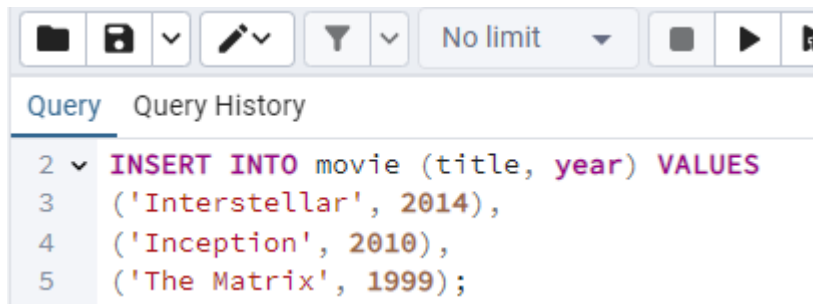


## Part 2

### Insert Data

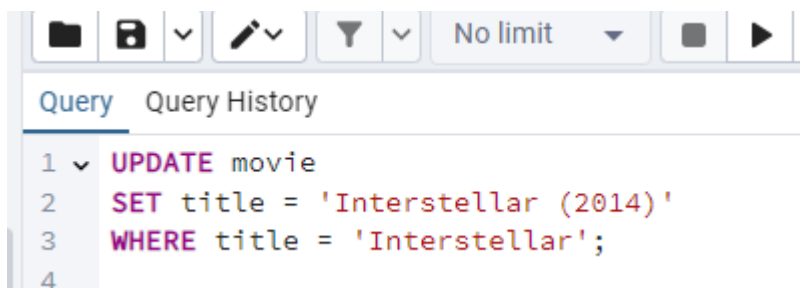


The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, editing, and execution. Below the toolbar are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query in a text editor. The query is an INSERT statement for a table named 'movie', inserting three rows of data: ('Interstellar', 2014), ('Inception', 2010), and ('The Matrix', 1999).

```
2 INSERT INTO movie (title, year) VALUES
3 ('Interstellar', 2014),
4 ('Inception', 2010),
5 ('The Matrix', 1999);
```

Kata kunci INSERT INTO digunakan untuk menambahkan satu atau lebih baris data ke dalam sebuah tabel dalam basis data.

### Update & Where Data

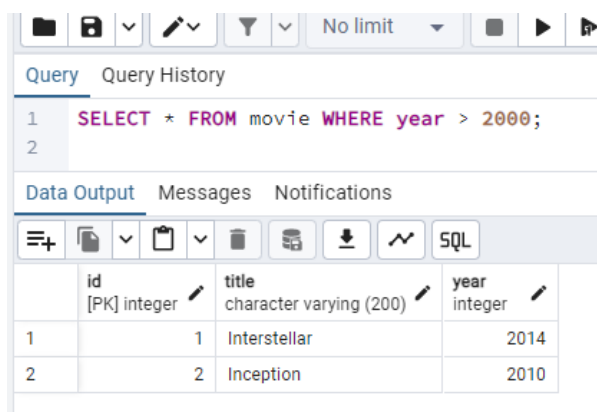


The screenshot shows a SQL IDE interface similar to the previous one. The 'Query' tab is active, displaying an UPDATE query. The query updates the 'title' column of the 'movie' table to 'Interstellar (2014)' for rows where the 'title' is 'Interstellar'.

```
1 UPDATE movie
2 SET title = 'Interstellar (2014)'
3 WHERE title = 'Interstellar';
4
```

Command ini digunakan untuk mengubah judul film "Interstellar" menjadi "Interstellar (2014)", hanya jika judulnya sama persis dengan 'Interstellar'.

### Select



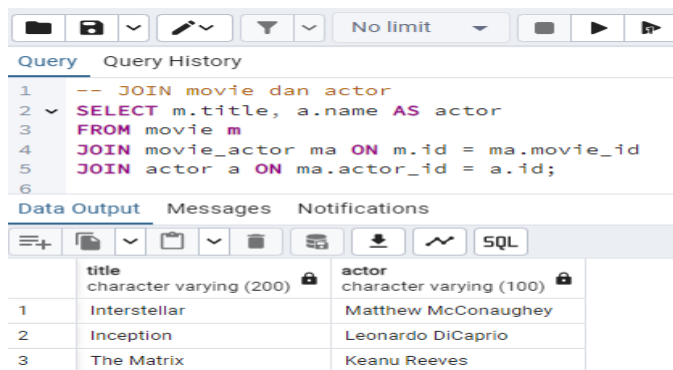
The screenshot shows a SQL IDE interface. The 'Query' tab is active, displaying a SELECT query. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the results of the query. The table has three columns: 'id' (integer, primary key), 'title' (character varying (200)), and 'year' (integer). The results show two rows: one for 'Interstellar' (2014) and one for 'Inception' (2010).

```
1 SELECT * FROM movie WHERE year > 2000;
2
```

	id [PK] integer	title character varying (200)	year integer
1	1	Interstellar	2014
2	2	Inception	2010

Perintah SELECT digunakan untuk mengambil (menampilkan) data dari satu atau lebih tabel dalam basis data.

## Join



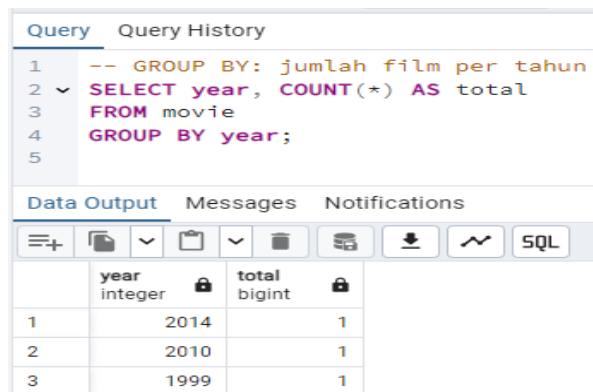
The screenshot shows a SQL query editor with a query window and a data output window. The query is a JOIN between a movie table and an actor table. The output window shows the results of the query, which are three rows of movie titles and actor names.

```
1 -- JOIN movie dan actor
2 SELECT m.title, a.name AS actor
3 FROM movie m
4 JOIN movie_actor ma ON m.id = ma.movie_id
5 JOIN actor a ON ma.actor_id = a.id;
```

	title character varying (200)	actor character varying (100)
1	Interstellar	Matthew McConaughey
2	Inception	Leonardo DiCaprio
3	The Matrix	Keanu Reeves

Menampilkan daftar judul film beserta nama aktornya

## Group by



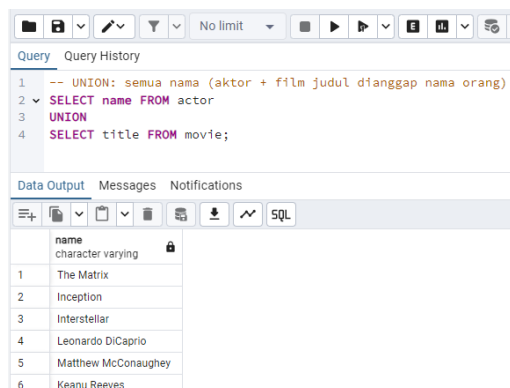
The screenshot shows a SQL query editor with a query window and a data output window. The query is a GROUP BY query that counts the number of movies for each year. The output window shows the results of the query, which are three rows of years and their corresponding counts.

```
1 -- GROUP BY: jumlah film per tahun
2 SELECT year, COUNT(*) AS total
3 FROM movie
4 GROUP BY year;
```

	year integer	total bigint
1	2014	1
2	2010	1
3	1999	1

Mengelompokkan film berdasarkan tahun dan menghitung jumlah film pada setiap tahun.

## Union



The screenshot shows a SQL query editor with a query window and a data output window. The query is a UNION query that combines the names of actors and movie titles. The output window shows the results of the query, which are six rows of names and titles.

```
1 -- UNION: semua nama (aktor + film judul dianggap nama orang)
2 SELECT name FROM actor
3 UNION
4 SELECT title FROM movie;
```

	name character varying
1	The Matrix
2	Inception
3	Interstellar
4	Leonardo DiCaprio
5	Matthew McConaughey
6	Keanu Reeves

Menggabungkan data dari dua kolom yang tipe datanya sama (di sini, VARCHAR)