1. **Query to Create a View Showing Employees in the IT Department:**

Table: `Employees`

```yaml
| Emp_ID | Emp_Name | Department | Salary |
|--------|----------|------------|--------|
| 1001   | Alice    | HR         | 5000   |
| 1002   | Bob      | IT         | 6000   |
| 1003   | Charlie  | Sales      | 5500   |
```

View:

```sql
CREATE VIEW IT_Employees AS
SELECT * FROM Employees WHERE Department = 'IT';
```

Output: Selecting from the view:

SELECT * FROM IT_Employees;

Expected output:

```yaml
| Emp_ID | Emp_Name | Department | Salary |
|--------|----------|------------|--------|
| 1002   | Bob      | IT         | 6000   |
```

2. **Query to Create a View Showing Employees with High Salaries:**

Table: `Employees`

View:

CREATE VIEW High_Salary_Employees AS

SELECT * FROM Employees WHERE Salary > 5500;

Output: Selecting from the view:

SELECT * FROM High_Salary_Employees;

Expected output:

```yaml
| Emp_ID | Emp_Name | Department | Salary |
|--------|----------|------------|--------|
| 1002   | Bob      | IT         | 6000   |
| 1003   | Charlie  | Sales      | 5500   |
```

3. **Query to Create a View Showing Total Sales by Department:**

Table: `Sales`

Table: `Sales`

```lua
| Sale_ID | Department | Amount |
|---------|------------|--------|
| 1       | HR         | 200    |
| 2       | IT         | 300    |
| 3       | HR         | 400    |
| 4       | Sales      | 500    |
```

View:

```sql
CREATE VIEW Total_Sales_By_Department AS
SELECT Department, SUM(Amount) AS Total_Sales
FROM Sales
GROUP BY Department;
```

Output: Selecting from the view:

SELECT * FROM Total_Sales_By_Department;

Expected output:

```lua
| Department | Total_Sales |
|------------|-------------|
| HR         | 600         |
| IT         | 300         |
| Sales      | 500         |
```

4. **Query to Create a View Showing Active Customers:**

Table: `Customers`

```mathematica
| Customer_ID | Customer_Name | Status |
|-------------|---------------|--------|
| 1           | John          | Active |
| 2           | Alice         | Active |
| 3           | Bob           | Inactive |
| 4           | Charlie       | Active |
```

View:

```sql
CREATE VIEW Active_Customers AS
SELECT * FROM Customers WHERE Stat = 'Active';
```

Output: Selecting from the view:

SELECT * FROM Active_Customers;

Expected output:

```
mathematica

| Customer_ID | Customer_Name | Status |
|-------------|---------------|--------|
| 1           | John          | Active |
| 2           | Alice         | Active |
| 4           | Charlie       | Active |
```

5. **Query to Create a View Showing Orders Placed Today:**

Table: `Orders`

Table: `` `Orders` ``

```
yaml                                                    Copy code

| Order_ID | Customer_ID | Order_Date | Total_Amount |
|----------|-------------|------------|--------------|
| 1        | 1           | 2024-04-25 | 150.00       |
| 2        | 2           | 2024-04-25 | 200.00       |
| 3        | 1           | 2024-04-24 | 100.00       |
| 4        | 3           | 2024-04-25 | 250.00       |
```

View:

CREATE VIEW Orders_Placed_Today AS

SELECT * FROM Orders WHERE Order_Date = CURDATE();

Output: Selecting from the view:

SELECT * FROM Orders_Placed_Today;

Expected output (if the current date is 2024-04-25):

```
| Order_ID | Customer_ID | Order_Date | Total_Amount |
|----------|-------------|------------|--------------|
| 1        | 1           | 2024-04-25 | 150.00       |
| 2        | 2           | 2024-04-25 | 200.00       |
| 4        | 3           | 2024-04-25 | 250.00       |
```

These examples illustrate how views can be used to create virtual tables based on the data in existing tables, enabling simplified access to specific subsets of data or providing summarized information.

Question: 2

1. **Query to Create a View Showing Students with High Grades:**

Table: `Grades`

```
css

| Student_ID | Course_ID | Grade |
|------------|-----------|-------|
| 1          | 101       | A     |
| 2          | 101       | B     |
| 3          | 101       | A     |
| 1          | 102       | B     |
| 2          | 102       | A     |
| 3          | 102       | A     |
```

View:

CREATE VIEW High_Grades_Students AS

SELECT DISTINCT Student_ID

FROM Grades

WHERE Grade = 'A';

Output: Selecting from the view:

SELECT * FROM High_Grades_Students;

Expected output:

```lua
| Student_ID |
|------------|
| 1          |
| 2          |
| 3          |
```

2. **Query to Create a View Showing Employees Who Have Been Promoted:**

Table: `Employee_Promotions`

Table: `Employee_Promotions`

```yaml
| Emp_ID | Promotion_Date | Promotion_Type |
|--------|----------------|----------------|
| 1001   | 2023-01-15     | Senior Analyst |
| 1002   | 2023-02-20     | Manager        |
| 1003   | 2023-03-10     | Senior Manager |
| 1001   | 2024-04-05     | Director       |
| 1004   | 2024-03-01     | Team Lead      |
```

CREATE VIEW Promoted_Employees AS

SELECT DISTINCT Emp_ID

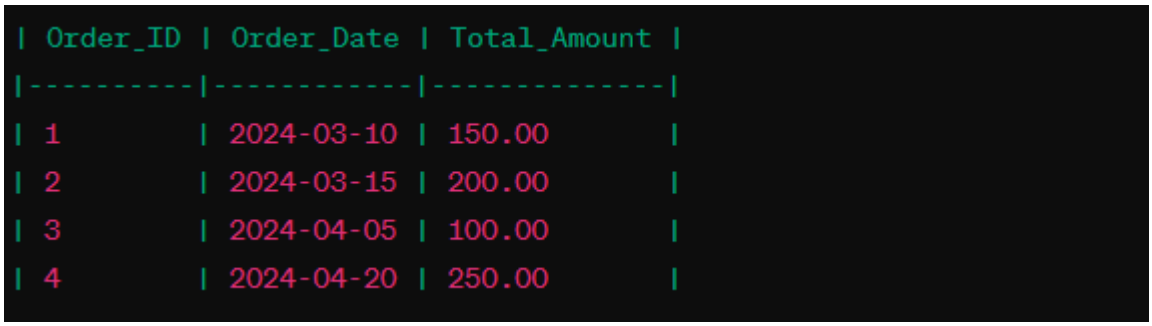
FROM Employee_Promotions;

Output: Selecting from the view:

SELECT * FROM Promoted_Employees;

3. **Query to Create a View Showing Orders Placed Last Month:**

Table: `Orders`



View:

CREATE VIEW Orders_Last_Month AS

SELECT *

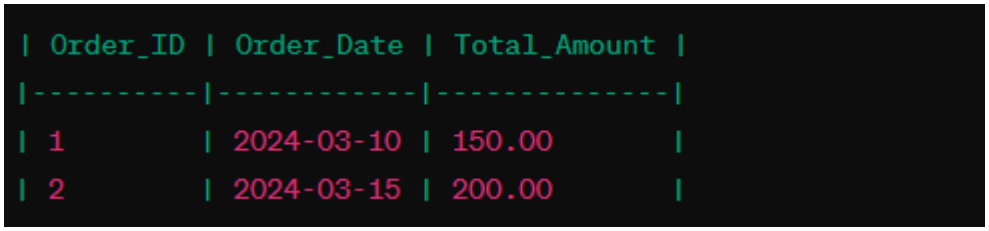
FROM Orders

WHERE MONTH(Order_Date) = MONTH(NOW()) - 1;

Output: Selecting from the view:

SELECT * FROM Orders_Last_Month;

Expected output (if the current month is April):



4. **Query to Create a View Showing Available Products:**

Tables: `Products` and `Orders`

```
Products:
| Product_ID | Product_Name | Quantity |
|------------|--------------|----------|
| 1          | Laptop       | 10       |
| 2          | Smartphone   | 5        |
| 3          | Tablet       | 3        |

Orders:
| Order_ID | Product_ID | Quantity |
|----------|------------|----------|
| 1        | 1          | 2        |
| 2        | 1          | 3        |
| 3        | 2          | 1        |
```

CREATE VIEW Available_Products AS

SELECT p.Product_ID, p.Product_Name, p.Quantity - COALESCE(SUM(o.Quantity), 0) AS Available_Quantity

FROM Products p

LEFT JOIN Orders o ON p.Product_ID = o.Product_ID

GROUP BY p.Product_ID, p.Product_Name, p.Quantity;

Output: Selecting from the view:

SELECT * FROM Available_Products;

Expected output:

```
| Product_ID | Product_Name | Available_Quantity |
|------------|--------------|--------------------|
| 1          | Laptop       | 5                  |
| 2          | Smartphone   | 4                  |
| 3          | Tablet       | 3                  |
```

5. **Query to Create a View Showing Students Enrolled in a Specific Course:**

Tables: `Students` and `Enrollments`

```
Students:
| Student_ID | Student_Name | Age |
|------------|--------------|-----|
| 1          | Alice        | 20  |
| 2          | Bob          | 21  |
| 3          | Charlie      | 22  |

Enrollments:
| Student_ID | Course_ID | Enrollment_Date |
|------------|-----------|-----------------|
| 1          | 101       | 2024-03-01      |
| 2          | 102       | 2024-03-10      |
| 3          | 101       | 2024-04-05      |
| 2          | 101       | 2024-04-10      |
```

View:

CREATE VIEW Course_Enrollments AS

SELECT s.Student_ID, s.Student_Name, e.Course_ID, e.Enrollment_Date

FROM Students s

JOIN Enrollments e ON s.Student_ID = e.Student_ID;

Output: Selecting from the view:

SELECT * FROM Course_Enrollments WHERE Course_ID = 101;

Expected output:

```
| Student_ID | Student_Name | Course_ID | Enrollment_Date |
|------------|--------------|-----------|-----------------|
| 1          | Alice        | 101       | 2024-03-01      |
| 3          | Charlie      | 101       | 2024-04-05      |
| 2          | Bob          | 101       | 2024-04-10      |
```