

Assignment 5: DVD Rental Queries

This assignment focuses on practicing SQL query operations using the **DVDRental sample database**.

The tasks involve filtering data with WHERE conditions, sorting results using ORDER BY, and limiting query outputs with LIMIT and OFFSET. Additionally, pattern matching with the LIKE operator and conditional filtering based on specific column values are applied.

These queries help strengthen understanding of data filtering, sorting logic, and result set control in SQL, which are essential skills for working with relational databases.

Query 1:

Description: Lists the top 5 longest films from the film table where the title ends with the letter 'n'. This query demonstrates filtering with pattern matching and ordering results by film length in descending order.

The screenshot shows a PostgreSQL query editor interface. The query editor displays the following SQL query:

```
1 SELECT * FROM film
2 WHERE title LIKE '%n'
3 ORDER BY length DESC
4 LIMIT 5;
```

Below the query editor, the results are displayed in a table. The table has 12 columns: title, description, release_year, language_id, rental_duration, rental_rate, length, replacement_cost, rating, mpaa_rating, and last_update. The results are sorted by length in descending order, showing the top 5 films.

	title	description	release_year	language_id	rental_duration	rental_rate	length	replacement_cost	rating	mpaa_rating	last_update
1	Soldiers Evolution	A Lacklustur...	2006	1	7	4.99	185	27.99	R		2013-05-26 14:04:50
2	Sorority Queen	A Fast-Pace...	2006	1	6	0.99	184	17.99	NC-17		2013-05-26 14:04:50
3	King Evolution	A Action-Pac...	2006	1	3	4.99	184	24.99	NC-17		2013-05-26 14:04:50
4	Frontier Cabin	A Emotional ...	2006	1	6	4.99	183	14.99	PG-13		2013-05-26 14:04:50
5	Wife Turn	A Awe-Inspir...	2006	1	3	4.99	183	27.99	NC-17		2013-05-26 14:04:50

At the bottom of the interface, the status bar shows: Total rows: 5 Query complete 00:00:00.480 CRLF Ln 4, Col 9.

Query 2:

Description: This query shows films from the film table whose titles end with the letter 'n', selecting the second group of 5 shortest films. The query uses pattern matching, sorts films by length in ascending order, and skips the first 5 results.

The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT * FROM film
2 WHERE title LIKE '%n'
3 ORDER BY length ASC
4 OFFSET 5
5 LIMIT 5;
```

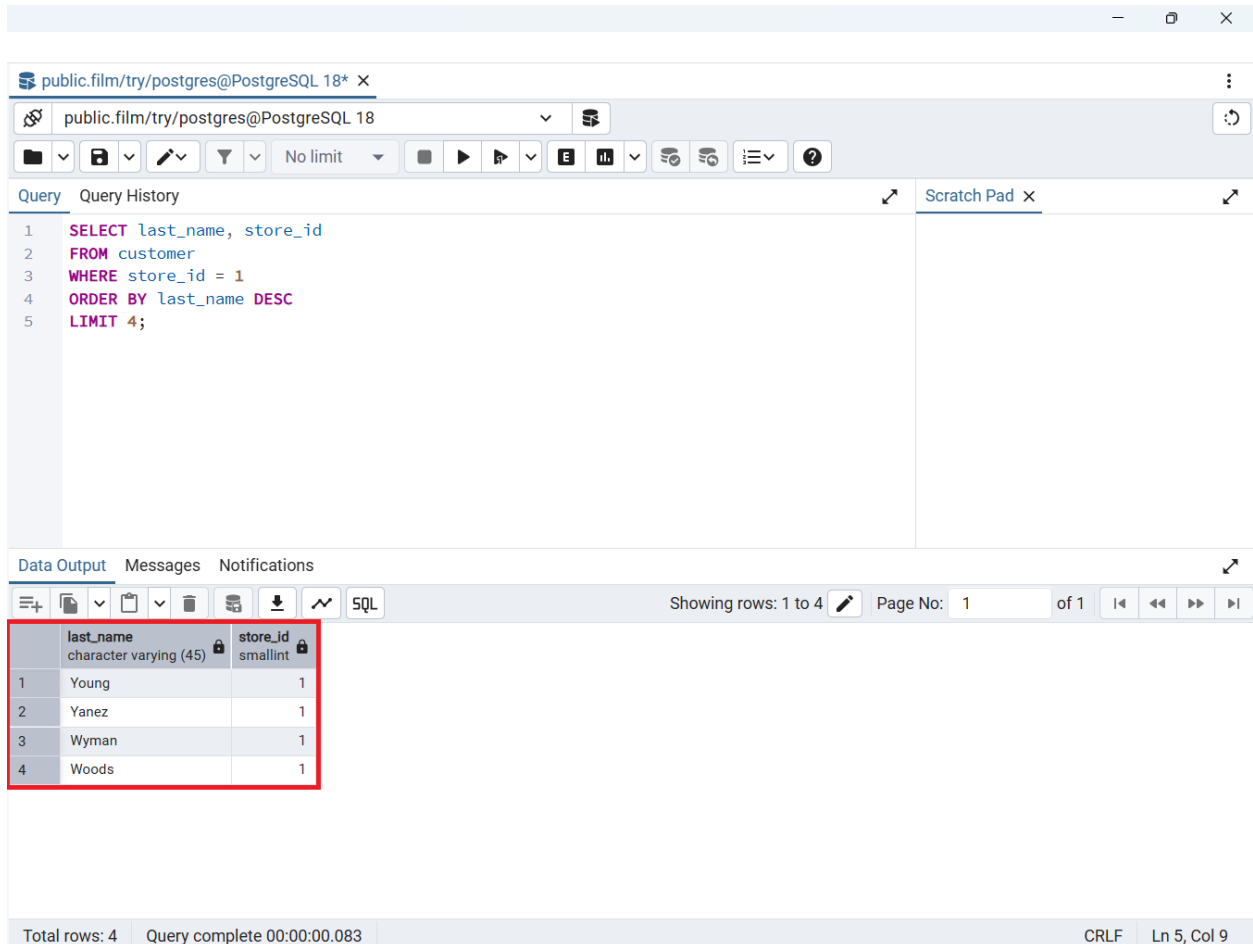
The results are displayed in a table with the following columns: film_id, title, description, release_year, language_id, rental_duration, rental_rate, length, replacement_cost, rating, and last_update. The first five rows of the results are highlighted with a red box.

	film_id	title	description	release_year	language_id	rental_duration	rental_rate	length	replacement_cost	rating	last_update
1	481	Jekyll Frogmen	A Fa...	2006	1	4	2.99	58	22.99	PG	2013-0
2	214	Daughter Madigan	A B...	2006	1	3	4.99	59	13.99	PG-13	2013-0
3	743	Room Roman	A A...	2006	1	7	0.99	60	27.99	PG	2013-0
4	114	Camelot Vacation	A To...	2006	1	3	0.99	61	26.99	NC-17	2013-0
5	77	Birds Perditiion	A B...	2006	1	5	4.99	61	15.99	G	2013-0

Total rows: 5 Query complete 00:00:00.160 CRLF Ln 6, Col 1

Query 3:

Description: Query 3 shows the first 4 customers from store_id = 1, ordered by their last names in descending order. This query demonstrates filtering by a specific column and sorting results alphabetically in reverse order.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT last_name, store_id
2 FROM customer
3 WHERE store_id = 1
4 ORDER BY last_name DESC
5 LIMIT 4;
```

The results are displayed in a table with the following data:

	last_name character varying (45)	store_id smallint
1	Young	1
2	Yanez	1
3	Wyman	1
4	Woods	1

The interface also shows a status bar at the bottom indicating "Total rows: 4", "Query complete 00:00:00.083", and "CRLF Ln 5, Col 9".