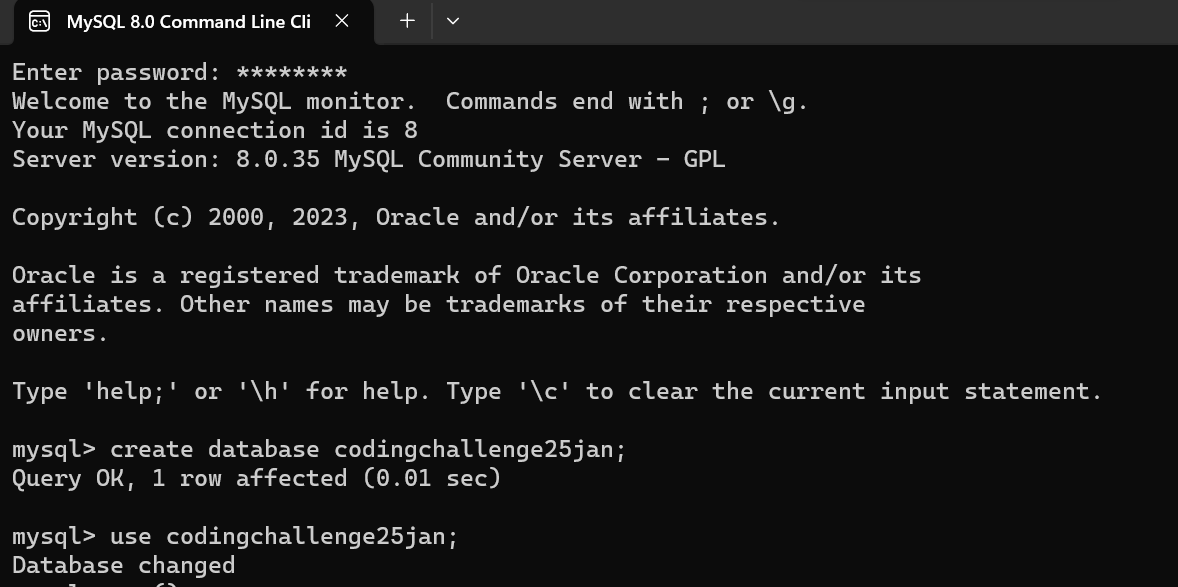
mysql> create database codingchallenge25jan;

Query OK, 1 row affected (0.01 sec)

mysql> use codingchallenge25jan;

Database changed



CREATE TABLE CUSTOMERS (

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

INSERT INTO CUSTOMERS VALUES

(1, 'Ramesh', 32, 'Ahmedabad', 2000.00),

(2, 'Khilan', 25, 'Delhi', 1500.00),

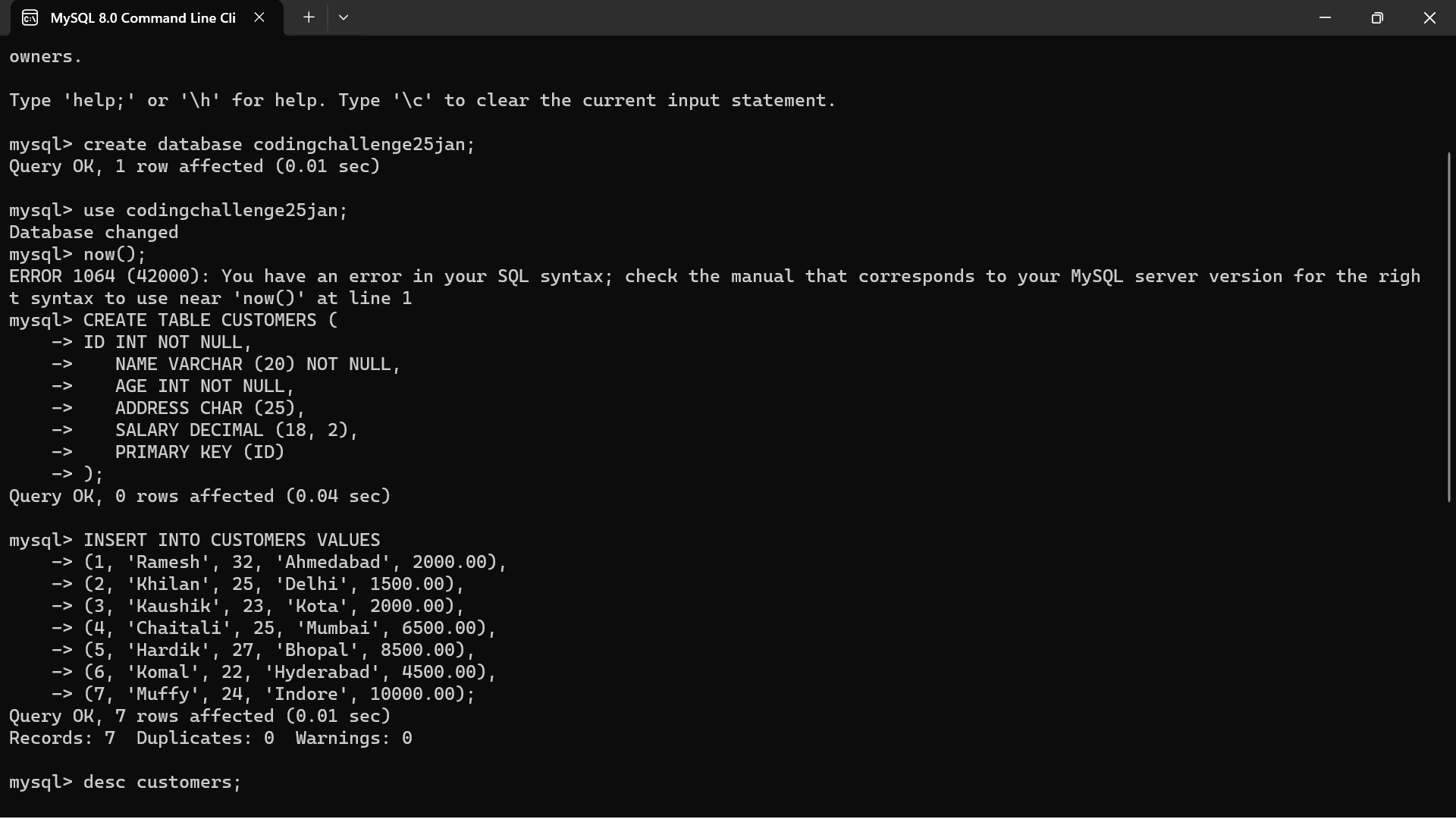
(3, 'Kaushik', 23, 'Kota', 2000.00),

(4, 'Chaitali', 25, 'Mumbai', 6500.00),

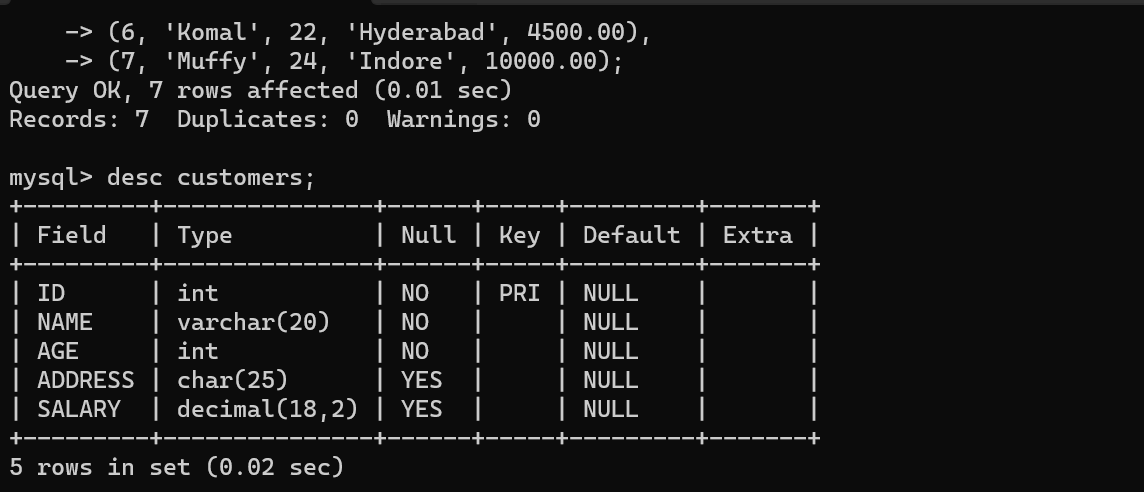
(5, 'Hardik', 27, 'Bhopal', 8500.00),

(6, 'Komal', 22, 'Hyderabad', 4500.00),

(7, 'Muffy', 24, 'Indore', 10000.00);



Desc customers;



CREATE TABLE ORDERS (

OID INT NOT NULL,

DATE VARCHAR (20) NOT NULL,

CUSTOMER\_ID INT NOT NULL,

AMOUNT DECIMAL (18, 2)

);

INSERT INTO ORDERS VALUES

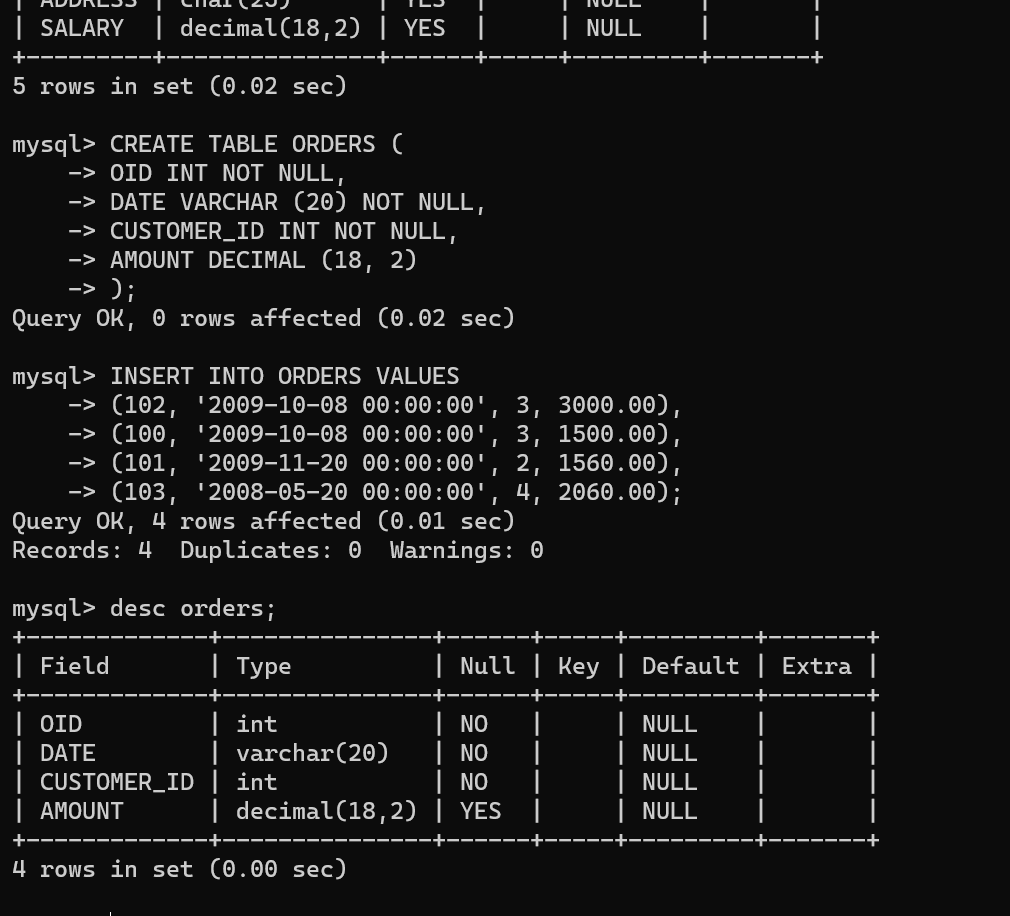
(102, '2009-10-08 00:00:00', 3, 3000.00),

(100, '2009-10-08 00:00:00', 3, 1500.00),

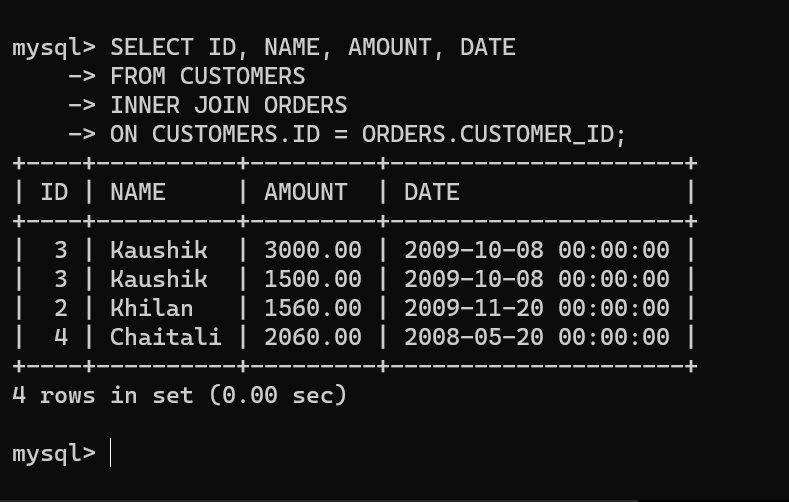
(101, '2009-11-20 00:00:00', 2, 1560.00),

(103, '2008-05-20 00:00:00', 4, 2060.00);

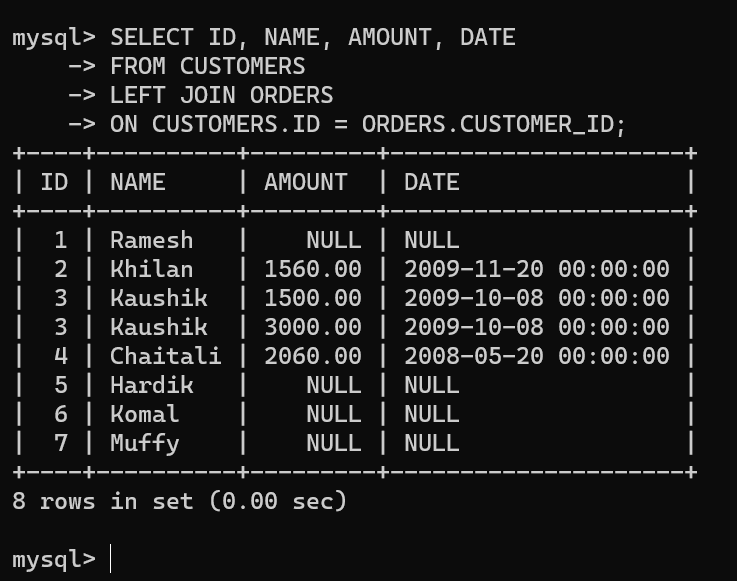
Desc orders;



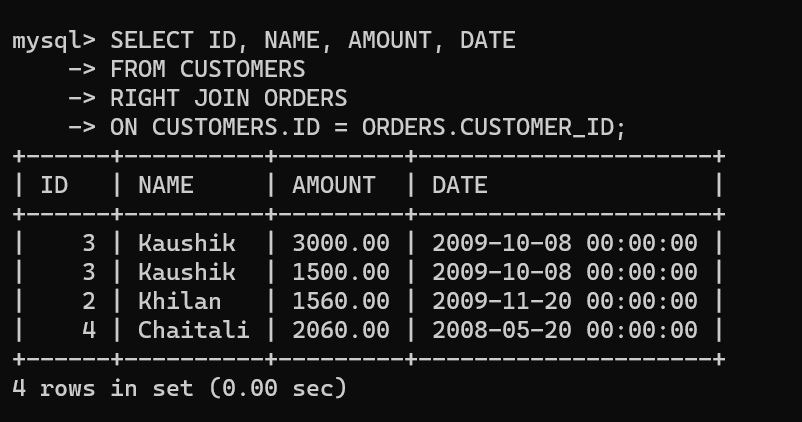
INNER JOIN- combines all tables by taking values which are matching in both tables.



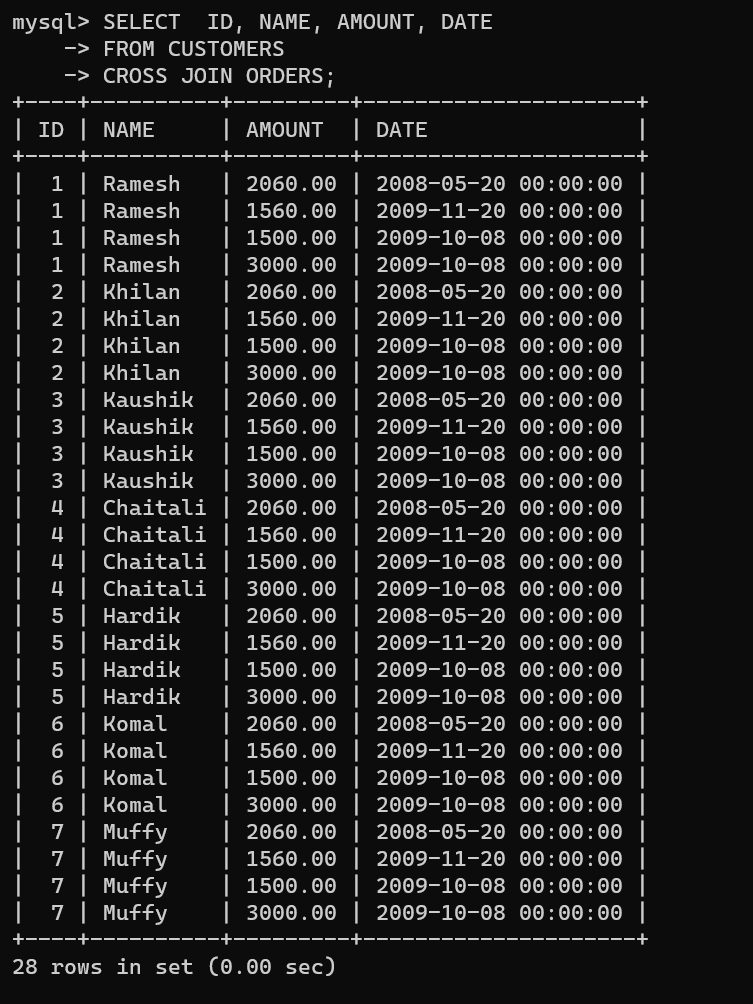
LEFT JOIN- returns all rows from the left table and matching values from right table or NULL in case no value is matching.



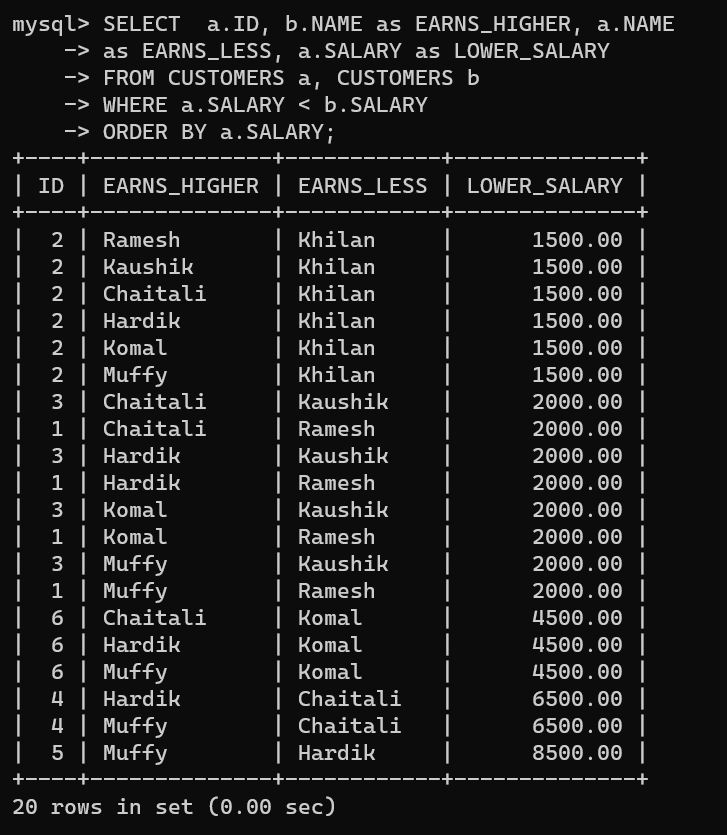
RIGHT JOIN – returns all rows from the right table and matching values from left table or NULL in case no value is matching.



CROSS JOIN- returns the cartesian product of two tables.

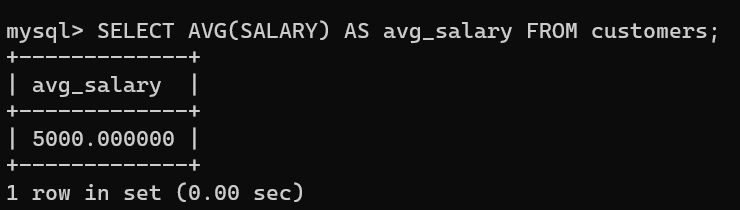


SELF JOIN- combines the records of table with itself

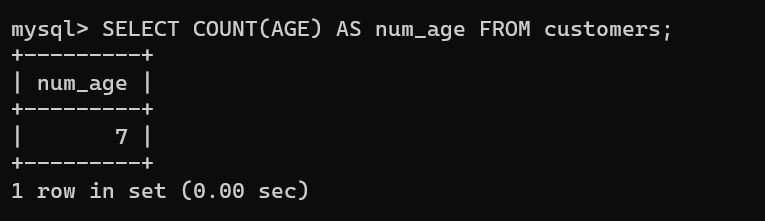


Aggregation

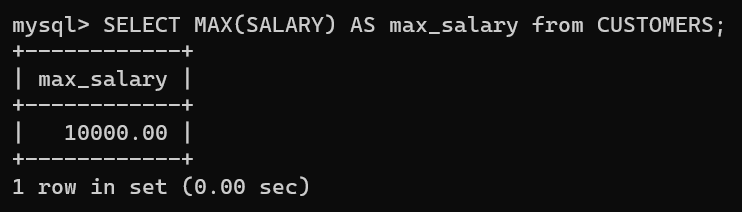
AVG- calculate the arithmetic mean of the selected column



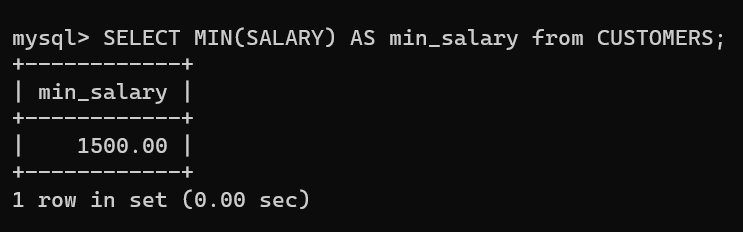
COUNT- count the number of values in the column.



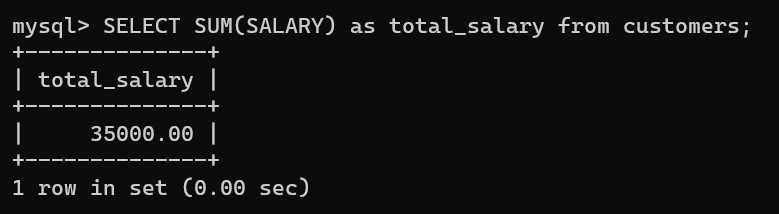
MAX- calculate the maximum value in the column.



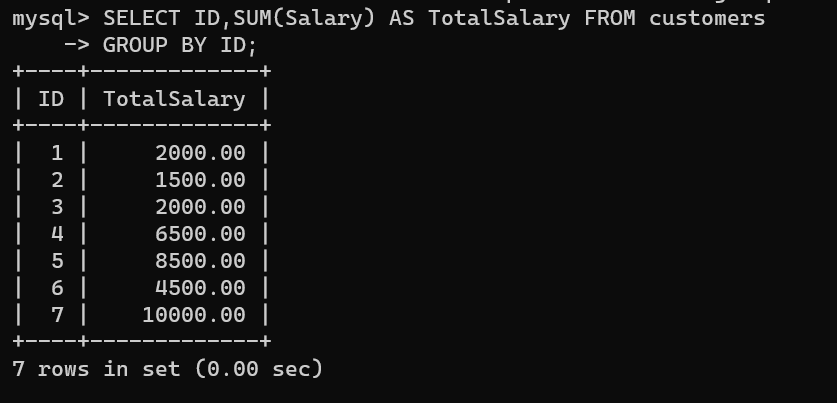
MIN- calculate the minimum value in the column.



SUM- calculates the sum of all the numeric data in a particular column.



SUBTOTAL- refers to sum of similar data but does not indicate final total, often used with group by clause.



OVER AND PARTITION BY- partition by is used for dividing the result into partitions and perform computation. Over is used to tell on which column the computation is working.

