Task 2: Data Analysis with Complex Queries

1. Introduction

In this task, we dive into advanced SQL techniques to analyze sales data. The focus is on using window functions, subqueries, and Common Table Expressions (CTEs) to extract meaningful insights from relational data.

2. Data Overview

To demonstrate these techniques, we use two tables: one containing sales records and the other listing employee information.

sales table:

sale_id	emp_id	sale_date	amount
1	101	2023-01-05	1000
2	101	2023-01-10	1500
3	102	2023-01-15	1200
4	103	2023-01-20	1300

employees table:

emp_id	emp_name	department
101	Alice	Sales
102	Bob	Marketing
103	Carol	Sales

3. SQL Queries and Analysis

A. Subquery

We start by identifying sales that are greater than the average sale amount.

```sql
SELECT emp_id, sale_date, amount
FROM sales
WHERE amount > (
    SELECT AVG(amount) FROM sales
);
```

Output:

emp_id	sale_date	amount
101	2023-01-10	1500
103	2023-01-20	1300

B. Common Table Expression (CTE)

Calculate total sales amount by each employee.

```sql
WITH total_sales AS (
    SELECT emp_id, SUM(amount) AS total_amount
    FROM sales
    GROUP BY emp_id
)
SELECT e.emp_name, t.total_amount
FROM total_sales t
JOIN employees e ON t.emp_id = e.emp_id;
```

Output:

| emp_name | total_amount |

|----------|--------------|

| Alice | 2500 |

| Bob | 1200 |

| Carol | 1300 |

C. Window Function

Show running total of sales for each employee.

```sql
SELECT
    emp_id,
    sale_date,
    amount,
    SUM(amount) OVER (PARTITION BY emp_id ORDER BY sale_date) AS running_total
FROM sales;
```

Output:

| emp_id | sale_date | amount | running_total |

|--------|------------|--------|---------------|

| 101 | 2023-01-05 | 1000 | 1000 |

| 101 | 2023-01-10 | 1500 | 2500 |

| 102 | 2023-01-15 | 1200 | 1200 |

| 103 | 2023-01-20 | 1300 | 1300 |

D. Monthly Trend Analysis

Calculate average monthly sales.

```sql
SELECT
    DATE_TRUNC('month', sale_date) AS sale_month,
    AVG(amount) AS avg_monthly_sale
FROM sales
GROUP BY sale_month
ORDER BY sale_month;
```

Output:

sale_month	avg_monthly_sale
2023-01-01	1250.00