

Term Project Final Report

Electricity Management System (EMS)

INFO 5707 - Data Modeling for Information Professionals

Prof: Abiodun Awojobi

Group 5 - Eagles

Pavan kalyan Naini

Mohith Sai Yandra

Sushanth Reddy Marreddy

Sharathchandra Reddy Yedavelli

Harshavardhan Reddy Basani

CONTENTS

Project Description	3
Objectives	3
Scope	4
Project Requirements	4
Database Requirements	4
User Requirements	4
Business Rules	5
Entity Relationship Diagram	6
Data Dictionary	6
Queries and Operations.....	11
 Section 1: Entity Generation and Data Entry.....	11
 Overview	11
 Entity Generation and Data Entry for Table Service_Provider	12
 Entity Generation and Data Entry for Table Region	14
 Entity Generation and Data Entry for Table Customers.....	16
 Entity Generation and Data Entry for Table Maintenance_Requests	20
 Entity Generation and Data Entry for Table Meter_Readings.....	22
 Entity Generation and Data Entry for Table Auto_Payments.....	25
 Entity Generation and Data Entry for Table Billing.....	26
 Entity Generation and Data Entry for Table Reminders	29
 Entity Generation and Data Entry for Table Usage_History	32
 Section 2: Data Retrieval and Analytical Reports	35
 Overview	35
 Data Analysis 1	35
 Data Analysis 2	36
 Data Analysis 3	37
 Data Analysis 4	38
 Data Analysis 5	39
 Data Analysis 6	40
 Data Analysis 7	41
 Data Analysis 8	42
 Data Analysis 9	43
 Data Analysis 10	44
Appendix 1.....	46

Project Description

The Electricity Management System represents a groundbreaking database initiative crafted to optimize the oversight of electricity consumption and distribution within a state. This sophisticated system serves as a central repository for crucial information, catering to the needs of both electricity authorities and consumers. It offers real-time monitoring capabilities for authorities, allowing them to track consumption patterns, device specifics, and billing information. This facilitates proactive decision-making and ensures the prompt identification and resolution of potential issues. The system's meticulous record-keeping feature is essential, securely capturing transaction details and safeguarding sensitive customer information.

Customers, in turn, experience enhanced convenience through a user-friendly interface that provides access to vital information. They can effortlessly track their monthly electricity usage, review payment history, stay informed about payment deadlines, and receive updates on potential outages. This transparency not only empowers consumers but also fosters responsible electricity usage. The system's integration of data from multiple electricity providers into a unified database is a cornerstone of its functionality. This integrated approach ensures the accuracy and consistency of information across diverse elements such as customer data, electricity bills, and transaction details, reinforcing the reliability and effectiveness of the state's electricity infrastructure.

In summary, the Electricity Management System is a comprehensive and user-centric solution designed to revolutionize electricity consumption and distribution management and monitoring. Its advanced features strengthen the operational efficiency of authorities, enhance customer satisfaction, and promote responsible energy consumption practices, thereby contributing to the overall reliability and transparency of the state's electricity infrastructure.

Objectives

- **Customer Information:** The system will store and manage detailed customer information, including personal details, contact information, and any relevant preferences.
- **Transaction History:** The system will maintain a record of all customer transactions, including payment history, date and time of transactions, and the corresponding amounts.
- **Electricity Usage Details:** It will store and display historical data on electricity usage for each customer to analyze consumption patterns and make informed decisions.
- **Date-Specific Usage:** The system will allow users to view electricity usage on specific dates. This feature is useful for both customers and authorities to understand consumption patterns on respective days.
- **Distributor Information:** Electricity distributors will have the information stored in the system to ensure that authorities can manage and track distributor-related data efficiently.
- **Employee Information:** All information related to employees involved in the electricity management process will be stored, including details about their roles, responsibilities, and contact information.
- **User Preferences for Automatic Payments:** The system will keep track of customer preferences, particularly those who have opted for automatic payments to ensure seamless billing processes for these users.

Scope

- **Electricity Usage Monitoring:** The system provides a platform for monitoring electricity usage, allowing authorities and customers to track consumption patterns.
- **Payment Details:** Customers can access and review their payment details, including transaction history and automatic payment preferences.
- **Comprehensive Customer Database:** A centralized database for customer information simplifies management tasks and enhances customer service.
- **Data Analysis for Decision-Making:** Previous data is analyzed to make informed decisions about electricity distribution and resource allocation.

Project Requirements

Operating System: Windows/ MacOs

Database: MSSQL

Applications: Azure Data Studio, Lucid chart, Microsoft word

Database Requirements

The following is the list that provides different table names that will be used.

1. Service_Provider Table
2. Region Table
3. Customers Table
4. Maintenance_Requests Table
5. Meter_Readings Table
6. Auto_Payments Table
7. Billing
8. Reminders Table
9. Usage_History

User Requirements

- Usage Monitoring: The system should be able to track and monitor electricity usage for different periods (daily, monthly, and annually) for both residential and commercial purposes.
- Real-Time Data: Users should have access to real-time consumption data to monitor and adjust their usage as necessary.
- Billing Information: The system should allow users to view and manage their billing information, including past bills and payment history.
- Alerts and Notifications: Users should receive alerts for unusual consumption patterns, power outages, or billing notifications.
- Usage Insights: The system should provide analysis and insights on usage trends to help users optimize their consumption and reduce costs.
- User Management: The system should allow admin/manager users to add, remove, or modify user access and permissions.
- System Maintenance: Tools for system maintenance should be included, such as meter diagnostics, updates, and data integrity checks.
- Billing and Invoicing: Admin/manager users should be able to manage billing cycles, generate invoices, and handle payment processing.

Group 5: EMS

- Compliance and Regulations: The system should ensure compliance with energy regulations and standards, with tools for reporting and adherence.
- Scalability: The system should be able to scale to accommodate a growing number of users or meters.
- Data Security: Robust security measures should be in place to protect user data and prevent unauthorized access.
- Data Backup and Recovery: Regular backups and a plan for data recovery in case of system failures or data loss.
- Compatibility and Integration: The system should be compatible with different metering systems and protocols for seamless integration.

Business Rules

1. Customer_Id, Billing_Id, Meter_Id should be unique.
2. Each customer can have only one service provider.
3. Each customer should have one meter for billing.
4. A service provider can have many customers.
5. Each customer may have many usage history records and billing records.
6. Each meter reading should be associated with a unique Meter_Id.
7. An auto-payment record ID or any payment method is associated with one customer and one bill.
8. Maintenancerequest_Id should be specific to one meter and one customer, and some customers may not have maintenancerequest_Id.
9. A particular region may have multiple customers and multiple service providers.
10. Each customer can be located in one region.
11. The total amount calculated for the bill should be based on the particular consumption period and the due balance from previous records.
12. The total payable amount should be equal to or greater than the bill amount generated.
13. Each customer must pay the bill amount by the specified due date.
14. If the total amount payable exceeds the 3-month billing period amount, then the meter connection should be removed.
15. Maintenance requests must be assigned to the nearest service provider to address them.
16. Each maintenance request must have a description and be associated with Customer_Id or Supplier_Id.
17. Each customer can opt for the auto-payment option if the customer has valid credit card details linking to their account.
18. Each customer must have valid account details such as address, phone number, email address, etc.
19. Each usage history record should be associated with a unique Usage_Id and Customer_Id.
20. Each usage history record must have a valid date associated with the meter reading of it.
21. Each customer should get a reminder for their bill payment before the due date.
22. Each customer can have multiple maintenance requests.

Entity-Relationship Diagram

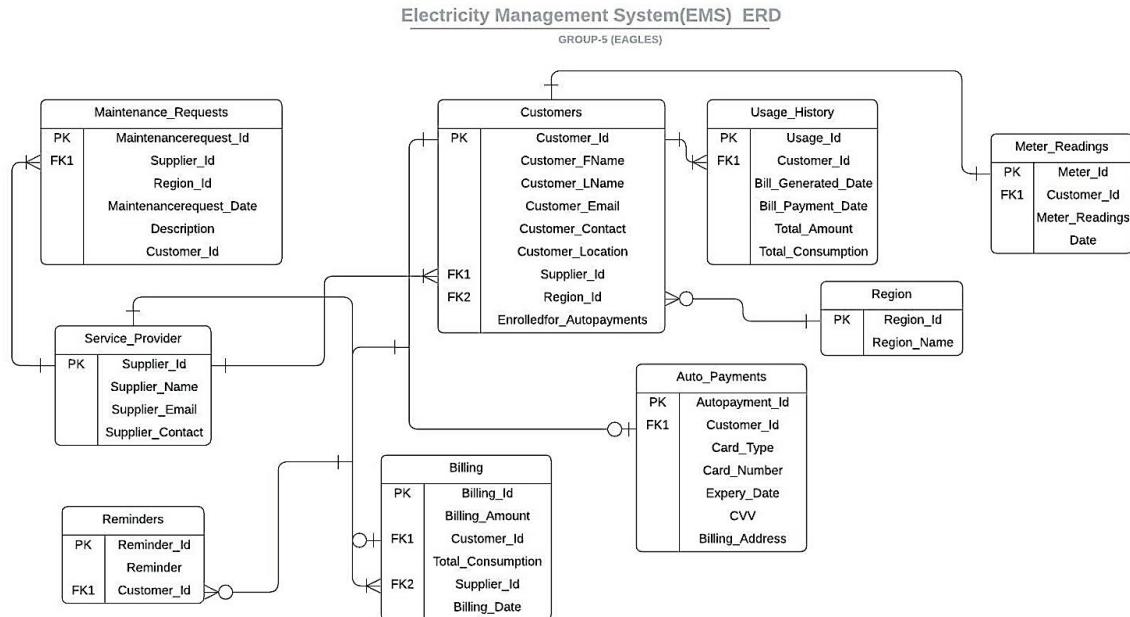


Table 1: Components of ERM

Entity	Relationship	Connectivity	Entity
Billing	Customers	1:1	Customers
Reminders	Bills	M:1	Billing
Maintenance_Requests	Maintenance	M:1	Service_Provider
Meter_Readings	Usage	1:1	Customers
Auto_Payments	Payments	1:1	Customers
Service_Provider	Suppliers	1:M	Customers
Region	Region	1:M	Customers
Usage_History	Usage	M:1	Customers

Data Dictionary

The data dictionary is vital when designing a data model as it allows users to quickly access the entities and their requirements. The dataset will suffer from inconsistency if there is no data dictionary. It lists all the entities, including Name, Description, Type, Format, and Range. It also provides the mapping and relationships between the tables, and the connection between them can be traced easily.

Table 2: Data Dictionary

S.no	Table Name	Attribute Name	Description	Attribute Type	Format	Range	Required	PK or FK	FK Reference Table
1	Customers	Customer_Id	Unique id of the Customer	INT	9	1-999999999	Y	PK	
		Customer_FName	First name of the customer	VARCHAR(50)	Xxxxxxx		Y		
		Customer_LName	Last name of the customer	VARCHAR(50)	Xxxxxxx		Y		
		Customer_Email	Email address of the customer	VARCHAR(50)	Xxxxxxx		Y		
		Customer_Contact	Contact number of the customer	VARCHAR(10)	Xxxxxxx		Y		
		Customer_Location	Address of the customer	VARCHAR(150)	Xxxx XXXXX XXXXX		Y		
		Supplier_Id	unique id of the supplier	INT	9	1-999999999	Y	FK1	Service_Provider
		Region_Id	Unique id of the region	INT	9	1-999999999	Y	FK2	Region
		Enrolledfor_Autopayments	Stores boolean type values	BIT	1/0		Y		
2	Usage_History	Usage_Id	Unique id of Electricity usage	INT	9	1-999999999	Y	PK	

Group 5: EMS

		Customer_Id	Unique id of the customer	INT	9	1-99999 9999	Y	FK1	Customers
		Bill_Generated_Date	The date on which the electricity bill was generated	DATE	yyyy-mm-dd		Y		
		Bill_Payment_Date	The date on which the electricity bill was paid	DATE	yyyy-mm-dd		Y		
		Total_Amount	The total amount of the bill generated	DECIMAL(999999999, 2)	99999.99		Y		
		Total_Consumption	The total consumption of the electricity in bill cycle	DECIMAL(999999999, 2)	9999999 9.99		Y		
3	Meter_ Readings	Meter_Id	Unique id of the meter	INT	9999	1000-99999 9999	Y	PK	
		Customer_Id	Unique id of the Customer	INT	9	1-99999 9999	Y	FK1	Customers
		Meter_ Readings	contains electricity usage	DECIMAL(999999999, 2)	9999999. 99		Y		
		Date	The date which the readings were recorded	DATE	yyyy-mm-dd		Y		
4	Region	Region_Id	Unique id of the region	INT	9	1-99999 9999	Y	PK	

Group 5: EMS

		Region_Name	Name of the region	VARCHAR(50)	Xxxxx		Y		
5	Maintenance_Requests	Maintenancerequest_Id	Unique id of the maintenance request	INT	9	1-999999999	Y	PK	
		Supplier_Id	unique id of the supplier	INT	9	1-999999999	Y	FK1	Service_Provider
		Region_Id	Unique id of the region	INT	9	1-999999999	Y		
		Customer_Id	Unique id of the customer	INT	9	1-999999999	Y		
		Maintenancerequest_Date	The date on which the maintenance is requested	DATE	yyyy-mm-dd		Y		
		Description	Description of the maintenance request	VARCHAR(500)	Xxxxxxx Xxxx Xxx		N		
6	Billing	Billing_Id	Unique Id of the Bill generated	INT	9	1-999999999	Y	PK	
		Billing_Amount	Total amount of the bill generated based on usage	DECIMAL(999999999, 2)	99999.99		Y		
		Customer_Id	Unique id of the Customer	INT	9	1-999999999	Y	FK1	Customers
		Total_Consumption	The total consumption of the	DECIMAL(999999999, 2)	99999999.99		Y		

Group 5: EMS

			electricity in bill cycle					
7	Reminders	Supplier_Id	unique id of the supplier	INT	9	1- 99999 9999	Y	FK2Service_P rovider
		Billing_Date	The date on which the bill is generated	DATE	yyyy- mm-dd		Y	
		Reminder_Id	Unique id of the reminder	INT	999	100- 99999 9999	Y	PK
8	Auto_Payments	Reminder	Description of the reminder	VARCHAR (250)	Xxxxx Xxxx Xxx		Y	
		Customer_Id	Unique id of the Customer	INT	9	1- 99999 9999	Y	FK1Customer s
		Autopayment_Id	Unique id for auto payment of bill	INT	9	1- 99999 9999	Y	PK
		Customer_Id	Unique id of the Customer	INT	9	1- 99999 9999	Y	FK1Customer s
		Card_Type	Type of the card using	VARCHAR (25)	Xxxxx Xxxxx		Y	
		Card_Number	Card number of the customer	VARCHAR (19)	XXXX- XXXX- XXXX- XXXX		Y	
		Expiry_Date	Date of expiry of the card	DATE	yyyy- mm		Y	
		CVV	Three- digit security code for payment	INT	999		Y	

		Billing_Address	Billing address of the customer	VARCHAR (250)	Xxxx Xxxx Xxxx		Y		
9	Service_Provider	Supplier_Id	unique id of the supplier	INT	9	1- 99999 9999	Y	PK	
		Supplier_Name	Name of the Supplier	VARCHAR (50)	Xxxxxx		Y		
		Supplier_Email	Email id of the Supplier	VARCHAR (50)	Xxxxxxxxxx x		Y		
		Supplier_Contact	Contact of the Supplier	VARCHAR (10)	Xxxxxxxxxx xx		Y		

Queries and Operations

Section 1: Entity Generation and Data insertion: Create Database, Create Table and Insert data into the Tables

Overview: EMS data model contains only one database named ‘EMS’ where it lists the database schema for the EMS database and associated tables listed in Table 3. To avoid inserting any type of invalid data, foreign key constraints are applied while the creation of the tables in the database. Each value in the foreign key must have an associated value taken from the primary key.

(*Comments: Before executing queries in section 2, queries in section 1 should be executed to avoid errors in extracting the results.)

Table 3: Database Schema

Database	Table Name	Table Explanation	Records	Insertion order
EMS	Service_Provider	This table contains information related to the power suppliers		1
EMS	Region	This Table contains all the region names		2
EMS	Customers	This Table provides information about customers		3
EMS	Maintenance_Requests	This Table contains all the Maintenance requests raised by the customers and suppliers		4

Group 5: EMS

EMS	Meter_ Readings	This Table provides the readings of meter in current billing period		5
EMS	Auto_Payments	This Table stores the preferred payment method for auto payment of bill in billing period		6
EMS	Billing	This Table provides the total consumption and amount due in current billing period		7
EMS	Reminders	This Table provides the reminders sent to customer related to Billing		8
EMS	Usage_History	This Table contains all the previous bill payment and consumption history		9

Table 4 Query steps and Explanation

Steps	Query	Example
1.Creating Database	Create database dbname;	Create database EMS;
2.Creating table	Use dbname; Create table tablename(A1 datatype, A2 datatype, A3 datatype,...);	<code>use EMS; create table Service_Provider(Supplier_Id Int identity(1,1) primary key, Supplier_Name Varchar(50) not null, Supplier_Email Varchar(50) not null, Supplier_Contact Varchar(10) not null)</code>
3. Inserting values into table	Use Dbname; Insert into tablename(A1,A2,A3,..) Values(B1,B2,B3,...);	<code>use EMS; insert into service_provider(Supplier_Name,Supplier_Contact ,Supplier_EMail) values('Reliant','8332359653','info@reliant.com)</code>
4.Retrieving the data	Use dbname; Select * from tablename;	<code>use EMS; select * from Service_Provider;</code>

Entity Generation and Data Entry for Table Service Provider:

Statements Explanation

- The Database EMS is created in the database schema using command “Create database EMS”.
- The Table Service_Provider is created in EMS database using command “Create table Service_Provider” and command “Use EMS” is used before creating the table to store the table in EMS database.
- Table Service_Provider is filled with relevant data using command “Insert into” and inserting values into the table.

Group 5: EMS

- Finally, the result will be displayed by using command “select * from Service_Provider” to display all inserted values in the table.

Queries

```
/*-----Create database EMS-----*/
Create database EMS;
/*-----Create Service_Provider Table-----*/
use EMS;
create table Service_Provider(
    Supplier_Id Int identity(1,1) primary key,
    Supplier_Name Varchar(50) not null,
    Supplier_Email Varchar(50) not null,
    Supplier_Contact Varchar(10) not null
);
/*-----Insert values to Service_Provider Table-----*/
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Reliant','8332359653','info@reliant.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Oncor','8263758293','info@oncor.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Compare Power','8462758364','info@comparepowwer.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Amigo Power','9162553475','info@Amigopower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Ampra Energy','7253856495','info@amprapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Energy Texas','8371698365','info@energytexas.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Tara energy','7239476849','info@taraenergy.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Onver Power','5837469382','info@onverpower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Tata Powers','8385945848','info@tatapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Cupa power','5837485737','info@cupapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Zampa Distributions','4938572647','info@zampadistributions.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('spectrum powers','9472837485','info@tspectrumpowers.com');
/*-----*/
```

Result

Table Explanation: Service_provider provides information about suppliers like their name, contact, and email. To query all the information command select * from service_Provider” is used.

Group 5: EMS

Query

```
select * from Service_Provider;
```

Screenshot

Supplier_Id	Supplier_Name	Supplier_Email	Supplier_Contact
1	Reliant	info@reliant.com	8332359653
2	Oncor	info@oncor.com	8263758293
3	Compare Power	info@comparepowwer.com	8462758364
4	Amigo Power	info@Amigopower.com	9162553475
5	Ampra Energy	info@amrapower.com	7253856495
6	Energy Texas	info@energytexas.com	8371698365
7	Tara energy	info@taraeenergy.com	7239476849
8	Onverpower	info@onverpower.com	5837469382
9	Tata Powers	info@tatapower.com	8385945848
10	Cupa power	info@cupapower.com	5837485737
11	Zampa Distributions	info@zampadistributions.com	4938572647
12	spectrum powers	info@tspectrumpowers.com	9472837485

Entity Generation and Data Entry for Table Region:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Table Region is created in EMS database. Command “Create table Region” is used to create the table.
- The table Region is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Region” to query all the inserted values to the table.

Queries

```
/*-----Create table Region-----*/
create table Region(
    Region_Id int identity(1,1) primary key,
    Region_Name varchar(50) not null
);
/*-----Inserting values into Region table-----*/
```

Group 5: EMS

```
insert into Region values('Denton');
insert into Region values('Dallas');
insert into Region values('Houston');
insert into Region values('Austin');
insert into Region values('Irving');
insert into Region values('Plano');
insert into Region values('Frisco');
insert into Region values('Fort-worth');
insert into Region values('Lewisville');
insert into Region values('Phoenix');

/*-----*/
```

Result

Table Explanation: Region table provides information about the region ID and name.

Query

```
select * from Region;
```

Screenshot

The screenshot shows the SSMS interface with the EMS database selected. The left pane displays the database structure, including tables like dbo.Region, dbo.Reminders, and dbo.Auto_P... . The right pane shows the results of the query 'select * from Region;'.

Region_Id	Region_Name
1	Denton
2	Dallas
3	Houston
4	Austin
5	Irving
6	Plano
7	Frisco
8	Fort-worth
9	Lewisville
10	Phoenix

Entity Generation and Data Entry for Table Customers:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Table Customers is created in EMS database. Command “Create table Customers” is used to create the table.
- The table Customers is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Customers” to query all the inserted values to the table.

Queries

```
/*-----Create table Customers-----*/  
  
create table Customers(  
Customer_Id int identity(1,1) primary key,  
Customer_FName varchar(50) not null,  
Customer_LName varchar(50) not null,  
Customer_Email varchar(50) not null,  
Customer_Contact varchar(10) not null,  
Customer_Location varchar(150) not null,  
Supplier_Id int not null,  
Region_Id int not null,  
Enrolledfor_Autopayments bit default 0 not null,  
foreign key(Supplier_Id) references Service_Provider(Supplier_Id),  
foreign key(Region_Id) references Region(Region_Id)  
);  
  
/*-----Inserting values into Customers table-----*/  
  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location, Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Raghu', 'nandhan', '4857362847', 'raghu@gmail.com', 'union circle, denton, TX', 1, 1, 1);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location, Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('ram', 'charan', '4837285674', 'ram@gmail.com', 'oaks of denton, denton, TX', 3, 1, 0);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Email, Customer_Contact, Customer_Location, Supplier_Id, Region_Id)  
values('Rishi', 'kumar', 'rishi@gmail.com', '8573746385', 'valley ranch pkwy, Irving, TX', 1, 5);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location, Supplier_Id, Region_Id, Enrolledfor_Autopayments)
```

Group 5: EMS

```
values('priya','chandran','9472847364','pricha@gmail.com','oaks at valley ranch,  
Irving, Tx',2,5,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Sruthi','Kiran','9263828462','kiran@gmail.com','Swadeshi plaza, Irving,  
TX',4,5,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Cristina','Rose','6284638472','cristina@gmail.com','Wilmington pkwy, Fort-  
worth, Tx',4,8,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Ronald','Wright','7483648264','ronald@gmail.com','Dallas pkwy, dallas,  
Tx',6,2,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Terissa','Tim','9273846284','teressa@gmail.com','Loscolinas pkwy, houston,  
Tx',5,3,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Mike','Tong','9374738472','Mike@gmail.com','Spring valley rd,plano,  
Tx',3,6,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Tom','Crissey','8284637463','Tom@gmail.com','Spring valley rd,plano,  
Tx',2,6,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Micheal','Tom','8264927482','Micheal@gmail.com','Montfort dr, frisco,  
Tx',1,7,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Pavan','Naini','9405959573','pavan@gmail.com','Ioof St, Houston, Tx',9,3,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Kalyan','Tinku','9890987890','tinku@gmail.com','Bernald St denton, Tx',1,1,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
```

Group 5: EMS

```
values('Kiran Kumar','Mintu','9309439890','kirankumar@gmail.com','lebin street,  
lw',7,9,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Adam','Smith','9453245679','adamsmith@gmail.com','Teasley Lane,Ph',7,10,12);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Anvar','mith','8293746738','anvar@gmail.com','Timber st, Lewisville,  
Tx',11,9,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('cane','walter','3947584839','cane@gmail.com','Twistley rd,Frisco,Tx',10,7,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Kane','county','5928385858','kane@gmail.com','winter ln, Plano, Tx',8,6,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Ruth','ranger','7838485868','ruth@gmail.com','Rinder pkwy, Dallas, Tx',6,2,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Birkey','wing','8273648573','birkey@gmail.com','oaks of denton, Denton,  
Tx',11,1,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('king','william','7394829485','kingwilliam@gmail.com','trails of austin,  
Austin,Tx',12,4,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('ram','singh','8483847275','ramsingh@gmail.com','lukes of fort-worth, Fort-  
worth, Tx',9,8,0);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('Trint','bing','8938475837','trinth@gmail.com','churchy st, Phoenix,  
Tx',5,10,1);  
insert into  
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,  
Supplier_Id,Region_Id,Enrolledfor_Autopayments)  
values('wright','wrong','8394388484','wright@gmail.com','trinty ln, Denton,  
Tx',8,1,0);
```

Group 5: EMS

```
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('miller', 'wright', '8273645273', 'miller@gmail.com', 'west street ln, Irving,
Tx', 9, 5, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Anil', 'kumbley', '7483746574', 'kumbley@gmail.com', 'Timber west ln, Austin,
Tx', 2, 4, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('virat', 'rona', '7384958274', 'virat@gmail.com', 'valley pkwy E, Plano,
Tx', 4, 6, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('vikranth', 'rana', '8938485828', 'vikranth@gmail.com', 'ruster rd, Fort-worth,
Tx', 8, 8, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Ringer', 'rale', '8394727475', 'ringer@gmail.com', 'winster pkwy, Austin,
Tx', 10, 4, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('stella', 'fring', '9284837475', 'stella@gmail.com', 'oaks of Phoenix, Phoenix,
Tx', 11, 10, 0);
/*-----*/
```

Result

Table Explanation: Table Customers provides information about all customer information like name, contact, email and address.

Query

```
select * from Customers;
```

Group 5: EMS

Screenshot

Customer_Id	Customer_FName	Customer_LName	Customer_Email	Customer_Contact	Customer_Location	Supplier_Id	
1	Raghu	nandhan	raghug@gmail.com	4857362847	union circle, denton, tx	1	
2	ram	charan	ram@gmail.com	4837285674	oaks of denton, denton, TX	3	
3	Rishi	kumar	rishi@gmail.com	8573746385	valley ranch pkwy, Irving, TX	1	
4	priya	chandran	pricha@gmail.com	9472847364	oaks at valley ranch, Irving, Tx	2	
5	Sruthi	Kiran	kirang@gmail.com	9263828462	Swadeshi plaza, Irving, TX	4	
6	Region	Cristina	Rose	6284638472	Wilmington pkwy, Fort-worth, Tx	4	
7	Ronald	Wright	ronald@gmail.com	7483648264	Dallas pkwy, dallas, Tx	6	
8	Service...	Tim	teressa@gmail.com	9273846284	Loscolinas pkwy, houston, Tx	5	
9	Usage...	Tong	Mike@gmail.com	9374738472	Spring valley rd,plano, Tx	3	
10	Dropped Le...	Tom	Crissey	Tom@gmail.com	Spring valley rd,plano, Tx	2	
11	Views	Micheal	Tom	Micheal@gmail.com	Montfort dr, frisco, Tx	1	
12	Synonyms	Pavan	Naini	pavan@gmail.com	9405959573	Ioof St, Houston, Tx	9
13	Programmabl...	Kalyan	Tinku	tinku@gmail.com	9896987890	Bernald St denton, Tx	1
14	External Res...	Kiran Kumar	Mintu	kirankumar@gmail.com	9389439890	lebin street, lw	7
15	Service Broker	Adam	Smith	adamsmith@gmail.com	9453245679	Teasley Lane,Ph	7
16	Storage	Anvar	mith	anvar@gmail.com	8293746738	Timber st, Lewisville, Tx	11
17	Security	cane	walter	cane@gmail.com	3947584839	Twistley rd,Frisco,Tx	10
18		Kane	county	Kane@gmail.com	5928385858	winter ln, Plano, Tx	8
19		Ruth	ranger	ruth@gmail.com	7838485668	Rinder pkwy, Dallas, Tx	6
20		Birkey	wing	birkey@gmail.com	8273648573	oaks of denton, Denton, Tx	11
21		king	william	Kingwilliam@gmail.com	7394829485	trails of austin, Austin,Tx	12
22		ram	singh	ramsingh@gmail.com	8483847275	lukes of fort-worth, Fort-worth, Tx	9
23		Trint	bing	trinth@gmail.com	8938475837	churchy st, Phoenix, Tx	5
24		wright	wrong	wright@gmail.com	8394388484	trinty ln, Denton, Tx	8
25		miller	wright	miller@gmail.com	8273645273	west street ln, Irving, Tx	9
26		Anil	kumbley	kumbley@gmail.com	7483746574	Timber west ln, Austin, Tx	2
27		virat	rona	virat@gmail.com	7384958274	valley pkwy E, Plano, Tx	4
79		vibekanth	vibekanth	vibekanth@gmail.com	0070605070	Futura Rd, Fortworth, Tx	6

Entity Generation and Data Entry for Table Maintenance Requests:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Table Maintenance_Requests is created in EMS database. Command “Create table Maintenance_Requests” is used to create the table.
- The table Maintenance_Requests is filled with relevant data using command “Insert into ***”.
- Then, the results will be displayed by using the command “select * from Maintenance_Requests” to query all the inserted values to the table.

Queries

```
/*-----Create Maintenance_Requests Table-----*/
create table Maintenance_Requests(
    Maintenancerequest_Id int identity(1,1) primary key,
    Supplier_Id int not null,
    Region_Id int default null,
    Customer_Id int default null,
    Maintenancerequest_Date date not null,
    Description varchar(500) default' ',
    foreign key(Supplier_Id) REFERENCES Service_Provider(Supplier_Id)
```

Group 5: EMS

```
);

/*-----Inserting values into Maintenance_Requests Table-----*/
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(2,'2023-06-22','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
values(1,5,4,'2023-08-23','Request for Installation of meter');
insert into Maintenance_Requests
values(1,1,1,'2023-06-12','Request for Installation of meter');
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(5,'2023-08-11','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
values(4,8,6,'2023-04-23','Request for Installation of meter');
insert into Maintenance_Requests
values(6,2,7,'2023-10-05','Request for Installation of meter');
insert into Maintenance_Requests
values(5,3,8,'2023-09-02','Request for Installation of meter');
insert into Maintenance_Requests
values(2,6,10,'2023-08-06','Request for Installation of meter');
insert into Maintenance_Requests
values(1,7,11,'2023-10-01','Request for Installation of meter');
insert into Maintenance_Requests
values(1,1,13,'2023-12-02','Request for Installation of meter');
insert into Maintenance_Requests
values(7,10,15,'2023-11-04','Request for change of meter');
insert into Maintenance_Requests
values(10,7,17,'2023-08-13','Request for Installation of meter');
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(3,'2023-11-10','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
values(6,12,19,'2023-09-12','Request for change of meter');
insert into Maintenance_Requests
values(11,1,20,'2023-08-13','Request for Installation of meter');
insert into Maintenance_Requests
values(9,8,22,'2023-10-04','Request for change of meter');
insert into Maintenance_Requests
values(5,10,23,'2023-11-13','Request for Installation of meter');
insert into Maintenance_Requests
values(2,4,26,'2023-09-06','Request for change of meter');
insert into Maintenance_Requests
values(8,8,28,'2023-08-03','Request for Installation of meter');
insert into Maintenance_Requests
values(10,4,29,'2023-07-12','Request for Installation of meter');
/*-----*/
```

Result

Table Explanation: Maintenance_Requests table provides the information about the maintenance requests raised by the customers and suppliers.

Group 5: EMS

Query

```
select * from Maintenance_Requests;
```

Screenshot

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost - SQLQuery_2 - localh...MS (sa)
- Databases:** EMS
- Table:** Maintenance_Requests
- Results:** The table contains 20 rows of data, each representing a maintenance request. The columns are: Maintenancerequest_Id, Supplier_Id, Region_Id, Customer_Id, Maintenancerequest_Date, and Description.
- Sample Data:**

Maintenancerequest_Id	Supplier_Id	Region_Id	Customer_Id	Maintenancerequest_Date	Description
1	2	NULL	NULL	2023-06-22	Maintenance is scheduled at the supplier end ann
2	1	5	4	2023-08-23	Request for Installation of meter
3	1	1	1	2023-06-12	Request for Installation of meter
4	5	NULL	NULL	2023-08-11	Maintenance is scheduled at the supplier end ann
5	4	8	6	2023-04-23	Request for Installation of meter
6	6	2	7	2023-10-05	Request for Installation of meter
7	5	3	8	2023-09-02	Request for Installation of meter
8	2	6	10	2023-08-06	Request for Installation of meter
9	1	7	11	2023-10-01	Request for Installation of meter
10	1	1	13	2023-12-02	Request for Installation of meter
11	7	10	15	2023-11-04	Request for change of meter
12	10	7	17	2023-08-13	Request for Installation of meter
13	3	NULL	NULL	2023-11-10	Maintenance is scheduled at the supplier end ann
14	6	12	19	2023-09-12	Request for change of meter
15	11	1	20	2023-08-13	Request for Installation of meter
16	9	8	22	2023-10-04	Request for change of meter
17	5	10	23	2023-11-13	Request for Installation of meter
18	2	4	26	2023-09-06	Request for change of meter
19	8	8	28	2023-08-03	Request for Installation of meter
20	10	4	29	2023-07-12	Request for Installation of meter

Entity Generation and Data Entry for Table Meter Readings:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Table Meter_Readings is created in EMS database. Command “Create table Meter_Readings” is used to create the table.
- The table Meter_Readings is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Meter_Readings” to query all the inserted values to the table.

Queries

```
/*-----Create Meter_Readings Table-----*/  
create table Meter_Readings(  
    Meter_Id int identity(1000,1) primary key,  
    Customer_Id int not null,  
    Meter_Readings decimal(9,2) default 0.00,  
    Readings_Date date not null,
```

Group 5: EMS

```
    foreign key(Customer_Id) references Customers(Customer_Id)
);
/*-----Inserting values into Meter_Readings Table-----*/
insert into Meter_Readings
values(1,23.56,'2023-11-20');
insert into Meter_Readings
values(2,13.27,'2023-11-20');
insert into Meter_Readings
values(3,20.46,'2023-11-20');
insert into Meter_Readings
values(4,32.76,'2023-11-20');
insert into Meter_Readings
values(5,53.2,'2023-11-20');
insert into Meter_Readings
values(6,49.5,'2023-11-20');
insert into Meter_Readings
values(7,42.3,'2023-11-20');
insert into Meter_Readings
values(8,64.2,'2023-11-20');
insert into Meter_Readings
values(9,40.3,'2023-11-20');
insert into Meter_Readings
values(10,54.5,'2023-11-20');
insert into Meter_Readings
values(11,46.7,'2023-11-20');
insert into Meter_Readings
values(12,34.4,'2023-11-20');
insert into Meter_Readings
values(13,10.1,'2023-11-20');
insert into Meter_Readings
values(14,19.2,'2023-11-20');
insert into Meter_Readings
values(15,43.2,'2023-11-20');
insert into Meter_Readings
values(16,53.5,'2023-11-20');
insert into Meter_Readings
values(17,23.27,'2023-11-20');
insert into Meter_Readings
values(18,22.46,'2023-11-20');
insert into Meter_Readings
values(19,22.76,'2023-11-20');
insert into Meter_Readings
values(20,12.51,'2023-11-20');
insert into Meter_Readings
values(21,33.56,'2023-11-20');
insert into Meter_Readings
values(22,23.27,'2023-11-20');
```

Group 5: EMS

```
insert into Meter_Readings
values(23,25.46,'2023-11-20');
insert into Meter_Readings
values(24,22.76,'2023-11-20');
insert into Meter_Readings
values(25,16.51,'2023-11-20');
insert into Meter_Readings
values(26,31.76,'2023-11-20');
insert into Meter_Readings
values(27,19.27,'2023-11-20');
insert into Meter_Readings
values(28,27.46,'2023-11-20');
insert into Meter_Readings
values(29,37.76,'2023-11-20');
insert into Meter_Readings
values(30,19.51,'2023-11-20');
/*-----*/
```

Result

Table Explanation: Meter_Readings table provides the information about the electricity usage by the customer with specific meter_Id in current billing period.

Query

```
select * from Meter_Readings;
```

Screenshot

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost - SQLQuery_2 - localhost\MS (sa)
- Databases:** EMS
- Tables:** EMS
- Results:** The results of the query "select * from Meter_Readings;" are displayed in a grid. The columns are: Meter_Id, Customer_Id, Meter_Readings, and Readings_Date. The data consists of 28 rows, each representing a different meter reading.

Meter_Id	Customer_Id	Meter_Readings	Readings_Date
1	1000	23.56	2023-11-20
2	1001	13.27	2023-11-20
3	1002	20.46	2023-11-20
4	1003	32.76	2023-11-20
5	1004	53.28	2023-11-20
6	1005	49.50	2023-11-20
7	1006	42.30	2023-11-20
8	1007	64.20	2023-11-20
9	1008	40.30	2023-11-20
10	1009	54.50	2023-11-20
11	1010	46.70	2023-11-20
12	1011	34.40	2023-11-20
13	1012	10.10	2023-11-20
14	1013	19.20	2023-11-20
15	1014	43.20	2023-11-20
16	1015	53.50	2023-11-20
17	1016	23.27	2023-11-20
18	1017	22.46	2023-11-20
19	1018	22.76	2023-11-20
20	1019	12.51	2023-11-20
21	1020	33.56	2023-11-20
22	1021	23.27	2023-11-20
23	1022	25.46	2023-11-20
24	1023	22.76	2023-11-20
25	1024	16.51	2023-11-20
26	1025	31.76	2023-11-20
27	1026	19.27	2023-11-20
28	1027	27.46	2023-11-20

Entity Generation and Data Entry for Table Auto Payments:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Table Auto_Payments is created in EMS database. Command “Create table Auto_Payments” is used to create the table.
- The table Auto_Payments is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Auto_Payments” to query all the inserted values to the table.

Queries

```
/*-----Create Auto_Payments Table-----*/
create table Auto_Payments(
    Autopayment_Id int identity(1,1) primary key,
    Customer_Id int unique not null,
    Card_Type varchar(25) not null,
    Card_Number varchar(19) not null unique,
    Expiry_Date Date not null,
    CVV int not null,
    Billing_Address varchar(250) not null,
    foreign key(Customer_Id) REFERENCES Customers(Customer_Id)
);
/*-----Inserting values into Auto_Payments Table-----*/
insert into Auto_Payments
values(1,'credit','4783927594716258','2024-02-01',346,'union circle');
insert into Auto_Payments
values(4,'debit','8264857102859374','2025-12-01',153,'oaks at valley ranch');
insert into Auto_Payments
values(6,'Debit','5263846273849270','2025-12-01',134,'Wilmington pkwy, Fort-worth, tx');
insert into Auto_Payments
values(8,'debit','8263745261846370','2026-04-01',143,'Loscolinas pkwy, houston, tx');
insert into Auto_Payments
values(11,'debit','8394728463746370','2024-03-01',256,'Montfort dr, frisco, tx');
insert into Auto_Payments
values(12,'Credit','12345','2027-01-01',123,'Ioof st,tx');
insert into Auto_Payments
values(14,'Debit','2341234564332','2030-11-01',533,'lebin street, lw');
insert into Auto_Payments
values(15,'Credit','234567812234','2025-12-01',324,'bernard street,denton');
insert into Auto_Payments
values(16,'credit','2150816150957480','2031-01-01',587,'Discovery park dr');
insert into Auto_Payments
values(17,'credit','4838485969485860','2024-07-01',564,'Twistley rd,Frisco,Tx');
insert into Auto_Payments
values(18,'debit','9483847562736450','2025-06-01',498,'winter ln, Plano, Tx');
```

Group 5: EMS

```
values(23,'credit','2938485862949280','2024-08-01',734,'churchy st, Phoenix, Tx');
insert into Auto_Payments
values(27,'debit','7374859384759280','2027-11-01',435,'valley pkwy E, Plano, Tx');
/*-----*/
```

Result

Table Explanation: Auto_Payments table provides the information about the preferred payment method by the customer to automatically debit the billing mount from the card details provided by the customers.

Query

```
select * from Auto_Payments;
```

Screenshot

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost - SQLQuery_2 - localhost\MS (sa)
- Databases:** EMS
- Tables:** EMS
- Query Results:** The results of the query "select * from Auto_Payments;" are displayed in a grid. The columns are: Autopayment_Id, Customer_Id, Card_Type, Card_Number, Expiry_Date, CVV, and Billing_Address. The data includes various payment records such as credit and debit entries for different customers across different locations.
- Status Bar:** Shows the count of rows (13 rows), encoding (UTF-8), and execution time (00:00:00).

Entity Generation and Data Entry for Table Billing:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Billing table is created in EMS database. Command “Create table Billing” is used to create the table.

Group 5: EMS

- The table Billing is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Billing” to query all the inserted values to the table.

Queries

```
/*-----Create Billing Table-----*/
create table Billing(
    Billing_Id int identity(1,1) primary key,
    Billing_Amount decimal(9,2) default 0.00,
    Customer_Id int unique not null,
    Total_Consumption decimal(9,2) default 0.00,
    Supplier_Id int not null,
    Billing_Date date,
    foreign key(Customer_Id) references Customers(Customer_Id)
);
/*-----Inserting values into Billing Table-----*/
insert into Billing
values(56.45,1,134.45,1,'2023-11-03');
insert into Billing
values(43.25,2,108.42,2,'2023-11-03');
insert into Billing
values(72.34,3,184.3,1,'2023-11-03');
insert into Billing
values(38.34,4,90.2,2,'2023-11-03');
insert into Billing
values(58.45,5,105.32,4,'2023-11-03');
insert into Billing
values(52.2,6,105.36,4,'2023-11-03');
insert into Billing
values(48.4,7,89.9,6,'2023-11-03');
insert into Billing
values(72.2,8,130.36,5,'2023-11-03');
insert into Billing
values(44.43,9,82.6,3,'2023-11-03');
insert into Billing
values(54.9,10,103.5,2,'2023-11-03');
insert into Billing
values(48.89,11,86.76,1,'2023-11-03');
insert into Billing
values(50.55,12,75,9,'2023-11-03');
insert into Billing
values(12,13,31,1,'2023-11-03');
insert into Billing
values(45,14,12,7,'2023-11-03');
insert into Billing
values(94,15,56,7,'2023-11-03');
insert into Billing
```

Group 5: EMS

```
values(100,16,200,11,'2023-11-03');
insert into Billing
values(86.65,17,164.6,10,'2023-11-03');
insert into Billing
values(46.87,18,128.8,8,'2023-11-03');
insert into Billing
values(55.23,19,104.45,6,'2023-11-03');
insert into Billing
values(37.87,20,74.45,11,'2023-11-03');
insert into Billing
values(64.57,21,124.45,12,'2023-11-03');
insert into Billing
values(37.43,22,84.55,9,'2023-11-03');
insert into Billing
values(42.45,23,90.43,5,'2023-11-03');
insert into Billing
values(39.78,24,80.15,8,'2023-11-03');
insert into Billing
values(37.67,25,82.55,9,'2023-11-03');
insert into Billing
values(46.23,26,101.86,2,'2023-11-03');
insert into Billing
values(48.65,27,96.23,4,'2023-11-03');
insert into Billing
values(39.78,28,92.45,8,'2023-11-03');
insert into Billing
values(52.78,29,108.36,10,'2023-11-03');
insert into Billing
values(49.67,30,103.28,11,'2023-11-03');
/*-----*/
```

Result

Table Explanation: Billing table provides the information about customer's total consumption in the previous bill cycle and amount due.

Query

```
select * from Billing;
```

Group 5: EMS

Screenshot

Billing_Id	Billing_Amount	Customer_Id	Total_Consumption	Supplier_Id	Billing_Date
1	56.45	1	134.45	1	2023-11-03
2	43.25	2	108.42	2	2023-11-03
3	72.34	3	184.30	1	2023-11-03
4	38.34	4	90.20	2	2023-11-03
5	52.20	5	105.32	4	2023-11-03
6	48.40	6	105.36	4	2023-11-03
7	72.20	7	89.90	6	2023-11-03
8	44.43	8	130.36	5	2023-11-03
9	54.90	9	82.60	3	2023-11-03
10	48.89	10	103.50	2	2023-11-03
11	50.55	11	86.76	1	2023-11-03
12	12.00	12	75.00	9	2023-11-03
13	45.00	13	31.00	1	2023-11-03
14	94.00	14	12.00	7	2023-11-03
15	100.00	15	56.00	7	2023-11-03
16	86.65	16	200.00	11	2023-11-03
17	46.87	17	164.60	10	2023-11-03
18	55.23	18	128.80	8	2023-11-03
19	37.87	19	104.45	6	2023-11-03
20	64.57	20	74.45	11	2023-11-03
21	37.43	21	124.45	12	2023-11-03
22	42.45	22	84.55	9	2023-11-03
23	39.78	23	90.43	5	2023-11-03
24	37.67	24	88.15	8	2023-11-03
25	46.23	25	82.55	9	2023-11-03
26	48.65	26	101.86	2	2023-11-03
27	39.78	27	96.23	4	2023-11-03
28	39.78	28	92.45	8	2023-11-03

Entity Generation and Data Entry for Table Reminders:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Reminders table is created in EMS database. Command “Create table Reminders” is used to create the table.
- The table Reminders is filled with relevant data using command “Insert into **”.
- Then, the results will be displayed by using the command “select * from Reminders” to query all the inserted values to the table.

Queries

```
/*----- Reminders Table-----*/
create table Reminders(
    Reminder_Id int identity(100,1) primary key,
    Reminder varchar(250) default 'No reminders or changed reminder preferences',
    Customer_Id int not null,
    foreign key(Customer_Id) REFERENCES Customers(Customer_Id)
);
/*-----Inserting values into Reminders Table-----*/
insert into Reminders
VALUES('Your bill has paid usig autopayment method',1);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',2);
```

Group 5: EMS

```
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',3);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',4);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',5);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',6);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',7);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',8);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',9);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',10);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',11);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',12);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',13);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',14);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',15);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',16);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',17);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',18);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',19);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',20);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',21);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',22);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',23);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',24);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',25);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',26);
```

Group 5: EMS

```
insert into Reminders
VALUES('Your bill has paid usig autopayment method',27);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',28);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',29);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',30);
/*-----*/
```

Result

Table Explanation: Reminders table provides the information about the reminders sent to customers regarding the bill payments.

Query

```
select * from Reminders;
```

Screenshot

The screenshot shows the SSMS interface with the 'EMS' database selected. The 'Tables' node under 'EMS' is expanded, and the 'Reminders' table is selected. The 'Results' tab is active, displaying the following data:

Reminder_Id	Reminder	Customer_Id
1	Your bill has paid usig autopayment method	1
2	Your generated bill was due on 11/20/2023	2
3	Your generated bill was due on 11/20/2023	3
4	Your bill has paid usig autopayment method	4
5	Your generated bill was due on 11/20/2023	5
6	Your bill has paid usig autopayment method	6
7	Your generated bill was due on 11/20/2023	7
8	Your bill has paid usig autopayment method	8
9	Your generated bill was due on 11/20/2023	9
10	Your generated bill was due on 11/20/2023	10
11	Your bill has paid usig autopayment method	11
12	Your generated bill was due on 11/20/2023	12
13	Your generated bill was due on 11/20/2023	13
14	Your bill has paid usig autopayment method	14
15	Your bill has paid usig autopayment method	15
16	Your bill has paid usig autopayment method	16
17	Your bill has paid usig autopayment method	17
18	Your bill has paid usig autopayment method	18
19	Your generated bill was due on 11/20/2023	19
20	Your generated bill was due on 11/20/2023	20
21	Your generated bill was due on 11/20/2023	21
22	Your generated bill was due on 11/20/2023	22
23	Your bill has paid usig autopayment method	23
24	Your generated bill was due on 11/20/2023	24
25	Your generated bill was due on 11/20/2023	25
26	Your generated bill was due on 11/20/2023	26
27	Your bill has paid usig autopayment method	27
28	Your generated bill was due on 11/20/2023	28

Entity Generation and Data Entry for Table Usage History:

Statements Explanation

- The database EMS is already created. Command “Use EMS” is used to call the database.
- Usage_History table is created in EMS database. Command “Create table Usage_History” is used to create the table.
- The table Usage_History is filled with relevant data using command “Insert into ***”.
- Then, the results will be displayed by using the command “select * from Usage_History” to query all the inserted values to the table.

Queries

```
/*-----Usage_History Table-----*/
create table Usage_History(
    Usage_Id int identity(1,1) primary key,
    Customer_Id int not null,
    Bill_Generated_Date date not null,
    Bill_payment_Date date not null,
    Total_Amount decimal(9,2),
    Total_Consumption decimal(9,2),
    foreign key(Customer_Id) references Customers(Customer_Id)
);
/*-----Inserting values into Usage_History Table-----*/
insert into Usage_History
values(1,'2023-10-3','2023-10-18',56.78,126.54);
insert into Usage_History
values(1,'2023-09-3','2023-09-15',53.18,106.54);
insert into Usage_History
values(1,'11/3/2023','11/6/2023',46.77,92.5);
insert into Usage_History
values(2,'2023-10-3','2023-10-18',46.78,88.54);
insert into Usage_History
values(2,'11/3/2023','11/12/2023',36.77,72.5);
insert into Usage_History
values(3,'11/3/2023','11/11/2023',33.7,66.9);
insert into Usage_History
values(4,'11/3/2023','11/6/2023',20.3,40.6);
insert into Usage_History
values(5,'2023-10-3','2023-10-18',36.3,70.1);
insert into Usage_History
values(5,'11/3/2023','11/07/2023',36.5,70.2);
insert into Usage_History
values(6,'11/3/2023','11/6/2023',33.6,67.3);
insert into Usage_History
values(7,'11/3/2023','11/08/2023',29.6,60.4);
insert into Usage_History
values(8,'11/3/2023','11/6/2023',39.5,80.4);
```

Group 5: EMS

```
insert into Usage_History
values(9,'11/3/2023','11/15/2023',60.3,130.4);
insert into Usage_History
values(10,'11/3/2023','11/8/2023',53.5,104.3);
insert into Usage_History
values(11,'2023-10-3','2023-06-18',44.7,95.6);
insert into Usage_History
values(11,'11/3/2023','11/6/2023',50.4,100.8);
insert into Usage_History
values(12,'2023-10-3','2023-06-18',39.5,80.9);
insert into Usage_History
values(12,'2023-09-3','2023-06-15',37.8,75.8);
insert into Usage_History
values(12,'11/3/2023','11/6/2023',41.7,90.2);
insert into Usage_History
values(13,'11/3/2023','11/13/2023',33.33,66.66);
insert into Usage_History
values(14,'11/3/2023','11/6/2023',42.27,82.7);
insert into Usage_History
values(15,'2023-10-3','2023-06-18',52.5,102.54);
insert into Usage_History
values(15,'11/3/2023','11/6/2023',37.77,82.54);
insert into Usage_History
values(16,'11/3/2023','11/6/2023',66.77,122.5);
insert into Usage_History
values(17,'2023-10-3','2023-06-18',37.5,60.5);
insert into Usage_History
values(17,'2023-09-3','2023-06-15',35.9,69.9);
insert into Usage_History
values(17,'11/3/2023','11/6/2023',35.7,69.8);
insert into Usage_History
values(18,'11/3/2023','11/08/2023',37.9,77.8);

insert into Usage_History
values(19,'2023-10-3','2023-07-18',41.5,92.4);
insert into Usage_History
values(19,'11/3/2023','11/05/2023',34.9,74.3);
insert into Usage_History
values(20,'11/3/2023','11/10/2023',44.4,88.8);
insert into Usage_History
values(21,'11/3/2023','11/15/2023',28.6,52.4);
insert into Usage_History
values(22,'11/3/2023','11/18/2023',34.8,74.6);
insert into Usage_History
values(23,'2023-10-3','2023-06-18',39.6,80.3);
insert into Usage_History
values(23,'2023-09-3','2023-06-15',36.7,71.4);
insert into Usage_History
```

Group 5: EMS

```
values(23,'11/3/2023','11/06/2023',38.9,72.4);
insert into Usage_History
values(24,'2023-10-3','2023-12-18',47.4,90.3);
insert into Usage_History
values(24,'11/3/2023','11/6/2023',55.9,112.5);
insert into Usage_History
values(25,'2023-10-3','2023-10-18',44.8,90.5);
insert into Usage_History
values(25,'2023-09-3','2023-09-15',51.7,103.2);
insert into Usage_History
values(25,'11/3/2023','11/12/2023',41.7,83.4);
insert into Usage_History
values(26,'2023-10-3','2023-11-18',37.6,72.4);
insert into Usage_History
values(26,'11/3/2023','11/13/2023',33.6,66.7);
insert into Usage_History
values(27,'2023-10-3','2023-06-18',61.2,120.5);
insert into Usage_History
values(27,'2023-09-3','2023-06-15',41.4,83.2);
insert into Usage_History
values(27,'11/3/2023','11/06/2023',39.5,76.8);
insert into Usage_History
values(28,'2023-10-3','2023-08-18',53.4,101.2);
insert into Usage_History
values(28,'11/3/2023','11/12/2023',28.9,60.2);

insert into Usage_History
values(29,'2023-10-3','2023-09-18',36.7,72.2);
insert into Usage_History
values(29,'11/3/2023','11/15/2023',54.6,102.2);

insert into Usage_History
values(30,'2023-10-3','2023-05-18',45.38,90.5);
insert into Usage_History
values(30,'11/3/2023','11/06/2023',43.34,86.9);
/*-----*/
```

Result

Table Explanation: Usage_History table provides the information about the reminders sent to customers regarding the bill payments.

Query

```
select * from Usage_History;
```

Group 5: EMS

Screenshot

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost - SQLQuery_2 - localhost\MS (sa)
- Database:** EMS
- Query:** 567 select * from Usage_History;
- Results:** The results grid displays data from the Usage_History table. The columns are: Usage_Id, Customer_Id, Bill_Generated_Date, Bill_payment_Date, Total_Amount, and Total_Consumption.
- Data Preview:** The table contains 25 rows of data, starting with row 1 and ending with row 25.
- Bottom Status Bar:** Shows Ln 567, Col 1 (28 selected), Spaces: 4, UTF-8, LF, 36 rows, MSSQL, 00:00:00, localhost : EMS.

Section 2: Data Retrieval and Analytical Reports

Overview: Raw data can be used to extract useful information using SQL, and it plays a major role in the data retrieval. This retrieved data can further be used by any company to generate insights such as generating analytics or reports, and to make data driven decisions. Here in this section of data analysis, SQL queries are generated using the SQL SELECT statement along with the brief explanation of the statement, and the result displayed is captured as a screenshot.

(*Comments: Before executing queries in this section 2, queries in section 1 should be executed avoid errors in extracting the results.)

Data Analysis 1

Statement: Get all the details related to customer and sort it by the customer first name.

Sql command explanation:

- Command select * is used to get all the columns from the table.
- Command from is used to retrieve data from the table.
- Command order by Customer_FName is used to sort the result by the Customer

SQL query:

```
select * from Customers  
order by Customer_FName;
```

Group 5: EMS

Result and screenshot:

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost, SQLQuery_3 - localh...MS (sa), SQLQuery_2 - localh...MS (sa), dbo.Reminders, SQLQuery_1 - localh...MS (sa).
- Database:** EMS
- Query:** select * from Customers order by Customer_FName;
- Results:** A grid of 25 rows showing customer data. The columns are Customer_Id, Customer_FName, Customer_LName, Customer_Email, Customer_Contact, Customer_Location, and Supplier_Id.
- Table Headers:** Customer_Id, Customer_FName, Customer_LName, Customer_Email, Customer_Contact, Customer_Location, Supplier_Id.
- Sample Data:** Row 1: Customer_Id 1, Customer_FName Adam, Customer_LName Smith, Customer_Email adamsmith@gmail.com, Customer_Contact 9453245679, Customer_Location Teasley Lane, Ph, Supplier_Id 7.

Data Analysis 2

Statement: Get all the details related to supplier and sort it by the supplier name.

Sql command explanation:

- Command select * is used to get all the columns from the table.
- Command from is used to retrieve data from the table.
- Command order by supplier name is used to sort the result by the supplier name.

SQL query:

```
select * from Service_Provider
order by Supplier_Name;
```

Group 5: EMS

Result and screenshot:

The screenshot shows the SSMS interface with the following details:

- Connections:** localhost, SQLQuery_3 - localh...MS (sa), SQLQuery_2 - localh...MS (sa), dbo.Reminders, SQLQuery_1 - localh...MS (sa).
- Database:** EMS.
- Query:** select * from Service_Provider order by Supplier_Name;
- Results Grid:** Displays 12 rows of data from the Service_Provider table. The columns are Supplier_Id, Supplier_Name, Supplier_Email, and Supplier_Contact.

Supplier_Id	Supplier_Name	Supplier_Email	Supplier_Contact
1	Amigo Power	info@Amigopower.com	9162553475
2	Ampra Energy	info@amrapower.com	7253856495
3	Compare Power	info@comparepower.com	8462758364
4	Cupa power	info@cupapower.com	5837485737
5	Energy Texas	info@energytexas.com	8371698365
6	Oncor	info@oncor.com	8263758293
7	Onver Power	info@onverpower.com	5837469382
8	Reliant	info@reliant.com	8332359653
9	spectru powers	info@spectrumpowers.com	9472837485
10	Tara energy	info@taraenergy.com	7239476849
11	Tata Powers	info@tatapower.com	8385945848
12	Zampa Distributions	info@zampadistributions.com	4938572647

- Bottom Status Bar:** Ln 46, Col 24, Spaces: 4, UTF-8, LF, 12 rows, MSSQL, 00:00:00, localhost : EMS.

Data Analysis 3

Statement: Get the total number of customers in a region and sort it by region.

Sql command explanation:

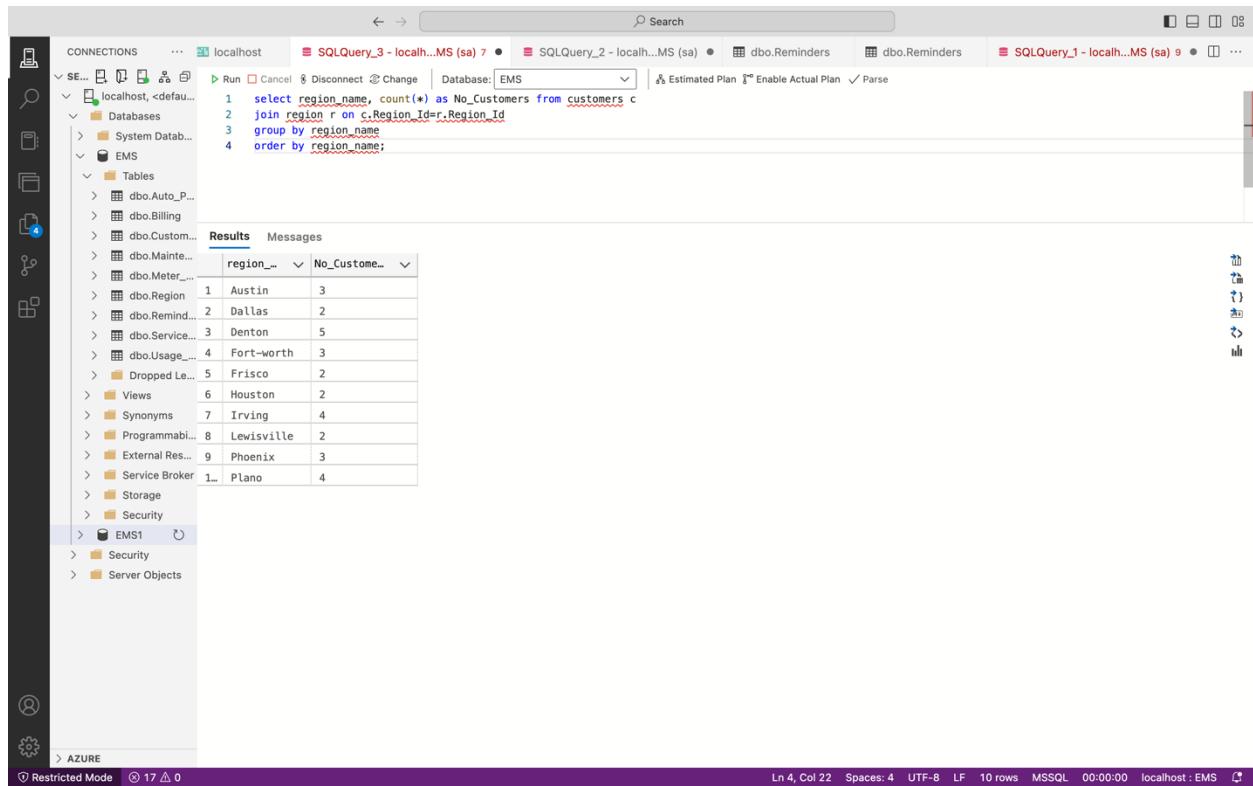
- Command “Use EMS” is used to call the EMS database.
- Command count(*) is used to get the number of customers.
- Command as is used to rename the column count(*) as an alias.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command group by Region is used to aggregate the data by Region.
- Command order by Region is used to sort the result by the Region name.

SQL query:

```
select region_name, count(*) as No_Customers from customers c
join region r on c.Region_Id=r.Region_Id
group by region_name
order by region_name;
```

Group 5: EMS

Result and screenshot:



The screenshot shows the SSMS interface with a query window open. The query is:

```
1 select region_name, count(*) as No_Customers from customers c
2 join region r on c.Region_Id=r.Region_Id
3 group by region_name
4 order by region_name;
```

The results grid displays the following data:

region_name	No_Customers
Austin	3
Dallas	2
Denton	5
Fort-worth	3
Frisco	2
Houston	2
Irving	4
Lewisville	2
Phoenix	3
Plano	4

Data Analysis 4

Statement: Get the total number of customers who signed-up for supplier and sort it by supplier name.

Sql command explanation:

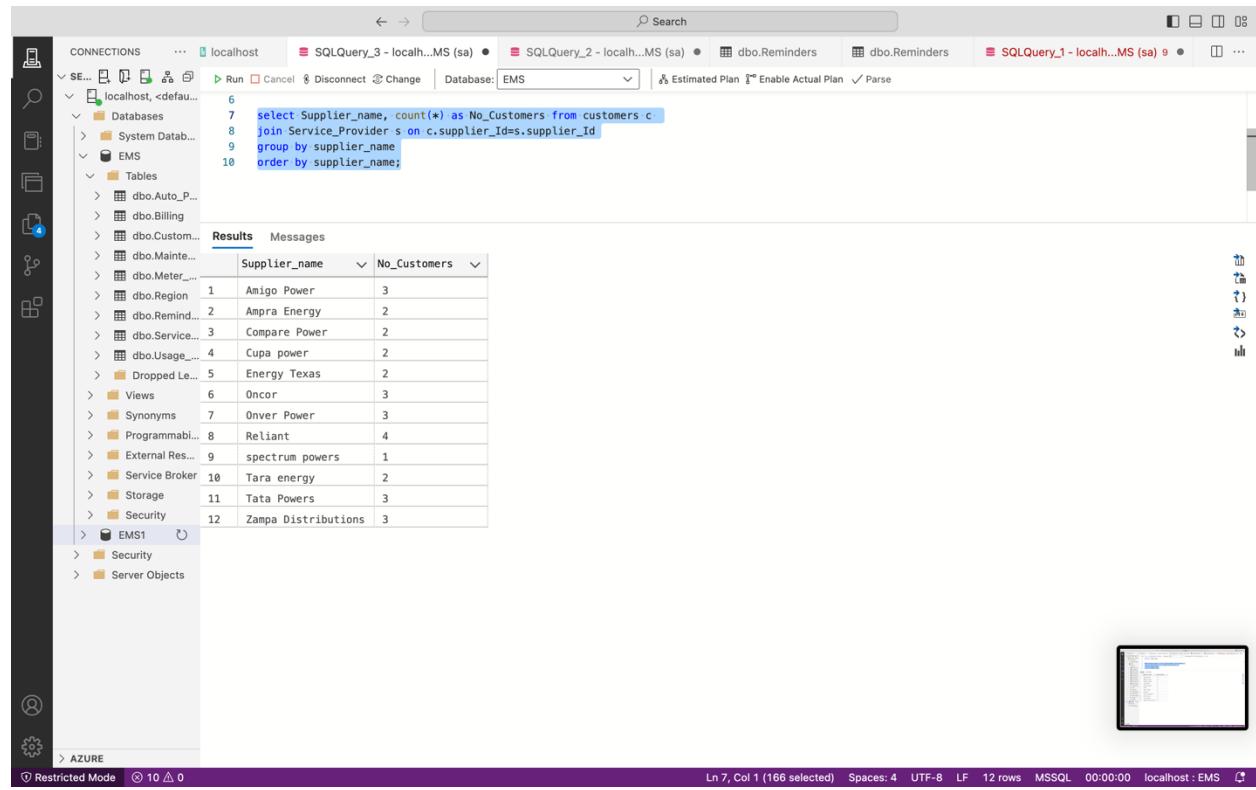
- Command count(*) is used to get the number of customers.
- Command as is used to rename the column count(*) as an alias.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command group by Supplier_name is used to aggregate the data by Supplier_Name.
- Command order by Supplier_name is used to sort the result by the Supplier_Name.

SQL query:

```
select Supplier_name, count(*) as No_Customers from customers c
join Service_Provider s on c.supplier_Id=s.supplier_Id
group by supplier_name
order by supplier_name;
```

Group 5: EMS

Result and screenshot:



The screenshot shows the SSMS interface with the following details:

- Connections:** localhost
- Database:** EMS
- Query:** A T-SQL script is displayed in the query pane:

```
6
7 select Supplier_name, count(*) as No_Customers from customers c
8 join Service_Provider s on c.supplier_Id=s.supplier_Id
9 group by supplier_name
10 order by supplier_name;
```
- Results:** A grid table titled "Results" shows the count of customers per supplier name. The data is as follows:

Supplier_name	No_Customers
Amigo Power	3
Ampra Energy	2
Compare Power	2
Cupa power	2
Energy Texas	2
Oncor	3
Onver Power	3
Reliant	4
spectrum powers	1
Tara energy	2
Tata Powers	3
Zamp Distributions	3

Data Analysis 5

Statement: List the customer name, reminders sent to customers and sort it by customer name.

Sql command explanation:

- Command from is used to retrieve data from the table.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command order by Customer_Name is used to sort the result by the Customer_Name.

SQL query:

```
select customer_Fname,Customer_LName, r.Reminder from Customers c
join Reminders r on c.Customer_Id=r.Customer_Id
order by Customer_FName;
```

Group 5: EMS

Result and screenshot:

```

14 select customer_Fname,Customer_LName, r.Reminder from Customers c
15 join Reminders r on c.Customer_Id=r.Customer_Id
16
17
18

```

	customer_Fname	Customer_LName	Reminder
1	Adam	Smith	Your bill has paid usig autopayment method
2	Anil	kumbley	Your generated bill was due on 11/20/2023
3	Anvar	mith	Your bill has paid usig autopayment method
4	Birkey	wing	Your generated bill was due on 11/20/2023
5	cane	walter	Your bill has paid usig autopayment method
6	Cristina	Rose	Your bill has paid usig autopayment method
7	Kalyan	Tinku	Your generated bill was due on 11/20/2023
8	Kane	county	Your bill has paid usig autopayment method
9	king	william	Your generated bill was due on 11/20/2023
10	Kiran Kumar	Mintu	Your bill has paid usig autopayment method
11	Micheal	Tom	Your bill has paid usig autopayment method
12	Mike	Tong	Your generated bill was due on 11/20/2023
13	miller	wright	Your generated bill was due on 11/20/2023
14	Pavan	Naini	Your generated bill was due on 11/20/2023
15	priya	chandran	Your bill has paid usig autopayment method
16	Raghu	nandhan	Your bill has paid usig autopayment method
17	ram	charan	Your generated bill was due on 11/20/2023
18	ram	singh	Your generated bill was due on 11/20/2023
19	Ringer	rale	Your generated bill was due on 11/20/2023
20	Rishi	kumar	Your generated bill was due on 11/20/2023
21	Ronald	Wright	Your generated bill was due on 11/20/2023
22	Ruth	ranger	Your generated bill was due on 11/20/2023
23	Sruthi	Kiran	Your generated bill was due on 11/20/2023
24	stella	fring	Your generated bill was due on 11/20/2023
25	Terissa	Tim	Your bill has paid usig autopayment method
26	Tom	Crissey	Your generated bill was due on 11/20/2023

Data Analysis 6

Statement: Get the meter readings and the customer full name and sort it by the meter readings.

Sql command explanation:

- Command from is used to retrieve data from the table.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command order by Meter_Readings is used to sort the result by the Meter_Readings

SQL query:

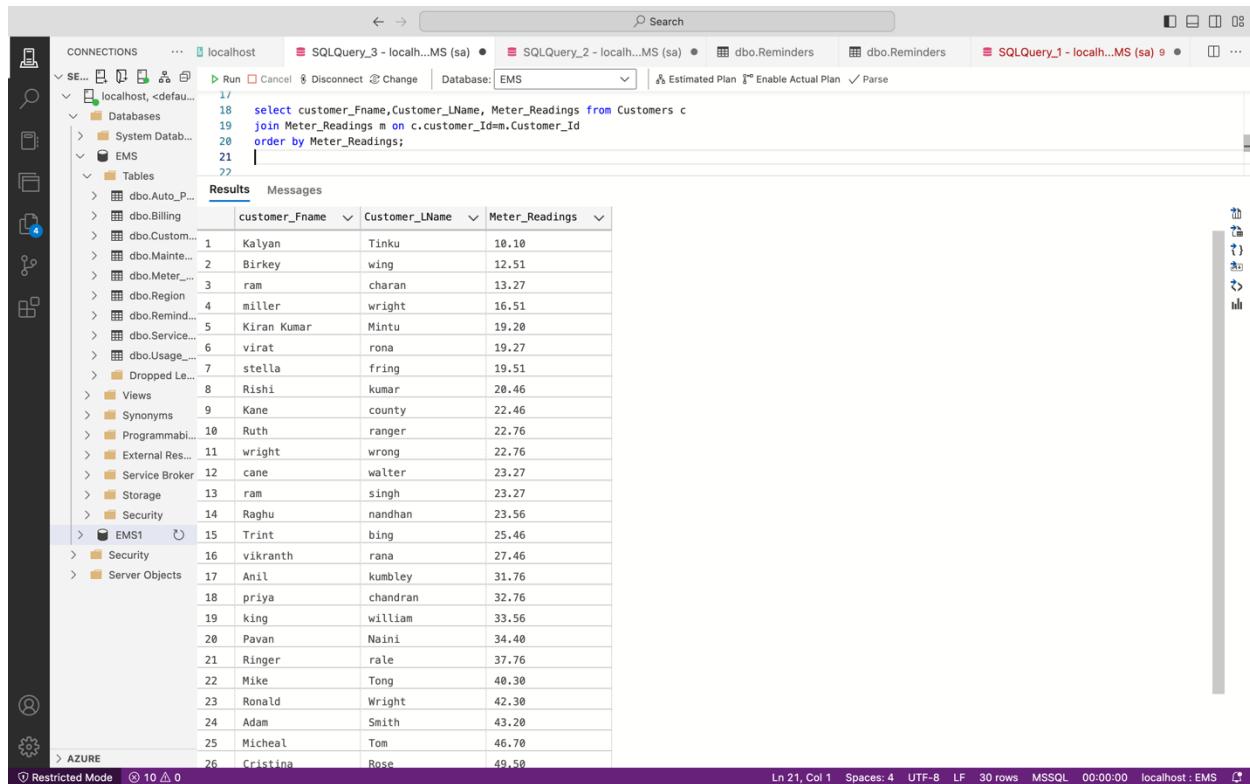
```

select customer_Fname,Customer_LName, Meter_Readings from Customers c
join Meter_Readings m on c.customer_Id=m.Customer_Id
order by Meter_Readings;

```

Group 5: EMS

Result and screenshot:



The screenshot shows the SSMS interface with the following details:

- Connections:** localhost, SQLQuery_3 - localh...MS (sa), SQLQuery_2 - localh...MS (sa), dbo.Reminders, SQLQuery_1 - localh...MS (sa).
- Database:** EMS.
- Query:** A T-SQL select statement:

```
17
18    select customer_Fname,Customer_LName, Meter_Readings from Customers c
19    join Meter_Readings m on c.customer_Id=m.Customer_Id
20    order by Meter_Readings;
```
- Results Grid:** Displays 30 rows of data with columns: customer_Fname, Customer_LName, and Meter_Readings. The data includes names like Kalyan, Tinku, Birkey, wing, ram, charan, miller, wright, Kiran Kumar, Mintu, virat, rona, stella, fring, Rishi, kumar, Kane, county, Ruth, ranger, External Res..., wrong, Service Broker, cane, walter, Storage, ram, singh, Raghu, nandhan, Trint, bing, vikranth, rana, Anil, kumbley, priya, chandran, King, william, Pavan, Naini, Ringer, rale, Mike, Tong, Ronald, Wright, Adam, Smith, Micheal, Tom, Cristina, Rose, and a final value of 49.50.
- Status Bar:** Shows Ln 21, Col 1, Spaces: 4, UTF-8, LF, 30 rows, MSSQL, 00:00:00, localhost : EMS.

Data Analysis 7

Statement: Get the billing details of customer whose billing amount is over \$50 and order the result in descending order using bill amount.

Sql command explanation:

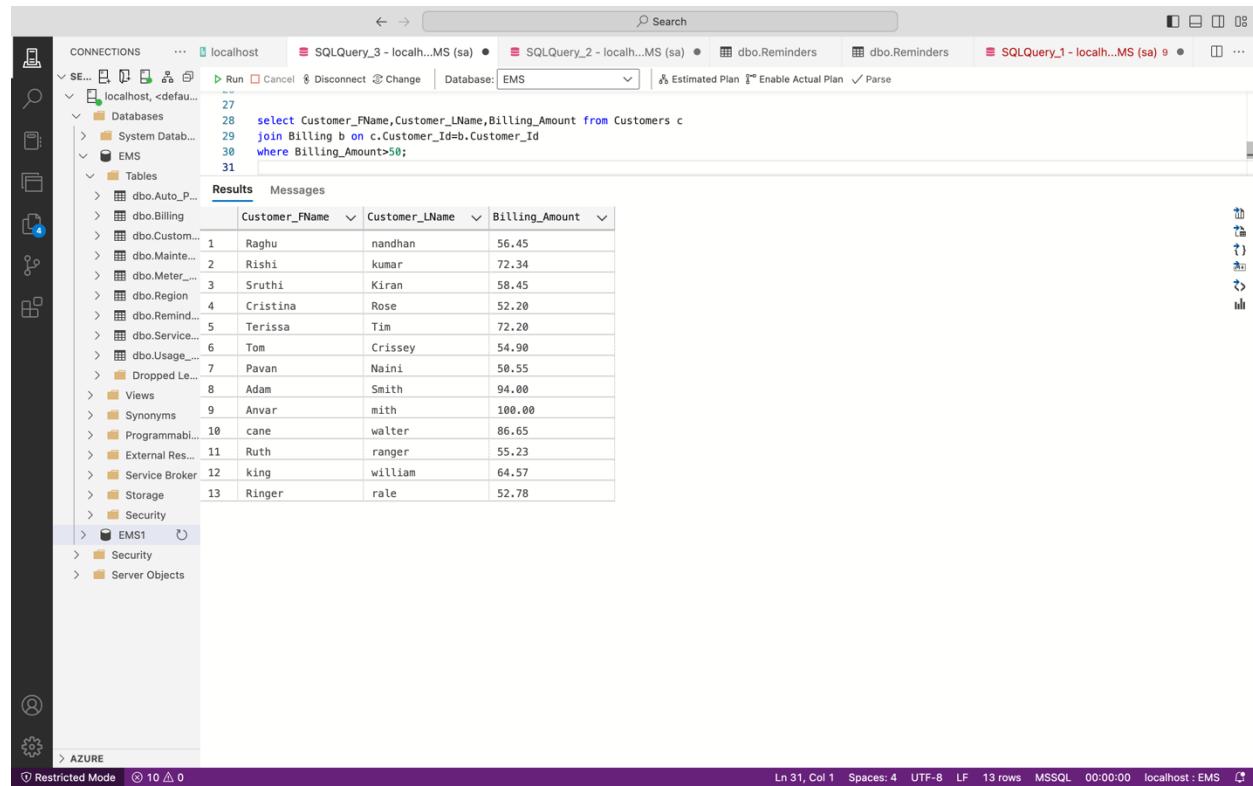
- Command from is used to retrieve data from the table.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command where is used to apply the condition to the query where the bill amount is over 50\$.
- Command order by Billing_Amount is used to sort the result by the Billing_Amount.

SQL query:

```
select Customer_FName,Customer_LName,Billing_Amount from Customers c
join Billing b on c.Customer_Id=b.Customer_Id
where Billing_Amount>50;
```

Group 5: EMS

Result and screenshot:



The screenshot shows the SSMS interface with the following details:

- Connections:** localhost, SQLQuery_3 - localhost..MS (sa), SQLQuery_2 - localhost..MS (sa), dbo.Reminders, SQLQuery_1 - localhost..MS (sa).
- Database:** EMS
- Query:** A T-SQL select statement:

```
select Customer_FName,Customer_LName,Billing_Amount from Customers c
join Billing b on c.Customer_Id=b.Customer_Id
where Billing_Amount>50;
```
- Results Grid:** Displays 13 rows of customer data with columns: Customer_FName, Customer_LName, and Billing_Amount.
- Bottom Status Bar:** Shows Ln 31, Col 1, Spaces: 4, UTF-8, LF, 13 rows, MSSQL, 00:00:00, localhost : EMS.

Data Analysis 8

Statement: Get the maintenance requests description raised by the customers with date and sort it by the request date.

Sql command explanation:

- Command from is used to retrieve data from the table.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command order by Maintenancerequest_Date is used to sort the result by the Maintenancerequest_Date.

SQL query:

```
select Customer_FName,Customer_LName,[Description],Maintenancerequest_Date from
Customers c
join Maintenance_Requests m on c.Customer_Id=m.Customer_Id
order by Maintenancerequest_Date;
```

Group 5: EMS

Result and screenshot:

```

SELECT Customer_FName, Customer_LName, [Description], Maintenancerequest_Date
FROM Customers c
JOIN Maintenance_Requests m ON c.Customer_Id = m.Customer_Id
ORDER BY Maintenancerequest_Date;

```

	Customer_FName	Customer_LName	Description	Maintenancerequest_Date
1	Cristina	Rose	Request for Installation of meter	2023-04-23
2	Raghu	nandhan	Request for Installation of meter	2023-06-12
3	Ringer	rale	Request for Installation of meter	2023-07-12
4	vikranth	rana	Request for Installation of meter	2023-08-03
5	Tom	Crisey	Request for Installation of meter	2023-08-06
6	cane	walter	Request for Installation of meter	2023-08-13
7	Birkey	wing	Request for Installation of meter	2023-08-13
8	priya	chandran	Request for Installation of meter	2023-08-23
9	Terissa	Tim	Request for Installation of meter	2023-09-02
10	Anil	kumbley	Request for change of meter	2023-09-06
11	Ruth	ranger	Request for change of meter	2023-09-12
12	Michaels	Tom	Request for Installation of meter	2023-10-01
13	ram	singh	Request for change of meter	2023-10-04
14	Ronald	Wright	Request for Installation of meter	2023-10-05
15	Adam	Smith	Request for change of meter	2023-11-04
16	Trint	bing	Request for Installation of meter	2023-11-13
17	Kalyan	Tinku	Request for Installation of meter	2023-12-02

Data Analysis 9

Statement: Get the usage history of the customers in the month of November whose billing amount is over 30\$ and sort it by bill payment date.

Sql command explanation:

- Command from is used to retrieve data from the table.
- Command join is used to join two tables together with ON to specify which column is used to join the table.
- Command where is used to apply the condition to the query where the bill amount is over 30\$ and bill generated in the month November.
- Command order by Bill_Payment_Date is used to sort the result by the Bill_Payment_Date.

SQL query:

```

SELECT Customer_FName, Customer_LName
,Bill_Generated_Date, Bill_payment_Date, Total_Amount, Total_Consumption FROM
Usage_History u
JOIN Customers c ON u.Customer_Id=c.Customer_Id
WHERE MONTH(Bill_Generated_Date)=11 AND Total_Amount>30
ORDER BY Bill_payment_Date;

```

Group 5: EMS

Result and screenshot:

```

39 select Customer_FName,Customer_LName ,Bill_Generated_Date,Bill_payment_Date,Total_Amount,Total_Consumption from Usage_History u
40 join Customers c on u.Customer_Id=c.Customer_Id
41 where MONTH(Bill_Generated_Date)=11 and Total_Amount>30
42 order by Bill_payment_Date;
43

```

	Customer_FName	Customer_LName	Bill_Generated_Date	Bill_payment_Date	Total_Amount	Total_Consumption
1	Ruth	ranger	2023-11-03	2023-11-05	34.90	74.30
2	Trint	bing	2023-11-03	2023-11-06	38.90	72.40
3	Kiran Kumar	Mintu	2023-11-03	2023-11-06	42.27	82.70
4	Adam	Smith	2023-11-03	2023-11-06	37.77	82.54
5	Anvar	mith	2023-11-03	2023-11-06	66.77	122.50
6	cane	walter	2023-11-03	2023-11-06	35.70	69.80
7	Raghu	nandhan	2023-11-03	2023-11-06	46.77	92.50
8	Raghu	nandhan	2023-11-03	2023-11-06	46.77	92.50
9	Cristina	Rose	2023-11-03	2023-11-06	33.60	67.30
1..	Terissa	Tim	2023-11-03	2023-11-06	39.50	80.40
1..	Micheal	Tom	2023-11-03	2023-11-06	50.40	100.80
1..	Pavan	Naini	2023-11-03	2023-11-06	41.70	90.20
1..	Sruthi	Kiran	2023-11-03	2023-11-07	36.50	70.20
1..	Kane	county	2023-11-03	2023-11-08	37.90	77.80
1..	Tom	Crissley	2023-11-03	2023-11-08	53.50	104.30
1..	Birkey	wing	2023-11-03	2023-11-10	44.40	88.80
1..	Rishi	kumar	2023-11-03	2023-11-11	33.70	66.90
1..	ram	charan	2023-11-03	2023-11-12	36.77	72.50
1..	Kalyan	Tinku	2023-11-03	2023-11-13	33.33	66.66
2..	Mike	Tong	2023-11-03	2023-11-15	60.30	130.40
2..	ram	singh	2023-11-03	2023-11-18	34.80	74.60

Data Analysis 10

Statement: Get the total consumption of electricity from suppliers which is used by the customers and display the column as Total_Consumption.

Sql command explanation:

- Command from is used to retrieve data from the table.
- Command join is used to join two or more tables together with ON to specify which column is used to join the table.
- Command group by Supplier_name is used to aggregate the data by Supplier_Name.
- Command order by total_consumption is used to sort the result by the total_Consumption.

SQL query:

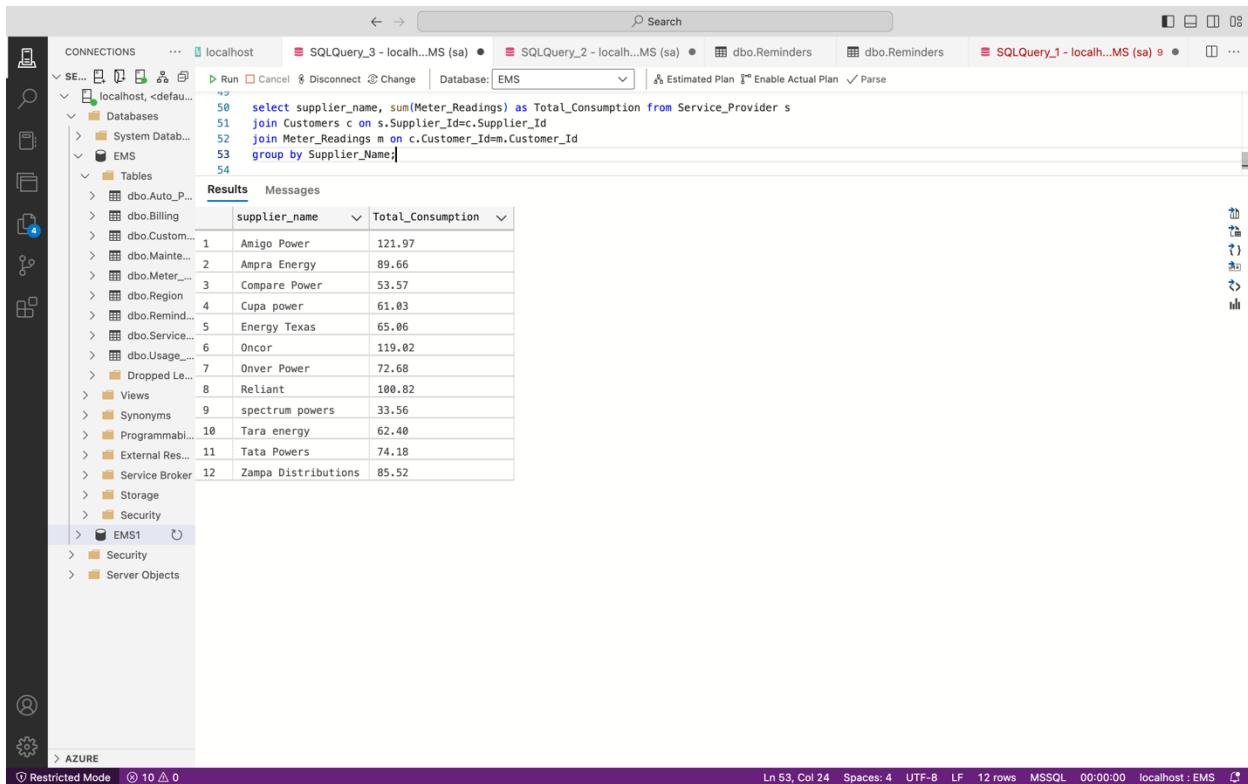
```

select supplier_name, sum(Meter_Readings) as Total_Consumption from Service_Provider s
join Customers c on s.Supplier_Id=c.Supplier_Id
join Meter_Readings m on c.Customer_Id=m.Customer_Id
group by Supplier_Name;

```

Group 5: EMS

Result and screenshot:



The screenshot shows the SSMS interface with a query window open. The query is:

```
select supplier_name, sum(Meter_Reading) as Total_Consumption from Service_Provider s  
join Customers c on s.Supplier_Id=c.Supplier_Id  
join Meter_Reading m on c.Customer_Id=m.Customer_Id  
group by Supplier_Name;
```

The results grid displays the following data:

supplier_name	Total_Consumption
Amigo Power	121.97
Ampra Energy	89.66
Compare Power	53.57
Cupa power	61.03
Energy Texas	65.06
Oncor	119.02
Onver Power	72.68
Reliant	100.82
spectrum powers	33.56
Tara energy	62.40
Tata Powers	74.18
Zampa Distributions	85.52

Appendix 1

```

/*-----Create database EMS-----*/
create database EMS;
/*-----Create Service_Provider Table-----*/
use EMS;
create table Service_Provider(
    Supplier_Id Int identity(1,1) primary key,
    Supplier_Name Varchar(50) not null,
    Supplier_Email Varchar(50) not null,
    Supplier_Contact Varchar(10) not null
);
/*-----Create table Region-----*/
create table Region(
    Region_Id int identity(1,1) primary key,
    Region_Name varchar(50) not null
);
/*-----Create table Customers-----*/

create table Customers(
Customer_Id int identity(1,1) primary key,
Customer_FName varchar(50) not null,
Customer_LName varchar(50) not null,
Customer_Email varchar(50) not null,
Customer_Contact varchar(10) not null,
Customer_Location varchar(150) not null,
Supplier_Id int not null,
Region_Id int not null,
Enrolledfor_Autopayments bit default 0 not null,
foreign key(Supplier_Id) references Service_Provider(Supplier_Id),
foreign key(Region_Id) references Region(Region_Id)
);
/*-----Create Maintenance_Requests Table-----*/
create table Maintenance_Requests(
    Maintenancerequest_Id int identity(1,1) primary key,
    Supplier_Id int not null,
    Region_Id int default null,
    Customer_Id int default null,
    Maintenancerequest_Date date not null,
    Description varchar(500) default ' ',
    foreign key(Supplier_Id) REFERENCES Service_Provider(Supplier_Id)
);
/*-----Create Meter_Readings Table-----*/
create table Meter_Readings(
    Meter_Id int identity(1000,1) primary key,
    Customer_Id int not null,
    Meter_Readings decimal(9,2) default 0.00,

```

Group 5: EMS

```
    Readings_Date date not null,
    foreign key(Customer_Id) references Customers(Customer_Id)
);
/*-----Create Auto_Payments Table-----*/
create table Auto_Payments(
    Autopayment_Id int identity(1,1) primary key,
    Customer_Id int unique not null,
    Card_Type varchar(25) not null,
    Card_Number varchar(19) not null unique,
    Expiry_Date Date not null,
    CVV int not null,
    Billing_Address varchar(250) not null,
    foreign key(Customer_Id) REFERENCES Customers(Customer_Id)
);
/*-----Create Billing Table-----*/
create table Billing(
    Billing_Id int identity(1,1) primary key,
    Billing_Amount decimal(9,2) default 0.00,
    Customer_Id int unique not null,
    Total_Consumption decimal(9,2) default 0.00,
    Supplier_Id int not null,
    Billing_Date date,
    foreign key(Customer_Id) references Customers(Customer_Id)
);
/*----- Reminders Table-----*/
create table Reminders(
    Reminder_Id int identity(100,1) primary key,
    Reminder varchar(250) default 'No reminders or changed reminder preferences',
    Customer_Id int not null,
    foreign key(Customer_Id) REFERENCES Customers(Customer_Id)
);
/*-----Usage_History Table-----*/
create table Usage_History(
    Usage_Id int identity(1,1) primary key,
    Customer_Id int not null,
    Bill_Generated_Date date not null,
    Bill_payment_Date date not null,
    Total_Amount decimal(9,2),
    Total_Consumption decimal(9,2),
    foreign key(Customer_Id) references Customers(Customer_Id)
);

/*-----Insert values to Service_Provider Table-----*/
use EMS;
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Reliant','8332359653','info@reliant.com');
```

Group 5: EMS

```
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Oncor','8263758293','info@oncor.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Compare Power','8462758364','info@comparepowwer.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Amigo Power','9162553475','info@Amigopower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Ampra Energy','7253856495','info@amprapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Energy Texas','8371698365','info@energytexas.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Tara energy','7239476849','info@taraenergy.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Onver Power','5837469382','info@onverpower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Tata Powers','8385945848','info@tatapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Cupa power','5837485737','info@cupapower.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('Zampa Distributions','4938572647','info@zampadistributions.com');
insert into service_provider(Supplier_Name,Supplier_Contact,Supplier_EMail)
values('spectrum powers','9472837485','info@tspectrumpowers.com');
/*-----Inserting values into Region table-----*/
insert into Region values('Denton');
insert into Region values('Dallas');
insert into Region values('Houston');
insert into Region values('Austin');
insert into Region values('Irving');
insert into Region values('Plano');
insert into Region values('Frisco');
insert into Region values('Fort-worth');
insert into Region values('Lewisville');
insert into Region values('Phoenix');
/*-----Inserting values into Customers table-----*/
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Raghu', 'nandhan', '4857362847', 'raghu@gmail.com', 'union circle, denton, tx', 1, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('ram', 'charan', '4837285674', 'ram@gmail.com', 'oaks of denton, denton, TX', 3, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Email, Customer_Contact, Customer_Location,
Supplier_Id, Region_Id)
```

Group 5: EMS

```
values('Rishi','kumar','rishi@gmail.com','8573746385','valley ranch pkwy, Irving,  
TX',1,5);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('priya','chandran','9472847364','pricha@gmail.com','oaks at valley ranch,  
Irving, Tx',2,5,1);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Sruthi','Kiran','9263828462','kiran@gmail.com','Swadeshi plaza, Irving,  
TX',4,5,0);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Cristina','Rose','6284638472','cristina@gmail.com','Wilmington pkwy, Fort-  
worth, Tx',4,8,1);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Ronald','Wright','7483648264','ronald@gmail.com','Dallas pkwy, dallas,  
Tx',6,2,0);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Terissa','Tim','9273846284','teressa@gmail.com','Loscolinas pkwy, houston,  
Tx',5,3,1);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Mike','Tong','9374738472','Mike@gmail.com','Spring valley rd,plano,  
Tx',3,6,0);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Tom','Crissey','8284637463','Tom@gmail.com','Spring valley rd,plano,  
Tx',2,6,0);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Micheal','Tom','8264927482','Micheal@gmail.com','Montfort dr, frisco,  
Tx',1,7,1);  
insert into  
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,  
Supplier_Id, Region_Id, Enrolledfor_Autopayments)  
values('Pavan','Naini','9405959573','pavan@gmail.com','Ioof St, Houston, Tx',9,3,1);
```

Group 5: EMS

```
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Kalyan', 'Tinku', '9890987890', 'tinku@gmail.com', 'Bernald St denton, Tx', 1, 1, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Kiran Kumar', 'Mintu', '9309439890', 'kirankumar@gmail.com', 'lebin street, lw', 7, 9, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Adam', 'Smith', '9453245679', 'adamsmith@gmail.com', 'Teasley Lane, Ph', 7, 10, 12);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Anvar', 'mith', '8293746738', 'anvar@gmail.com', 'Timber st, Lewisville, Tx', 11, 9, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('cane', 'walter', '3947584839', 'cane@gmail.com', 'Twistley rd, Frisco, Tx', 10, 7, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Kane', 'county', '5928385858', 'kane@gmail.com', 'winter ln, Plano, Tx', 8, 6, 1);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Ruth', 'ranger', '7838485868', 'ruth@gmail.com', 'Rinder pkwy, Dallas, Tx', 6, 2, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('Birkey', 'wing', '8273648573', 'birkey@gmail.com', 'oaks of denton, Denton, Tx', 11, 1, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('king', 'william', '7394829485', 'kingwilliam@gmail.com', 'trails of austin, Austin, Tx', 12, 4, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
values('ram', 'singh', '8483847275', 'ramsingh@gmail.com', 'lukes of fort-worth, Fort-worth, Tx', 9, 8, 0);
insert into
Customers(Customer_FName, Customer_LName, Customer_Contact, Customer_Email, Customer_Location,
Supplier_Id, Region_Id, Enrolledfor_Autopayments)
```

Group 5: EMS

```
values('Trint','bing','8938475837','trinth@gmail.com','churchy st, Phoenix,
Tx',5,10,1);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('wright','wrong','8394388484','wright@gmail.com','trinty ln, Denton,
Tx',8,1,0);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('miller','wright','8273645273','miller@gmail.com','west street ln, Irving,
Tx',9,5,0);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('Anil','kumbley','7483746574','kumbley@gmail.com','Timber west ln, Austin,
Tx',2,4,0);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('virat','rona','7384958274','virat@gmail.com','valley pkwy E, Plano,
Tx',4,6,1);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('vikranth','rana','8938485828','vikranth@gmail.com','ruster rd, Fort-worth,
Tx',8,8,0);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('Ringer','rale','8394727475','ringer@gmail.com','winster pkwy, Austin,
Tx',10,4,0);
insert into
Customers(Customer_FName,Customer_LName,Customer_Contact,Customer_Email,Customer_Location,
Supplier_Id,Region_Id,Enrolledfor_Autopayments)
values('stella','fring','9284837475','stella@gmail.com','oaks of Phoenix, Phoenix,
Tx',11,10,0);
/*-----Inserting values into Maintenance_Requests Table-----*/
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(2,'2023-06-22','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
values(1,5,4,'2023-08-23','Request for Installation of meter');
insert into Maintenance_Requests
values(1,1,1,'2023-06-12','Request for Installation of meter');
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(5,'2023-08-11','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
```

Group 5: EMS

```
values(4,8,6,'2023-04-23','Request for Installation of meter');
insert into Maintenance_Requests
values(6,2,7,'2023-10-05','Request for Installation of meter');
insert into Maintenance_Requests
values(5,3,8,'2023-09-02','Request for Installation of meter');
insert into Maintenance_Requests
values(2,6,10,'2023-08-06','Request for Installation of meter');
insert into Maintenance_Requests
values(1,7,11,'2023-10-01','Request for Installation of meter');
insert into Maintenance_Requests
values(1,1,13,'2023-12-02','Request for Installation of meter');
insert into Maintenance_Requests
values(7,10,15,'2023-11-04','Request for change of meter');
insert into Maintenance_Requests
values(10,7,17,'2023-08-13','Request for Installation of meter');
insert into Maintenance_Requests(Supplier_Id,Maintenancerequest_Date,[Description])
values(3,'2023-11-10','Maintenance is scheduled at the supplier end annually');
insert into Maintenance_Requests
values(6,12,19,'2023-09-12','Request for change of meter');
insert into Maintenance_Requests
values(11,1,20,'2023-08-13','Request for Installation of meter');
insert into Maintenance_Requests
values(9,8,22,'2023-10-04','Request for change of meter');
insert into Maintenance_Requests
values(5,10,23,'2023-11-13','Request for Installation of meter');
insert into Maintenance_Requests
values(2,4,26,'2023-09-06','Request for change of meter');
insert into Maintenance_Requests
values(8,8,28,'2023-08-03','Request for Installation of meter');
insert into Maintenance_Requests
values(10,4,29,'2023-07-12','Request for Installation of meter');
/*-----Inserting values into Meter_Readings Table-----*/
insert into Meter_Readings
values(1,23.56,'2023-11-20');
insert into Meter_Readings
values(2,13.27,'2023-11-20');
insert into Meter_Readings
values(3,20.46,'2023-11-20');
insert into Meter_Readings
values(4,32.76,'2023-11-20');
insert into Meter_Readings
values(5,53.2,'2023-11-20');
insert into Meter_Readings
values(6,49.5,'2023-11-20');
insert into Meter_Readings
values(7,42.3,'2023-11-20');
insert into Meter_Readings
```

Group 5: EMS

```
values(8,64.2,'2023-11-20');
insert into Meter_Readings
values(9,40.3,'2023-11-20');
insert into Meter_Readings
values(10,54.5,'2023-11-20');
insert into Meter_Readings
values(11,46.7,'2023-11-20');
insert into Meter_Readings
values(12,34.4,'2023-11-20');
insert into Meter_Readings
values(13,10.1,'2023-11-20');
insert into Meter_Readings
values(14,19.2,'2023-11-20');
insert into Meter_Readings
values(15,43.2,'2023-11-20');
insert into Meter_Readings
values(16,53.5,'2023-11-20');
insert into Meter_Readings
values(17,23.27,'2023-11-20');
insert into Meter_Readings
values(18,22.46,'2023-11-20');
insert into Meter_Readings
values(19,22.76,'2023-11-20');
insert into Meter_Readings
values(20,12.51,'2023-11-20');
insert into Meter_Readings
values(21,33.56,'2023-11-20');
insert into Meter_Readings
values(22,23.27,'2023-11-20');
insert into Meter_Readings
values(23,25.46,'2023-11-20');
insert into Meter_Readings
values(24,22.76,'2023-11-20');
insert into Meter_Readings
values(25,16.51,'2023-11-20');
insert into Meter_Readings
values(26,31.76,'2023-11-20');
insert into Meter_Readings
values(27,19.27,'2023-11-20');
insert into Meter_Readings
values(28,27.46,'2023-11-20');
insert into Meter_Readings
values(29,37.76,'2023-11-20');
insert into Meter_Readings
values(30,19.51,'2023-11-20');
/*-----Inserting values into Auto_Payments Table-----*/
insert into Auto_Payments
```

Group 5: EMS

```
values(1,'credit','4783927594716258','2024-02-01',346,'union circle');
insert into Auto_Payments
values(4,'debit','8264857102859374','2025-12-01',153,'oaks at valley ranch');
insert into Auto_Payments
values(6,'Debit','5263846273849270','2025-12-01',134,'Wilmington pkwy, Fort-worth, tx');
insert into Auto_Payments
values(8,'debit','8263745261846370','2026-04-01',143,'Loscolinas pkwy, houston, tx');
insert into Auto_Payments
values(11,'debit','8394728463746370','2024-03-01',256,'Montfort dr, frisco, tx');
insert into Auto_Payments
values(12,'Credit','12345','2027-01-01',123,'Ioof st,tx');
insert into Auto_Payments
values(14,'Debit','2341234564332','2030-11-01',533,'lebin street, lw');
insert into Auto_Payments
values(15,'Credit','234567812234','2025-12-01',324,'bernard street,denton');
insert into Auto_Payments
values(16,'credit','2150816150957480','2031-01-01',587,'Discovery park dr');
insert into Auto_Payments
values(17,'credit','4838485969485860','2024-07-01',564,'Twistley rd,Frisco,Tx');
insert into Auto_Payments
values(18,'debit','9483847562736450','2025-06-01',498,'winter ln, Plano, Tx');
insert into Auto_Payments
values(23,'credit','2938485862949280','2024-08-01',734,'churchy st, Phoenix, Tx');
insert into Auto_Payments
values(27,'debit','7374859384759280','2027-11-01',435,'valley pkwy E, Plano, Tx');
/*-----Inserting values into Billing Table-----*/
insert into Billing
values(56.45,1,134.45,1,'2023-11-03');
insert into Billing
values(43.25,2,108.42,2,'2023-11-03');
insert into Billing
values(72.34,3,184.3,1,'2023-11-03');
insert into Billing
values(38.34,4,90.2,2,'2023-11-03');
insert into Billing
values(58.45,5,105.32,4,'2023-11-03');
insert into Billing
values(52.2,6,105.36,4,'2023-11-03');
insert into Billing
values(48.4,7,89.9,6,'2023-11-03');
insert into Billing
values(72.2,8,130.36,5,'2023-11-03');
insert into Billing
values(44.43,9,82.6,3,'2023-11-03');
insert into Billing
values(54.9,10,103.5,2,'2023-11-03');
```

Group 5: EMS

```
insert into Billing
values(48.89,11,86.76,1,'2023-11-03');
insert into Billing
values(50.55,12,75,9,'2023-11-03');
insert into Billing
values(12,13,31,1,'2023-11-03');
insert into Billing
values(45,14,12,7,'2023-11-03');
insert into Billing
values(94,15,56,7,'2023-11-03');
insert into Billing
values(100,16,200,11,'2023-11-03');
insert into Billing
values(86.65,17,164.6,10,'2023-11-03');
insert into Billing
values(46.87,18,128.8,8,'2023-11-03');
insert into Billing
values(55.23,19,104.45,6,'2023-11-03');
insert into Billing
values(37.87,20,74.45,11,'2023-11-03');
insert into Billing
values(64.57,21,124.45,12,'2023-11-03');
insert into Billing
values(37.43,22,84.55,9,'2023-11-03');
insert into Billing
values(42.45,23,90.43,5,'2023-11-03');
insert into Billing
values(39.78,24,80.15,8,'2023-11-03');
insert into Billing
values(37.67,25,82.55,9,'2023-11-03');
insert into Billing
values(46.23,26,101.86,2,'2023-11-03');
insert into Billing
values(48.65,27,96.23,4,'2023-11-03');
insert into Billing
values(39.78,28,92.45,8,'2023-11-03');
insert into Billing
values(52.78,29,108.36,10,'2023-11-03');
insert into Billing
values(49.67,30,103.28,11,'2023-11-03');
/*-----Inserting values into Reminders Table-----*/
insert into Reminders
VALUES('Your bill has paid usig autopayment method',1);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',2);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',3);
```

Group 5: EMS

```
insert into Reminders
VALUES('Your bill has paid usig autopayment method',4);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',5);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',6);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',7);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',8);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',9);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',10);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',11);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',12);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',13);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',14);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',15);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',16);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',17);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',18);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',19);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',20);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',21);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',22);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',23);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',24);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',25);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',26);
insert into Reminders
VALUES('Your bill has paid usig autopayment method',27);
```

Group 5: EMS

```
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',28);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',29);
insert into Reminders
VALUES('Your generated bill was due on 11/20/2023',30);
/*-----Inserting values into Usage_History Table-----*/
insert into Usage_History
values(1,'2023-10-3','2023-10-18',56.78,126.54);
insert into Usage_History
values(1,'2023-09-3','2023-09-15',53.18,106.54);
insert into Usage_History
values(1,'11/3/2023','11/6/2023',46.77,92.5);

insert into Usage_History
values(2,'2023-10-3','2023-10-18',46.78,88.54);
insert into Usage_History
values(2,'11/3/2023','11/12/2023',36.77,72.5);

insert into Usage_History
values(3,'11/3/2023','11/11/2023',33.7,66.9);

insert into Usage_History
values(4,'11/3/2023','11/6/2023',20.3,40.6);

insert into Usage_History
values(5,'2023-10-3','2023-10-18',36.3,70.1);
insert into Usage_History
values(5,'11/3/2023','11/07/2023',36.5,70.2);

insert into Usage_History
values(6,'11/3/2023','11/6/2023',33.6,67.3);

insert into Usage_History
values(7,'11/3/2023','11/08/2023',29.6,60.4);

insert into Usage_History
values(8,'11/3/2023','11/6/2023',39.5,80.4);

insert into Usage_History
values(9,'11/3/2023','11/15/2023',60.3,130.4);
```

Group 5: EMS

```
insert into Usage_History
values(10,'11/3/2023','11/8/2023',53.5,104.3);

insert into Usage_History
values(11,'2023-10-3','2023-06-18',44.7,95.6);
insert into Usage_History
values(11,'11/3/2023','11/6/2023',50.4,100.8);

insert into Usage_History
values(12,'2023-10-3','2023-06-18',39.5,80.9);
insert into Usage_History
values(12,'2023-09-3','2023-06-15',37.8,75.8);
insert into Usage_History
values(12,'11/3/2023','11/6/2023',41.7,90.2);

insert into Usage_History
values(13,'11/3/2023','11/13/2023',33.33,66.66);

insert into Usage_History
values(14,'11/3/2023','11/6/2023',42.27,82.7);

insert into Usage_History
values(15,'2023-10-3','2023-06-18',52.5,102.54);
insert into Usage_History
values(15,'11/3/2023','11/6/2023',37.77,82.54);

insert into Usage_History
values(16,'11/3/2023','11/6/2023',66.77,122.5);

insert into Usage_History
values(17,'2023-10-3','2023-06-18',37.5,60.5);
insert into Usage_History
values(17,'2023-09-3','2023-06-15',35.9,69.9);
insert into Usage_History
values(17,'11/3/2023','11/6/2023',35.7,69.8);

insert into Usage_History
values(18,'11/3/2023','11/08/2023',37.9,77.8);

insert into Usage_History
values(19,'2023-10-3','2023-07-18',41.5,92.4);
insert into Usage_History
values(19,'11/3/2023','11/05/2023',34.9,74.3);
```

Group 5: EMS

```
insert into Usage_History
values(20,'11/3/2023','11/10/2023',44.4,88.8);

insert into Usage_History
values(21,'11/3/2023','11/15/2023',28.6,52.4);

insert into Usage_History
values(22,'11/3/2023','11/18/2023',34.8,74.6);

insert into Usage_History
values(23,'2023-10-3','2023-06-18',39.6,80.3);
insert into Usage_History
values(23,'2023-09-3','2023-06-15',36.7,71.4);
insert into Usage_History
values(23,'11/3/2023','11/06/2023',38.9,72.4);

insert into Usage_History
values(24,'2023-10-3','2023-12-18',47.4,90.3);
insert into Usage_History
values(24,'11/3/2023','11/6/2023',55.9,112.5);

insert into Usage_History
values(25,'2023-10-3','2023-10-18',44.8,90.5);
insert into Usage_History
values(25,'2023-09-3','2023-09-15',51.7,103.2);
insert into Usage_History
values(25,'11/3/2023','11/12/2023',41.7,83.4);

insert into Usage_History
values(26,'2023-10-3','2023-11-18',37.6,72.4);
insert into Usage_History
values(26,'11/3/2023','11/13/2023',33.6,66.7);

insert into Usage_History
values(27,'2023-10-3','2023-06-18',61.2,120.5);
insert into Usage_History
values(27,'2023-09-3','2023-06-15',41.4,83.2);
insert into Usage_History
values(27,'11/3/2023','11/06/2023',39.5,76.8);

insert into Usage_History
values(28,'2023-10-3','2023-08-18',53.4,101.2);
insert into Usage_History
values(28,'11/3/2023','11/12/2023',28.9,60.2);

insert into Usage_History
```

Group 5: EMS

```
values(29,'2023-10-3','2023-09-18',36.7,72.2);
insert into Usage_History
values(29,'11/3/2023','11/15/2023',54.6,102.2);

insert into Usage_History
values(30,'2023-10-3','2023-05-18',45.38,90.5);
insert into Usage_History
values(30,'11/3/2023','11/06/2023',43.34,86.9);
```