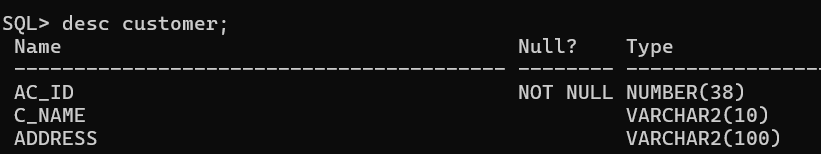
Practical: -8

Aim: - implement outer join, inner join and self-join operations.

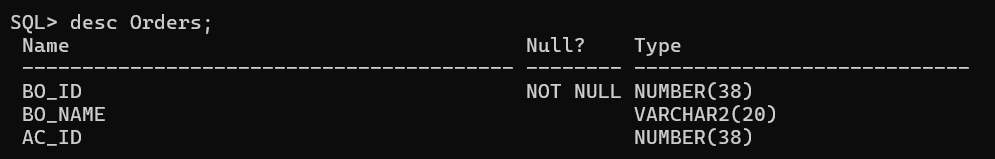
SQL> create table customer (Ac\_id int primary key, c\_name varchar (10), address varchar (100));

Table created.



SQL> create table Orders (Bo\_id int primary key, Bo\_name varchar (20), Ac\_id int, foreign key (Ac\_id) references customer (Ac\_id) on delete cascade);

Table created.



SQL> insert into customer (Ac\_id, c\_name, address) values (1, 'Ramu’, 'New York');

SQL> insert into customer (Ac\_id, c\_name, address) values (2, 'Raju', 'Delhi');

SQL> insert into customer (Ac\_id, c\_name, address) values (3, 'Anshu’, 'ap');

SQL> insert into customer (Ac\_id, c\_name, address) values (4, 'Uma', 'ts');

SQL> insert into Orders (Bo\_id, Bo\_name, Ac\_id) values (101, 'Ramu’, 1);

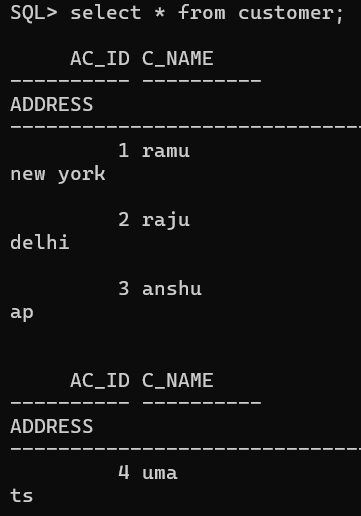
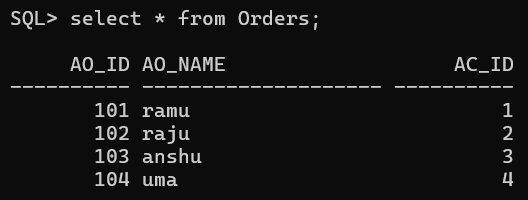
SQL> insert into Orders (Bo\_id, Bo\_name, Ac\_id) values (102, 'raja' ,2);

SQL> insert into Orders (Bo\_id, Bo\_name, Ac\_id) values (103, 'Anshu' ,3);

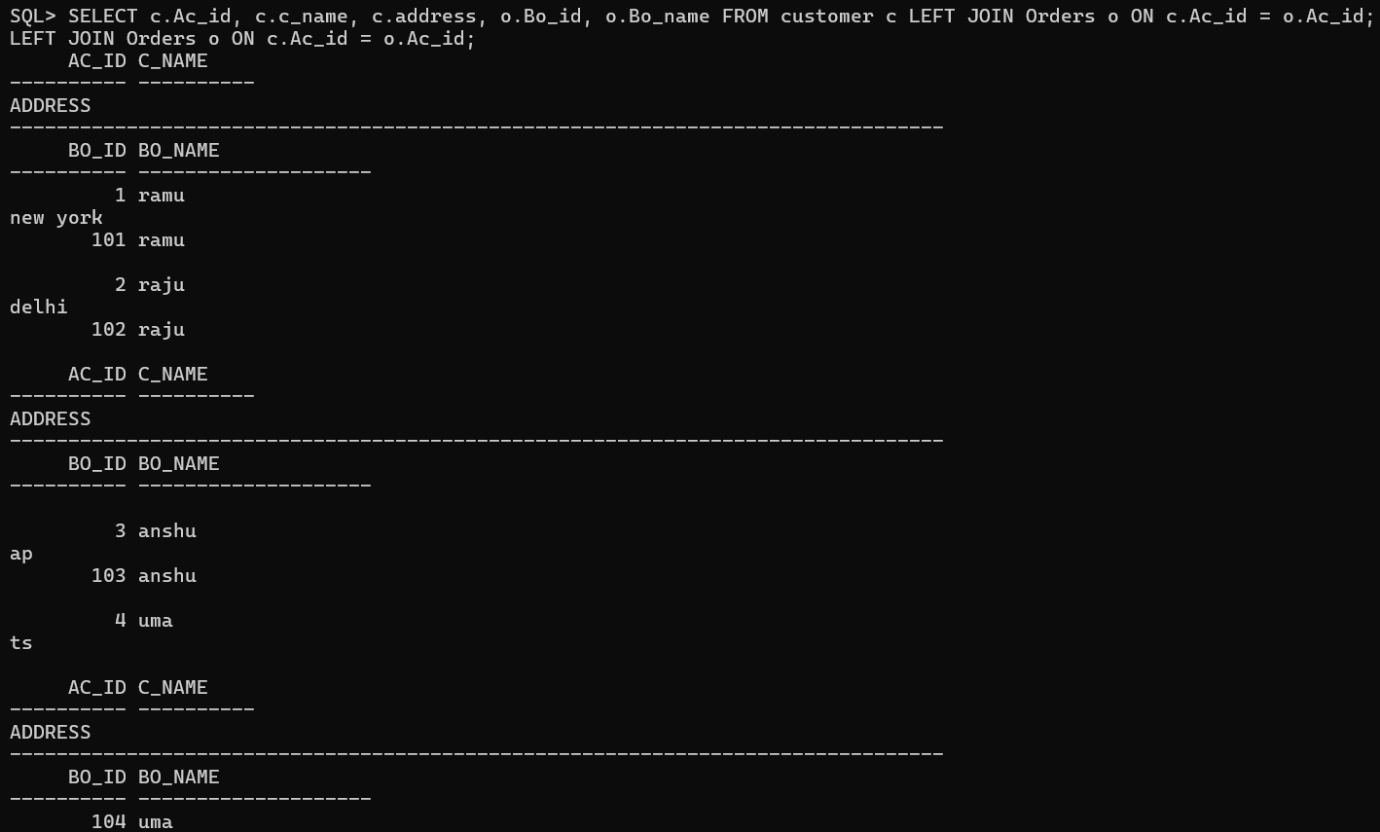
SQL> insert into Orders (Bo\_id, Bo\_name, Ac\_id) values (104, 'Uma' ,4);

SQL>select \* from customer;

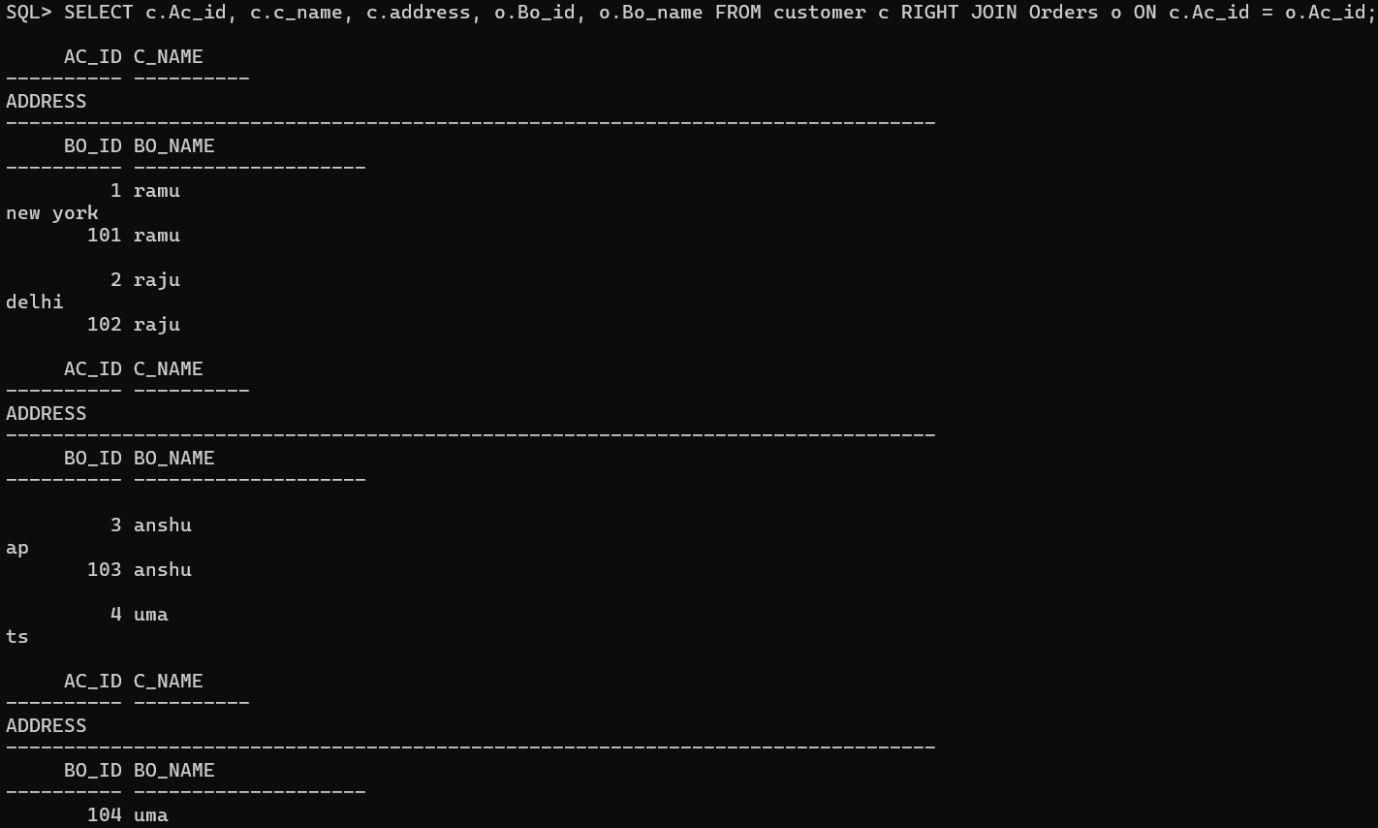
SQL>select \* from orders;



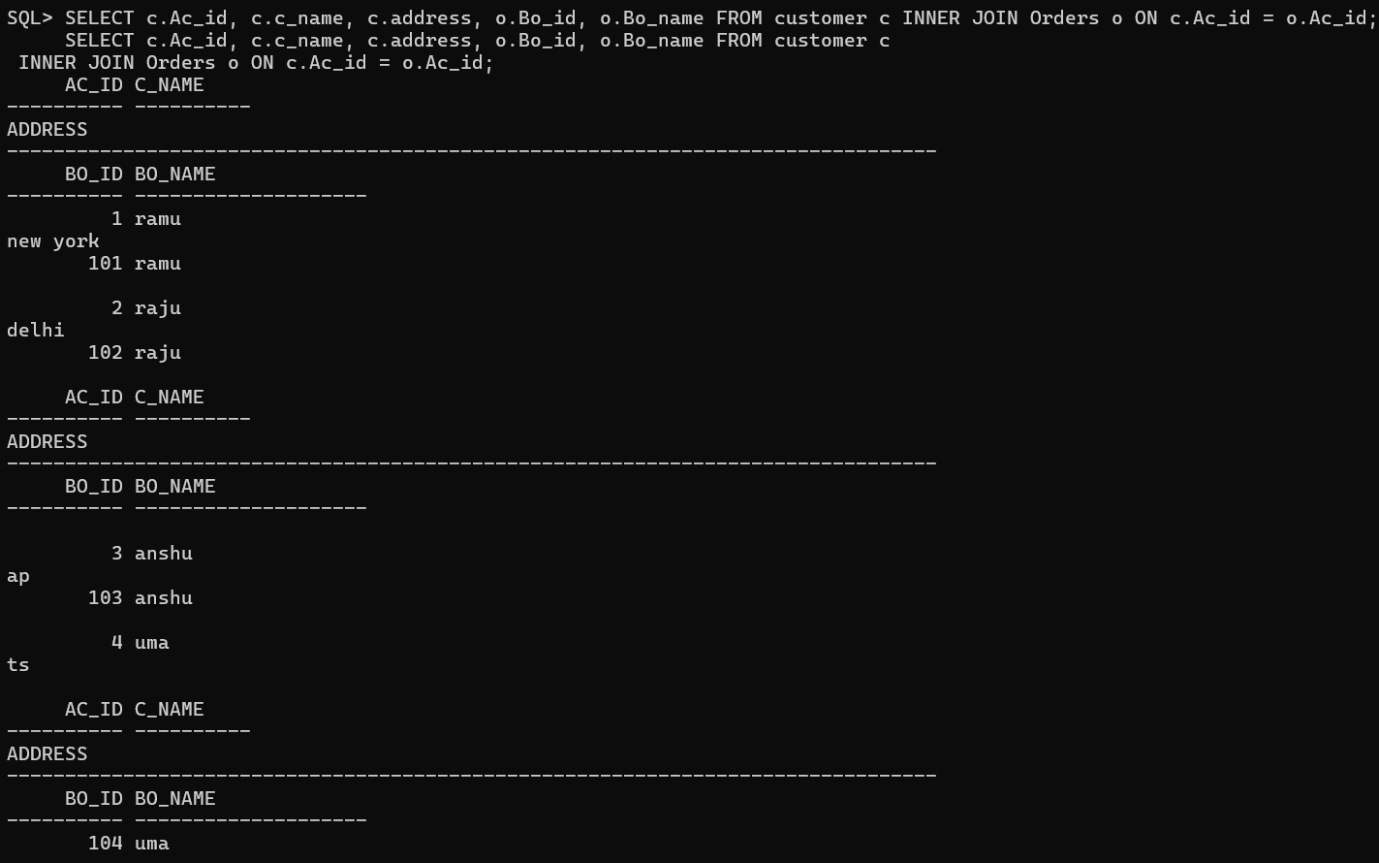
SQL> SELECT c. Ac\_id, c\_name, c. address, o. Bo\_id, o. Bo\_name FROM customer c LEFT JOIN Orders o ON c. Ac\_id = o. Ac\_id;



SQL> SELECT c. Ac\_id, c\_name, c. address, o. Bo\_id, o. Bo\_name FROM customer c RIGHT JOIN Orders o ON c. Ac\_id = o. Ac\_id;

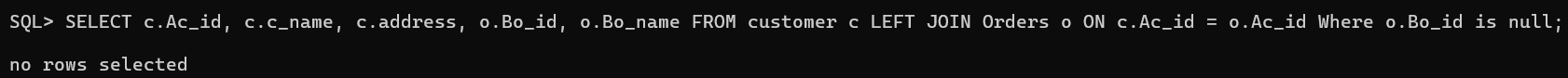


SQL> SELECT c. Ac\_id, c\_name, c. address, o. Bo\_id, o. Bo\_name FROM customer c INNER JOIN Orders o ON c. Ac\_id = o. Ac\_id;



SQL> SELECT c. Ac\_id, c. c\_name, c. address, o. Bo\_id, o. Bo\_name FROM customer c LEFT JOIN Orders o ON c. Ac\_id = o. Ac\_id Where o. Bo\_id is null;

no rows selected



SQL> SELECT c. Ac\_id, c\_name, c. address, o. Bo\_id, o. Bo\_name

2 FROM customer c

3 RIGHT JOIN Orders o ON c. Ac\_id = o. Ac\_id

4 WHERE c. Ac\_id IS NULL;

