MAHARISHI INTERNATIONAL UNIVERSITY

Subject: CS 422

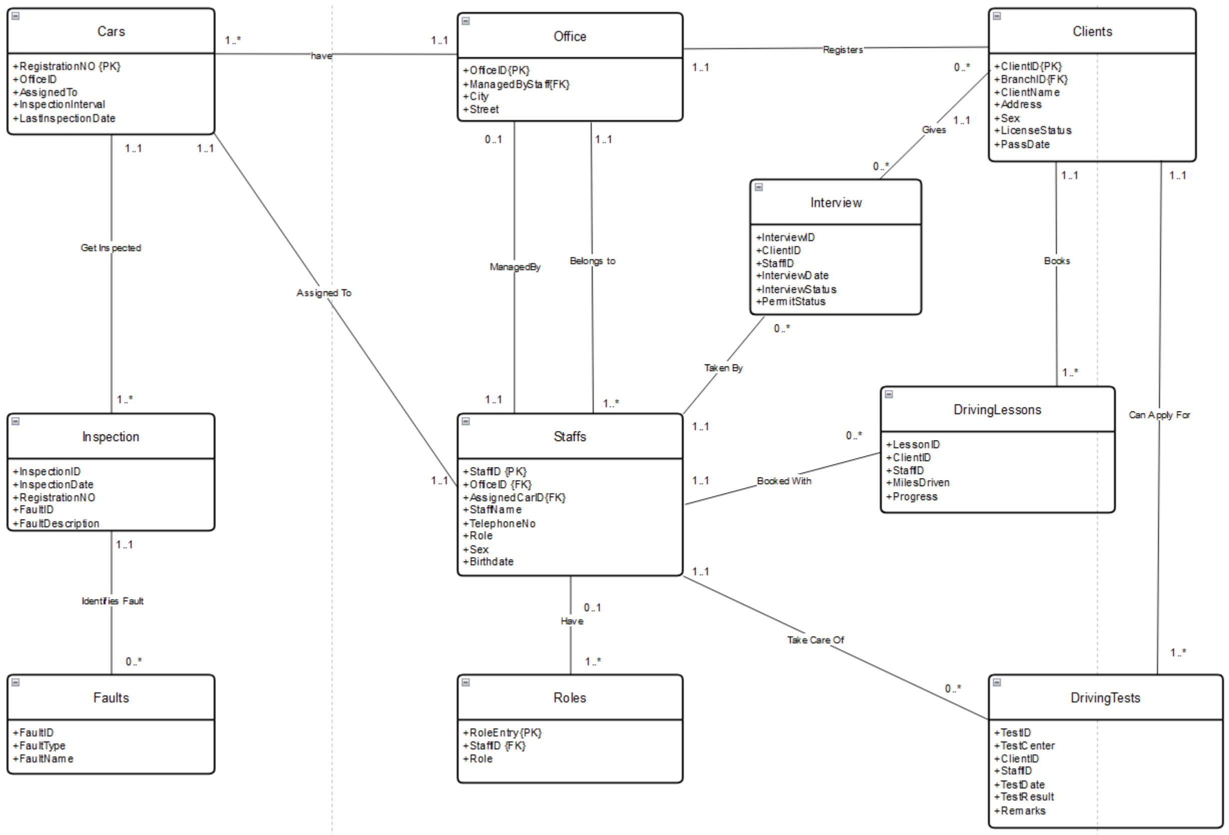
Assignment Topic: Project - Final Project Submission

Submitted by – Dip Ranjon Das (Student ID – 616979)

PART 1

Question 1: Draw the ER diagram for the EasyDrive School of Motoring that you find in paragraph B.2 in appendix B of the book.

Answer:



Question 2: Translate the ER diagram to tables (logical DB design).

Answer:

Office (OfficeID {PK}, ManagedbyStaff{FK}, City, Street)

Cars (RegistrationNo{PK}, OfficeID {FK}, AssignedTo {FK}, InspectionInterval, LastInspectionDate)

Staffs (StaffID {PK}, OfficeID {FK}, StaffName, TelephoneNo, Role, Sex, Birthdate)

Clients (ClientID {PK}, BranchID {FK},ClientName, Address, Sex, LicenseStatus, PassDate)

Inspection (InspectionID {PK},InspectionDate, RegistrationNo{FK}, FaultNo, FaultDescription)

Faults (FaultID, FaultType, FaultName)

Interview (InterviewID {PK}, ClientID {FK}, StaffID{FK}, InterviewDate, InterviewStatus, PermitStatus)

DrivingLessons (LessonID {PK}, ClientID {FK}, StaffID{FK}, MilesDriven, Progress)

DrivingTest (TestID {PK},TestCenter, ClientID{FK}, StaffID{FK}, TestDate, TestResult, Remarks)

Roles (RoleEntry{PK}, StaffID {FK}, DateFrom, role)

Question 3: Normalize the above tables to the highest normal form that you can.
Answer:

0NF Table	1NF Table	2NF Table	3NF Table
Inspection → FaultDescription is TD on FaultID		Clients → LicenseStatus is TD on PassDate Interview → InterviewStatus has dependency on InterviewDate and PermitStatus	Office Cars Staffs → Role is redundant Data Faults DrivingLessons DrivingTest Roles

3NF Clients Table:
1. Clients (ClientID {PK}, BranchID {FK},ClientName, Address, Sex, LicenseStatus, PassDate)

If PassDate is not NULL LicenseStatus is complete otherwise Null. We can determine LicenseStatus from PassDate therefore we can remove it from the table.

3NF Form: **Clients** (ClientID {PK}, BranchID {FK},ClientName, Address, Sex, PassDate)

2. Inspection (InspectionID {PK},InspectionDate, RegistrationNo{FK}, FaultNo, FaultDescription)

For every inspection there can be a list of Faults from the available faults data from faults table. For every fault there is a faultNo and description. Fault description depends on FaultNo and InspectionID therefore it is a partial dependency too. Therefore, we copy InspectionID and move FaultNo, FaultDescription from Inspection table and move to a new table name as IdentifiedFaults.

3NF Form:

Inspection (InspectionID {PK},InspectionDate, RegistrationNo{FK})
IdentifiedFaults ((InspectionID, FaultNo) {PK}, FaultDescription)

3. **Interview** (InterviewID {PK}, ClientID {FK}, StaffID{FK}, InterviewDate, InterviewStatus, PermitStatus)

According to defined problem InterviewStatus depends on valid permit status at the interview date. Therefore interview status is functional dependent on PermitStatus and InterviewDate. We can therefore move InterviewStatus, PermitStatus to a new table called PermitValidation.

3NF Form:
Interview (InterviewID {PK}, ClientID {FK}, StaffID{FK}, InterviewDate, PermitValidationID)
PermitValidation (PermitValidationID {PK}, InterviewStatus, PermitStatus)

4. Staffs (StaffID {PK}, OfficeID {FK}, StaffName, TelephoneNo, Role, Sex, Birthdate)
Role is a redundant data in Staff because we are keeping Role information is separate Role table. So, to ensure normalization we can move Role attribute from Staffs Table.

3NF Form:
Staffs (StaffID {PK}, OfficeID {FK}, StaffName, TelephoneNo, Sex, Birthdate)

So, our final normalized Tables are:

Office (OfficeID {PK}, ManagedbyStaff{FK}, City, Street)

Cars (RegistrationNo{PK}, OfficeID {FK}, AssignedTo {FK}, InspectionInterval, LastInspectionDate)

Faults (FaultID, FaultType, FaultName)

DrivingLessons (LessonID {PK}, ClientID {FK}, StaffID{FK}, MilesDriven, Progress)

DrivingTest (TestID {PK},TestCenter, ClientID{FK}, StaffID{FK}, TestDate, TestResult, Remarks)

Roles (RoleEntry{PK}, StaffID {FK}, DateFrom, role)

Clients (ClientID {PK}, BranchID {FK},ClientName, Address, Sex, PassDate)

Inspection (InspectionID {PK},InspectionDate, RegistrationNo{FK})

IdentifiedFaults ((InspectionID, FaultNo) {PK}, FaultDescription)

Interview (InterviewID {PK}, ClientID {FK}, StaffID{FK}, InterviewDate, PermitValidationID)

PermitValidation (PermitValidationID {PK}, InterviewStatus, PermitStatus)

Staffs (StaffID {PK}, OfficeID {FK}, StaffName, TelephoneNo, Sex, Birthdate)

Question 4: Input these tables into SQL-Server and put some data into them (Populate them).
Answer:

```
CREATE TABLE [dbo].[Cars](
        [RegistrationNo] [varchar](20) NOT NULL,
        [OfficeID] [int] NULL,
        [AssignedTo] [int] NULL,
        [InspectionInterval] [int] NOT NULL,
        [LastInspectionDate] [date] NOT NULL,
PRIMARY KEY CLUSTERED
(
        [RegistrationNo] ASC
)

ALTER TABLE [dbo].[Cars]  WITH CHECK ADD FOREIGN
KEY([AssignedTo])
REFERENCES [dbo].[Staffs] ([StaffID])

ALTER TABLE [dbo].[Cars]  WITH CHECK ADD FOREIGN
KEY([OfficeID])
REFERENCES [dbo].[Office] ([OfficeID])
```

```
CREATE TABLE [dbo].[Staffs](
        [StaffID] [int] NOT NULL,
        [StaffName] [varchar](50) NOT NULL,
        [TelephoneNo] [varchar](20) NOT NULL,
        [Sex] [char](1) NOT NULL,
        [Birthdate] [date] NOT NULL,
        [OfficeID] [int] NULL,
        [NoOfClients] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [StaffID] ASC
)
ALTER TABLE [dbo].[Staffs]  WITH CHECK ADD FOREIGN
KEY([OfficeID])
REFERENCES [dbo].[Office] ([OfficeID])
```

```
CREATE TABLE [dbo].[Clients](
        [ClientID] [int] NOT NULL,
        [OfficeID] [int] NOT NULL,
        [ClientName] [varchar](50) NOT NULL,
        [Address] [varchar](255) NOT NULL,
        [Sex] [char](1) NOT NULL,
        [PassDate] [date] NULL,
        [DateOfBirth] [date] NULL,
        [AssignedToStaff] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [ClientID] ASC
)
ALTER TABLE [dbo].[Clients]  WITH CHECK ADD
FOREIGN KEY([OfficeID])
REFERENCES [dbo].[Office] ([OfficeID])
```

```
CREATE TABLE [dbo].[Faults](
        [FaultID] [int] NOT NULL,
        [FaultType] [varchar](50) NOT NULL,
        [FaultName] [varchar](200) NOT NULL,
PRIMARY KEY CLUSTERED
(
        [FaultID] ASC
)
```

```
CREATE TABLE [dbo].[DrivingTest](
        [TestID] [int] IDENTITY(1,1) NOT NULL,
        [TestCenter] [varchar](50) NOT NULL,
        [ClientID] [int] NOT NULL,
        [StaffID] [int] NOT NULL,
        [TestDate] [date] NOT NULL,
        [TestResult] [varchar](10) NOT NULL,
        [Remarks] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
        [TestID] ASC
)
ALTER TABLE [dbo].[DrivingTest]  WITH CHECK ADD
FOREIGN KEY([ClientID])
REFERENCES [dbo].[Clients] ([ClientID])

ALTER TABLE [dbo].[DrivingTest]  WITH CHECK ADD
FOREIGN KEY([StaffID])
REFERENCES [dbo].[Staffs] ([StaffID])
```

```
CREATE TABLE [dbo].[DrivingLessons](
        [LessonID] [int] IDENTITY(1,1) NOT NULL,
        [ClientID] [int] NOT NULL,
        [StaffID] [int] NOT NULL,
        [MilesDriven] [int] NULL,
        [BunchID] [int] NULL,
        [LessonDateTime] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
        [LessonID] ASC
)
ALTER TABLE [dbo].[DrivingLessons]  WITH CHECK ADD
FOREIGN KEY([BunchID])
REFERENCES [dbo].[BunchList] ([BunchID])

ALTER TABLE [dbo].[DrivingLessons]  WITH CHECK ADD
FOREIGN KEY([ClientID])
REFERENCES [dbo].[Clients] ([ClientID])

ALTER TABLE [dbo].[DrivingLessons]  WITH CHECK ADD
FOREIGN KEY([StaffID])
REFERENCES [dbo].[Staffs] ([StaffID])
```

```
CREATE TABLE [dbo].[IdentifiedFault](
        [IdentifiedFaultID] [int] IDENTITY(1,1)
NOT NULL,
        [InspectionID] [int] NULL,
        [FaultID] [int] NOT NULL,
        [FaultDescription] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
        [IdentifiedFaultID] ASC
)
ALTER TABLE [dbo].[IdentifiedFault]  WITH CHECK
ADD FOREIGN KEY([FaultID])
REFERENCES [dbo].[Faults] ([FaultID])

ALTER TABLE [dbo].[IdentifiedFault]  WITH CHECK
ADD FOREIGN KEY([InspectionID])
REFERENCES [dbo].[Inspection] ([InspectionID])
```

```
CREATE TABLE [dbo].[Inspection](
        [InspectionID] [int] NOT NULL,
        [InspectionDate] [date] NOT NULL,
        [RegistrationNo] [varchar](20) NOT NULL,
PRIMARY KEY CLUSTERED
(
        [InspectionID] ASC
)
ALTER TABLE [dbo].[Inspection]  WITH CHECK ADD
FOREIGN KEY([RegistrationNo])
REFERENCES [dbo].[Cars] ([RegistrationNo])
```

```
CREATE TABLE [dbo].[Interview](
        [InterviewID] [int] IDENTITY(1,1) NOT
NULL,
        [ClientID] [int] NOT NULL,
        [StaffID] [int] NOT NULL,
        [InterviewDate] [date] NULL,
        [PermitValidationID] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
        [InterviewID] ASC
)
ALTER TABLE [dbo].[Interview]  WITH CHECK ADD
FOREIGN KEY([ClientID])
REFERENCES [dbo].[Clients] ([ClientID])

ALTER TABLE [dbo].[Interview]  WITH CHECK ADD
FOREIGN KEY([PermitValidationID])
REFERENCES [dbo].[PermitValidation]
([PermitValidationID])

ALTER TABLE [dbo].[Interview]  WITH CHECK ADD
FOREIGN KEY([StaffID])
REFERENCES [dbo].[Staffs] ([StaffID])
```

```
CREATE TABLE [dbo].[Office](
        [OfficeID] [int] IDENTITY(1,1) NOT NULL,
        [ManagedbyStaff] [int] NULL,
        [City] [varchar](50) NOT NULL,
        [Street] [varchar](100) NOT NULL,
PRIMARY KEY CLUSTERED
(
        [OfficeID] ASC
)
ALTER TABLE [dbo].[Office]  WITH CHECK ADD FOREIGN
KEY([ManagedbyStaff])
REFERENCES [dbo].[Staffs] ([StaffID])
```

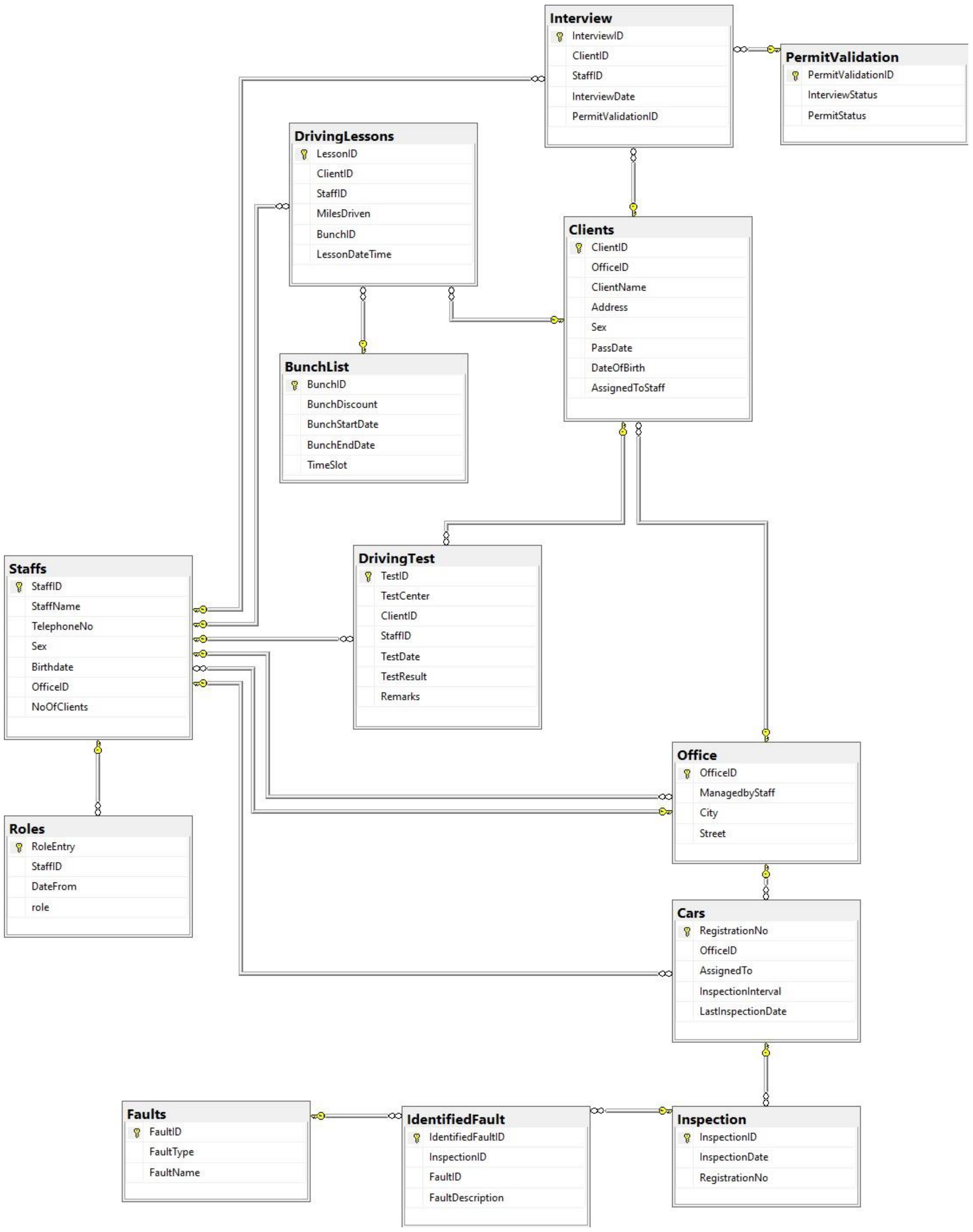
```
CREATE TABLE [dbo].[PermitValidation](
	[PermitValidationID] [int] NOT NULL,
	[InterviewStatus] [varchar](20) NULL,
	[PermitStatus] [varchar](20) NULL,
PRIMARY KEY CLUSTERED
(
	[PermitValidationID] ASC
)
ALTER TABLE [dbo].[PermitValidation] ADD
DEFAULT (NULL) FOR [InterviewStatus]

ALTER TABLE [dbo].[PermitValidation] ADD  DEFAULT
(NULL) FOR [PermitStatus]
```

```
CREATE TABLE [dbo].[Roles](
	[RoleEntry] [int] IDENTITY(1,1) NOT NULL,
	[StaffID] [int] NOT NULL,
	[DateFrom] [date] NOT NULL,
	[role] [varchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
	[RoleEntry] ASC
)
ALTER TABLE [dbo].[Roles]  WITH CHECK ADD FOREIGN
KEY([StaffID])
REFERENCES [dbo].[Staffs] ([StaffID])
```

```
CREATE TABLE [dbo].[BunchList](
	[BunchID] [int] NOT NULL,
	[BunchDiscount] [float] NULL,
	[BunchStartDate] [date] NULL,
	[BunchEndDate] [date] NULL,
	[TimeSlot] [varchar](20) NULL,
 CONSTRAINT [PK_BunchList] PRIMARY KEY CLUSTERED
(
	[BunchID] ASC
)
```

PART 2

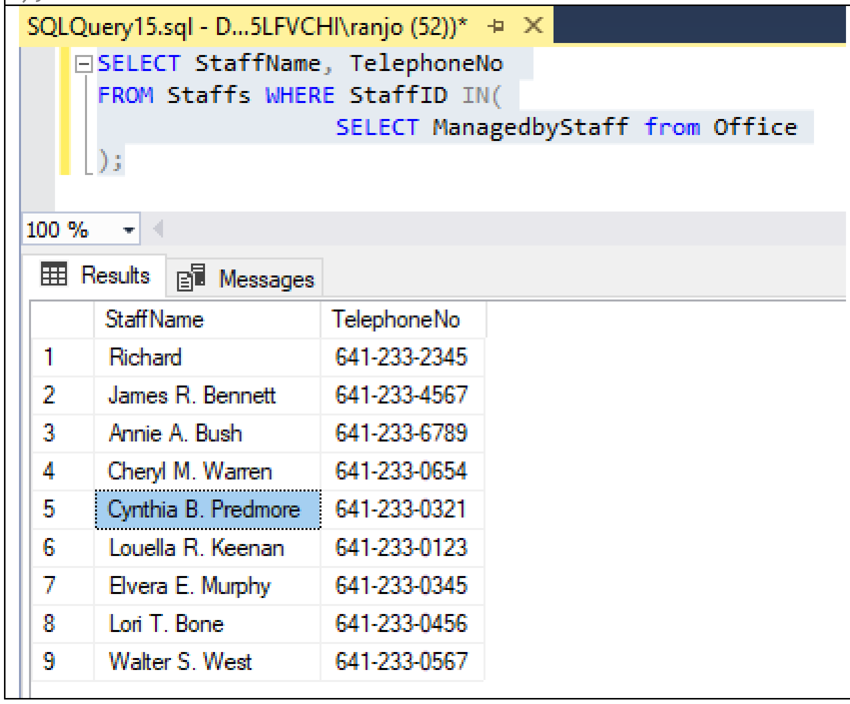
After studying all requirements from the customer and required queries we changed few tables and modified the ER diagram. The final ER diagram taken from MSSQL Server is below:



B.2.2 Query Transactions (Sample)

1. The names and the telephone numbers of the Managers of each office.
Answer:

```
SELECT StaffName, TelephoneNo
FROM Staffs WHERE StaffID IN(
                                                    SELECT ManagedbyStaff from Office
);
```



SQLQuery15.sql - D...5LFVCHI\ranjo (52))*

```
SELECT StaffName, TelephoneNo
FROM Staffs WHERE StaffID IN(
                              SELECT ManagedbyStaff from Office
);
```

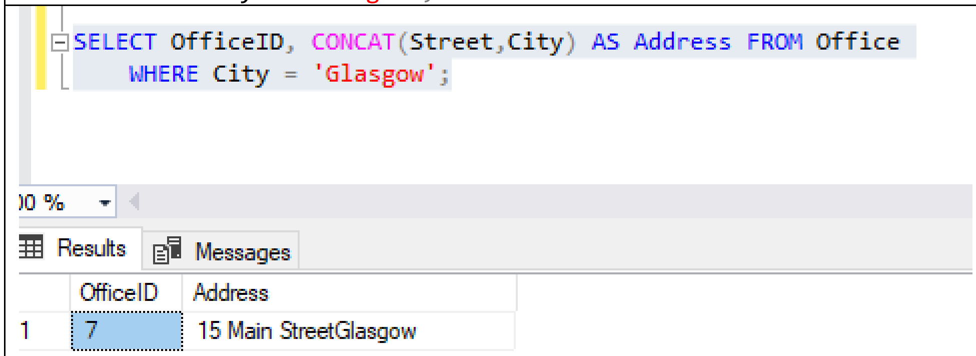
100 %

Results | Messages

	StaffName	TelephoneNo
1	Richard	641-233-2345
2	James R. Bennett	641-233-4567
3	Annie A. Bush	641-233-6789
4	Cheryl M. Warren	641-233-0654
5	Cynthia B. Predmore	641-233-0321
6	Louella R. Keenan	641-233-0123
7	Elvera E. Murphy	641-233-0345
8	Lori T. Bone	641-233-0456
9	Walter S. West	641-233-0567

2. The full address of all the Offices in Glasgow.
Answer:

```
SELECT OfficeID, CONCAT(Street,City) AS Address FROM Office
       WHERE City = 'Glasgow';
```



```
SELECT OfficeID, CONCAT(Street,City) AS Address FROM Office
    WHERE City = 'Glasgow';
```

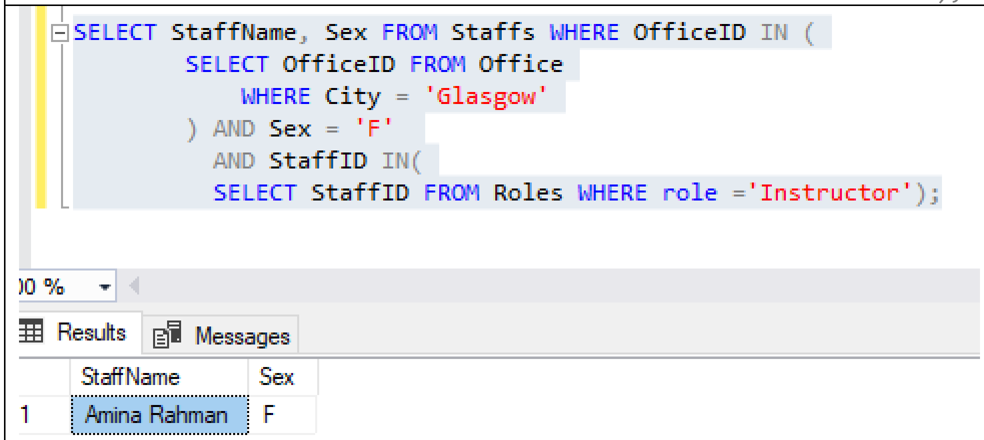
)0 %

Results | Messages

	OfficeID	Address
1	7	15 Main StreetGlasgow

3. The names of all female Instructors based in Glasgow, Bearsden office.
Answer:

```
SELECT StaffName, Sex FROM Staffs WHERE OfficeID IN (
              SELECT OfficeID FROM Office
                     WHERE City = 'Glasgow'
              ) AND Sex = 'F'
                AND StaffID IN(
                SELECT StaffID FROM Roles WHERE role ='Instructor');
```

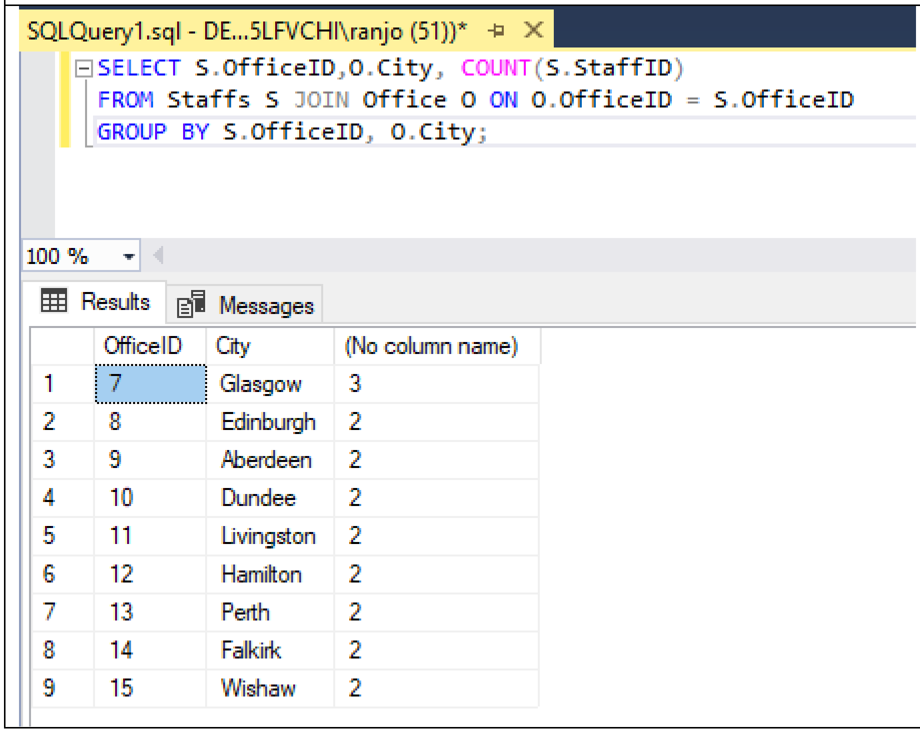


```
SELECT StaffName, Sex FROM Staffs WHERE OfficeID IN (
        SELECT OfficeID FROM Office
            WHERE City = 'Glasgow'
        ) AND Sex = 'F'
          AND StaffID IN(
          SELECT StaffID FROM Roles WHERE role ='Instructor');
```

```
   StaffName      Sex
1  Amina Rahman   F
```

4. The total number of staff at each office.
Answer:

```
SELECT S.OfficeID,O.City, COUNT(S.StaffID)
FROM Staffs S JOIN Office O ON O.OfficeID = S.OfficeID
GROUP BY S.OfficeID, O.City;
```



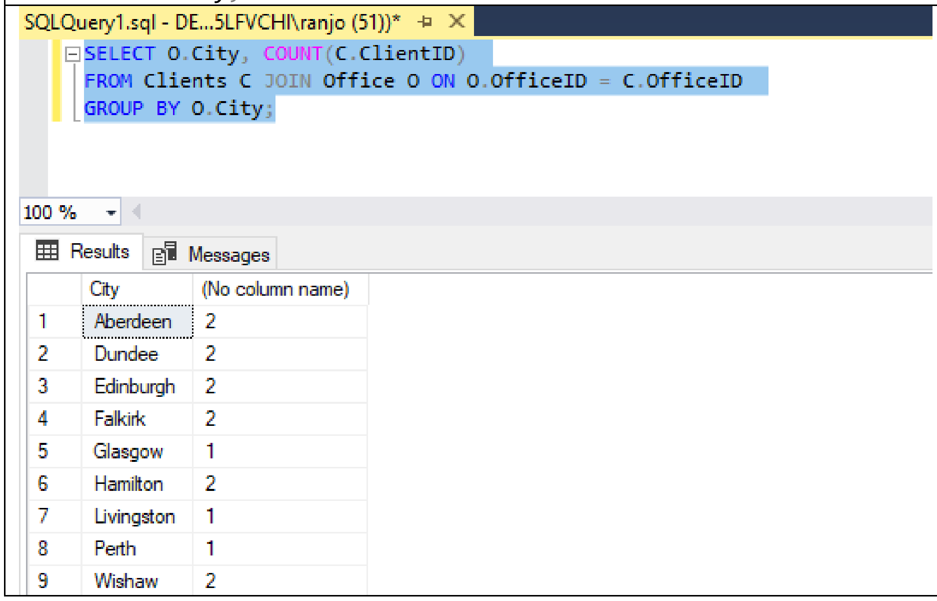
SQLQuery1.sql - DE...5LFVCHI\ranjo (51))*

```
SELECT S.OfficeID,O.City, COUNT(S.StaffID)
FROM Staffs S JOIN Office O ON O.OfficeID = S.OfficeID
GROUP BY S.OfficeID, O.City;
```

```
   OfficeID  City        (No column name)
1  7         Glasgow     3
2  8         Edinburgh   2
3  9         Aberdeen    2
4  10        Dundee      2
5  11        Livingston  2
6  12        Hamilton    2
7  13        Perth       2
8  14        Falkirk     2
9  15        Wishaw      2
```

5. The total number of clients (past and present) in each city.
Answer:

```
SELECT O.City, COUNT(C.ClientID)
FROM Clients C JOIN Office O ON O.OfficeID = C.OfficeID
GROUP BY O.City;
```



SQLQuery1.sql - DE...5LFVCHI\ranjo (51))*

```
SELECT O.City, COUNT(C.ClientID)
FROM Clients C JOIN Office O ON O.OfficeID = C.OfficeID
GROUP BY O.City;
```

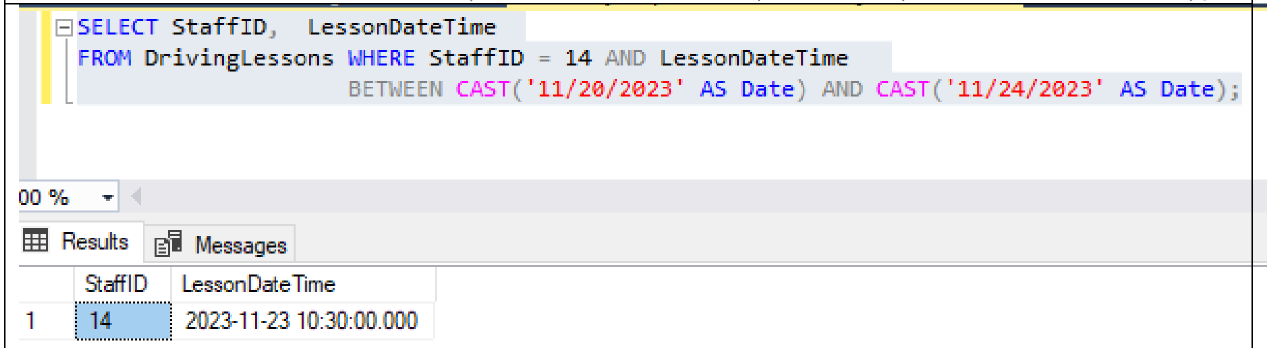
100 %

Results | Messages

	City	(No column name)
1	Aberdeen	2
2	Dundee	2
3	Edinburgh	2
4	Falkirk	2
5	Glasgow	1
6	Hamilton	2
7	Livingston	1
8	Perth	1
9	Wishaw	2

6. The timetable of appointments for a given Instructor next week.
Answer:

```
SELECT StaffID,  LessonDateTime
FROM DrivingLessons WHERE StaffID = 14 AND LessonDateTime
                    BETWEEN CAST('11/20/2023' AS Date) AND CAST('11/24/2023' AS Date);
```



```
SELECT StaffID,  LessonDateTime
FROM DrivingLessons WHERE StaffID = 14 AND LessonDateTime
                    BETWEEN CAST('11/20/2023' AS Date) AND CAST('11/24/2023' AS Date);
```

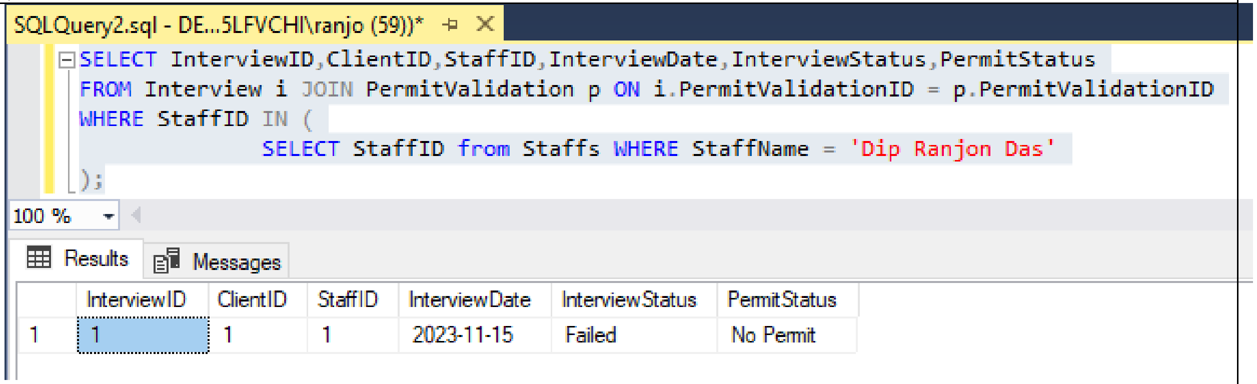
00 %

Results | Messages

	StaffID	LessonDateTime
1	14	2023-11-23 10:30:00.000

7. The details of Interviews conducted by a given Instructor.
Answer:

```
SELECT InterviewID,ClientID,StaffID,InterviewDate,InterviewStatus,PermitStatus
FROM Interview i JOIN PermitValidation p ON i.PermitValidationID = p.PermitValidationID
WHERE StaffID IN (
                       SELECT StaffID from Staffs WHERE StaffName = 'Dip Ranjon Das'
);
```



SQLQuery2.sql - DE...5LFVCHI\ranjo (59))*

```
SELECT InterviewID,ClientID,StaffID,InterviewDate,InterviewStatus,PermitStatus
FROM Interview i JOIN PermitValidation p ON i.PermitValidationID = p.PermitValidationID
WHERE StaffID IN (
                       SELECT StaffID from Staffs WHERE StaffName = 'Dip Ranjon Das'
);
```

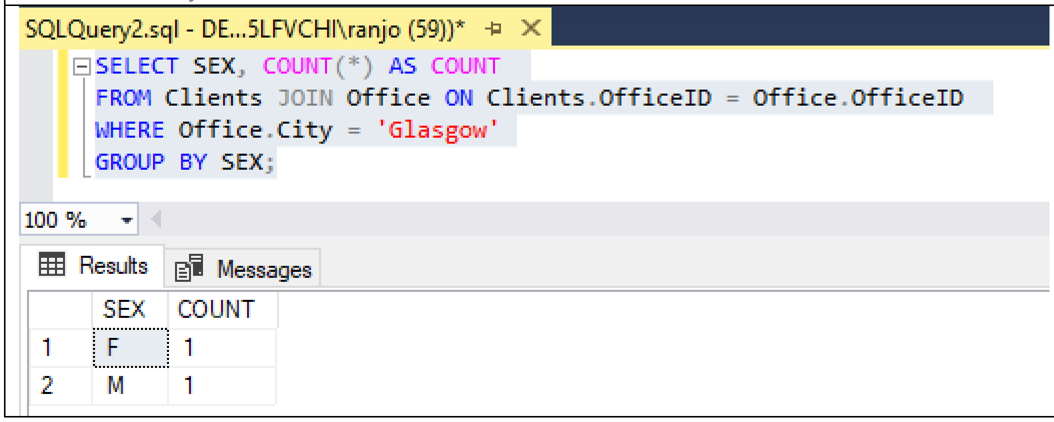
100 %

Results | Messages

	InterviewID	ClientID	StaffID	InterviewDate	InterviewStatus	PermitStatus
1	1	1	1	2023-11-15	Failed	No Permit

8. The total number of Female and male clients (past and present) in the Glasgow, Bearsden office.
Answer:

```
SELECT SEX, COUNT(*) AS COUNT
FROM Clients JOIN Office ON Clients.OfficeID = Office.OfficeID
WHERE Office.City = 'Glasgow'
GROUP BY SEX;
```



SQLQuery2.sql - DE...5LFVCHI\ranjo (59))*

```
SELECT SEX, COUNT(*) AS COUNT
FROM Clients JOIN Office ON Clients.OfficeID = Office.OfficeID
WHERE Office.City = 'Glasgow'
GROUP BY SEX;
```

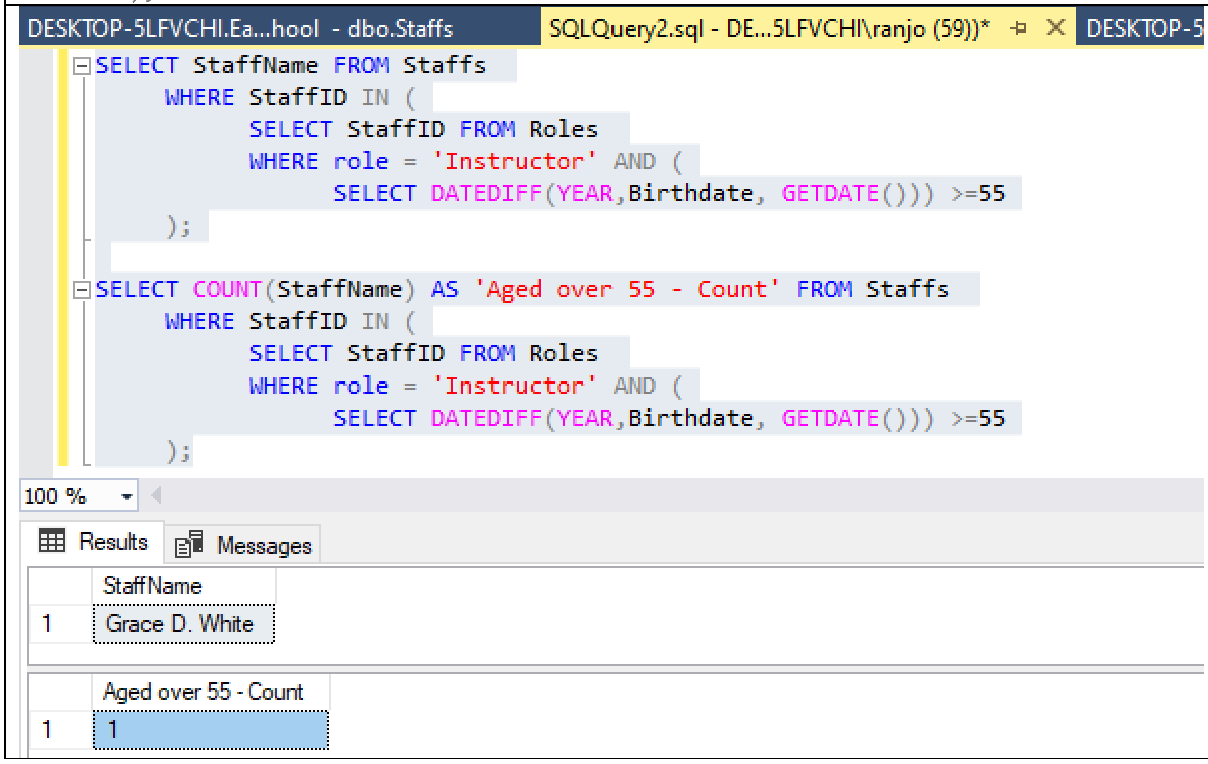
100 %

Results | Messages

	SEX	COUNT
1	F	1
2	M	1

9. The numbers and name of staffs who are Instructors and over 55 years old.
Answer:

```
SELECT StaffName FROM Staffs
        WHERE StaffID IN (
              SELECT StaffID FROM Roles
                 WHERE role = 'Instructor' AND (
                       SELECT DATEDIFF(YEAR,Birthdate, GETDATE())) >=55
        );

SELECT COUNT(StaffName) AS 'Aged over 55 - Count' FROM Staffs
        WHERE StaffID IN (
              SELECT StaffID FROM Roles
                 WHERE role = 'Instructor' AND (
                       SELECT DATEDIFF(YEAR,Birthdate, GETDATE())) >=55
        );
```


DESKTOP-5LFVCHI.Ea...hool - dbo.Staffs
SQLQuery2.sql - DE...5LFVCHI\ranjo (59))*
DESKTOP-5
100 %
Results
Messages

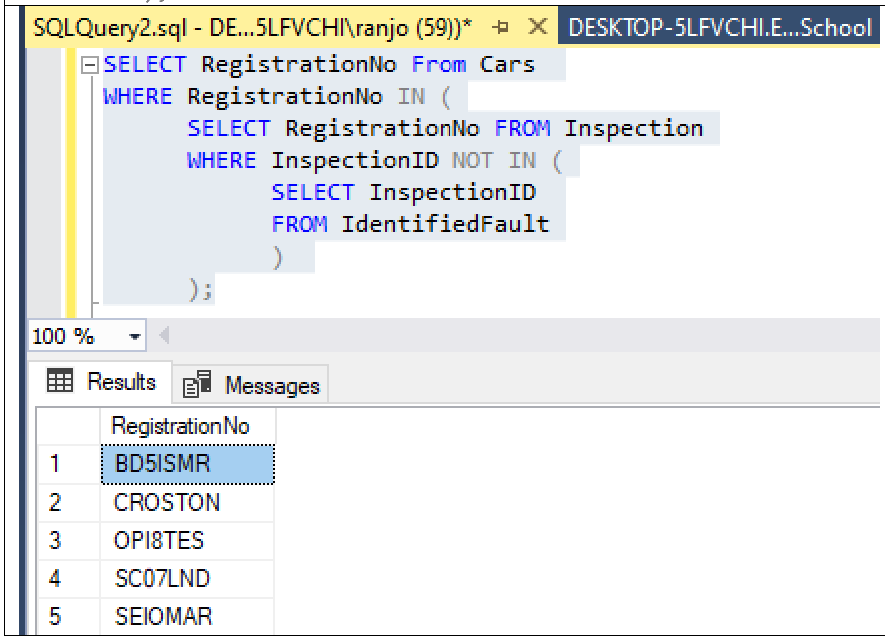
```
SELECT StaffName FROM Staffs
     WHERE StaffID IN (
           SELECT StaffID FROM Roles
           WHERE role = 'Instructor' AND (
                 SELECT DATEDIFF(YEAR,Birthdate, GETDATE())) >=55
     );

SELECT COUNT(StaffName) AS 'Aged over 55 - Count' FROM Staffs
     WHERE StaffID IN (
           SELECT StaffID FROM Roles
           WHERE role = 'Instructor' AND (
                 SELECT DATEDIFF(YEAR,Birthdate, GETDATE())) >=55
     );
```

	StaffName
1	Grace D. White

	Aged over 55 - Count
1	1

10. The registration number of cars that have had no faults found.
Answer:

```
SELECT RegistrationNo From Cars
WHERE RegistrationNo IN (
         SELECT RegistrationNo FROM Inspection
         WHERE InspectionID NOT IN (
               SELECT InspectionID
                     FROM IdentifiedFault
                     )
         );
```



SQLQuery2.sql - DE...5LFVCHI\ranjo (59))* DESKTOP-5LFVCHI.E...School

```
SELECT RegistrationNo From Cars
WHERE RegistrationNo IN (
      SELECT RegistrationNo FROM Inspection
      WHERE InspectionID NOT IN (
            SELECT InspectionID
            FROM IdentifiedFault
            )
      );
```

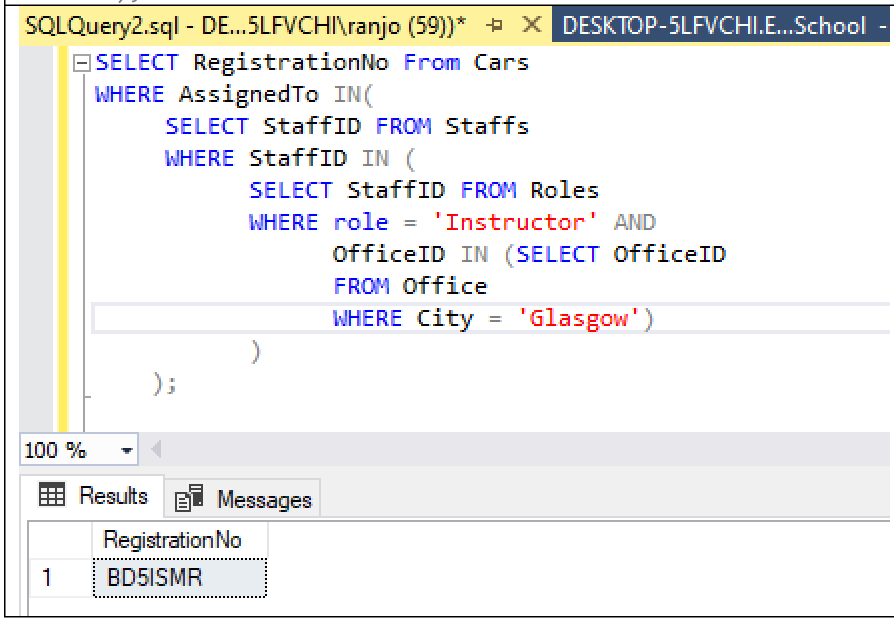
100 %

Results | Messages

	RegistrationNo
1	BD5ISMR
2	CROSTON
3	OPI8TES
4	SC07LND
5	SEIOMAR

11. The registration number of the cars used by Instructors at the Glasgow, Bearsden Office.
Answer:

```
SELECT RegistrationNo From Cars
WHERE AssignedTo IN(
     SELECT StaffID FROM Staffs
        WHERE StaffID IN (
              SELECT StaffID FROM Roles
                 WHERE role = 'Instructor' AND
                       OfficeID IN (SELECT OfficeID
                             FROM Office
                             WHERE City = 'Glasgow')
                 )
        );
```


SQLQuery2.sql - DE...5LFVCHI\ranjo (59))*   DESKTOP-5LFVCHI.E...School -

```
SELECT RegistrationNo From Cars
WHERE AssignedTo IN(
     SELECT StaffID FROM Staffs
     WHERE StaffID IN (
           SELECT StaffID FROM Roles
           WHERE role = 'Instructor' AND
                 OfficeID IN (SELECT OfficeID
                 FROM Office
                 WHERE City = 'Glasgow')
           )
     );
```

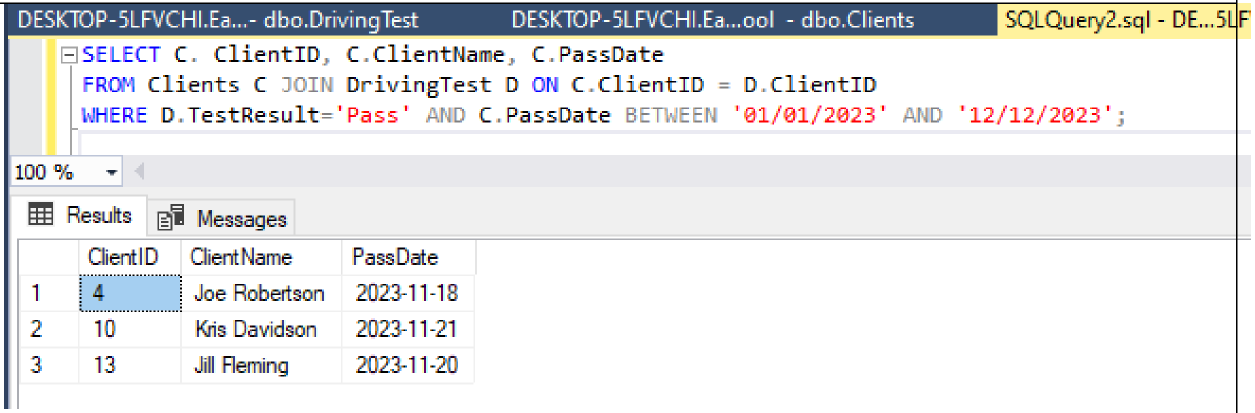
100 %

Results   Messages

	RegistrationNo
1	BD5ISMR

12. The names of the clients who passed the driving test in January 2013.

```
SELECT C. ClientID, C.ClientName, C.PassDate
FROM Clients C JOIN DrivingTest D ON C.ClientID = D.ClientID
WHERE D.TestResult='Pass' AND C.PassDate BETWEEN '01/01/2013' AND '01/31/2013';
NB: There is no data in this range in the database. Similar query below is given as a
demonstration for a year range data.

SELECT C. ClientID, C.ClientName, C.PassDate
FROM Clients C JOIN DrivingTest D ON C.ClientID = D.ClientID
WHERE D.TestResult='Pass' AND C.PassDate BETWEEN '01/01/2023' AND '12/12/2023';
```

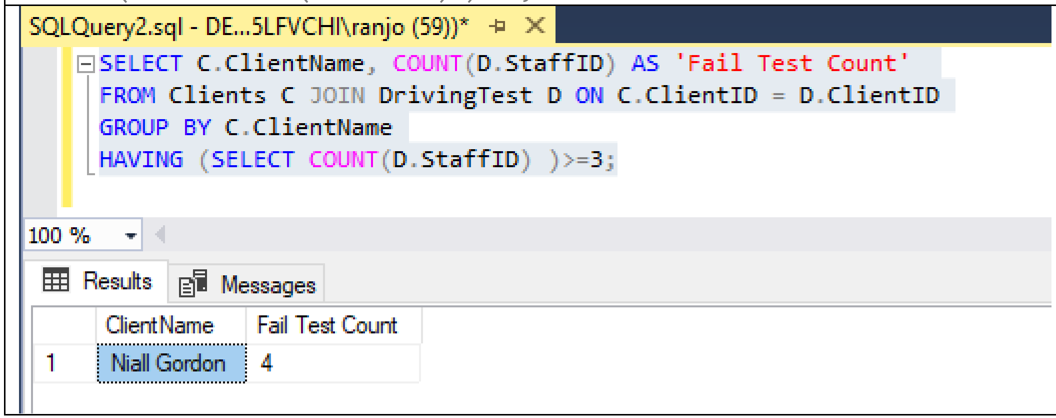

DESKTOP-5LFVCHI.Ea...- dbo.DrivingTest
DESKTOP-5LFVCHI.Ea...ool - dbo.Clients
SQLQuery2.sql - DE...5LF
100 %
Results
Messages

```
SELECT C. ClientID, C.ClientName, C.PassDate
FROM Clients C JOIN DrivingTest D ON C.ClientID = D.ClientID
WHERE D.TestResult='Pass' AND C.PassDate BETWEEN '01/01/2023' AND '12/12/2023';
```

	ClientID	ClientName	PassDate
1	4	Joe Robertson	2023-11-18
2	10	Kris Davidson	2023-11-21
3	13	Jill Fleming	2023-11-20

13. The names of clients who have sat the driving test more than three time and have still not passed.

```
SELECT C.ClientName, COUNT(D.StaffID) AS 'Fail Test Count'
FROM Clients C JOIN DrivingTest D ON C.ClientID = D.ClientID
GROUP BY C.ClientName
HAVING (SELECT COUNT(D.StaffID) )>=3;
```

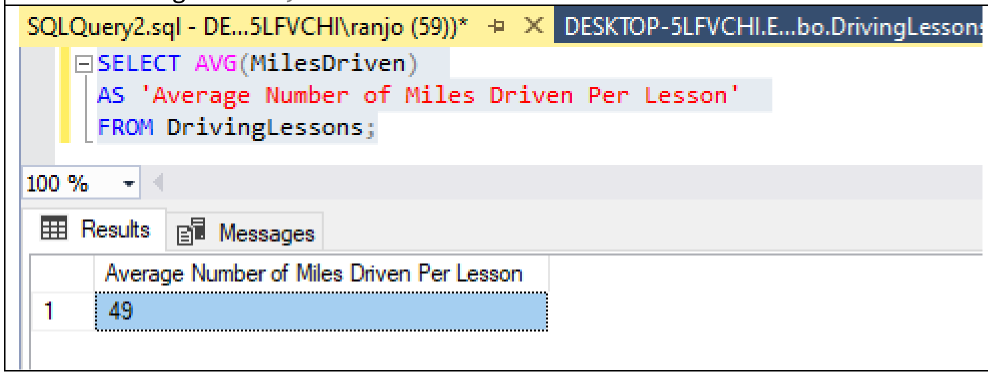


```
SELECT C.ClientName, COUNT(D.StaffID) AS 'Fail Test Count'
FROM Clients C JOIN DrivingTest D ON C.ClientID = D.ClientID
GROUP BY C.ClientName
HAVING (SELECT COUNT(D.StaffID) )>=3;
```

	ClientName	Fail Test Count
1	Niall Gordon	4

14. The average number of miles driven during a one-hour lesson.
Answer:

```
SELECT AVG(MilesDriven)
AS 'Average Number of Miles Driven Per Lesson'
FROM DrivingLessons;
```


SQLQuery2.sql - DE...5LFVCHI\ranjo (59))* DESKTOP-5LFVCHI.E...bo.DrivingLessons

```
SELECT AVG(MilesDriven)
AS 'Average Number of Miles Driven Per Lesson'
FROM DrivingLessons;
```

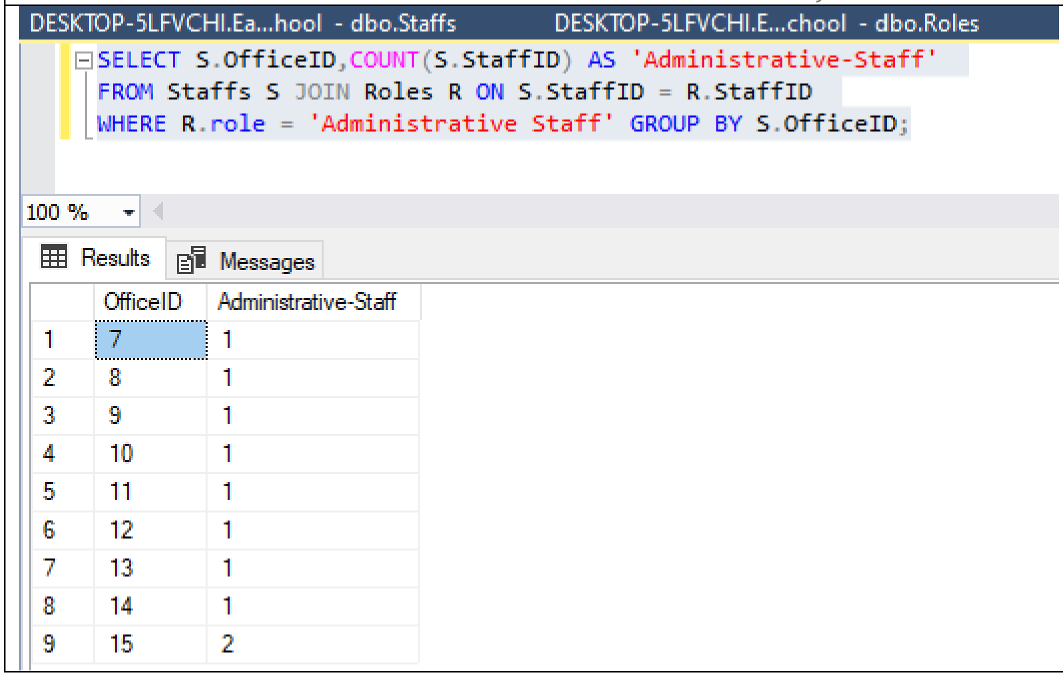
100 %

Results Messages

	Average Number of Miles Driven Per Lesson
1	49

15. The number of administrative staffs located at each office.
Answer:

```
SELECT S.OfficeID,COUNT(S.StaffID) AS 'Administrative-Staff'
FROM Staffs S JOIN Roles R ON S.StaffID = R.StaffID
WHERE R.role = 'Administrative Staff' GROUP BY S.OfficeID;
```


DESKTOP-5LFVCHI.Ea...hool - dbo.Staffs DESKTOP-5LFVCHI.E...chool - dbo.Roles

```
SELECT S.OfficeID,COUNT(S.StaffID) AS 'Administrative-Staff'
FROM Staffs S JOIN Roles R ON S.StaffID = R.StaffID
WHERE R.role = 'Administrative Staff' GROUP BY S.OfficeID;
```

100 %

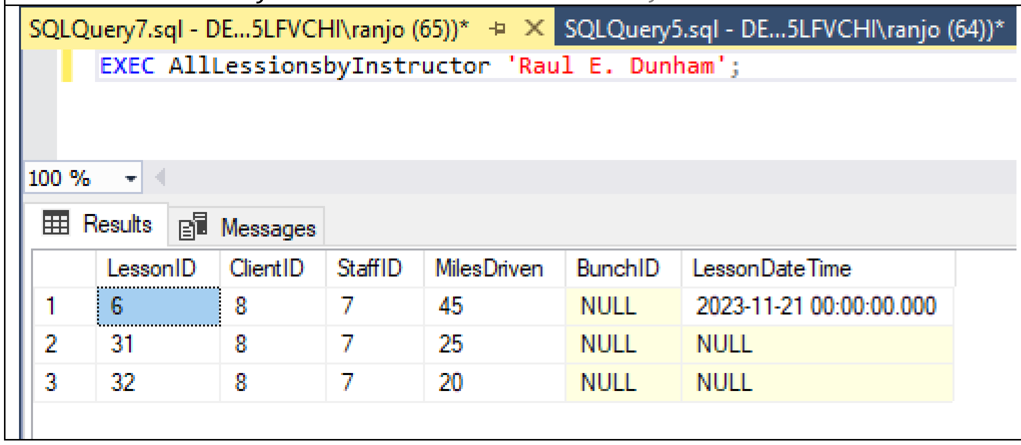
Results Messages

	OfficeID	Administrative-Staff
1	7	1
2	8	1
3	9	1
4	10	1
5	11	1
6	12	1
7	13	1
8	14	1
9	15	2

Part 2
Stored Procedures

Question 1: Write a stored procedure that takes in one argument, the staff number of an instructor. The procedure outputs all details of all the lessons for that instructor.
Answer:

```
USE [EasyDriveSchool ]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:     Provide instructors name and findout details of all lessons
-- =============================================
CREATE OR ALTER PROCEDURE [dbo].[AllLessionsbyInstructor]

       @InstructorName VARCHAR(20)

AS
BEGIN

       SET NOCOUNT ON;
       SELECT * FROM [dbo].[DrivingLessons]
       WHERE [StaffID] IN (
                                   SELECT [StaffID]
                                   FROM [dbo].[Staffs]
                                   WHERE [StaffName] = @InstructorName
                                   );
END
GO

EXEC AllLessionsbyInstructor 'Raul E. Dunham';
```



SQLQuery7.sql - DE...5LFVCHI\ranjo (65))*   SQLQuery5.sql - DE...5LFVCHI\ranjo (64))*

```
EXEC AllLessionsbyInstructor 'Raul E. Dunham';
```

100 %

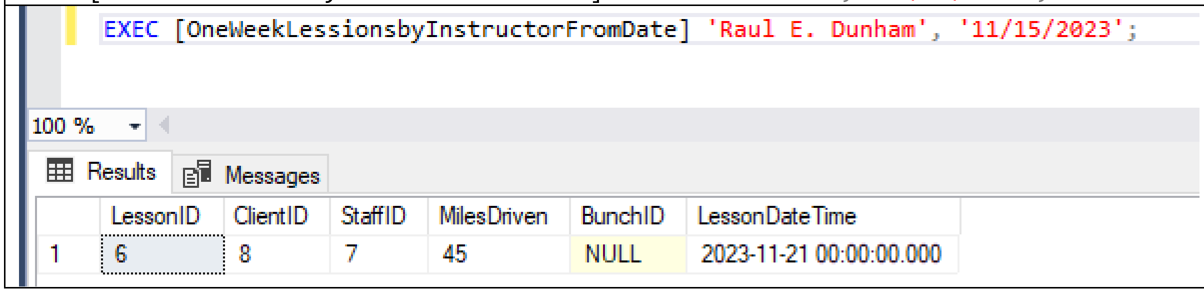
Results Messages

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime
1	6	8	7	45	NULL	2023-11-21 00:00:00.000
2	31	8	7	25	NULL	NULL
3	32	8	7	20	NULL	NULL

Question 2: Write a stored procedure that takes in two arguments, a staff number and a date. The procedure shows details of all lessons for that staff Instructor, starting at the date of the argument, and ending seven days later.

Answer:

```
USE [EasyDriveSchool ]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Dip Ranjon Das
-- Create date: 11/22/2023
-- Description: Provide instructors name and findout details of all lessons
-- =============================================
CREATE OR ALTER PROCEDURE [dbo].[OneWeekLessionsbyInstructorFromDate]

       @InstructorName VARCHAR(20),
       @DateFrom DATE
AS
BEGIN
       DECLARE @NextDate DATE
       SET @NextDate= DATEADD(DAY,7, CAST(@DateFrom AS DATE));
       SELECT * FROM [dbo].[DrivingLessons]
       WHERE [StaffID] IN (
                                   SELECT [StaffID] FROM [dbo].[Staffs]
                                   WHERE [StaffName] = @InstructorName)
                               AND CAST(LessonDateTime AS Date)
                                   BETWEEN @DateFrom AND @NextDate;

       SET NOCOUNT ON;

END
GO

EXEC [OneWeekLessionsbyInstructorFromDate] 'Raul E. Dunham', '11/15/2023';
```



```
EXEC [OneWeekLessionsbyInstructorFromDate] 'Raul E. Dunham', '11/15/2023';
```

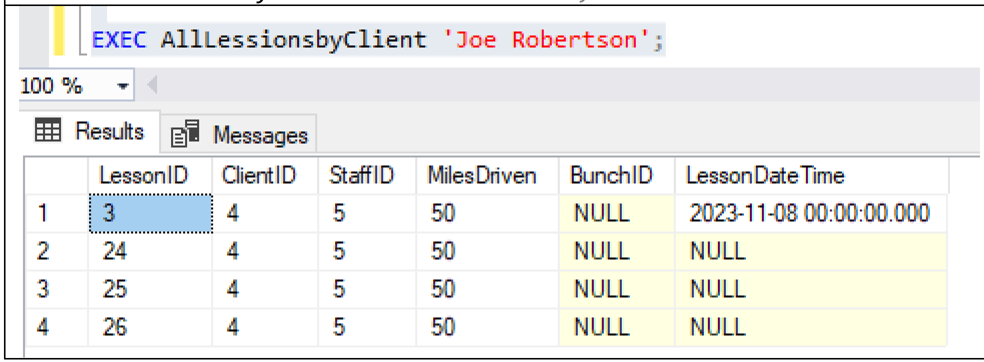
100 %

Results | Messages

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime
1	6	8	7	45	NULL	2023-11-21 00:00:00.000

Question 3: Do the same question 1 and 2 above, but for a client number instead of a staff number.
Answer:

```
USE [EasyDriveSchool ]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:          Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:     Provide Client name and findout details of all lessons
-- =============================================
CREATE OR ALTER   PROCEDURE [dbo].[AllLessionsbyClient]

       @ClientName VARCHAR(20)

AS
BEGIN

       SET NOCOUNT ON;

       SELECT * FROM [dbo].[DrivingLessons]
       WHERE [ClientID] IN (
                                   SELECT [ClientID]
                                   FROM [dbo].[Clients]
                                   WHERE [ClientName] = @ClientName
                                   );
END

EXEC AllLessionsbyClient 'Joe Robertson';
```



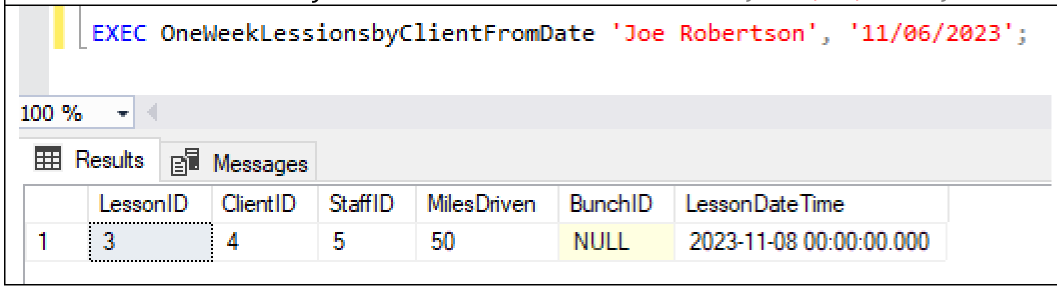
```
EXEC AllLessionsbyClient 'Joe Robertson';
```

100 %

Results | Messages

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime
1	3	4	5	50	NULL	2023-11-08 00:00:00.000
2	24	4	5	50	NULL	NULL
3	25	4	5	50	NULL	NULL
4	26	4	5	50	NULL	NULL

```sql
USE [EasyDriveSchool ]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:      Provide Client name and findout details of all lessons
-- =============================================
CREATE OR ALTER   PROCEDURE [dbo].[OneWeekLessionsbyClientFromDate]

       @ClientName VARCHAR(20),
       @DateFrom DATE

AS
BEGIN
       DECLARE @NextDate DATE
       SET @NextDate= DATEADD(DAY,7, CAST(@DateFrom AS DATE));
       SELECT * FROM [dbo].[DrivingLessons]
       WHERE [ClientID] IN (
                                   SELECT [ClientID]
                                   FROM [dbo].[Clients]
                                   WHERE [ClientName] = @ClientName)
                                   AND CAST(LessonDateTime AS Date)
                                       BETWEEN @DateFrom AND @NextDate;


       SET NOCOUNT ON;

END

EXEC OneWeekLessionsbyClientFromDate 'Joe Robertson', '11/06/2023';
```



```sql
EXEC OneWeekLessionsbyClientFromDate 'Joe Robertson', '11/06/2023';
```

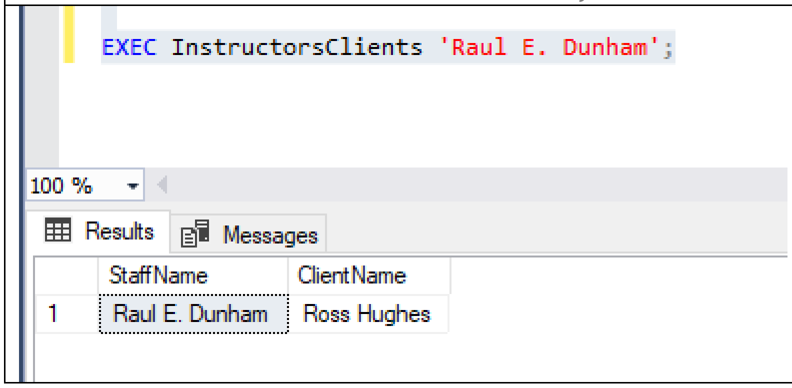
100 %

Results | Messages

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime
1	3	4	5	50	NULL	2023-11-08 00:00:00.000

Question 4: Create some stored procedures yourself which do something you would like to see being done.
Returning name of all their Clients for a given Instructor.
Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Dip Ranjon Das
-- Create date: 11/22/2023
-- Description: List of all Staffs and their Clients
-- =============================================
CREATE OR ALTER PROCEDURE [dbo].[InstructorsClients]
       @InstructorName VARCHAR(20)
AS
BEGIN
       SET NOCOUNT ON;
       SELECT  S.StaffName, C.ClientName FROM Staffs S JOIN DrivingLessons ON S.StaffID
= DrivingLessons.StaffID

        JOIN Clients C ON DrivingLessons.ClientID = C.ClientID
                                                         WHERE S.StaffName =
@InstructorName GROUP BY S.StaffName, C.ClientName;

END
GO

EXEC InstructorsClients 'Raul E. Dunham';
```



```
EXEC InstructorsClients 'Raul E. Dunham';
```

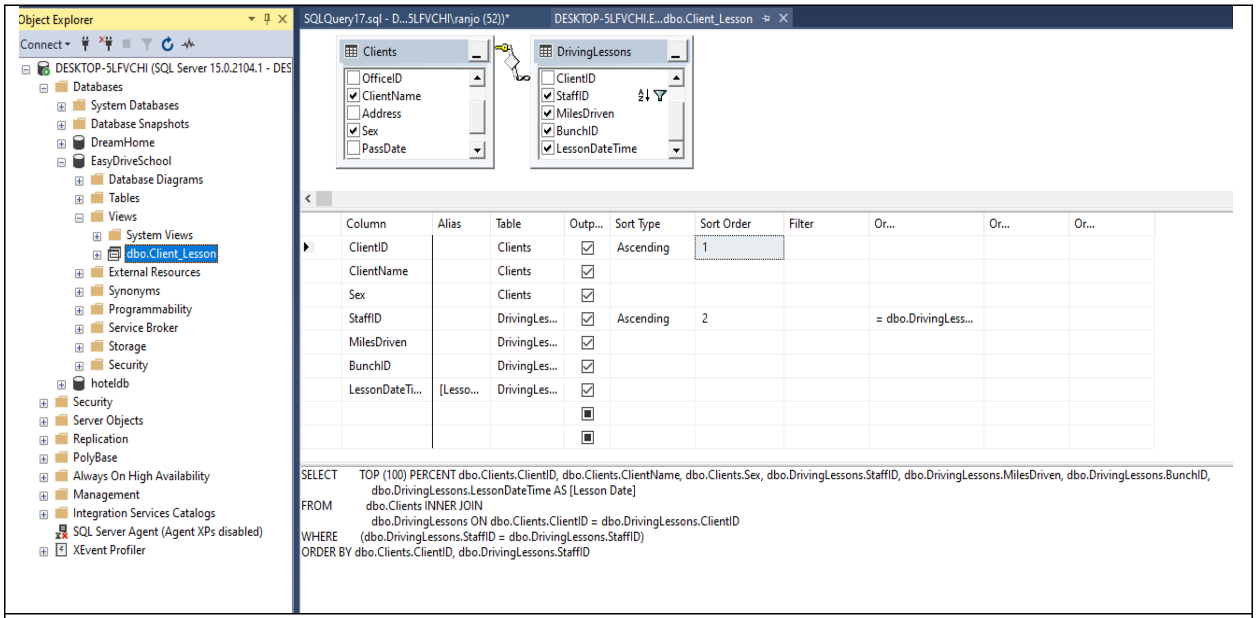
100 %

Results | Messages

	StaffName	ClientName
1	Raul E. Dunham	Ross Hughes

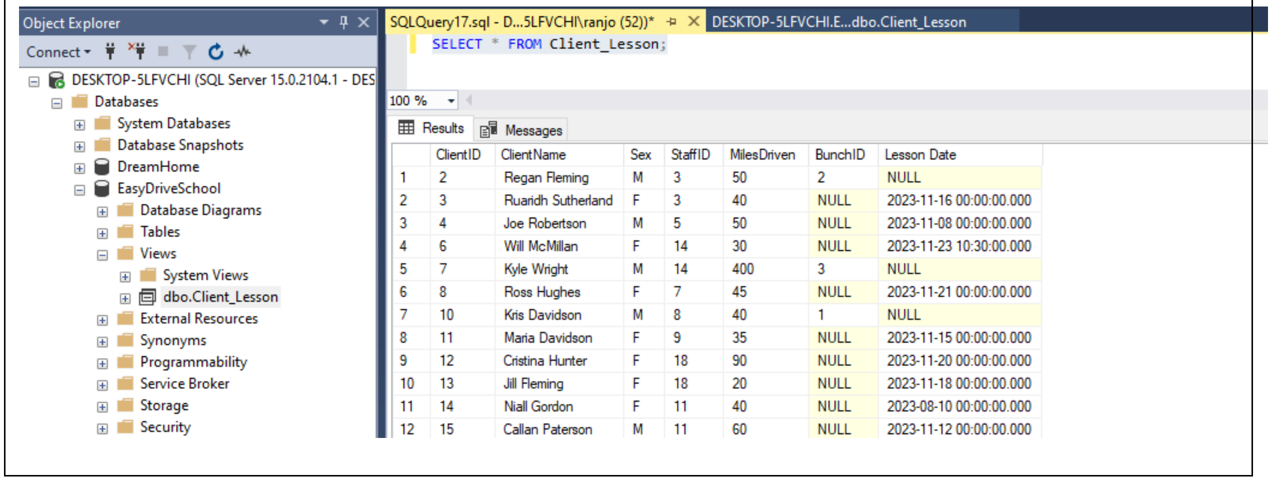
Views

Research how to make views in SQL server:
Question 5: Create a view called Client_Lesson which does an inner join on the Client and Lesson tables. Run it to make sure it works properly.
Answer:



	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▸	ClientID		Clients	✓	Ascending	1				
	ClientName		Clients	✓						
	Sex		Clients	✓						
	StaffID		DrivingLes...	✓	Ascending	2		= dbo.DrivingLess...		
	MilesDriven		DrivingLes...	✓						
	BunchID		DrivingLes...	✓						
	LessonDateTi...	[Lesso...	DrivingLes...	✓						
				■						
				■						

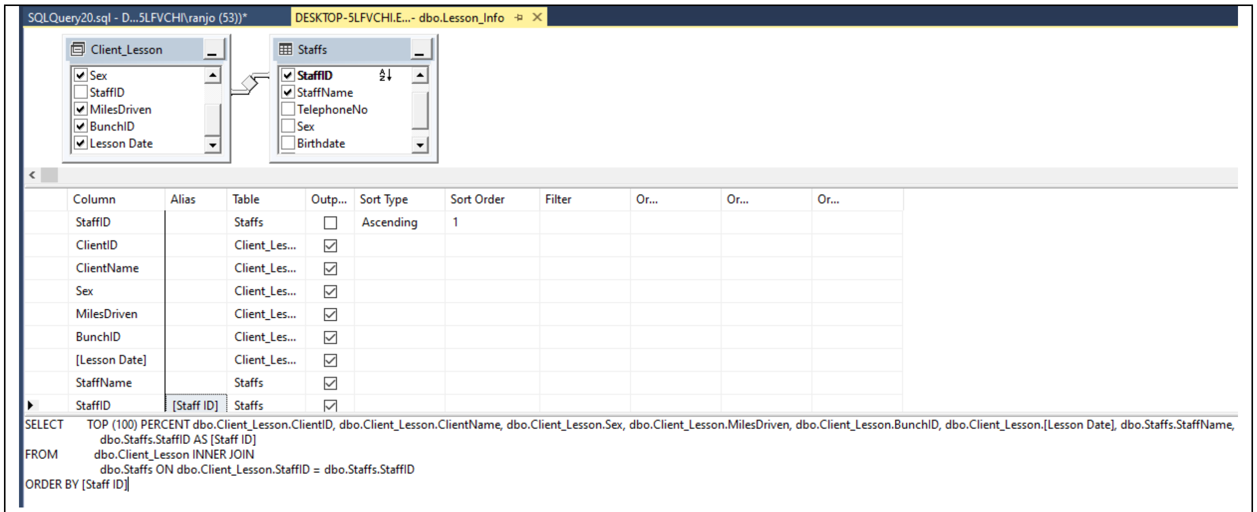
```
SELECT      TOP (100) PERCENT dbo.Clients.ClientID, dbo.Clients.ClientName, dbo.Clients.Sex, dbo.DrivingLessons.StaffID, dbo.DrivingLessons.MilesDriven, dbo.DrivingLessons.BunchID,
              dbo.DrivingLessons.LessonDateTime AS [Lesson Date]
FROM         dbo.Clients INNER JOIN
              dbo.DrivingLessons ON dbo.Clients.ClientID = dbo.DrivingLessons.ClientID
WHERE      (dbo.DrivingLessons.StaffID = dbo.DrivingLessons.StaffID)
ORDER BY dbo.Clients.ClientID, dbo.DrivingLessons.StaffID
```



```
SELECT * FROM Client_Lesson;
```

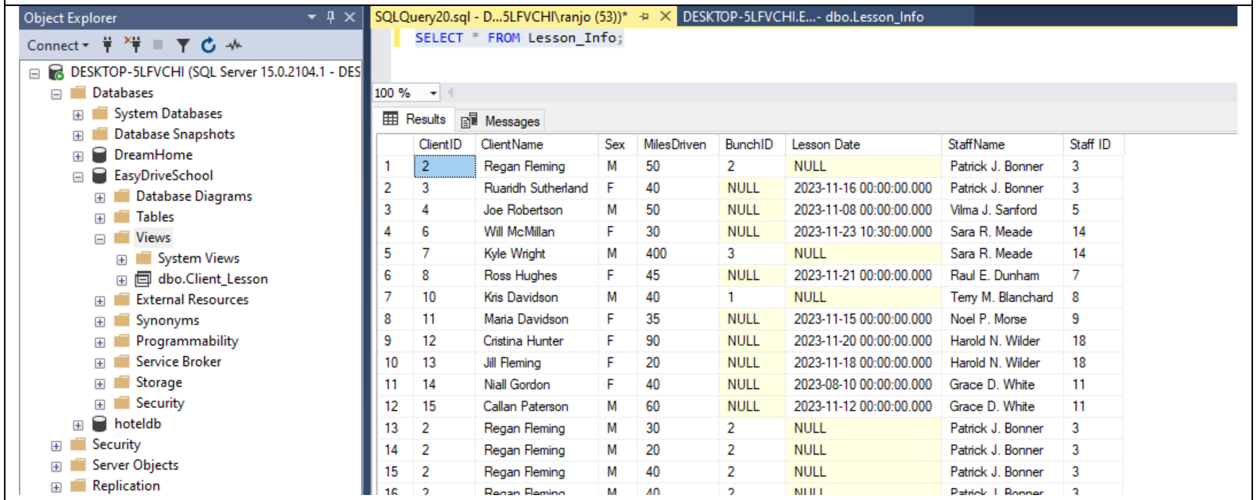
	ClientID	ClientName	Sex	StaffID	MilesDriven	BunchID	Lesson Date
1	2	Regan Fleming	M	3	50	2	NULL
2	3	Ruaridh Sutherland	F	3	40	NULL	2023-11-16 00:00:00.000
3	4	Joe Robertson	M	5	50	NULL	2023-11-08 00:00:00.000
4	6	Will McMillan	F	14	30	NULL	2023-11-23 10:30:00.000
5	7	Kyle Wright	M	14	400	3	NULL
6	8	Ross Hughes	F	7	45	NULL	2023-11-21 00:00:00.000
7	10	Kris Davidson	M	8	40	1	NULL
8	11	Maria Davidson	F	9	35	NULL	2023-11-15 00:00:00.000
9	12	Cristina Hunter	F	18	90	NULL	2023-11-20 00:00:00.000
10	13	Jill Fleming	F	18	20	NULL	2023-11-18 00:00:00.000
11	14	Niall Gordon	F	11	40	NULL	2023-08-10 00:00:00.000
12	15	Callan Paterson	M	11	60	NULL	2023-11-12 00:00:00.000

Question 6: Create a View called Lesson_Info which calls the View above Client_Lesson, and outputs all the information from Client_Lesson, along with who the staff person is for the lesson i.e. the staff person's name and staffID.
One, View can call other view that makes things flexible.
Answer:



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
StaffID		Staffs	☐	Ascending	1				
ClientID		Client_Les...	✓						
ClientName		Client_Les...	✓						
Sex		Client_Les...	✓						
MilesDriven		Client_Les...	✓						
BunchID		Client_Les...	✓						
[Lesson Date]		Client_Les...	✓						
StaffName		Staffs	✓						
StaffID	[Staff ID]	Staffs	✓						

```
SELECT     TOP (100) PERCENT dbo.Client_Lesson.ClientID, dbo.Client_Lesson.ClientName, dbo.Client_Lesson.Sex, dbo.Client_Lesson.MilesDriven, dbo.Client_Lesson.BunchID, dbo.Client_Lesson.[Lesson Date], dbo.Staffs.StaffName,
             dbo.Staffs.StaffID AS [Staff ID]
FROM       dbo.Client_Lesson INNER JOIN
             dbo.Staffs ON dbo.Client_Lesson.StaffID = dbo.Staffs.StaffID
ORDER BY [Staff ID]
```



```sql
SELECT * FROM Lesson_Info;
```

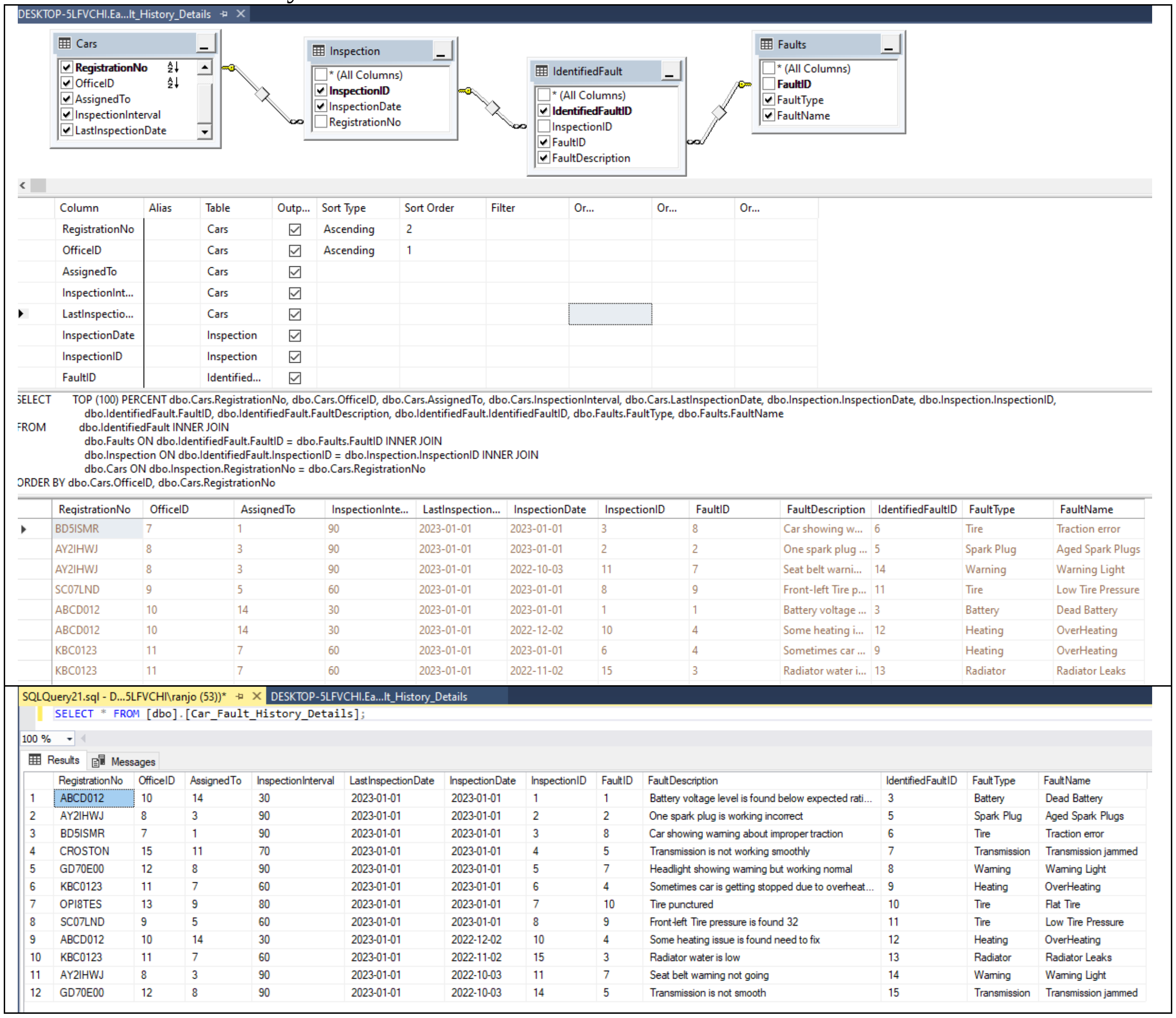
	ClientID	ClientName	Sex	MilesDriven	BunchID	Lesson Date	StaffName	Staff ID
1	2	Regan Fleming	M	50	2	NULL	Patrick J. Bonner	3
2	3	Ruaridh Sutherland	F	40	NULL	2023-11-16 00:00:00.000	Patrick J. Bonner	3
3	4	Joe Robertson	M	50	NULL	2023-11-08 00:00:00.000	Vilma J. Sanford	5
4	6	Will McMillan	F	30	NULL	2023-11-23 10:30:00.000	Sara R. Meade	14
5	7	Kyle Wright	M	400	3	NULL	Sara R. Meade	14
6	8	Ross Hughes	F	45	NULL	2023-11-21 00:00:00.000	Raul E. Dunham	7
7	10	Kris Davidson	M	40	1	NULL	Terry M. Blanchard	8
8	11	Maria Davidson	F	35	NULL	2023-11-15 00:00:00.000	Noel P. Morse	9
9	12	Cristina Hunter	F	90	NULL	2023-11-20 00:00:00.000	Harold N. Wilder	18
10	13	Jill Fleming	F	20	NULL	2023-11-18 00:00:00.000	Harold N. Wilder	18
11	14	Niall Gordon	F	40	NULL	2023-08-10 00:00:00.000	Grace D. White	11
12	15	Callan Paterson	M	60	NULL	2023-11-12 00:00:00.000	Grace D. White	11
13	2	Regan Fleming	M	30	2	NULL	Patrick J. Bonner	3
14	2	Regan Fleming	M	20	2	NULL	Patrick J. Bonner	3
15	2	Regan Fleming	M	40	2	NULL	Patrick J. Bonner	3
16	2	Regan Fleming	M	40	2	NULL	Patrick J. Bonner	3

Question 7: Create two more views that may be useful to you. Test them!

Answer:

Two more views can be:

1. Cars fault history details.



DESKTOP-5LFVCHI.Ea...lt_History_Details

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
RegistrationNo		Cars	✓	Ascending	2				
OfficeID		Cars	✓	Ascending	1				
AssignedTo		Cars	✓						
InspectionInt...		Cars	✓						
LastInspectio...		Cars	✓						
InspectionDate		Inspection	✓						
InspectionID		Inspection	✓						
FaultID		Identified...	✓						

```
SELECT     TOP (100) PERCENT dbo.Cars.RegistrationNo, dbo.Cars.OfficeID, dbo.Cars.AssignedTo, dbo.Cars.InspectionInterval, dbo.Cars.LastInspectionDate, dbo.Inspection.InspectionDate, dbo.Inspection.InspectionID,
             dbo.IdentifiedFault.FaultID, dbo.IdentifiedFault.FaultDescription, dbo.IdentifiedFault.IdentifiedFaultID, dbo.Faults.FaultType, dbo.Faults.FaultName
FROM        dbo.IdentifiedFault INNER JOIN
             dbo.Faults ON dbo.IdentifiedFault.FaultID = dbo.Faults.FaultID INNER JOIN
             dbo.Inspection ON dbo.IdentifiedFault.InspectionID = dbo.Inspection.InspectionID INNER JOIN
             dbo.Cars ON dbo.Inspection.RegistrationNo = dbo.Cars.RegistrationNo
ORDER BY dbo.Cars.OfficeID, dbo.Cars.RegistrationNo
```

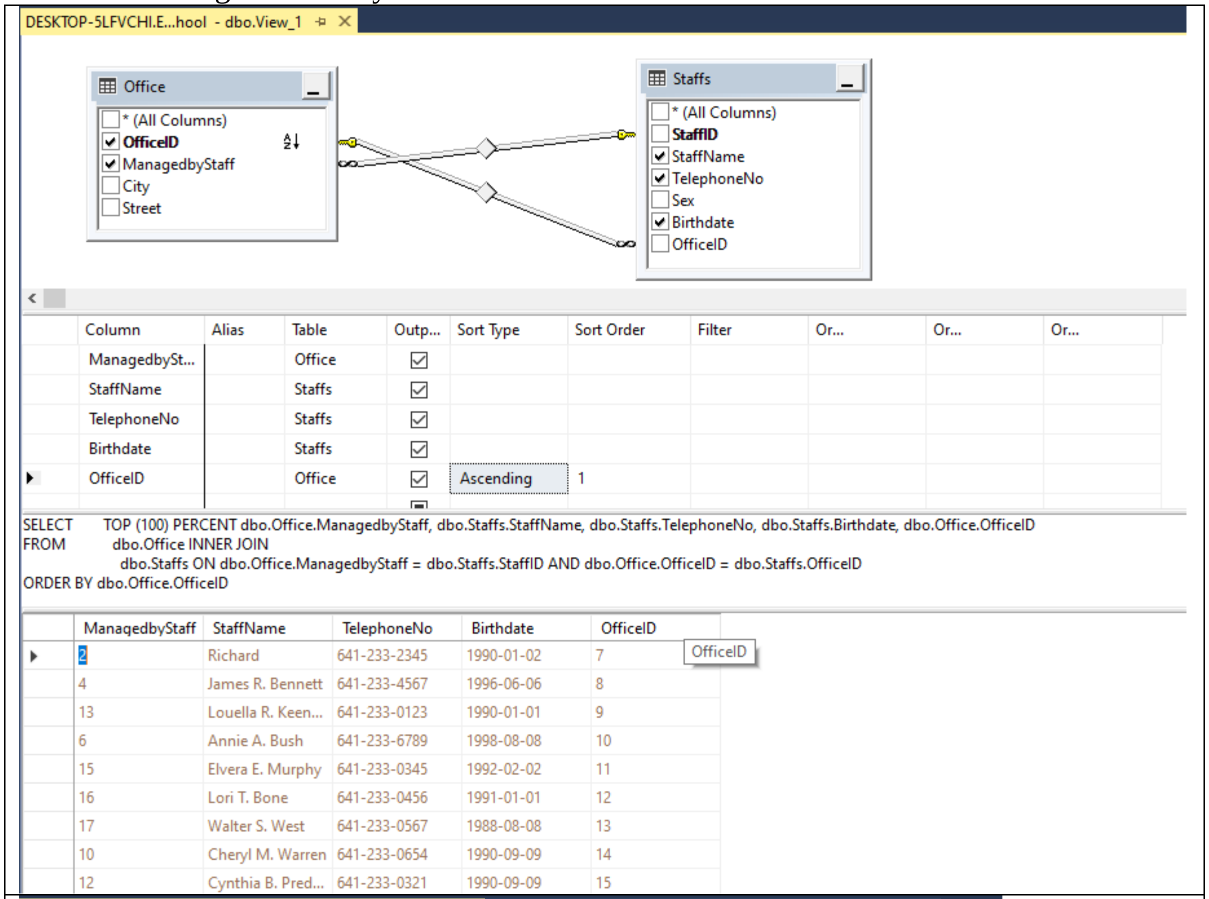
RegistrationNo	OfficeID	AssignedTo	InspectionInte...	LastInspection...	InspectionDate	InspectionID	FaultID	FaultDescription	IdentifiedFaultID	FaultType	FaultName
BD5ISMR	7	1	90	2023-01-01	2023-01-01	3	8	Car showing w...	6	Tire	Traction error
AY2IHWJ	8	3	90	2023-01-01	2023-01-01	2	2	One spark plug ...	5	Spark Plug	Aged Spark Plugs
AY2IHWJ	8	3	90	2023-01-01	2022-10-03	11	7	Seat belt warni...	14	Warning	Warning Light
SC07LND	9	5	60	2023-01-01	2023-01-01	8	9	Front-left Tire p...	11	Tire	Low Tire Pressure
ABCD012	10	14	30	2023-01-01	2023-01-01	1	1	Battery voltage ...	3	Battery	Dead Battery
ABCD012	10	14	30	2023-01-01	2022-12-02	10	4	Some heating i...	12	Heating	OverHeating
KBC0123	11	7	60	2023-01-01	2023-01-01	6	4	Sometimes car ...	9	Heating	OverHeating
KBC0123	11	7	60	2023-01-01	2022-11-02	15	3	Radiator water i...	13	Radiator	Radiator Leaks

SQLQuery21.sql - D...5LFVCHI\ranjo (53))* DESKTOP-5LFVCHI.Ea...lt_History_Details

```sql
SELECT * FROM [dbo].[Car_Fault_History_Details];
```

	RegistrationNo	OfficeID	AssignedTo	InspectionInterval	LastInspectionDate	InspectionDate	InspectionID	FaultID	FaultDescription	IdentifiedFaultID	FaultType	FaultName
1	ABCD012	10	14	30	2023-01-01	2023-01-01	1	1	Battery voltage level is found below expected rati...	3	Battery	Dead Battery
2	AY2IHWJ	8	3	90	2023-01-01	2023-01-01	2	2	One spark plug is working incorrect	5	Spark Plug	Aged Spark Plugs
3	BD5ISMR	7	1	90	2023-01-01	2023-01-01	3	8	Car showing warning about improper traction	6	Tire	Traction error
4	CROSTON	15	11	70	2023-01-01	2023-01-01	4	5	Transmission is not working smoothly	7	Transmission	Transmission jammed
5	GD70E00	12	8	90	2023-01-01	2023-01-01	5	7	Headlight showing warning but working normal	8	Warning	Warning Light
6	KBC0123	11	7	60	2023-01-01	2023-01-01	6	4	Sometimes car is getting stopped due to overheat...	9	Heating	OverHeating
7	OPI8TES	13	9	80	2023-01-01	2023-01-01	7	10	Tire punctured	10	Tire	Flat Tire
8	SC07LND	9	5	60	2023-01-01	2023-01-01	8	9	Front-left Tire pressure is found 32	11	Tire	Low Tire Pressure
9	ABCD012	10	14	30	2023-01-01	2022-12-02	10	4	Some heating issue is found need to fix	12	Heating	OverHeating
10	KBC0123	11	7	60	2023-01-01	2022-11-02	15	3	Radiator water is low	13	Radiator	Radiator Leaks
11	AY2IHWJ	8	3	90	2023-01-01	2022-10-03	11	7	Seat belt warning not going	14	Warning	Warning Light
12	GD70E00	12	8	90	2023-01-01	2022-10-03	14	5	Transmission is not smooth	15	Transmission	Transmission jammed

2. Office Managers Birthday.


DESKTOP-5LFVCHI.E...hool - dbo.View_1

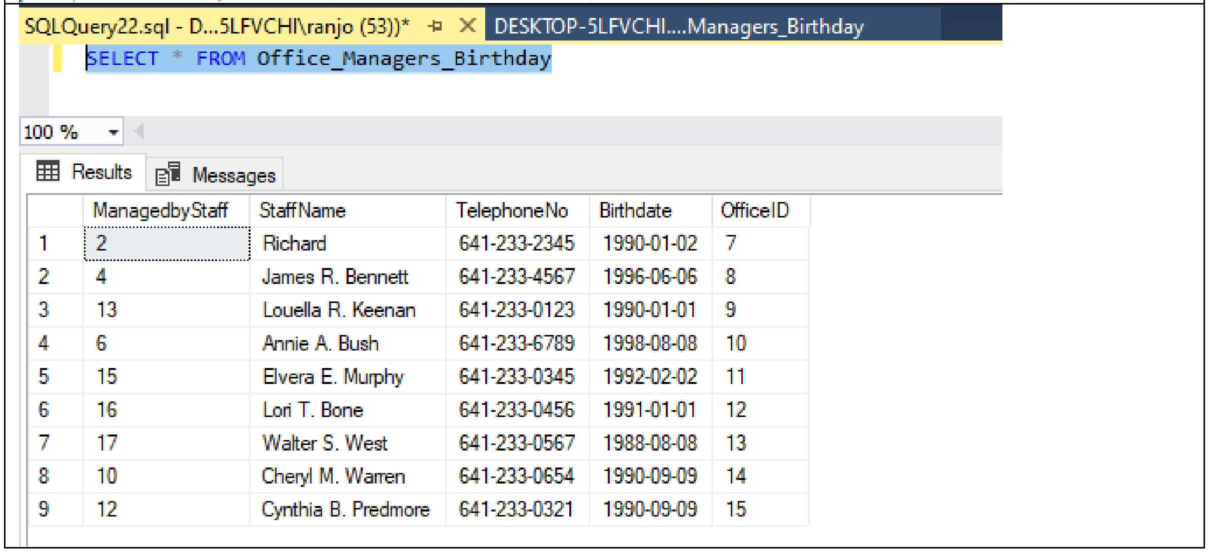
Office
- [ ] * (All Columns)
- [✓] **OfficeID**
- [✓] ManagedbyStaff
- [ ] City
- [ ] Street

Staffs
- [ ] * (All Columns)
- [ ] **StaffID**
- [✓] StaffName
- [✓] TelephoneNo
- [ ] Sex
- [✓] Birthdate
- [ ] OfficeID

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
ManagedbySt...		Office	✓						
StaffName		Staffs	✓						
TelephoneNo		Staffs	✓						
Birthdate		Staffs	✓						
OfficeID		Office	✓	Ascending	1				

```
SELECT     TOP (100) PERCENT dbo.Office.ManagedbyStaff, dbo.Staffs.StaffName, dbo.Staffs.TelephoneNo, dbo.Staffs.Birthdate, dbo.Office.OfficeID
FROM         dbo.Office INNER JOIN
                  dbo.Staffs ON dbo.Office.ManagedbyStaff = dbo.Staffs.StaffID AND dbo.Office.OfficeID = dbo.Staffs.OfficeID
ORDER BY dbo.Office.OfficeID
```

ManagedbyStaff	StaffName	TelephoneNo	Birthdate	OfficeID
2	Richard	641-233-2345	1990-01-02	7
4	James R. Bennett	641-233-4567	1996-06-06	8
13	Louella R. Keen...	641-233-0123	1990-01-01	9
6	Annie A. Bush	641-233-6789	1998-08-08	10
15	Elvera E. Murphy	641-233-0345	1992-02-02	11
16	Lori T. Bone	641-233-0456	1991-01-01	12
17	Walter S. West	641-233-0567	1988-08-08	13
10	Cheryl M. Warren	641-233-0654	1990-09-09	14
12	Cynthia B. Pred...	641-233-0321	1990-09-09	15


SQLQuery22.sql - D...5LFVCHI\ranjo (53))*   DESKTOP-5LFVCHI....Managers_Birthday

```
SELECT * FROM Office_Managers_Birthday
```

	ManagedbyStaff	StaffName	TelephoneNo	Birthdate	OfficeID
1	2	Richard	641-233-2345	1990-01-02	7
2	4	James R. Bennett	641-233-4567	1996-06-06	8
3	13	Louella R. Keenan	641-233-0123	1990-01-01	9
4	6	Annie A. Bush	641-233-6789	1998-08-08	10
5	15	Elvera E. Murphy	641-233-0345	1992-02-02	11
6	16	Lori T. Bone	641-233-0456	1991-01-01	12
7	17	Walter S. West	641-233-0567	1988-08-08	13
8	10	Cheryl M. Warren	641-233-0654	1990-09-09	14
9	12	Cynthia B. Predmore	641-233-0321	1990-09-09	15

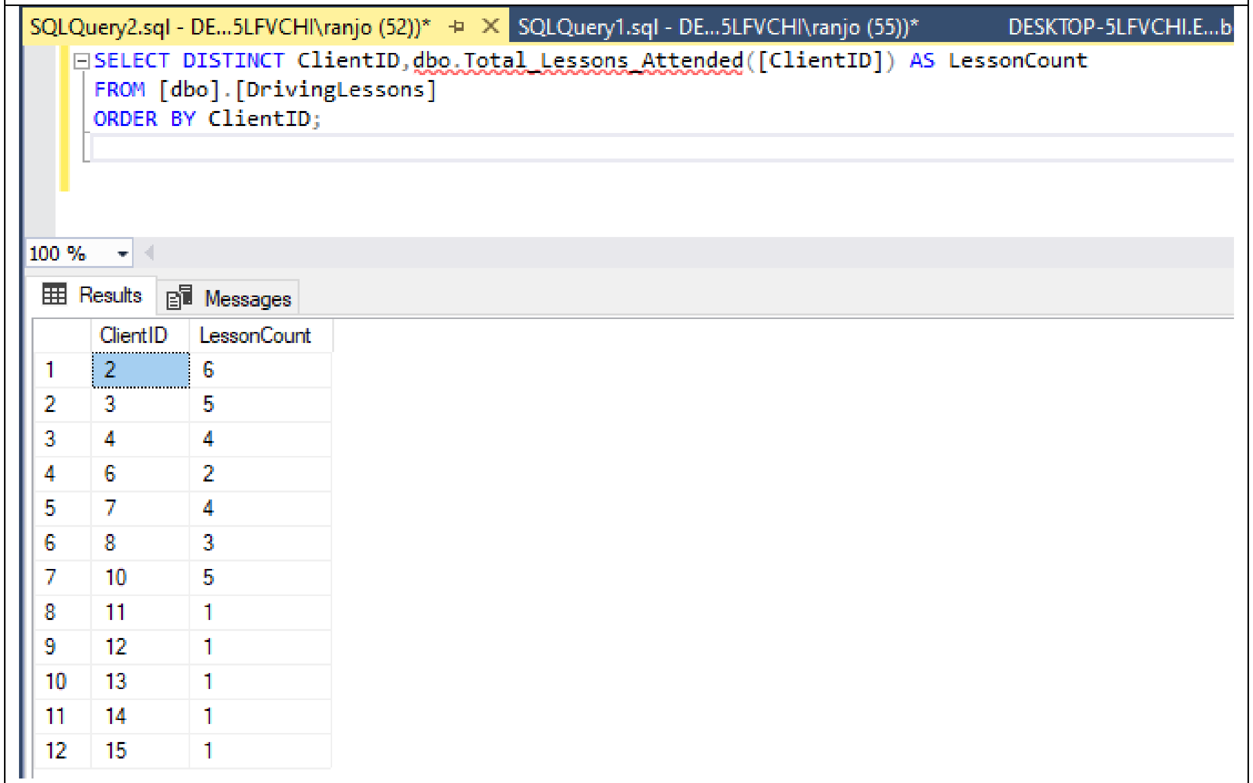
User Defined Functions

Research what user defined functions are and how to make them in SQL server.

Question 8: Create a user defined function that returns the total lessons that a client has taken up to today.

Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:              Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:  Function tp return total lesson a client attended
-- =============================================
CREATE FUNCTION Total_Lessons_Attended
(
        @ClientID int
)
RETURNS int
AS
BEGIN
        DECLARE @LessonCOUNT int;
        SET @LessonCOUNT  = (SELECT COUNT([ClientID])
        FROM [dbo].[DrivingLessons]
        WHERE [ClientID]=@ClientID);

        IF (@LessonCOUNT IS NULL)
        SET @LessonCOUNT = 0;
    RETURN @LessonCOUNT;
END
GO
```



SQLQuery2.sql - DE...5LFVCHI\ranjo (52))* SQLQuery1.sql - DE...5LFVCHI\ranjo (55))* DESKTOP-5LFVCHI.E...b

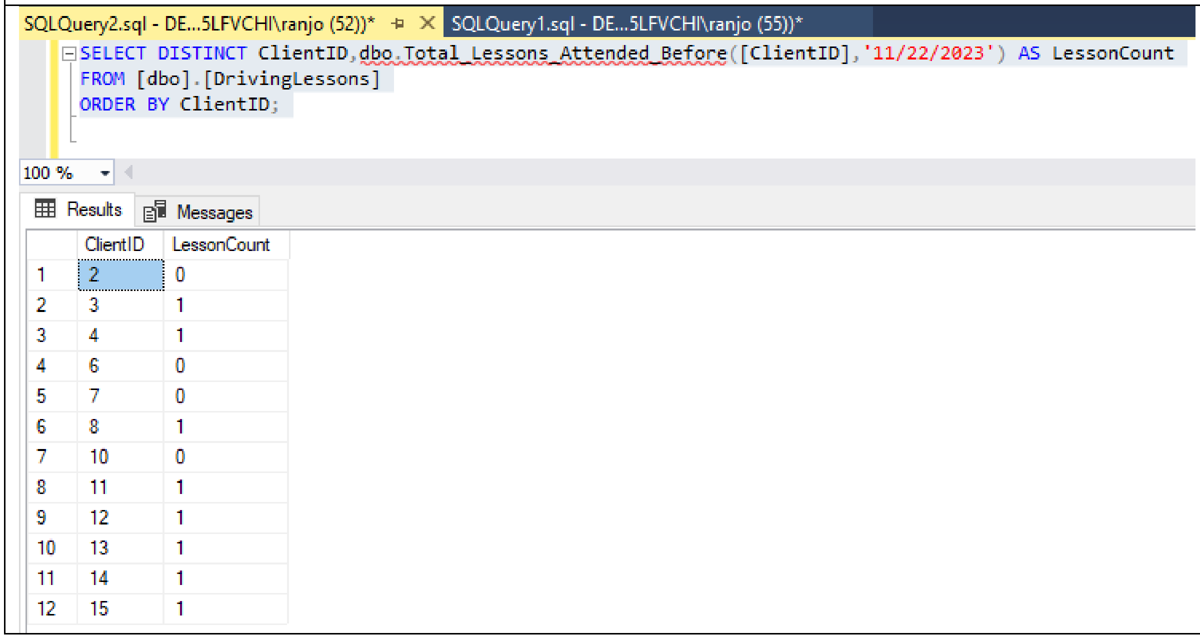
```
SELECT DISTINCT ClientID,dbo.Total_Lessons_Attended([ClientID]) AS LessonCount
FROM [dbo].[DrivingLessons]
ORDER BY ClientID;
```

	ClientID	LessonCount
1	2	6
2	3	5
3	4	4
4	6	2
5	7	4
6	8	3
7	10	5
8	11	1
9	12	1
10	13	1
11	14	1
12	15	1

Question 9: Create a user defined function that returns the total lessons that a client has taken before a date supplied by the user.

Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:                 Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:  Function tp return total lesson a client attended
-- =============================================
CREATE FUNCTION Total_Lessons_Attended_Before
(
        @ClientID int,
        @ProvidedDate Date
)
RETURNS int
AS
BEGIN
        DECLARE @LessonCOUNT int;
        SET @LessonCOUNT  = (SELECT COUNT([ClientID])
        FROM [dbo].[DrivingLessons]
        WHERE [ClientID]=@ClientID AND [LessonDateTime]<@ProvidedDate);

        IF (@LessonCOUNT IS NULL)
        SET @LessonCOUNT = 0;
    RETURN @LessonCOUNT;
END
GO
```

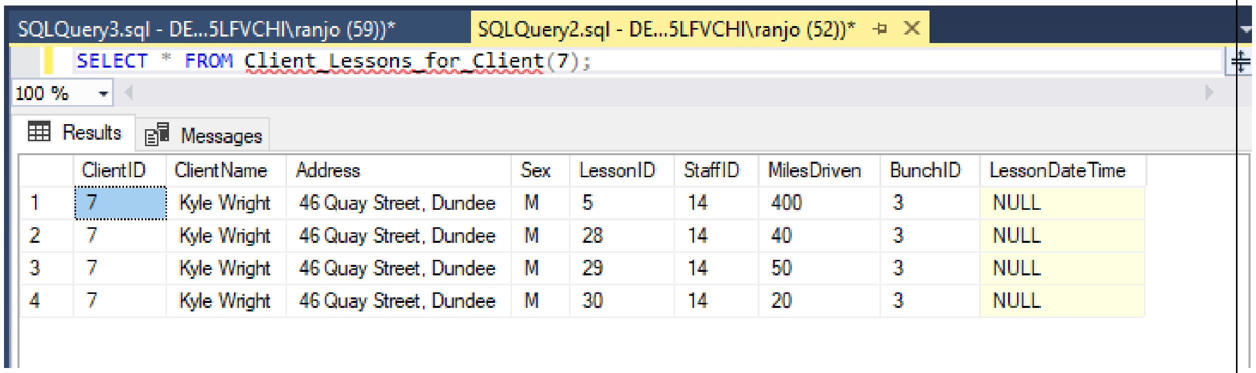

```
SELECT DISTINCT ClientID,dbo.Total_Lessons_Attended_Before([ClientID],'11/22/2023') AS LessonCount
FROM [dbo].[DrivingLessons]
ORDER BY ClientID;
```

	ClientID	LessonCount
1	2	0
2	3	1
3	4	1
4	6	0
5	7	0
6	8	1
7	10	0
8	11	1
9	12	1
10	13	1
11	14	1
12	15	1

Question 10: Create a user defined function that returns a table which does an inner join on the Client and lesson tables, for a particular client which is supplied by the user. Run it to make sure it works properly.

Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:              Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:  Client_Lessons_for_Client
-- =============================================
CREATE FUNCTION Client_Lessons_for_Client
(
        @ClientID int
)
RETURNS TABLE
AS
RETURN
(
        SELECT C.ClientID, C.ClientName, C.Address,
               C.Sex, D.LessonID, D.StaffID,
                   D.MilesDriven, D.BunchID, D.LessonDateTime
        FROM Clients C JOIN DrivingLessons D
        ON C.ClientID = D.ClientID
        WHERE C.ClientID = @ClientID
        GROUP BY C.ClientID, C.ClientName, C.Address,
               C.Sex, D.LessonID, D.StaffID,
                   D.MilesDriven, D.BunchID, D.LessonDateTime
);
GO
```


SQLQuery3.sql - DE...5LFVCHI\ranjo (59))*
SQLQuery2.sql - DE...5LFVCHI\ranjo (52))*
100 %
Results
Messages

```
SELECT * FROM Client_Lessons_for_Client(7);
```

	ClientID	ClientName	Address	Sex	LessonID	StaffID	MilesDriven	BunchID	LessonDateTime
1	7	Kyle Wright	46 Quay Street, Dundee	M	5	14	400	3	NULL
2	7	Kyle Wright	46 Quay Street, Dundee	M	28	14	40	3	NULL
3	7	Kyle Wright	46 Quay Street, Dundee	M	29	14	50	3	NULL
4	7	Kyle Wright	46 Quay Street, Dundee	M	30	14	20	3	NULL

TRIGGERS

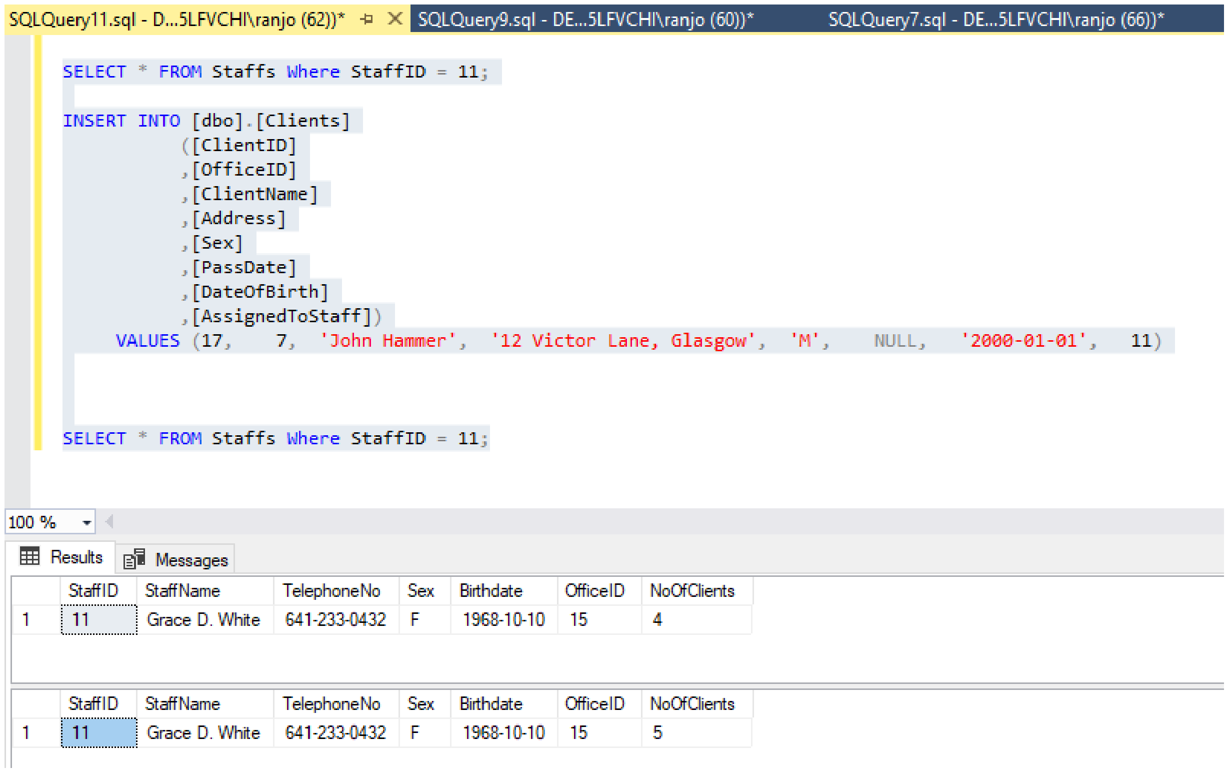
Research what Triggers are and how to make them in SQL Server.

Question 11: In the Staff Table, add an attribute to keep track of the total number of clients that an instructor has. Whenever a new client is added to the client table, we add one to the above new attribute, to the staff person who is working with this new client. A similar thing is done if a client is removed from our client Table.

Answer:

```
USE [EasyDriveSchool ]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:              Dip Ranjon Das
-- Create date: 11/22/2023
-- Description:  Getting Client Count in Staff Table
-- =============================================
CREATE OR ALTER   TRIGGER [dbo].[ClientCount_Assignedto_Staffs]
   ON  [dbo].[Clients]
   AFTER INSERT,DELETE,UPDATE

AS
BEGIN
                CREATE TABLE #StaffIDs
                (
                        StaffID INT
                );  --- Creating a temporary table

                INSERT INTO #StaffIDs (StaffID)(SELECT DISTINCT [AssignedToStaff] FROM
[dbo].[Clients]);
                --- Reading all Assigned staff ID and storing in a new table


                DECLARE @row_variable1 INT;
                DECLARE @CountClient INT;
                --- Cursor1 code
                DECLARE staffid_cursor CURSOR FOR
                SELECT * FROM #StaffIDs;

                OPEN staffid_cursor;
                FETCH NEXT FROM staffid_cursor INTO @row_variable1;  --- Fetch first item of the cursor
to variable

                WHILE @@FETCH_STATUS = 0
                BEGIN
                  SET @CountClient = (SELECT COUNT(DISTINCT ClientID) FROM [dbo].[Clients] WHERE
AssignedToStaff = @row_variable1);
                  --- Stored CountClient to @CountClient from Clients table
                  UPDATE [dbo].[Staffs] SET [NoOfClients] = @CountClient WHERE StaffID =
@row_variable1;

                  FETCH NEXT FROM staffid_cursor INTO @row_variable1;
                END;

                CLOSE staffid_cursor;
                DEALLOCATE staffid_cursor;
                --- cursor1 code End

                DROP TABLE #StaffIDs;   --- Deleteing the temporary table

END
```

Test 1: Insert Test

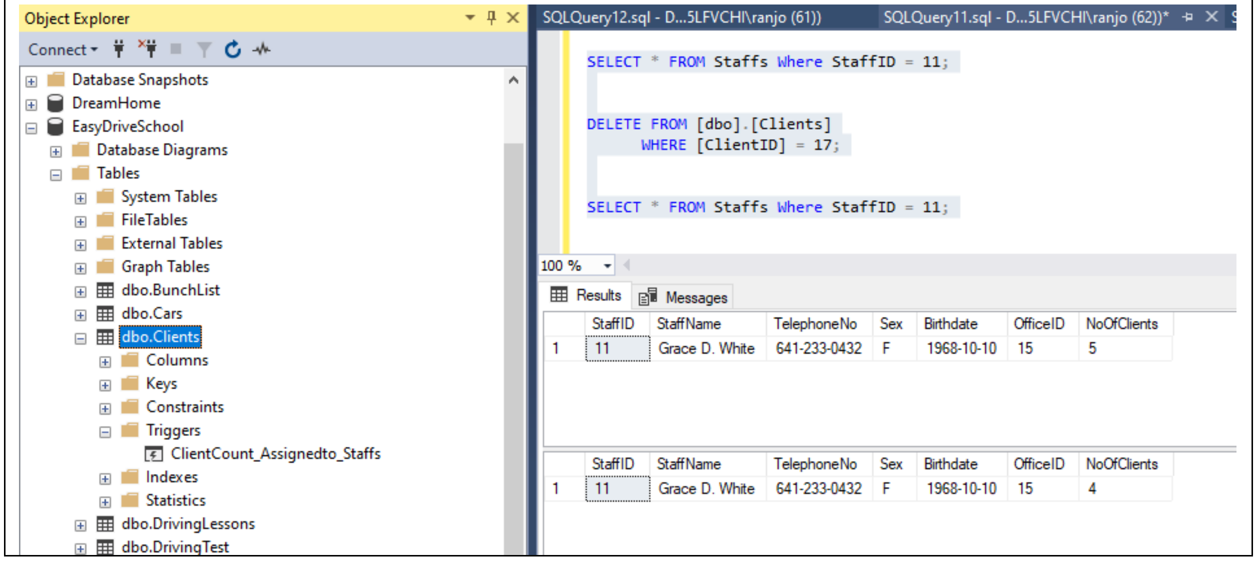

SQLQuery11.sql - D...5LFVCHI\ranjo (62))*
SQLQuery9.sql - DE...5LFVCHI\ranjo (60))*
SQLQuery7.sql - DE...5LFVCHI\ranjo (66))*
100 %
Results
Messages

```
SELECT * FROM Staffs Where StaffID = 11;

INSERT INTO [dbo].[Clients]
           ([ClientID]
           ,[OfficeID]
           ,[ClientName]
           ,[Address]
           ,[Sex]
           ,[PassDate]
           ,[DateOfBirth]
           ,[AssignedToStaff])
     VALUES (17,    7,  'John Hammer',  '12 Victor Lane, Glasgow',  'M',    NULL,    '2000-01-01',    11)


SELECT * FROM Staffs Where StaffID = 11;
```

	StaffID	StaffName	TelephoneNo	Sex	Birthdate	OfficeID	NoOfClients
1	11	Grace D. White	641-233-0432	F	1968-10-10	15	4

	StaffID	StaffName	TelephoneNo	Sex	Birthdate	OfficeID	NoOfClients
1	11	Grace D. White	641-233-0432	F	1968-10-10	15	5

Test 2: Delete Test



```
SELECT * FROM Staffs Where StaffID = 11;

DELETE FROM [dbo].[Clients]
      WHERE [ClientID] = 17;

SELECT * FROM Staffs Where StaffID = 11;
```

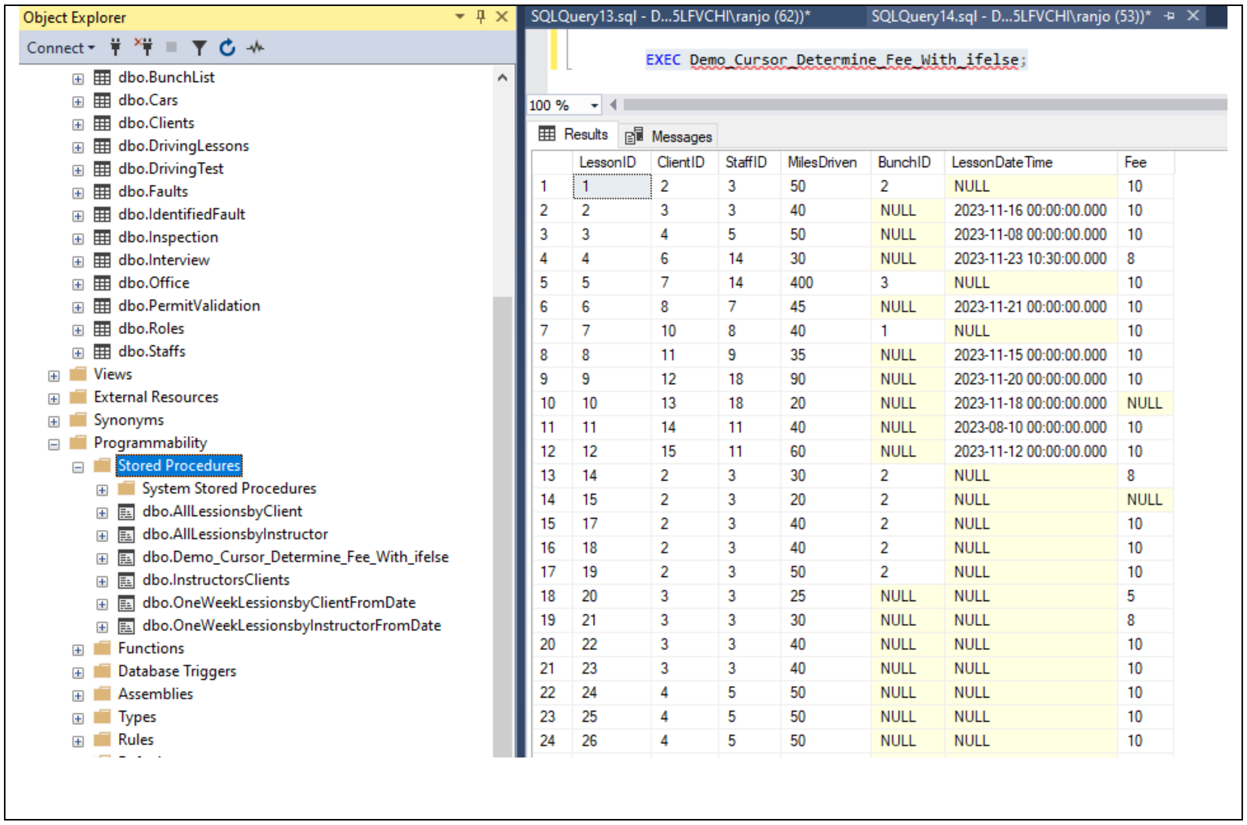
	StaffID	StaffName	TelephoneNo	Sex	Birthdate	OfficeID	NoOfClients
1	11	Grace D. White	641-233-0432	F	1968-10-10	15	5

	StaffID	StaffName	TelephoneNo	Sex	Birthdate	OfficeID	NoOfClients
1	11	Grace D. White	641-233-0432	F	1968-10-10	15	4

CURSOR

Research what a cursor is and how to make them in SQl server.
Question 12: use a cursor to read the rows of the Lesson table.
If the mileage for the lesson was over 20 miles, increase the fee by $5.
If the mileage for the lesson was over 25 miles, increase the fee by $8.
If the mileage for the lesson was over 30 miles, increase the fee by $10.
You can use an IF ….. ELSE …. Statement.
Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:               Dip Ranjon Das
-- Create date: 11/23/2023
-- Description:  Cursor code with IF Else to increase Fees
-- =============================================
CREATE PROCEDURE Demo_Cursor_Determine_Fee_With_ifelse

AS
BEGIN
                CREATE TABLE #LessonTable
                (
                        LessonID INT, ClientID INT, StaffID INT, MilesDriven INT,
                        BunchID INT, LessonDateTime datetime, Fee int
                );  --- Creating a temporary table

                INSERT INTO #LessonTable
(LessonID,ClientID,StaffID,MilesDriven,BunchID,LessonDateTime)(SELECT * FROM [dbo].[DrivingLessons]);
                --- Reading all Lessons and storing in a new temporary table

                DECLARE @row_variable1 INT;
                DECLARE @LessonID int,@ClientID int,@StaffID int ,@MilesDriven int,@BunchID
int,@LessonDateTime datetime,@Fee int;
                --- Cursor1 code
                DECLARE LessonTable_cursor CURSOR FOR
                SELECT * FROM #LessonTable;

                OPEN LessonTable_cursor;
                FETCH NEXT FROM LessonTable_cursor INTO
@LessonID,@ClientID,@StaffID,@MilesDriven,@BunchID,@LessonDateTime,@Fee;
                --- Fetch first item of the cursor to variable
                WHILE @@FETCH_STATUS = 0
                BEGIN
                        IF @MilesDriven>30 SET @Fee = 10
                        ELSE IF @MilesDriven>25 SET @Fee = 8
                        ELSE IF @MilesDriven>20  SET @Fee = 5
                        ELSE SET @Fee = @Fee
                  UPDATE #LessonTable SET [Fee] = @FEE WHERE [LessonID] = @LessonID;

                  FETCH NEXT FROM LessonTable_cursor INTO
@LessonID,@ClientID,@StaffID,@MilesDriven,@BunchID,@LessonDateTime,@Fee;
                END;

                CLOSE LessonTable_cursor;
                DEALLOCATE LessonTable_cursor;

               SELECT * FROM #LessonTable;
               DROP Table #LessonTable;
END
GO
```


Object Explorer
Connect
SQLQuery13.sql - D...5LFVCHI\ranjo (62))*
SQLQuery14.sql - D...5LFVCHI\ranjo (53))*
100 %
Results
Messages

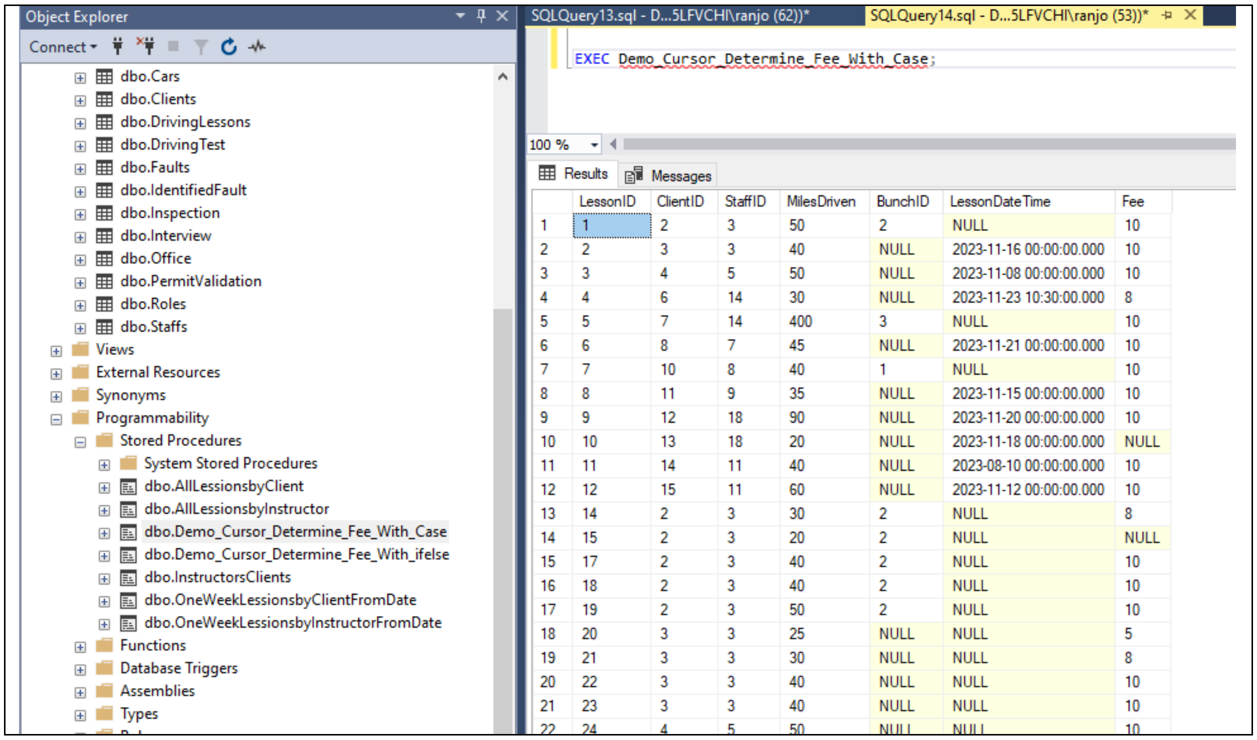
- dbo.BunchList
- dbo.Cars
- dbo.Clients
- dbo.DrivingLessons
- dbo.DrivingTest
- dbo.Faults
- dbo.IdentifiedFault
- dbo.Inspection
- dbo.Interview
- dbo.Office
- dbo.PermitValidation
- dbo.Roles
- dbo.Staffs
- Views
- External Resources
- Synonyms
- Programmability
  - Stored Procedures
    - System Stored Procedures
    - dbo.AllLessionsbyClient
    - dbo.AllLessionsbyInstructor
    - dbo.Demo_Cursor_Determine_Fee_With_ifelse
    - dbo.InstructorsClients
    - dbo.OneWeekLessionsbyClientFromDate
    - dbo.OneWeekLessionsbyInstructorFromDate
  - Functions
  - Database Triggers
  - Assemblies
  - Types
  - Rules

```sql
EXEC Demo_Cursor_Determine_Fee_With_ifelse;
```

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime	Fee
1	1	2	3	50	2	NULL	10
2	2	3	3	40	NULL	2023-11-16 00:00:00.000	10
3	3	4	5	50	NULL	2023-11-08 00:00:00.000	10
4	4	6	14	30	NULL	2023-11-23 10:30:00.000	8
5	5	7	14	400	3	NULL	10
6	6	8	7	45	NULL	2023-11-21 00:00:00.000	10
7	7	10	8	40	1	NULL	10
8	8	11	9	35	NULL	2023-11-15 00:00:00.000	10
9	9	12	18	90	NULL	2023-11-20 00:00:00.000	10
10	10	13	18	20	NULL	2023-11-18 00:00:00.000	NULL
11	11	14	11	40	NULL	2023-08-10 00:00:00.000	10
12	12	15	11	60	NULL	2023-11-12 00:00:00.000	10
13	14	2	3	30	2	NULL	8
14	15	2	3	20	2	NULL	NULL
15	17	2	3	40	2	NULL	10
16	18	2	3	40	2	NULL	10
17	19	2	3	50	2	NULL	10
18	20	3	3	25	NULL	NULL	5
19	21	3	3	30	NULL	NULL	8
20	22	3	3	40	NULL	NULL	10
21	23	3	3	40	NULL	NULL	10
22	24	4	5	50	NULL	NULL	10
23	25	4	5	50	NULL	NULL	10
24	26	4	5	50	NULL	NULL	10

Question 13: Do the same thing as Question 12, but now use a case statement.
Answer:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:		Dip Ranjon Das
-- Create date: 11/23/2023
-- Description:	Cursor code with IF Else to increase Fees
-- =============================================
CREATE OR ALTER PROCEDURE Demo_Cursor_Determine_Fee_With_Case

AS
BEGIN
		CREATE TABLE #LessonTable
		(
			LessonID INT,
			ClientID INT,
			StaffID INT,
			MilesDriven INT,
			BunchID INT,
			LessonDateTime datetime,
			Fee int
		);  --- Creating a temporary table

		INSERT INTO #LessonTable
(LessonID,ClientID,StaffID,MilesDriven,BunchID,LessonDateTime)(SELECT * FROM [dbo].[DrivingLessons]);
		--- Reading all Lessons and storing in a new temporary table


		DECLARE @row_variable1 INT;
		DECLARE @LessonID int,@ClientID int,@StaffID int ,@MilesDriven int,@BunchID
int,@LessonDateTime datetime,@Fee int;
		--- Cursor1 code
		DECLARE LessonTable_cursor CURSOR FOR
		SELECT * FROM #LessonTable;

		OPEN LessonTable_cursor;
		FETCH NEXT FROM LessonTable_cursor INTO
@LessonID,@ClientID,@StaffID,@MilesDriven,@BunchID,@LessonDateTime,@Fee;
		--- Fetch first item of the cursor to variable


		WHILE @@FETCH_STATUS = 0
		BEGIN
			SET @Fee = CASE
				WHEN @MilesDriven>30 THEN 10
				WHEN @MilesDriven>25 THEN 8
				WHEN @MilesDriven>20 THEN 5
				ELSE @Fee
			END;
		  UPDATE #LessonTable SET [Fee] = @FEE WHERE [LessonID] = @LessonID;

		 FETCH NEXT FROM LessonTable_cursor INTO
		 @LessonID,@ClientID,@StaffID,@MilesDriven,@BunchID,@LessonDateTime,@Fee;
		END;

		CLOSE LessonTable_cursor;
		DEALLOCATE LessonTable_cursor;

		SELECT * FROM #LessonTable;
		DROP Table #LessonTable;
END
GO
```



```sql
EXEC Demo_Cursor_Determine_Fee_With_Case;
```

	LessonID	ClientID	StaffID	MilesDriven	BunchID	LessonDateTime	Fee
1	1	2	3	50	2	NULL	10
2	2	3	3	40	NULL	2023-11-16 00:00:00.000	10
3	3	4	5	50	NULL	2023-11-08 00:00:00.000	10
4	4	6	14	30	NULL	2023-11-23 10:30:00.000	8
5	5	7	14	400	3	NULL	10
6	6	8	7	45	NULL	2023-11-21 00:00:00.000	10
7	7	10	8	40	1	NULL	10
8	8	11	9	35	NULL	2023-11-15 00:00:00.000	10
9	9	12	18	90	NULL	2023-11-20 00:00:00.000	10
10	10	13	18	20	NULL	2023-11-18 00:00:00.000	NULL
11	11	14	11	40	NULL	2023-08-10 00:00:00.000	10
12	12	15	11	60	NULL	2023-11-12 00:00:00.000	10
13	14	2	3	30	2	NULL	8
14	15	2	3	20	2	NULL	NULL
15	17	2	3	40	2	NULL	10
16	18	2	3	40	2	NULL	10
17	19	2	3	50	2	NULL	10
18	20	3	3	25	NULL	NULL	5
19	21	3	3	30	NULL	NULL	8
20	22	3	3	40	NULL	NULL	10
21	23	3	3	40	NULL	NULL	10
22	24	4	5	50	NULL	NULL	10