SQL WORKSHEET 3

FlipRobo

1. Write SQL query to create table Customers.

Ans. Create a SQL table called *customers* that stores customer ID, name, and address information.

```
CREATE TABLE customers
( customer_id int NOT NULL,
  customer_name char(50) NOT NULL,
  address char(50),
  city char(50),
  state char(25),
  zip_code char(10),
  CONSTRAINT customers_pk PRIMARY KEY (customer_id)
);
```

2. Write SQL query to create table Orders.

Ans. mysql> create table `order`

- > (

- > Id int,

- > Price int

- >);

Query OK, 0 rows affected (0.66 sec)

Example: mysql> insert into `order` values(1,200);
Query OK, 1 row affected (0.21 sec)
mysql> insert into `order` values(2,100);
Query OK, 1 row affected (0.17 sec)
mysql> insert into `order` values(3,300);
Query OK, 1 row affected (0.20 sec)
mysql> insert into `order` values(4,1200);
Query OK, 1 row affected (0.13 sec)
mysql> insert into `order` values(5,1000);
Query OK, 1 row affected (0.18 sec)
mysql> insert into `order` values(6,7000);
Query OK, 1 row affected (0.20 sec)
mysql> insert into `order` values(7,900);
Query OK, 1 row affected (0.16 sec)
mysql> insert into `order` values(8,10000);
Query OK, 1 row affected (0.18 sec)
mysql> insert into `order` values(9,1100);
Query OK, 1 row affected (0.30 sec)
mysql> insert into `order` values(10,500);
Query OK, 1 row affected (0.18 sec)

mysql> select *from `order`;

3. Write SQL query to show all the columns data from the **Orders** Table.

Ans. SELECT ord_date, customer_id, ord_no, shipped_date, req_date, status_1,com_ments
FROM orders;

4. Write SQL query to show all the comments from the **Orders** Table.

Ans. /* multi-line comment
another comment */
SELECT * FROM Orders;

/*Select all the columns
of all the records
in the Orders table:*/
SELECT * FROM Orders;

5. Write a SQL query to show orderDate and Total number of orders placed on that date, from **Orders** table.

Ans. SELECT date(order_placed_date), COUNT(order_id) AS num_orders, SUM(order_total) AS daily_total
FROM orders
WHERE order_placed_date>=date_sub(current_date, INTERVAL 31 DAY)
GROUP BY date(order_placed_date)

6. Write a SQL query to show employeNumber, lastName, firstName of all the employees from **employees table.**

Ans. SELECT employee_id, first_name, last_name,
(SELECT department_name FROM departments d
WHERE e.department_id = d.department_id) department
FROM employees e ORDER BY department;

7. Write a SQL query to show all orderNumber, customerName of the person who placed the respective order.

Ans. SELECT o.orderNumber
FROM orders o
LEFT JOIN customers c
ON c.customerNumber = o.customerNumber
WHERE c.customerName LIKE 'N%'

/*Declares your variables*/
DECLARE @Initial varchar(1) --Declaring our Initial: Note this is why SQL is self documenting
DECLARE @FieldChoice = int --Declaring our field choice
SET @Initial = 'N' --Initial to search
SET @FieldChoice = 1 -- {1,2,3} = {customerName,contactFirstName,contactLastName}

/*Your query*/
SELECT O.orderNumber
FROM orders O
INNER JOIN customers C
ON C.customerNumber = O.customerNumber
WHERE (LEFT(C.customerName,1) IN (@Initial) AND @FieldChoice = 1)
OR (LEFT(C.contactFirstName,1) IN (@Initial) AND @FieldChoice = 2)
OR (LEFT(C.contactLastName,1) IN (@Initial) AND @FieldChoice = 3)

8. Write a SQL query to show name of all the customers in one column and salerepemployee name in another column.

Ans. SELECT orders.ord_no, customer.cust_name
FROM orders, customer
WHERE orders.customer_id = customer.customer_id;

```
SELECT a.cust_name,a.city, b.ord_no,
b.ord_date,b.purch_amt AS "Order Amount"
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
order by b.ord_date;
```

9. Write a SQL query to show Date in one column and total payment amount of the payments made on that date from the **payments** table.

Ans. SELECT COUNT(*) AS day_of_payment
FROM payment
WHERE TO_CHAR (payment_date,'DAU') = '09/12/2022'
SELECT dau.date, count(*) FROM DAU
WHERE EXISTS (select 1 from Payer
WHERE Payer.user_id=DAU.user_id
AND Payer.payment_timestamp BETWEEN trunc(dau.date)-90 AND trunc(DAU.date))
GROUP BY dau.date
ORDER BY Date;

10. Write a SQL query to show all the products productName, MSRP, productDescription from the **products** table.

Ans. SELECT item_mast.pro_name, pro_price, company_mast.com_name
FROM item_mast
INNER JOIN company_mast
ON item_mast.pro_com = company_mast.com_id;

```
SELECT pro_name as "Item Name", pro_price AS "Price in Rs."
FROM item_mast;
```

11. Write a SQL query to print the productName, productDescription of the most ordered product.

Ans. SELECT P.pro_name AS "Product Name",
P.pro_price AS "Price",
C.com_name AS "Company"
FROM item_mast P, company_mast C
WHERE P.pro_com = C.com_id
AND P.pro_price =(SELECT MAX(P.pro_price)
FROM item_mast P
WHERE P.pro_com = C.com_id
);

12. Write a SQL query to print the city name where maximum number of orders were placed.

Ans. SELECT * FROM customer WHERE grade >
ALL (SELECT grade FROM customer
WHERE city = 'NewYork');
SELECT customer_id, COUNT(DISTINCT ord_no),
MAX(purch_amt)
FROM orders
GROUP BY customer_id
ORDER BY 2 DESC;

13. Write a SQL query to get the name of the state having maximum number of customers.

Ans. SELECT c.STATE, c.CITY, SUM(s.QTY * s.SALEPRICE)
FROM DWSALE s
INNER JOIN DWCUST c
ON s.dwcustid = c.dwcustid
GROUP BY c.STATE, c.CITY
ORDER BY c.STATE;
SELECT state, city, total
FROM (SELECT c.STATE, c.CITY, SUM(s.QTY * s.SALEPRICE) as total,
ROW_NUMBER() OVER (PARTITION BY c.STATE ORDER BY SUM(s.QTY * s.SALEPRICE)) as seqnum
FROM DWSALE s INNER JOIN
DWCUST c ON s.dwcustid = c.dwcustid
GROUP BY c.STATE, c.CITY
) sc
WHERE seqnum = 1
ORDER BY STATE;

14. Write a SQL query to print the employee number in one column and Full name of the employee in the second column for all the employees.

Ans. SELECT e.employee_name, m.employee_name AS manager_name
FROM employee e
JOIN employee m on e.manager_id = m.employee_id
SELECT employee_name AS Employee_Name,(SELECT employee_name
FROM employee
where
employeeid=ManagerID) AS Manager_Name FROM employee

SELECT e.Employee Name,
e.Employee Name as Manager Name
FROM Employee e JOIN Employee m
ON e.Employee id = m.Manager id

15. Write a SQL query to print the orderNumber, customer Name and total amount paid by the customer for that order (quantityOrdered × priceEach).

Ans. SELECT c.CustomerID,o.OrderID,(ord.Quantity*p.Price) as
Total_Amount
from Customers c inner join Orders o
inner join Products p
inner join OrderDetails ord
on c.CustomerID = o.CustomerID

```
and o.OrderID = ord.OrderID
and ord.ProductID = p.ProductID;

SELECT c.CustomerID, SUM(ord.Quantity*p.Price) as
Total_Amount
FROM Customers c inner join Orders o
    on c.CustomerID = o.CustomerID join
    OrderDetails ord
    on o.OrderID = ord.OrderID join
    Products p
    on ord.ProductID = p.ProductID
GROUP BY CustomerID;
```