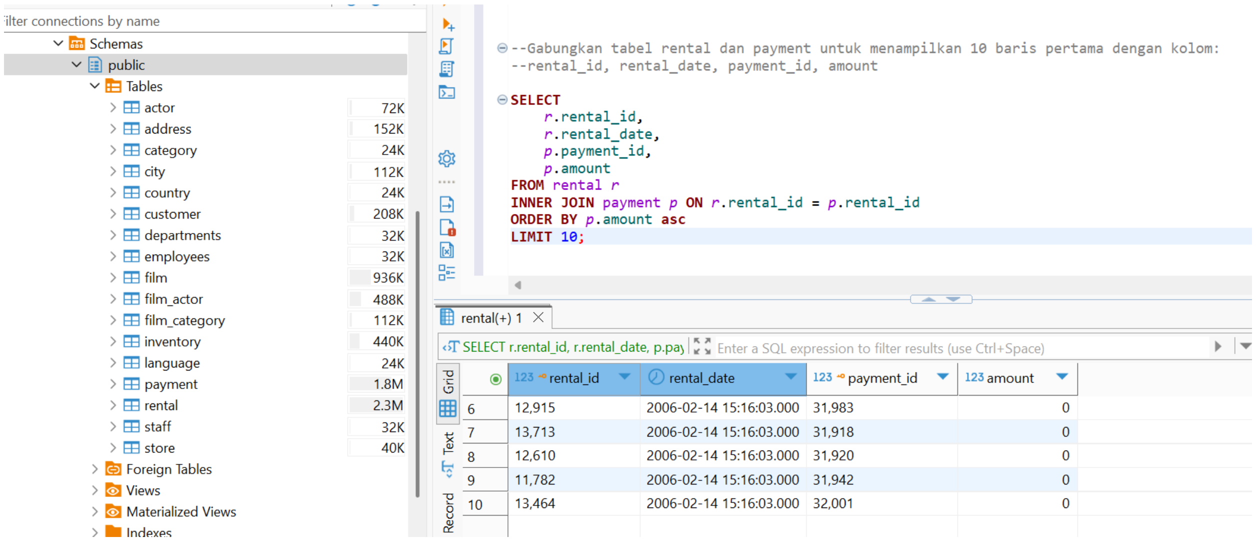
Gabungkan tabel `rental` dan `payment` untuk menampilkan **10 baris pertama** dengan kolom:

- `rental_id`
- `rental_date`
- `payment_id`
- `amount`

Urutkan hasil berdasarkan `amount` secara **ascending**.

```
--Gabungkan tabel rental dan payment untuk menampilkan 10 baris pertama dengan kolom:
--rental_id, rental_date, payment_id, amount

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 p.amount asc
LIMIT 10;
```



ilter connections by name

- Schemas
  - public
    - Tables

actor	72K
address	152K
category	24K
city	112K
country	24K
customer	208K
departments	32K
employees	32K
film	936K
film_actor	488K
film_category	112K
inventory	440K
language	24K
payment	1.8M
rental	2.3M
staff	32K
store	40K

    - Foreign Tables
    - Views
    - Materialized Views
    - Indexes

```
--Gabungkan tabel rental dan payment untuk menampilkan 10 baris pertama dengan kolom:
--rental_id, rental_date, payment_id, amount

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 p.amount asc
LIMIT 10;
```

rental(+) 1

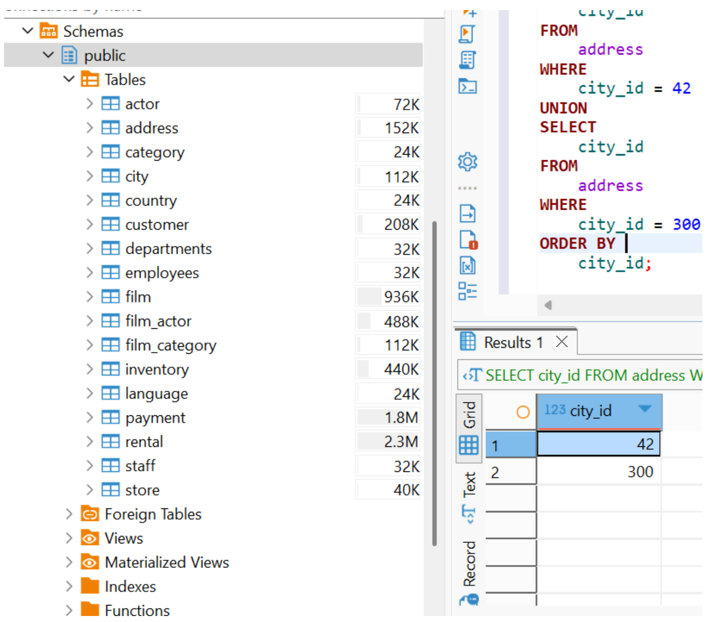
SELECT r.rental_id, r.rental_date, p.pay | Enter a SQL expression to filter results (use Ctrl+Space)

	rental_id	rental_date	payment_id	amount
6	12,915	2006-02-14 15:16:03.000	31,983	0
7	13,713	2006-02-14 15:16:03.000	31,918	0
8	12,610	2006-02-14 15:16:03.000	31,920	0
9	11,782	2006-02-14 15:16:03.000	31,942	0
10	13,464	2006-02-14 15:16:03.000	32,001	0

Dari tabel `address`, **gabungkan seluruh kolom** dari alamat yang memiliki `city_id = 42` dan `city_id = 300` menggunakan **`UNION`**.

```
--Dari tabel address, gabungkan seluruh kolom dari alamat yang memiliki
--city_id = 42 dan city_id = 300 menggunakan UNION.

SELECT
        city_id
FROM
        address
WHERE
        city_id = 42
UNION
SELECT
        city_id
FROM
        address
WHERE
        city_id = 300
ORDER BY
        city_id;
```



- Schemas
  - public
    - Tables

actor	72K
address	152K
category	24K
city	112K
country	24K
customer	208K
departments	32K
employees	32K
film	936K
film_actor	488K
film_category	112K
inventory	440K
language	24K
payment	1.8M
rental	2.3M
staff	32K
store	40K

    - Foreign Tables
    - Views
    - Materialized Views
    - Indexes
    - Functions

```sql
FROM
    address
WHERE
    city_id = 42
UNION
SELECT
    city_id
FROM
    address
WHERE
    city_id = 300
ORDER BY
    city_id;
```

Results 1

SELECT city_id FROM address W

	city_id
1	42
2	300