Big Data & EDW Summer Internship 2023

(Batch 2)

Project Title: E-commerce Platform Database and Reporting System

Database creation

```
Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\priya>mysql -u root -p
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> -- DATABASE SETUP(DDL)
mysql> CREATE DATABASE ecommerce_platform;
Query OK, 1 row affected (0.41 sec)

mysql> USE ecommerce_platform;
Database changed
```

Products Table Creation

```
mysql> CREATE TABLE products (
    -> product_id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(255) NOT NULL,
    ->  price DECIMAL(10, 2) NOT NULL,
    ->  category VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (1.05 sec)
```

Description

```
mysql> DESC products;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| product_id | int          | NO   | PRI | NULL    | auto_increment |
| name       | varchar(255) | NO   |     | NULL    |                |
| price      | decimal(10,2)| NO   |     | NULL    |                |
| category   | varchar(50)  | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
4 rows in set (0.30 sec)
```

Insertion

```
mysql> INSERT INTO products (name, price, category) VALUES
    -> ('Product A', 10.99, 'Electronics'),
    -> ('Product B', 5.99, 'Clothing'),
    -> ('Product C', 15.49, 'Home & Kitchen');
Query OK, 3 rows affected (0.54 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Display

```
mysql> SELECT * FROM products;
+------------+-----------+-------+----------------+
| product_id | name      | price | category       |
+------------+-----------+-------+----------------+
|          1 | Product A | 10.99 | Electronics    |
|          2 | Product B |  5.99 | Clothing       |
|          3 | Product C | 15.49 | Home & Kitchen |
+------------+-----------+-------+----------------+
```

Customers Table Creation

```
mysql> CREATE TABLE customers (
    -> customer_id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(100) NOT NULL,
    -> email VARCHAR(100) NOT NULL,
    -> address VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (1.25 sec)
```

Description

```
mysql> DESC customers;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| customer_id | int          | NO   | PRI | NULL    | auto_increment |
| name        | varchar(100) | NO   |     | NULL    |                |
| email       | varchar(100) | NO   |     | NULL    |                |
| address     | varchar(255) | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
4 rows in set (0.03 sec)
```

Insertion

```
mysql> -- Insert sample data into customers table
mysql> INSERT INTO customers (name, email, address) VALUES
    -> ('John Doe', 'john@example.com', '123 Main St'),
    -> ('Jane Smith', 'jane@example.com', '456 Oak Ave');
Query OK, 2 rows affected (0.13 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Display

```
mysql> SELECT * FROM customers;
+-------------+------------+------------------+-------------+
| customer_id | name       | email            | address     |
+-------------+------------+------------------+-------------+
|           1 | John Doe   | john@example.com | 123 Main St |
|           2 | Jane Smith | jane@example.com | 456 Oak Ave |
+-------------+------------+------------------+-------------+
2 rows in set (0.00 sec)
```

Order Table Creation

```
mysql> CREATE TABLE orders (
    -> order_id INT AUTO_INCREMENT PRIMARY KEY,
    ->  customer_id INT NOT NULL,
    -> order_date DATE NOT NULL,
    -> total_amount DECIMAL(10, 2) NOT NULL,
    ->  FOREIGN KEY (customer_id) REFERENCES customers (customer_id)
    -> );
Query OK, 0 rows affected (0.54 sec)
```

Description

```
mysql> DESC orders;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| order_id     | int           | NO   | PRI | NULL    | auto_increment |
| customer_id  | int           | NO   | MUL | NULL    |                |
| order_date   | date          | NO   |     | NULL    |                |
| total_amount | decimal(10,2) | NO   |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Insertion

```
mysql> -- Insert sample data into orders and order_items tables
mysql> INSERT INTO orders (customer_id, order_date, total_amount) VALUES
    -> (1, '2023-08-01', 26.98);
Query OK, 1 row affected (0.17 sec)
```

Display

```
mysql> SELECT * FROM orders;
+----------+-------------+------------+--------------+
| order_id | customer_id | order_date | total_amount |
+----------+-------------+------------+--------------+
|        1 |           1 | 2023-08-01 |        26.98 |
+----------+-------------+------------+--------------+
1 row in set (0.00 sec)
```

Order item table creation

```
mysql> CREATE TABLE order_items (
    -> item_id INT AUTO_INCREMENT PRIMARY KEY,
    -> order_id INT NOT NULL,
    -> product_id INT NOT NULL,
    -> quantity INT NOT NULL,
    -> FOREIGN KEY (order_id) REFERENCES orders (order_id),
    -> FOREIGN KEY (product_id) REFERENCES products (product_id)
    -> );
Query OK, 0 rows affected (0.35 sec)
```

Description

```
mysql> DESC order_items;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| item_id    | int  | NO   | PRI | NULL    | auto_increment |
| order_id   | int  | NO   | MUL | NULL    |                |
| product_id | int  | NO   | MUL | NULL    |                |
| quantity   | int  | NO   |     | NULL    |                |
+------------+------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Insertion

```
mysql> INSERT INTO order_items (order_id, product_id, quantity) VALUES
    ->  (1, 1, 2), -- 2 units of Product A
    -> (1, 2, 3); -- 3 units of Product B
Query OK, 2 rows affected (0.18 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Display

```
mysql> SELECT * FROM order_items;
+---------+----------+------------+----------+
| item_id | order_id | product_id | quantity |
+---------+----------+------------+----------+
|       1 |        1 |          1 |        2 |
|       2 |        1 |          2 |        3 |
+---------+----------+------------+----------+
2 rows in set (0.00 sec)
```

Payment method Table Creation

```
mysql> CREATE TABLE payment_methods (
    ->     payment_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     customer_id INT NOT NULL,
    ->     payment_type VARCHAR(50) NOT NULL,
    ->     card_number VARCHAR(16) NOT NULL,
    ->     expiration_date DATE NOT NULL,
    ->     cvv VARCHAR(4) NOT NULL,
    ->     FOREIGN KEY (customer_id) REFERENCES customers (customer_id)
    -> );
Query OK, 0 rows affected (2.04 sec)
```

Description

```
mysql> DESC payment_methods;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| payment_id      | int         | NO   | PRI | NULL    | auto_increment |
| customer_id     | int         | NO   | MUL | NULL    |                |
| payment_type    | varchar(50) | NO   |     | NULL    |                |
| card_number     | varchar(16) | NO   |     | NULL    |                |
| expiration_date | date        | NO   |     | NULL    |                |
| cvv             | varchar(4)  | NO   |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
6 rows in set (0.11 sec)
```

Insertion

```
mysql> INSERT INTO payment_methods (customer_id, payment_type, card_number, expiration_date, cvv)
    -> VALUES
    ->     (1, 'Credit Card', '1234567890123456', '2025-12-31', '123'),
    ->     (2, 'Debit Card', '9876543210987654', '2024-06-30', '789');
Query OK, 2 rows affected (0.20 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Display

```
mysql> SELECT * FROM payment_methods;
+------------+------------+--------------+------------------+-----------------+-----+
| payment_id | customer_id | payment_type | card_number      | expiration_date | cvv |
+------------+------------+--------------+------------------+-----------------+-----+
|          1 |          1 | Credit Card  | 1234567890123456 | 2025-12-31      | 123 |
|          2 |          2 | Debit Card   | 9876543210987654 | 2024-06-30      | 789 |
+------------+------------+--------------+------------------+-----------------+-----+
2 rows in set (0.00 sec)
```

Shipping Providers Table Creation

```
mysql> CREATE TABLE shipping_providers (
    ->     provider_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100) NOT NULL,
    ->     contact_email VARCHAR(100) NOT NULL,
    ->     contact_phone VARCHAR(20) NOT NULL
    -> );
Query OK, 0 rows affected (0.42 sec)
```

Description

```
mysql> DESC shipping_providers;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| provider_id   | int          | NO   | PRI | NULL    | auto_increment |
| name          | varchar(100) | NO   |     | NULL    |                |
| contact_email | varchar(100) | NO   |     | NULL    |                |
| contact_phone | varchar(20)  | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Insertion

```
mysql> INSERT INTO shipping_providers (name, contact_email, contact_phone)
    -> VALUES
    ->     ('Fast Ship', 'info@fastship.com', '+1 (123) 456-7890'),
    ->     ('Express Delivery', 'support@express.com', '+1 (987) 654-3210');
Query OK, 2 rows affected (0.16 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Display

```
mysql> SELECT * FROM shipping_providers;
+-------------+------------------+---------------------+-------------------+
| provider_id | name             | contact_email       | contact_phone     |
+-------------+------------------+---------------------+-------------------+
|           1 | Fast Ship        | info@fastship.com   | +1 (123) 456-7890 |
|           2 | Express Delivery | support@express.com | +1 (987) 654-3210 |
+-------------+------------------+---------------------+-------------------+
2 rows in set (0.02 sec)
```

Performing Queries

1. Retrieve top-selling products.

```
mysql> SELECT p.name, SUM(oi.quantity) AS total_quantity
    -> FROM products p
    -> JOIN order_items oi ON p.product_id = oi.product_id
    -> GROUP BY p.product_id
    -> ORDER BY total_quantity DESC
    -> LIMIT 5;
+-----------+----------------+
| name      | total_quantity |
+-----------+----------------+
| Product B |              3 |
| Product A |              2 |
+-----------+----------------+
2 rows in set (0.19 sec)
```

2.Retrieve all customer information.

```
mysql> -- Retrieve all customer information
mysql> SELECT * FROM customers;
+-------------+------------+------------------+-------------+
| customer_id | name       | email            | address     |
+-------------+------------+------------------+-------------+
|           1 | John Doe   | john@example.com | 123 Main St |
|           2 | Jane Smith | jane@example.com | 456 Oak Ave |
+-------------+------------+------------------+-------------+
2 rows in set (0.00 sec)
```

3.Retrieve a specific customer by email.

```
mysql> -- Retrieve a specific customer by email
mysql> SELECT * FROM customers WHERE email = 'john@example.com';
+-------------+----------+------------------+-------------+
| customer_id | name     | email            | address     |
+-------------+----------+------------------+-------------+
|           1 | John Doe | john@example.com | 123 Main St |
+-------------+----------+------------------+-------------+
1 row in set (0.12 sec)
```

4.Calculate total revenue for August 2023.

```
mysql> -- Calculate total revenue for August 2023
mysql> SELECT SUM(total_amount) AS total_revenue
    -> FROM orders
    -> WHERE order_date >= '2023-08-01' AND order_date < '2023-09-01';
+---------------+
| total_revenue |
+---------------+
|         26.98 |
+---------------+
1 row in set (0.13 sec)
```

5.Identify customers with the highest number of orders.

```
mysql> -- Identify customers with the highest number of orders
mysql> SELECT c.name, COUNT(o.order_id) AS num_orders
    -> FROM customers c
    -> JOIN orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.name
    -> ORDER BY num_orders DESC
    -> LIMIT 5;
+----------+------------+
| name     | num_orders |
+----------+------------+
| John Doe |          1 |
+----------+------------+
1 row in set (0.12 sec)
```

6.Customer Loyalty View.

```
mysql> -- Customer Loyalty View (Repeat Customers and Purchase History)
mysql> CREATE VIEW customer_loyalty_view AS
    -> SELECT c.customer_id, c.name,
    -> COUNT(o.order_id) AS num_orders,
    -> SUM(o.total_amount) AS total_spent
    -> FROM customers c
    -> JOIN orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.customer_id, c.name;
Query OK, 0 rows affected (0.37 sec)
```

7.Shipping performance View.

```
mysql> -- Shipping Performance View (Average Delivery Times and Delayed Orders)
mysql> CREATE VIEW shipping_performance_view AS
    -> SELECT o.order_id,
    -> o.order_date,
    ->  DATEDIFF(o.order_date, MIN(o.order_date)) AS delivery_time,
    ->  CASE WHEN DATEDIFF(o.order_date, MIN(o.order_date)) > 5
    -> THEN 'Delayed' ELSE 'On Time' END AS delivery_status
    -> FROM orders o
    -> GROUP BY o.order_id, o.order_date;
Query OK, 0 rows affected (0.19 sec)
```

8.Most Popular Product Categories.

```
mysql> -- Most Popular Product Categories
mysql> SELECT category, COUNT(*) AS num_products
    -> FROM products
    -> GROUP BY category
    -> ORDER BY num_products DESC;
+---------------+--------------+
| category      | num_products |
+---------------+--------------+
| Electronics   |            1 |
| Clothing      |            1 |
| Home & Kitchen |            1 |
+---------------+--------------+
3 rows in set (0.03 sec)
```

9.Region with the Highest Sales.

```
mysql> -- Region with the Highest Sales
mysql> SELECT c.address, SUM(o.total_amount) AS total_sales
    -> FROM customers c
    -> JOIN orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.address
    -> ORDER BY total_sales DESC
    -> LIMIT 1;
+-------------+-------------+
| address     | total_sales |
+-------------+-------------+
| 123 Main St |       26.98 |
+-------------+-------------+
1 row in set (0.00 sec)
```

10.List all products along with their average rating.

```
mysql> SELECT p.name AS product_name, AVG(pr.rating) AS average_rating
    -> FROM products p
    -> LEFT JOIN product_reviews pr ON p.product_id = pr.product_id
    -> GROUP BY p.product_id, product_name;
+--------------+----------------+
| product_name | average_rating |
+--------------+----------------+
| Product A    |         5.0000 |
| Product B    |         4.0000 |
| Product C    |           NULL |
+--------------+----------------+
3 rows in set (0.16 sec)
```

11.Get Shipping details for an order.

```
mysql> SELECT o.order_id, o.order_date, sp.name AS shipping_provider, s.shipping_date, s.delivery_status, s.tracking_number
    -> FROM orders o
    -> LEFT JOIN shipping_details s ON o.order_id = s.order_id
    -> LEFT JOIN shipping_providers sp ON s.provider_id = sp.provider_id
    -> WHERE o.order_id = 1;
+----------+------------+------------------+---------------+-----------------+-----------------+
| order_id | order_date | shipping_provider | shipping_date | delivery_status | tracking_number |
+----------+------------+------------------+---------------+-----------------+-----------------+
|        1 | 2023-08-01 | Fast Ship         | 2023-08-02    | Shipped         | FS123456789     |
|        1 | 2023-08-01 | Express Delivery  | 2023-08-03    | In Transit      | ED987654321     |
+----------+------------+------------------+---------------+-----------------+-----------------+
2 rows in set (0.03 sec)
```

12.Retrieve the number of products in each category.

```
mysql> SELECT category, COUNT(*) AS num_products
    -> FROM products
    -> GROUP BY category;
+-----------------+--------------+
| category        | num_products |
+-----------------+--------------+
| Electronics     |            1 |
| Clothing        |            1 |
| Home & Kitchen  |            1 |
+-----------------+--------------+
3 rows in set (0.00 sec)
```

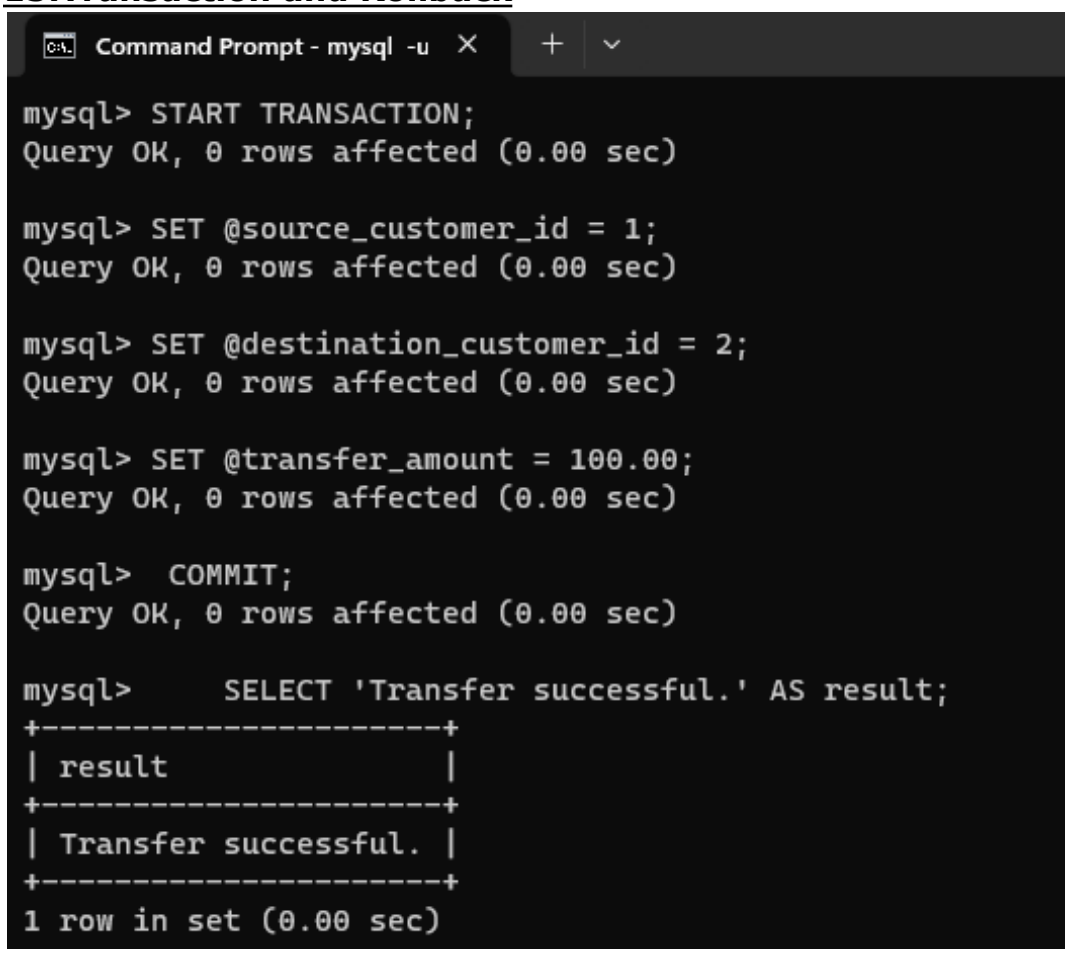
13.Calculate the average order total amount.

```
mysql> SELECT AVG(total_amount) AS avg_order_total
    -> FROM orders;
+------------------+
| avg_order_total  |
+------------------+
|       26.980000  |
+------------------+
1 row in set (0.11 sec)
```

14.Update the table products.

```
mysql> UPDATE products SET price = 49.99
    -> WHERE product_id = 3;
Query OK, 1 row affected (0.53 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

15.Transaction and Rollback



----------X----------

QUESTIONS & ANSWERS

1) Write an SQL query to fetch unique values of DEPARTMENT from the Worker table.

Ans:
```
SELECT DISTINCT DEPARTMENT
FROM Worker;
```

2) Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from the Worker table.

Ans:
```
SELECT INSTR(FIRST_NAME, 'a') AS position_of_a
FROM Worker
WHERE FIRST_NAME = 'Amitabh';
```

3) Write an SQL query that fetches the unique values of DEPARTMENT from the Worker table and prints its length.

Ans: :
```
SELECT DISTINCT DEPARTMENT,
LENGTH(DEPARTMENT) AS department_length
FROM Worker;
```

4) Write an SQL query to print the FIRST_NAME and LAST_NAME from the Worker table into a single column COMPLETE_NAME. A space char should separate them.

Ans:
```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS
COMPLETE_NAME
FROM Worker;
```

5) Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.

Ans: SELECT *
FROM Worker
ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;

6) Write an SQL query to print details of the Workers whose FIRST_NAME contains ‘a’.

Ans: SELECT *
FROM Worker
WHERE FIRST_NAME LIKE '%a%';

7) Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

Ans: SELECT *
FROM Worker
WHERE SALARY BETWEEN 100000 AND 500000;

8) Write an SQL query to print details of the Workers who joined in Feb’2014.

Ans: SELECT *FROM Worker
WHERE DATEPART(YEAR, JOINING_DATE) = 2014 AND
DATEPART(MONTH, JOINING_DATE) = 2;

9) Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

Ans: SELECT FIRST_NAME, LAST_NAME
FROM Worker
WHERE SALARY >= 50000 AND SALARY <= 100000;

10) Write an SQL query to fetch the no. of workers for each department in descending order.

Ans: SELECT DEPARTMENT, COUNT(*) AS NumOfWorkers
FROM Worker
GROUP BY DEPARTMENT
ORDER BY NumOfWorkers DESC;

11) Write an SQL query to show only odd rows from a table.

Ans: SELECT *
FROM (
SELECT *, ROW_NUMBER() OVER () AS rn
FROM YourTableName
)
AS temp WHERE rn % 2 = 1;

12) Write an SQL query to clone a new table from another table.

Ans: SELECT *
INTO New_Table
FROM Worker;

13) Write an SQL query to determine the nth (say n=5) highest salary from a table.

Ans: SELECT Salary
FROM Worker w1
WHERE 6 - 1 = (
SELECT COUNT(DISTINCT Salary)
FROM Worker w2
WHERE w2.Salary > w1.Salary
);

14) Write an SQL query to determine the 5th highest salary without using the TOP or limit method.

Ans: SELECT DISTINCT Salary
FROM Worker w1
WHERE 5 = (
SELECT COUNT(DISTINCT Salary)
FROM Worker w2
WHERE w2.Salary >= w1.Salary
)
ORDER BY Salary DESC;

15) Write an SQL query to fetch the list of employees with the same salary.
Ans: SELECT FIRST_NAME, LAST_NAME, SALARY
FROM Worker
GROUP BY FIRST_NAME, LAST_NAME, SALARY
HAVING COUNT(*) > 1;

16) Write an SQL query to show one row twice in the results from a table.
Ans: SELECT *
FROM YourTableName
WHERE SomeCondition
UNION ALL
SELECT *
FROM YourTableName
WHERE SomeCondition;

17) Write an SQL query to fetch the last five records from worker table.
Ans: SELECT TOP 5 *
FROM Worker
ORDER BY Worker_ID DESC;

18) Write an SQL query to print the name of employees having the highest salary in each department.

Ans: SELECT Department, MAX(Salary) AS MaxSalary
FROM Worker
GROUP BY Department;

19) Write an SQL query to fetch the names of workers who earn the highest salary.

Ans: SELECT First_Name, Last_Name, Salary
FROM Worker
WHERE Salary = (SELECT MAX(Salary) FROM Worker);

20) Write an SQL query to fetch nth max salaries from a table.

Ans: SELECT Salary
FROM Worker w1
WHERE 0 = (
SELECT COUNT(DISTINCT Salary)
FROM Worker w2
WHERE w2.Salary > w1.Salary
);

Submitted By : Priya Yadav
College : Raj Kumar Goel Institute Of Technology , Ghaziabad
Roll No: 2000330100164
SQL (Batch 2)