


**NAME:** Prashant Sharma

**UID:** 20BCS9390

**SECTION:** 20BCS-KRG-MM-G1

**Q1.** You have a order table with 2 column order no and customer no. write a sql to find the customer no for the customer whose has largest no of order.

### CODE AND OUTPUT:



The screenshot shows the 'Live SQL' web interface. The SQL Worksheet contains the following code:

```
2 CREATE TABLE ORDERS (  
3 ORDER_ID NUMBER(10) PRIMARY KEY,  
4 CUSTOMER_ID NUMBER(10) NOT NULL  
5 );  
6  
7 INSERT INTO ORDERS VALUES (1, 9390);  
8 INSERT INTO ORDERS VALUES (2, 9391);  
9 INSERT INTO ORDERS VALUES (3, 9390);  
10 INSERT INTO ORDERS VALUES (4, 9391);  
11 INSERT INTO ORDERS VALUES (5, 9390);  
12  
13 SELECT CUSTOMER_ID, COUNT(*) as NUM_ORDERS  
14 FROM ORDERS  
15 GROUP BY CUSTOMER_ID  
16 ORDER BY NUM_ORDERS DESC  
17 FETCH FIRST ROW ONLY;
```

Below the code, it says "1 row(s) inserted." and displays the following result table:

CUSTOMER_ID	NUM_ORDERS
9390	3

At the bottom, there is a footer with version information: "2023 Oracle - Live SQL 22.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym" and "Built with using Oracle APEX - Privacy - Terms of Use".

**Q2.** Write an SQL query to display the records with three or more rows with consecutive id's, and the number of people is greater than or equal to 100 for each.

### CODE:

Live SQL

FeedbackHelpprashantonusharma@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1
2 CREATE TABLE STADIUM (
3   ID NUMBER(10) PRIMARY KEY,
4   VISIT_DATE DATE NOT NULL,
5   PEOPLE NUMBER(10) NOT NULL
6 );
7
8 INSERT INTO STADIUM VALUES (1, TO_DATE('2017-01-01', 'YYYY-MM-DD'), 10);
9 INSERT INTO STADIUM VALUES (2, TO_DATE('2017-01-02', 'YYYY-MM-DD'), 109);
10 INSERT INTO STADIUM VALUES (3, TO_DATE('2017-01-03', 'YYYY-MM-DD'), 150);
11 INSERT INTO STADIUM VALUES (4, TO_DATE('2017-01-04', 'YYYY-MM-DD'), 99);
12 INSERT INTO STADIUM VALUES (5, TO_DATE('2017-01-05', 'YYYY-MM-DD'), 145);
13 INSERT INTO STADIUM VALUES (6, TO_DATE('2017-01-06', 'YYYY-MM-DD'), 1455);
14 INSERT INTO STADIUM VALUES (7, TO_DATE('2017-01-07', 'YYYY-MM-DD'), 199);
15 INSERT INTO STADIUM VALUES (8, TO_DATE('2017-01-09', 'YYYY-MM-DD'), 188);
16
17 WITH cte AS (
18   SELECT ID, VISIT_DATE, PEOPLE,
19          ROW_NUMBER() OVER (ORDER BY VISIT_DATE) - ROW_NUMBER() OVER (PARTITION BY ID - 1 ORDER BY VISIT_DATE) AS grp
20   FROM STADIUM
21   WHERE PEOPLE >= 100)
22 SELECT ID, VISIT_DATE, PEOPLE
23 FROM cte
24 WHERE grp >= 2
25 GROUP BY grp, ID, VISIT_DATE, PEOPLE
26 ORDER BY VISIT_DATE;
```

## OUTPUT:

Live SQL

FeedbackHelpprashantonusharma@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1
2 CREATE TABLE STADIUM (
3   ID NUMBER(10) PRIMARY KEY,
4   VISIT_DATE DATE NOT NULL,
5   PEOPLE NUMBER(10) NOT NULL
6 );
```

1 row(s) inserted.

1 row(s) inserted.

ID	VISIT_DATE	PEOPLE
5	05-JAN-17	145
6	06-JAN-17	1455
7	07-JAN-17	199
8	09-JAN-17	188

Download CSV

4 rows selected.