

# SQL Basics Part 3 Notes

Team Treehouse

June 4, 2020

## 1 Searching Tables with 'WHERE'

- WHERE clause
  - **Syntax:** *SELECT columns FROM table name WHERE condition;*
  - **Syntax (Condition):** *Columns Operator Value*
- Equality Operator
  - **Syntax:** *SELECT columns FROM table name WHERE column name = value;*

### Examples:

```
1  SELECT * FROM contacts WHERE first_name = "Andrew";
2
3
4  SELECT first_name, email FROM users WHERE last_name = "Chalkley";
5
6
7  SELECT name AS "Product Name" FROM products WHERE stock_count = 0;
8
9
10 SELECT title "Book Title" FROM books WHERE year_published = 1999;
11
```

- Inequality Operator
  - **Syntax:** *SELECT columns FROM table name WHERE column name != value;*

### Examples:

```
1  SELECT * FROM contacts WHERE first_name != "Kenneth";
2
3
4  SELECT first_name, email FROM users WHERE last_name != "L:one";
5
6
7  SELECT name AS "Product Name" FROM products WHERE stock_count !=
8  0;
9
10 SELECT title "Book Title" FROM books WHERE year_published != 2015;
11
```

- Greater than/ Less than Operator
  - **Syntax (less than):** SELECT *columns* FROM *table name* WHERE *column name* < *value*;
  - **Syntax (greater than):** SELECT *columns* FROM *table name* WHERE *column name* > *value*;
- Cheat Sheet: [Link](#)

## 2 Exercise 1

- Solution included in *exercise\_1.sql*

## 3 Filtering by Comparing Values

- **Syntax (Less than):** SELECT *columns* FROM *table name* WHERE *column name* < *value*;
- **Syntax (Less than or equal):** SELECT *columns* FROM *table name* WHERE *column name* <= *value*;
- **Syntax (Greater than):** SELECT *columns* FROM *table name* WHERE *column name* > *value*;
- **Syntax (Greater than or equal):** SELECT *columns* FROM *table name* WHERE *column name* >= *value*;

### Example:

```
1  SELECT first_name, last_name FROM users WHERE date_of_birth < '
2  1998-12-01';
3
4  SELECT title AS "Book Title", author AS Author FROM books WHERE
5  year_released <= 2015;
6
7  SELECT name, description FROM products WHERE price > 9.99;
8
9
10 SELECT title FROM movies WHERE release_year >= 2000;
11
```

## 4 Exercise 2

- Solution included in *exercise\_2.sql*

## 5 Filtering on More than One Condition

- Is used when filtering with multiple conditions
- Can be done using *AND* and/or *OR* operator
- **Syntax (AND):** `SELECT columns FROM table name WHERE (condition 1) AND (condition 2) ...;`
- **Syntax (OR):** `SELECT columns FROM table name WHERE (condition 1) OR (condition 2) ...;`

### Examples:

```
1  SELECT username FROM users WHERE last_name = "Chalkley" AND
2  first_name = "Andrew";
3
4  SELECT * FROM products WHERE category = "Games Consoles" AND price
5  < 400;
6
7  SELECT * FROM movies WHERE title = "The Matrix" OR title = "The
8  Matrix Reloaded" OR title = "The Matrix Revolutions";
```

```
9
10  SELECT country FROM countries WHERE population < 1000000 OR
11  population > 100000000;
```

## 6 Exercise 3

- Solution included in *exercise\_3.sql*

## 7 Filtering By Dates

- Is done using comparison operators (same as part 3).
- **Syntax (Less than):** `SELECT columns FROM table name WHERE column name < value;`
- **Syntax (Less than or equal):** `SELECT columns FROM table name WHERE column name <= value;`
- **Syntax (Greater than):** `SELECT columns FROM table name WHERE column name > value;`
- **Syntax (Greater than or equal):** `SELECT columns FROM table name WHERE column name >= value;`

### Examples:

```
1  SELECT first_name, last_name FROM users WHERE date_of_birth < '
2  1998-12-01';
3
4  SELECT title AS "Book Title", author AS Author FROM books WHERE
5  year_released <= 2015;
6
7  SELECT name, description FROM products WHERE price > 9.99;
8
9
10 SELECT title FROM movies WHERE release_year >= 2000;
11
```

## 8 Exercise 4

- Solution included in *exercise\_4.sql*

## 9 Searching Within a Set of Values

- Returns results with matching sets of values in a columns
- Is similar to Python's *x in [Value1, value2, ....]*
- **Syntax:** `SELECT columns FROM table name WHERE column name IN (value 1, value 2, ...);`
- **Syntax (Negation):** `SELECT columns FROM table name WHERE column name NOT IN (value 1, value 2, ...);`

### Examples:

```
1  SELECT name FROM islands WHERE id IN (4, 8, 15, 16, 23, 42);
2
3
4  SELECT * FROM products WHERE category IN ("eBooks", "Books", "
   Comics");
5
6
7  SELECT title FROM courses WHERE topic IN ("JavaScript", "Databases
   ", "CSS");
8
9
10 SELECT * FROM campaigns WHERE medium IN ("email", "blog", "ppc");
11
12
13 SELECT * FROM products WHERE category NOT IN ("Electronics");
14
15
16 SELECT title FROM courses WHERE topic NOT IN ("SQL", "NoSQL");
17
```

## 10 Exercise 5

- Solution included in *exercise\_5.sql*

## 11 Searching Within a Range of Values

- Returns results between *lesser value* and *greater value*
- **Syntax:** `SELECT columns FROM table name WHERE column name BETWEEN lesser value AND greater value;`

### Examples:

```
1  SELECT * FROM movies WHERE release_year BETWEEN 2000 AND 2010;
2
3
4  SELECT name, description FROM products WHERE price BETWEEN 9.99
   AND 19.99;
5
6
7  SELECT name, appointment_date FROM appointments WHERE
   appointment_date BETWEEN "2015-01-01" AND "2015-01-07";
8
```

## 12 Exercise 6

- Solution included in *exercise\_6.sql*

## 13 Finding Data that Matches a Pattern

- LIKE operator
  - Is used inside of *WHERE* clause to match a pattern
  - **Syntax:** `SELECT columns FROM table name WHERE column name LIKE pattern;`
  - Can be used to make search case insensitive

```
1  SELECT title FROM books WHERE title LIKE "Harry Potter";
2  // returns items like 'Harry potter', 'harry potter'
3
4
```

- LIKE operator with wild card %
  - Works to match zero or more unspecified characters

- works the same as ‘ in regex

```
1  SELECT title FROM books WHERE title LIKE "Harry Potter%Fire";
2  // returns items like 'Harry Potter and Dragon Fire', 'Harry
   Potter and Fire', 'Harry Potter Rising Fire'
3
4  SELECT title FROM movies WHERE title LIKE "Alien%";
5  // Returns items like 'Alien attack', 'Alien', "Alienate"
6
7
8  SELECT * FROM contacts WHERE first_name LIKE "%drew";
9  // Returns items like 'tigerdrew', 'mountaindrew', 'morning drew',
   'andrew'
10
11
12 SELECT * FROM books WHERE title LIKE "%Brief History%";
13 // Returns items like 'Canadian Brief History Channel', 'Brief
   History'
14
```

## 14 Exercise 7

- Solution included in *exercise\_7.sql*

## 15 Filtering Out or Finding Missing Information

- Using IS NULL
  - Is used in WHERE
  - Retrieve rows with information missing.
  - **Syntax:** SELECT *columns* FROM *table name* WHERE *column name* IS NULL;
- Using IS NOT NULL
  - **Syntax:** SELECT *columns* FROM *table name* WHERE *column name* IS NOT NULL;

### Example:

```
1  SELECT address FROM records WHERE address IS NOT NULL;
2
```

## 16 Exercise 8

- Solution included in *exercise\_8.sql*

## 17 Review & Practice with SQL Playgrounds

## 18 Quiz 1

1. Which keyword could you use to rewrite this query in a shorter form?

```
1 SELECT <columns> FROM <table> WHERE <column 1> = <value 1> OR <  
2 column 1> = <value 2> OR <column 1> = <value 3>;
```

- A. ALL
- B. BETWEEN
- C. IN

**Answer: C**

2. Please fill in the correct answer in each blank provided below.

I want to categorize products by price on a website. Cheap is defined by the prices from 0.01 and 9.99. Enter the missing keywords.

```
1 SELECT name, description FROM products WHERE price ____ 0.01 ____  
2 9.99;
```

**Answer: BETWEEN, AND**

3. Imagine you wanted to retrieve all appointments in for the upcoming week. Monday is 7th October 2019 and Friday is 11th October 2019.

Which query is the correct one to use?

- A. SELECT \* FROM appointments WHERE day BETWEEN "2019-10-07" AND "2019-10-11";



B. `SELECT * FROM appointments WHERE day < "2019-10-07" AND day < "2019-10-11";`

**Answer: A**

4. What's missing from this query to find all contacts without a phone number?

```
1 SELECT * FROM contacts WHERE phone _____ NULL;  
2
```

A. IS NOT

B. IS

**Answer: B**

5. Please fill in the correct answer in each blank provided below.

```
1 SELECT <columns> FROM <table> WHERE <column> ____ <value>;  
2
```

**Answer: =**

6. Please fill in the correct answer in each blank provided below.

Enter the inequality operator:

```
1 SELECT <columns> FROM <table> WHERE <column> ____ <value>;  
2
```

**Answer: >**

7. Which operator is the greater than operator?

A. <

B. >

C. <=

D. >=

**Answer: B**