

INFO 5707 PROJECT PROPOSAL

DRIVING SCHOOL APPOINTMENT SCHEDULING SYSTEM

TEAM MEMBERS:

1. Manasa Cherukupally, Team coordinator (11604712)
2. Keerthi Boggiti (11560357)
3. Harshith Kumar Reddy Kothur (11549819)

OVERVIEW:

Driving School is a place where they teach how to drive. People willing to get a driving license usually take Driving lessons in the Driving School. To get the driving Lessons, the user should first place the request to the driving school. Admin of the Driving school will check the slots and based on the slot availability user will get the reservation to get driving lessons. Also, the user will be assigned an available instructor who will teach the person to drive.

SCOPE:

The driving school reservation system is a model that helps people make reservations for the slots in driving school. This system helps the driving school register the applicants, take requests from the applicants, schedule the driving lessons, allocate the available instructors, and select the car model as requested. The system should also allow the applicant to cancel the reservation before the scheduled date.

SPECIFIC USER REQUIREMENTS:

The system will be able to provide applicants to make a request for driving school. Applicants can make reservations for the driving lessons. The system will allocate the available instructor. Applicants can also select car models for their lessons. They will be able to cancel the reservation before their scheduled date.

TECHNICAL REQUIREMENTS:

For creating table and executing the SQL queries we will be using **MYSQL** database.

For designing the ER-Diagram as part of Design phase, we will be using **Draw.io**.

For creating the report we used **Microsoft Word**

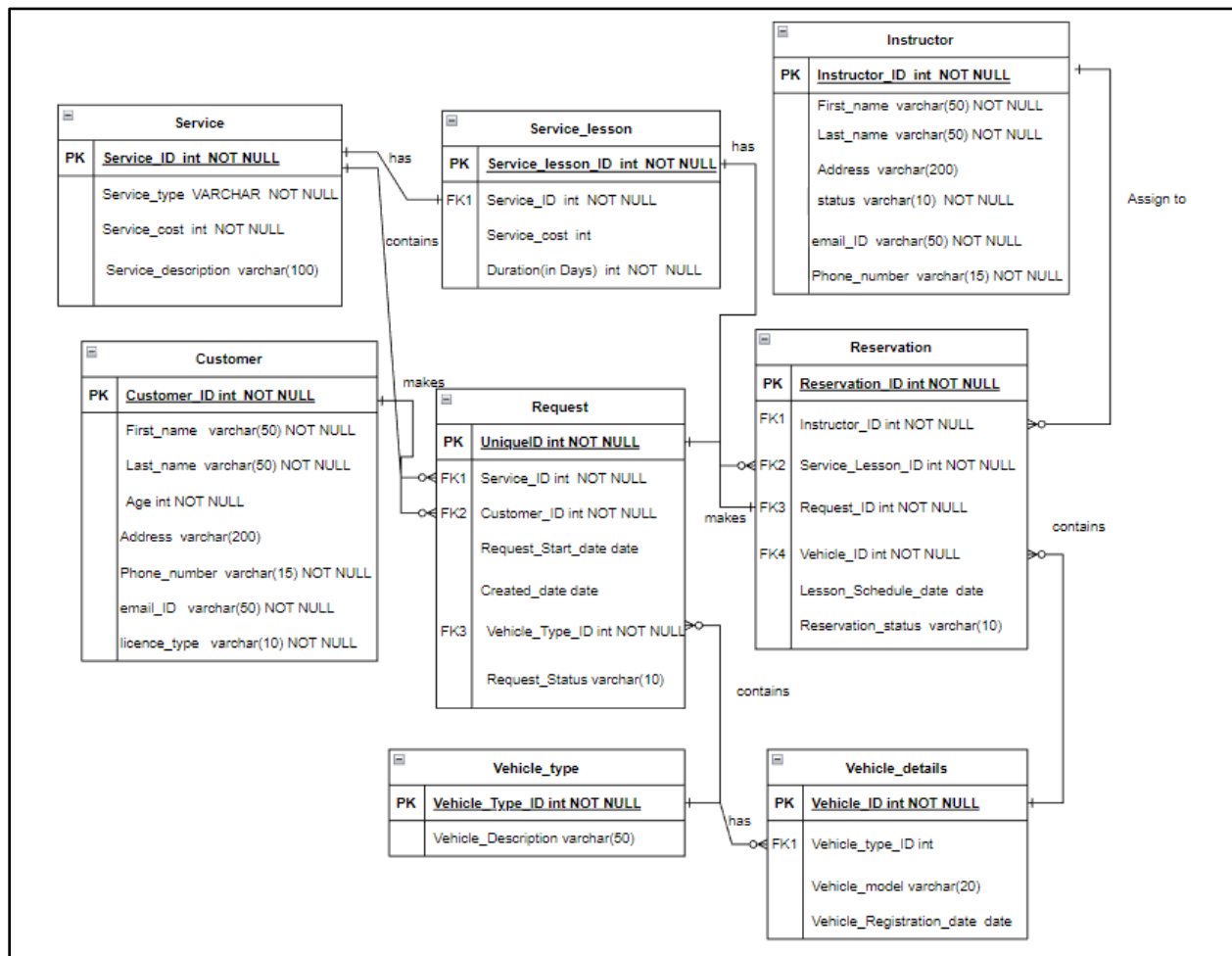
DESIGN PHASE

ER-DIAGRAM:

As part of the project design phase we have taken eight entities. To start with first and main entity is the “Customer” who will access the system to make reservations. We have “Request” entity to keep the request for specific driving lesson. There is “Reservation entity which is related with Request entity. Since there is option to choose from different vehicle models, we have “Vehicle type” and “Vehicle details” entity. There is “Instructor” entity so that available instructor is allocated. We also have “Service” entity and “Service lesson” entity for offering different services.

“Customer table” has one to many relationships with “Request table”. Request table has one to one relation with “Reservation table”. “Reservation table” has one to many relation with “Instructor table” . “Service” and “vehicle type ” tables have one to many relation with “Request table”. Reservation table has many to one relation with “Vehicle details” table.

ER-Diagram of Driving School Appointment Scheduling System:



Business Rules:

1. One customer can make many requests.
2. One request makes only one reservation.
3. Each vehicle type can be in many requests.
4. One vehicle type can have many vehicles.
5. One service lesson can have many reservations.
6. Each instructor is assigned with one or many reservations.
7. One service makes one service lesson
8. Each vehicle can be used in many reservations.

Data Dictionary:

Table Name	Attribute Name	Contents	Type	Format	Range	Required	P K or F K	FK referenced Table
<i>Service</i>	Service_ID	Service number	INT	99999	00001-99999	Y	PK	
	Service_type	Type of Service	VARCHAR			Y		
	Service_cost	Service's Cost	INT	99999	00001-99999	Y		
	Service_description	Brief of Service	VARCHAR (100)	XXXXXX				
<i>Customer</i>	Customer_ID	Customer number	INT	99999	00001-99999	Y	PK	
	First_name	Customer's first name	VARCHAR (50)	XXXXXX		Y		
	Last_name	Customer's first name	VARCHAR (50)	XXXXXX		Y		
	Age	Customer's Age	INT	999	001 - 999	Y		
	Address	Customer's location details	VARCHAR (200)	XXXXXX				
	Phone_number	Customer's Contact no.	VARCHAR (15)	XXXXXX		Y		

	email_ID	Customer's email	VARCHAR (50)	XXXXXX		Y		
	licence_type	Customer's type of licence	VARCHAR (10)	XXXXXX		Y		
Instructor	Instructor_ID	Instructor number	INT	99999	00001-99999	Y	PK	
	First_name	Instructor's first name	VARCHAR (50)	XXXXXX		Y		
	Last_name	Instructor's last name	VARCHAR (50)	XXXXXX		Y		
	Address	Instructor's location details	VARCHAR (200)	XXXXXX				
	status	Instructor's status	VARCHAR (10)	Active (or) inactive		Y		
	email_ID	Instructor's email	VARCHAR (50)	XXXXXX		Y		
	Phone_number	Instructor's contact no.	VARCHAR (15)	XXXXXX		Y		
Reservation	Reservation_ID	Reservation number	INT	99999	00001-99999	Y	PK	
	Instructor_ID	Instructor number	INT	99999	00001-99999	Y	FK	Instructor
	Service_Lesson_ID	Service Lesson number	INT	99999	00001-99999	Y	FK	Service_lesson
	Request_ID	Request number	INT	99999	00001-99999	Y	FK	Request
	Vehicle_ID	Vehicle number	INT	99999	00001-99999	Y	FK	Vehicle_details
	Lesson_Schedule_date	Lesson Schedule Date	DATE	YYYY-MM-DD				
	Reservation_status	Reservation's Status	VARCHAR (50)	XXXXXX				
Vehicle_details	Vehicle_ID	Vehicle number	INT	99999	00001-99999	Y	PK	
	Vehicle_type_ID	Vehicle type number	INT	99999	00001-99999		FK	Vehicle_type
	Vehicle_model	Model of the Vehicle	VARCHAR (20)	XXXXXX				
	Vehicle_Registration_date	Registration date of Vehicle	DATE	YYYY-MM-DD				

Vehicle_type	Vehicle_Type_ID	Vehicle type number	INT	99999	00001-99999	Y	PK	
	Vehicle_Description	Vehicle's brief	VARCHAR (50)	XXXXXX				
Service_lesson	Service_lesson_ID	Service lesson number	INT	99999	00001 - 99999	Y	PK	
	Service_ID	Service number	INT	99999	00001 - 99999	Y	FK	Service
	Service_cost	Cost of Service	INT	99999	00001 - 99999			
	Duration (in Days)	Duration of Service lesson	INT	999	001 - 999	Y		
Request	UniqueID	Request number	INT	99999	00001-99999	Y	PK	
	Service_ID	Service number	INT	99999	00001-99999	Y	FK	Service
	Customer_ID	Customer number	INT	99999	00001-99999	Y	FK	Customer
	Request_Start_date	Request's Start date	DATE	YYYY-MM-DD				
	Created_date	Created Date of Request	DATE	YYYY-MM-DD				
	Vehicle_Type_ID	Vehicle Type number	INT	99999	00001 - 99999	Y	FK	Vehicle_type
	Request_Status	Request status	VARCHAR (10),			Y		

Implementation Phase

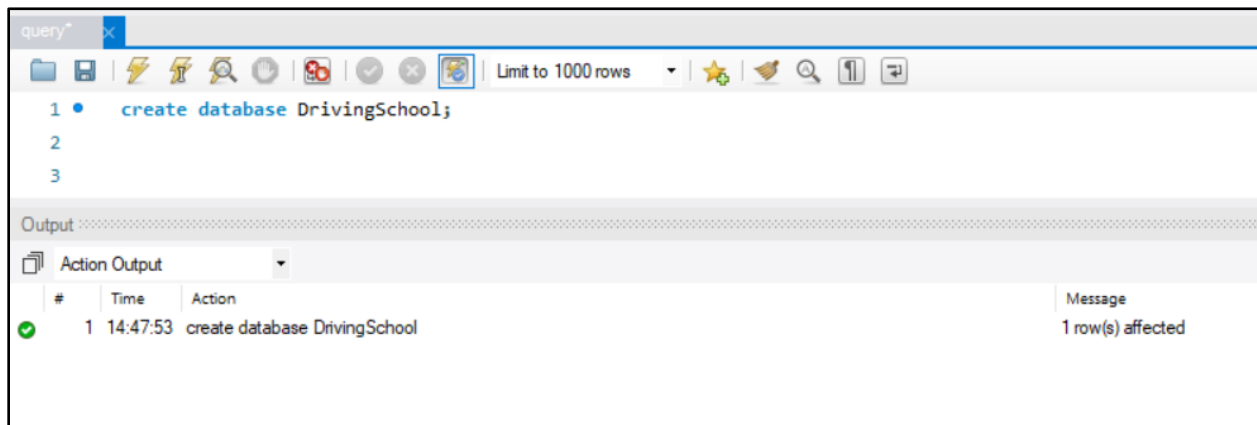
Data Entry and Update:

Database Creation:

SQL Query to create database:

Create database DrivingSchool;

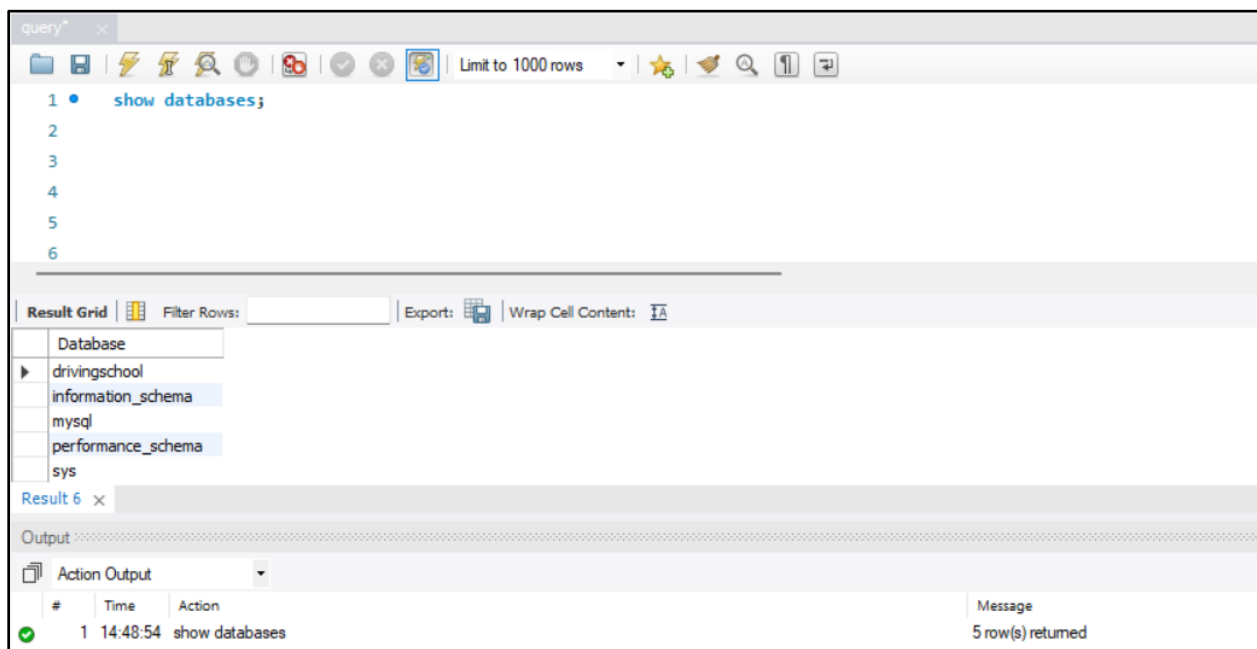
Output:



SQL Query to view databases:

Show databases;

Output:

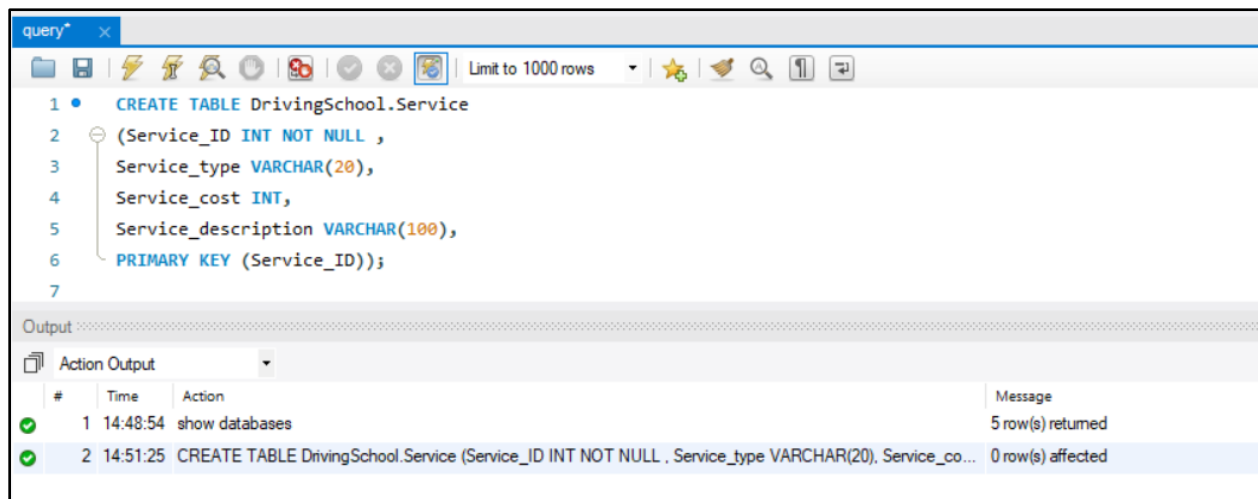


Tables Creation:

Creating Service table:

```
CREATE TABLE DrivingSchool.Service  
(Service_ID INT NOT NULL ,  
Service_type VARCHAR(20),  
Service_cost INT,  
Service_description VARCHAR(100),  
PRIMARY KEY (Service_ID));
```

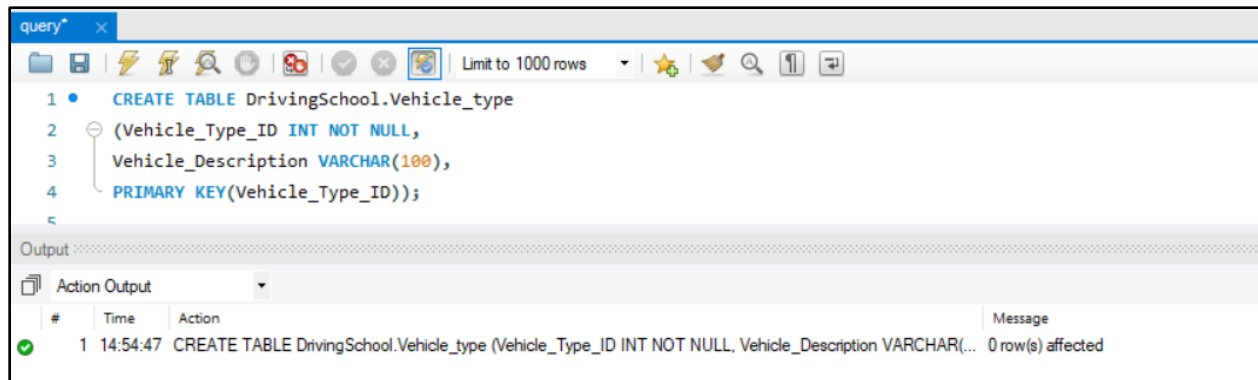
Output:



Creating Vehicle_Type table:

```
CREATE TABLE DrivingSchool.Vehicle_type  
(Vehicle_Type_ID INT NOT NULL,  
Vehicle_Description VARCHAR(100),  
PRIMARY KEY(Vehicle_Type_ID));
```

Output:



The screenshot shows a SQL query editor window titled 'query'. The query text is as follows:

```
1 • CREATE TABLE DrivingSchool.Vehicle_type
2 (Vehicle_Type_ID INT NOT NULL,
3   Vehicle_Description VARCHAR(100),
4   PRIMARY KEY(Vehicle_Type_ID));
```

Below the query editor, the 'Output' tab is selected, showing the 'Action Output' table. The table has four columns: '#', 'Time', 'Action', and 'Message'. A single row is displayed, indicating successful execution:

#	Time	Action	Message
1	14:54:47	CREATE TABLE DrivingSchool.Vehicle_type (Vehicle_Type_ID INT NOT NULL, Vehicle_Description VARCHAR(100), PRIMARY KEY(Vehicle_Type_ID));	0 row(s) affected

Creating Customer table:

```
CREATE TABLE DrivingSchool.Customer
```

```
(Customer_ID INT NOT NULL,
```

```
First_name VARCHAR(50),
```

```
Last_name VARCHAR(50),
```

```
Age INT,
```

```
Address VARCHAR(200),
```

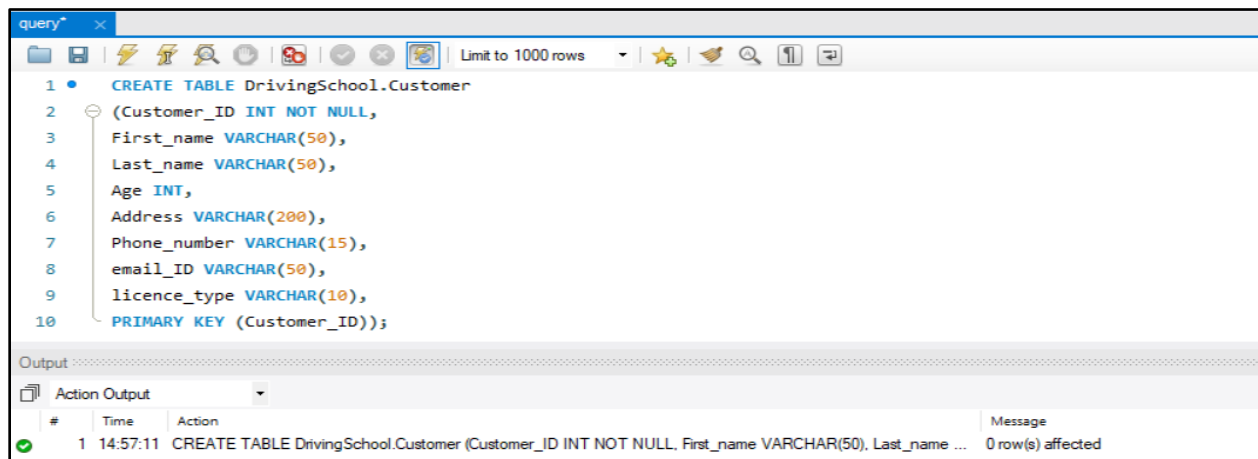
```
Phone_number VARCHAR(15),
```

```
email_ID VARCHAR(50),
```

```
licence_type VARCHAR(10),
```

```
PRIMARY KEY (Customer_ID));
```

Output:



The screenshot shows a SQL query editor window titled 'query'. The query text is as follows:

```
1 • CREATE TABLE DrivingSchool.Customer
2 (Customer_ID INT NOT NULL,
3   First_name VARCHAR(50),
4   Last_name VARCHAR(50),
5   Age INT,
6   Address VARCHAR(200),
7   Phone_number VARCHAR(15),
8   email_ID VARCHAR(50),
9   licence_type VARCHAR(10),
10  PRIMARY KEY (Customer_ID));
```

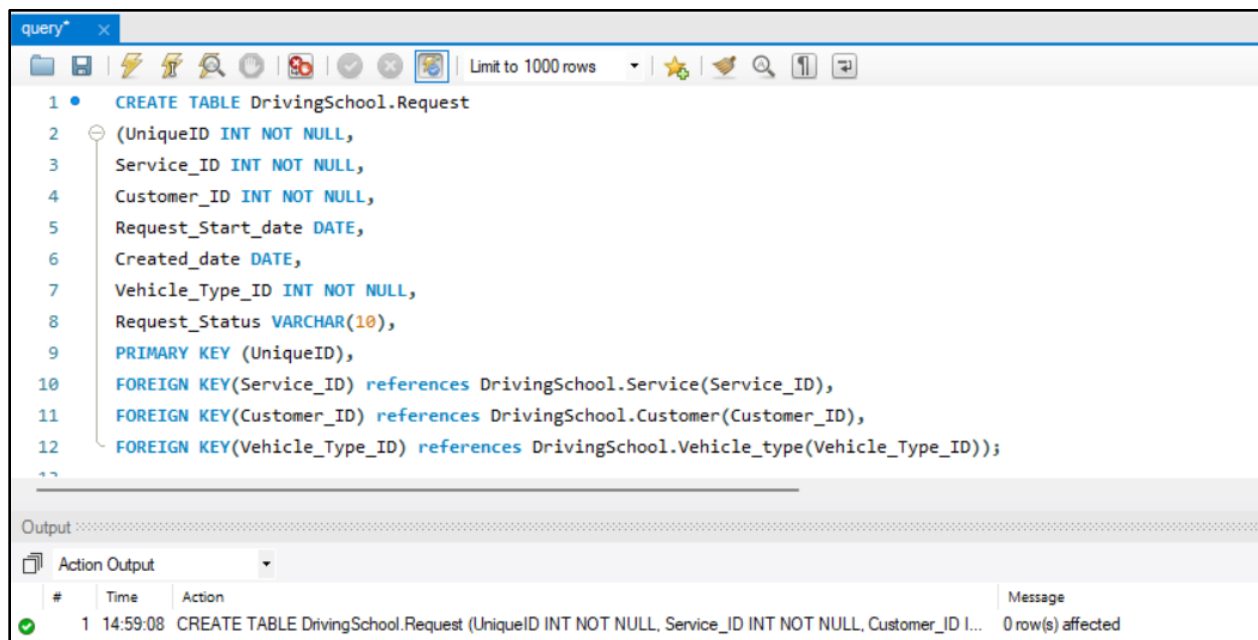
Below the query editor, the 'Output' tab is selected, showing the 'Action Output' table. The table has four columns: '#', 'Time', 'Action', and 'Message'. A single row is displayed, indicating successful execution:

#	Time	Action	Message
1	14:57:11	CREATE TABLE DrivingSchool.Customer (Customer_ID INT NOT NULL, First_name VARCHAR(50), Last_name VARCHAR(50), Age INT, Address VARCHAR(200), Phone_number VARCHAR(15), email_ID VARCHAR(50), licence_type VARCHAR(10), PRIMARY KEY (Customer_ID));	0 row(s) affected

Creating Request table:

```
CREATE TABLE DrivingSchool.Request  
(UniqueID INT NOT NULL,  
Service_ID INT NOT NULL,  
Customer_ID INT NOT NULL,  
Request_Start_date DATE,  
Created_date DATE,  
Vehicle_Type_ID INT NOT NULL,  
Request_Status VARCHAR(10),  
PRIMARY KEY (UniqueID),  
FOREIGN KEY(Service_ID) references DrivingSchool.Service(Service_ID),  
FOREIGN KEY(Customer_ID) references DrivingSchool.Customer(Customer_ID),  
FOREIGN KEY(Vehicle_Type_ID) references DrivingSchool.Vehicle_type(Vehicle_Type_ID));
```

Output:



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

```
1 • CREATE TABLE DrivingSchool.Request  
2 (UniqueID INT NOT NULL,  
3 Service_ID INT NOT NULL,  
4 Customer_ID INT NOT NULL,  
5 Request_Start_date DATE,  
6 Created_date DATE,  
7 Vehicle_Type_ID INT NOT NULL,  
8 Request_Status VARCHAR(10),  
9 PRIMARY KEY (UniqueID),  
10 FOREIGN KEY(Service_ID) references DrivingSchool.Service(Service_ID),  
11 FOREIGN KEY(Customer_ID) references DrivingSchool.Customer(Customer_ID),  
12 FOREIGN KEY(Vehicle_Type_ID) references DrivingSchool.Vehicle_type(Vehicle_Type_ID));  
13
```

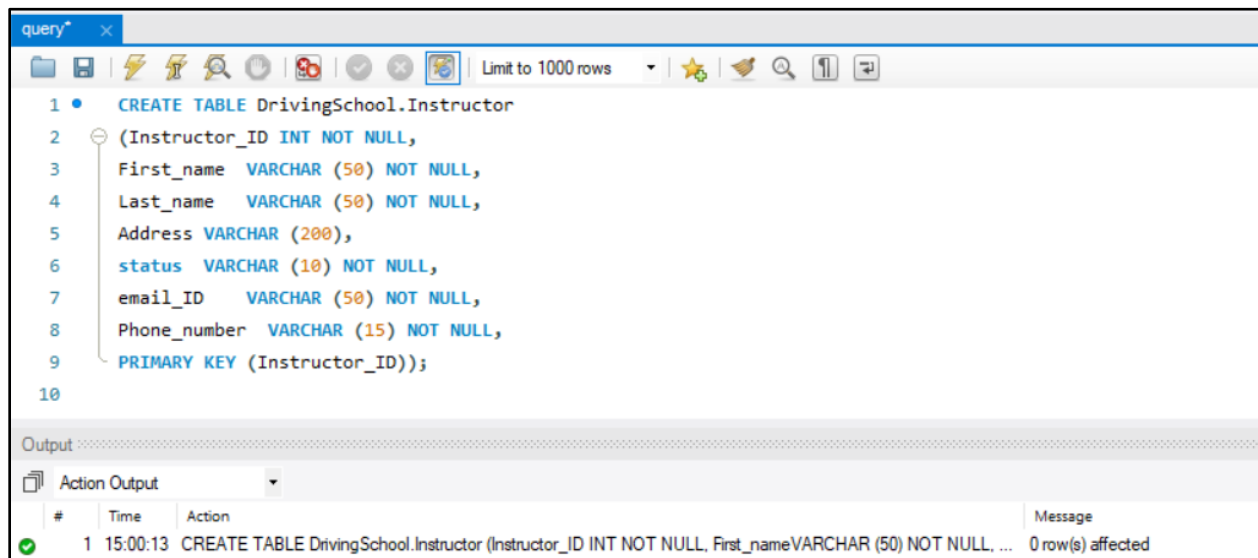
Below the query editor is an "Output" section with a tab labeled "Action Output". It contains a table with the following data:

#	Time	Action	Message
1	14:59:08	CREATE TABLE DrivingSchool.Request (UniqueID INT NOT NULL, Service_ID INT NOT NULL, Customer_ID I...	0 row(s) affected

Creating Instructor table:

```
CREATE TABLE DrivingSchool.Instructor
(Instructor_ID INT NOT NULL,
First_name  VARCHAR (50) NOT NULL,
Last_name   VARCHAR (50) NOT NULL,
Address     VARCHAR (200),
status      VARCHAR (20) NOT NULL,
email_ID    VARCHAR (50) NOT NULL,
Phone_number VARCHAR (15) NOT NULL,
PRIMARY KEY (Instructor_ID));
```

Output:



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

```
1 • CREATE TABLE DrivingSchool.Instructor
2 (Instructor_ID INT NOT NULL,
3   First_name  VARCHAR (50) NOT NULL,
4   Last_name   VARCHAR (50) NOT NULL,
5   Address     VARCHAR (200),
6   status      VARCHAR (10) NOT NULL,
7   email_ID    VARCHAR (50) NOT NULL,
8   Phone_number VARCHAR (15) NOT NULL,
9   PRIMARY KEY (Instructor_ID));
10
```

Below the query editor, the 'Output' tab is selected, showing a table with columns '#', 'Time', 'Action', and 'Message'. The first row indicates successful execution:

#	Time	Action	Message
1	15:00:13	CREATE TABLE DrivingSchool.Instructor (Instructor_ID INT NOT NULL, First_name VARCHAR (50) NOT NULL, ...	0 row(s) affected

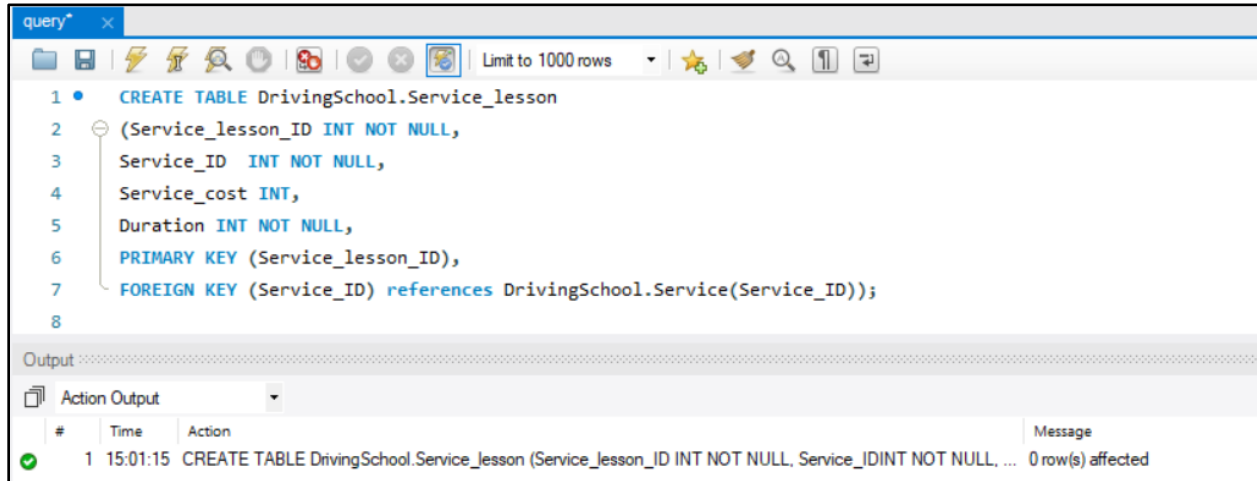
Creating Service_Lesson table:

```
CREATE TABLE DrivingSchool.Service_lesson
(Service_lesson_ID INT NOT NULL,
Service_ID  INT NOT NULL,
Service_cost INT,
Duration INT NOT NULL,
```

PRIMARY KEY (Service_lesson_ID),

FOREIGN KEY (Service_ID) references DrivingSchool.Service(Service_ID));

Output:



The screenshot shows a SQL query editor window titled "query*". The query is as follows:

```
1 • CREATE TABLE DrivingSchool.Service_lesson
2   (Service_lesson_ID INT NOT NULL,
3     Service_ID INT NOT NULL,
4     Service_cost INT,
5     Duration INT NOT NULL,
6     PRIMARY KEY (Service_lesson_ID),
7     FOREIGN KEY (Service_ID) references DrivingSchool.Service(Service_ID));
8
```

Below the query editor, the "Output" tab is selected, showing the "Action Output" table:

#	Time	Action	Message
1	15:01:15	CREATE TABLE DrivingSchool.Service_lesson (Service_lesson_ID INT NOT NULL, Service_ID INT NOT NULL, ...	0 row(s) affected

Creating Vehicle_Details table:

CREATE TABLE DrivingSchool.Vehicle_details

(Vehicle_ID INT NOT NULL,

Vehicle_type_ID INT,

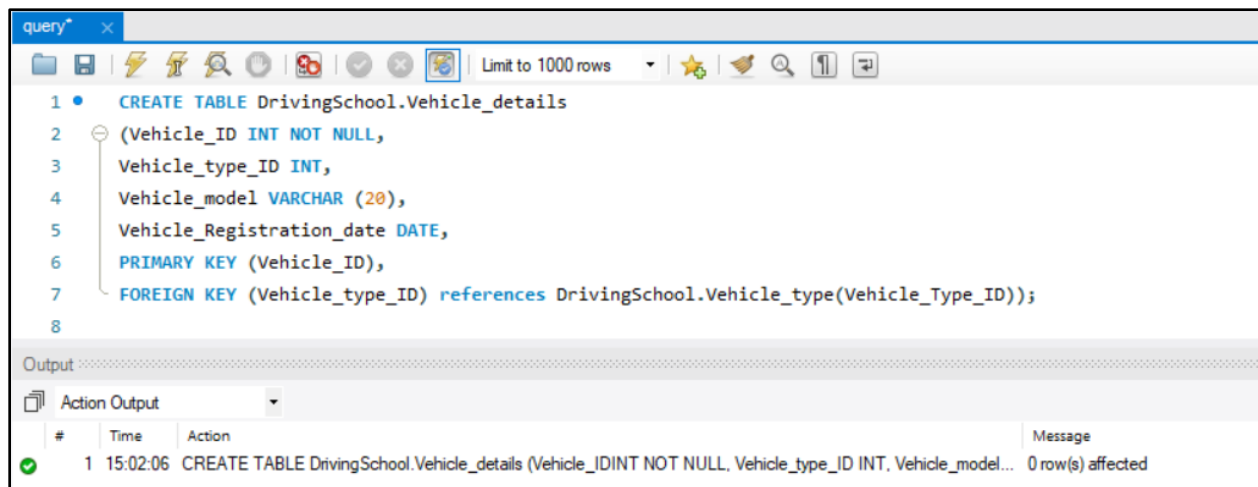
Vehicle_model VARCHAR (20),

Vehicle_Registration_date DATE,

PRIMARY KEY (Vehicle_ID),

FOREIGN KEY (Vehicle_type_ID) references DrivingSchool.Vehicle_type(Vehicle_Type_ID));

Output:



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

```
1 • CREATE TABLE DrivingSchool.Vehicle_details
2 (Vehicle_ID INT NOT NULL,
3  Vehicle_type_ID INT,
4  Vehicle_model VARCHAR (20),
5  Vehicle_Registration_date DATE,
6  PRIMARY KEY (Vehicle_ID),
7  FOREIGN KEY (Vehicle_type_ID) references DrivingSchool.Vehicle_type(Vehicle_Type_ID));
8
```

Below the query editor, the 'Output' tab is selected, showing a table with columns: #, Time, Action, and Message. The first row contains the following data:

#	Time	Action	Message
1	15:02:06	CREATE TABLE DrivingSchool.Vehicle_details (Vehicle_ID INT NOT NULL, Vehicle_type_ID INT, Vehicle_model...	0 row(s) affected

Creating Reservation Table:

```
CREATE TABLE DrivingSchool.Reservation
```

```
(Reservation_ID INT NOT NULL,
```

```
Instructor_ID INT NOT NULL,
```

```
Service_Lesson_ID INT NOT NULL,
```

```
Request_ID INT NOT NULL,
```

```
Vehicle_ID INT NOT NULL,
```

```
Lesson_Schedule_date DATE,
```

```
Reservation_status VARCHAR (50),
```

```
PRIMARY KEY (Reservation_ID),
```

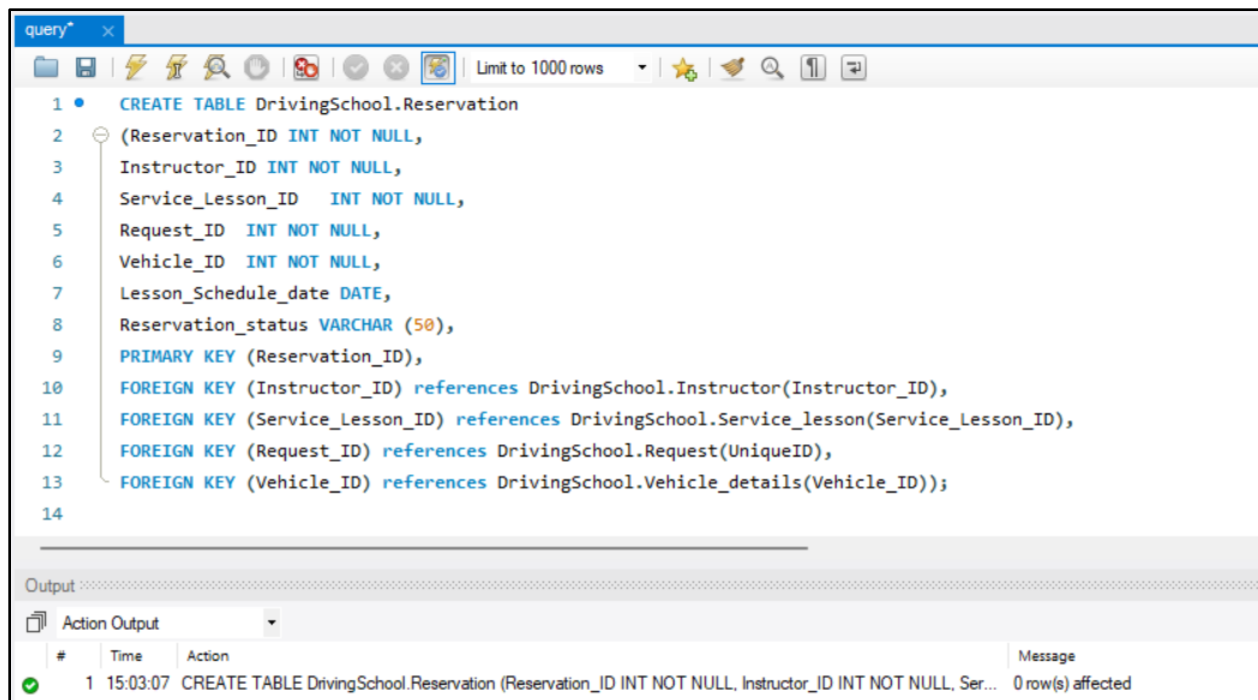
```
FOREIGN KEY (Instructor_ID) references DrivingSchool.Instructor(Instructor_ID),
```

```
FOREIGN KEY (Service_Lesson_ID) references DrivingSchool.Service_lesson(Service_Lesson_ID),
```

```
FOREIGN KEY (Request_ID) references DrivingSchool.Request(UniqueID),
```

```
FOREIGN KEY (Vehicle_ID) references DrivingSchool.Vehicle_details(Vehicle_ID));
```

Output:



The screenshot shows a SQL query editor window titled 'query'. The query is a CREATE TABLE statement for 'DrivingSchool.Reservation'. The table has the following columns: Reservation_ID (INT NOT NULL), Instructor_ID (INT NOT NULL), Service_Lesson_ID (INT NOT NULL), Request_ID (INT NOT NULL), Vehicle_ID (INT NOT NULL), Lesson_Schedule_date (DATE), and Reservation_status (VARCHAR (50)). The primary key is Reservation_ID. There are four foreign keys: Instructor_ID references DrivingSchool.Instructor(Instructor_ID), Service_Lesson_ID references DrivingSchool.Service_lesson(Service_Lesson_ID), Request_ID references DrivingSchool.Request(UniqueID), and Vehicle_ID references DrivingSchool.Vehicle_details(Vehicle_ID). The output section shows the execution of the query, with a message indicating that 0 row(s) were affected.

```
1 • CREATE TABLE DrivingSchool.Reservation
2   (Reservation_ID INT NOT NULL,
3     Instructor_ID INT NOT NULL,
4     Service_Lesson_ID INT NOT NULL,
5     Request_ID INT NOT NULL,
6     Vehicle_ID INT NOT NULL,
7     Lesson_Schedule_date DATE,
8     Reservation_status VARCHAR (50),
9     PRIMARY KEY (Reservation_ID),
10    FOREIGN KEY (Instructor_ID) references DrivingSchool.Instructor(Instructor_ID),
11    FOREIGN KEY (Service_Lesson_ID) references DrivingSchool.Service_lesson(Service_Lesson_ID),
12    FOREIGN KEY (Request_ID) references DrivingSchool.Request(UniqueID),
13    FOREIGN KEY (Vehicle_ID) references DrivingSchool.Vehicle_details(Vehicle_ID));
14
```

Output

#	Time	Action	Message
1	15:03:07	CREATE TABLE DrivingSchool.Reservation (Reservation_ID INT NOT NULL, Instructor_ID INT NOT NULL, Ser...	0 row(s) affected

Data Insertion in Tables:

Data Insertion in Service table:

INSERT INTO DrivingSchool.Service value(1,"MINOR SERVICE",50,"Teen 7X7 Driving classes");

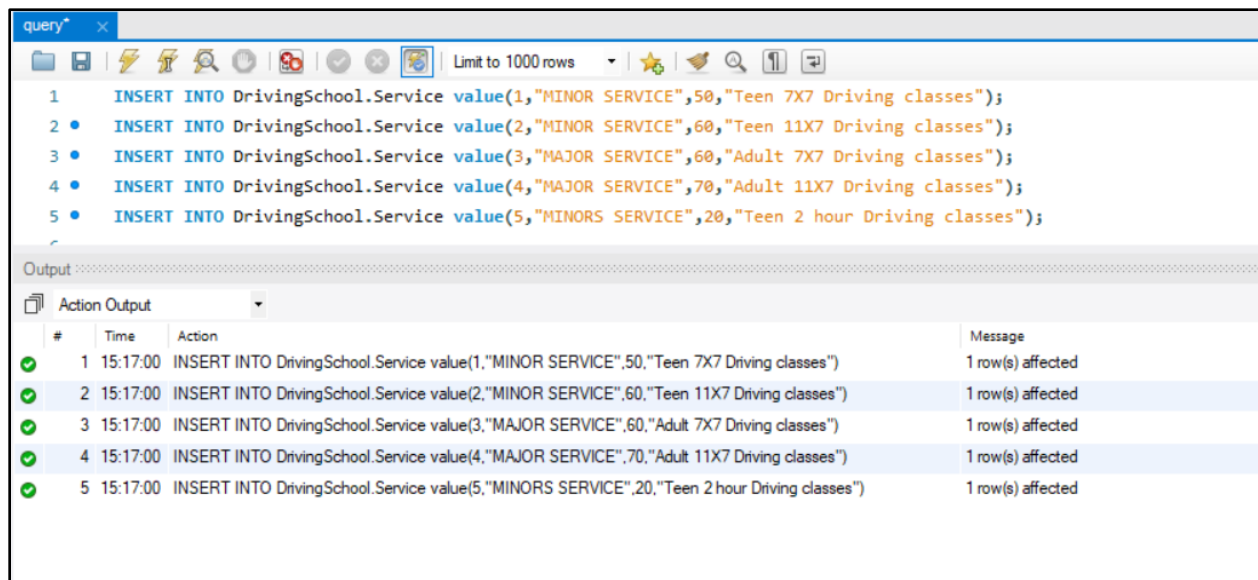
INSERT INTO DrivingSchool.Service value(2,"MINOR SERVICE",60,"Teen 11X7 Driving classes");

INSERT INTO DrivingSchool.Service value(3,"MAJOR SERVICE",60,"Adult 7X7 Driving classes");

INSERT INTO DrivingSchool.Service value(4,"MAJOR SERVICE",70,"Adult 11X7 Driving classes");

INSERT INTO DrivingSchool.Service value(5,"MINORS SERVICE",20,"Teen 2 hour Driving classes");

Output:



The screenshot shows a SQL query window with five INSERT statements for the DrivingSchool.Service table. The statements are as follows:

```
1 INSERT INTO DrivingSchool.Service value(1,"MINOR SERVICE",50,"Teen 7X7 Driving classes");
2 INSERT INTO DrivingSchool.Service value(2,"MINOR SERVICE",60,"Teen 11X7 Driving classes");
3 INSERT INTO DrivingSchool.Service value(3,"MAJOR SERVICE",60,"Adult 7X7 Driving classes");
4 INSERT INTO DrivingSchool.Service value(4,"MAJOR SERVICE",70,"Adult 11X7 Driving classes");
5 INSERT INTO DrivingSchool.Service value(5,"MINORS SERVICE",20,"Teen 2 hour Driving classes");
```

The Action Output window shows the results of these statements:

#	Time	Action	Message
1	15:17:00	INSERT INTO DrivingSchool.Service value(1,"MINOR SERVICE",50,"Teen 7X7 Driving classes")	1 row(s) affected
2	15:17:00	INSERT INTO DrivingSchool.Service value(2,"MINOR SERVICE",60,"Teen 11X7 Driving classes")	1 row(s) affected
3	15:17:00	INSERT INTO DrivingSchool.Service value(3,"MAJOR SERVICE",60,"Adult 7X7 Driving classes")	1 row(s) affected
4	15:17:00	INSERT INTO DrivingSchool.Service value(4,"MAJOR SERVICE",70,"Adult 11X7 Driving classes")	1 row(s) affected
5	15:17:00	INSERT INTO DrivingSchool.Service value(5,"MINORS SERVICE",20,"Teen 2 hour Driving classes")	1 row(s) affected

Data Insertion in Vehicle_Type table:

INSERT INTO DrivingSchool.Vehicle_type values(1, "Truck");

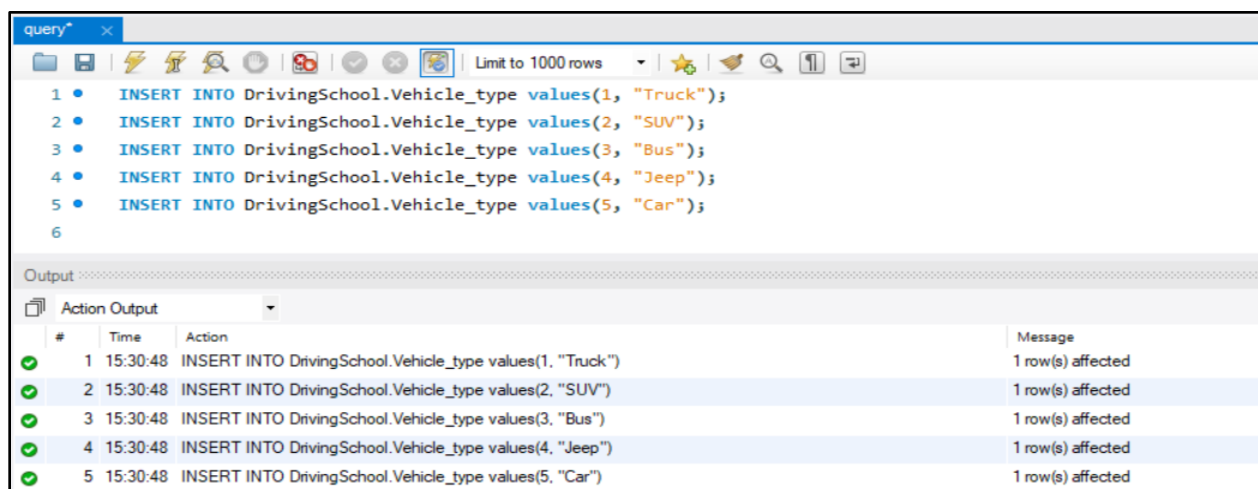
INSERT INTO DrivingSchool.Vehicle_type values(2, "SUV");

INSERT INTO DrivingSchool.Vehicle_type values(3, "Bus");

INSERT INTO DrivingSchool.Vehicle_type values(4, "Jeep");

INSERT INTO DrivingSchool.Vehicle_type values(5, "Car");

Output:



The screenshot shows a SQL query window with five INSERT statements for the DrivingSchool.Vehicle_type table. The statements are as follows:

```
1 INSERT INTO DrivingSchool.Vehicle_type values(1, "Truck");
2 INSERT INTO DrivingSchool.Vehicle_type values(2, "SUV");
3 INSERT INTO DrivingSchool.Vehicle_type values(3, "Bus");
4 INSERT INTO DrivingSchool.Vehicle_type values(4, "Jeep");
5 INSERT INTO DrivingSchool.Vehicle_type values(5, "Car");
```

The Action Output window shows the results of these statements:

#	Time	Action	Message
1	15:30:48	INSERT INTO DrivingSchool.Vehicle_type values(1, "Truck")	1 row(s) affected
2	15:30:48	INSERT INTO DrivingSchool.Vehicle_type values(2, "SUV")	1 row(s) affected
3	15:30:48	INSERT INTO DrivingSchool.Vehicle_type values(3, "Bus")	1 row(s) affected
4	15:30:48	INSERT INTO DrivingSchool.Vehicle_type values(4, "Jeep")	1 row(s) affected
5	15:30:48	INSERT INTO DrivingSchool.Vehicle_type values(5, "Car")	1 row(s) affected

Data Insertion in Customer table:

```
INSERT INTO DrivingSchool.Customer values(1, "Manasa","Cherukupally",28,"3925 N Elm St,Denton",4695866122, "manasa@gmail.com", "Class C");
```

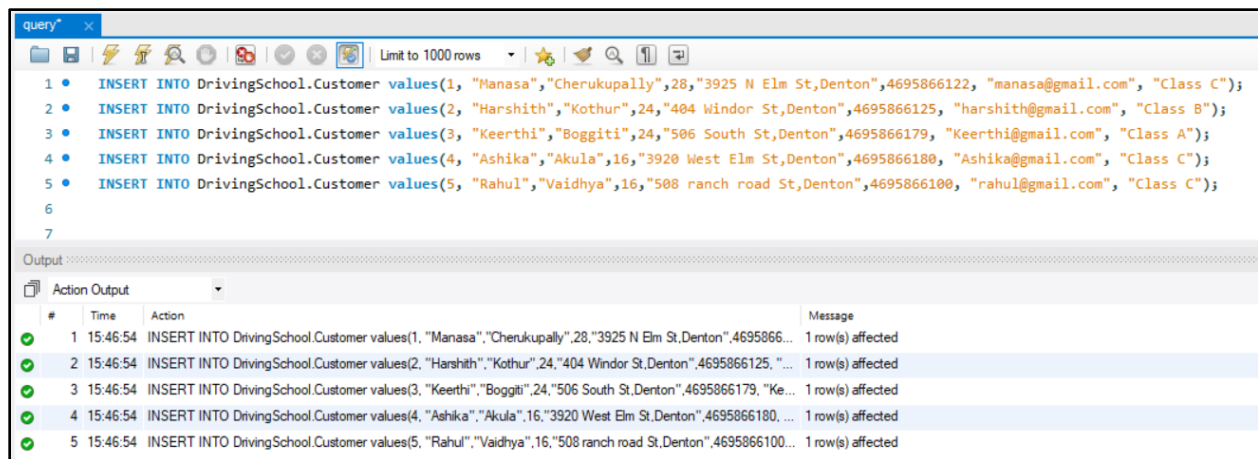
```
INSERT INTO DrivingSchool.Customer values(2, "Harshith","Kothur",24,"404 Windor St,Denton",4695866125, "harshith@gmail.com", "Class B");
```

```
INSERT INTO DrivingSchool.Customer values(3, "Keerthi","Boggiti",24,"506 South St,Denton",4695866179, "Keerthi@gmail.com", "Class A");
```

```
INSERT INTO DrivingSchool.Customer values(4, "Ashika","Akula",16,"3920 West Elm St,Denton",4695866180, "Ashika@gmail.com", "Class C");
```

```
INSERT INTO DrivingSchool.Customer values(5, "Rahul","Vaidhya",16,"508 ranch road St,Denton",4695866100, "rahul@gmail.com", "Class C");
```

Output:



The screenshot shows a SQL query window with five INSERT statements. Below the queries, the 'Output' tab displays the execution results, showing that each statement successfully affected one row.

#	Time	Action	Message
1	15:46:54	INSERT INTO DrivingSchool.Customer values(1, "Manasa","Cherukupally",28,"3925 N Elm St,Denton",4695866122, "manasa@gmail.com", "Class C");	1 row(s) affected
2	15:46:54	INSERT INTO DrivingSchool.Customer values(2, "Harshith","Kothur",24,"404 Windor St,Denton",4695866125, "harshith@gmail.com", "Class B");	1 row(s) affected
3	15:46:54	INSERT INTO DrivingSchool.Customer values(3, "Keerthi","Boggiti",24,"506 South St,Denton",4695866179, "Keerthi@gmail.com", "Class A");	1 row(s) affected
4	15:46:54	INSERT INTO DrivingSchool.Customer values(4, "Ashika","Akula",16,"3920 West Elm St,Denton",4695866180, "Ashika@gmail.com", "Class C");	1 row(s) affected
5	15:46:54	INSERT INTO DrivingSchool.Customer values(5, "Rahul","Vaidhya",16,"508 ranch road St,Denton",4695866100, "rahul@gmail.com", "Class C");	1 row(s) affected

Data Insertion in Request table:

```
INSERT INTO DrivingSchool.Request values(1,3,1,"2022-12-19","2022-12-15",5,"active");
```

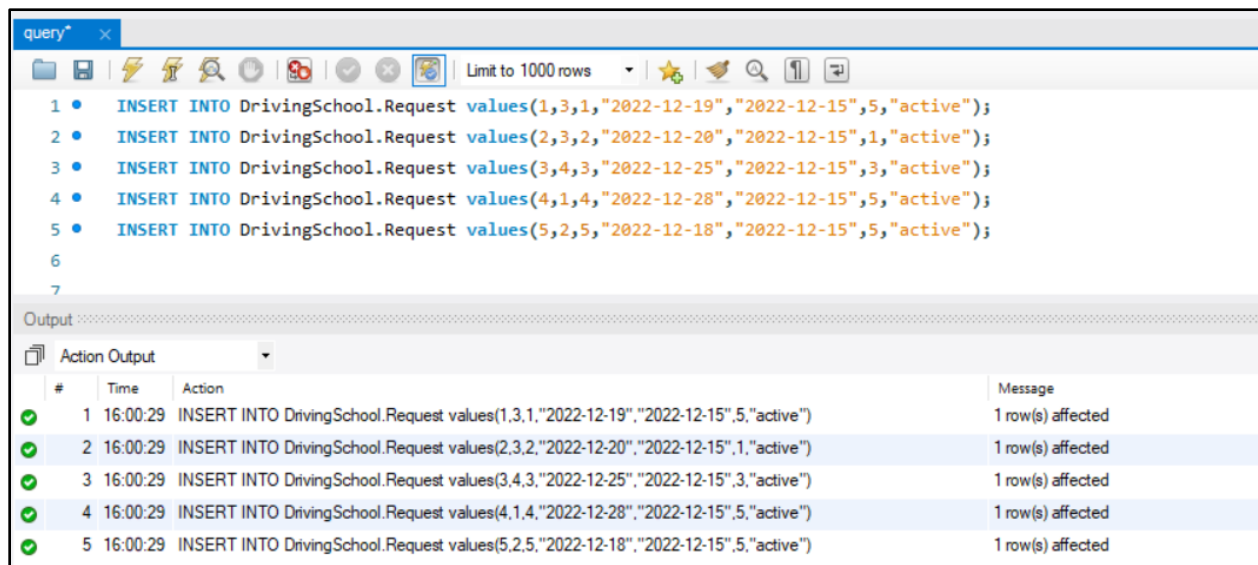
```
INSERT INTO DrivingSchool.Request values(2,3,2,"2022-12-20","2022-12-15",1,"active");
```

```
INSERT INTO DrivingSchool.Request values(3,4,3,"2022-12-25","2022-12-15",3,"active");
```

```
INSERT INTO DrivingSchool.Request values(4,1,4,"2022-12-28","2022-12-15",5,"active");
```

```
INSERT INTO DrivingSchool.Request values(5,2,5,"2022-12-18","2022-12-15",5,"active");
```

Output:



The screenshot shows a SQL query editor with five INSERT statements. The output pane below shows the execution results for each statement, indicating that each row was successfully inserted and affected.

#	Time	Action	Message
1	16:00:29	INSERT INTO DrivingSchool.Request values(1,3,1,"2022-12-19","2022-12-15",5,"active")	1 row(s) affected
2	16:00:29	INSERT INTO DrivingSchool.Request values(2,3,2,"2022-12-20","2022-12-15",1,"active")	1 row(s) affected
3	16:00:29	INSERT INTO DrivingSchool.Request values(3,4,3,"2022-12-25","2022-12-15",3,"active")	1 row(s) affected
4	16:00:29	INSERT INTO DrivingSchool.Request values(4,1,4,"2022-12-28","2022-12-15",5,"active")	1 row(s) affected
5	16:00:29	INSERT INTO DrivingSchool.Request values(5,2,5,"2022-12-18","2022-12-15",5,"active")	1 row(s) affected

Data Insertion in Instructor table:

```
INSERT INTO DrivingSchool.Instructor values(1, "Prashanth","Dadi","3925 N eml  
St,Denton","Available","Prashanth@gmail.com","4695861234");
```

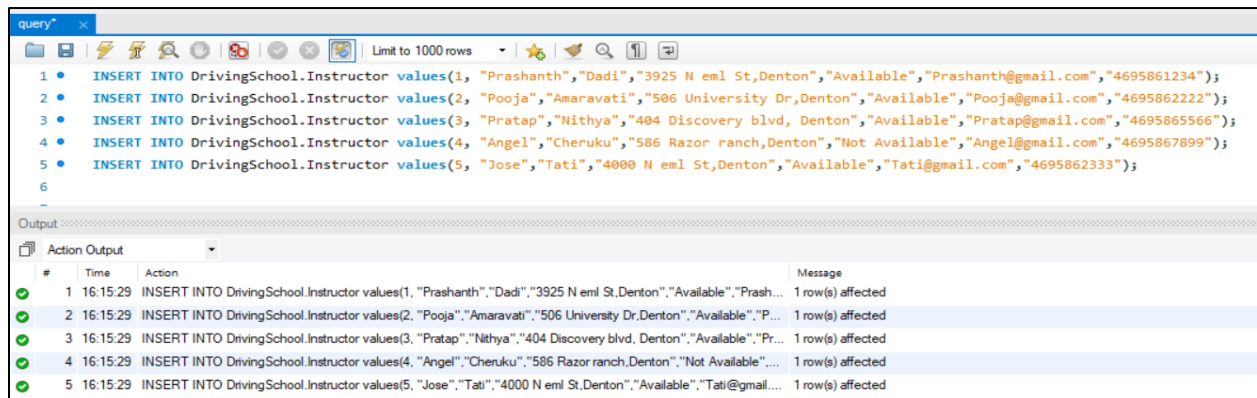
```
INSERT INTO DrivingSchool.Instructor values(2, "Pooja","Amaravati","506 University  
Dr,Denton","Available","Pooja@gmail.com","4695862222");
```

```
INSERT INTO DrivingSchool.Instructor values(3, "Pratap","Nithya","404 Discovery blvd,  
Denton","Available","Pratap@gmail.com","4695865566");
```

```
INSERT INTO DrivingSchool.Instructor values(4, "Angel","Cheruku","586 Razor  
ranch,Denton","Not Available","Angel@gmail.com","4695867899");
```

```
INSERT INTO DrivingSchool.Instructor values(5, "Jose","Tati","4000 N eml  
St,Denton","Available","Tati@gmail.com","4695862333");
```


Output:



The screenshot shows a SQL query window with five INSERT statements for the DrivingSchool.Instructor table. The output pane displays a table with columns: #, Time, Action, and Message. Each row shows a successful insert operation with a message '1 row(s) affected'.

#	Time	Action	Message
1	16:15:29	INSERT INTO DrivingSchool.Instructor values(1, "Prashanth","Dadi","3925 N eml St,Denton","Available","Prashanth@gmail.com","4695861234");	1 row(s) affected
2	16:15:29	INSERT INTO DrivingSchool.Instructor values(2, "Pooja","Amaravati","506 University Dr,Denton","Available","Pooja@gmail.com","4695862222");	1 row(s) affected
3	16:15:29	INSERT INTO DrivingSchool.Instructor values(3, "Pratap","Nithya","404 Discovery blvd, Denton","Available","Pratap@gmail.com","4695865566");	1 row(s) affected
4	16:15:29	INSERT INTO DrivingSchool.Instructor values(4, "Angel","Cheruku","586 Razor ranch,Denton","Not Available","Angel@gmail.com","4695867899");	1 row(s) affected
5	16:15:29	INSERT INTO DrivingSchool.Instructor values(5, "Jose","Tati","4000 N eml St,Denton","Available","Tati@gmail.com","4695862333");	1 row(s) affected

Data Insertion in Service_Lesson table:

INSERT INTO DrivingSchool.Service_Lesson values(1,1,350,7);

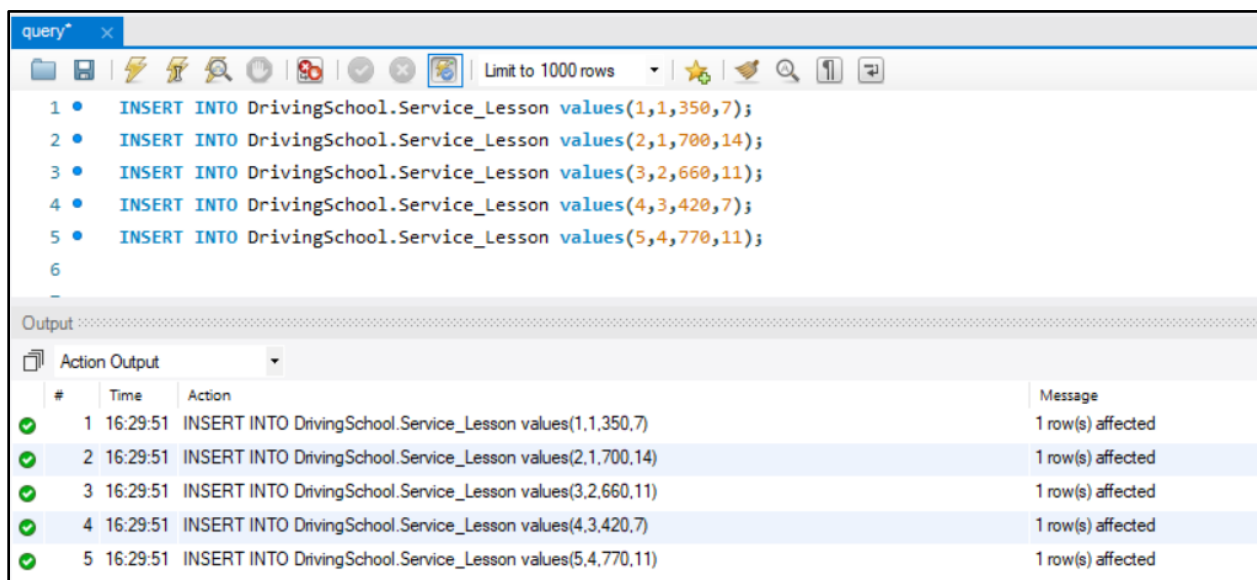
INSERT INTO DrivingSchool.Service_Lesson values(2,1,700,14);

INSERT INTO DrivingSchool.Service_Lesson values(3,2,660,11);

INSERT INTO DrivingSchool.Service_Lesson values(4,3,420,7);

INSERT INTO DrivingSchool.Service_Lesson values(5,4,770,11);

Output:



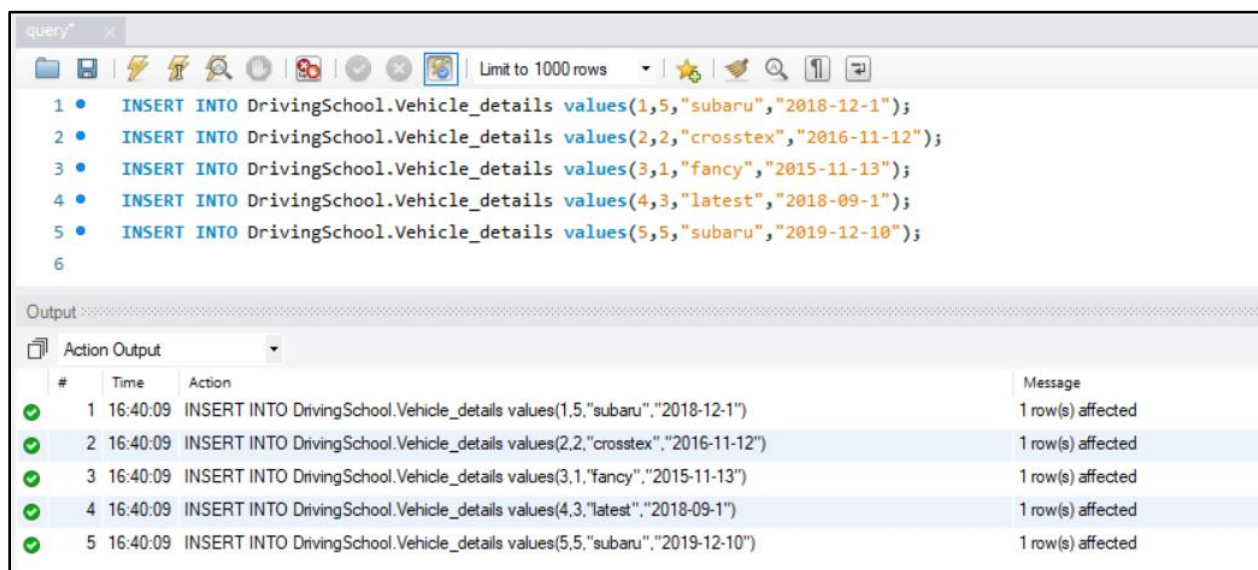
The screenshot shows a SQL query window with five INSERT statements for the DrivingSchool.Service_Lesson table. The output pane displays a table with columns: #, Time, Action, and Message. Each row shows a successful insert operation with a message '1 row(s) affected'.

#	Time	Action	Message
1	16:29:51	INSERT INTO DrivingSchool.Service_Lesson values(1,1,350,7);	1 row(s) affected
2	16:29:51	INSERT INTO DrivingSchool.Service_Lesson values(2,1,700,14);	1 row(s) affected
3	16:29:51	INSERT INTO DrivingSchool.Service_Lesson values(3,2,660,11);	1 row(s) affected
4	16:29:51	INSERT INTO DrivingSchool.Service_Lesson values(4,3,420,7);	1 row(s) affected
5	16:29:51	INSERT INTO DrivingSchool.Service_Lesson values(5,4,770,11);	1 row(s) affected

Data Insertion in Vehicle_details table:

```
INSERT INTO DrivingSchool.Vehicle_details values(1,5,"subaru","2018-12-1");  
INSERT INTO DrivingSchool.Vehicle_details values(2,2,"crosstex","2016-11-12");  
INSERT INTO DrivingSchool.Vehicle_details values(3,1,"fancy","2015-11-13");  
INSERT INTO DrivingSchool.Vehicle_details values(4,3,"latest","2018-09-1");  
INSERT INTO DrivingSchool.Vehicle_details values(5,5,"subaru","2019-12-10");
```

Output:



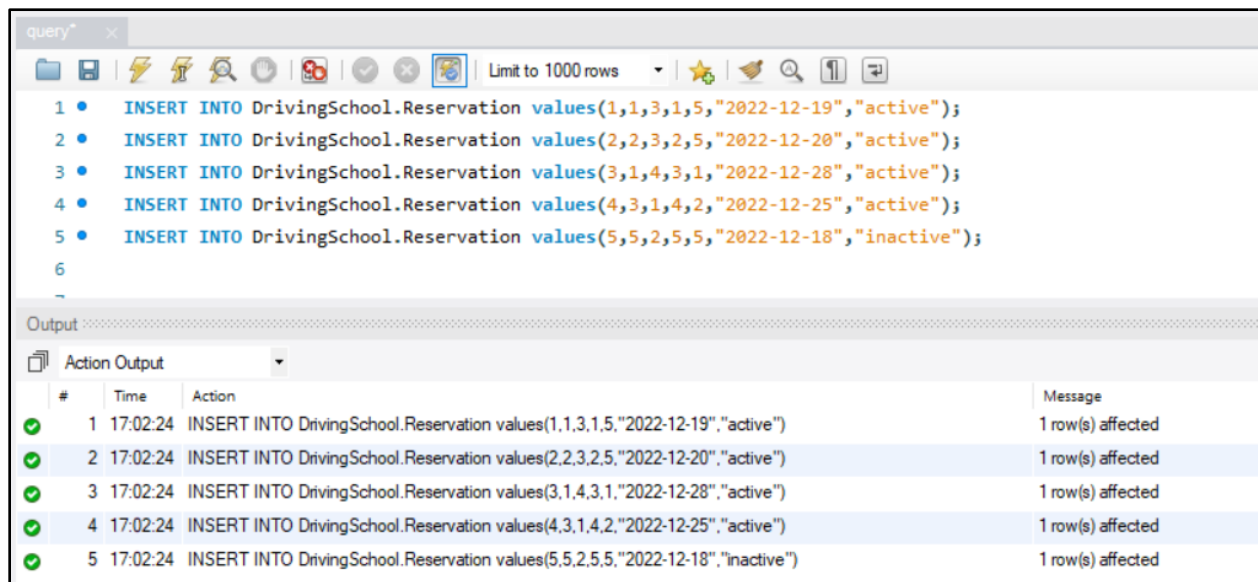
The screenshot shows a SQL query window with five INSERT statements. Below the query editor, the 'Output' tab is selected, displaying the 'Action Output' table. The table has four columns: '#', 'Time', 'Action', and 'Message'. It shows five rows of successful insertions, each with a green checkmark in the first column and a message indicating '1 row(s) affected'.

#	Time	Action	Message
✓ 1	16:40:09	INSERT INTO DrivingSchool.Vehicle_details values(1,5,"subaru","2018-12-1")	1 row(s) affected
✓ 2	16:40:09	INSERT INTO DrivingSchool.Vehicle_details values(2,2,"crosstex","2016-11-12")	1 row(s) affected
✓ 3	16:40:09	INSERT INTO DrivingSchool.Vehicle_details values(3,1,"fancy","2015-11-13")	1 row(s) affected
✓ 4	16:40:09	INSERT INTO DrivingSchool.Vehicle_details values(4,3,"latest","2018-09-1")	1 row(s) affected
✓ 5	16:40:09	INSERT INTO DrivingSchool.Vehicle_details values(5,5,"subaru","2019-12-10")	1 row(s) affected

Data Insertion in Reservation table:

```
INSERT INTO DrivingSchool.Reservation values(1,1,3,1,5,"2022-12-19","active");  
INSERT INTO DrivingSchool.Reservation values(2,2,3,2,5,"2022-12-20","active");  
INSERT INTO DrivingSchool.Reservation values(3,1,4,3,1,"2022-12-28","active");  
INSERT INTO DrivingSchool.Reservation values(4,3,1,4,2,"2022-12-25","active");  
INSERT INTO DrivingSchool.Reservation values(5,5,2,5,5,"2022-12-18","inactive");
```

Output:



The screenshot shows a database query editor with a toolbar at the top. The query area contains five INSERT statements into the DrivingSchool.Reservation table. Below the query, the 'Output' section is set to 'Action Output', displaying a log of the executed actions with their timestamps and messages.

```
1 • INSERT INTO DrivingSchool.Reservation values(1,1,3,1,5,"2022-12-19","active");
2 • INSERT INTO DrivingSchool.Reservation values(2,2,3,2,5,"2022-12-20","active");
3 • INSERT INTO DrivingSchool.Reservation values(3,1,4,3,1,"2022-12-28","active");
4 • INSERT INTO DrivingSchool.Reservation values(4,3,1,4,2,"2022-12-25","active");
5 • INSERT INTO DrivingSchool.Reservation values(5,5,2,5,5,"2022-12-18","inactive");
6
```

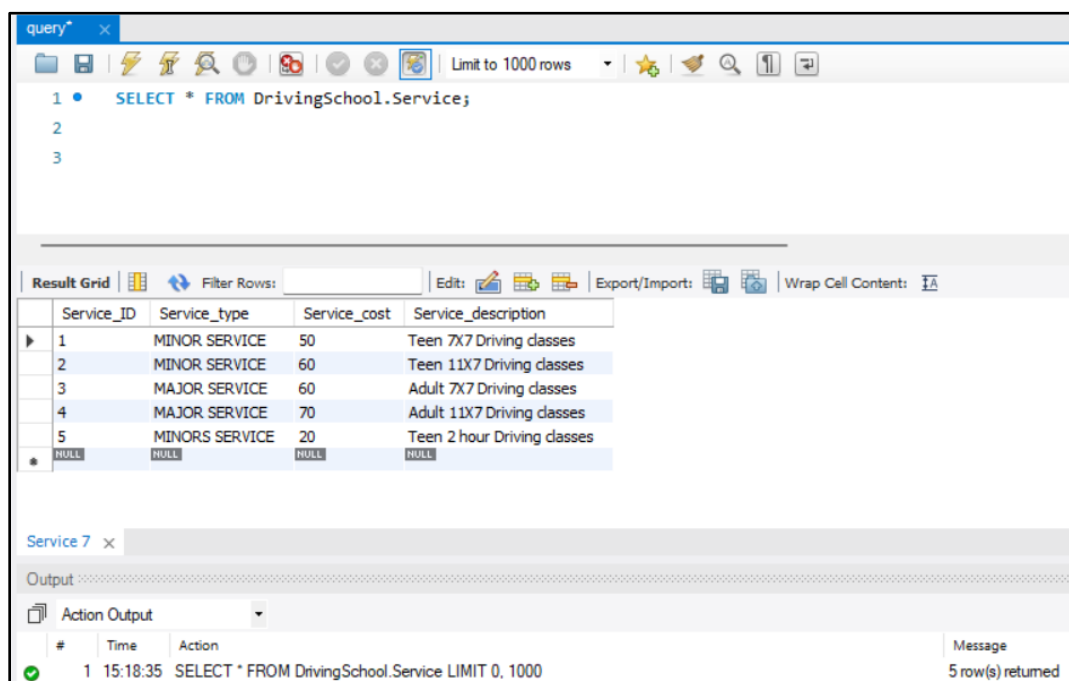
#	Time	Action	Message
✓ 1	17:02:24	INSERT INTO DrivingSchool.Reservation values(1,1,3,1,5,"2022-12-19","active")	1 row(s) affected
✓ 2	17:02:24	INSERT INTO DrivingSchool.Reservation values(2,2,3,2,5,"2022-12-20","active")	1 row(s) affected
✓ 3	17:02:24	INSERT INTO DrivingSchool.Reservation values(3,1,4,3,1,"2022-12-28","active")	1 row(s) affected
✓ 4	17:02:24	INSERT INTO DrivingSchool.Reservation values(4,3,1,4,2,"2022-12-25","active")	1 row(s) affected
✓ 5	17:02:24	INSERT INTO DrivingSchool.Reservation values(5,5,2,5,5,"2022-12-18","inactive")	1 row(s) affected

Data Selection from Tables:

Select query on Service table:

Select * from DrivingSchool.Service;

Output:



The screenshot shows a database query editor with a toolbar. The query area contains a single SELECT statement. Below the query, the 'Result Grid' is displayed, showing the results of the query. The 'Output' section is also visible, showing the execution log.

```
1 • SELECT * FROM DrivingSchool.Service;
2
3
```

Service_ID	Service_type	Service_cost	Service_description
1	MINOR SERVICE	50	Teen 7X7 Driving classes
2	MINOR SERVICE	60	Teen 11X7 Driving classes
3	MAJOR SERVICE	60	Adult 7X7 Driving classes
4	MAJOR SERVICE	70	Adult 11X7 Driving classes
5	MINORS SERVICE	20	Teen 2 hour Driving classes
* NULL	NULL	NULL	NULL

Service 7 x

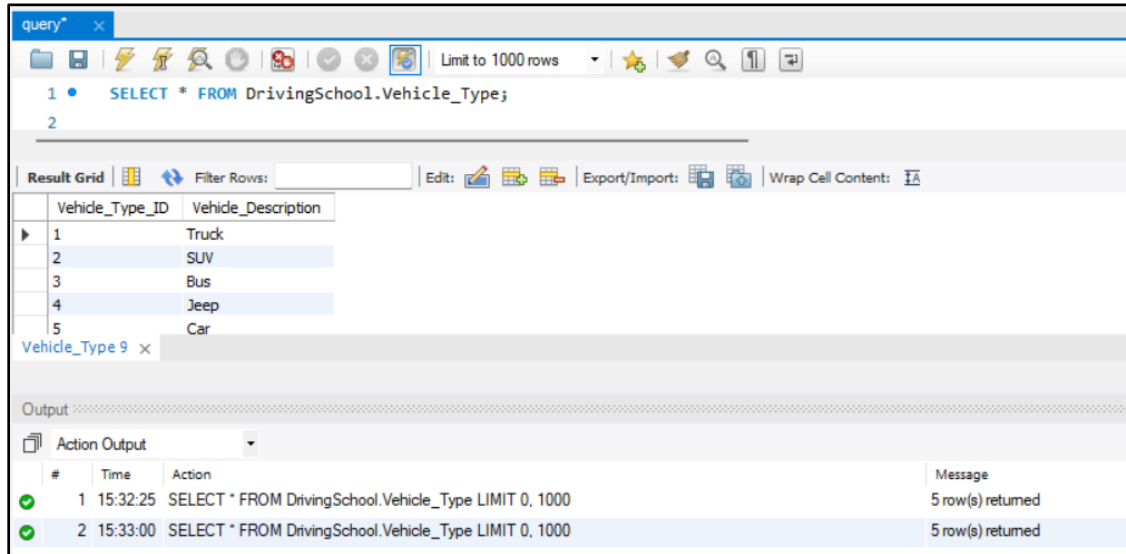
Output

#	Time	Action	Message
✓ 1	15:18:35	SELECT * FROM DrivingSchool.Service LIMIT 0, 1000	5 row(s) returned

Select query on Vehicle_type table:

Select * from DrivingSchool.Vehicle_type;

Output:



query* x

Limit to 1000 rows

```
1 • SELECT * FROM DrivingSchool.Vehicle_Type;
```

2

Result Grid

Vehicle_Type_ID	Vehicle_Description
1	Truck
2	SUV
3	Bus
4	Jeep
5	Car

Vehicle_Type 9 x

Output

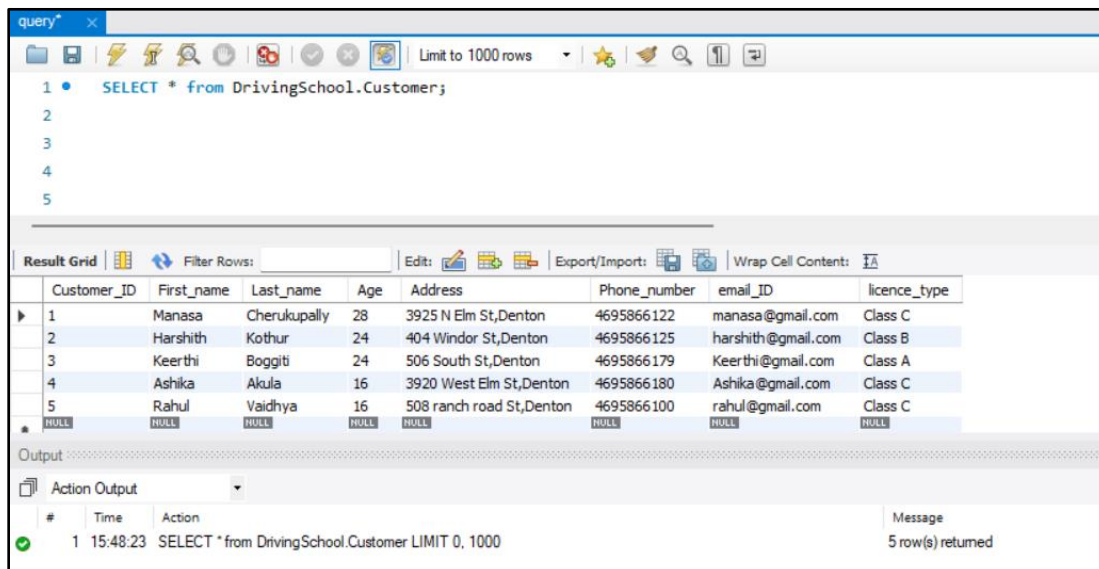
Action Output

#	Time	Action	Message
1	15:32:25	SELECT * FROM DrivingSchool.Vehicle_Type LIMIT 0, 1000	5 row(s) returned
2	15:33:00	SELECT * FROM DrivingSchool.Vehicle_Type LIMIT 0, 1000	5 row(s) returned

Select query on Customer table:

Select * from DrivingSchool.Customer;

Output:



query* x

Limit to 1000 rows

```
1 • SELECT * from DrivingSchool.Customer;
```

2

3

4

5

Result Grid

Customer_ID	First_name	Last_name	Age	Address	Phone_number	email_ID	licence_type
1	Manasa	Cherukupally	28	3925 N Elm St,Denton	4695866122	manasa@gmail.com	Class C
2	Harshith	Kothur	24	404 Windor St,Denton	4695866125	harshith@gmail.com	Class B
3	Keerthi	Boggiti	24	506 South St,Denton	4695866179	Keerthi@gmail.com	Class A
4	Ashika	Akula	16	3920 West Elm St,Denton	4695866180	Ashika@gmail.com	Class C
5	Rahul	Vaidhya	16	508 ranch road St,Denton	4695866100	rahul@gmail.com	Class C
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output

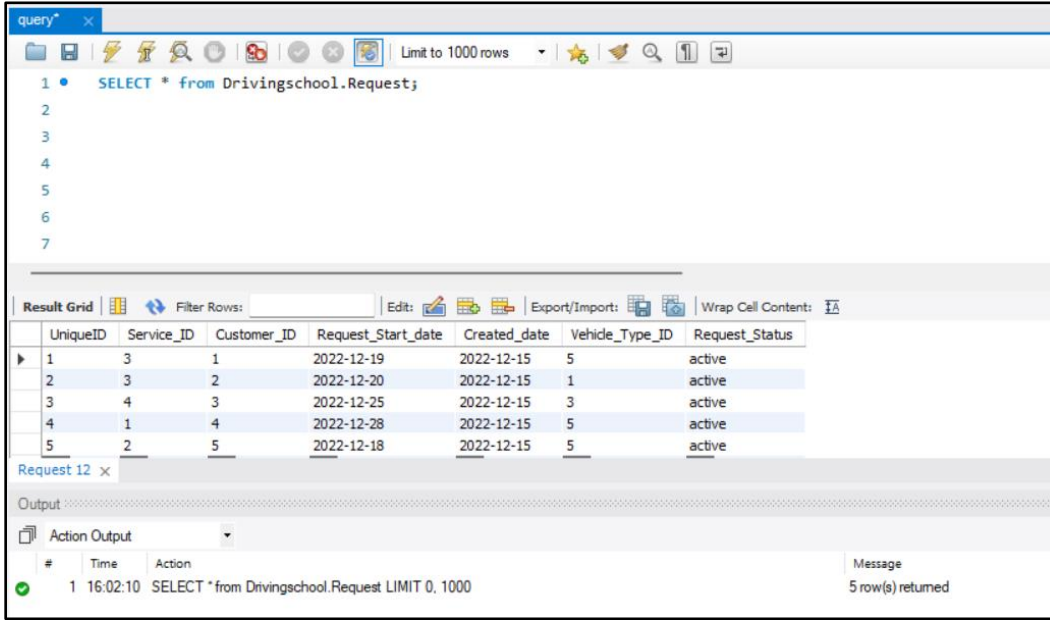
Action Output

#	Time	Action	Message
1	15:48:23	SELECT * from DrivingSchool.Customer LIMIT 0, 1000	5 row(s) returned

Select query on Request table:

Select * from DrivingSchool.Request;

Output:



The screenshot shows a query editor window titled 'query*'. The SQL query entered is 'SELECT * from Drivingschool.Request;'. Below the query, a 'Result Grid' displays the results of the query. The table has 7 columns: UniqueID, Service_ID, Customer_ID, Request_Start_date, Created_date, Vehicle_Type_ID, and Request_Status. There are 5 rows of data, all with a status of 'active'.

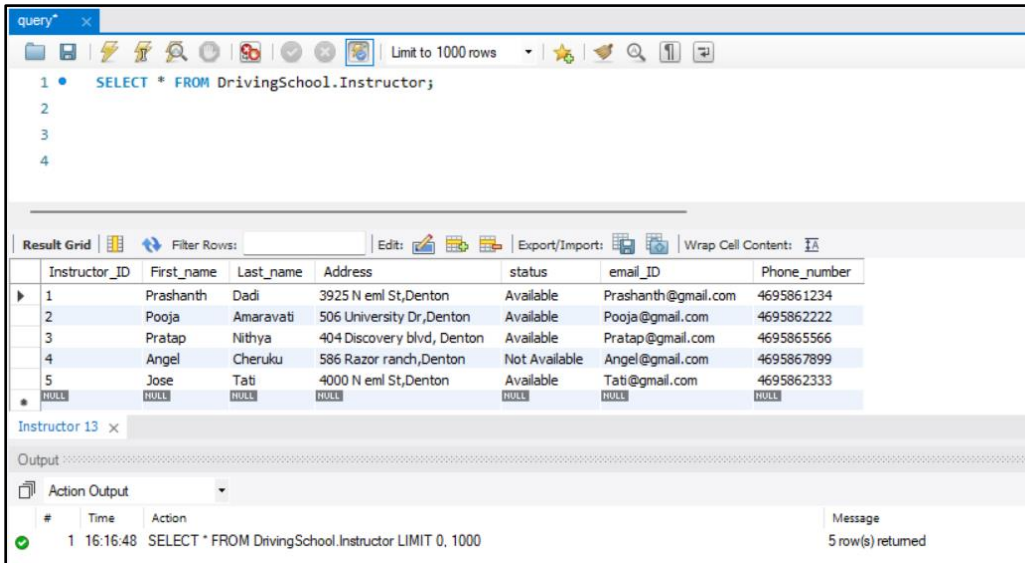
UniqueID	Service_ID	Customer_ID	Request_Start_date	Created_date	Vehicle_Type_ID	Request_Status
1	3	1	2022-12-19	2022-12-15	5	active
2	3	2	2022-12-20	2022-12-15	1	active
3	4	3	2022-12-25	2022-12-15	3	active
4	1	4	2022-12-28	2022-12-15	5	active
5	2	5	2022-12-18	2022-12-15	5	active

Below the table, the 'Output' section shows the execution details: 'Action Output' is selected, and the message indicates '5 row(s) returned'.

Select query on Instructor table:

Select * from DrivingSchool.Instructor;

Output:



The screenshot shows a query editor window titled 'query*'. The SQL query entered is 'SELECT * FROM DrivingSchool.Instructor;'. Below the query, a 'Result Grid' displays the results of the query. The table has 8 columns: Instructor_ID, First_name, Last_name, Address, status, email_ID, and Phone_number. There are 5 rows of data, all with a status of 'Available'.

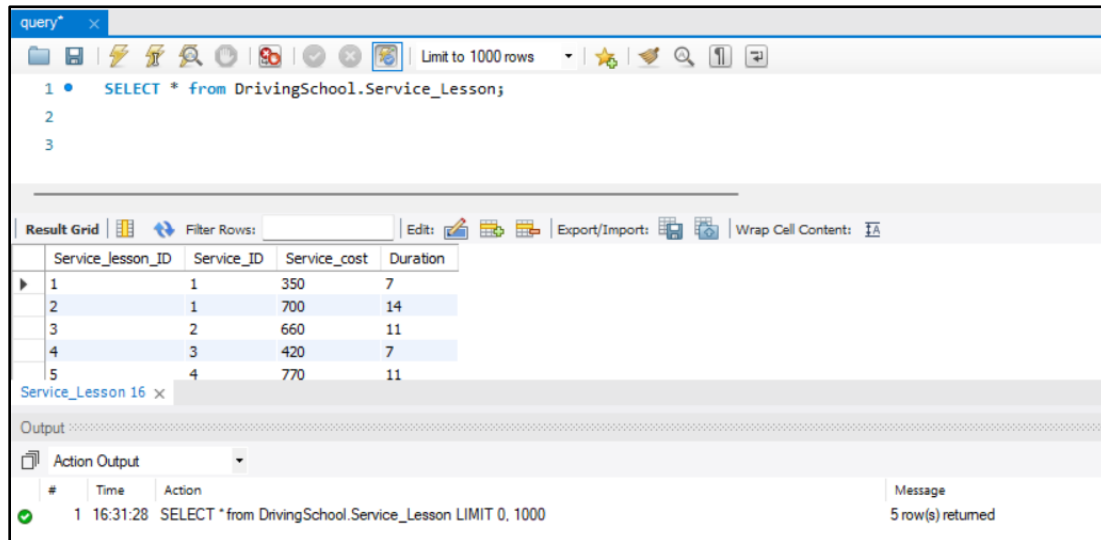
Instructor_ID	First_name	Last_name	Address	status	email_ID	Phone_number
1	Prashanth	Dadi	3925 N eml St,Denton	Available	Prashanth@gmail.com	4695861234
2	Pooja	Amaravati	506 University Dr,Denton	Available	Pooja@gmail.com	4695862222
3	Pratap	Nithya	404 Discovery blvd, Denton	Available	Pratap@gmail.com	4695865566
4	Angel	Cheruku	586 Razor ranch, Denton	Not Available	Angel@gmail.com	4695867899
5	Jose	Tati	4000 N eml St,Denton	Available	Tati@gmail.com	4695862333

Below the table, the 'Output' section shows the execution details: 'Action Output' is selected, and the message indicates '5 row(s) returned'.

Select query on Service_lesson table:

Select * from DrivingSchool.Service_Lesson;

Output:



query*

1 • SELECT * from DrivingSchool.Service_Lesson;

2

3

Result Grid

	Service_lesson_ID	Service_ID	Service_cost	Duration
▶ 1	1	1	350	7
2	1	1	700	14
3	2	2	660	11
4	3	3	420	7
5	4	4	770	11

Service_Lesson 16 x

Output

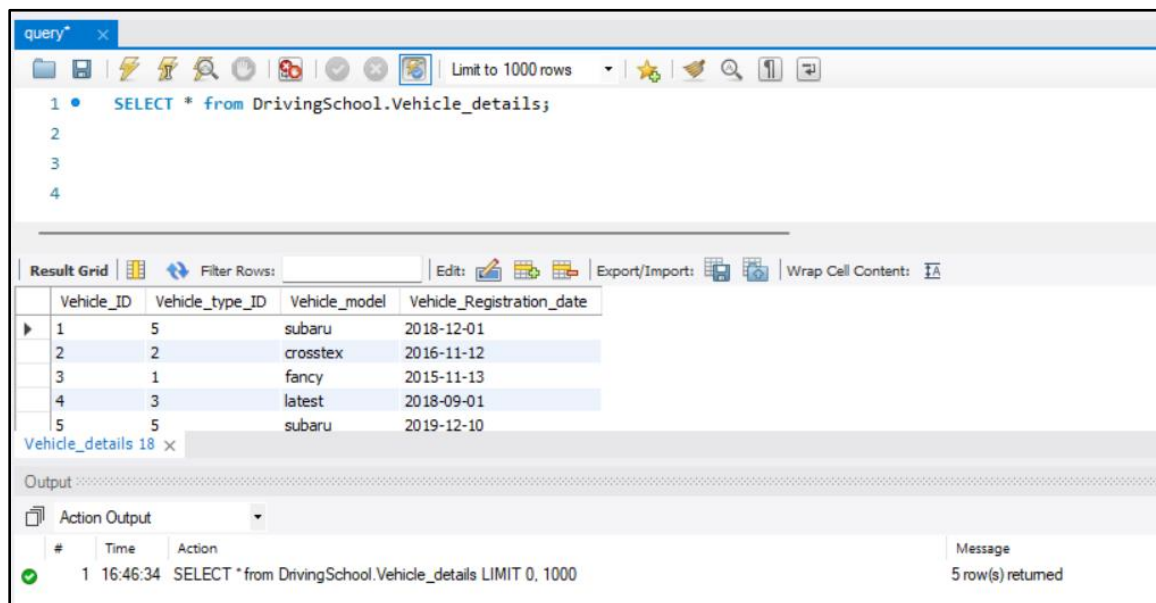
Action Output

#	Time	Action	Message
✓ 1	16:31:28	SELECT * from DrivingSchool.Service_Lesson LIMIT 0, 1000	5 row(s) returned

Select query on Vehicle_details table:

Select * from DrivingSchool.Vehicle_details;

Output:



query*

1 • SELECT * from DrivingSchool.Vehicle_details;

2

3

4

Result Grid

	Vehicle_ID	Vehicle_type_ID	Vehicle_model	Vehicle_Registration_date
▶ 1	5	5	subaru	2018-12-01
2	2	2	crosstex	2016-11-12
3	1	1	fancy	2015-11-13
4	3	3	latest	2018-09-01
5	5	5	subaru	2019-12-10

Vehicle_details 18 x

Output

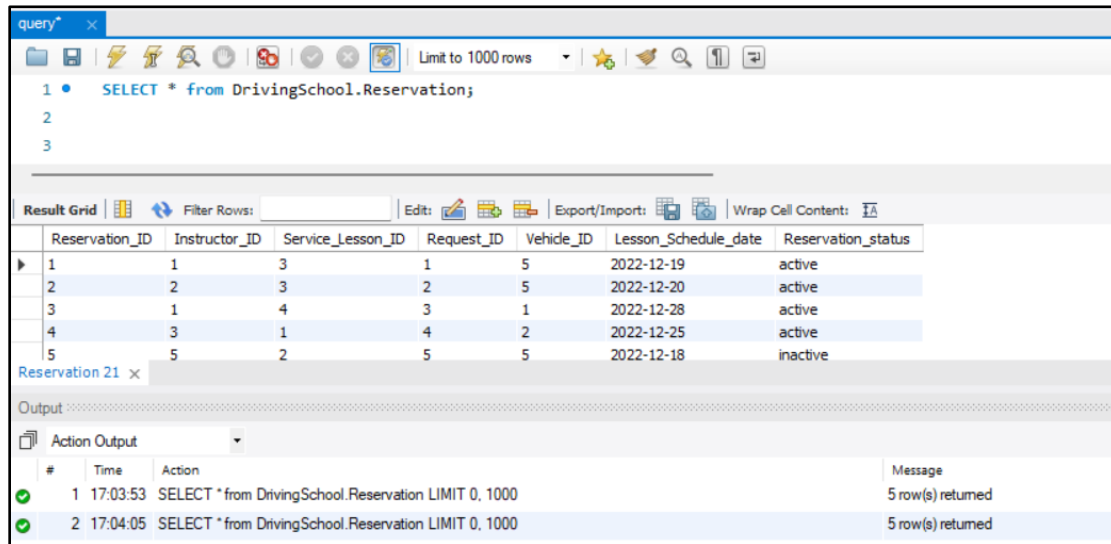
Action Output

#	Time	Action	Message
✓ 1	16:46:34	SELECT * from DrivingSchool.Vehicle_details LIMIT 0, 1000	5 row(s) returned

Select query on Reservation table:

Select * from DrivingSchool.Reservation;

Output:



query*

```
1 • SELECT * from DrivingSchool.Reservation;
2
3
```

Result Grid

	Reservation_ID	Instructor_ID	Service_Lesson_ID	Request_ID	Vehicle_ID	Lesson_Schedule_date	Reservation_status
1	1	1	3	1	5	2022-12-19	active
2	2	2	3	2	5	2022-12-20	active
3	1	4	3	1	1	2022-12-28	active
4	3	1	4	2	2	2022-12-25	active
5	5	2	5	5	5	2022-12-18	inactive

Reservation 21 x

Output

Action Output

#	Time	Action	Message
1	17:03:53	SELECT * from DrivingSchool.Reservation LIMIT 0, 1000	5 row(s) returned
2	17:04:05	SELECT * from DrivingSchool.Reservation LIMIT 0, 1000	5 row(s) returned

Modifying Tables:

Modify first_name in Customer table:

UPDATE DrivingSchool.Customer

SET first_name="Manasaa"

WHERE first_name="Manasa";

Output:

query*

Update Last_name in Instructor table:

```
UPDATE DrivingSchool.Instructor
```

```
SET last_name="Joy"
```

```
WHERE first_name="Pooja";
```

Output:

The screenshot shows a database query tool interface. The query editor contains the following SQL statements:

```
1 • UPDATE DrivingSchool.Instructor
2   SET last_name="Joy"
3   WHERE first_name="Pooja";
4
5 • Select * from DrivingSchool.Instructor;
```

Below the query editor is a 'Result Grid' showing the data from the 'Instructor' table. The grid has columns: Instructor_ID, First_name, Last_name, Address, status, email_ID, and Phone_number. The data is as follows:

Instructor_ID	First_name	Last_name	Address	status	email_ID	Phone_number
1	Prashanth	Dadi	3925 N emil St,Denton	Available	Prashanth@gmail.com	4695861234
2	Pooja	Joy	506 University Dr,Denton	Available	Pooja@gmail.com	4695862222
3	Pratap	Nithya	404 Discovery blvd, Denton	Available	Pratap@gmail.com	4695865566
4	Angel	Cheruku	586 Razor ranch,Denton	Not Available	Angel@gmail.com	4695867899
5	Jose	Tabi	4000 N emil St,Denton	Available	Tati@gmail.com	4695862333
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Below the result grid is an 'Output' section showing the execution of the queries. The first query is an UPDATE statement that affected 1 row. The second query is a SELECT statement that returned 5 rows.

#	Time	Action	Message
1	20:06:20	UPDATE DrivingSchool.Instructor SET last_name="Joy" WHERE first_name="Pooja"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	20:06:20	Select * from DrivingSchool.Instructor LIMIT 0, 1000	5 row(s) returned

Update Request_Start_date in Request table:

```
UPDATE DrivingSchool.Request
```

```
SET request_Start_date='2022-12-20'
```

```
WHERE UniqueID=1;
```


Output:

The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

```
1 • UPDATE DrivingSchool.Request
2   SET request_Start_date='2022-12-20'
3   WHERE UniqueID=1;
4
5 • Select * from DrivingSchool.Request;
```

Below the query editor is a 'Result Grid' with the following columns: UniqueID, Service_ID, Customer_ID, Request_Start_date, Created_date, Vehicle_Type_ID, and Request_Status. The grid contains 5 rows of data:

UniqueID	Service_ID	Customer_ID	Request_Start_date	Created_date	Vehicle_Type_ID	Request_Status
1	3	1	2022-12-20	2022-12-15	5	active
2	3	2	2022-12-20	2022-12-15	1	active
3	4	3	2022-12-25	2022-12-15	3	active
4	1	4	2022-12-28	2022-12-15	5	active
5	2	5	2022-12-18	2022-12-15	5	active

Below the result grid is an 'Output' section with a table showing the execution of the queries:

#	Time	Action	Message
1	20:10:26	UPDATE DrivingSchool.Request SET request_Start_date='2022-12-20' WHERE UniqueID=1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	20:10:26	Select * from DrivingSchool.Request LIMIT 0, 1000	5 row(s) returned

Data Retrieval and Simple Reports:

1. Display the Lesson_Schedule_date for reservation made by the customer whose name is Manasaa

```
SELECT Lesson_Schedule_date FROM DrivingSchool.Reservation R
JOIN DrivingSchool.Request Re
ON R.Request_ID=Re.UniqueID
JOIN DrivingSchool.Customer C
ON Re.Customer_ID=C.Customer_ID
WHERE C.First_name='Manasaa';
```

Answer:

Lesson_Schedule_date
2022-12-19

Output:

query

```
1 • select Lesson_Schedule_date from DrivingSchool.Reservation R
2   join DrivingSchool.Request Re
3   on R.Request_ID=Re.UniqueID
4   join DrivingSchool.Customer C
5   on RE.Customer_ID=C.Customer_ID
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Lesson_Schedule_date
2022-12-19

Output

Action Output

#	Time	Action	Message
1	20:24:22	select Lesson_Schedule_date from DrivingSchool.Reservation R join DrivingSchool.Request Re on R.Request_...	1 row(s) returned

2. Display the Service Description of the Request with UniqueID 3

select Service_description from DrivingSchool.Service S

join DrivingSchool.Request Re

on S.Service_ID=Re.Service_ID

where Re.UniqueID=3;

Answer:

Service_Description
Adult 11X7 DrivingClasses

output:

query

```
1 • select Service_description from DrivingSchool.Service S
2   join DrivingSchool.Request Re
3   on S.Service_ID=Re.Service_ID
4   where Re.UniqueID=3;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Service_description
Adult 11X7 Driving classes

Result 30

Output

Action Output

#	Time	Action	Message
1	20:31:56	select Service_description from DrivingSchool.Service S join DrivingSchool.Request Re on S.Service_ID=Re.Se...	1 row(s) returned

3. Display the Lesson_Schedule_date made for services whose service_type is MINOR_SERVICES and sort them. Print Service_ID and Lesson_Schedule_date

```
SELECT R.Lesson_Schedule_date,S.Service_ID FROM DrivingSchool.Reservation R
JOIN DrivingSchool.Request Re
on R.Request_ID=Re.UniqueID
JOIN DrivingSchool.Service S
ON S.Service_ID=Re.Service_ID
WHERE S.Service_Type="MINOR SERVICE"
ORDER BY Lesson_schedule_date;
```

Answer:

Lesson_Schedule_Date	Service_ID
2022-12-18	2
2022-12-25	1

Output:

The screenshot shows a SQL query editor with a query window titled 'query*'. The query is as follows:

```
1 • SELECT R.Lesson_Schedule_date,S.Service_ID FROM DrivingSchool.Reservation R
2 JOIN DrivingSchool.Request Re
3 on R.Request_ID=Re.UniqueID
4 JOIN DrivingSchool.Service S
5 ON S.Service_ID=Re.Service_ID
6 WHERE S.Service_Type="MINOR SERVICE"
7 ORDER BY Lesson_schedule_date;
```

Below the query, there is a 'Result Grid' section with a table showing the results:

Lesson_Schedule_date	Service_ID
2022-12-18	2
2022-12-25	1

At the bottom, there is an 'Output' section with a table showing the execution details:

#	Time	Action	Message
1	20:44:56	SELECT R.Lesson_Schedule_date,S.Service_ID FROM DrivingSchool.Reservation R JOIN DrivingSchool.Request Re	2 row(s) returned

4. Display the Vehicle_Description of the Vehicle who was reserved with Reservation_ID 4

```
SELECT Vehicle_Description FROM DrivingSchool.Vehicle_Type V
JOIN DrivingSchool.Vehicle_Details Vd
on V.Vehicle_Type_ID=Vd.Vehicle_Type_ID
JOIN DrivingSchool.Reservation R
ON Vd.Vehicle_ID=R.Vehicle_ID
WHERE R.Reservation_ID=4;
```

Answer:

Vehicle_Description
SUV

Output:

The screenshot shows a SQL query editor window titled 'query'. The query is as follows:

```
1 • SELECT Vehicle_Description FROM DrivingSchool.Vehicle_Type V
2 JOIN DrivingSchool.Vehicle_Details Vd
3 on V.Vehicle_Type_ID=Vd.Vehicle_Type_ID
4 JOIN DrivingSchool.Reservation R
5 ON Vd.Vehicle_ID=R.Vehicle_ID
6 WHERE R.Reservation_ID=4;
7
```

Below the query, the 'Result Grid' shows the following data:

Vehicle_Description
SUV

The 'Output' section shows the execution details:

#	Time	Action	Message
1	21:19:42	SELECT Vehicle_Description FROM DrivingSchool.Vehicle_Type V JOIN DrivingSchool.Vehicle_Details Vd on ...	1 row(s) returned

5. Display the total Amount paid for Service_Lessons of each Service_Type for all the reservations done

```
SELECT SUM(Sl.Service_Cost) FROM DrivingSchool.Service_Lesson Sl
JOIN DrivingSchool.Service S
on S.Service_ID=Sl.Service_ID
JOIN DrivingSchool.Reservation R
ON Sl.Service_Lesson_ID=R.Service_Lesson_ID
GROUP BY S.Service_Type;
```

Answer:

SUM
2370
420

Output:

The screenshot shows a SQL query editor window titled 'query'. The query is as follows:

```
1 SELECT SUM(Sl.Service_Cost) FROM DrivingSchool.Service_Lesson Sl
2 JOIN DrivingSchool.Service S
3 on S.Service_ID=Sl.Service_ID
4 JOIN DrivingSchool.Reservation R
5 ON Sl.Service_Lesson_ID=R.Service_Lesson_ID
6 GROUP BY S.Service_Type;
```

Below the query, the 'Result Grid' tab is active, displaying the following results:

SUM(Sl.Service_Cost)
2370
420

At the bottom, the 'Output' tab shows the execution details:

#	Time	Action	Message
1	21:37:29	SELECT SUM(Sl.Service_Cost) FROM DrivingSchool.Service_Lesson Sl JOIN DrivingSchool.Service S on S.S...	2 row(s) returned

Conclusion:

Driving School Reservation system is modeled in a way to help the user and the administrative services of the driving school to enter and retrieve the data easily. This data model also helps to perform computation on the daily data generated by the reservation process. It improves the overall efficiency of the Driving School Reservation System.