Pada skema dvdrental

1. Tunjukan first_name dan last_name actor yang memiliki first_name Jennifer, Nick, dan Ed.

```sql
-- SELECT with WHERE
SELECT first_name, last_name
FROM actor
WHERE first_name IN ('Jennifer', 'Nick', 'Ed');
```
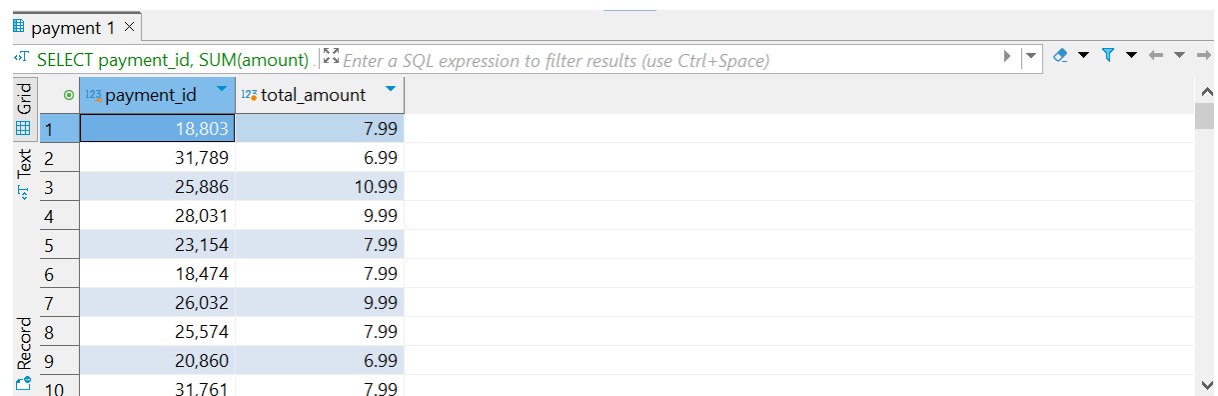
	first_name	last_name
1	Nick	Wahlberg
2	Ed	Chase
3	Jennifer	Davis
4	Nick	Stallone
5	Ed	Mansfield
6	Nick	Degeneres
7	Ed	Guiness

2. Hitung Total Amount untuk setiap payment_id yang Total Amount-nya lebih dari 5.99

```sql
-- SELECT with HAVING
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
8	25,574	7.99
9	20,860	6.99
10	31,761	7.99

3. Tunjukkan film.id, film.title, film.description and film_length. Kelompokkan film.length ke dalam 4 categories(over 100, 87-100, 72-86 and under 72).

```sql
-- CASE
SELECT film_id, title, description, length,
CASE
    WHEN length > 100 THEN 'Over 100 minutes'
    WHEN length BETWEEN 87 AND 100 THEN '87-100 minutes'
    WHEN length BETWEEN 72 AND 86 THEN '72-86 minutes'
    ELSE 'Under 72 minutes'
END AS duration
FROM
    film;
```

4. Dari tabel rental dan payment, tunjukkan 10 baris rental_id, rental_date, payment_id, dan amount. Ordered by amount in ascending order.

```sql
-- JOIN
SELECT r.rental_id, r.rental_date, p.payment_id, p.amount
FROM rental r
JOIN payment p ON r.rental_id = p.rental_id
ORDER BY p.amount ASC
LIMIT 10;
```


5. Gabungkan address (seluruh kolom) yang memiliki city_id = 42 dengan city_id=300. Gunakan UNION

```sql
-- UNION
SELECT *
FROM address
WHERE city_id = 42
UNION
SELECT *
FROM address
WHERE city_id = 300;
```