# Coffee Shop Sales Analysis Project

## 1. Introduction

The purpose of this project is to analyze coffee shop sales data using SQL, focusing on extracting key insights, cleaning and transforming the data, and generating meaningful reports.

## 2. Database Setup

### 2.1 Database Creation

```sql  
CREATE DATABASE coffee\_shop;  
```

### 2.2 Importing the Data

```sql  
SELECT \* FROM coffee\_shop\_sales;  
DESCRIBE coffee\_shop\_sales;  
```

## 3. Data Cleaning and Transformation

### 3.1 Formatting and Changing Data Types

#### 3.1.1 Formatting `transaction\_date`

```sql  
SET SQL\_SAFE\_UPDATES = 0;  
UPDATE coffee\_shop\_sales  
SET transaction\_date = STR\_TO\_DATE(transaction\_date, '%d-%m-%Y');  
ALTER TABLE coffee\_shop\_sales  
MODIFY COLUMN transaction\_date DATE;  
```

#### 3.1.2 Formatting `transaction\_time`

```sql  
UPDATE coffee\_shop\_sales  
SET transaction\_time = STR\_TO\_DATE(transaction\_time, '%H:%i:%s');  
ALTER TABLE coffee\_shop\_sales  
MODIFY COLUMN transaction\_time TIME;  
```

#### 3.1.3 Renaming a Column

```sql  
ALTER TABLE coffee\_shop\_sales  
CHANGE COLUMN ï»¿transaction\_id transaction\_id INT;  
```

## 4. Data Analysis and Insights

### 4.1 Month-Wise Total Sales

```sql  
SELECT MONTHNAME(transaction\_date), SUM(transaction\_qty \* unit\_price) AS total\_sales  
FROM coffee\_shop\_sales  
GROUP BY MONTHNAME(transaction\_date);  
```

### 4.2 Month-on-Month Sales Difference (%)

```sql  
DELIMITER //  
CREATE FUNCTION get\_salesdiff\_from\_previous\_month(month\_number INT)   
RETURNS DECIMAL(10,2)  
DETERMINISTIC  
BEGIN  
 DECLARE sales\_diff DECIMAL(10,2);  
 WITH monthly\_sales AS (  
 SELECT MONTH(transaction\_date) AS month\_no, SUM(transaction\_qty \* unit\_price) AS total\_sales  
 FROM coffee\_shop\_sales  
 GROUP BY MONTH(transaction\_date)  
 )  
 SELECT ((SELECT total\_sales  
 FROM monthly\_sales  
 WHERE month\_no = month\_number) -   
 (SELECT total\_sales  
 FROM monthly\_sales  
 WHERE month\_no = (month\_number-1))) /   
 (SELECT total\_sales  
 FROM monthly\_sales  
 WHERE month\_no = (month\_number-1)) \* 100   
 INTO sales\_diff;  
 RETURN sales\_diff;  
END;  
//  
DELIMITER ;  
```

#### Example of Usage:

```sql  
SELECT get\_salesdiff\_from\_previous\_month(2) AS feb\_d;  
```

### 4.3 Number of Orders Per Month

```sql  
SELECT MONTHNAME(transaction\_date) AS month\_name, COUNT(transaction\_id)  
FROM coffee\_shop\_sales  
GROUP BY MONTHNAME(transaction\_date);  
```

### 4.4 Month-on-Month Order Difference

```sql  
SELECT MONTH(transaction\_date) AS month\_no,   
 COUNT(transaction\_id) AS orders,  
 COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1) OVER(ORDER BY MONTH(transaction\_date)) AS orders\_diff,  
 (COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1) OVER(ORDER BY MONTH(transaction\_date))) /  
 LAG(COUNT(transaction\_id), 1) OVER(ORDER BY MONTH(transaction\_date)) \* 100 AS orders\_diff\_perc  
FROM coffee\_shop\_sales  
GROUP BY MONTH(transaction\_date);  
```

## 5. Advanced Analysis

## 6. Key Reports

## 7. Conclusion

This project demonstrates how SQL can be used for effective data cleaning, transformation, and generating actionable insights. It includes key performance metrics such as sales trends, order volume changes, and product category performance.