TASK-2

DATA ANALYSIS WITH COMPLEX QUERIES

Introduction

Analyzing sales data effectively is crucial for identifying trends, improving performance, and making informed business decisions. This guide provides a complete SQL script for creating a sample sales dataset and executing advanced queries to uncover meaningful insights. You can use this as a practice exercise or adapt it to your real-world datasets.

Dataset Creation and Overview

The dataset models a simplified sales record, including:

- sale_id: Unique identifier for each sale.
- sale_date: The date of the sale.
- product_id: ID of the product sold.
- region: Region where the sale occurred.
- sales_amount: The monetary value of the sale.

Steps:

1. Create the sales_data Table: Use the CREATE TABLE statement to structure the dataset.

2. Insert Sample Data: Populate the table with diverse sales data across regions, products, and time periods.

Advanced Query Applications

A. Monthly Sales Trend with Growth:

This query calculates total sales per month, compares it to the previous month, and computes the growth or decline.

B. Top 5 Regions by Total Sales:

Identifies regions generating the highest sales.

Use Case: Allocate resources to high-performing regions or address underperforming ones.

C. Sales Rank by Product:

Ranks products based on their total sales volume.

Use Case: Highlight best-sellers and underperforming products to refine

D. Regional Sales Distribution:

Calculates the percentage contribution of each region to the overall sales.

Use Case: Gauge regional market share and prioritize regions with higher growth potential.

E. Running Total of Sales by Region:

Provides a cumulative view of sales within each region over time.

Use Case: Evaluate regional growth trends and track progress towards targets.

Insights and Recommendations

1. Visualization Techniques:

- Monthly Sales Trends: Line charts to highlight changes over time.
- Regional Sales Distribution: Pie charts to emphasize percentage contributions.
- Top Performing Products: Bar charts to rank sales figures.

2. Report Writing:

- Summarize key findings, such as which months, regions, or products performed best.
- Provide actionable insights, e.g., focus on expanding in regions with high sales growth or optimizing inventory for top products.

Coding

SQL*Plus: Release 21.0.0.0.0 - Production on Sat Jan 18 10:16:03 2025

Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Enter user-name: system

Enter password:

Last Successful login time: Sat Jan 18 2025 10:15:22 +05:30

Connected to:

Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production

Version 21.3.0.0.0

#CREATING TABLE

```
SQL> create table sales_data(
  2  sale_id number primary key,
  3  sale_date date not null,
  4  product_id int not null,
  5  region varchar(50) not null,
  6  sales_amount decimal(10,2) not null);
```

Table created.

#DESCRIBING TABLE

SQL> desc sales_data;

Name	Null?	Type
SALE_ID	NOT NULL	NUMBER
SALE_DATE	NOT NULL	DATE
PRODUCT_ID	NOT NULL	NUMBER(38)
REGION	NOT NULL	VARCHAR2(50)
SALES_AMOUNT	NOT NULL	NUMBER(10,2)

#INSERTING VALUES

SQL> insert into sales_data values(001,to_date('2025-01-01','yy-mm-dd'),1,'NORTH',500.00);

1 row created.

SQL> insert into sales_data values(002,to_date('2025-01-02','yy-mm-dd'),2,'SOUTH',300.00);

1 row created.

```
SQL> insert into sales_data values(003,to_date('2025-01-03','yy-mm-
dd'),1,'NORTH',450.00);

1 row created.

SQL> insert into sales_data values(004,to_date('2025-01-03','yy-mm-
dd'),3,'WEST',200.00);

1 row created.

SQL> insert into sales_data values(005,to_date('2025-01-04','yy-mm-
dd'),2,'SOUTH',350.00);

1 row created.

SQL> insert into sales_data values(006,to_date('2025-01-05','yy-mm-
dd'),3,'EAST',400.00);

1 row created.

SQL> insert into sales_data values(007,to_date('2025-02-01','yy-mm-
dd'),1,'NORTH',600.00);
```

1 row created.

SQL> insert into sales_data values(008,to_date('2025-02-02','yy-mm-dd'),2,'SOUTH',500.00);

1 row created.

SQL> insert into sales_data values(009,to_date('2025-02-03','yy-mm-dd'),3,'WEST',700.00);

1 row created.

SQL> insert into sales_data values(010,to_date('2025-02-04','yy-mm-dd'),3,'EAST',650.00);

1 row created.

SQL> insert into sales_data values(011,to_date('2025-02-05','yy-mm-dd'),2,'NORTH',300.00);

1 row created.

#SQL QUERIES

(1)MONTHLY SALES TREND WITH GROWTH:

```
SQL> with monthlysales as (select trunc(sale_date,'month') as month,
  2  sum (sales_amount) as total_sales from sales_data group by
trunc(sale_date,'month'))
  3  select month,total_sales,lag (total_sales) over (order by month) as
previous_month_sales,
  4  total_sales - coalesce (lag (total_sales) over (order by month),0) as
sales_change
  5  from monthlysales order by month;
```

MONTH	TOTAL_SALES	PREVIOUS_MONTH_SALES	SALES_CHANGE
01-JAN-25	2200		2200
01-FEB-25	2750	2200	550

(2)TOP 5 REGION BY TOTAL SALES:

```
SQL> select region,sum(sales_amount)as total_sales from sales_data
  2  group by region order by total_sales desc fetch first 5 rows only;
```

REGION TOTAL_SALES

---------- ----------

NORTH 1850

SOUTH 1150

EAST 1050

WEST 900

(3)SALES RANK BY PRODUCT:

SQL> with productsales as (select product_id,sum (sales_amount) as total_sales

2 from sales_data group by product_id) select product_id,total_sales,rank()over

3 (order by total_sales desc) as sales_rank from productsales;

no rows selected

(4)REGIONAL SALES DISTRIBUTION:

SQL> select region,sum(sales_amount)as total_sales,

2 round(sum(sales_amount)*100.0/

3 sum(sum(sales_amount))over(),2)as sales_percentage

```
  4  from sales_data group by region order by total_sales desc;

no rows selected
```

(5)RUNNING TOTAL OF SALES BY REGION:

```
SQL> select region,sale_date,sum(sales_amount)over(partition by
region
  2  order by sale_date)as running_total from sales_data
  3  order by region,sale_date;

no rows selected
```

Output

```
SQL*Plus: Release 21.0.0.0.0 - Production on Sat Jan 18 10:16:03 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Sat Jan 18 2025 10:15:22 +05:30

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> create table sales_data(
  2  sale_id number primary key,
  3  sale_date date not null,
  4  product_id int not null,
  5  region varchar(50) not null,
  6  sales_amount decimal(10,2) not null);

Table created.

SQL> desc sales_data;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALE_ID                                   NOT NULL NUMBER
 SALE_DATE                                 NOT NULL DATE
 PRODUCT_ID                                NOT NULL NUMBER(38)
 REGION                                    NOT NULL VARCHAR2(50)
 SALES_AMOUNT                              NOT NULL NUMBER(10,2)
```

```
SQL> insert into sales_data values(001,to_date('2025-01-01','yy-mm-dd'),1,'NORTH',500.00);

1 row created.

SQL> insert into sales_data values(002,to_date('2025-01-02','yy-mm-dd'),2,'SOUTH',300.00);

1 row created.

SQL> insert into sales_data values(003,to_date('2025-01-03','yy-mm-dd'),1,'NORTH',450.00);

1 row created.

SQL> insert into sales_data values(004,to_date('2025-01-03','yy-mm-dd'),3,'WEST',200.00);

1 row created.

SQL> insert into sales_data values(005,to_date('2025-01-04','yy-mm-dd'),2,'SOUTH',350.00);

1 row created.

SQL> insert into sales_data values(006,to_date('2025-01-05','yy-mm-dd'),3,'EAST',400.00);

1 row created.

SQL> insert into sales_data values(007,to_date('2025-02-01','yy-mm-dd'),1,'NORTH',600.00);

1 row created.

SQL> insert into sales_data values(008,to_date('2025-02-02','yy-mm-dd'),2,'SOUTH',500.00);

1 row created.

SQL> insert into sales_data values(009,to_date('2025-02-03','yy-mm-dd'),3,'WEST',700.00);

1 row created.

SQL> insert into sales_data values(010,to_date('2025-02-04','yy-mm-dd'),3,'EAST',650.00);

1 row created.
```

```
SQL> with monthlysales as (select trunc(sale_date,'month') as month,
  2  sum (sales_amount) as total_sales from sales_data group by trunc(sale_date,'month'))
  3  select month,total_sales,lag (total_sales) over (order by month) as previous_month_sales,
  4  total_sales - coalesce (lag (total_sales) over (order by month),0) as sales_change
  5  from monthlysales order by month;

MONTH     TOTAL_SALES PREVIOUS_MONTH_SALES SALES_CHANGE
--------- ----------- -------------------- ------------
01-JAN-25        2200                            2200
01-FEB-25        2750                 2200         550
```

```
SQL> select region,sum(sales_amount)as total_sales from sales_data
  2  group by region order by total_sales desc fetch first 5 rows only;

REGION                                             TOTAL_SALES
-------------------------------------------------- -----------
NORTH                                                     1850
SOUTH                                                     1150
EAST                                                      1050
WEST                                                       900
```

```
SQL> with productsales as (select product_id,sum (sales_amount) as total_sales
  2  from sales_data group by product_id) select product_id,total_sales,rank()over
  3  (order by total_sales desc) as sales_rank from productsales;

no rows selected
```

```
SQL> select region,sum(sales_amount)as total_sales,
  2  round(sum(sales_amount)*100.0/
  3  sum(sum(sales_amount))over(),2)as sales_percentage
  4  from sales_data group by region order by total_sales desc;

no rows selected
```

```
SQL> select region,sale_date,sum(sales_amount)over(partition by region
  2  order by sale_date)as running_total from sales_data
  3  order by region,sale_date;

no rows selected
```