UNIVERSITY OF MAURITIUS

Group Assignment by: RUNGIAH Tirouvalen (2110209)

IMRIT Azhar Ali Osamah (2211017)

JHEELAN Pawan (2210207)

VAVAVROOM

ACKNOWLEDGEMENT

We would especially want to thank Mrs. Sudha Cheerkoot-Jalim, our database lecturer, for providing us with this wonderful opportunity to experience what it would be like to be a database designer through this small project. This small project helped us develop as computer science students by covering all the material we learned in both semesters about normalisation, queries, procedures, triggers, and much more. By conducting our own study and pushing ourselves outside our comfort zones, we were also able to discover new things.

STUDENT CONTRIBUTION

Everyone gave the assignment their equal effort. Every stage of the project was completed with full participation from everyone. We used Google Meets to efficiently interact with the other members of our group. To reduce errors, all of the work was completed concurrently and was cross-checked at each stage. We can therefore affirm that each of us has contributed equally to the small project.

Table of Content

VavaVroomVroom Database

Project Description

In Mauritius, the automotive dealership VavaVroomVroom offers both new and used vehicles for sale. Customers can choose from our premium fleet of cars, vans, lorries, trucks, and motorcycles. The business operates a number of showrooms on the island. Each showroom has a unique ID, as well as a location, a phone number, and an email address. There are departments in each showroom, and each department has a department number. A manager with a manager ID is assigned to each department in a showroom. It is also kept track of how many people work in each department. Vehicles with a specific vehicle ID, a type, a make, a model, a year of manufacture, a color, a mileage, an engine capacity, a price (including registration, declaration, and fitness fees), and a discount are shown in showrooms.

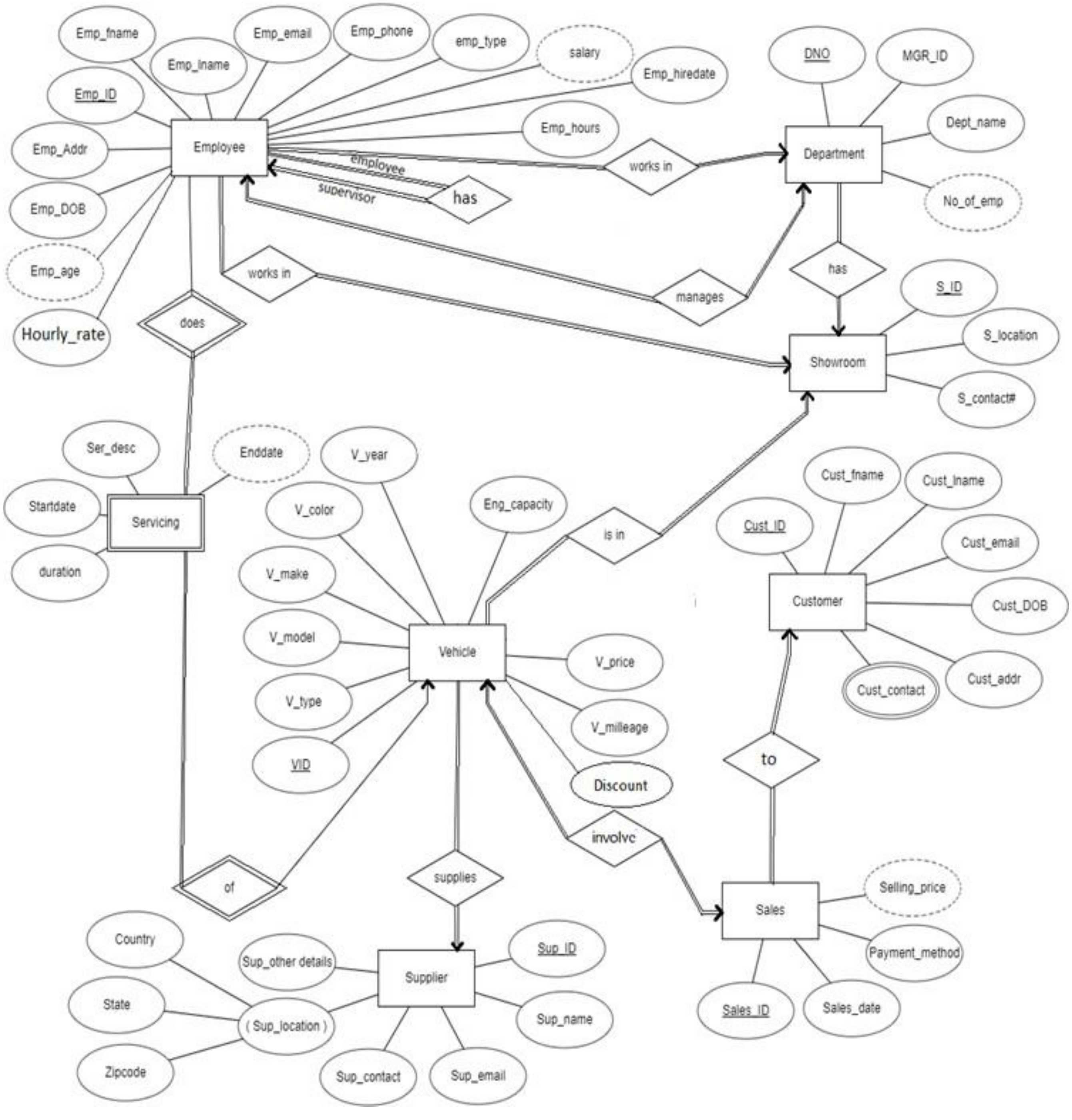
Each vehicle provider has a specific supplier ID, a name, a location (country, state, zip code), and more information. Suppliers can be contacted via email or at their listed phone number. A customer can purchase many vehicles. Every customer has a distinct customer ID that includes their name, address, phone number(s), email, and date of birth.

There are multiple personnel at each showroom. A special ID is connected to each employee. The employee's name, email address, phone number, date of birth, address, number of hours worked, type of employment, and hourly rate. The following formula is used to determine an employee's salary: (hours worked * hourly rate). Each employee is limited to working in one showroom department. Each employee has a supervisor assigned to them. Many employees can be under a supervisor's control.

Any sales that take place are tracked using a special sales ID. One customer and one vehicle are used for each sale. These steps can be used to determine the selling price :(Vehicle price less any discounts).

When necessary, the organization performs servicing on the cars. On a certain day, several employees may work on car maintenance. A description and a timeframe (number of days) are given for the servicing. VavaVroomVroom's owner, Azhar Imrit, has made the decision to create a database management system for the organization's data storage, management, and access.

ERD



1.0 Functional Dependencies:

S_ID ❼ {S_location, S_contact# }

Emp_ID ❼ {Emp_fname, Emp _lname, Emp _email, Emp _phone, Emp _addr, Emp _DOB, Emp_hiredate, Emp_hours, emp_type, Supervisor_ID, DNO,S_ID }

emp_type ❼ {Hourly_rate }

DNO ❼ {Dept_name, MGR_ID}

V_ID ❼ {V_type, V_make, V_model, V_color, V_year, Eng_capacity, V_price, V_mileage, discount, Sup_ID,S_ID}

Sup_ID ❼ {Sup_name, Sup_email, Sup_contact, Country, State, Zipcode, Sup_otherdetails}

V_ID, Emp_ID, Startdate ❼ {Ser_desc, duration}

SalesID ❼ {Sales_date, Payment_method, Cust_ID}

Cust_ID ❼ {Cust_fname, Cust_lname, Cust_email, Cust_DOB, Cust_addr, Cust_contact}

S_ID, SalesID ❼ {V_ID}

1.1 Normalisation

UNF

{Emp_ID, Emp_fname, Emp _lname, Emp _email, Emp _phone, Emp _addr, Emp _DOB, Emp_hiredate, Emp_hours, emp_type, Hourly_rate, S_ID, S_location, S_contact#, DNO, Dept_name, MGR_ID, Supervisor_ID, V_ID, V_ type, V_ make, V_ model, V_ color, V_ year, Eng_capacity, V_ price, V_ mileage, discount, Sup_ID, Sup_name, Sup_email, Sup_contact, Country, State, Zipcode, Sup_otherdetails, Startdate, Ser_desc, duration, SalesID, Sales_date, Payment_method, Cust_ID, Cust_fname, Cust_lname, Cust_email, Cust_DOB, Cust_addr, Cust_contact}

1NF

{ Emp_ID, Emp_fname, Emp _lname, Emp _email, Emp _phone, Emp _addr, Emp _DOB, Emp_hiredate, Emp_hours, emp_type, Hourly_rate, S_ID, S_location, S_contact#, DNO, Dept_name, MGR_ID, Supervisor_ID, V_ID, V_ type, V_ make, V_ model, V_ color, V_ year, Eng_capacity, V_ price, V_ mileage, discount, Sup_ID, Sup_name, Sup_email, Sup_contact, Country, State, Zipcode, Sup_otherdetails, Startdate, Ser_desc, duration, SalesID, Sales_date, Payment_method, Cust_ID, Cust_fname, Cust_lname, Cust_email, Cust_DOB, Cust_addr, Cust_contact }

2NF

{S_ID, S_location, S_contact#, DNO, Dept_name, MGR_ID, Supervisor_ID, Emp_ID, Emp_fname, Emp _lname, Emp _email, Emp _phone, Emp _addr, Emp _DOB, Emp_hiredate, Emp_hours, emp_type, Hourly_rate, V_ID, V_ type, V_ make, V_ model, V_ color, V_ year, Eng_capacity, V_ price, V_ mileage, discount, Sup_ID, Sup_name, Sup_email, Sup_contact, Country, State, Zipcode, Sup_otherdetails, Startdate, Ser_desc, duration }

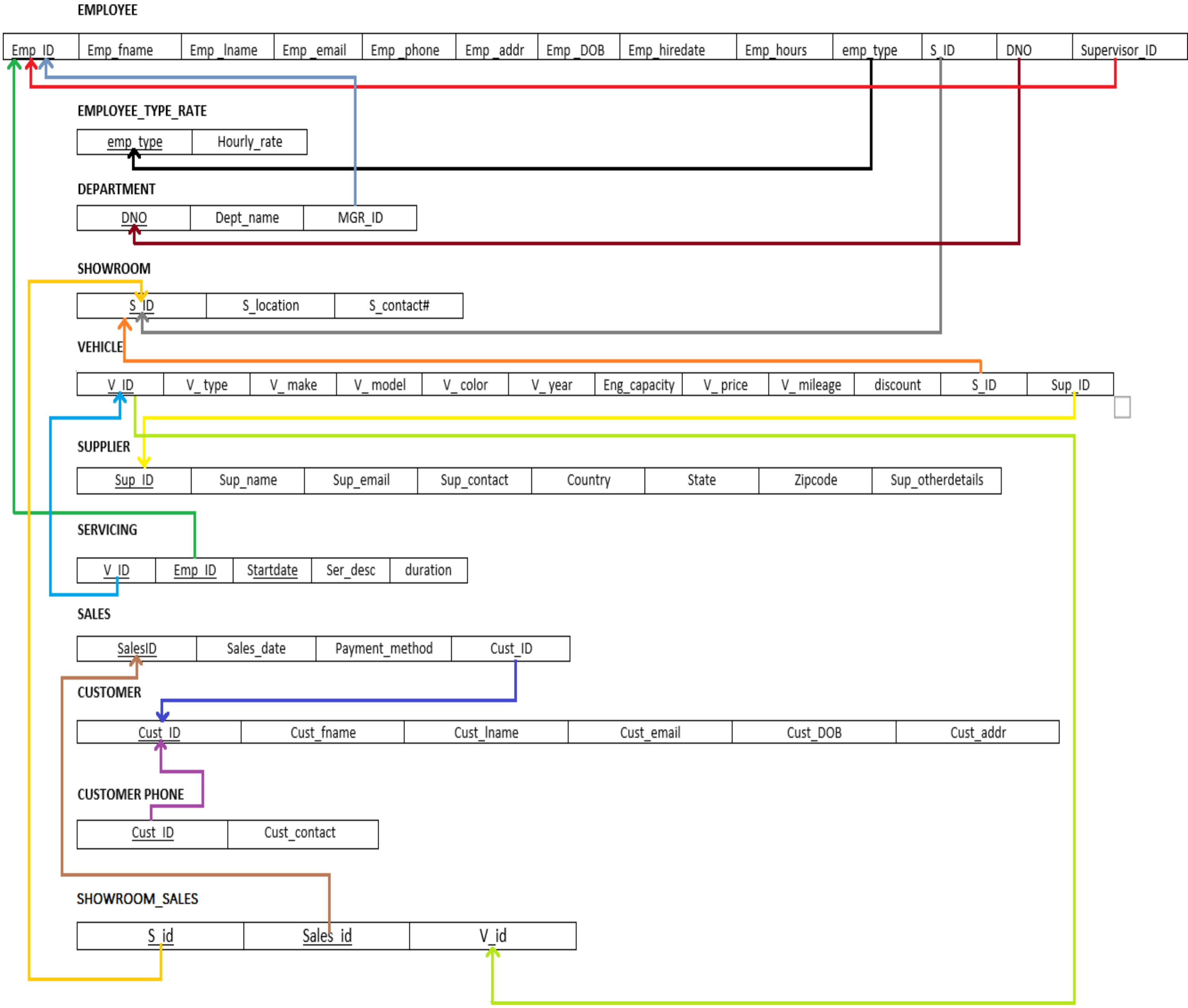
{SalesID, Sales_date, Payment_method, Cust_ID, Cust_fname, Cust_lname, Cust_email, Cust_DOB, Cust_addr, Cust_contact }

{ S_ID, SalesID, V_ID }

3NF

SHOWROOM_SALES	{S_ID, SalesID, V_ID}
SHOWROOM	{S_ID, S_location, S_contact# }
EMPLOYEE	{Emp_ID, Emp_fname, Emp _lname, Emp _email, Emp _phone, Emp _addr, Emp _DOB, Emp_hiredate, Emp_hours, emp_type, S_ID, Supervisor_ID, DNO}
EMPLOYEE_TYPE_RATE	{emp_type, Hourly_rate}
DEPARTMENT	{DNO, Dept_name, MGR_ID }
VEHICLE	{V_ID, V_ type, V_ make, V_ model, V_ color, V_ year, Eng_capacity, V_ price, V_ mileage, discount, Sup_ID,S_ID}
SUPPLIER	{Sup_ID, Sup_name, Sup_email, Sup_contact, Country, State, Zipcode, Sup_otherdetails}
SERVICING	{V_ID, Emp_ID, Startdate, Ser_desc, duration}
SALES	{SalesID, Sales_date, Payment_method, Cust_ID}
CUSTOMER	{Cust_ID, Cust_fname, Cust_lname, Cust_email, Cust_DOB, Cust_addr}
CUSTOMER PHONE	{Cust_ID, Cust_contact}

Schema Diagram



EMPLOYEE

Emp_ID	Emp_fname	Emp_lname	Emp_email	Emp_phone	Emp_addr	Emp_DOB	Emp_hiredate	Emp_hours	emp_type	S_ID	DNO	Supervisor_ID

EMPLOYEE_TYPE_RATE

emp_type	Hourly_rate

DEPARTMENT

DNO	Dept_name	MGR_ID

SHOWROOM

S_ID	S_location	S_contact#

VEHICLE

V_ID	V_type	V_make	V_model	V_color	V_year	Eng_capacity	V_price	V_mileage	discount	S_ID	Sup_ID

SUPPLIER

Sup_ID	Sup_name	Sup_email	Sup_contact	Country	State	Zipcode	Sup_otherdetails

SERVICING

V_ID	Emp_ID	Startdate	Ser_desc	duration

SALES

SalesID	Sales_date	Payment_method	Cust_ID

CUSTOMER

Cust_ID	Cust_fname	Cust_lname	Cust_email	Cust_DOB	Cust_addr

CUSTOMER PHONE

Cust_ID	Cust_contact

SHOWROOM_SALES

S_id	Sales_id	V_id

Diagram 2: Schema Diagram

The schema diagram above shows the relationships between the tables via Primary key-Foreign Key relationships. It gives a brief idea of how the tables will look without any records in them. Arrows go from Foreign Keys to their respective Primary Keys.

1.2 The Functionality of the Fields and Description

The Functionality of the Fields and Description Respective table is shown below:

1.2.1 EMPLOYEE

Attribute	Description	Data Type	Constraint
Emp_ID	Unique ID for the employee (Employee ID)	VARCHAR (5)	• Primary Key • Auto-Increment • Should start with "E" followed by 4 integers
Emp_fname	First name of the employee	VARCHAR (50)	• Not Null
Emp _lname	Last name of the employee	VARCHAR (50)	• Not Null
Emp _email	Email of employee	VARCHAR (30)	
Emp _phone	Phone Number of employees	INTEGER	• Not Null
Emp _addr	Address of the employee	VARCHAR (50)	• Not Null
Emp _DOB	Date of Birth of the employee	DATE	• Date less than current date • NOT NULL
Emp_hiredate	Date on which the employee joined the company	DATE	• Date less than current date • NOT NULL
Emp_hours	Number of hours worked	Float	• Not Null
emp_type	Type of employee	VARCHAR (10)	• Foreign Key referencing to employee_type_rate table
S_ID	Unique ID for the showroom (Showroom ID)	INTEGER	• Foreign Key referencing to Showroom table
DNO	Department Number	INTEGER	• Foreign Key referencing to Department table
Supervisor_ID	ID for supervisor	VARCHAR (5)	• Foreign Key referencing to Employee table

1.2.2 EMPLOYEE_TYPE_RATE

Attribute	Description	Data Type	Constraint
emp_type	Type of employee	VARCHAR (10)	• Primary Key • Possible Values: "Full-time", "Part-time"
Hourly_rate	Amount paid per hour	INTEGER	• Not Null

1.2.3 DEPARTMENT

Attribute	Description	Data Type	Constraint
DNO	Unique Number for the Department (Department Number)	INTEGER	• Primary Key • Not Null
Dept_name	Name of Department	VARCHAR (20)	• Not Null
MGR_ID	ID of Manager	VARCHAR (5)	• Not Null • Foreign Key referencing to Employee table

1.2.4 SHOWROOM

Attribute	Description	Data Type	Constraint
S_ID	Unique Number for the Showroom (Department ID)	INTEGER	• Primary Key • NOT NULL
S_location	Location of Showroom	VARCHAR (50)	• Not Null
S_contact#	Contact Number of Showroom	INTEGER	• Not Null

1.2.5 VEHICLE

Attribute	**Description**	**Data Type**	**Constraint**
V_ID	Unique Number for the Vehicle (Vehicle ID)	VARCHAR (6)	• Primary Key • Should start with "V" followed by 5 integers
V_ type	Type of Vehicle	VARCHAR (20)	• NOT NULL
V_ make	Make of Vehicle	VARCHAR (20)	• NOT NULL
V_ model	Model of Vehicle	VARCHAR (20)	• NOT NULL
V_ color	Color of Vehicle	VARCHAR (20)	• NOT NULL
V_ year	Year of make of Vehicle	INTEGER	• Less or equal to Current Year
Eng_capacity	Engine Capacity	INTEGER	• NOT NULL
V_ price	Price of Vehicle	FLOAT	• NOT NULL
V_ mileage	Mileage of Vehicle	INTEGER	• NOT NULL
discount	Discount on vehicle	FLOAT	
S_ID	Showroom ID	INTEGER	• Foreign Key referencing to Showroom table
Sup_ID	Supplier ID	VARCHAR (7)	• Foreign Key referencing to Supplier table

1.2.6 SUPPLIER

Attribute	**Description**	**Data Type**	**Constraint**
Sup_ID	Unique Number for the Supplier (Supplier ID)	VARCHAR (7)	• Primary Key • Should start with "SUP" followed by 4 integers
Sup_name	Name of Supplier	VARCHAR (30)	• NOT NULL
Sup_email	Email of Supplier	VARCHAR (30)	• Should contain a • "." and a "@"
Sup_contact	Contact of Supplier	VARCHAR (15)	• NOT NULL
Country	Country of Supplier	VARCHAR (30)	• NOT NULL
State	State of Supplier	VARCHAR (30)	• NOT NULL
Zipcode	Zipcode of Supplier	INTEGER	• NOT NULL

Sup_otherdetails	Other Details of Supplier	VARCHAR (50)	

1.2.7 SERVICING

Attribute	Description	Data Type	Constraint
V_ID	Vehicle ID	VARCHAR (6)	• Foreign Key referencing to Vehicle table • Compound key
Emp_ID	Employee ID	VARCHAR (5)	• Foreign Key referencing to Employee table • Compound key
Startdate	Start date of servicing	DATE	• Less than current date • Compound key
Ser_desc	Description of servicing	VARCHAR (50)	
duration	Duration of servicing	INTEGER	• NOT NULL

1.2.8 SALES

Attribute	Description	Data Type	Constraint
SalesID	Unique Number for the Sales (SalesID)	VARCHAR (5)	• Primary Key • Should start with “S” followed by 4 integers
Sales_date	Date of sales	DATE	• NOT NULL
Payment_method	Method of Payment	VARCHAR (20)	• Possible Values:Cash,Cheque,
Cust_ID	Customer ID	VARCHAR (5)	• Foreign Key referencing to Customer table

1.2.9 CUSTOMER

Attribute	Description	Data Type	Constraint
Cust_ID	Unique Number for the Customer (Customer ID)	VARCHAR (5)	• Primary Key, • Should start with “C” followed by 4 integers
Cust_fname	Firstname of Customer	VARCHAR (30)	• NOT NULL
Cust_lname	Lastname of Customer	VARCHAR (30)	• NOT NULL
Cust_email	Email of Customer	VARCHAR (30)	• Should contain a • “.” and a “@”
Cust_DOB	Date of Birth of Customer	DATE	• Less than current date
Cust_addr	Address of Customer	VARCHAR (30)	• NOT NULL

1.2.10 CUSTOMER PHONE

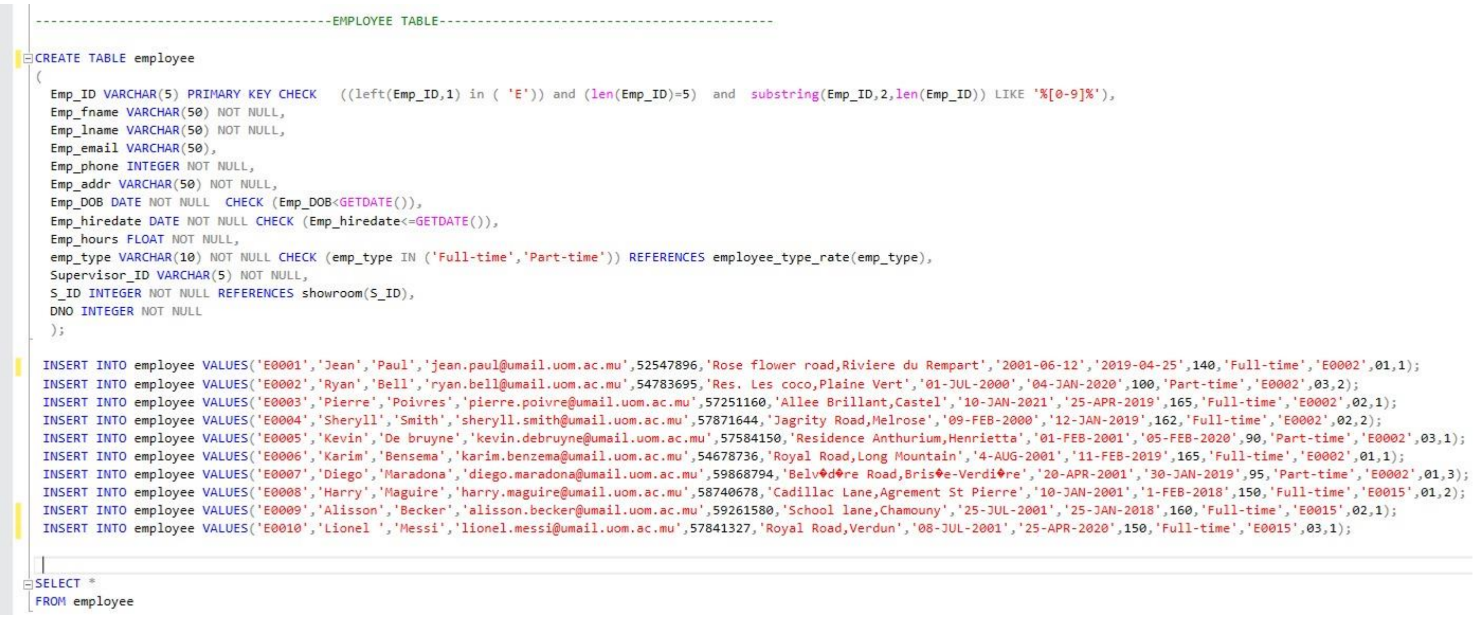
Attribute	Description	Data Type	Constraint
Cust_ID	Customer ID	VARCHAR (5)	• Foreign Key referencing to Customer table
Cust_contact	Phone number of customers	VARCHAR (15)	• NOT NULL, together with customer, they are the composite key.

1.2.11 SHOWROOM SALES

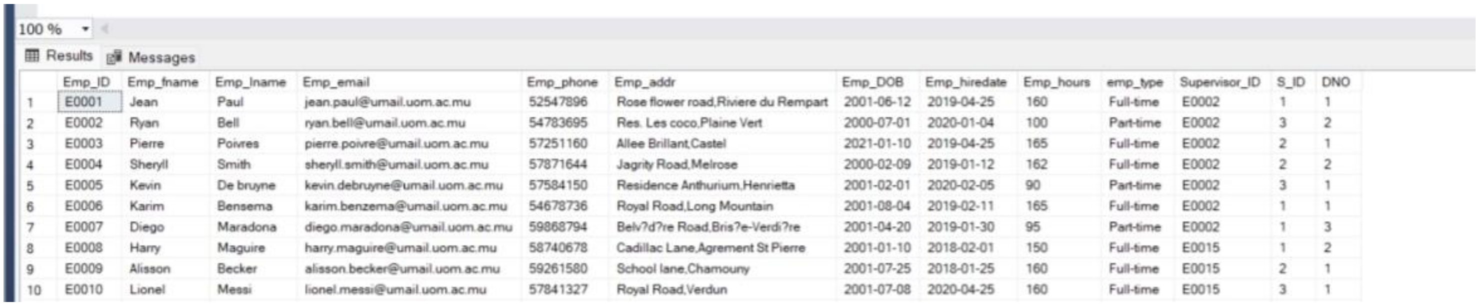
Attribute	Description	Data Type	Constraint
S_ID	Showroom ID	INTEGER	• Foreign Key referencing to Showroom table • Compound key
Sales_ID	Sales_ID	VARCHAR (5)	• Foreign Key referencing to Sales table • Compound key
V_ID	Vehicle ID	VARCHAR (6)	• Foreign Key referencing to Vehicles table

CREATION OF TABLES AND INSERTION OF DATA

Employee Table

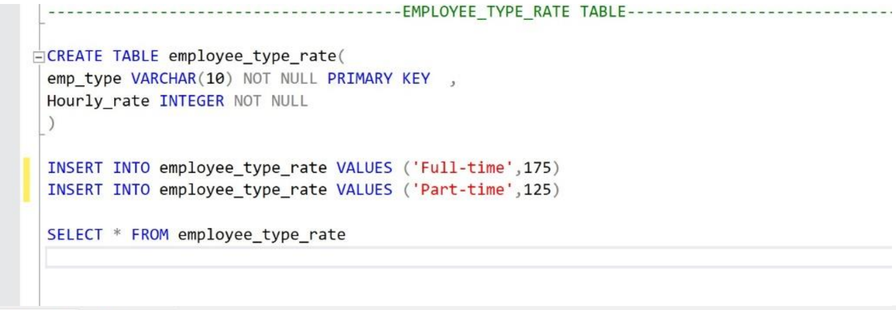


```
--------------------------------------EMPLOYEE TABLE------------------------------------------

CREATE TABLE employee
(
  Emp_ID VARCHAR(5) PRIMARY KEY CHECK   ((left(Emp_ID,1) in ( 'E')) and (len(Emp_ID)=5)  and  substring(Emp_ID,2,len(Emp_ID)) LIKE '%[0-9]%'),
  Emp_fname VARCHAR(50) NOT NULL,
  Emp_lname VARCHAR(50) NOT NULL,
  Emp_email VARCHAR(50),
  Emp_phone INTEGER NOT NULL,
  Emp_addr VARCHAR(50) NOT NULL,
  Emp_DOB DATE NOT NULL  CHECK (Emp_DOB<GETDATE()),
  Emp_hiredate DATE NOT NULL CHECK (Emp_hiredate<=GETDATE()),
  Emp_hours FLOAT NOT NULL,
  emp_type VARCHAR(10) NOT NULL CHECK (emp_type IN ('Full-time','Part-time')) REFERENCES employee_type_rate(emp_type),
  Supervisor_ID VARCHAR(5) NOT NULL,
  S_ID INTEGER NOT NULL REFERENCES showroom(S_ID),
  DNO INTEGER NOT NULL
  );

 INSERT INTO employee VALUES('E0001','Jean','Paul','jean.paul@umail.uom.ac.mu',52547896,'Rose flower road,Riviere du Rempart','2001-06-12','2019-04-25',140,'Full-time','E0002',01,1);
 INSERT INTO employee VALUES('E0002','Ryan','Bell','ryan.bell@umail.uom.ac.mu',54783695,'Res. Les coco,Plaine Vert','01-JUL-2000','04-JAN-2020',100,'Part-time','E0002',03,2);
 INSERT INTO employee VALUES('E0003','Pierre','Poivres','pierre.poivre@umail.uom.ac.mu',57251160,'Allee Brillant,Castel','10-JAN-2021','25-APR-2019',165,'Full-time','E0002',02,1);
 INSERT INTO employee VALUES('E0004','Sheryll','Smith','sheryll.smith@umail.uom.ac.mu',57871644,'Jagrity Road,Melrose','09-FEB-2000','12-JAN-2019',162,'Full-time','E0002',02,2);
 INSERT INTO employee VALUES('E0005','Kevin','De bruyne','kevin.debruyne@umail.uom.ac.mu',57584150,'Residence Anthurium,Henrietta','01-FEB-2001','05-FEB-2020',90,'Part-time','E0002',03,1);
 INSERT INTO employee VALUES('E0006','Karim','Bensema','karim.benzema@umail.uom.ac.mu',54678736,'Royal Road,Long Mountain','4-AUG-2001','11-FEB-2019',165,'Full-time','E0002',01,1);
 INSERT INTO employee VALUES('E0007','Diego','Maradona','diego.maradona@umail.uom.ac.mu',59868794,'Belv�d�re Road,Bris�e-Verdi�re','20-APR-2001','30-JAN-2019',95,'Part-time','E0002',01,3);
 INSERT INTO employee VALUES('E0008','Harry','Maguire','harry.maguire@umail.uom.ac.mu',58740678,'Cadillac Lane,Agrement St Pierre','10-JAN-2001','1-FEB-2018',150,'Full-time','E0015',01,2);
 INSERT INTO employee VALUES('E0009','Alisson','Becker','alisson.becker@umail.uom.ac.mu',59261580,'School lane,Chamouny','25-JUL-2001','25-JAN-2018',160,'Full-time','E0015',02,1);
 INSERT INTO employee VALUES('E0010','Lionel ','Messi','lionel.messi@umail.uom.ac.mu',57841327,'Royal Road,Verdun','08-JUL-2001','25-APR-2020',150,'Full-time','E0015',03,1);

SELECT *
FROM employee
```

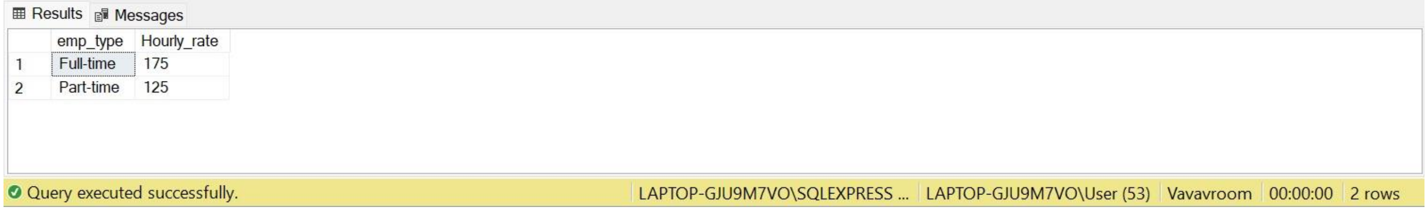


	Emp_ID	Emp_fname	Emp_lname	Emp_email	Emp_phone	Emp_addr	Emp_DOB	Emp_hiredate	Emp_hours	emp_type	Supervisor_ID	S_ID	DNO
1	E0001	Jean	Paul	jean.paul@umail.uom.ac.mu	52547896	Rose flower road,Riviere du Rempart	2001-06-12	2019-04-25	160	Full-time	E0002	1	1
2	E0002	Ryan	Bell	ryan.bell@umail.uom.ac.mu	54783695	Res. Les coco,Plaine Vert	2000-07-01	2020-01-04	100	Part-time	E0002	3	2
3	E0003	Pierre	Poivres	pierre.poivre@umail.uom.ac.mu	57251160	Allee Brillant,Castel	2021-01-10	2019-04-25	165	Full-time	E0002	2	1
4	E0004	Sheryll	Smith	sheryll.smith@umail.uom.ac.mu	57871644	Jagrity Road,Melrose	2000-02-09	2019-01-12	162	Full-time	E0002	2	2
5	E0005	Kevin	De bruyne	kevin.debruyne@umail.uom.ac.mu	57584150	Residence Anthurium,Henrietta	2001-02-01	2020-02-05	90	Part-time	E0002	3	1
6	E0006	Karim	Bensema	karim.benzema@umail.uom.ac.mu	54678736	Royal Road,Long Mountain	2001-08-04	2019-02-11	165	Full-time	E0002	1	1
7	E0007	Diego	Maradona	diego.maradona@umail.uom.ac.mu	59868794	Belv?d?re Road,Bris?e-Verdi?re	2001-04-20	2019-01-30	95	Part-time	E0002	1	3
8	E0008	Harry	Maguire	harry.maguire@umail.uom.ac.mu	58740678	Cadillac Lane,Agrement St Pierre	2001-01-10	2018-02-01	150	Full-time	E0015	1	2
9	E0009	Alisson	Becker	alisson.becker@umail.uom.ac.mu	59261580	School lane,Chamouny	2001-07-25	2018-01-25	160	Full-time	E0015	2	1
10	E0010	Lionel	Messi	lionel.messi@umail.uom.ac.mu	57841327	Royal Road,Verdun	2001-07-08	2020-04-25	160	Full-time	E0015	3	1

Employee Type Rate Table



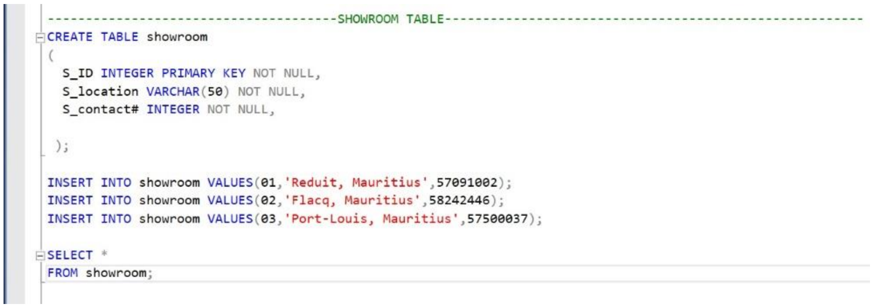
```
---------------------------------------EMPLOYEE_TYPE_RATE TABLE----------------------------

CREATE TABLE employee_type_rate(
emp_type VARCHAR(10) NOT NULL PRIMARY KEY  ,
Hourly_rate INTEGER NOT NULL
)

INSERT INTO employee_type_rate VALUES ('Full-time',175)
INSERT INTO employee_type_rate VALUES ('Part-time',125)

SELECT * FROM employee_type_rate
```



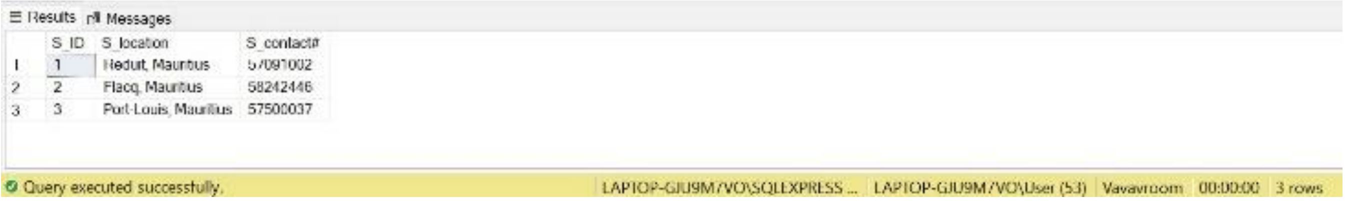
	emp_type	Hourly_rate
1	Full-time	175
2	Part-time	125

Query executed successfully. | LAPTOP-GJU9M7VO\SQLEXPRESS ... | LAPTOP-GJU9M7VO\User (53) | Vavavroom | 00:00:00 | 2 rows

Showroom Table



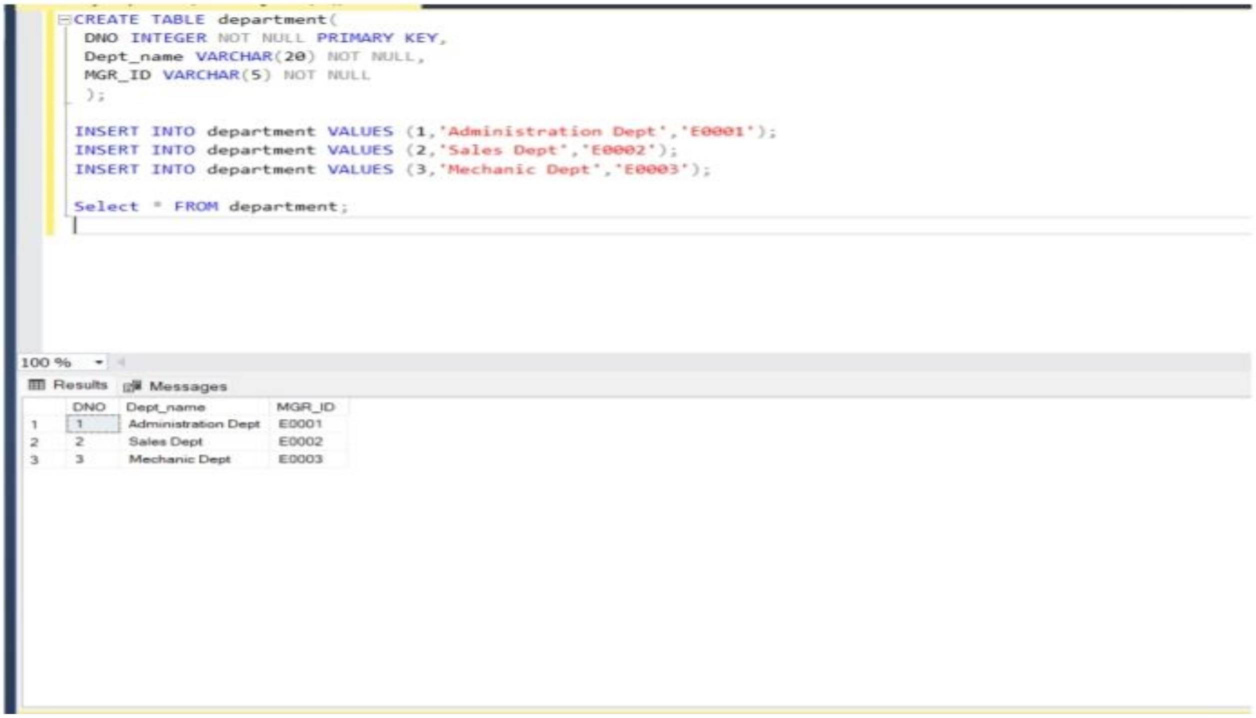
```
-------------------------------------SHOWROOM TABLE-------------------------------------------------------
CREATE TABLE showroom
(
  S_ID INTEGER PRIMARY KEY NOT NULL,
  S_location VARCHAR(50) NOT NULL,
  S_contact# INTEGER NOT NULL,

 );

INSERT INTO showroom VALUES(01,'Reduit, Mauritius',57091002);
INSERT INTO showroom VALUES(02,'Flacq, Mauritius',58242446);
INSERT INTO showroom VALUES(03,'Port-Louis, Mauritius',57500037);

SELECT *
FROM showroom;
```



	S_ID	S_location	S_contact#
1	1	Reduit, Mauritius	57091002
2	2	Flacq, Mauritius	58242446
3	3	Port-Louis, Mauritius	57500037

Query executed successfully. LAPTOP-GJU9M7VO\SQLEXPRESS ... LAPTOP-GJU9M7VO\User (53) Vavavroom 00:00:00 3 rows

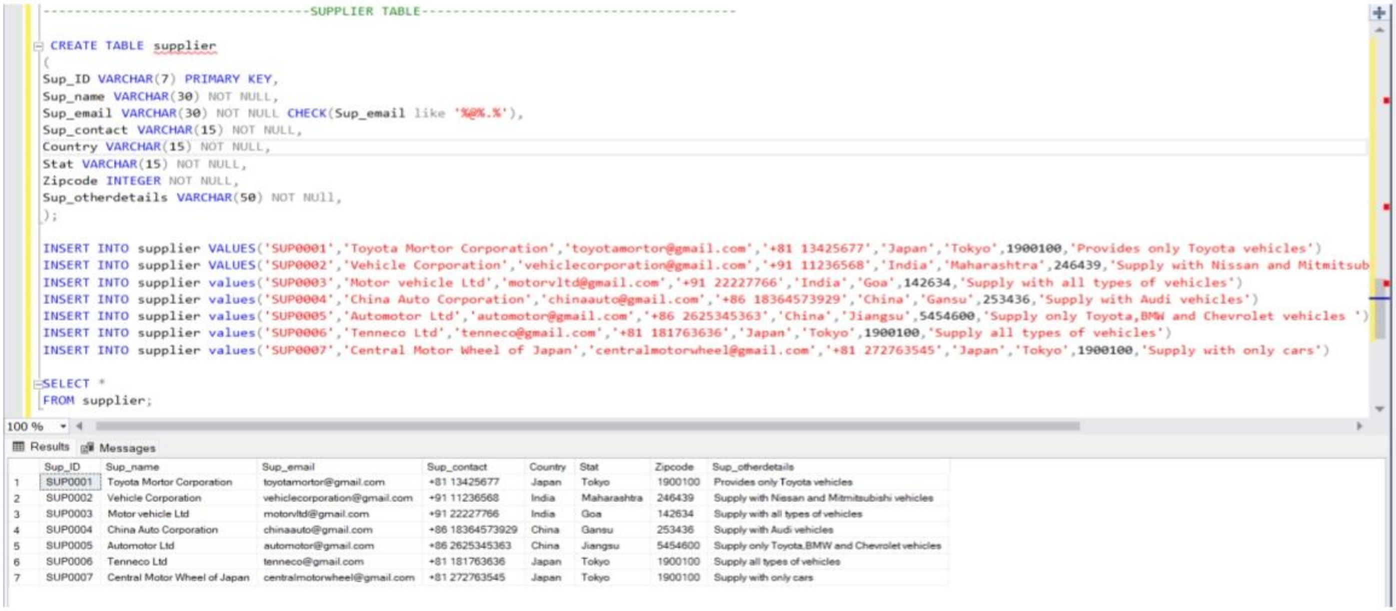
Department Table



```
CREATE TABLE department(
  DNO INTEGER NOT NULL PRIMARY KEY,
  Dept_name VARCHAR(20) NOT NULL,
  MGR_ID VARCHAR(5) NOT NULL
  );

INSERT INTO department VALUES (1,'Administration Dept','E0001');
INSERT INTO department VALUES (2,'Sales Dept','E0002');
INSERT INTO department VALUES (3,'Mechanic Dept','E0003');

Select * FROM department;
```

	DNO	Dept_name	MGR_ID
1	1	Administration Dept	E0001
2	2	Sales Dept	E0002
3	3	Mechanic Dept	E0003

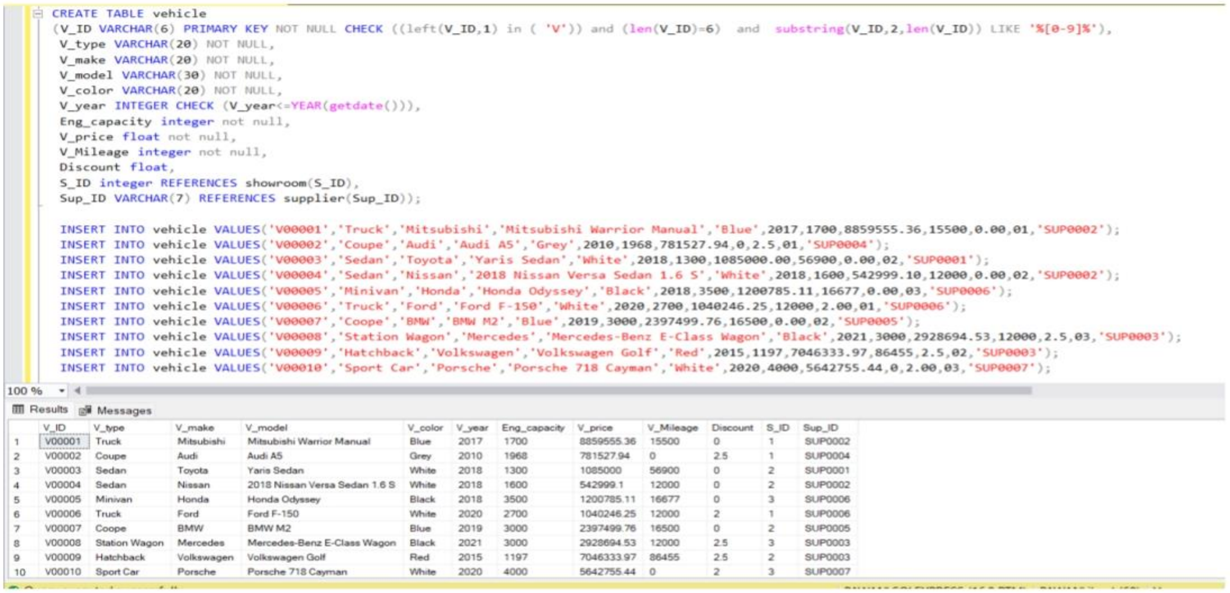
Supplier Table



```
---------------------------------SUPPLIER TABLE---------------------------------------

CREATE TABLE supplier
(
Sup_ID VARCHAR(7) PRIMARY KEY,
Sup_name VARCHAR(30) NOT NULL,
Sup_email VARCHAR(30) NOT NULL CHECK(Sup_email like '%@%.%'),
Sup_contact VARCHAR(15) NOT NULL,
Country VARCHAR(15) NOT NULL,
Stat VARCHAR(15) NOT NULL,
Zipcode INTEGER NOT NULL,
Sup_otherdetails VARCHAR(50) NOT NUll,
);

INSERT INTO supplier VALUES('SUP0001','Toyota Mortor Corporation','toyotamortor@gmail.com','+81 13425677','Japan','Tokyo',1900100,'Provides only Toyota vehicles')
INSERT INTO supplier VALUES('SUP0002','Vehicle Corporation','vehiclecorporation@gmail.com','+91 11236568','India','Maharashtra',246439,'Supply with Nissan and Mitmitsub
INSERT INTO supplier values('SUP0003','Motor vehicle Ltd','motorvltd@gmail.com','+91 22227766','India','Goa',142634,'Supply with all types of vehicles')
INSERT INTO supplier values('SUP0004','China Auto Corporation','chinaauto@gmail.com','+86 18364573929','China','Gansu',253436,'Supply with Audi vehicles')
INSERT INTO supplier values('SUP0005','Automotor Ltd','automotor@gmail.com','+86 2625345363','China','Jiangsu',5454600,'Supply only Toyota,BMW and Chevrolet vehicles ')
INSERT INTO supplier values('SUP0006','Tenneco Ltd','tenneco@gmail.com','+81 181763636','Japan','Tokyo',1900100,'Supply all types of vehicles')
INSERT INTO supplier values('SUP0007','Central Motor Wheel of Japan','centralmotorwheel@gmail.com','+81 272763545','Japan','Tokyo',1900100,'Supply with only cars')

SELECT *
FROM supplier;
```

	Sup_ID	Sup_name	Sup_email	Sup_contact	Country	Stat	Zipcode	Sup_otherdetails
1	SUP0001	Toyota Mortor Corporation	toyotamortor@gmail.com	+81 13425677	Japan	Tokyo	1900100	Provides only Toyota vehicles
2	SUP0002	Vehicle Corporation	vehiclecorporation@gmail.com	+91 11236568	India	Maharashtra	246439	Supply with Nissan and Mitmitsubishi vehicles
3	SUP0003	Motor vehicle Ltd	motorvltd@gmail.com	+91 22227766	India	Goa	142634	Supply with all types of vehicles
4	SUP0004	China Auto Corporation	chinaauto@gmail.com	+86 18364573929	China	Gansu	253436	Supply with Audi vehicles
5	SUP0005	Automotor Ltd	automotor@gmail.com	+86 2625345363	China	Jiangsu	5454600	Supply only Toyota,BMW and Chevrolet vehicles
6	SUP0006	Tenneco Ltd	tenneco@gmail.com	+81 181763636	Japan	Tokyo	1900100	Supply all types of vehicles
7	SUP0007	Central Motor Wheel of Japan	centralmotorwheel@gmail.com	+81 272763545	Japan	Tokyo	1900100	Supply with only cars

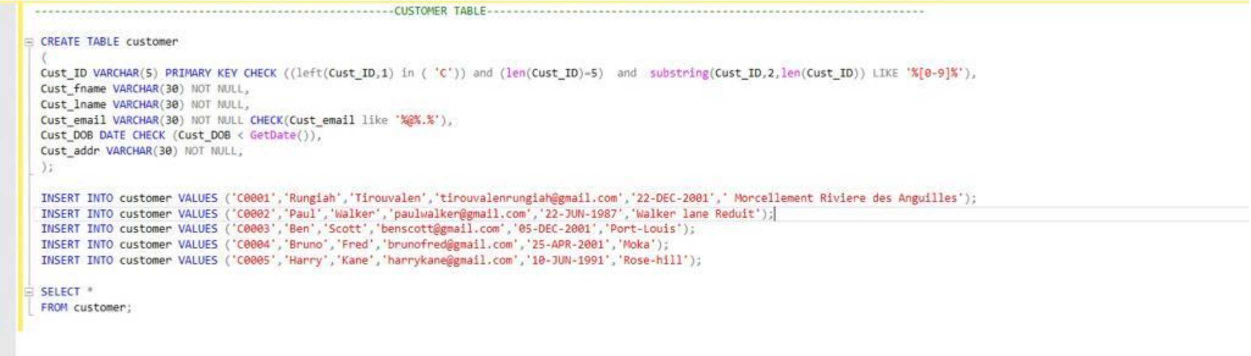
Vehicle Table



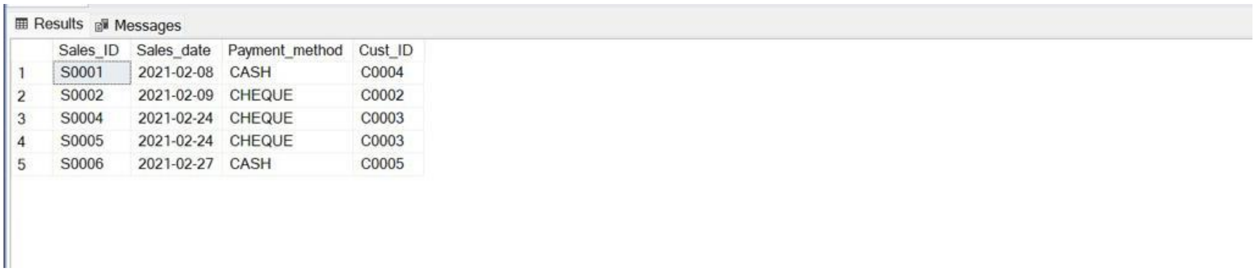
```
CREATE TABLE vehicle
(V_ID VARCHAR(6) PRIMARY KEY NOT NULL CHECK ((left(V_ID,1) in ( 'V')) and (len(V_ID)=6)  and  substring(V_ID,2,len(V_ID)) LIKE '%[0-9]%'),
 V_type VARCHAR(20) NOT NULL,
 V_make VARCHAR(20) NOT NULL,
 V_model VARCHAR(30) NOT NULL,
 V_color VARCHAR(20) NOT NULL,
 V_year INTEGER CHECK (V_year<=YEAR(getdate())),
 Eng_capacity integer not null,
 V_price float not null,
 V_Mileage integer not null,
 Discount float,
 S_ID integer REFERENCES showroom(S_ID),
 Sup_ID VARCHAR(7) REFERENCES supplier(Sup_ID));

 INSERT INTO vehicle VALUES('V00001','Truck','Mitsubishi','Mitsubishi Warrior Manual','Blue',2017,1700,8859555.36,15500,0.00,01,'SUP0002');
 INSERT INTO vehicle VALUES('V00002','Coupe','Audi','Audi A5','Grey',2010,1968,781527.94,0,2.5,01,'SUP0004');
 INSERT INTO vehicle VALUES('V00003','Sedan','Toyota','Yaris Sedan','White',2018,1300,1085000.00,56900,0.00,02,'SUP0001');
 INSERT INTO vehicle VALUES('V00004','Sedan','Nissan','2018 Nissan Versa Sedan 1.6 S','White',2018,1600,542999.10,12000,0.00,02,'SUP0002');
 INSERT INTO vehicle VALUES('V00005','Minivan','Honda','Honda Odyssey','Black',2018,3500,1200785.11,16677,0.00,03,'SUP0006');
 INSERT INTO vehicle VALUES('V00006','Truck','Ford','Ford F-150','White',2020,2700,1040246.25,12000,2.00,01,'SUP0006');
 INSERT INTO vehicle VALUES('V00007','Coope','BMW','BMW M2','Blue',2019,3000,2397499.76,16500,0.00,02,'SUP0005');
 INSERT INTO vehicle VALUES('V00008','Station Wagon','Mercedes','Mercedes-Benz E-Class Wagon','Black',2021,3000,2928694.53,12000,2.5,03,'SUP0003');
 INSERT INTO vehicle VALUES('V00009','Hatchback','Volkswagen','Volkswagen Golf','Red',2015,1197,7046333.97,86455,2.5,02,'SUP0003');
 INSERT INTO vehicle VALUES('V00010','Sport Car','Porsche','Porsche 718 Cayman','White',2020,4000,5642755.44,0,2.00,03,'SUP0007');
```

	V_ID	V_type	V_make	V_model	V_color	V_year	Eng_capacity	V_price	V_Mileage	Discount	S_ID	Sup_ID
1	V00001	Truck	Mitsubishi	Mitsubishi Warrior Manual	Blue	2017	1700	8859555.36	15500	0	1	SUP0002
2	V00002	Coupe	Audi	Audi A5	Grey	2010	1968	781527.94	0	2.5	1	SUP0004
3	V00003	Sedan	Toyota	Yaris Sedan	White	2018	1300	1085000	56900	0	2	SUP0001
4	V00004	Sedan	Nissan	2018 Nissan Versa Sedan 1.6 S	White	2018	1600	542999.1	12000	0	2	SUP0002
5	V00005	Minivan	Honda	Honda Odyssey	Black	2018	3500	1200785.11	16677	0	3	SUP0006
6	V00006	Truck	Ford	Ford F-150	White	2020	2700	1040246.25	12000	2	1	SUP0006
7	V00007	Coope	BMW	BMW M2	Blue	2019	3000	2397499.76	16500	0	2	SUP0005
8	V00008	Station Wagon	Mercedes	Mercedes-Benz E-Class Wagon	Black	2021	3000	2928694.53	12000	2.5	3	SUP0003
9	V00009	Hatchback	Volkswagen	Volkswagen Golf	Red	2015	1197	7046333.97	86455	2.5	2	SUP0003
10	V00010	Sport Car	Porsche	Porsche 718 Cayman	White	2020	4000	5642755.44	0	2	3	SUP0007

Customer Table

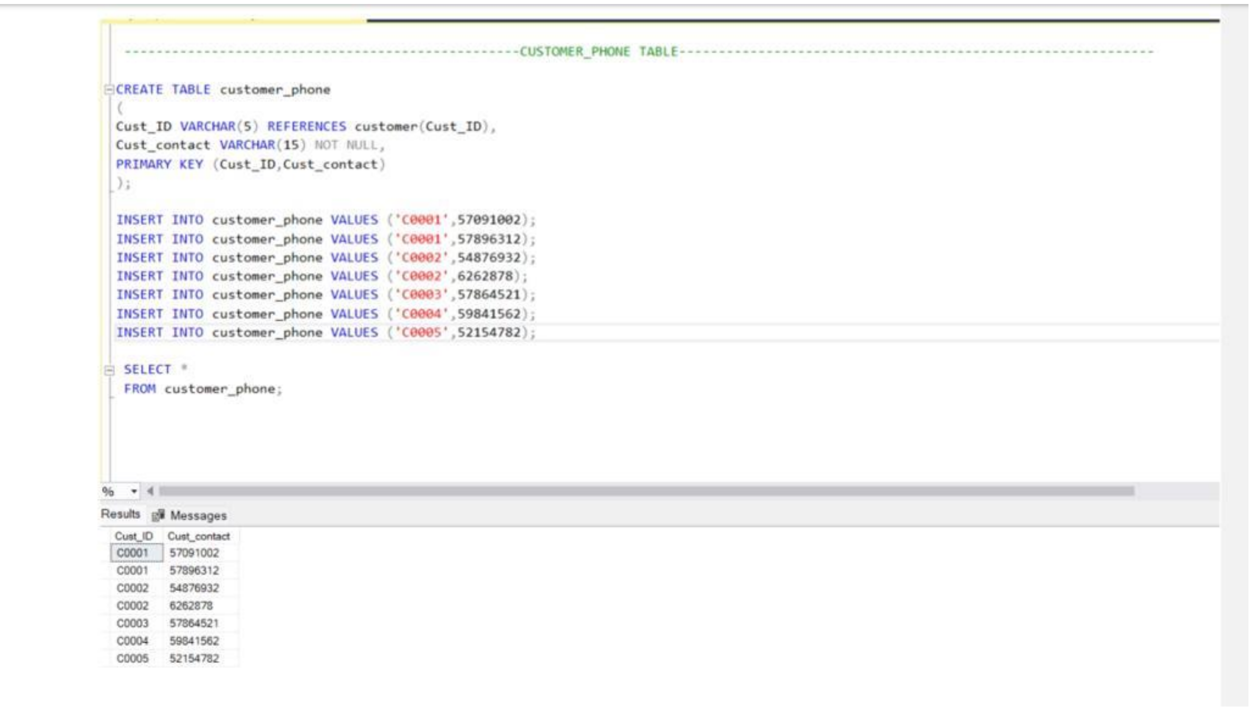


```
-----------------------------------------------------CUSTOMER TABLE-------------------------------------------------------------------

CREATE TABLE customer
(
Cust_ID VARCHAR(5) PRIMARY KEY CHECK ((left(Cust_ID,1) in ( 'C')) and (len(Cust_ID)=5)  and  substring(Cust_ID,2,len(Cust_ID)) LIKE '%[0-9]%'),
Cust_fname VARCHAR(30) NOT NULL,
Cust_lname VARCHAR(30) NOT NULL,
Cust_email VARCHAR(30) NOT NULL CHECK(Cust_email like '%@%.%'),
Cust_DOB DATE CHECK (Cust_DOB < GetDate()),
Cust_addr VARCHAR(30) NOT NULL,
);

INSERT INTO customer VALUES ('C0001','Rungiah','Tirouvalen','tirouvalenrungiah@gmail.com','22-DEC-2001',' Morcellement Riviere des Anguilles');
INSERT INTO customer VALUES ('C0002','Paul','Walker','paulwalker@gmail.com','22-JUN-1987','Walker lane Reduit');
INSERT INTO customer VALUES ('C0003','Ben','Scott','benscott@gmail.com','05-DEC-2001','Port-Louis');
INSERT INTO customer VALUES ('C0004','Bruno','Fred','brunofred@gmail.com','25-APR-2001','Moka');
INSERT INTO customer VALUES ('C0005','Harry','Kane','harrykane@gmail.com','10-JUN-1991','Rose-hill');

SELECT *
FROM customer;
```



	Sales_ID	Sales_date	Payment_method	Cust_ID
1	S0001	2021-02-08	CASH	C0004
2	S0002	2021-02-09	CHEQUE	C0002
3	S0004	2021-02-24	CHEQUE	C0003
4	S0005	2021-02-24	CHEQUE	C0003
5	S0006	2021-02-27	CASH	C0005

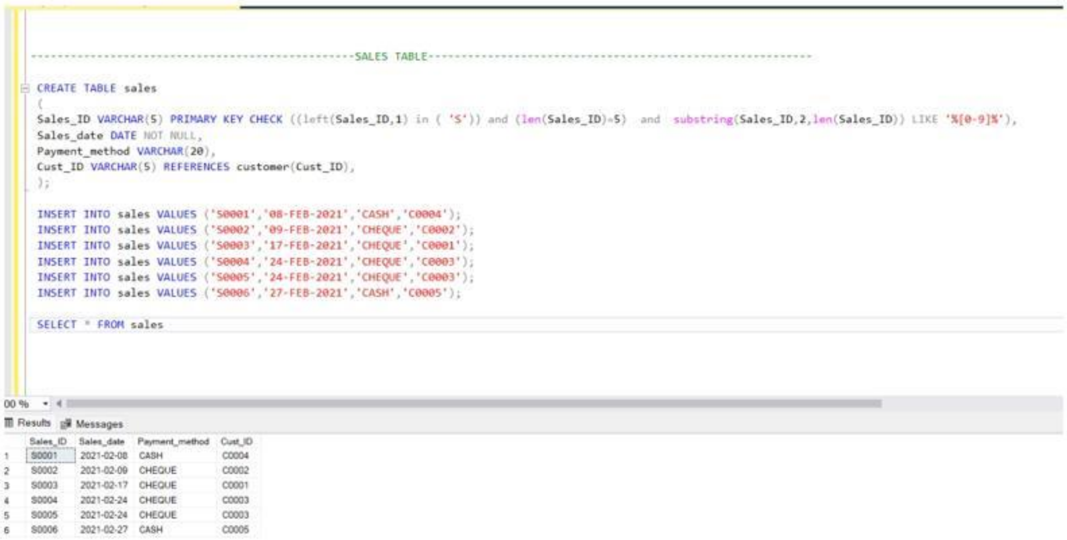
Customer Phone Table



```
---------------------------------------------------CUSTOMER_PHONE TABLE----------------------------------------------------------

CREATE TABLE customer_phone
(
Cust_ID VARCHAR(5) REFERENCES customer(Cust_ID),
Cust_contact VARCHAR(15) NOT NULL,
PRIMARY KEY (Cust_ID,Cust_contact)
);

INSERT INTO customer_phone VALUES ('C0001',57091002);
INSERT INTO customer_phone VALUES ('C0001',57896312);
INSERT INTO customer_phone VALUES ('C0002',54876932);
INSERT INTO customer_phone VALUES ('C0002',6262878);
INSERT INTO customer_phone VALUES ('C0003',57864521);
INSERT INTO customer_phone VALUES ('C0004',59841562);
INSERT INTO customer_phone VALUES ('C0005',52154782);

 SELECT *
 FROM customer_phone;
```

Cust_ID	Cust_contact
C0001	57091002
C0001	57896312
C0002	54876932
C0002	6262878
C0003	57864521
C0004	59841562
C0005	52154782

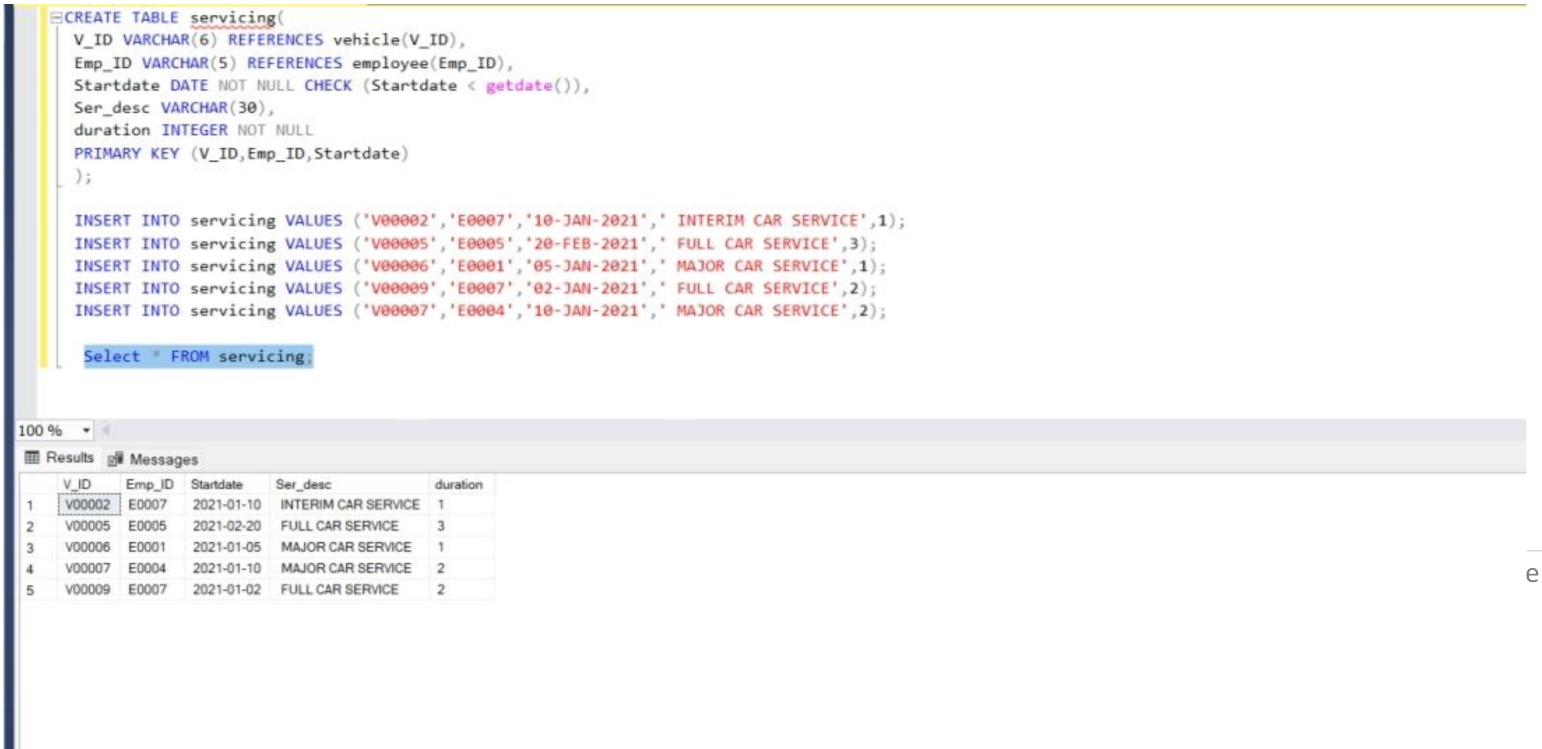
Sales Table



```
-------------------------------------------------SALES TABLE---------------------------------------------------------

CREATE TABLE sales
(
Sales_ID VARCHAR(5) PRIMARY KEY CHECK ((left(Sales_ID,1) in ( 'S')) and (len(Sales_ID)=5)  and  substring(Sales_ID,2,len(Sales_ID)) LIKE '%[0-9]%'),
Sales_date DATE NOT NULL,
Payment_method VARCHAR(20),
Cust_ID VARCHAR(5) REFERENCES customer(Cust_ID),
);

INSERT INTO sales VALUES ('S0001','08-FEB-2021','CASH','C0004');
INSERT INTO sales VALUES ('S0002','09-FEB-2021','CHEQUE','C0002');
INSERT INTO sales VALUES ('S0003','17-FEB-2021','CHEQUE','C0001');
INSERT INTO sales VALUES ('S0004','24-FEB-2021','CHEQUE','C0003');
INSERT INTO sales VALUES ('S0005','24-FEB-2021','CHEQUE','C0003');
INSERT INTO sales VALUES ('S0006','27-FEB-2021','CASH','C0005');

SELECT * FROM sales
```

```
   Sales_ID  Sales_date  Payment_method  Cust_ID
1  S0001     2021-02-08  CASH            C0004
2  S0002     2021-02-09  CHEQUE          C0002
3  S0003     2021-02-17  CHEQUE          C0001
4  S0004     2021-02-24  CHEQUE          C0003
5  S0005     2021-02-24  CHEQUE          C0003
6  S0006     2021-02-27  CASH            C0005
```

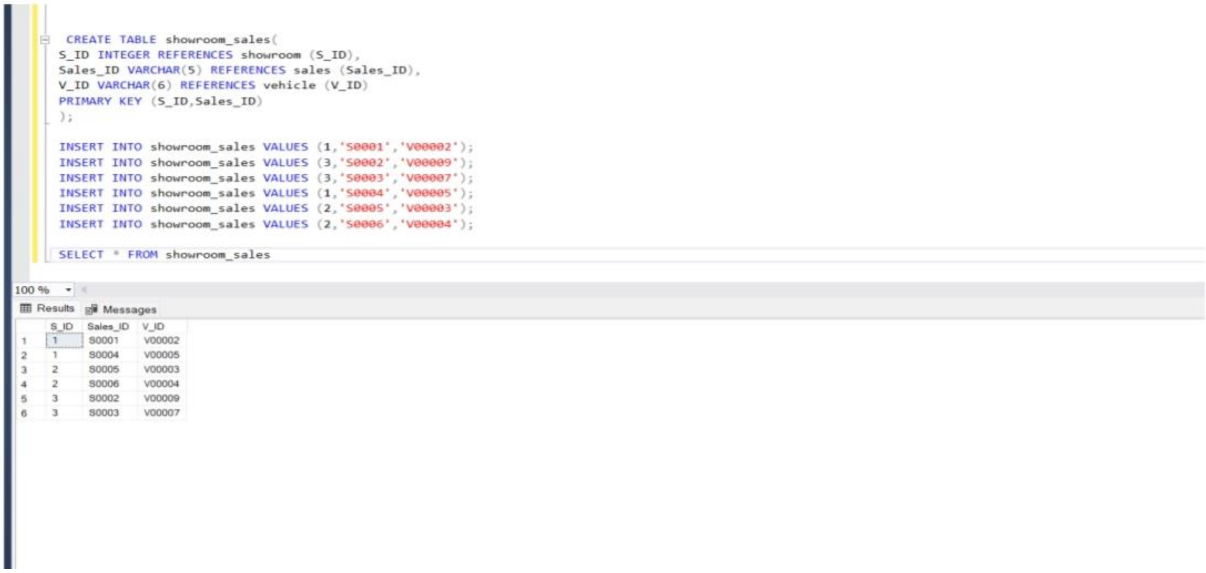
Servicing Table



```
CREATE TABLE servicing(
  V_ID VARCHAR(6) REFERENCES vehicle(V_ID),
  Emp_ID VARCHAR(5) REFERENCES employee(Emp_ID),
  Startdate DATE NOT NULL CHECK (Startdate < getdate()),
  Ser_desc VARCHAR(30),
  duration INTEGER NOT NULL
  PRIMARY KEY (V_ID,Emp_ID,Startdate)
  );

  INSERT INTO servicing VALUES ('V00002','E0007','10-JAN-2021',' INTERIM CAR SERVICE',1);
  INSERT INTO servicing VALUES ('V00005','E0005','20-FEB-2021',' FULL CAR SERVICE',3);
  INSERT INTO servicing VALUES ('V00006','E0001','05-JAN-2021',' MAJOR CAR SERVICE',1);
  INSERT INTO servicing VALUES ('V00009','E0007','02-JAN-2021',' FULL CAR SERVICE',2);
  INSERT INTO servicing VALUES ('V00007','E0004','10-JAN-2021',' MAJOR CAR SERVICE',2);

   Select * FROM servicing;
```

```
   V_ID    Emp_ID  Startdate   Ser_desc              duration
1  V00002  E0007   2021-01-10  INTERIM CAR SERVICE   1
2  V00005  E0005   2021-02-20  FULL CAR SERVICE      3
3  V00006  E0001   2021-01-05  MAJOR CAR SERVICE     1
4  V00007  E0004   2021-01-10  MAJOR CAR SERVICE     2
5  V00009  E0007   2021-01-02  FULL CAR SERVICE      2
```

Showroom Sales Table



```
CREATE TABLE showroom_sales(
S_ID INTEGER REFERENCES showroom (S_ID),
Sales_ID VARCHAR(5) REFERENCES sales (Sales_ID),
V_ID VARCHAR(6) REFERENCES vehicle (V_ID)
PRIMARY KEY (S_ID,Sales_ID)
);

INSERT INTO showroom_sales VALUES (1,'S0001','V00002');
INSERT INTO showroom_sales VALUES (3,'S0002','V00009');
INSERT INTO showroom_sales VALUES (3,'S0003','V00007');
INSERT INTO showroom_sales VALUES (1,'S0004','V00005');
INSERT INTO showroom_sales VALUES (2,'S0005','V00003');
INSERT INTO showroom_sales VALUES (2,'S0006','V00004');

SELECT * FROM showroom_sales
```

```
   S_ID  Sales_ID  V_ID
1  1     S0001     V00002
2  1     S0004     V00005
3  2     S0005     V00003
4  2     S0006     V00004
5  3     S0002     V00009
6  3     S0003     V00007
```

Stored Procedures and Their Functionalities

Stored Procedure's name	**Functionality**
sp_ins_emp	• **Stored procedure to insert values into the employee Table** • **Takes 12 parameters** (@Emp_fname,@Emp_lname,@Emp_email,@Emp_phone,@Emp_addr, @Emp_DOB,@Emp_hiredate,@Emp_hours,@emp_type,@Supervisor_ID ,@S_ID,@DNO) • **Emp_ID is not included because it is auto-incremented** (The procedure separates the alphabet 'E' and numerical part of Emp_ID and increments the numerical part and then concatenates it back to form the new Emp_ID) • **Catches any errors during the insert and displays which error occured**

sp_ins_cust	• **Stored procedure to insert values into** 1) **customer Table** 2) **customer_phone Table** • **Takes 7 parameters** (@Cust_fname, @Cust_lname, @Cust_email, @Cust_DOB, @Cust_addr, @Cust_contact1, @Cust_contact2) • **Cust_ID is not included because it is auto-incremented** (The procedure separates the alphabet ‘C’ and numerical part of Cust_ID and increments the numerical part and then concatenates it back to form the new Cust_ID) • Automatically inserts @Cust_contact1 if in correct format

	• Only Inserts @Cust_contact2 if it is not input as a NULL value • **Catches any errors during the insert and displays which error occured**
sp_ins_vehicle	• **Stored procedure to insert values into the vehicle Table** • **Takes 11 parameters** (@V_type, @V_make, @V_model, @V_color, @V_year, @Eng_capacity, @V_price, @V_mileage, @discount, @S_ID, @Sup_ID) • **V_ID is not included because it is auto-incremented** (The procedure separates the alphabet 'V' and numerical part of V_ID and increments the numerical part and then concatenates it back to form the new V_ID) • **Catches any errors during the insert and displays which error occured**
sp_ins_sup	• **Stored procedure to insert values into the supplier Table** • **Takes 3 parameters** (@Sup_name, @Sup_email, @Sup_contact, @Country, @Stat, @Zipcode, @Sup_otherdetails) • **Sales_ID is not included because it is auto-incremented** (The procedure separates the alphabets 'SUP' and numerical part of Sup_ID and increments the numerical part and then concatenates it back to form the new Sup_ID) • **Catches any errors during the insert and displays which error occurred**
sp_Insert_Sales	• **Stored procedure to insert values into** **1) sales Table**

	2) sales_showroom Table • **Takes 4 parameters** (@Sales_date, @Payment_method, @Cust_ID, @V_ID) • **Sales_ID is not included because it is auto-incremented** (The procedure separates the alphabets 'S' and numerical part of Sales_ID and increments the numerical part and then concatenates it back to form the new Sales_ID) • **S_ID is not included because it is retrieved from the vehicle Table by using the @V_ID** • **Catches any errors during the insert and displays which error occurred**
sp_Insert _Servicing	• **Stored procedure to insert values into the servicing Table** • **Takes 3 parameters** (@V_ID, @Emp_ID, @Ser_desc, @duration) • **Start date is not included because it is retrieved automatically** The procedure uses the in-built function GETDATE () to store startdate • **Catches any errors during the insert and displays which error occurred**
sp_display_all_salaries	• Stored Procedure to calculate and display all salaries of employees along with their IDs.
sp_display_employee_salary	• **Stored Procedure to calculate and display salary of an employee along with his/her name.** • **Takes 1 parameter** (@emp_id)

	• **The salary of an employee is calculated using the rate per hour and the number of hours worked by the employee.** (@salary=CAST((@hour * CAST(@hourly_rate AS FLOAT)) AS INTEGER); • **Then the salary is converted to string datatype to be able to print.** (@STRINGSALARY=CAST (@salary AS VARCHAR (10))
sp_display_supplied_vehicle	• Stored Procedure to display all vehicles which have been supplied by a particular supplier • **Takes 1 parameter** (@supplier_id)
sp_display_vehicleprice	• **Stored Procedure to check if a vehicle has a discount, then displays the vehicle ID, the original price, the discount (if any) and the new price** • **Takes 1 parameter** (@vehicle_id) • **The discount value is the calculated by using the discount rate and the original price.** (@discount_value=((SELECT Discount FROM vehicle WHERE @vehicle_id=V_ID) / 100) * @original_price) • **Then the new price is calculated using the discount value and the original price.** (@new_price= @original_price - @discount_value) • **The discount value, original price and the new price is then converted to string datatype to be able to print.** (@STRING_discountvalue=CAST (@discount_value AS VARCHAR (15)) @STRING_originalvalue=CAST (@original_price AS VARCHAR (15)) • @STRING_newprice=CAST (@new_price AS VARCHAR (15)))
sp_emp_age	• **Stored Procedure to display age of an employee along with his/her name and ID.** • **Takes 1 parameter** (@emp_id)

	• **Employee age is then calculated by using the current year and the date of birth of the employee.** (@age= year (getdate ()) - year(@emp_dob) • **The age is then converted to string datatype to be able to print.** **(@**STRING_age = CAST (@age AS VARCHAR (3))**)**
sp_emp_service_period	• **Stored Procedure to display period of service of an employee in years, along with his/her name and ID** • **Takes 1 parameter** (@emp_id) • **Service period is then calculated by using current year and the hire date of the employee** (year(getdate()) - year(@hire_date)
sp_sales_emp_commission	• **Stored Procedure to increase salary of all employees by 2.5% in sales department if more than 5 cars are sold in a showroom in the last month** *The year and month of salesdates in the sales table are compared to the year and month of the inbuilt GETDATE() function. *The number of cars sold for each showroom in the last month is counted *The salaries of employees who work in the showrooms with count>=5 are increased by 2.5%

Triggers and Their Functionalities

Trigger	Functionality
tg_checkdata_emp	• Reinforces constraint for Firstname, Lastname, Date of Birth, phone number, address, employee type, Supervisor ID, Department Number. • If the fields Firstname, Lastname, Date of Birth, address, Supervisor ID, Department Number are left NULL, it returns an error message. • It also returns an error message if the Date of Birth is after the current date. • A built-in function GET DATE () is used. • If all of the above are working correctly, the data is inserted accordingly and a message is printed if insert is successful.
tg_checkdata_customer	• Reinforces constraint for Firstname, Lastname, Email, address and Date of Birth. • If Firstname, Lastname, Email, address is left NULL, an error message is printed. • If DOB is above current date an error message is printed. • A built-in function GET DATE () is used. • If all of the above are working correctly, the data is inserted accordingly and a message is printed if insert is successful.

tg_checkdata_vehicles	• Reinforces constraint for Type, Make, Model, color, year, engine capacity, price and the other fields. • It returns appropriate message in case of errors. • If the fields such as Type, Make, Model, color, year, engine capacity are left null, an error message is printed. • If the price of the vehicle is 0, it does not accept such input. • The vehicle should be in a showroom which exists, otherwise there is an error message.

tg_checkdata_supplier	• Reinforces constraint for the supplier’s name, email, contact number, country, state, zip code. • Supplier ID is auto incremented in the insert procedure for supplier. • It returns appropriate message in case of errors. • If the name, email, contact number, country, state, zip code is NULL, there is an error message. • Email should be of a particular format.
tg_checkdata_customer	• Reinforces constraint for customer ID and his/her contact numbers. • It returns appropriate message in case of errors. • The customer ID which will be input should exist in the customer table. It therefore checks for this constraint and in case of error, a warning message is output. • The first contact should be filled whereas the second can be left NULL.
tg_checkdata_servicing	• Reinforces constraint vehicle ID, employee ID, date of the servicing, description as well as the duration. • It returns appropriate message in case of errors. • The user cannot input a vehicle ID which does not exist. • The user cannot also input an employee ID which does not exist. • Moreover, the description and duration fields should not be left blank.
tg_checkdata_sales	• Reinforces constraint for Sales ID, Sales Date, Payment method, Customer ID. • Sales ID is auto incremented in its procedure. • It returns appropriate message in case of errors. • Sales Date and Payment should not be left NULL. • The customer ID should be a valid customer (which exists in our table customer).

	• In December, there is an EOY discount. An appropriate message is printed to show this information. • If all of the above are working correctly, the data is inserted accordingly and a message is printed if insert is successful.
tg_checkdata_showroomsales	• Reinforces constraint for showroom ID, sales ID, vehicle ID. • The showroom and vehicle ID input should exist in their respective tables: showroom, vehicle. • For each sale, the sales ID should be recorded in the showroom_sales table. • IF the above conditions are not met, it returns appropriate message in case of errors.

THE END