## ****Name : ntwari patrick****

**Id : 29089**

**Course : PL/SQL**

## ****Step 1: Problem Definition (2 pts)****

**Business Context:**  
A small-to-medium fruit distribution company supplies fresh fruits (bananas, mangoes, oranges, pineapples, etc.) to households and local shops across different regions in the society.

**Data Challenge (2–3 sentences):**  
The company struggles to identify which fruits generate the highest revenue in each region and season. They also want to understand customer purchasing trends, growth patterns, and segment customers into spending groups to guide marketing campaigns.

**Expected Outcome:**  
Insights into the top 5 fruits by revenue per region/quarter, customer purchase quartiles, and sales trends (running totals, growth rates, and moving averages) to support targeted promotions and better inventory management.

## ****Step 2: Success Criteria (5 measurable goals)****

**Top 5 fruits per region/quarter** → RANK()

**Running monthly sales totals** → SUM() OVER()

**Month-over-month growth** → LAG() / LEAD()

**Customer quartiles by total fruit purchases** → NTILE(4)

**3-month moving average of fruit sales** → AVG() OVER()

## ****Step 3: Database Schema (ERD + Tables)****

### Tables

**customers** → customer info (buyers of fruits)

**fruits** → product catalog (different fruits and categories)

**transactions** → sales (records of each fruit sale)

### Example rows

**customers**: (1001, "Alice", "Kigali")  
**fruits**: (2001, "Banana", "Tropical")  
**transactions**: (3001, 1001, 2001, 2025-01-15, 500)

CREATE TABLE transactions (

transaction\_id NUMBER PRIMARY KEY,

customer\_id NUMBER NOT NULL,

fruit\_id NUMBER NOT NULL,

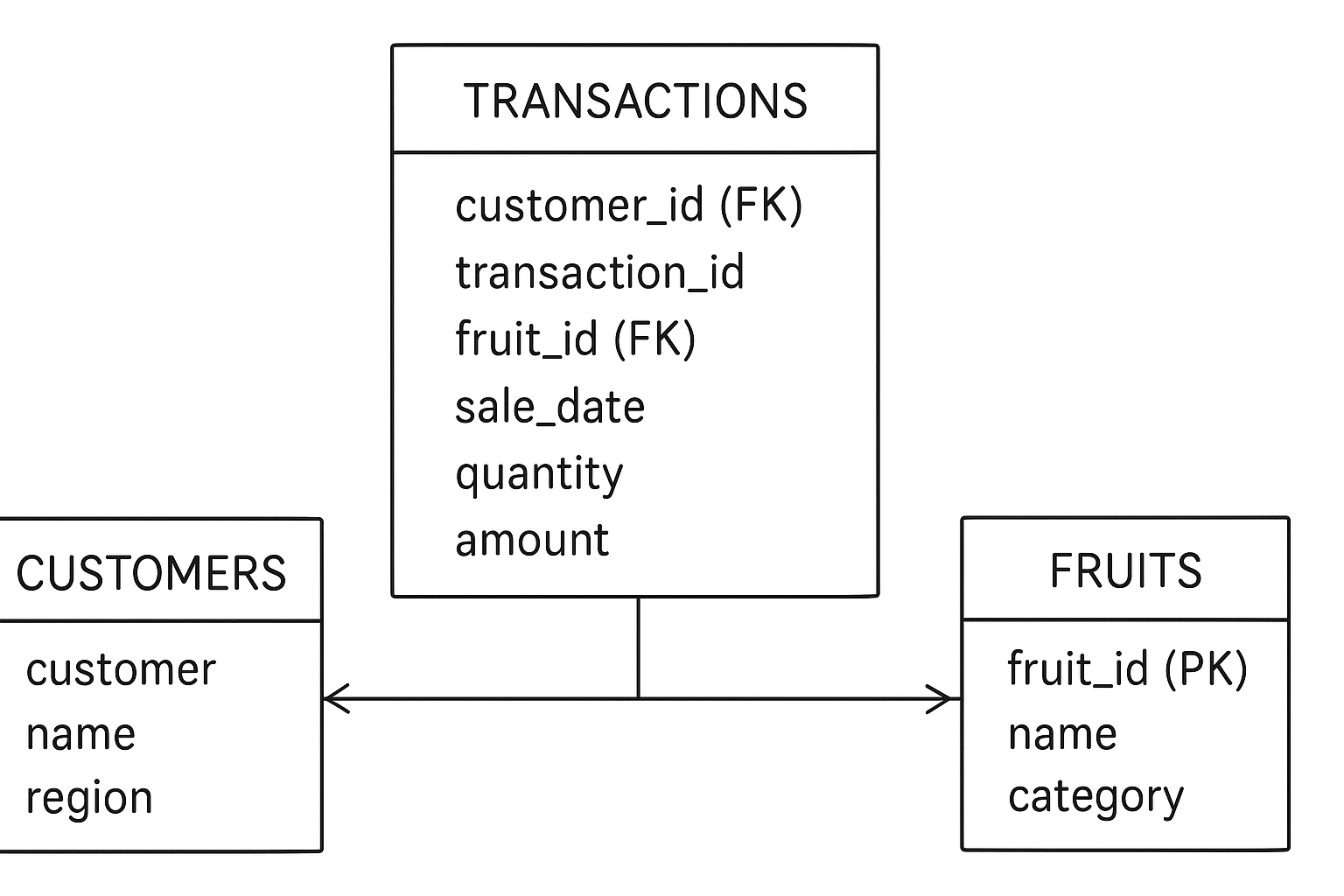
sale\_date DATE NOT NULL,

quantity NUMBER NOT NULL,

amount NUMBER(12,2),

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id)

REFERENCES customers(customer\_id),



CONSTRAINT fk\_fruit FOREIGN KEY (fruit\_id)

REFERENCES fruits(fruit\_id)

## ****Step 4: Window Function Examples (adapted to fruits)****

### A. Ranking – Top fruits per region & quarter

WITH fruit\_sales AS (

SELECT

f.fruit\_id,

f.name AS fruit\_name,

c.region,

TRUNC(t.sale\_date, 'Q') AS quarter\_start,

SUM(t.amount) AS total\_revenue

FROM transactions t

JOIN fruits f ON t.product\_id = f.fruit\_id

JOIN customers c ON t.customer\_id = c.customer\_id

GROUP BY f.fruit\_id, f.name, c.region, TRUNC(t.sale\_date, 'Q')

)SELECT

fruit\_name,

region,

quarter\_start,

total\_revenue,

RANK() OVER (PARTITION BY region, quarter\_start ORDER BY total\_revenue DESC) AS fruit\_rankFROM fruit\_salesWHERE RANK() OVER (PARTITION BY region, quarter\_start ORDER BY total\_revenue DESC) <= 5;

### B. Aggregate – Running monthly totals

SELECT

TRUNC(sale\_date, 'MM') AS month\_start,

SUM(amount) AS month\_total,

SUM(SUM(amount)) OVER (ORDER BY TRUNC(sale\_date,'MM')) AS running\_totalFROM transactionsGROUP BY TRUNC(sale\_date, 'MM')ORDER BY month\_start;

### C. Navigation – Month-over-month growth

WITH monthly\_sales AS (

SELECT TRUNC(sale\_date,'MM') AS month\_start,

SUM(amount) AS total\_sales

FROM transactions

GROUP BY TRUNC(sale\_date,'MM')

)SELECT

month\_start,

total\_sales,

LAG(total\_sales) OVER (ORDER BY month\_start) AS prev\_month,

ROUND(((total\_sales - LAG(total\_sales) OVER (ORDER BY month\_start))

/ NULLIF(LAG(total\_sales) OVER (ORDER BY month\_start),0)) \* 100, 2) AS growth\_pctFROM monthly\_sales;

### D. Distribution – Customer quartiles by spending

WITH customer\_revenue AS (

SELECT c.customer\_id, c.name,

SUM(t.amount) AS total\_spent

FROM customers c

JOIN transactions t ON c.customer\_id = t.customer\_id

GROUP BY c.customer\_id, c.name

)SELECT

customer\_id, name, total\_spent,

NTILE(4) OVER (ORDER BY total\_spent DESC) AS spending\_quartileFROM customer\_revenue;