

WHERE > AND < →

```
SELECT *  
FROM actor  
WHERE first_name = 'Penelope' AND last_name = 'Monroe' ;
```

```
WHERE first_name = 'Penelope' OR first_name = 'Bob' ;
```

WHERE NOT → CLAUSE

```
WHERE NOT (rental_rate = 4.99 OR rental_rate = 2.99)
```

Homework-1

- 1- Sort the data in the title and description columns in the first film table.

```
SELECT title, description FROM film;
```

- 2- Sort the data in all columns in the movie table with the film length greater than 60 AND less than 75.

```
SELECT * FROM film  
WHERE length > 60 and length < 75;
```

- 3- Sort the data in all columns in the film table with rental_rate 0.99 AND replacement_cost 12.99 OR 28.99.

```
SELECT * from film  
WHERE rental_rate = 0.99  
AND replacement_cost = 28.99;
```

- 4- What is the value in the last_name column of the customer whose value is 'Mary' in the first_name column of the customer table?

```
SELECT first_name, last_name FROM customer  
WHERE first_name = 'Mary';
```

- 5- Sort the data in the movie table whose length is NOT greater than 50, but whose rental_rate is NOT 2.99 or 4.99.

```
SELECT * FROM film  
WHERE NOT (length < 50)  
AND NOT (rental_rate = 2.99 OR rental_rate = 4.99);
```

BETWEEN AND SYNTAX

```
SELECT <column_name>, <column_name>, ...  
FROM <table_name>  
WHERE <condition>;
```

```
-----  
SELECT *  
FROM film  
WHERE length BETWEEN 100 AND 140;
```

```
-----  
-- WHERE length > 100 AND length < 140
```

IN SYNTAX

```
SELECT *  
FROM film  
WHERE length IN (30,60,90,120);
```

We can also use the NOT IN construct for values out of the list.

Homework-2

- 1- Sort all column data in the film table provided that the replacement cost value is greater than 12.99, equal and less than 16.99 (Use BETWEEN - AND structure.)

```
SELECT * FROM film  
WHERE replacement_cost BETWEEN 12.98 AND 16.98;  
--12.99 and 16.99 included
```

- 2- Sort the data in the first_name and last_name columns in the actor table provided that first_name is the values 'Penelope' or 'Nick' or 'Ed'. (Use the IN operator.)

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

- 3- Sort the data in all columns in the film table with rental_rate 0.99, 2.99, 4.99 AND replacement_cost 12.99, 15.99, 28.99. (Use the IN operator.)

```
SELECT * FROM film  
WHERE rental_rate IN (0.99, 2.99, 4.99)  
AND replacement_cost IN (12.99, 15.99, 28.99);
```

LIKE / NOT LIKE

For multi character use '%' but for single character use '_' symbol

```
SELECT *  
FROM actor  
WHERE first_name LIKE 'P%';
```

```
SELECT *  
FROM actor  
WHERE first_name -- 'P%';
```

Both uses are same 😊

```
--* → ILIKE  
--  → LIKE  
!-- → NOT LIKE  
!--* → NOT ILIKE
```

NOTE : The ILIKE operator is the case - insensitive version of the LIKE operator!

Homework-3

- 1- List the country names in the country column of the country table, starting with the 'A' character and ending with the 'a' character.

```
SELECT * FROM country  
WHERE country ILIKE 'A%a';
```

- 2- List the country names in the country column of the country table, consisting of at least 6 characters and ending with the 'n' character.

```
SELECT country FROM country  
WHERE country ILIKE '____%n';
```

- 3- In the title column of the film table, list the movie names containing at least 4 'T' characters, regardless of upper- or lower-case letters.

```
SELECT title FROM film  
WHERE title ILIKE '%T%T%T%T%';
```

- 4- From the data in all the columns in the film table, sort the data that starts with the title 'C' character, has a length greater than 90 and a rental_rate of 2.99.

```
SELECT title, length, rental_rate FROM film  
WHERE title LIKE 'C%' AND  
length > 90 AND rental_rate = 2.99;
```

SELECT DISTINCT SYNTAX

```
SELECT DISTINCT <columnName>, <columnName>, ...  
FROM <tableName>;
```

SELECT COUNT SYNTAX

```
SELECT COUNT(*)  
FROM actor  
WHERE first_name = 'Penelope';  
MORE
```

```
SELECT COUNT(DISTINCT <columnName>)  
FROM actor
```

Homework-4

- 1- Sort the different values in the replacement cost column in the film table.

```
SELECT DISTINCT replacement_cost FROM film;
```

- 2- How many different data are there in the replacement cost column in the film table?

```
SELECT COUNT(DISTINCT replacement_cost) FROM film;
```

- 3- How many of the film titles in the film table start with the character T and at the same time the rating is equal to 'G'?

```
SELECT COUNT(title) FROM film  
WHERE title LIKE 'T%' AND  
rating = 'G';
```

- 4- How many of the country names (country) in the country table consist of 5 characters?

```
SELECT COUNT(country) FROM country  
WHERE country LIKE '_____';
```

- 5- How many of the city names in the city table end with the character 'R' or r?

```
SELECT COUNT(city) FROM city  
WHERE city ILIKE '%r';
```

ORDER BY SYNTAX

```
SELECT <columnName>, <columnName>, ...  
FROM <tableName>  
ORDER BY <columnName>, <columnName>, ... ASC|DESC;
```

ASC → INCREASING

DESC → DECREASING

```
SELECT *  
FROM film  
WHERE title LIKE 'A%'  
ORDER BY title ASC length DESC;
```

LIMIT

```
SELECT *  
FROM film  
WHERE title LIKE 'B%'  
ORDER BY length DESC  
LIMIT 10;
```

→ Gives the 10 longest films.

OFFSET

```
SELECT *  
FROM film  
WHERE title LIKE 'B%'  
ORDER BY length DESC  
OFFSET 6  
LIMIT 4;
```

→ Skips the 6 longest film and gives other 4 film.

Homework-5

- 1- List the 5 longest (length) films in the film table and the film title (title) ends with the 'n' character.

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

- 2- List the shortest (length) second (6,7,8,9,10) 5 films (6,7,8,9,10) in the film table and the film title ends with the 'n' character.

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

- 3- Sort the first 4 data, provided that store_id is 1 in the descending order according to the last_name column in the customer table.

```
SELECT * from customer
WHERE store_id = 1
ORDER BY last_name DESC
LIMIT 4;
```

Aggregate Functions - MIN, MAX, SUM, AVG

```
SELECT AVG(length)
FROM film;
```

Homework-6

- 1- What is the average of the values in the rental_rate column in the film table?

```
SELECT AVG(rental_rate) FROM film;
```

- 2- How many of the movies in the film table start with the character 'C'?

```
SELECT COUNT(title) FROM film
WHERE title LIKE 'C%';
```

- 3- Among the movies in the film table, how many minutes is the longest (length) film with a rental_rate equal to 0.99?

```
SELECT MAX(length) FROM film
WHERE rental_rate = 0.99;
```

- 4- How many different replacement_cost values are there for the films longer than 150 minutes in the film table?

```
SELECT COUNT(replacement_cost) FROM film
WHERE length > 150 ;
```