4 - SHOW and SELECT

```
SHOW DATABASES

SHOW TABLES

SHOW COLUMNS FROM customers

SELECT city FROM customers
```

5 - Basic Rules for SQL Statements

```
---------------------------------------

SELECT city FROM customers;

SELECT id FROM customers;

----------------------------------------
```

6 - Getting Multiple Columns

```
SELECT name, zip, country FROM customers

SELECT * country FROM customers // zaznacza wszystkie kolumny
```

7 - DISTINCT and LIMIT

```
SELECT DISTINCT state FROM customers // zwraca unikalne wartości z kolumny state

returns row one time for each value in this column

SELECT id, name FROM customers LIMIT 5

SELECT id, name FROM customers LIMIT 5, 10 // zaczyna od numeru 6 i zwraca kolejne 10 pozycji
```

8 - Sorting Results

```
SELECT name FROM customers ORDER BY name

SELECT name, adress FROM customers ORDER BY id

SELECT state, city, name FROM customers ORDER BY state, name
```

9 - Sort Direction

```
SELECT name, zip FROM customers ORDER BY zip // od najmniejszej wartości do największej

SELECT name, zip FROM customers ORDER BY zip DESC // od największej wartości do najmniejszej

SELECT name, zip FROM customers ORDER BY name DESC // od z do a
```

```
SELECT name, id FROM customers ORDER BY id DESC LIMIT 1 // zwraca wartość z największym id

SELECT name, id FROM customers ORDER BY name  LIMIT 1 // zwraca 1 wartość w kolejności alfabetycznej ( np. abbot)
```

10 - Basic Data Filtering and WHERE

```
SELECT id, name FROM customers WHERE id=54

SELECT id, name FROM customers WHERE id != 54

SELECT id, name FROM customers WHERE id < 8

SELECT id, name FROM customers WHERE id BETWEEN 25 AND 30

SELECT name, state FROM customers WHERE state = 'CA'
```

11 - Advanced Filtering Using AND and OR

```
SELECT name, state, city FROM customers WHERE state = 'CA' AND city = 'Hollywood'

SELECT name, state, city FROM customers WHERE state = 'CA' OR city = 'Hollywood'

SELECT id, name, city FROM customers WHERE id = 1 OR id = 2 AND city = 'Raleigh'

SELECT id, name, city FROM customers WHERE ( id = 1 OR id = 2 ) AND city = 'Raleigh'
```

12 - Are you IN or are you NOT IN?

```
SELECT name, state FROM cusomers WHERE state = 'CA' OR state = 'NC' OR state = 'NY'
// to samo

SELECT name, state FROM customers WHERE state IN ("CA' , 'NY' , 'NC' ) // to samo

SELECT name, state FROM customers WHERE state NOT IN ("CA' , 'NY' , 'NC' ) // wsystkie stany oprócz
```

13 - How Search Engines Work

```
// wszystko, co zaczyna sie od słowa new np: new iPad, new courtains, newspaper

SELECT name FROM items WHERE name LIKE 'new%'

SELECT name FROM items WHERE name LIKE '%computer%' // computer byle gdzie w kol. 'name'

SELECT city FROM cusomers WHERE city LIKE 'h%d' // miasto zaczynajace sie na h i kończące na d
```

14 - More on Wildcards

```
SELECT name FROM items WHERE name LIKE ' _ boxes ' // zwraca 7 boxes, 2 boxes

SELECT name FROM items WHERE name LIKE ' % boxes ' // zwraca 27 boxes, 22 boxes
```

15 - Regular Expressions

```
SELECT name FROM items WHERE name REGEXP ' new ' // wszystkie row co zawierają new w kol. name

SELECT name FROM items WHERE name REGEXP ' .new ' // wszystkie row co zawierają new w kol. name

SELECT name FROM items WHERE name REGEXP ' gold | car ' // wszystko co zawiera gold lub car zawierają new w kol. name

SELECT name FROM items WHERE name REGEXP ' [12345] boxes' // zwraca np. 3 boxes , 5 boxes

SELECT name FROM items WHERE name REGEXP ' [1 - 5] boxes' // zwraca np. 3 boxes , 5 boxes

SELECT name FROM items WHERE name REGEXP ' [^ 12345] boxes' // wszystko oprócz  np. 3 boxes
```

16 - Creating Custom Columns

```
SELECT CONTACT (city,  ' , '    ,state) FROM customers // rezultat: NEW YORK, NY

SELECT CONTACT (city,  ' , '    ,state) AS new_adress FROM customers // nadaje nazwe nowej kolumnie

SELECT name, cost, cost - 1 AS new_cost FROM items // tworzy nową kolumnę o nazwie new_cost z wartością mniejszą o 1
```

17 - Functions

```
SELECT name, UPPER (name) FROM customers

SELECT AVG (cost) FROM items // zwraca średnią cene

SELECT SUM (bids) FROM items
```

18 - More on Aggregate Functions

```
SELECT COUNT (name) FROM items WHERE seller_id = 6

SELECT AVG (cost) FROM items WHERE seller_id = 6

--------------------------------------------------------------------------------
```

```
SELECT count (*) AS item_count,

MAX (cost) AS max,

AVG (cost) AS avg

FROM items WHERE seller_id = 12
```
