***Assignment – 16***

**Creating Tables and Indexes.**

1. Write a command that will enable a user to pull orders grouped by date out of the Orders table quickly.

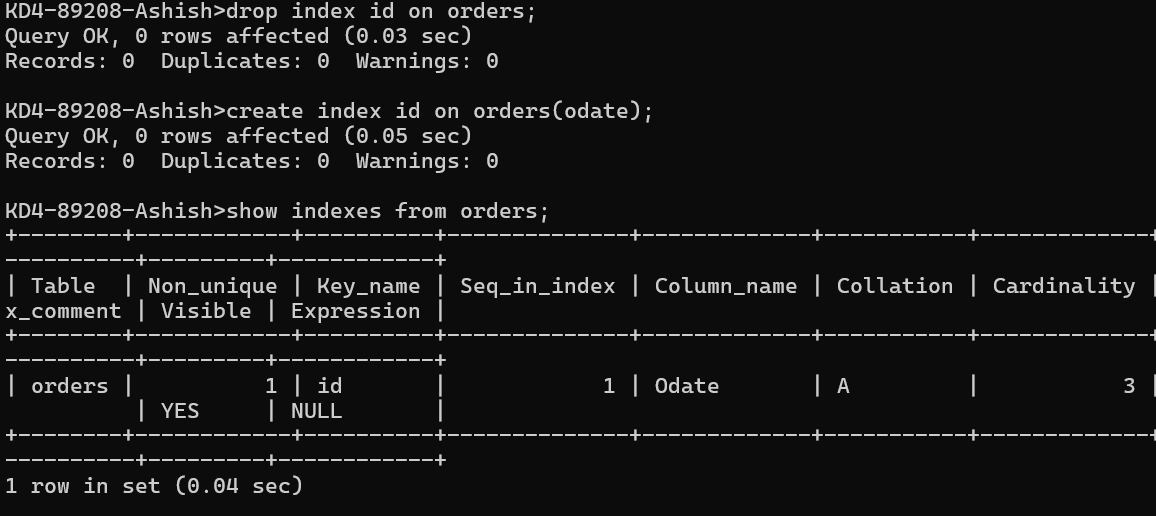
ANS :

KD4-89208-Ashish>create index id on orders(odate);

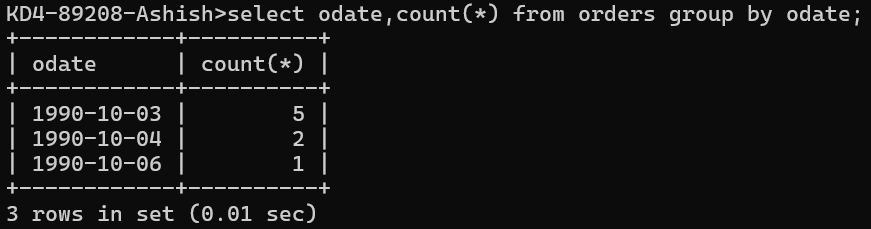
Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

KD4-89208-Ashish>show indexes from orders;



KD4-89208-Ashish>select odate,count(\*) from orders group by odate;



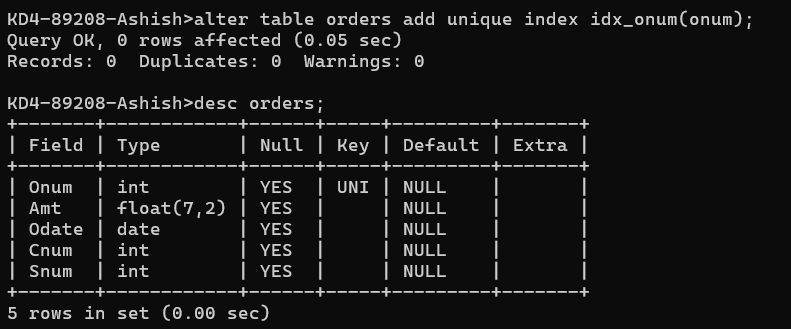
1. If the Orders table has already been created, how can you force the onum field to be unique (assume all current values are unique)?

ANS: KD4-89208-Ashish>alter table orders add unique index idx\_onum(onum);

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

KD4-89208-Ashish>desc orders;



1. Create an index that would permit each salesperson to retrieve his or her orders grouped by date quickly.

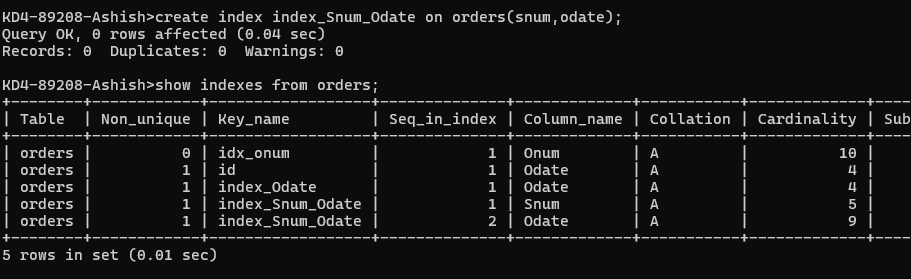
ANS:

KD4-89208-Ashish>create index index\_Snum\_Odate on orders(snum,odate);

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

KD4-89208-Ashish>show indexes from orders;



1. Let us assume that each salesperson is to have only one customer of a given rating, and that this is currently the case. Enter a command that enforces it.

ANS: KD4-89208-Ashish>drop index index\_cnum\_rating on customers;

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

KD4-89208-Ashish>create index index\_cnum\_rating on customers(cnum,rating);

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

KD4-89208-Ashish>select cnum,rating from customers group by cnum,rating;

