Project

Solution to part One 1 – 4

Tables:

Appointment({PK: staffed, appointmentDate, appointmentTime})

staffID	appointmentDate	appointmentTime	toDo
101	2023-08-10	09:00:00	Interview a new Client
101	2023-08-18	13:00:00	Lesson
102	2023-08-12	14:30:00	take a client to Driving Test
103	2023-08-15	11:00:00	Take a car to Inspection
104	2023-08-20	10:30:00	Lesson
NULL	NULL	NULL	NULL

Branch({PK: branchID})

branchID	address	city
1	123 Main St	Exampleville
2	456 Elm St	Citytown
3	789 Oak Ave	Glasgow
4	567 Pine St	London
5	901 Maple Rd	Manchester
6	234 Cedar Ln	Birmingham
7	678 Birch Blvd	Edinburgh
8	345 Walnut Way	Liverpool
9	123 Cherry Dr	Leeds
10	456 Sycamore Rd	Bristol
11	789 Redwood Ave	Glasgow
12	987 Elm Rd	London
NULL	NULL	NULL

Car({PK: carNo})

carNo	numberFaults	registrationNumber
1	3	ABC123
2	2	XYZ789
3	0	xyz1234
NULL	NULL	NULL

Client({PK: clientNo)

clientNo	name	validProvisionalLicence	specialNeeds	writtenTestPassed	gender	staffID
301	John Doe	Yes	No	Yes	Male	101
302	Jane Smith	Yes	Yes	No	Male	102
303	Michael Johnson	Yes	Yes	Yes	Male	103
304	Emily Brown	No	No	No	Female	102
305	William Wilson	Yes	No	Yes	Male	103
306	Olivia Davis	Yes	Yes	Yes	Female	109
307	James Jones	No	No	No	Male	104
308	Emma Taylor	Yes	No	Yes	Female	101
309	Alexander Martinez	Yes	Yes	Yes	Male	102
310	Sophia Anderson	Yes	No	No	Female	105
314	Bacha	yes	yes	yes	male	104
401	John Doe	Yes	No	Yes	Male	102
402	Jane Smith	Yes	Yes	No	Female	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

DrivingTest({PK: clientNo, date})

clientNo	date	testPassed	reasonForFailing
301	2023-07-15	Yes	NULL
302	2013-01-18	Yes	Incorrect lane changes
303	2020-09-15	No	Failure to yield
303	2021-11-08	No	NULL
303	2022-05-10	No	Speeding
303	2023-06-30	No	NULL
304	2023-02-20	Yes	NULL
305	2021-11-08	Yes	NULL

Inspection({PK: carNo, date})

carNo	date	inspectionResult	faultDescription
1	2023-01-15	failed	engine issue
1	2023-01-17	passed	NULL
2	2023-02-10	passed	NULL
NULL	NULL	NULL	NULL

Interview({PK: date, clientNo})

data	clientNo	InterviewerName
2023-07-20	301	Bacha
2023-07-22	302	Ayantu
NULL	NULL	NULL

Lesson({PK: staffID, clientNo, startDate, time})

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
101	301	2023-08-01	10:00:00	A	1	50	80%	50.00	NULL
101	301	2023-08-08	09:00:00	A	1	100	80%	50.00	2023-08-15
101	301	2023-08-08	12:30:00	A	2	40	100%	45.00	2023-08-15
101	302	2023-08-03	14:30:00	B	2	30	60%	45.00	NULL
102	302	2023-08-08	10:30:00	B	2	75	100%	60.00	2023-08-15
103	303	2023-08-09	11:15:00	C	3	120	60%	45.00	2023-08-09
104	301	2023-08-09	14:00:00	A	4	50	80%	55.00	2023-08-09
105	302	2023-08-10	16:30:00	B	1	90	40%	40.00	2023-08-17
106	303	2023-08-10	08:45:00	C	2	70	100%	65.00	2023-08-10
107	301	2023-08-11	13:00:00	A	3	110	80%	70.00	2023-08-11
108	302	2023-08-11	10:15:00	B	4	30	60%	55.00	2023-08-18
109	303	2023-08-12	14:45:00	C	1	85	40%	60.00	2023-08-12
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Staff({PK: staffID})

staffID	position	carNo	gender	DOB	phoneNo	name	branchID	totalClients
101	Instructor	1	Male	1985-05-10	NULL	NULL	1	0
102	Administrative	NULL	Female	1990-12-15	NULL	NULL	2	0
103	Instructor	2	Male	1993-09-20	123-456-7890	John Doe	3	0
104	Administrative	1	Female	1988-03-12	987-654-3210	Jane Smith	1	1
105	Manager	NULL	Male	1980-11-05	555-123-4567	Michael Johnson	1	0
106	Administrative	1	Female	1995-07-25	888-555-1234	Emily Williams	2	0
107	Instructor	2	Male	1992-01-18	777-222-3333	David Wilson	1	0
108	Manager	NULL	Female	1975-06-30	444-666-8888	Laura Martin	1	0
109	Instructor	3	Male	1987-04-15	333-111-9999	Kevin Turner	1	0
110	Instructor	3	Female	1998-08-08	222-444-6666	Michelle Harris	2	0
111	Manager	NULL	Male	1983-12-01	111-777-5555	Brian Clark	3	0
112	Instructor	2	Female	1991-02-22	999-333-2222	Rachel Lewis	3	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Solution to part One 5: the 1st 8 queries:

a)
SELECT name, phoneNo
from Staff
where position = 'Manager';

name	phoneNo
Michael Johnson	555-123-4567
Laura Martin	444-666-8888
Brian Clark	111-777-5555

b)
SELECT address
FROM Branch
Where city = 'Glasgow';

address
789 Oak Ave
789 Redwood Ave

C)
SELECT name
from Staff s, Branch b
where position = 'Instructor' AND gender = 'Female' AND s.branchId = b.branchID AND
city= 'Glasgow';

name
Rachel Lewis

d)
SELECT branchId, COUNT(*) As 'Total Number'
from Staff
group by branchId;

branchId	Total Number
1	6
2	3
3	3

E)
SELECT b.branchId, count(c.clientNo)
from Branch b, Staff s, Client c
where c.staffID = s.staffId and s.branchId = b.branchID
group by b.branchID;

branchId	count(c.clientNo)
1	6
2	4
3	2

f)

```
SELECT s.name, a.appointmentDate, a.appointmentTime, a.toDo
FROM Appointments a
JOIN Staff s ON a.staffID = s.staffID
WHERE a.appointmentDate BETWEEN CURDATE() + INTERVAL 7 DAY AND CURDATE()
+ INTERVAL 14 DAY;
```

name	appointmentDate	appointmentTime	toDo
Sara Andrew	2023-08-18	13:00:00	Lesson
Jane Smith	2023-08-20	10:30:00	Lesson

g)

```
SELECT v.*, s.staffID, s.name
from Interview v, Client c, Staff s
where v.clientNo = c.clientNo and c.staffID = s.staffID;
```

data	clientNo	InterviewerName	staffID	name
2023-07-20	301	Bacha	101	Sara Andrew
2023-07-22	302	Ayantu	102	Nuru Ahmad

h)

```
SELECT COUNT(c.gender)
from Branch b, Staff s, Lesson l, Client c
where b.branchID = s.branchID AND l.staffID = s.staffID AND c.clientNo
= l.clientNo AND b.city = 'Glasgow'
```

COUNT(c.gender)
1

Solution to part Two: 7 t0 12:

7)

```
DELIMITER //
CREATE PROCEDURE getAllLessons(IN staffNum INT)
BEGIN
   SELECT *
   FROM Lesson l
   WHERE l.staffID = staffNum;
END //getAllLessons
DELIMITER ;

CALL getAllLessons(101);
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
101	301	2023-08-01	10:00:00	A	1	50	80%	50.00	NULL
101	301	2023-08-08	09:00:00	A	1	100	80%	50.00	2023-08-15
101	301	2023-08-08	12:30:00	A	2	40	100%	45.00	2023-08-15
101	302	2023-08-03	14:30:00	B	2	30	60%	45.00	NULL

8)

```
DELIMITER //
create Procedure getLesson(staffNum Int, sDate Date)
BEGIN
SELECT *
from Lesson l
where l.staffID = staffNum and l.startDate = sDate and l.endDate = sDate + interval 7 Day;
END //
DELIMITER ;

call getLesson(101, '2023-08-08');
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
101	301	2023-08-08	09:00:00	A	1	100	80%	50.00	2023-08-15
101	301	2023-08-08	12:30:00	A	2	40	100%	45.00	2023-08-15

8.5 A)

```
DELIMITER //
CREATE PROCEDURE getClientLessons(IN cNum INT)
BEGIN
   SELECT *
   FROM Lesson l
   WHERE l.clientNo = cNum;
END //
DELIMITER ;
```

```
CALL getClientLessons(302);
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
101	302	2023-08-03	14:30:00	B	2	30	60%	45.00	NULL
102	302	2023-08-08	10:30:00	B	2	75	100%	60.00	2023-08-15
105	302	2023-08-10	16:30:00	B	1	90	40%	40.00	2023-08-17
108	302	2023-08-11	10:15:00	B	4	30	60%	55.00	2023-08-18

```
DELIMITER //
create Procedure getClientLessonByDate(cNum Int, sDate Date)
BEGIN
SELECT *
from Lesson l
where l.clientNo = cNum and l.startDate = sDate and l.endDate = sDate + interval 7 Day;
END //
DELIMITER ;
```

```
call getClientLessonByDate(302, '2023-08-08');
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
102	302	2023-08-08	10:30:00	B	2	75	100%	60.00	2023-08-15

8.5 B)

```
DELIMITER //
create Procedure getClientWhoPassed(testDate Date)
BEGIN
SELECT *
from Client l, DrivingTest d
where l.clientNo = d.clientNo and l.startDate = testDate - interval 7 Day;
END //
DELIMITER ;
```

```
call getClientWhoPassed('2023-07-22')
```

clientNo	name	validProvisionalLicence	specialNeeds	writtenTestPassed	gender	staffID	clientNo	date	testPassed	reasonForFailing
301	John Doe	Yes	No	Yes	male	101	301	2023-07-15	Yes	NULL

9 A)
```
CREATE VIEW Client_Lesson  AS
SELECT *
FROM Client c natural join  Lesson l;

select *
from Client_Lesson;
```

clientNo	staffID	name	validProvisionalLicen...	specialNeeds	writtenTestPass...	gender	startDate	time	block	carNo	mileage	progress	fee	endDate
301	101	John Doe	Yes	No	Yes	male	2023-08-01	10:00:00	A	1	50	80%	50.00	NULL
301	101	John Doe	Yes	No	Yes	male	2023-08-08	09:00:00	A	1	100	80%	50.00	2023-08-15
301	101	John Doe	Yes	No	Yes	male	2023-08-08	12:30:00	A	2	40	100%	45.00	2023-08-15
302	102	Jane Smith	Yes	Yes	No	Male	2023-08-08	10:30:00	B	2	75	100%	60.00	2023-08-15
303	103	Michael Johnson	Yes	Yes	Yes	Male	2023-08-09	11:15:00	C	3	120	60%	45.00	2023-08-09

9B)
```
CREATE VIEW Lesson_Info AS
SELECT cl.*, s.staffName
FROM Client_Lesson cl natural join Staff s;

SELECT *
FROM  Lesson  Info;
```

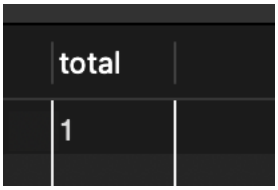
clientNo	staffID	clientName	validProvisionalLicen...	specialNeeds	writtenTestPass...	gender	startDate	time	block	carNo	mileage	progress	fee	endDate	staffName
301	101	John Doe	Yes	No	Yes	male	2023-08-01	10:00:00	A	1	50	80%	50.00	NULL	Sara Andrew
301	101	John Doe	Yes	No	Yes	male	2023-08-08	09:00:00	A	1	100	80%	50.00	2023-08-15	Sara Andrew

10 A)
```
DELIMITER //
CREATE FUNCTION totalLessons()
RETURNS INT
READS SQL DATA
BEGIN
   DECLARE totalLessons INT;
   SET totalLessons = 0;
   SELECT COUNT(*) INTO totalLessons
   FROM Lesson;
   RETURN totalLessons;
END;
//
DELIMITER ;

SELECT totalLessons();
```

totalLessons()
12

10B)
```
DELIMITER //
```

```
CREATE FUNCTION totalLessonsBeforeDate(clientId INT, endDate DATE)
RETURNS INT
READS SQL DATA
BEGIN
   DECLARE totalLessons INT;
   SELECT COUNT(*) INTO totalLessons
   FROM Lesson
   WHERE clientNo = clientId AND startDate < endDate;
   RETURN totalLessons;
END;
//
DELIMITER ;

select totalLessonsBeforeDate(301, '2023-08-08') as total
```



total
1

11)

```
DELIMITER //
CREATE PROCEDURE getClientLessons(clientId INT)
BEGIN
   CREATE TEMPORARY TABLE tempTable AS
   SELECT *
   FROM Client natural join Lesson
   WHERE Client.clientNo = clientId;

   SELECT *
   FROM tempTable;
   DROP TEMPORARY TABLE tempTable;
END //
DELIMITER ;

call getClientLessons(302);
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate
101	302	2023-08-03	14:30:00	B	2	30	60%	45.00	NULL
102	302	2023-08-08	10:30:00	B	2	75	100%	60.00	2023-08-15
105	302	2023-08-10	16:30:00	B	1	90	40%	40.00	2023-08-17
108	302	2023-08-11	10:15:00	B	4	30	60%	55.00	2023-08-18

12) **Triggers on Insert**

```
DROP TRIGGER IF EXISTS TotalClient;
DELIMITER //
CREATE TRIGGER TotalClient AFTER INSERT ON Client
FOR EACH ROW
BEGIN
   UPDATE Staff
   SET totalClients = totalClients + 1
   WHERE staffID = NEW.staffID;
END//
DELIMITER ;
```

@Before insert

staffID	position	carNo	gender	DOB	phoneNo	staffName	branchID	totalClients
101	Instructor	1	Male	1985-05-10	NULL	Sara Andrew	1	0
102	Administrative	NULL	Female	1990-12-15	NULL	Nuru Ahmad	2	0
103	Instructor	2	Male	1993-09-20	123-456-7890	John Doe	3	0
104	Administrative	1	Female	1988-03-12	987-654-3210	Jane Smith	1	1
105	Manager	NULL	Male	1980-11-05	555-123-4567	Michael Johnson	1	0
106	Administrative	1	Female	1995-07-25	888-555-1234	Emily Williams	2	0
107	Instructor	2	Male	1992-01-18	777-222-3333	David Wilson	1	0
108	Manager	NULL	Female	1975-06-30	444-666-8888	Laura Martin	1	0
109	Instructor	3	Male	1987-04-15	333-111-9999	Kevin Turner	1	0
110	Instructor	3	Female	1998-08-08	222-444-6666	Michelle Harris	2	0
111	Manager	NULL	Male	1983-12-01	111-777-5555	Brian Clark	3	0
112	Instructor	2	Female	1991-02-22	999-333-2222	Rachel Lewis	3	0

```
Insert into Client(clientNo, clientName, validProvisionalLicence, specialNeeds,
writtenTestPassed, gender, staffID)
values(315,'Bacha','yes', 'yes','yes', 'male',104);
```

@after insert

staffID	position	carNo	gender	DOB	phoneNo	staffName	branchID	totalClien
101	Instructor	1	Male	1985-05-10	NULL	Sara Andrew	1	0
102	Administrative	NULL	Female	1990-12-15	NULL	Nuru Ahmad	2	0
103	Instructor	2	Male	1993-09-20	123-456-7890	John Doe	3	0
104	Administrative	1	Female	1988-03-12	987-654-3210	Jane Smith	1	2

Triggers on delete.

```
DELIMITER //
CREATE TRIGGER TotalClientOnRemove AFTER Delete ON Client
FOR EACH ROW
BEGIN
   UPDATE Staff
   SET totalClients = totalClients - 1
```

```
    WHERE staffID = Old.staffID;
END //
DELIMITER ;

delete from client
where clientNo = 315;
```

@after deletion

101	Instructor	1	Male	1985-05-10	NULL	Sara Andrew	1	0
102	Administrative	NULL	Female	1990-12-15	NULL	Nuru Ahmad	2	0
103	Instructor	2	Male	1993-09-20	123-456-7890	John Doe	3	0
104	Administrative	1	Female	1988-03-12	987-654-3210	Jane Smith	1	1

Solution to part Two: the last 7 queries:

j)

```
select registrationNumber
from Car
where numberFaults = 0;
```

registrationNumber
xyz1234

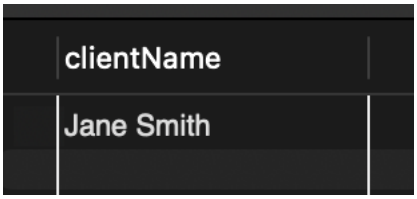
k)

```
select registrationNumber
from Car c, Staff s, Branch b
where c.carNo = s.carNo
and s.position='Instructor'
and s.branchID = b.branchID
and city='Glasgow';
```

registrationNumber
XYZ789
XYZ789

l)

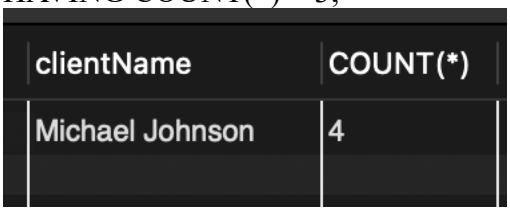
```
select c.clientName
from Client c, DrivingTest d
where c.clientNo = d.clientNo
and d.date Between'2013-01-01' and '2013-01-31'
and d.testPassed = 'Yes';
```



clientName
Jane Smith

m)

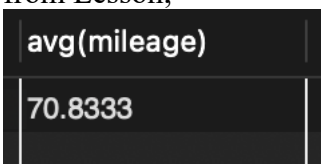
```
SELECT c.clientName, COUNT(*)
FROM Client c
JOIN DrivingTest d ON c.clientNo = d.clientNo
WHERE testPassed = 'No'
GROUP BY c.clientNo, c.clientName
HAVING COUNT(*) > 3;
```



clientName	COUNT(*)
Michael Johnson	4

n)

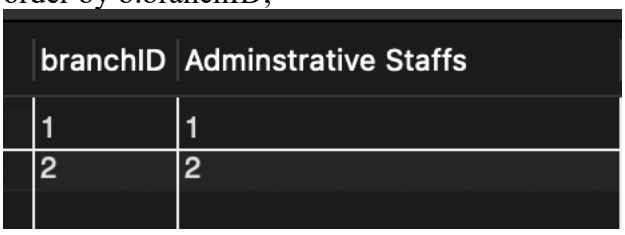
```
select avg(mileage)
from Lesson;
```



avg(mileage)
70.8333

0)

```
select b.branchID, count(*) as 'Adminstrative Staffs'
from Staff s, Branch b
where s.branchID = b.branchID and position='Administrative'
group by b.branchID
order by b.branchID;
```



branchID	Adminstrative Staffs
1	1
2	2

Solution to part Two : 15 Cursors :

```
DELIMITER //
CREATE PROCEDURE UpdateLessonFeesBasedOnMileage()
BEGIN
```

```
    DECLARE done INT DEFAULT 0;
    DECLARE lessonID INT;
    DECLARE mileage DECIMAL(10, 2);
    DECLARE fee DECIMAL(10, 2);

    DECLARE cur CURSOR FOR
    SELECT lessonID, mileage, fee
    FROM Lesson;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO lessonID, mileage, fee;
        IF done THEN
            LEAVE read_loop;
        END IF;

        IF mileage > 30 THEN
            SET fee = fee + 10;
        ELSEIF mileage > 25 THEN
            SET fee = fee + 8;
        ELSEIF mileage > 20 THEN
            SET fee = fee + 5;
        END IF;

        UPDATE Lesson
        SET fee = fee
        WHERE lessonID = lessonID;
    END LOOP;

    CLOSE cur;
END //
DELIMITER ;
```

Using case:

```
SET SQL_SAFE_UPDATES = 0;


DELIMITER //
CREATE PROCEDURE UpdateLessonFeesBasedOnMileage()
BEGIN
```

```
    UPDATE Lesson l
    SET l.fee =
        CASE
            WHEN l.mileage > 30 THEN l.fee + 10
            WHEN l.mileage > 25 THEN l.fee + 8
            WHEN l.mileage > 20 THEN l.fee + 5
            ELSE l.fee
        END;
END //
DELIMITER ;
```

@Before calling ....

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate	LessonID
101	301	2023-08-01	10:00:00	A	1	50.00	80%	70.00	NULL	1
101	301	2023-08-08	09:00:00	A	1	100.00	80%	70.00	2023-08-15	2
101	301	2023-08-08	12:30:00	A	2	40.00	100%	65.00	2023-08-15	3
101	302	2023-08-03	14:30:00	B	2	30.00	60%	46.00	NULL	4
102	302	2023-08-08	10:30:00	B	2	75.00	100%	80.00	2023-08-15	5
103	303	2023-08-09	11:15:00	C	3	120.00	60%	65.00	2023-08-09	6
104	301	2023-08-09	14:00:00	A	4	50.00	80%	110.00	2023-08-09	7
105	302	2023-08-10	16:30:00	B	1	90.00	40%	96.00	2023-08-17	8
106	303	2023-08-10	08:45:00	C	2	70.00	100%	67.00	2023-08-10	9
107	301	2023-08-11	13:00:00	A	3	110.00	80%	118.00	2023-08-11	10
108	302	2023-08-11	10:15:00	B	4	30.00	60%	39.00	2023-08-18	11
109	303	2023-08-12	14:45:00	C	1	85.00	40%	80.00	2023-08-12	12
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

@After calling ....

```
CALL UpdateLessonFeesBasedOnMileage();
```

staffID	clientNo	startDate	time	block	carNo	mileage	progress	fee	endDate	LessonID
101	301	2023-08-01	10:00:00	A	1	50.00	80%	80.00	NULL	1
101	301	2023-08-08	09:00:00	A	1	100.00	80%	80.00	2023-08-15	2
101	301	2023-08-08	12:30:00	A	2	40.00	100%	75.00	2023-08-15	3
101	302	2023-08-03	14:30:00	B	2	30.00	60%	54.00	NULL	4
102	302	2023-08-08	10:30:00	B	2	75.00	100%	90.00	2023-08-15	5
103	303	2023-08-09	11:15:00	C	3	120.00	60%	75.00	2023-08-09	6
104	301	2023-08-09	14:00:00	A	4	50.00	80%	120.00	2023-08-09	7
105	302	2023-08-10	16:30:00	B	1	90.00	40%	106.00	2023-08-17	8
106	303	2023-08-10	08:45:00	C	2	70.00	100%	77.00	2023-08-10	9
107	301	2023-08-11	13:00:00	A	3	110.00	80%	128.00	2023-08-11	10
108	302	2023-08-11	10:15:00	B	4	30.00	60%	47.00	2023-08-18	11
109	303	2023-08-12	14:45:00	C	1	85.00	40%	90.00	2023-08-12	12
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

16)

```
DELIMITER //
CREATE PROCEDURE MyLoopExample()
BEGIN
```

```
    DECLARE Counter INT DEFAULT 1;

    WHILE Counter <= 10 DO
        SET @message = CONCAT('Counter: ', Counter);
        SELECT @message;
        SET Counter = Counter + 1;
    END WHILE;
END //
DELIMITER ;

call  MyLoopExample();
```