Code –

--creating a database

create database assignment

use assignment

CREATE TABLE User\_Details (

user\_id INT PRIMARY KEY,

user\_name VARCHAR(255),

birth\_year INT,

nationality VARCHAR(255)

);

INSERT INTO User\_Details (user\_id, user\_name, birth\_year, nationality)

VALUES

(1, 'Aarav Patel', 1988, 'Ind'),

(2, 'Zara Khan', 1995, 'uk'),

(3, 'Ravi Singh', 1982, 'pak'),

(4, 'Meera Sharma', 1990, 'ger'),

(5, 'Priya Verma', 1987, 'rus'),

(6, 'Rahul Desai', 1993, 'chn'),

(7, 'Lila Choudhury', 1984, 'Ind'),

(8, 'Anaya Kapoor', 1989, 'pak'),

(9, 'Kiran Mehra', 1996, 'Ind'),

(10, 'Vikram Gupta', 1981, 'Ind');

CREATE TABLE SKU\_Details (

sku\_id INT PRIMARY KEY,

sku\_name VARCHAR(255)

);

INSERT INTO SKU\_Details (sku\_id, sku\_name)

VALUES

(21, 'LTOP7890'),

(22, 'PHN4567'),

(23, 'HPH2345'),

(24, 'DCAM8910'),

(25, 'DCHR6789'),

(26, 'MTOOL1234'),

(27, 'HSET5678'),

(28, 'BSHB4321'),

(29, 'CKTL7890'),

(30, 'SHOE1234');

CREATE TABLE Order\_Details (

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 User\_Details(user\_id),

FOREIGN KEY (sku\_id) REFERENCES SKU\_Details(sku\_id)

);

INSERT INTO Order\_Details (order\_id, user\_id, sku\_id, amount, quantity, order\_date)

VALUES

(1, 1, 21, 500.00, 2, '2023-01-05'),

(2, 1, 22, 350.50, 1, '2023-02-15'),

(3, 1, 23, 200.75, 3, '2024-03-20'),

(4, 4, 24, 1200.00, 5, '2023-04-10'),

(5, 4, 25, 800.25, 4, '2023-05-18'),

(6, 6, 26, 300.90, 2, '2023-06-25'),

(7, 7, 27, 750.00, 3, '2023-07-12'),

(8, 8, 28, 420.75, 2, '2024-08-05'),

(9, 8, 29, 600.50, 1, '2024-08-28'),

(10, 8, 30, 950.25, 3, '2024-08-30');

CREATE TABLE Feedback (

review\_id INT PRIMARY KEY,

sku\_id INT,

user\_id INT,

order\_id INT,

review\_rating INT,

question\_1 TEXT,

question\_2 TEXT,

question\_3 TEXT,

FOREIGN KEY (sku\_id) REFERENCES SKU\_Details(sku\_id),

FOREIGN KEY (user\_id) REFERENCES User\_Details(user\_id),

FOREIGN KEY (order\_id) REFERENCES Order\_Details(order\_id)

);

INSERT INTO Feedback (review\_id, sku\_id, user\_id, order\_id, review\_rating, question\_1, question\_2, question\_3)

VALUES

(1, 21, 1, 1, 5, 'Excellent service!', 'The product quality is amazing.', null),

(2, 21, 2, 2, 4, 'Good experience overall.', NULL, 'Slightly expensive.'),

(3, 23, 3, 3, 3, 'Average service.', 'Product was okay.', NULL),

(4, 25, 4, 4, 4, 'Outstanding product!', 'Arrived on time.', 'Will buy again for sure.'),

(5, 25, 5, 5, 4, NULL, 'Reasonable price.', 'Had a minor issue with packaging.'),

(6, 26, 6, 6, 2, 'Not happy with the purchase.', NULL, 'Poor customer service response.'),

(7, 27, 7, 7, 3, 'Needs improvement.', NULL, 'Delivery was delayed.'),

(8, 28, 8, 8, 4, 'Satisfied with the purchase.', 'Good customer support.', NULL),

(9, 29, 9, 9, 5, 'Exceptional service!', 'stopped working', 'Will be a repeat customer.'),

(10, 30, 10, 10, NULL, 'Overall good experience.', 'Prompt delivery.', 'Would recommend with minor improvements.');

SELECT user\_id, SUM(amount) AS total\_order\_amount

FROM Order\_Details

WHERE YEAR(order\_date) = 2023

GROUP BY user\_id

HAVING COUNT(order\_id) >= 2;

WITH RankedSKUs AS (

SELECT

f.sku\_id,

YEAR(o.order\_date) AS review\_year,

MONTH(o.order\_date) AS review\_month,

f.review\_rating,

ROW\_NUMBER() OVER (PARTITION BY YEAR(o.order\_date), MONTH(o.order\_date) ORDER BY f.review\_rating DESC) AS rank

FROM

Feedback f

JOIN

Order\_Details o ON f.sku\_id = o.sku\_id

)

SELECT

review\_year,

review\_month,

sku\_id,

review\_rating

FROM

RankedSKUs

WHERE

rank = 1;

SELECT

COUNT(\*) AS num\_users,

SUM(CASE WHEN question\_1 IS NOT NULL AND question\_2 IS NOT NULL AND question\_3 IS NOT NULL THEN 1 ELSE 0 END) AS num\_users\_with\_all\_responses

FROM

Feedback;