Scenario 1:

```
-- Create or Replace Database Wonderla
CREATE OR REPLACE DATABASE wonderla;
USE wonderla;



-- Create or Replace User Details Table
CREATE OR REPLACE TABLE UserDetails (
   user_id INT PRIMARY KEY,
   user_name VARCHAR(200),
   birth_year INT,
   nationality VARCHAR(200)
);

-- Create or Replace SKU Details Table
CREATE OR REPLACE TABLE SKUDetails (
   sku_id INT PRIMARY KEY,
   sku_name VARCHAR(200)
);

-- Create or Replace Order Details Table
CREATE OR REPLACE TABLE OrderDetails (
   order_id INT PRIMARY KEY,
   user_id INT,
   sku_id INT,
   amount DECIMAL(10, 2),
   quantity INT,
   order_date DATE,
   FOREIGN KEY (user_id) REFERENCES UserDetails(user_id),
   FOREIGN KEY (sku_id) REFERENCES SKUDetails(sku_id)
);

-- Create or Replace Feedback Table
CREATE OR REPLACE TABLE Feedback (
   review_id INT PRIMARY KEY,
   user_id INT,
   order_id INT,
   review_rating INT,
   question_1 VARCHAR(250),
   question_2 VARCHAR(250),
   question_3 VARCHAR(250),
   FOREIGN KEY (user_id) REFERENCES UserDetails(user_id),
   FOREIGN KEY (order_id) REFERENCES OrderDetails(order_id)
);
```

--Q1--Write a query to find the total order amount for each user in the year 2023 who has made at least two orders.

```
SELECT user_id, SUM(amount) AS total_order_amount
FROM OrderDetails
WHERE EXTRACT(YEAR FROM order_date) = 2023
GROUP BY user_id
HAVING COUNT(order_id) >= 2;
```

--Q2--Write a query to find top-reviewed SKU for each month

```
SELECT EXTRACT(MONTH FROM o.order_date) AS month,
    EXTRACT(YEAR FROM o.order_date) AS year,
    s.sku_id,
    s.sku_name,
    MAX(f.review_rating) AS max_review_rating
FROM OrderDetails o
JOIN Feedback f ON o.order_id = f.order_id
JOIN SKUDetails s ON o.sku_id = s.sku_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2023
GROUP BY EXTRACT(MONTH FROM o.order_date), EXTRACT(YEAR FROM o.order_date),
s.sku_id, s.sku_name
ORDER BY EXTRACT(YEAR FROM o.order_date), EXTRACT(MONTH FROM o.order_date),
max_review_rating DESC;
```

--Q3--Write a query to create a pivot table that shows the number of users who gave response for all three questions

```
SELECT question_1, question_2, question_3, COUNT(*) AS num_users
FROM Feedback
WHERE question_1 IS NOT NULL AND question_2 IS NOT NULL AND question_3 IS NOT NULL
GROUP BY question_1, question_2, question_3;
```