

## TASK 5

SHRI SHARANG D

TABLE CREATED:

```
SQL> CREATE TABLE online_sales (  
2     order_id INT PRIMARY KEY,  
3     order_date DATE,  
4     amount DECIMAL(10, 2),  
5     product_id INT  
6 );
```

Table created.

INPUTS ARE INSERTED:

```
SQL> INSERT ALL  
2     INTO online_sales VALUES (1001, TO_DATE('2023-01-05', 'YYYY-MM-DD'), 125.50, 101)  
3     INTO online_sales VALUES (1002, TO_DATE('2023-01-12', 'YYYY-MM-DD'), 200.00, 102)  
4     INTO online_sales VALUES (1003, TO_DATE('2023-01-18', 'YYYY-MM-DD'), 75.25, 103)  
5     INTO online_sales VALUES (1004, TO_DATE('2023-02-03', 'YYYY-MM-DD'), 300.00, 101)  
6     INTO online_sales VALUES (1005, TO_DATE('2023-02-10', 'YYYY-MM-DD'), 150.75, 104)  
7     INTO online_sales VALUES (1006, TO_DATE('2023-02-15', 'YYYY-MM-DD'), 225.50, 102)  
8     INTO online_sales VALUES (1007, TO_DATE('2023-03-02', 'YYYY-MM-DD'), 180.00, 103)  
9     INTO online_sales VALUES (1008, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 90.25, 101)  
10    INTO online_sales VALUES (1009, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 310.00, 104)  
11    INTO online_sales VALUES (1010, TO_DATE('2023-03-20', 'YYYY-MM-DD'), 175.75, 102)  
12    SELECT * FROM dual;
```

10 rows created.

BASIC MONTHLY SALARY ANALYSIS:

```
SQL> SELECT  
2     EXTRACT(YEAR FROM order_date) AS year,  
3     EXTRACT(MONTH FROM order_date) AS month,  
4     SUM(amount) AS monthly_revenue,  
5     COUNT(DISTINCT order_id) AS order_volume  
6 FROM  
7     online_sales  
8 GROUP BY  
9     EXTRACT(YEAR FROM order_date),  
10    EXTRACT(MONTH FROM order_date)  
11 ORDER BY  
12     year ASC,  
13     month ASC;
```

YEAR	MONTH	MONTHLY_REVENUE	ORDER_VOLUME
2023	1	400.75	3
2023	2	676.25	3
2023	3	756	4

## TIME PERIOD ANALYSIS:

```
SQL> SELECT
  2     EXTRACT(YEAR FROM order_date) AS year,
  3     EXTRACT(MONTH FROM order_date) AS month,
  4     SUM(amount) AS monthly_revenue,
  5     COUNT(DISTINCT order_id) AS order_volume,
  6     ROUND(SUM(amount) / COUNT(DISTINCT order_id), 2) AS avg_order_value
  7 FROM
  8     online_sales
  9 WHERE
10     order_date BETWEEN TO_DATE('2023-01-01', 'YYYY-MM-DD')
11     AND TO_DATE('2023-03-31', 'YYYY-MM-DD')
12 GROUP BY
13     EXTRACT(YEAR FROM order_date),
14     EXTRACT(MONTH FROM order_date)
15 ORDER BY
16     year ASC,
17     month ASC;
```

YEAR	MONTH	MONTHLY_REVENUE	ORDER_VOLUME	AVG_ORDER_VALUE
2023	1	400.75	3	133.58
2023	2	676.25	3	225.42
2023	3	756	4	189

## PRODUCT PERFORMANCE BY MONTH:

```
SQL> SELECT
  2     EXTRACT(YEAR FROM order_date) AS year,
  3     EXTRACT(MONTH FROM order_date) AS month,
  4     product_id,
  5     SUM(amount) AS product_revenue,
  6     COUNT(DISTINCT order_id) AS order_count
  7 FROM
  8     online_sales
  9 GROUP BY
10     EXTRACT(YEAR FROM order_date),
11     EXTRACT(MONTH FROM order_date),
12     product_id
13 ORDER BY
14     year ASC,
15     month ASC,
16     product_revenue DESC;
```

YEAR	MONTH	PRODUCT_ID	PRODUCT_REVENUE	ORDER_COUNT
2023	1	102	200	1
2023	1	101	125.5	1
2023	1	103	75.25	1
2023	2	101	300	1
2023	2	102	225.5	1
2023	2	104	150.75	1
2023	3	104	310	1
2023	3	103	180	1
2023	3	102	175.75	1
2023	3	101	90.25	1

10 rows selected.

TOP PERFORMING MONTHS:

```
SQL> SELECT
  2     EXTRACT(YEAR FROM order_date) AS year,
  3     EXTRACT(MONTH FROM order_date) AS month,
  4     SUM(amount) AS monthly_revenue,
  5     RANK() OVER (ORDER BY SUM(amount) DESC) AS revenue_rank
  6 FROM
  7     online_sales
  8 GROUP BY
  9     EXTRACT(YEAR FROM order_date),
10     EXTRACT(MONTH FROM order_date)
11 ORDER BY
12     monthly_revenue DESC
13 FETCH FIRST 3 ROWS ONLY;
```

YEAR	MONTH	MONTHLY_REVENUE	REVENUE_RANK
2023	3	756	1
2023	2	676.25	2
2023	1	400.75	3