Scenario mySQL Set question 2

Scenario 1: Identify Consistent High Performers

Question: Find employees who have consistently received a salary increase for the past 3 years.

Logic:

1. Use a **self-join** on the `salary` table based on the `employee_id`.
2. Compare the employee's **salary data across consecutive years** using the `year` column.
3. Filter results to include only those employees whose **salary has increased each year** for the last three years.

Scenario 2: Customer Retention Analysis

Question: Find customers who made purchases in the last 6 months but not in the previous 6 months.

Logic:

1. Use two subqueries on the orders table to identify customers who made purchases within each specific 6-month period.
2. The first subquery captures customers who purchased in the most recent 6 months.
3. The second subquery captures customers who purchased in the earlier 6-month window.
4. Use the NOT IN condition to exclude customers who made purchases in the previous 6 months, retaining only new or reactivated customers.

Scenario 3: Identify Products with Declining Sales

Question: Find products whose sales decreased by more than 30% compared to the previous month.

Logic:

1. Perform a **self-join** on the `sales` table using `product_id` to compare data across months.
2. Match each product's **current month** with its **previous month** using the month field.
3. Calculate the **percentage decrease** in sales between the two months.
4. Filter and display only those products where the decline exceeds **30%**.

Scenario 4: Calculate Moving Average of Sales

Question: Calculate a 3-month moving average for product sales.

Logic:

1. Use a **window function** with the `AVG()` function to compute the moving average over a 3-month period.
2. **Partition** the data by `product_id` so that calculations are done separately for each product.
3. **Order** the data by `sales_date` to ensure the moving average is computed chronologically for each product.

Scenario 5: Detect Duplicate Records

Question: Find all duplicate records from a customer table based on email and phone number.

Logic:

1. Use the **GROUP BY** clause on `email` and `phone_number` to group identical records.
2. Apply the **HAVING** clause to filter groups that have **more than one occurrence**, indicating duplicate entries.

Scenario 6: Identify Products with No Sales for 3 Consecutive Months

Question: List products that have not had any sales for 3 consecutive months.

Logic:

1. Use a **LEFT JOIN** or a **subquery** on the `sales` table to identify months where a product has no sales data.
2. Apply **date functions** to detect gaps between sales months for each product.
3. Filter and list products that have **no sales activity for three consecutive months**.