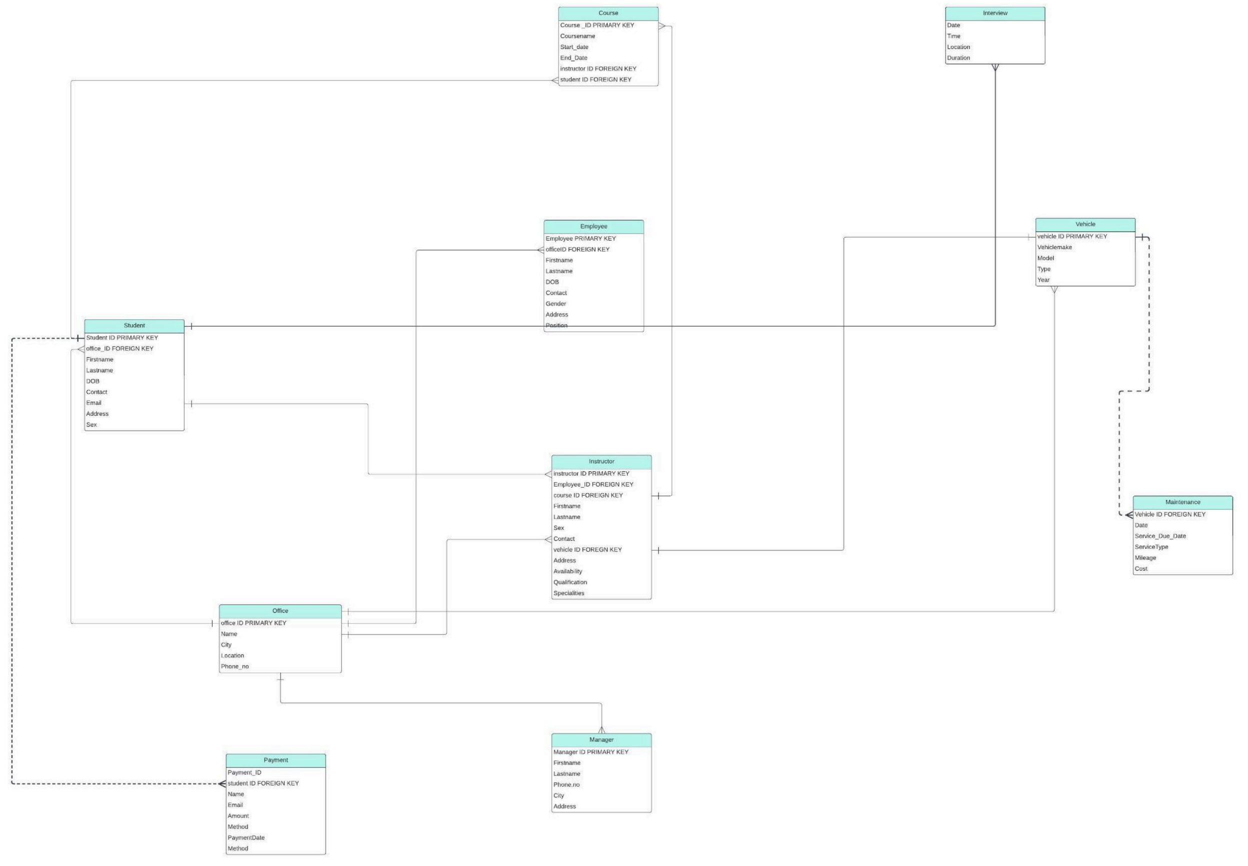
Team member

Sushant humagain(19760hs)

Swagat Neupane(19698ns)

Ronish shrestha(19707sr)

4.1 Map the ER model developed at the end of Module 3 to a relational model.



5.1 Update the data dictionary and list of assumptions as needed. For each table,

write the table name and write out the names, data types, and sizes of all the data
items, and identify any constraints, using the conventions of the DBMS you will use
for implementation. 10
Write the queries written in MySQL to create the table here .

→

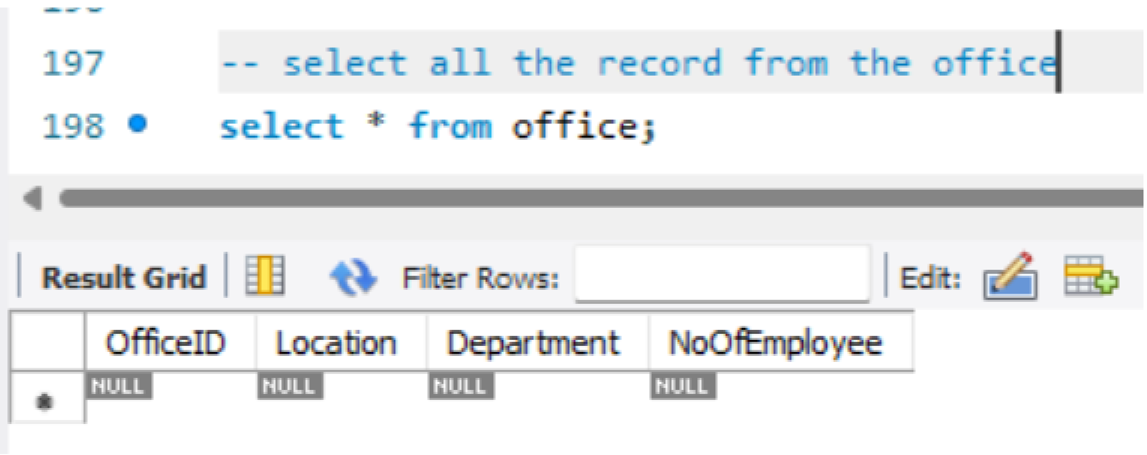
```
CREATE TABLE Office (
    OfficeID INT PRIMARY KEY,
    Location VARCHAR(255),
    Department VARCHAR(255),
    NoOfEmployee INT
);

CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(255),
    DOB DATE,
    Address VARCHAR(255),
    Position VARCHAR(255),
    OfficeID INT,
    FOREIGN KEY (OfficeID) REFERENCES Office(OfficeID)
);

CREATE TABLE Student (
   StudentID INT PRIMARY KEY,
   FirstName VARCHAR(255),
   LastName VARCHAR(255),
   DOB DATE,
   ContactNo VARCHAR(15),
   Email VARCHAR(255),
   Sex CHAR,
   Address VARCHAR(255),
   OfficeID INT,
   FOREIGN KEY (OfficeID) REFERENCES Office (OfficeID)
);
```

```
CREATE TABLE Instructor (
  InstructorID INT PRIMARY KEY,
  EmployeeID INT,
  Course_ID INT,
  FirstName VARCHAR (255),
  LastName VARCHAR (255),
  Addresss varchar (500),
  Vehicle_ID int,
  Sex CHAR,
  Availability BOOLEAN ,
  FOREIGN KEY(EmployeeID) REFERENCES Employee(EmployeeID)
);

CREATE TABLE Course(
Course_ID int primary key ,
InstructorID int ,
StudentID int ,
Start_Date date ,
End_Date date ,
Duration varchar (50) ,
Level varchar (50) ,
FOREIGN KEY(InstructorID) REFERENCES Instructor(InstructorID),
FOREIGN Key(StudentId ) references student(studentId )
);


CREATE TABLE Vehicle(
Vehicle_Id int primary key ,
Vehicle_Make varchar (1000 ),
Vehicle_Type varchar (1000 ),
Model_Year date ,
Plate_Number varchar (1000 ),
officeID int ,
FOREIGN Key (OfficeID) references office(OfficeID)
);
```

```
Create table Maintenance(
vehicle_ID int,
Vehicle_Type Varchar(50) not null,
Date_Serviced Date not null ,
service_type varchar(250),
Cost decimal(10,2) not null check(cost >0 ) default '0' not null ,
Next_Schedule_Maintenance Date Not Null,
FOREIGN KEY (vehicle_ID) references Vehicle(vehicle_id)
);
```

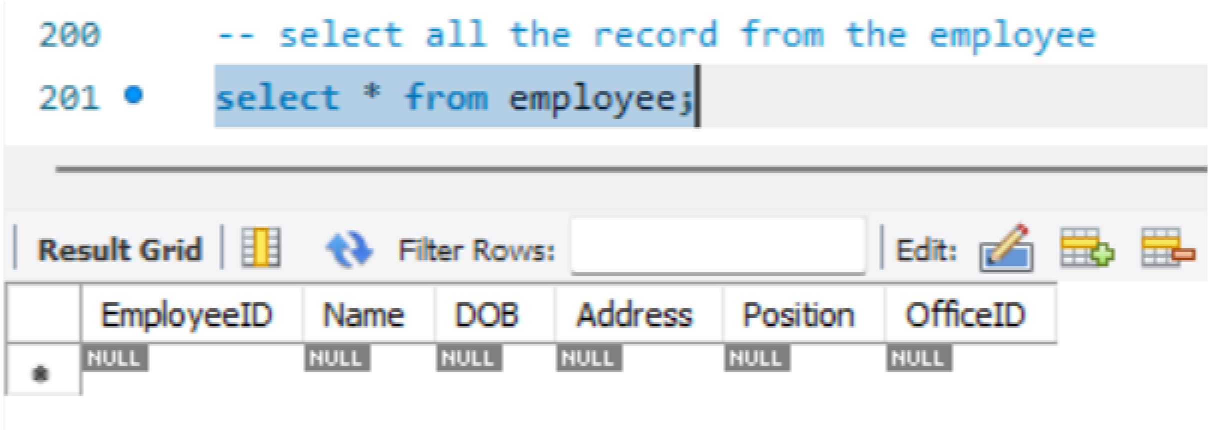
```
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    Payment_Date DATE,
    Name VARCHAR(255),
    Amount DECIMAL(10, 2),
    Email VARCHAR(255),
    studentID INT
);
```

5.2 Write and execute SQL statements to create all tables needed to implement the design.



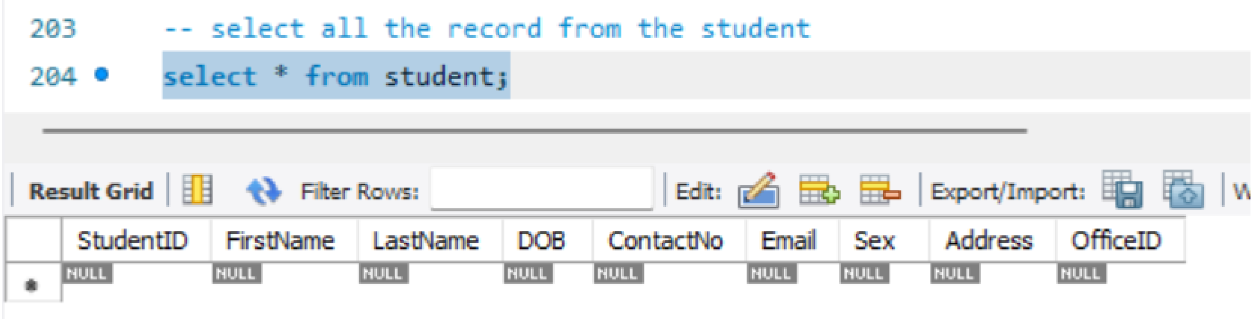
```sql
-- select all the record from the office
select * from office;
```

	OfficeID	Location	Department	NoOfEmployee
*	NULL	NULL	NULL	NULL



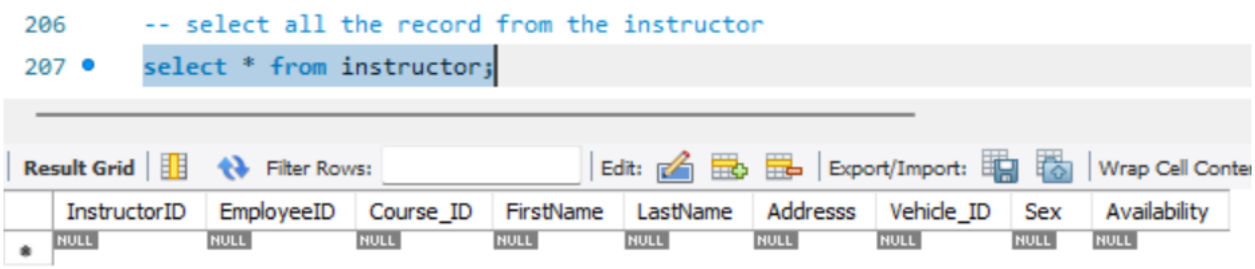
```sql
-- select all the record from the employee
select * from employee;
```

	EmployeeID	Name	DOB	Address	Position	OfficeID
*	NULL	NULL	NULL	NULL	NULL	NULL



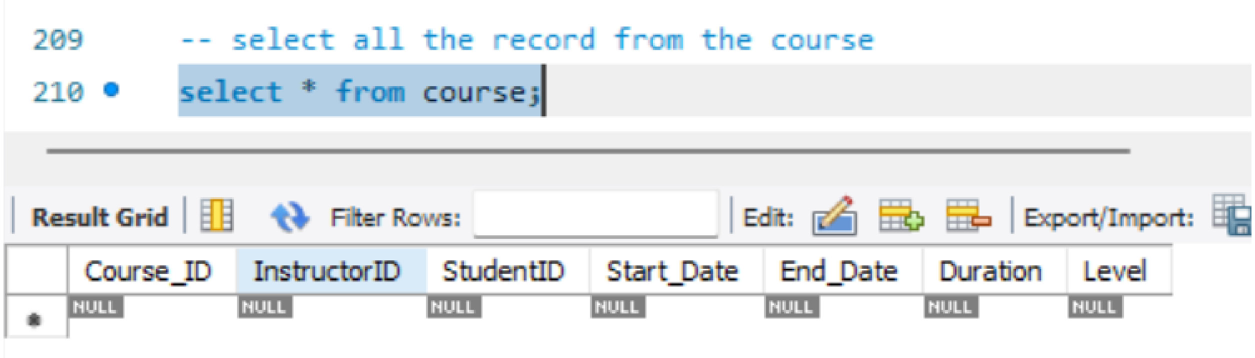
```sql
-- select all the record from the student
select * from student;
```

	StudentID	FirstName	LastName	DOB	ContactNo	Email	Sex	Address	OfficeID
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



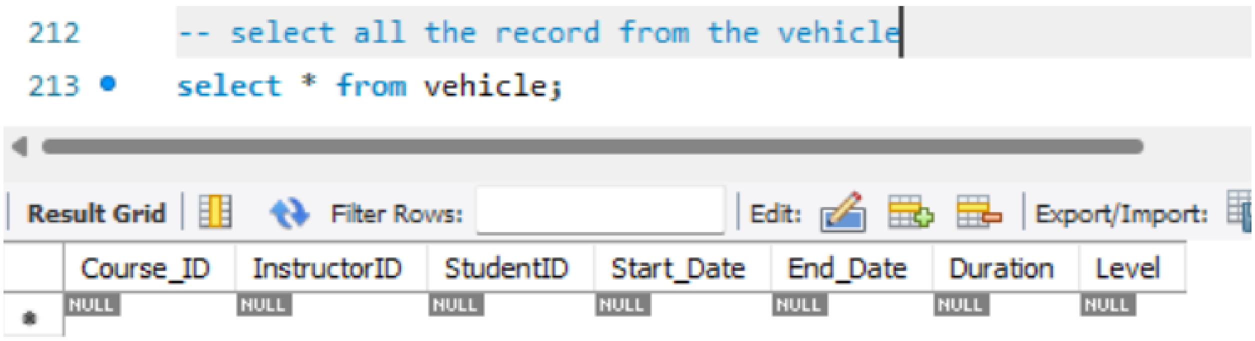
```
-- select all the record from the instructor
select * from instructor;
```

InstructorID	EmployeeID	Course_ID	FirstName	LastName	Addresss	Vehicle_ID	Sex	Availability
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



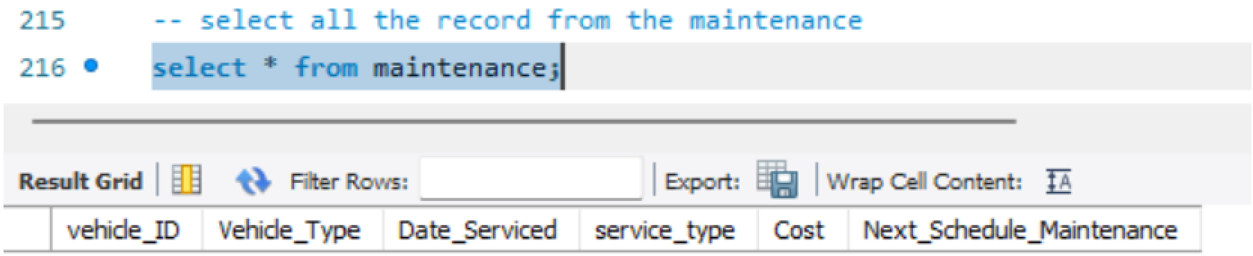
```
-- select all the record from the course
select * from course;
```

Course_ID	InstructorID	StudentID	Start_Date	End_Date	Duration	Level
NULL	NULL	NULL	NULL	NULL	NULL	NULL



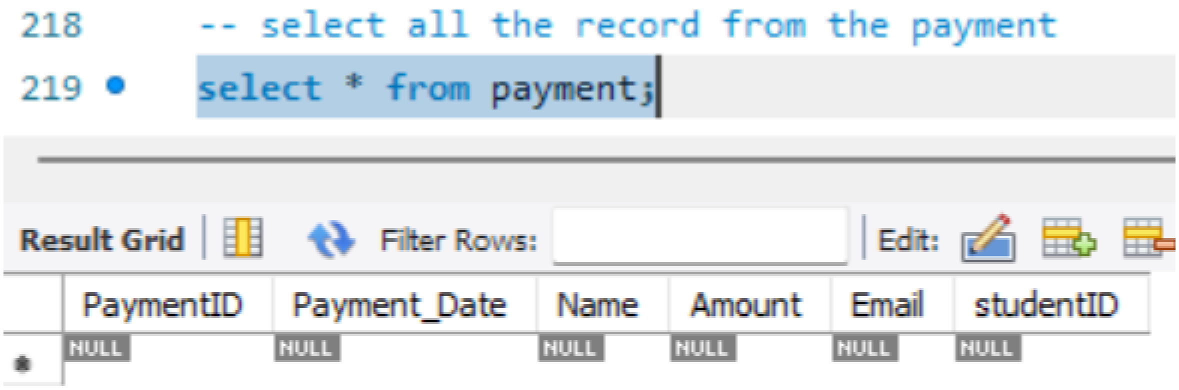
```
-- select all the record from the vehicle
select * from vehicle;
```

Course_ID	InstructorID	StudentID	Start_Date	End_Date	Duration	Level
NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
-- select all the record from the maintenance
select * from maintenance;
```

vehicle_ID	Vehicle_Type	Date_Serviced	service_type	Cost	Next_Schedule_Maintenance

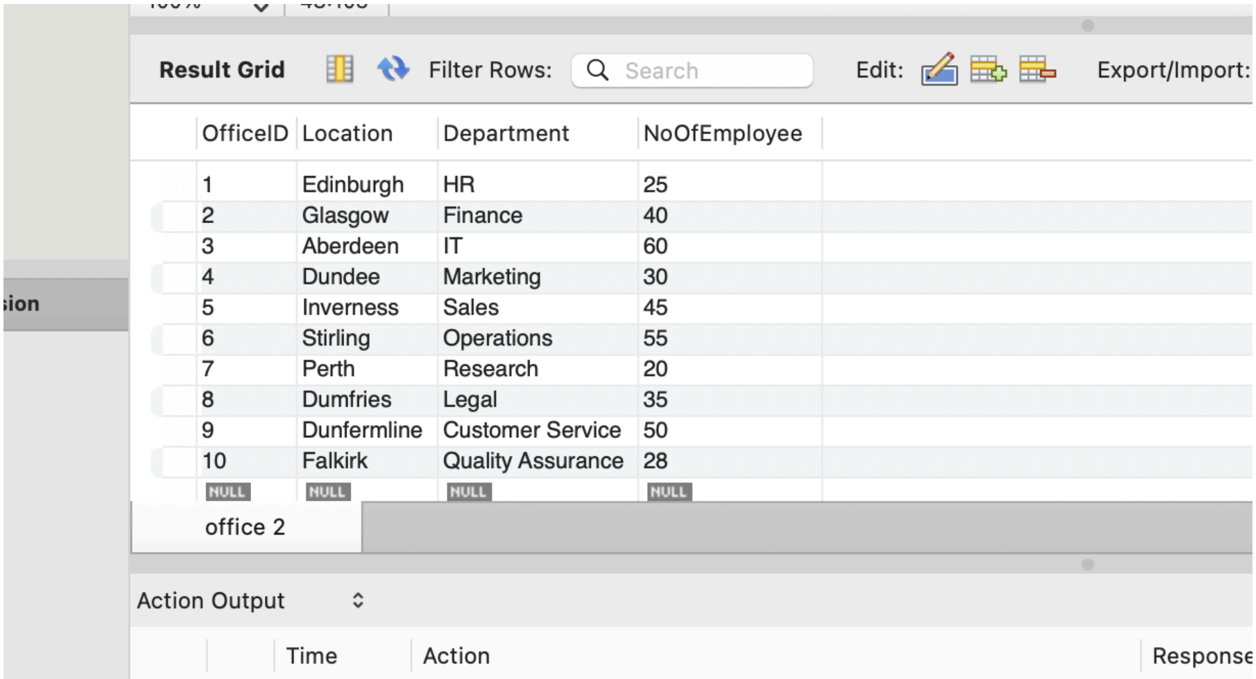


```
-- select all the record from the payment
select * from payment;
```

PaymentID	Payment_Date	Name	Amount	Email	studentID
NULL	NULL	NULL	NULL	NULL	NULL

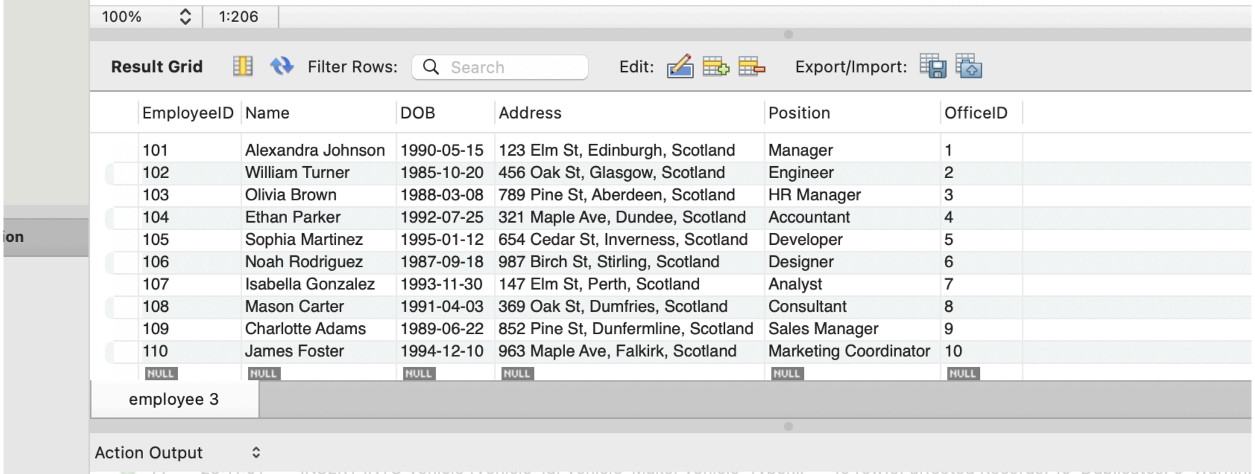
Step 5.3 Insert at least ten records in each table, preserving all constraints. Put in enough data to demonstrate how the database will function.

```
INSERT INTO Office (OfficeID, Location, Department, NoOfEmployee)
VALUES
 (1, 'Edinburgh', 'HR', 25),
  (2, 'Glasgow', 'Finance', 40),
  (3, 'Aberdeen', 'IT', 60),
  (4, 'Dundee', 'Marketing', 30),
  (5, 'Inverness', 'Sales', 45),
  (6, 'Stirling', 'Operations', 55),
  (7, 'Perth', 'Research', 20),
  (8, 'Dumfries', 'Legal', 35),
  (9, 'Dunfermline', 'Customer Service', 50),
  (10, 'Falkirk', 'Quality Assurance', 28);
```



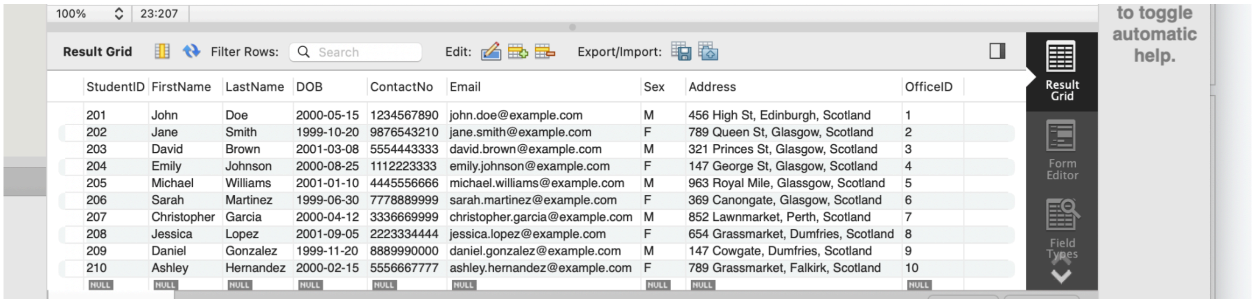
OfficeID	Location	Department	NoOfEmployee
1	Edinburgh	HR	25
2	Glasgow	Finance	40
3	Aberdeen	IT	60
4	Dundee	Marketing	30
5	Inverness	Sales	45
6	Stirling	Operations	55
7	Perth	Research	20
8	Dumfries	Legal	35
9	Dunfermline	Customer Service	50
10	Falkirk	Quality Assurance	28
NULL	NULL	NULL	NULL

```
    (101, 'Alexandra Johnson', '1990-05-15', '123 Elm St, Edinburgh,
Scotland', 'Manager', 1),
    (102, 'William Turner', '1985-10-20', '456 Oak St, Glasgow, Scotland',
'Engineer', 2),
    (103, 'Olivia Brown', '1988-03-08', '789 Pine St, Aberdeen, Scotland', 'HR
Manager', 3),
    (104, 'Ethan Parker', '1992-07-25', '321 Maple Ave, Dundee, Scotland',
'Accountant', 4),
    (105, 'Sophia Martinez', '1995-01-12', '654 Cedar St, Inverness, Scotland',
'Developer', 5),
    (106, 'Noah Rodriguez', '1987-09-18', '987 Birch St, Stirling, Scotland',
'Designer', 6),
    (107, 'Isabella Gonzalez', '1993-11-30', '147 Elm St, Perth, Scotland',
'Analyst', 7),
    (108, 'Mason Carter', '1991-04-03', '369 Oak St, Dumfries, Scotland',
'Consultant', 8),
    (109, 'Charlotte Adams', '1989-06-22', '852 Pine St, Dunfermline,
Scotland', 'Sales Manager', 9),
    (110, 'James Foster', '1994-12-10', '963 Maple Ave, Falkirk, Scotland',
'Marketing Coordinator', 10);
```



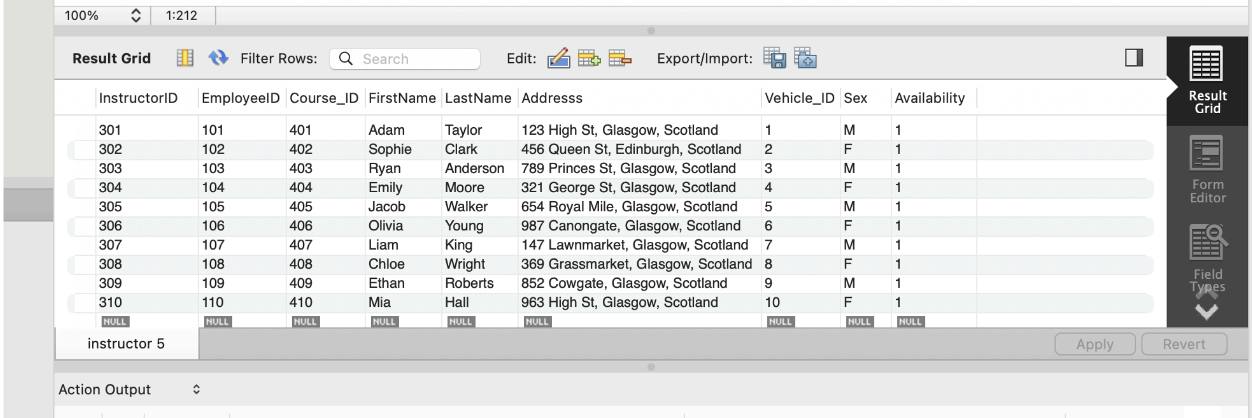
EmployeeID	Name	DOB	Address	Position	OfficeID
101	Alexandra Johnson	1990-05-15	123 Elm St, Edinburgh, Scotland	Manager	1
102	William Turner	1985-10-20	456 Oak St, Glasgow, Scotland	Engineer	2
103	Olivia Brown	1988-03-08	789 Pine St, Aberdeen, Scotland	HR Manager	3
104	Ethan Parker	1992-07-25	321 Maple Ave, Dundee, Scotland	Accountant	4
105	Sophia Martinez	1995-01-12	654 Cedar St, Inverness, Scotland	Developer	5
106	Noah Rodriguez	1987-09-18	987 Birch St, Stirling, Scotland	Designer	6
107	Isabella Gonzalez	1993-11-30	147 Elm St, Perth, Scotland	Analyst	7
108	Mason Carter	1991-04-03	369 Oak St, Dumfries, Scotland	Consultant	8
109	Charlotte Adams	1989-06-22	852 Pine St, Dunfermline, Scotland	Sales Manager	9
110	James Foster	1994-12-10	963 Maple Ave, Falkirk, Scotland	Marketing Coordinator	10
NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Student (StudentID, FirstName, LastName, DOB, ContactNo,
Email, Sex, Address, OfficeID)
VALUES
(201, 'John', 'Doe', '2000-05-15', '1234567890', 'john.doe@example.com', 'M',
'456 High St, Edinburgh, Scotland', 1),
   (202, 'Jane', 'Smith', '1999-10-20', '9876543210',
'jane.smith@example.com', 'F', '789 Queen St, Glasgow, Scotland', 2),
   (203, 'David', 'Brown', '2001-03-08', '5554443333',
'david.brown@example.com', 'M', '321 Princes St, Glasgow, Scotland', 3),
   (204, 'Emily', 'Johnson', '2000-08-25', '1112223333',
'emily.johnson@example.com', 'F', '147 George St, Glasgow, Scotland', 4),
   (205, 'Michael', 'Williams', '2001-01-10', '4445556666',
'michael.williams@example.com', 'M', '963 Royal Mile, Glassgow, Scotland',
5),
   (206, 'Sarah', 'Martinez', '1999-06-30', '7778889999',
'sarah.martinez@example.com', 'F', '369 Canongate, Glasgow, Scotland', 6),
   (207, 'Christopher', 'Garcia', '2000-04-12', '3336669999',
'christopher.garcia@example.com', 'M', '852 Lawnmarket, Perth, Scotland',
7),
   (208, 'Jessica', 'Lopez', '2001-09-05', '2223334444',
'jessica.lopez@example.com', 'F', '654 Grassmarket, Dumfries, Scotland',
8),
    (209, 'Daniel', 'Gonzalez', '1999-11-20', '8889990000',
'daniel.gonzalez@example.com', 'M', '147 Cowgate, Dumfries, Scotland', 9),
    (210, 'Ashley', 'Hernandez', '2000-02-15', '5556667777',
'ashley.hernandez@example.com', 'F', '789 Grassmarket, Falkirk, Scotland',
10);
```



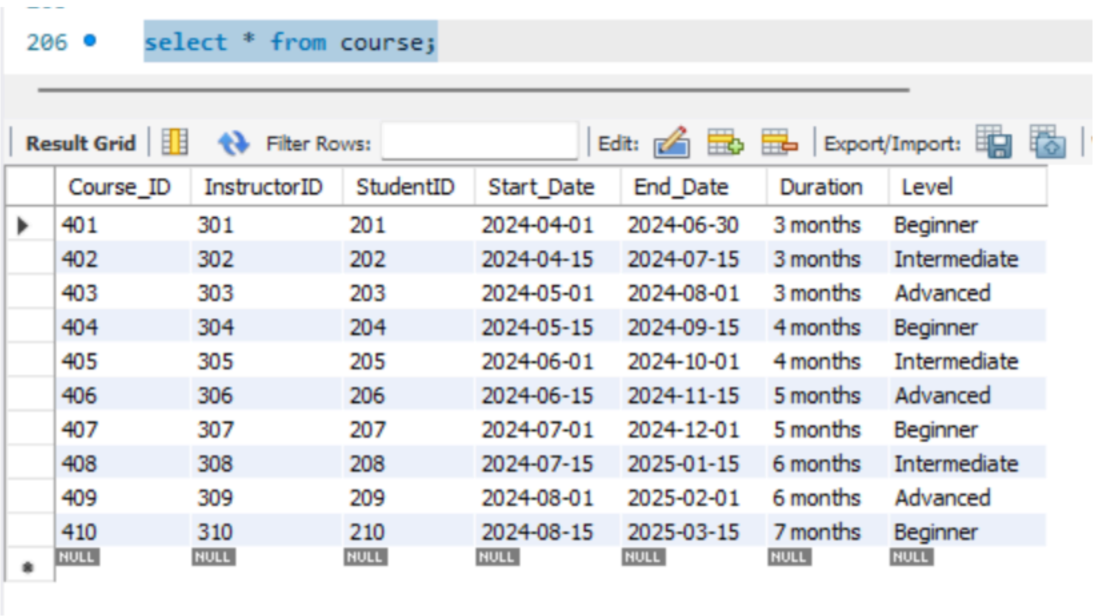
StudentID	FirstName	LastName	DOB	ContactNo	Email	Sex	Address	OfficeID
201	John	Doe	2000-05-15	1234567890	john.doe@example.com	M	456 High St, Edinburgh, Scotland	1
202	Jane	Smith	1999-10-20	9876543210	jane.smith@example.com	F	789 Queen St, Glasgow, Scotland	2
203	David	Brown	2001-03-08	5554443333	david.brown@example.com	M	321 Princes St, Glasgow, Scotland	3
204	Emily	Johnson	2000-08-25	1112223333	emily.johnson@example.com	F	147 George St, Glasgow, Scotland	4
205	Michael	Williams	2001-01-10	4445556666	michael.williams@example.com	M	963 Royal Mile, Glassgow, Scotland	5
206	Sarah	Martinez	1999-06-30	7778889999	sarah.martinez@example.com	F	369 Canongate, Glasgow, Scotland	6
207	Christopher	Garcia	2000-04-12	3336669999	christopher.garcia@example.com	M	852 Lawnmarket, Perth, Scotland	7
208	Jessica	Lopez	2001-09-05	2223334444	jessica.lopez@example.com	F	654 Grassmarket, Dumfries, Scotland	8
209	Daniel	Gonzalez	1999-11-20	8889990000	daniel.gonzalez@example.com	M	147 Cowgate, Dumfries, Scotland	9
210	Ashley	Hernandez	2000-02-15	5556667777	ashley.hernandez@example.com	F	789 Grassmarket, Falkirk, Scotland	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Instructor (InstructorID, EmployeeID, Course_ID, FirstName,
LastName, Addresss, Vehicle_ID, Sex, Availability) VALUES
(301, 101, 401, 'Adam', 'Taylor', '123 High St, Glasgow, Scotland', 1, 'M', 1),
(302, 102, 402, 'Sophie', 'Clark', '456 Queen St, Edinburgh, Scotland', 2, 'F',
1),
(303, 103, 403, 'Ryan', 'Anderson', '789 Princes St, Glasgow, Scotland', 3,
'M', 1),
(304, 104, 404, 'Emily', 'Moore', '321 George St, Glasgow, Scotland', 4, 'F', 1),
(305, 105, 405, 'Jacob', 'Walker', '654 Royal Mile, Glasgow, Scotland', 5, 'M',
1),
(306, 106, 406, 'Olivia', 'Young', '987 Canongate, Glasgow, Scotland', 6, 'F',
1),
(307, 107, 407, 'Liam', 'King', '147 Lawnmarket, Glasgow, Scotland', 7, 'M',
1),
(308,108, 408, 'Chloe', 'Wright', '369 Grassmarket, Glasgow, Scotland', 8, 'F',
1),
(309, 109, 409, 'Ethan', 'Roberts', '852 Cowgate, Glasgow, Scotland', 9, 'M',
1),
(310, 110, 410, 'Mia', 'Hall', '963 High St, Glasgow, Scotland', 10, 'F', 1);
```



InstructorID	EmployeeID	Course_ID	FirstName	LastName	Addresss	Vehicle_ID	Sex	Availability
301	101	401	Adam	Taylor	123 High St, Glasgow, Scotland	1	M	1
302	102	402	Sophie	Clark	456 Queen St, Edinburgh, Scotland	2	F	1
303	103	403	Ryan	Anderson	789 Princes St, Glasgow, Scotland	3	M	1
304	104	404	Emily	Moore	321 George St, Glasgow, Scotland	4	F	1
305	105	405	Jacob	Walker	654 Royal Mile, Glasgow, Scotland	5	M	1
306	106	406	Olivia	Young	987 Canongate, Glasgow, Scotland	6	F	1
307	107	407	Liam	King	147 Lawnmarket, Glasgow, Scotland	7	M	1
308	108	408	Chloe	Wright	369 Grassmarket, Glasgow, Scotland	8	F	1
309	109	409	Ethan	Roberts	852 Cowgate, Glasgow, Scotland	9	M	1
310	110	410	Mia	Hall	963 High St, Glasgow, Scotland	10	F	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

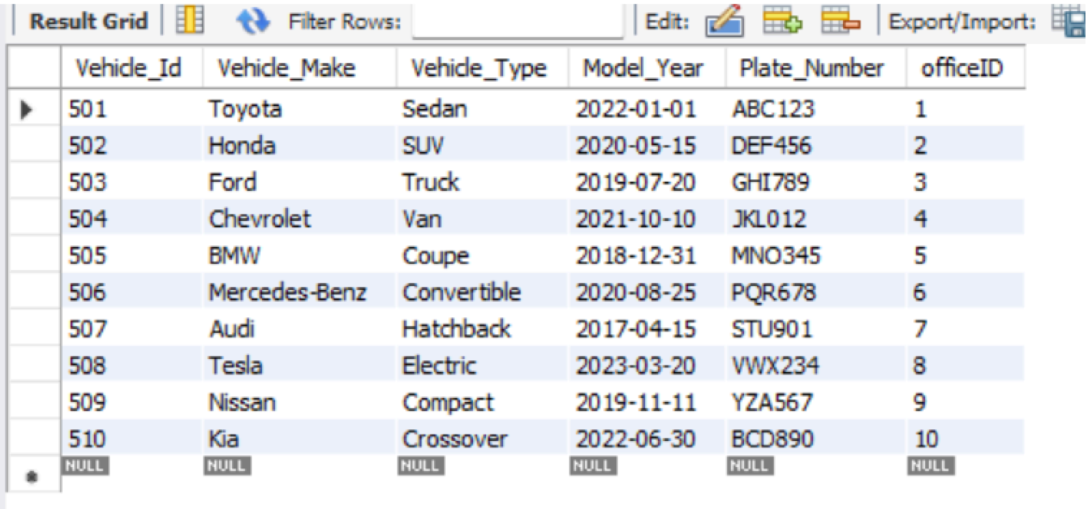
```
INSERT INTO Course (Course_ID, InstructorID, StudentID, Start_Date,
End_Date, Duration, Level) VALUES
(401, 301, 201, '2024-04-01', '2024-06-30', '3 months', 'Beginner'),
(402, 302, 202, '2024-04-15', '2024-07-15', '3 months', 'Intermediate'),
(403, 303, 203, '2024-05-01', '2024-08-01', '3 months', 'Advanced'),
(404, 304, 204, '2024-05-15', '2024-09-15', '4 months', 'Beginner'),
(405, 305, 205, '2024-06-01', '2024-10-01', '4 months', 'Intermediate'),
(406, 306, 206, '2024-06-15', '2024-11-15', '5 months', 'Advanced'),
(407, 307, 207, '2024-07-01', '2024-12-01', '5 months', 'Beginner'),
(408, 308, 208, '2024-07-15', '2025-01-15', '6 months', 'Intermediate'),
(409, 309, 209, '2024-08-01', '2025-02-01', '6 months', 'Advanced'),
(410, 310, 210, '2024-08-15', '2025-03-15', '7 months', 'Beginner');
```



```
select * from course;
```

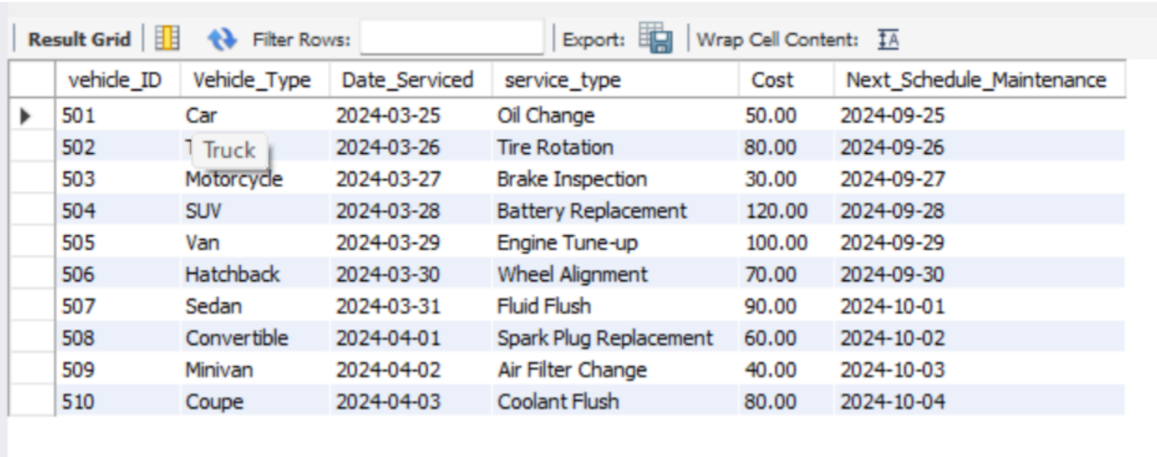
Course_ID	InstructorID	StudentID	Start_Date	End_Date	Duration	Level
401	301	201	2024-04-01	2024-06-30	3 months	Beginner
402	302	202	2024-04-15	2024-07-15	3 months	Intermediate
403	303	203	2024-05-01	2024-08-01	3 months	Advanced
404	304	204	2024-05-15	2024-09-15	4 months	Beginner
405	305	205	2024-06-01	2024-10-01	4 months	Intermediate
406	306	206	2024-06-15	2024-11-15	5 months	Advanced
407	307	207	2024-07-01	2024-12-01	5 months	Beginner
408	308	208	2024-07-15	2025-01-15	6 months	Intermediate
409	309	209	2024-08-01	2025-02-01	6 months	Advanced
410	310	210	2024-08-15	2025-03-15	7 months	Beginner
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Vehicle (Vehicle_Id, Vehicle_Make, Vehicle_Type, Model_Year,
Plate_Number, officeid) VALUES
(501, 'Toyota', 'Sedan', '2022-01-01', 'ABC123', 1),
(502, 'Honda', 'SUV', '2020-05-15', 'DEF456', 2),
(503, 'Ford', 'Truck', '2019-07-20', 'GHI789', 3),
(504, 'Chevrolet', 'Van', '2021-10-10', 'JKL012', 4),
(505, 'BMW', 'Coupe', '2018-12-31', 'MNO345', 5),
(506, 'Mercedes-Benz', 'Convertible', '2020-08-25', 'PQR678', 6),
(507, 'Audi', 'Hatchback', '2017-04-15', 'STU901', 7),
(508, 'Tesla', 'Electric', '2023-03-20', 'VWX234', 8),
(509, 'Nissan', 'Compact', '2019-11-11', 'YZA567', 9),
(510, 'Kia', 'Crossover', '2022-06-30', 'BCD890', 10);
```



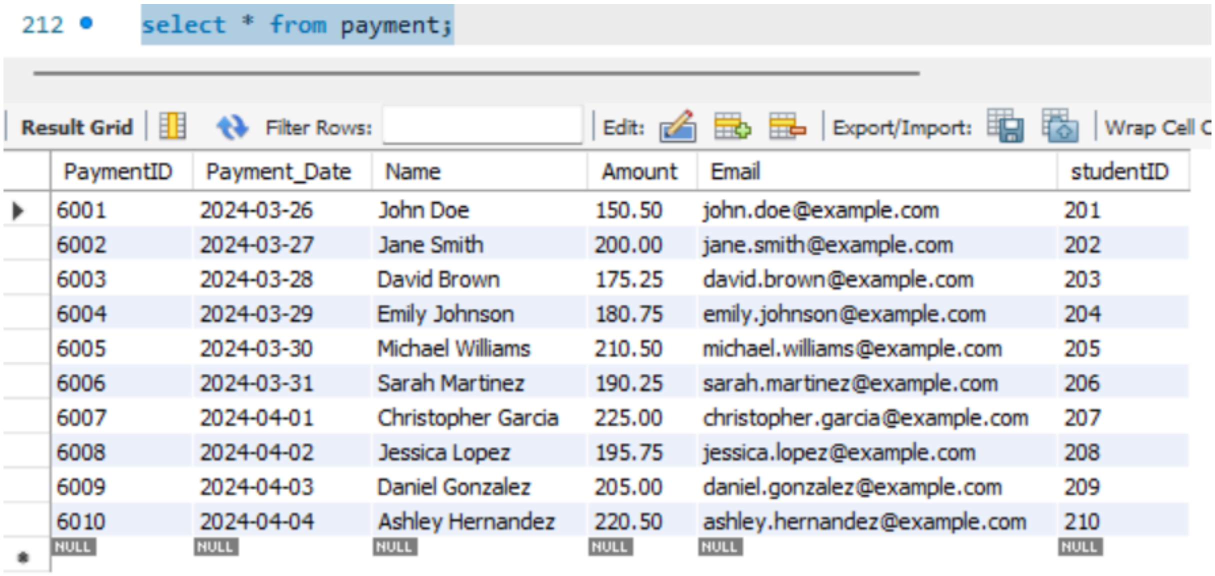
Vehicle_Id	Vehicle_Make	Vehicle_Type	Model_Year	Plate_Number	officeID
501	Toyota	Sedan	2022-01-01	ABC123	1
502	Honda	SUV	2020-05-15	DEF456	2
503	Ford	Truck	2019-07-20	GHI789	3
504	Chevrolet	Van	2021-10-10	JKL012	4
505	BMW	Coupe	2018-12-31	MNO345	5
506	Mercedes-Benz	Convertible	2020-08-25	PQR678	6
507	Audi	Hatchback	2017-04-15	STU901	7
508	Tesla	Electric	2023-03-20	VWX234	8
509	Nissan	Compact	2019-11-11	YZA567	9
510	Kia	Crossover	2022-06-30	BCD890	10
NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Maintenance (vehicle_ID, Vehicle_Type, Date_Serviced,
service_type, Cost, Next_Schedule_Maintenance)
VALUES
    (501, 'Car', '2024-03-25', 'Oil Change', 50.00, '2024-09-25'),
    (502, 'Truck', '2024-03-26', 'Tire Rotation', 80.00, '2024-09-26'),
    (503, 'Motorcycle', '2024-03-27', 'Brake Inspection', 30.00, '2024-09-27'),
    (504, 'SUV', '2024-03-28', 'Battery Replacement', 120.00, '2024-09-28'),
    (505, 'Van', '2024-03-29', 'Engine Tune-up', 100.00, '2024-09-29'),
    (506, 'Hatchback', '2024-03-30', 'Wheel Alignment', 70.00, '2024-09-30'),
    (507, 'Sedan', '2024-03-31', 'Fluid Flush', 90.00, '2024-10-01'),
    (508, 'Convertible', '2024-04-01', 'Spark Plug Replacement', 60.00,
'2024-10-02'),
    (509, 'Minivan', '2024-04-02', 'Air Filter Change', 40.00, '2024-10-03'),
    (510, 'Coupe', '2024-04-03', 'Coolant Flush', 80.00, '2024-10-04');
```



vehicle_ID	Vehicle_Type	Date_Serviced	service_type	Cost	Next_Schedule_Maintenance
501	Car	2024-03-25	Oil Change	50.00	2024-09-25
502	Truck	2024-03-26	Tire Rotation	80.00	2024-09-26
503	Motorcycle	2024-03-27	Brake Inspection	30.00	2024-09-27
504	SUV	2024-03-28	Battery Replacement	120.00	2024-09-28
505	Van	2024-03-29	Engine Tune-up	100.00	2024-09-29
506	Hatchback	2024-03-30	Wheel Alignment	70.00	2024-09-30
507	Sedan	2024-03-31	Fluid Flush	90.00	2024-10-01
508	Convertible	2024-04-01	Spark Plug Replacement	60.00	2024-10-02
509	Minivan	2024-04-02	Air Filter Change	40.00	2024-10-03
510	Coupe	2024-04-03	Coolant Flush	80.00	2024-10-04

```
INSERT INTO Payment (PaymentID, Payment_Date, Name, Amount, Email,
StudentID)
VALUES
   (6001, '2024-03-26', 'John Doe', 150.50, 'john.doe@example.com', 201),
   (6002, '2024-03-27', 'Jane Smith', 200.00, 'jane.smith@example.com', 202),
   (6003, '2024-03-28', 'David Brown', 175.25, 'david.brown@example.com',
203),
   (6004, '2024-03-29', 'Emily Johnson', 180.75,
'emily.johnson@example.com', 204),
   (6005, '2024-03-30', 'Michael Williams', 210.50,
'michael.williams@example.com', 205),
   (6006, '2024-03-31', 'Sarah Martinez', 190.25,
'sarah.martinez@example.com', 206),
   (6007, '2024-04-01', 'Christopher Garcia', 225.00,
'christopher.garcia@example.com', 207),
   (6008, '2024-04-02', 'Jessica Lopez', 195.75,
'jessica.lopez@example.com', 208),
   (6009, '2024-04-03', 'Daniel Gonzalez', 205.00,
'daniel.gonzalez@example.com', 209),
   (6010, '2024-04-04', 'Ashley Hernandez', 220.50,
'ashley.hernandez@example.com', 210);
```



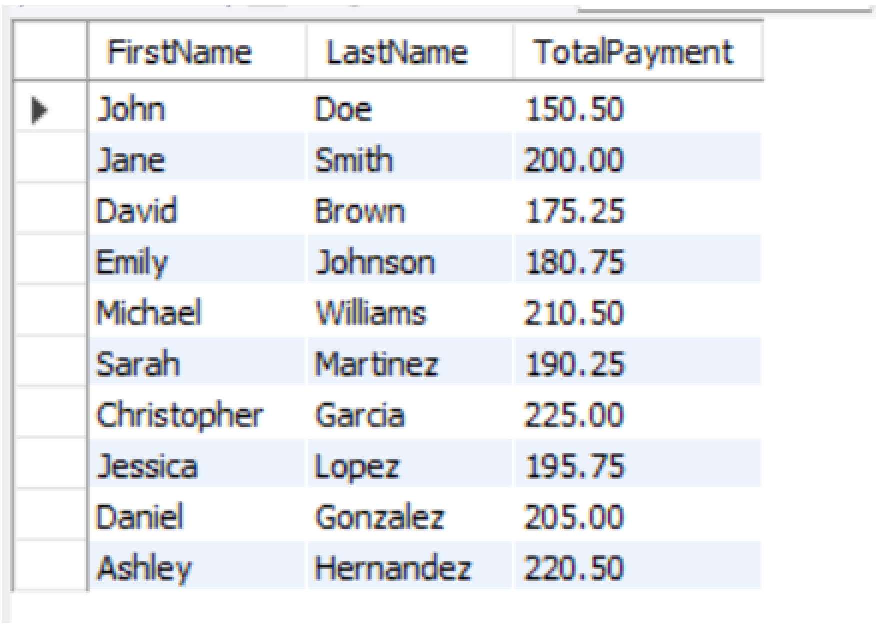
```
select * from payment;
```

PaymentID	Payment_Date	Name	Amount	Email	studentID
6001	2024-03-26	John Doe	150.50	john.doe@example.com	201
6002	2024-03-27	Jane Smith	200.00	jane.smith@example.com	202
6003	2024-03-28	David Brown	175.25	david.brown@example.com	203
6004	2024-03-29	Emily Johnson	180.75	emily.johnson@example.com	204
6005	2024-03-30	Michael Williams	210.50	michael.williams@example.com	205
6006	2024-03-31	Sarah Martinez	190.25	sarah.martinez@example.com	206
6007	2024-04-01	Christopher Garcia	225.00	christopher.garcia@example.com	207
6008	2024-04-02	Jessica Lopez	195.75	jessica.lopez@example.com	208
6009	2024-04-03	Daniel Gonzalez	205.00	daniel.gonzalez@example.com	209
6010	2024-04-04	Ashley Hernandez	220.50	ashley.hernandez@example.com	210
NULL	NULL	NULL	NULL	NULL	NULL

Step 5.4 Write SQL statements that will process five non-routine requests for information from the database just created. For each, write the request in English, followed by the corresponding SQL command.

1. **Find the total payment amount received from each student**

```
SELECT Student.FirstName, Student.LastName,
SUM(Payment.Amount) AS TotalPayment
FROM Student
JOIN Payment ON Student.StudentID = Payment.studentID
GROUP BY Student.StudentID;
```

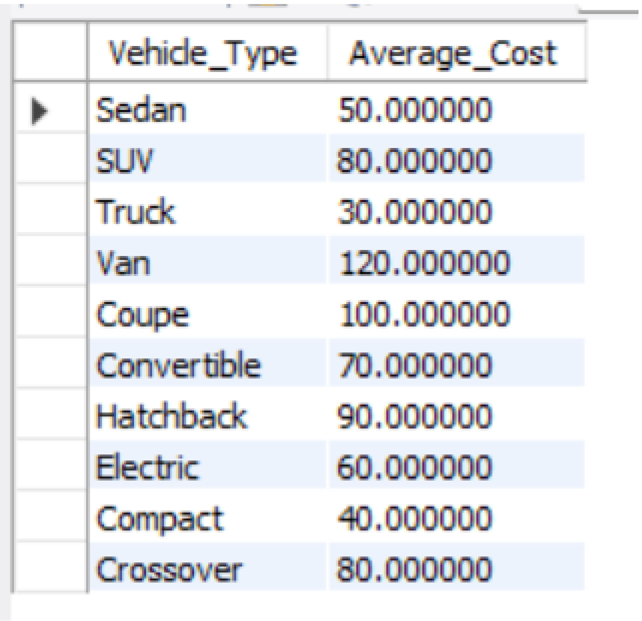


	FirstName	LastName	TotalPayment
▶	John	Doe	150.50
	Jane	Smith	200.00
	David	Brown	175.25
	Emily	Johnson	180.75
	Michael	Williams	210.50
	Sarah	Martinez	190.25
	Christopher	Garcia	225.00
	Jessica	Lopez	195.75
	Daniel	Gonzalez	205.00
	Ashley	Hernandez	220.50

2. **Find the average maintenance cost for each type of vehicle:**

```
SELECT Vehicle.Vehicle_Type, AVG(Maintenance.Cost) AS
Average_Cost
```

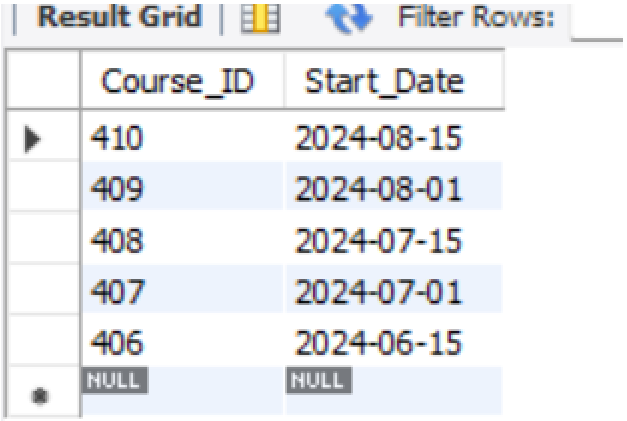
```sql
FROM Vehicle
JOIN Maintenance ON Vehicle.Vehicle_Id = Maintenance.vehicle_ID
GROUP BY Vehicle.Vehicle_Type;
```



Vehicle_Type	Average_Cost
Sedan	50.000000
SUV	80.000000
Truck	30.000000
Van	120.000000
Coupe	100.000000
Convertible	70.000000
Hatchback	90.000000
Electric	60.000000
Compact	40.000000
Crossover	80.000000

3. Retrieve the courses with the most recent start dates:

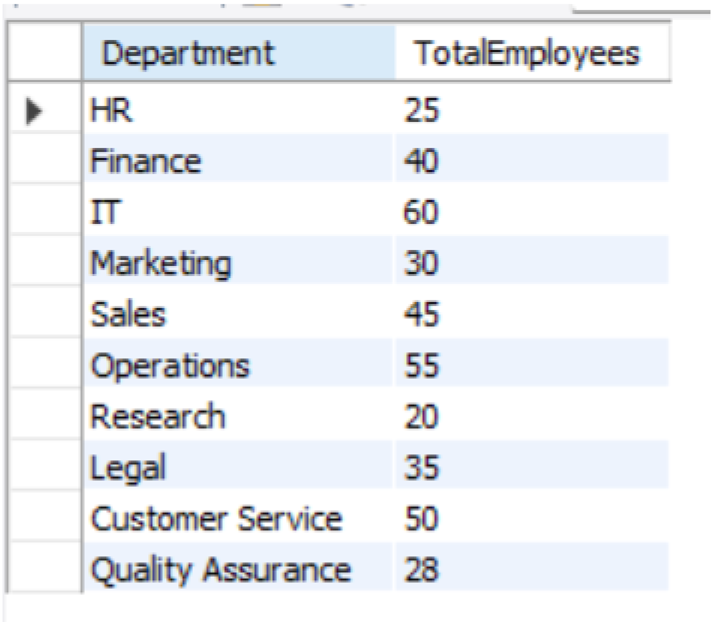
```sql
SELECT Course.Course_ID, Course.Start_Date
FROM Course
ORDER BY Course.Start_Date DESC
LIMIT 5;
```



Course_ID	Start_Date
410	2024-08-15
409	2024-08-01
408	2024-07-15
407	2024-07-01
406	2024-06-15
NULL	NULL

4. Retrieve the total number of employees in each department:

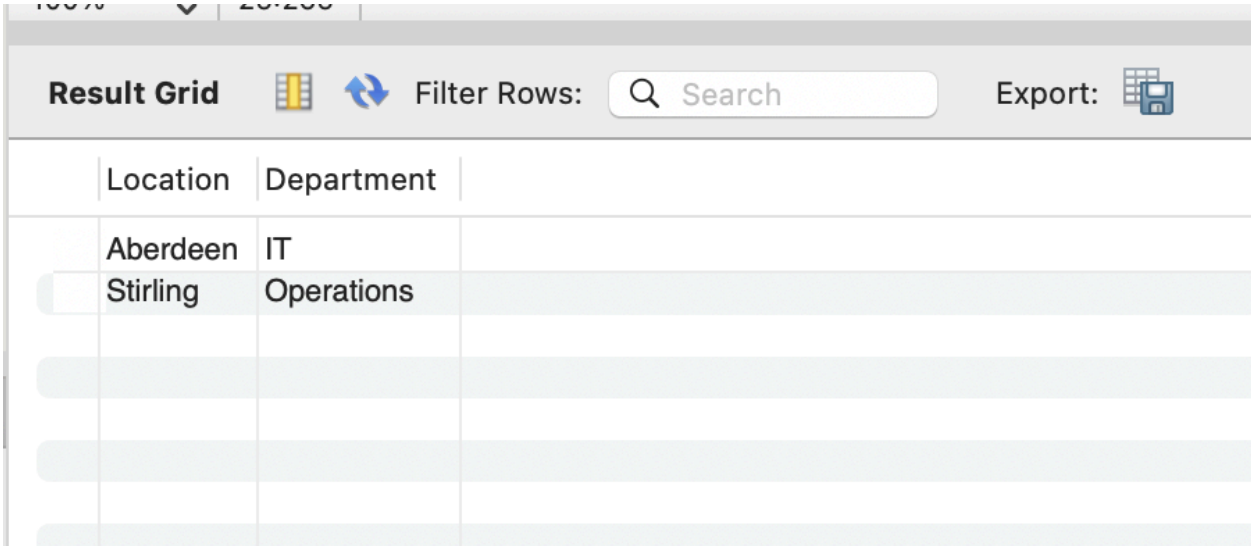
```
SELECT Department, SUM(NoOfEmployee) AS TotalEmployees
FROM Office
GROUP BY Department;
```



Department	TotalEmployees
HR	25
Finance	40
IT	60
Marketing	30
Sales	45
Operations	55
Research	20
Legal	35
Customer Service	50
Quality Assurance	28

5. List all offices located in cities with more than 50 employees:

```
SELECT Location, Department
FROM Office
WHERE NoOfEmployee > 50;
```



Location	Department
Aberdeen	IT
Stirling	Operations