

# DATA MINING LAB 2

Name: Uday Bolla

Roll No :22cs3024

Branch: Cse

## Query for Creating Table and Inserting Values :

#Creating table

USE my\_database;

```
CREATE TABLE FactSales (  
    DateKey INT,  
    ProductKey INT,  
    CustomerKey INT,  
    StoreKey INT,  
    QtySold INT,  
    SalesAmount DECIMAL(10, 2),  
    DiscountAmount DECIMAL(10, 2),  
    TaxAmount DECIMAL(10, 2),  
    NetSalesAmount DECIMAL(10, 2)  
);
```

```
CREATE TABLE DimProduct (  
    ProductKey INT PRIMARY KEY,  
    ProductName VARCHAR(100),  
    Category VARCHAR(50),  
    SubCategory VARCHAR(50),  
    Brand VARCHAR(50),  
    Price DECIMAL(10, 2)  
);
```

```
CREATE TABLE DimCustomer (  
    CustomerKey INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Email VARCHAR(100),  
    PhoneNo VARCHAR(15),  
    Address VARCHAR(200),  
    City VARCHAR(50),  
    State VARCHAR(50),  
    Country VARCHAR(50),  
    LoyaltyLevel VARCHAR(50)
```

```
);
```

```
CREATE TABLE DimDate (  
    DateKey INT PRIMARY KEY,  
    FullDate DATE,  
    Day INT,  
    Month INT,  
    Year INT,  
    DayName VARCHAR(15),  
    MonthName VARCHAR(15),  
    Quarter INT  
);
```

```
CREATE TABLE DimStore (  
    StoreKey INT PRIMARY KEY,  
    Region VARCHAR(50),  
    ManagerName VARCHAR(50),  
    OpeningDate DATE  
);
```

#### # Inserting Values

```
INSERT INTO DimProduct (ProductKey, ProductName, Category, SubCategory, Brand, Price) SELECT  
1, 'Laptop', 'Electronics', 'Computers', 'BrandA', 1000.00  
FROM DUAL  
WHERE NOT EXISTS (SELECT 1 FROM DimProduct WHERE ProductKey = 1);
```

```
INSERT INTO DimProduct (ProductKey, ProductName, Category, SubCategory, Brand, Price) SELECT  
2, 'Smartphone', 'Electronics', 'Mobile Phones', 'BrandB', 500.00  
FROM DUAL  
WHERE NOT EXISTS (SELECT 1 FROM DimProduct WHERE ProductKey = 2);
```

```
INSERT INTO DimProduct (ProductKey, ProductName, Category, SubCategory, Brand, Price) SELECT  
3, 'Tablet', 'Electronics', 'Tablets', 'BrandC', 300.00  
FROM DUAL  
WHERE NOT EXISTS (SELECT 1 FROM DimProduct WHERE ProductKey = 3);
```

```
INSERT INTO DimCustomer (CustomerKey, FirstName, LastName, Email, PhoneNo, Address, City, State, Country, LoyaltyLevel)  
SELECT 1, 'John', 'Doe', 'john@example.com', '1234567890', '123 Main St', 'New York', 'NY', 'USA', 'Gold'  
FROM DUAL  
WHERE NOT EXISTS (SELECT 1 FROM DimCustomer WHERE CustomerKey = 1);
```

```
INSERT INTO DimCustomer (CustomerKey, FirstName, LastName, Email, PhoneNo, Address, City, State, Country, LoyaltyLevel)  
SELECT 2, 'Jane', 'Smith', 'jane@example.com', '0987654321', '456 Elm St', 'Los Angeles', 'CA', 'USA', 'Silver'  
FROM DUAL  
WHERE NOT EXISTS (SELECT 1 FROM DimCustomer WHERE CustomerKey = 2);
```

```
INSERT INTO DimDate (DateKey, FullDate, Day, Month, Year, DayName, MonthName, Quarter) SELECT  
20230101, '2023-01-01', 1, 1, 2023, 'Sunday', 'January', 1  
FROM DUAL
```

```
WHERE NOT EXISTS (SELECT 1 FROM DimDate WHERE DateKey = 20230101);
```

```
INSERT INTO DimDate (DateKey, FullDate, Day, Month, Year, DayName, MonthName, Quarter)
SELECT 20230102, '2023-01-02', 2, 1, 2023, 'Monday', 'January', 1
FROM DUAL
WHERE NOT EXISTS (SELECT 1 FROM DimDate WHERE DateKey = 20230102);
```

```
INSERT INTO DimStore (StoreKey, Region, ManagerName, OpeningDate)
SELECT 1, 'North', 'Alice', '2020-01-01'
FROM DUAL
WHERE NOT EXISTS (SELECT 1 FROM DimStore WHERE StoreKey = 1);
```

```
INSERT INTO DimStore (StoreKey, Region, ManagerName, OpeningDate)
SELECT 2, 'South', 'Bob', '2021-01-01'
FROM DUAL
WHERE NOT EXISTS (SELECT 1 FROM DimStore WHERE StoreKey = 2);
```

```
INSERT INTO FactSales (DateKey, ProductKey, CustomerKey, StoreKey, QtySold, SalesAmount, DiscountAmount, TaxAmount,
NetSalesAmount)
SELECT 20230101, 1, 1, 1, 2, 2000.00, 100.00, 180.00, 1720.00
FROM DUAL
WHERE NOT EXISTS (SELECT 1 FROM FactSales WHERE DateKey = 20230101 AND ProductKey = 1 AND CustomerKey = 1 AND
StoreKey = 1);
```

```
INSERT INTO FactSales (DateKey, ProductKey, CustomerKey, StoreKey, QtySold, SalesAmount, DiscountAmount, TaxAmount,
NetSalesAmount)
SELECT 20230102, 2, 2, 2, 3, 1500.00, 50.00, 135.00, 1315.00
FROM DUAL
```

```
WHERE NOT EXISTS (SELECT 1 FROM FactSales WHERE DateKey = 20230102 AND ProductKey = 2 AND CustomerKey = 2
AND StoreKey = 2);
```

## 4. Distributive and Algebraic Functions

### Total Sales

```
SELECT SUM(SalesAmount) AS TotalSales
FROM FactSales;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Co
	TotalSales			
▶	7000.00			

## Count of Transactions Per Store

```
SELECT
  StoreKey,
  COUNT(*) AS TransactionCount
FROM FactSales
GROUP BY StoreKey;
```

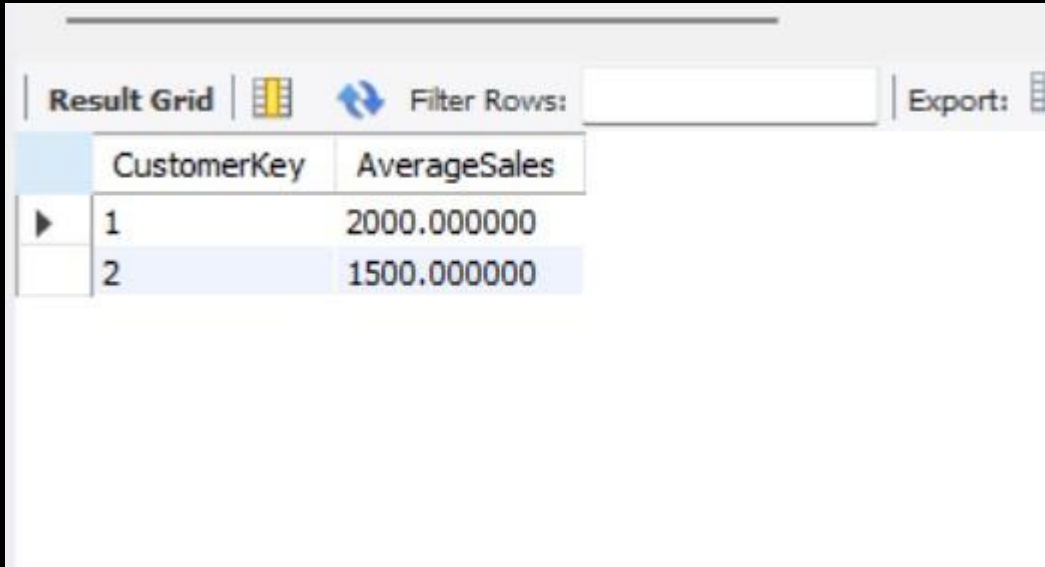
Result Grid		Filter Rows:	Export:
	StoreKey	TransactionCount	
▶	1	2	
	2	2	

## Average Sales Per Customer

```

SELECT
    CustomerKey,
    AVG(SalesAmount) AS AverageSales
FROM FactSales
GROUP BY CustomerKey;

```



The screenshot shows a 'Result Grid' window with a toolbar containing icons for 'Filter Rows' and 'Export'. The grid displays the results of the SQL query above, with columns 'CustomerKey' and 'AverageSales'.

	CustomerKey	AverageSales
▶	1	2000.000000
	2	1500.000000

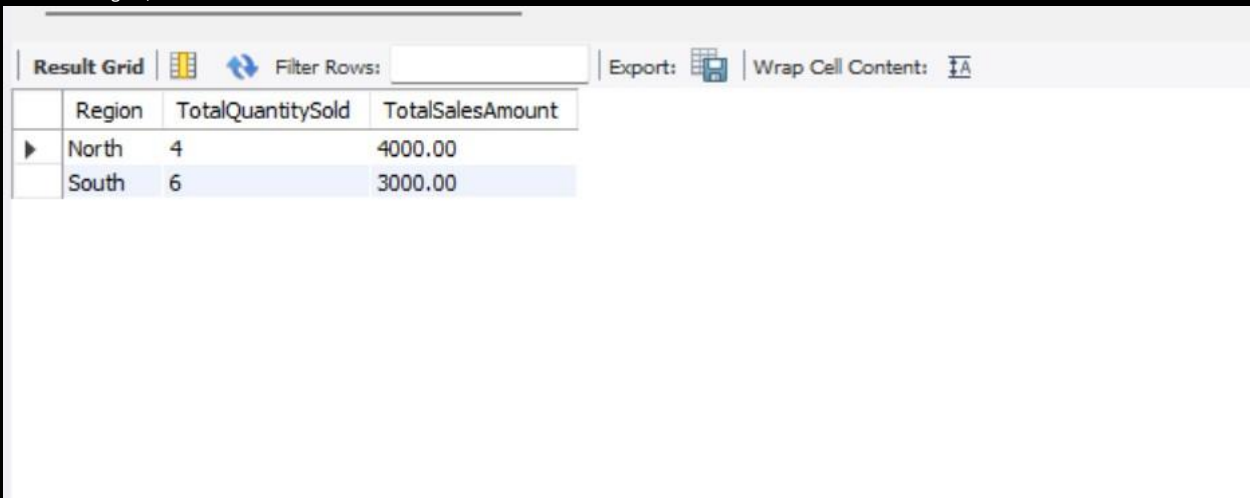
## 5. Summarisability

### Aggregate Data at the Region Level

```

SELECT
    s.Region,
    SUM(f.QtySold) AS TotalQuantitySold,
    SUM(f.SalesAmount) AS TotalSalesAmount
FROM FactSales f
JOIN DimStore s ON f.StoreKey = s.StoreKey
GROUP BY s.Region;

```

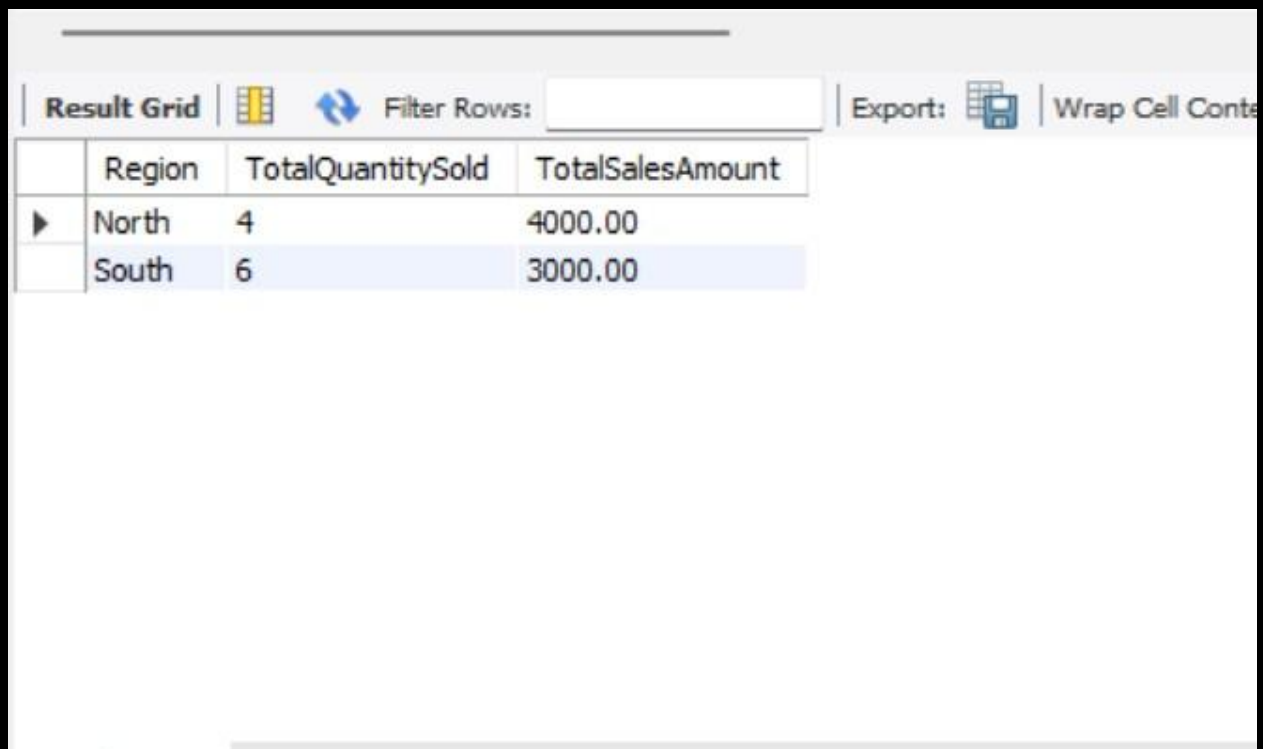


The screenshot shows a 'Result Grid' window with a toolbar containing icons for 'Filter Rows', 'Export', and 'Wrap Cell Content'. The grid displays the results of the SQL query above, with columns 'Region', 'TotalQuantitySold', and 'TotalSalesAmount'.

	Region	TotalQuantitySold	TotalSalesAmount
▶	North	4	4000.00
	South	6	3000.00

## Aggregate Data at the Store Level

```
SELECT
    s.Region,
    SUM(StoreTotal.TotalQuantitySold) AS TotalQuantitySold,
    SUM(StoreTotal.TotalSalesAmount) AS TotalSalesAmount
FROM (
    SELECT
        f.StoreKey,
        SUM(f.QtySold) AS TotalQuantitySold,
        SUM(f.SalesAmount) AS TotalSalesAmount
    FROM FactSales f
    GROUP BY f.StoreKey
) AS StoreTotal
JOIN DimStore s ON StoreTotal.StoreKey = s.StoreKey
GROUP BY s.Region;
```



	Region	TotalQuantitySold	TotalSalesAmount
▶	North	4	4000.00
	South	6	3000.00

## 7. Drill Across

```
SELECT
    p.ProductName AS Product,
    s.Region AS Region,
    SUM(CASE WHEN d.Year = 2023 THEN f.SalesAmount ELSE 0 END) AS Sales_2023,
    SUM(CASE WHEN d.Year = 2024 THEN f.SalesAmount ELSE 0 END) AS Sales_2024,
    (SUM(CASE WHEN d.Year = 2023 THEN f.SalesAmount ELSE 0 END) -
     SUM(CASE WHEN d.Year = 2024 THEN f.SalesAmount ELSE 0 END)) AS Sales_Diff
FROM FactSales f
JOIN DimDate d ON f.DateKey = d.DateKey
JOIN DimStore s ON f.StoreKey = s.StoreKey
JOIN DimProduct p ON f.ProductKey = p.ProductKey
WHERE d.Year IN (2022, 2023)
```

GROUP BY p.ProductName, s.Region  
ORDER BY s.Region, p.ProductName;

Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	Product	Region	Sales_2023	Sales_2024	Sales_Diff
▶	Laptop	North	4000.00	0.00	4000.00
	Smartphone	South	3000.00	0.00	3000.00

## 10. Range Query and Range Sum Query

Query to retrieve sales data for all transactions between two dates:

```
SELECT
    *
FROM FactSales f
JOIN DimDate d ON f.DateKey = d.DateKey
WHERE d.FullDate BETWEEN '2023-01-01' AND '2023-12-31'
ORDER BY d.FullDate;
```

Result Grid									
Filter Rows:									
Export:									
Wrap Cell Content:									
	DateKey	ProductKey	CustomerKey	StoreKey	QtySold	SalesAmount	DiscountAmount	TaxAmount	NetSalesAmount
▶	20230101	1	1	1	2	2000.00	100.00	180.00	1720.00
	20230101	1	1	1	2	2000.00	100.00	180.00	1720.00
	20230102	2	2	2	3	1500.00	50.00	135.00	1315.00
	20230102	2	2	2	3	1500.00	50.00	135.00	1315.00

Query to calculate the range sum of sales for a specific product category over a given time period:

```
SELECT
    p.ProductName,
    SUM(f.SalesAmount) AS TotalSales
FROM FactSales f
JOIN DimProduct p ON f.ProductKey = p.ProductKey
JOIN DimDate d ON f.DateKey = d.DateKey
WHERE p.Category = 'Electronics' -- Specify the product category
AND d.FullDate BETWEEN '2023-01-01' AND '2023-12-31'
GROUP BY p.ProductName
```



ORDER BY TotalSales DESC;

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	ProductName	TotalSales		
▶	Laptop	4000.00		
	Smartphone	3000.00		

## 11. Lattice of Cuboids

```
SELECT
  'all' AS product,
  'all' AS region,
  'all' AS time,
  SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimProduct p ON f.ProductKey = p.ProductKey
JOIN DimStore s ON f.StoreKey = s.StoreKey
JOIN DimDate d ON f.DateKey = d.DateKey;
```

Result Grid

Filter Rows:




Export

	product	region	time	total_sales
▶	all	all	all	7000.00

### One-Dimensional Cuboids

```
SELECT
  'all' AS product,
  'all' AS region,
  'all' AS time,
  SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimProduct p ON f.ProductKey = p.ProductKey
JOIN DimStore s ON f.StoreKey = s.StoreKey
JOIN DimDate d ON f.DateKey = d.DateKey
GROUP BY p.ProductName;
```



Result Grid   Filter Rows:  | Export:  | Wrap




	product	region	time	total_sales
▶	all	all	all	4000.00
	all	all	all	3000.00

all

```

SELECT
    'all' AS product,
    s.Region AS region,
    'all' AS time,
    SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimStore s ON f.StoreKey = s.StoreKey
JOIN DimDate d ON f.DateKey = d.DateKey
GROUP BY s.Region;

```

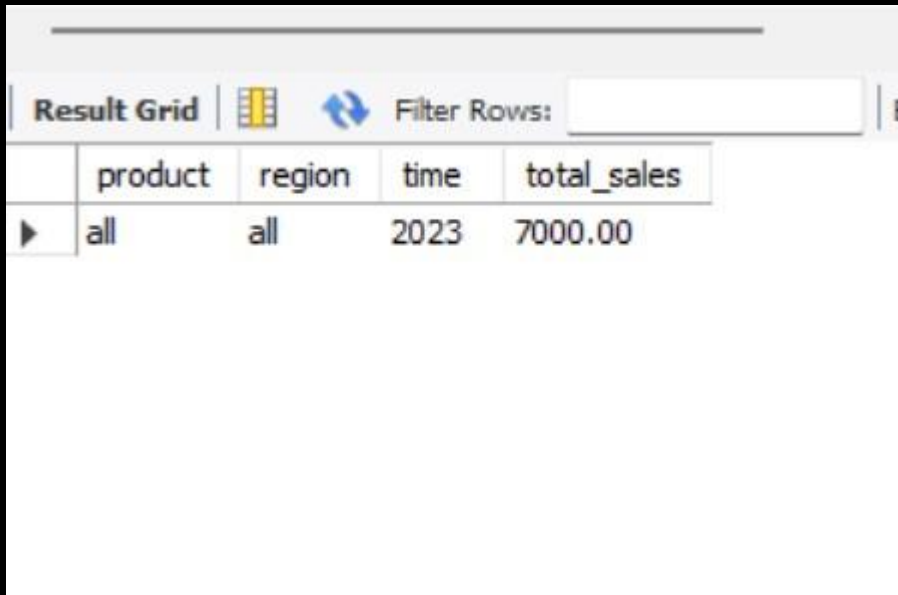
Result Grid   Filter Rows:  | Export: 

	product	region	time	total_sales
▶	all	North	all	4000.00
	all	South	all	3000.00

```

SELECT
    'all' AS product,
    'all' AS region,
    d.Year AS time,
    SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimDate d ON f.DateKey = d.DateKey
GROUP BY d.Year;

```



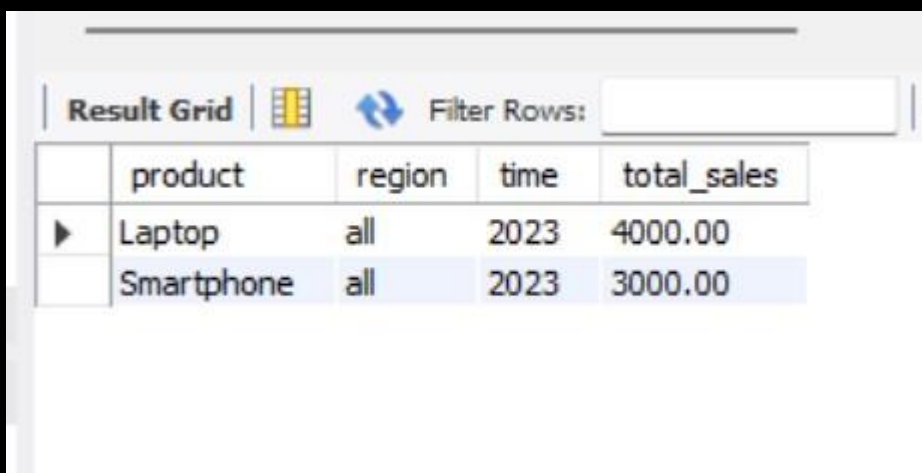
The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains one row of data with the following columns: product, region, time, and total\_sales.

	product	region	time	total_sales
▶	all	all	2023	7000.00

```

SELECT
    p.ProductName AS product,
    'all' AS region,
    d.Year AS time,
    SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimProduct p ON f.ProductKey = p.ProductKey
JOIN DimDate d ON f.DateKey = d.DateKey
GROUP BY p.ProductName, d.Year;

```





The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two rows of data with the following columns: product, region, time, and total\_sales.

	product	region	time	total_sales
▶	Laptop	all	2023	4000.00
	Smartphone	all	2023	3000.00

```

SELECT
    p.ProductName AS product,
    s.Region AS region,
    'all' AS time,
    SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimProduct p ON f.ProductKey = p.ProductKey
JOIN DimStore s ON f.StoreKey = s.StoreKey
GROUP BY p.ProductName, s.Region;



```

Result Grid   Filter Rows: <input type="text"/>				
	product	region	time	total_sales
▶	Laptop	North	all	4000.00
	Smartphone	South	all	3000.00

```

SELECT
    'all' AS product,
    s.Region AS region,
    d.Year AS time,
    SUM(f.SalesAmount) AS total_sales
FROM FactSales f
JOIN DimStore s ON f.StoreKey = s.StoreKey
JOIN DimDate d ON f.DateKey = d.DateKey
GROUP BY d.Year, s.Region;

```

Result Grid   Filter Rows: <input type="text"/>				
	product	region	time	total_sales
▶	all	North	2023	4000.00
	all	South	2023	3000.00