Group 4:

Group Members:

1.Zahir Ayub Khan(Group Leader)

2.Jamal khan

3.Nauman Ali

4.Syed Shan E Ali

Table of Contents

Lab 5 Common Solution :

```
create table Student (
ID nchar(30),
Name varchar(30),
);

create table Transcript (
Subject nchar(30),
GPA nchar(30),
ID  nchar(30),
);

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-001', 'ahmad khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-002', 'Hassan ali');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-003', 'Bilal Khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-004', 'Rana Noon');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-005', 'Zoya Yusufzai');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-006', 'Zain Ahmad');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-007', 'Ghulam mujtaba');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-008', 'Hamza ali khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-009', 'Ali Joiya');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-010', 'Alex markovich');


INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Math','2.3','Sp17-bse-001');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('English','3.3','Sp17-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Database system','2.7','Sp17-bse-003');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('OOP','3.7','Sp17-bse-004');
```

```sql
INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Islamic studies','3.0','Sp17-bse-005');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('CA','2.0','Sp17-bse-006');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Advance OOP','2.7','Sp17-bse-007');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('statistics','2.3','Sp17-bse-008');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Cal1','2.3','Sp17-bse-009');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('SQE','2.7','Sp17-bse-010');

/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/



create database StudentTranscriptDB
use StudentTranscriptDB


create table Student(
StudentId nchar(30),
Name(30))

  insert into Student(StudentId,Name) values
  ('fa20-bcs-001','ali')
 insert into Student(StudentId,Name) values
  ('fa20-bcs-002','Aftab')
 insert into Student(StudentId,Name) values
 ('fa20-bcs-003','Ammar')
  insert into Student(StudentId,Name) values
('fa20-bcs-004','Mohsin')
insert into Student(StudentId,Name) values
('fa20-bcs-005','Zubair')
insert into Student(StudentId,Name) values
('fa20-bcs-006','Daniyal')
insert into Student(StudentId,Name) values
('fa20-bcs-007','Usman')
insert into Student(StudentId,Name) values
('fa20-bcs-008','Abbass')
insert into Student(StudentId,Name) values
('fa20-bcs-009','Saddique')
insert into Student(StudentId,Name) values
('fa20-bcs-010','Umar')



create table Transcript(
CourseName nchar(30),
GPA float(30),
StudentId nchar(30)
);

  insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-001')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-002')
 insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','2.7','fa20-bcs-003')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.3','fa20-bcs-004')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','4.0','fa20-bcs-005')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.7','fa20-bcs-006')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','2.0','fa20-bcs-007')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.7','fa20-bcs-008')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.5','fa20-bcs-009')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.0','fa20-bcs-010')


Select Count(*) As [Number of std]
  FROM [StudentTranscriptDB].[dbo].[Transcript]

  Group By CourseName

  Select CourseName, AVG(GPA) As [AVG GPA]
  FROM [StudentTranscriptDB].[dbo].[Transcript]

  Group By CourseName
```

Zahir Ayub Khan:

```
create table Student (
ID nchar(30),
Name varchar(30),
);

create table Transcript (
Subject nchar(30),
GPA nchar(30),
ID  nchar(30),
);

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-001', 'ahmad khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-002', 'Hassan ali');
```

```
INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-003', 'Bilal Khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-004', 'Rana Noon');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-005', 'Zoya Yusufzai');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-006', 'Zain Ahmad');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-007', 'Ghulam mujtaba');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-008', 'Hamza ali khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-009', 'Ali Joiya');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-010', 'Alex markovich');


INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Math','2.3','Sp17-bse-001');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('English','3.3','Sp17-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Database system','2.7','Sp17-bse-003');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('OOP','3.7','Sp17-bse-004');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Islamic studies','3.0','Sp17-bse-005');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('CA','2.0','Sp17-bse-006');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Advance OOP','2.7','Sp17-bse-007');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('statistics','2.3','Sp17-bse-008');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Cal1','2.3','Sp17-bse-009');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('SQE','2.7','Sp17-bse-010');
```

Jamal Khan:

```
create database StudentTranscriptDB
use StudentTranscriptDB


create table Student(
StudentId nchar(30),
Name(30))

  insert into Student(StudentId,Name) values
  ('fa20-bcs-001','ali')
 insert into Student(StudentId,Name) values
  ('fa20-bcs-002','Aftab')
 insert into Student(StudentId,Name) values
 ('fa20-bcs-003','Ammar')
  insert into Student(StudentId,Name) values
('fa20-bcs-004','Mohsin')
insert into Student(StudentId,Name) values
('fa20-bcs-005','Zubair')
insert into Student(StudentId,Name) values
('fa20-bcs-006','Daniyal')
insert into Student(StudentId,Name) values
('fa20-bcs-007','Usman')
insert into Student(StudentId,Name) values
('fa20-bcs-008','Abbass')
insert into Student(StudentId,Name) values
('fa20-bcs-009','Saddique')
insert into Student(StudentId,Name) values
('fa20-bcs-010','Umar')



create table Transcript(
CourseName nchar(30),
GPA float(30),
StudentId nchar(30)
);

  insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-001')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-002')
 insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','2.7','fa20-bcs-003')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.3','fa20-bcs-004')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','4.0','fa20-bcs-005')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.7','fa20-bcs-006')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','2.0','fa20-bcs-007')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.7','fa20-bcs-008')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.5','fa20-bcs-009')
insert into Transcript (CourseName,GPA,StudentId) values
```

```
  ('Database Systems','3.0','fa20-bcs-010')


Select Count(*) As [Number of std]
  FROM [StudentTranscriptDB].[dbo].[Transcript]

  Group By CourseName

  Select CourseName, AVG(GPA) As [AVG GPA]
  FROM [StudentTranscriptDB].[dbo].[Transcript]

  Group By CourseName
```

Update And Delete Common Solution:

```
create table Student (
ID nchar(30),
Name varchar(30),
);

create table Transcript (
Subject nchar(30),
GPA nchar(30),
ID  nchar(30),
);

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-001', 'ahmad khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-002', 'Hassan ali');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-003', 'Bilal Khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-004', 'Rana Noon');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-005', 'Zoya Yusufzai');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-006', 'Zain Ahmad');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-007', 'Ghulam mujtaba');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-008', 'Hamza ali khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-009', 'Ali Joiya');
```

```sql
INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-010', 'Alex markovich');


INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Math','2.3','Sp17-bse-001');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('English','3.3','Sp17-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Database system','2.7','Sp17-bse-003');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('OOP','3.7','Sp17-bse-004');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Islamic studies','3.0','Sp17-bse-005');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('CA','2.0','Sp17-bse-006');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Advance OOP','2.7','Sp17-bse-007');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('statistics','2.3','Sp17-bse-008');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Cal1','2.3','Sp17-bse-009');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('SQE','2.7','Sp17-bse-010');

UPDATE Transcript
Set GPA='2.7'
WHERE ID='Sp17-bse-001';

UPDATE Transcript
Set GPA='1.7'
WHERE ID='Sp17-bse-006';

UPDATE Transcript
Set GPA='2.3'
WHERE ID='Sp17-bse-010';

DELETE FROM Transcript WHERE ID='Sp17-bse-009';
DELETE FROM Transcript WHERE ID='Sp17-bse-008';


/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/


create database StudentTranscriptDB
use StudentTranscriptDB
```

```
create table Student(
StudentId nchar(30),
Name(30))

  insert into Student(StudentId,Name) values
  ('fa20-bcs-001','ali')
 insert into Student(StudentId,Name) values
  ('fa20-bcs-002','Aftab')
 insert into Student(StudentId,Name) values
 ('fa20-bcs-003','Ammar')
  insert into Student(StudentId,Name) values
('fa20-bcs-004','Mohsin')
insert into Student(StudentId,Name) values
('fa20-bcs-005','Zubair')
insert into Student(StudentId,Name) values
('fa20-bcs-006','Daniyal')
insert into Student(StudentId,Name) values
('fa20-bcs-007','Usman')
insert into Student(StudentId,Name) values
('fa20-bcs-008','Abbass')
insert into Student(StudentId,Name) values
('fa20-bcs-009','Saddique')
insert into Student(StudentId,Name) values
('fa20-bcs-010','Umar')



create table Transcript(
CourseName nchar(30),
GPA float(30),
StudentId nchar(30)
);

  insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-001')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-002')
 insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','2.7','fa20-bcs-003')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.3','fa20-bcs-004')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','4.0','fa20-bcs-005')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.7','fa20-bcs-006')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','2.0','fa20-bcs-007')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.7','fa20-bcs-008')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.5','fa20-bcs-009')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.0','fa20-bcs-010')



  DELETE FROM Transcript WHERE StudentId='fa20-bcs-008';
  select * from Transcript

  UPDATE Transcript set GPA='3.3'WHERE StudentId='fa20-bcs-010';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-002';
  select * from Transcript

  UPDATE Transcript set GPA='3.3'WHERE StudentId='fa20-bcs-004';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-001';
  select * from Transcript

  UPDATE Transcript set GPA='1.3'WHERE StudentId='fa20-bcs-003';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-006';
  select * from Transcript

  UPDATE Transcript set GPA='0'WHERE StudentId='fa20-bcs-009';
  select * from Transcript
```

Zahir Ayub Khan:

```
create table Student (
ID nchar(30),
Name varchar(30),
);

create table Transcript (
Subject nchar(30),
GPA nchar(30),
ID  nchar(30),
);

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-001', 'ahmad khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-002', 'Hassan ali');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-003', 'Bilal Khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-004', 'Rana Noon');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-005', 'Zoya Yusufzai');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-006', 'Zain Ahmad');
```

```
INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-007', 'Ghulam mujtaba');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-008', 'Hamza ali khan');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-009', 'Ali Joiya');

INSERT INTO Student (ID, Name)
VALUES ('Sp17-bse-010', 'Alex markovich');


INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Math','2.3','Sp17-bse-001');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('English','3.3','Sp17-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Database system','2.7','Sp17-bse-003');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('OOP','3.7','Sp17-bse-004');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Islamic studies','3.0','Sp17-bse-005');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('CA','2.0','Sp17-bse-006');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Advance OOP','2.7','Sp17-bse-007');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('statistics','2.3','Sp17-bse-008');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('Cal1','2.3','Sp17-bse-009');

INSERT INTO Transcript(Subject,GPA,ID)
VALUES ('SQE','2.7','Sp17-bse-010');

UPDATE Transcript
Set GPA='2.7'
WHERE ID='Sp17-bse-001';

UPDATE Transcript
Set GPA='1.7'
WHERE ID='Sp17-bse-006';

UPDATE Transcript
Set GPA='2.3'
WHERE ID='Sp17-bse-010';

DELETE FROM Transcript WHERE ID='Sp17-bse-009';
DELETE FROM Transcript WHERE ID='Sp17-bse-008';
```

Jamal Khan:

```
create database StudentTranscriptDB
use StudentTranscriptDB


create table Student(
StudentId nchar(30),
Name(30))

  insert into Student(StudentId,Name) values
  ('fa20-bcs-001','ali')
 insert into Student(StudentId,Name) values
  ('fa20-bcs-002','Aftab')
 insert into Student(StudentId,Name) values
 ('fa20-bcs-003','Ammar')
  insert into Student(StudentId,Name) values
('fa20-bcs-004','Mohsin')
insert into Student(StudentId,Name) values
('fa20-bcs-005','Zubair')
insert into Student(StudentId,Name) values
('fa20-bcs-006','Daniyal')
insert into Student(StudentId,Name) values
('fa20-bcs-007','Usman')
insert into Student(StudentId,Name) values
('fa20-bcs-008','Abbass')
insert into Student(StudentId,Name) values
('fa20-bcs-009','Saddique')
insert into Student(StudentId,Name) values
('fa20-bcs-010','Umar')



create table Transcript(
CourseName nchar(30),
GPA float(30),
StudentId nchar(30)
);

  insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-001')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','1.3','fa20-bcs-002')
 insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','2.7','fa20-bcs-003')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.3','fa20-bcs-004')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database System','4.0','fa20-bcs-005')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.7','fa20-bcs-006')
```

```
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','2.0','fa20-bcs-007')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.7','fa20-bcs-008')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Operating Systems','2.5','fa20-bcs-009')
insert into Transcript (CourseName,GPA,StudentId) values
  ('Database Systems','3.0','fa20-bcs-010')
```

```
  DELETE FROM Transcript WHERE StudentId='fa20-bcs-008';
  select * from Transcript

  UPDATE Transcript set GPA='3.3'WHERE StudentId='fa20-bcs-010';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-002';
  select * from Transcript

  UPDATE Transcript set GPA='3.3'WHERE StudentId='fa20-bcs-004';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-001';
  select * from Transcript

  UPDATE Transcript set GPA='1.3'WHERE StudentId='fa20-bcs-003';
  select * from Transcript

  DELETE FROM Transcript WHERE StudentId='fa20-bcs-006';
  select * from Transcript

  UPDATE Transcript set GPA='0'WHERE StudentId='fa20-bcs-009';
  select * from Transcript
```

Lab 6 Common Solution:

```
select * from branch;
INSERT into Branch (branchNo, street, city, postcode) VALUES ('B005',
 'H#88 j-13/3', 'ABT', '54000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B006',
 'H#77 k-14/6', 'KARI', '56000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B007',
 'H#89 m-11/7', 'QUA', '63000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B008',
 'H#69 l-10/2', 'ISL', '52200');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B009',
 'H#35 l-61/2', 'HAR', '73000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0010',
 'H#66 l-01/5', 'MUL', '32100');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0011',
 'H#99 l-11/3', 'ABT', '53300');
```

```sql
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0012',
 'H#12 I-10/2', 'SWA', '57700');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0013',
 'H#19 I-13/6', 'KARI', '56600');
  INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0014',
 'H#35 I-62/2', 'HAR', '73800');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0015',
 'H#66 I-02/5', 'MUL', '23700');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0016',
 'H#99 I-17/3', 'ABT', '81900');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0017',
 'H#12 I-80/6', 'SWA', '65100');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0018',
 'H#19 I-93/6', 'KARI', '89100');

 /* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/



create database DreamHome;
use Dreamhome;
create table Branch(branchNo varchar(20) NOT NULL PRIMARY KEY, street varchar(50)
NOT NULL, city varchar(50) NOT NULL,
postcode varchar(20) NOT NULL);


INSERT into Branch (branchNo, street, city, postcode) VALUES
  ('B0019','X#11 Y-11/1', 'ABBOTTABAD', '22500');
  INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0020','X#77 Y-22/2', 'MANSERA', '23400');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0021','X#89 Y-33/3', 'PESHAWER', '24500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0022','X#69 Y-44/4', 'MARDAN', '26500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0023','X#35 Y-55/5', 'KOHAT', '27500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0024','X#66 Y-66/6', 'NOWSHERA', '28500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0025','X#99 Y-77/7', 'SWABI', '29500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0026','X#12 Y-88/8', 'ISLAMABAD', '31500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0027','X#19 Y-99/9', 'RAWALPINDI', '32500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0028','X#35 Y-12/1', 'LAHORE', '33500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0029','X#66 Y-13/2', 'KARACHI', '34500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0030','X#99 Y-14/3', 'MULTAN', '35500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0031','X#12 Y-15/4', 'HYDERABAD', '36500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0032','X#19 Y-16/5', 'OKHARA', '37500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0033','X#35 Y-12/1', 'LAHORE', '8000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0034','X#66 Y-13/2', 'KARACHI', '9000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0035','X#99 Y-14/3', 'MULTAN', '11000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0036','X#12 Y-15/4', 'HYDERABAD', '15000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0037','X#19 Y-16/5', 'OKHARA', '34500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0038','X#35 Y-12/1', 'LAHORE', '1000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0039','X#66 Y-13/2', 'KARACHI', '19000');
```

Zahir Ayub Khan:

```
select * from branch;
INSERT into Branch (branchNo, street, city, postcode) VALUES ('B005',
 'H#88 j-13/3', 'ABT', '54000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B006',
 'H#77 k-14/6', 'KARI', '56000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B007',
 'H#89 m-11/7', 'QUA', '63000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B008',
 'H#69 I-10/2', 'ISL', '52200');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B009',
 'H#35 I-61/2', 'HAR', '73000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0010',
 'H#66 I-01/5', 'MUL', '32100');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0011',
 'H#99 I-11/3', 'ABT', '53300');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0012',
 'H#12 I-10/2', 'SWA', '57700');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0013',
 'H#19 I-13/6', 'KARI', '56600');
  INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0014',
 'H#35 I-62/2', 'HAR', '73800');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0015',
 'H#66 I-02/5', 'MUL', '23700');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0016',
 'H#99 I-17/3', 'ABT', '81900');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0017',
 'H#12 I-80/6', 'SWA', '65100');
 INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0018',
 'H#19 I-93/6', 'KARI', '89100');
```

Jamal Khan:

```
create database DreamHome;
use Dreamhome;
create table Branch(branchNo varchar(20) NOT NULL PRIMARY KEY, street varchar(50)
NOT NULL, city varchar(50) NOT NULL,
postcode varchar(20) NOT NULL);


INSERT into Branch (branchNo, street, city, postcode) VALUES
  ('B0019','X#11 Y-11/1', 'ABBOTTABAD', '22500');
  INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0020','X#77 Y-22/2', 'MANSERA', '23400');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0021','X#89 Y-33/3', 'PESHAWER', '24500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0022','X#69 Y-44/4', 'MARDAN', '26500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0023','X#35 Y-55/5', 'KOHAT', '27500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0024','X#66 Y-66/6', 'NOWSHERA', '28500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0025','X#99 Y-77/7', 'SWABI', '29500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0026','X#12 Y-88/8', 'ISLAMABAD', '31500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0027','X#19 Y-99/9', 'RAWALPINDI', '32500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0028','X#35 Y-12/1', 'LAHORE', '33500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0029','X#66 Y-13/2', 'KARACHI', '34500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0030','X#99 Y-14/3', 'MULTAN', '35500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0031','X#12 Y-15/4', 'HYDERABAD', '36500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0032','X#19 Y-16/5', 'OKHARA', '37500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0033','X#35 Y-12/1', 'LAHORE', '8000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0034','X#66 Y-13/2', 'KARACHI', '9000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0035','X#99 Y-14/3', 'MULTAN', '11000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0036','X#12 Y-15/4', 'HYDERABAD', '15000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0037','X#19 Y-16/5', 'OKHARA', '34500');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0038','X#35 Y-12/1', 'LAHORE', '1000');
 INSERT into Branch (branchNo, street, city, postcode) VALUES
 ('B0039','X#66 Y-13/2', 'KARACHI', '19000');
```

Lab 7 Common Solution:

Q1: Print the list of postcodes without any repetition

```
      Select distinct(postcode) from Branch;
```

Q2: Print all fName from Staff without repetition

```
       Select distinct(fName) from Staff;
```

Q3: List all staff with renaming all its columns in results

```
       Select staffNo as ID, fName as FirstNAme, lName as LastName, [position]
as       Allocation, sex as Gender, DOB as Birth, salary as Wages, branchNo as
Branch from Staff;
```

Q4: List all clients with re-naming all its columns to synonyms.

```
     Select clientNo as StakeHolderID, fName as
FirstName, lName as LastName, telNo as PhoneNo, prefType as
Preference, maxRent as MaximumRent from  Client;
```

Q5: List all staff with a salary greater than 10,000.

```
   Select salary from Staff where salary >10000;
```

Q6: List all managers and supervisors.

```
Select [position] from Staff Where [position] = 'supervisor' OR [position]
='manager'

/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

Q1: Print the list of postcodes without any repetition

ANS: Select distinct(postcode) from Branch;

Q2: Print all fName from Staff without repetition

ANS: Select distinct(fName) from Staff;

Q3: List all staff with renaming all its columns in results

ANS: Select staffNo as ID, fName as FirstNAme, lName as LastName, [position] as Allocation, sex as Gender, DOB as Birth, salary as Wages, branchNo as Branch from Staff;

Q4: List all clients with re-naming all its columns to synonyms.

ANS: SELECT clientNo as StakeHolderID, fName as FirstName, lName as LastName, telNo as PhoneNo, prefType as Preference, maxRent as MaximumRent
FROM Client;

Q5: List all staff with a salary greater than 10,000.
ANS SELECT salary FROM Staff where salary >10000;
Q6: List all managers and supervisors.
ANS: SELECT [position] FROM Staff Where [position] = 'supervisor' OR [position] ='manager'

Zahir Ayub khan:

Q1: Print the list of postcodes without any repetition
Select distinct(postcode) from Branch;
Q2: Print all fName from Staff without repetition
Select distinct(fName) from Staff;
Q3: List all staff with renaming all its columns in results
Select staffNo as ID, fName as FirstNAme, lName as LastName, [position] as Allocation, sex as Gender, DOB as Birth, salary as Wages, branchNo as Branch from Staff;
Q4: List all clients with re-naming all its columns to synonyms.
Select clientNo as StakeHolderID, fName as FirstName, lName as LastName, telNo as PhoneNo, prefType as Preference, maxRent as MaximumRent from Client;
Q5: List all staff with a salary greater than 10,000.
Select salary from Staff where salary >10000;
Q6: List all managers and supervisors.
Select [position] from Staff Where [position] = 'supervisor' OR [position] ='manager'

Jamal Khan:

Q1: Print the list of postcodes without any repetition
ANS: Select distinct(postcode) from Branch;
Q2: Print all fName from Staff without repetition
ANS: Select distinct(fName) from Staff;

Q3: List all staff with renaming all its columns in results

```
ANS: Select staffNo as ID, fName as FirstNAme, lName
as LastName, [position] as Allocation, sex as Gender,
DOB as Birth, salary as Wages, branchNo as Branch
from Staff;
```

Q4: List all clients with re-naming all its columns to synonyms.

```
ANS: SELECT clientNo as StakeHolderID, fName as
FirstName, lName as LastName, telNo as
PhoneNo, prefType as Preference, maxRent as
MaximumRent FROM Client;
```

Q5: List all staff with a salary greater than 10,000.

```
ANS SELECT salary FROM Staff where salary >10000;
```

Q6: List all managers and supervisors.

```
 ANS: SELECT [position] FROM Staff Where [position] =
'supervisor' OR [position] ='manager'
```

LAB 8 Common Solution:

Q1:

```
select staffNo,fName,lName,salary from staff order by salary desc
```

Q2:

```
select propertyNo,type,rooms,rent from PropertyForRent order by type
select propertyNo,type,rooms,rent from PropertyForRent order by type,rent desc
```

Q3:

```
select count(*) as myCount from PropertyForRent where rent<=500
```

Q4:

```
 select count(Distinct propertyNo) As myCount from
Viewing where viewDate BETWEEN '1-May-04' AND '31-May-04';
```

Q5:

```
 select count(staffNo) as myCount,sum(salary) as mySalary from staff where
position='Manager'
```

Q6:

```
 select MIN(salary) as myMin, MAX(salary) as MyMax, AVG(salary)
as myAVG from Staff
```

Q7:

```
 select staffNo, fName, lName, position,
salary from Staff where (select AVG(salary) from Staff) < salary;
```

Q8:

```
select *from Staff where salary> any(select salary from Staff
where branchNo='B002')
```

Qno9:-

```
 select *from Staff where salary> all(select salary from Staff
where branchNo='B002')
```

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

Q1:

Ans: select staffNo,fName,lName,salary from staff order by salary desc

Q2:

Ans: select propertyNo,type,rooms,rent from PropertyForRent order by type select propertyNo,type,rooms,rent from PropertyForRent order by type,rent desc

Q3:

Ans: select count(*) as myCount from PropertyForRent where rent<=500

Q4:

Ans: select count(Distinct propertyNo) As myCount from Viewing WHERE viewDate BETWEEN '1-May-04' AND '31-May-04';

Q5:

Ans: select count(staffNo) as myCount,sum(salary) as mySalary from staff where position='Manager'

Q6:

Ans: select MIN(salary) as myMin, MAX(salary) as myMax, AVG(salary) as myAVG from Staff

Q7:

Ans: SELECT staffNo, fName, lName, position, salary FROM Staff WHERE (SELECT AVG(salary) FROM Staff) < salary;

Q8:

Ans: select *from Staff where salary> any(select salary from Staff where branchNo='B003')

Q9:

Ans: select *from Staff where salary> all(select salary from Staff where branchNo='B003')

Zahir Ayub Khan:

Q1:

select staffNo,fName,lName,salary from staff order by salary desc

Q2:

select propertyNo,type,rooms,rent from PropertyForRent order by type
select propertyNo,type,rooms,rent from PropertyForRent order by type,rent desc

Q3:

select count(*) as myCount from PropertyForRent where rent<=500

Q4:

select count(Distinct propertyNo) As myCount from
Viewing where viewDate BETWEEN '1-May-04' AND '31-May-04';

Q5:

select count(staffNo) as myCount,sum(salary) as mySalary from staff where position='Manager'

Q6:

select MIN(salary) as myMin, MAX(salary) as MyMax, AVG(salary) as myAVG from Staff

Q7:

```
 select staffNo, fName, lName, position,
salary from Staff where (select AVG(salary) from Staff) < salary;
```

Q8:

```
select *from Staff where salary> any(select salary from Staff
where branchNo='B002')
```

Qno9:-

```
 select *from Staff where salary> all(select salary from Staff
where branchNo='B002')
```

Jamal Khan:

Q1:

Ans: select staffNo,fName,lName,salary from staff order by salary desc

Q2:

Ans: select propertyNo,type,rooms,rent from PropertyForRent order by type select propertyNo,type,rooms,rent from PropertyForRent order by type,rent desc

Q3:

Ans: select count(*) as myCount from PropertyForRent where rent<=500

Q4:

Ans: select count(Distinct propertyNo) As myCount from Viewing WHERE viewDate BETWEEN '1-May-04' AND '31-May-04';

Q5:

Ans: select count(staffNo) as myCount,sum(salary) as mySalary from staff where position='Manager'

Q6:

Ans: select MIN(salary) as myMin, MAX(salary) as myMax, AVG(salary) as myAVG from Staff

Q7:

Ans: SELECT staffNo, fName, lName, position, salary FROM Staff WHERE (SELECT AVG(salary) FROM Staff) < salary;

Q8:

Ans: select *from Staff where salary> any(select salary from Staff where branchNo='B003')

Q9:

Ans: select *from Staff where salary> all(select salary from Staff where branchNo='B003')

LAB 9 Common Solution:

```
CREATE DATABASE employeese;


SELECT FIRST_NAME, LAST_NAME, SALARY

FROM employees

WHERE SALARY >

(SELECT salary FROM employees WHERE last_name = 'popp');



SELECT first_name, last_name

FROM employees

WHERE department_id

IN (SELECT department_id FROM departments WHERE
department_name='IT');
```

```
 /* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
Q1:
SELECT * FROM employees;
```

Q2:

```
SELECT FIRST_NAME, LAST_NAME, SALARY
FROM employees
WHERE SALARY >
(SELECT salary FROM employees WHERE last_name = 'Bull');
```

Q3:

```
SELECT first_name, last_name
FROM employees
WHERE department_id
IN (SELECT department_id FROM departments WHERE
department_name='IT');
```

Jamal Khan:

Q1:

```
SELECT * FROM employees;
```

Q2:

```
SELECT FIRST_NAME, LAST_NAME, SALARY
FROM employees
WHERE SALARY >
(SELECT salary FROM employees WHERE last_name = 'Bull');
```

Q3:

```
SELECT first_name, last_name
FROM employees
WHERE department_id
IN (SELECT department_id FROM departments WHERE
department_name='IT');
```

Zahir Ayub Khan:

```
CREATE DATABASE employeese;
```

```
SELECT FIRST_NAME, LAST_NAME, SALARY

FROM employees

WHERE SALARY >

(SELECT salary FROM employees WHERE last_name = 'popp');
```

```
SELECT first_name, last_name

FROM employees

WHERE department_id

IN (SELECT department_id FROM departments WHERE department_name='IT');
```

LAB 10 Common Solution:

1.1

```
SELECT first_name, last_name FROM employees

WHERE manager_id in (select employee_id

FROM employees WHERE department_id

IN (SELECT department_id FROM departments WHERE location_id

IN (select location_id from locations where country_id='US')));
```

1.2

```
SELECT first_name, last_name

FROM employees

WHERE (employee_id IN (SELECT manager_id FROM employees));
```

1.3

```
SELECT first_name, last_name, salary FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);
```

1.4

```
SELECT first_name, last_name, salary

FROM employees

WHERE employees.salary = (SELECT min_salary

FROM jobs

WHERE employees.job_id = jobs.job_id);
```

1.5

```
SELECT first_name, last_name, salary

FROM employees

WHERE department_id IN

(SELECT department_id FROM departments WHERE department_name LIKE 'IT')

AND salary > (SELECT avg(salary) FROM employees);
```

1.6

```
SELECT first_name, last_name, salary

FROM employees

WHERE salary >

(SELECT salary FROM employees WHERE last_name = 'Bell') ORDER BY first_name;
```

1.7

```
SELECT * FROM employees

WHERE salary = (SELECT MIN(salary) FROM employees);
```

1.8

```
SELECT * FROM employees

WHERE salary >

ALL(SELECT avg(salary)FROM employees GROUP BY department_id);
```

1.12

```
SELECT employee_id, first_name, last_name,

(SELECT department_name FROM departments d

 WHERE e.department_id = d.department_id) department

 FROM employees e ORDER BY department;
```

1.13

```
SELECT employee_id FROM employees

WHERE employee_id%2 = 0;
```

1.14

```
SELECT DISTINCT salary

FROM employees e1

WHERE 5 = (SELECT COUNT(DISTINCT salary)

FROM employees  e2
WHERE e2.salary >= e1.salary);
```

1.15

```
SELECT DISTINCT salary
FROM employees e1
WHERE 4 = (SELECT COUNT(DISTINCT salary)
FROM employees  e2
WHERE e2.salary <= e1.salary);
```

1.16

```
SELECT * FROM (
SELECT * FROM employees ORDER BY employee_id DESC
LIMIT 10) sub
ORDER BY employee_id ASC;
```

1.17

```
SELECT * FROM departments
WHERE department_id
NOT IN (select department_id FROM employees);
```

1.18

```
SELECT DISTINCT salary
FROM employees a
WHERE 3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary >= a.salary)
```

```
ORDER BY a.salary DESC;
```

1.19

```
SELECT DISTINCT salary
FROM employees a
WHERE  3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary <= a.salary)
ORDER BY a.salary DESC;
```

1.20

```
SELECT *
FROM employees emp1
WHERE (1) = (
SELECT COUNT(DISTINCT(emp2.salary))
FROM employees emp2
WHERE emp2.salary > emp1.salary);
```

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

Q1:

```
SELECT first_name, last_name FROM employees
WHERE manager_id in (select employee_id
FROM employees WHERE department_id
IN (SELECT department_id FROM departments WHERE location_id
IN (select location_id from locations where
country_id='US')));
```

Q2:

```
SELECT first_name, last_name
FROM employees
WHERE (employee_id IN (SELECT manager_id FROM employees));
```

Q3:

```
SELECT first_name, last_name, salary FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

Q4:

```
SELECT first_name, last_name, salary
FROM employees
WHERE employees.salary = (SELECT min_salary
FROM jobs
WHERE employees.job_id = jobs.job_id);
```

Q5:

```
SELECT first_name, last_name, salary
FROM employees
WHERE department_id IN
(SELECT department_id FROM departments WHERE department_name
LIKE 'IT%')
AND salary > (SELECT avg(salary) FROM employees);
```

Q6:

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary >
(SELECT salary FROM employees WHERE last_name = 'Bell') ORDER
BY first_name;
```

Q7:

```
SELECT * FROM employees
WHERE salary = (SELECT MIN(salary) FROM employees);
```

Q8: Write a query to find the names (first_name, last_name), the salary of the em-ployees whose salary greater than the average salary of all departments?

```
SELECT * FROM employees
WHERE salary >
ALL(SELECT avg(salary)FROM employees GROUP BY department_id);
```

Q9: Write a query to find the names (first_name, last_name) and salary of the em-ployees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = ’SH_CLERK’). Sort the results of the salary of the lowest to highest.

```
SELECT first_name,last_name, job_id, salary
FROM employees
WHERE salary >
ALL (SELECT salary FROM employees WHERE job_id = 'SH_CLERK')
```

```
ORDER BY salary;
```

Q10: Write a query to find the names (first_name, last_name) of the employees who are not supervisors?

```
SELECT b.first_name,b.last_name
FROM employees b
WHERE NOT EXISTS (SELECT 'X' FROM employees a WHERE
a.manager_id = b.employee_id);
```

Q11: Write a query to display the employee ID, first name, last names, and depart-ment names of all employees.

```
SELECT employee_id, first_name, last_name,
(SELECT department_name FROM departments d
 WHERE e.department_id = d.department_id) department
 FROM employees e ORDER BY department;
```

Q12: Write a query to display the employee ID, first name, last names, salary of all employees whose salary is above average for their departments.

```
SELECT employee_id, first_name
FROM employees AS A
WHERE salary >
(SELECT AVG(salary) FROM employees WHERE department_id =
A.department_id);
```

Q13: Write a query to fetch even numbered records from employees table.

```
SET @i = 0;
SELECT i, employee_id
FROM (SELECT @i := @i + 1 AS i, employee_id FROM employees)
a WHERE MOD(a.i, 2) = 0;
```

Q14: Write a query to find the 5th maximum salary in the employees table.

```
SELECT DISTINCT salary
FROM employees e1
WHERE 5 = (SELECT COUNT(DISTINCT salary)
FROM employees  e2
WHERE e2.salary >= e1.salary);
```

Q15: Write a query to find the 4th minimum salary in the employees table.

```
SELECT DISTINCT salary
FROM employees e1
WHERE 4 = (SELECT COUNT(DISTINCT salary)
FROM employees  e2
WHERE e2.salary <= e1.salary);
```

Q16: Write a query to select last 10 records from a table.

```
SELECT * FROM (
SELECT * FROM employees ORDER BY employee_id DESC LIMIT 10)
sub
ORDER BY employee_id ASC;
```

Q17: Write a query to list department number, name for all the departments in which there are no employees in the department.

```
SELECT * FROM departments
WHERE department_id
NOT IN (select department_id FROM employees);
```

Q18: Write a query to get 3 maximum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE 3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary >= a.salary)
ORDER BY a.salary DESC;
```

Q19: Write a query to get 3 minimum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE  3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary <= a.salary)
ORDER BY a.salary DESC;
```

Q20: Write a query to get nth max salaries of employees.Further practice with nested queries

```
SELECT *
FROM employees emp1
WHERE (1) = (
SELECT COUNT(DISTINCT(emp2.salary))
FROM employees emp2
WHERE emp2.salary > emp1.salary);
```

Zahir Ayub Khan:

Jamal Khan:

Q1:

```
SELECT first_name, last_name FROM employees
WHERE manager_id in (select employee_id
FROM employees WHERE department_id
IN (SELECT department_id FROM departments WHERE location_id
IN (select location_id from locations where
country_id='US')));
```

Q2:

```
SELECT first_name, last_name
FROM employees
```

```
WHERE (employee_id IN (SELECT manager_id FROM employees));
```

Q3:

```
SELECT first_name, last_name, salary FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

Q4:

```
SELECT first_name, last_name, salary
FROM employees
WHERE employees.salary = (SELECT min_salary
FROM jobs
WHERE employees.job_id = jobs.job_id);
```

Q5:

```
SELECT first_name, last_name, salary
FROM employees
WHERE department_id IN
(SELECT department_id FROM departments WHERE department_name
LIKE 'IT%')
AND salary > (SELECT avg(salary) FROM employees);
```

Q6:

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary >
(SELECT salary FROM employees WHERE last_name = 'Bell') ORDER
BY first_name;
```

Q7:

```
SELECT * FROM employees
WHERE salary = (SELECT MIN(salary) FROM employees);
```

Q8: Write a query to find the names (first_name, last_name), the salary of the em-ployees whose salary greater than the average salary of all departments?

```
SELECT * FROM employees
WHERE salary >
ALL(SELECT avg(salary)FROM employees GROUP BY department_id);
```

Q9: Write a query to find the names (first_name, last_name) and salary of the em-ployees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = ’SH_CLERK’). Sort the results of the salary of the lowest to highest.

```
SELECT first_name,last_name, job_id, salary
FROM employees
WHERE salary >
ALL (SELECT salary FROM employees WHERE job_id = 'SH_CLERK')
ORDER BY salary;
```

Q10: Write a query to find the names (first_name, last_name) of the employees who are not supervisors?

```
SELECT b.first_name,b.last_name
FROM employees b
WHERE NOT EXISTS (SELECT 'X' FROM employees a WHERE
a.manager_id = b.employee_id);
```

Q11: Write a query to display the employee ID, first name, last names, and depart-ment names of all employees.

```
SELECT employee_id, first_name, last_name,
(SELECT department_name FROM departments d
 WHERE e.department_id = d.department_id) department
 FROM employees e ORDER BY department;
```

Q12: Write a query to display the employee ID, first name, last names, salary of all employees whose salary is above average for their departments.

```
SELECT employee_id, first_name
FROM employees AS A
WHERE salary >
(SELECT AVG(salary) FROM employees WHERE department_id =
A.department_id);
```

Q13: Write a query to fetch even numbered records from employees table.

```
SET @i = 0;
SELECT i, employee_id
FROM (SELECT @i := @i + 1 AS i, employee_id FROM employees)
a WHERE MOD(a.i, 2) = 0;
```

Q14: Write a query to find the 5th maximum salary in the employees table.

```
SELECT DISTINCT salary
FROM employees e1
WHERE 5 = (SELECT COUNT(DISTINCT salary)
FROM employees  e2
WHERE e2.salary >= e1.salary);
```

Q15: Write a query to find the 4th minimum salary in the employees table.

```
SELECT DISTINCT salary
FROM employees e1
WHERE 4 = (SELECT COUNT(DISTINCT salary)
FROM employees  e2
WHERE e2.salary <= e1.salary);
```

Q16: Write a query to select last 10 records from a table.

```
SELECT * FROM (
SELECT * FROM employees ORDER BY employee_id DESC LIMIT 10)
sub
ORDER BY employee_id ASC;
```

Q17: Write a query to list department number, name for all the departments in which there are no employees in the department.

```
SELECT * FROM departments
```

```
WHERE department_id
NOT IN (select department_id FROM employees);
```

Q18: Write a query to get 3 maximum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE 3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary >= a.salary)
ORDER BY a.salary DESC;
```

Q19: Write a query to get 3 minimum salaries.

```
SELECT DISTINCT salary
FROM employees a
WHERE  3 >= (SELECT COUNT(DISTINCT salary)
FROM employees b
WHERE b.salary <= a.salary)
ORDER BY a.salary DESC;
```

Q20: Write a query to get nth max salaries of employees.Further practice with nested queries

```
SELECT *
FROM employees emp1
WHERE (1) = (
SELECT COUNT(DISTINCT(emp2.salary))
FROM employees emp2
WHERE emp2.salary > emp1.salary);
```

LAB 11 Common Solution:

```
CREATE TABLE table_name (

   Name  VARCHAR(20),

   Family_name VarChar(20),



);

Insert into table_name(Name,Family_name)
```

```
select

'Ali' as Name,

'Khan' as Family_name;

Insert into table_name(Name,Family_name)

select

'Ahmad' as Name,

'Iqbal' as Family_name;

Insert into table_name(Name,Family_name)

select

'Zubar' as Name,

'Akram' as Family_name;

SELECT Name,Family_name ,Name+"+Family_name as fullname

FROM   table_name;

/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

```
Use dreamhome;
create table Name (FName varchar(50) NOT NULL,familyName varchar (50) NOT NULL);
insert into Name
values ('Jamal', 'Nawaz'),
('Aftab', 'Kurshid'),
('Abdul', 'Rehman');
-- CONCATINATION
SELECT CONCAT("Ali ", "Shoukat") AS full String;
-- extra string funtions
SELECT LENGTH(Fname) AS LengthOfString from Name;
select LOCATE("i", "Ali");
SELECT upper("Abdullah");
SELECT lower("Saifullah");
SELECT REPEAT(familyName, 3) from Name;
SELECT STRCMP("Afaaq", "Afaq");
SELECT SUBSTR("Jawad", 4) AS ExtractString;
SELECT LEFT("Amaan", 5) AS ExtractString;
SELECT ASCII(FName) from Name;
```

Zahir Ayub Khan:

```
CREATE TABLE table_name (
   Name  VARCHAR(20),
   Family_name VarChar(20),



);
Insert into table_name(Name,Family_name)
select
'Ali' as Name,
'Khan' as Family_name;
Insert into table_name(Name,Family_name)
select
'Ahmad' as Name,
'Iqbal' as Family_name;
Insert into table_name(Name,Family_name)
select
'Zubar' as Name,
'Akram' as Family_name;
SELECT Name,Family_name ,Name+''+Family_name as fullname
FROM   table_name;
```

Jamal Khan:

```
Use dreamhome;
create table Name (FName varchar(50) NOT NULL,familyName varchar (50) NOT NULL);
insert into Name
values ('Jamal', 'Nawaz'),
('Aftab', 'Kurshid'),
('Abdul', 'Rehman');
-- CONCATINATION
SELECT CONCAT("Ali ", "Shoukat") AS full String;
-- extra string funtions
SELECT LENGTH(Fname) AS LengthOfString from Name;
select LOCATE("i", "Ali");
SELECT upper("Abdullah");
SELECT lower("Saifullah");
SELECT REPEAT(familyName, 3) from Name;
SELECT STRCMP("Afaaq", "Afaq");
SELECT SUBSTR("Jawad", 4) AS ExtractString;
SELECT LEFT("Amaan", 5) AS ExtractString;
SELECT ASCII(FName) from Name;
```

LAB 12 Common Solution:

Q1

select CountryCode,sum(Percentage) from countrylanguage group by CountryCode;

Q2

select sum(Population) from country;

Q3

select count(*) from country;

Q4

select count(distinct Language) from countrylanguage;

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

```
Q1:
SELECT country_code, sum(percentage)
FROM country_language group by country_code;
Q2:
SELECT sum(Population) From country;
Q3:
```

```
SELECT count(*) From country;
Q4:
SELECT count(distinct language)
 from country_language;
```

Zahir Ayub Khan:

Q1

```
 select CountryCode,sum(Percentage) from countrylanguage
group by CountryCode;
```

Q2

```
 select sum(Population) from country;
```

Q3

```
 select count(*) from country;
```

Q4

```
 select count(distinct Language) from countrylanguage;
```

Jamal Khan:

```
Q1:
SELECT country_code, sum(percentage)
FROM country_language group by country_code;
Q2:
SELECT sum(Population) From country;
Q3:
SELECT count(*) From country;
Q4:
SELECT count(distinct language)
 from country_language;
```

Lab 13 Common Solution:

```
Q1
SELECT customers.customerNumber,customers.customerName,
payments.checkNumber,payments.paymentDate, payments.amount FROM
customers inner JOIN payments ON
customers.customerNumber=payments.customerNumber;
Q2
 SELECT products.productName, orderdetails.quantityOrdered ,
orderdetails.priceEach FROM orderdetails INNER JOIN products ON
orderdetails.productCode=products.productCode;
 Q3
 SELECT products.productName, productlines.productLine FROM products
Right JOIN productlines ON products.productLine=productlines.productLine;
Q4
 SELECT customers.customerName, orders.orderNumber FROM
customers inner JOIN orders ON
customers.customerNumber=orders.customerNumber and
customers.customerNumber=103;

/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/

Q1:
SELECT Customers.Customer_Number,Customers.Customer_Name,
Payments.Check_Number,Payments.Payment_Date,
Payments.Payments_amount  FROM customers inner JOIN payments ON
Customers.Customer_Number=Payments.Customer_Number;
Q2:
SELECT Products.Product_Name, Order_details.Quantity_Ordered ,
Order_details.Price_Each FROM Order_details INNER JOIN products ON
Order_details.Product_Code=Products.Product_Code;
Q3:
SELECT Products.Product_Name, Product_lines.Product_Line FROM
products Right JOIN Product_lines ON
Products.Product_Line=Product_lines.Product_Line;
Q4:
SELECT Customers.Customer_Name, Orders.Order_Number FROM
Customers inner JOIN orders ON
Customers.Customer_Number=Orders.Customer_Number and
Customers.Customer_Number=103;
```

Zahir Ayub Khan:

```
SELECT customers.customerNumber,customers.customerName,
payments.checkNumber,payments.paymentDate, payments.amount FROM
customers inner JOIN payments ON
customers.customerNumber=payments.customerNumber;
```

Q2

```
 SELECT products.productName, orderdetails.quantityOrdered ,
orderdetails.priceEach FROM orderdetails INNER JOIN products ON
orderdetails.productCode=products.productCode;
```

Q3

```
 SELECT products.productName, productlines.productLine FROM products
Right JOIN productlines ON products.productLine=productlines.productLine;
```

Q4

```
 SELECT customers.customerName, orders.orderNumber FROM
customers inner JOIN orders ON
customers.customerNumber=orders.customerNumber and
customers.customerNumber=103;
```

Jamal Khan:

Q1:

```
SELECT Customers.Customer_Number,Customers.Customer_Name,
Payments.Check_Number,Payments.Payment_Date,
Payments.Payments_amount  FROM customers inner JOIN payments ON
Customers.Customer_Number=Payments.Customer_Number;
```

Q2:

```
SELECT Products.Product_Name, Order_details.Quantity_Ordered ,
Order_details.Price_Each FROM Order_details INNER JOIN products ON
Order_details.Product_Code=Products.Product_Code;
```

Q3:

```
SELECT Products.Product_Name, Product_lines.Product_Line FROM
products Right JOIN Product_lines ON
Products.Product_Line=Product_lines.Product_Line;
```

Q4:

```
SELECT Customers.Customer_Name, Orders.Order_Number FROM
Customers inner JOIN orders ON
Customers.Customer_Number=Orders.Customer_Number and
Customers.Customer_Number=103;
```

Lab 14 Common Solution:

Q1 update employees set firstName='office number 6' where employeeNumber=1002;

Q2

update customers set customerName= 10124 where customerNumber=112;

Q3

select orderNumber, status from orders union select orderNumber,productCode from orderdetails;

Q4

select officecode from offices where officecode=5 union select officecode from employees where officecode=5;

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

Q1:

update employees set first_Name='office number 6' where employee_Number=1002;

Q2:

update customers set customer_Name= 10124 where customer_Number=112;

Q3:

select order_Number, status from orders union select order_Number,product_Code from order_details;

Q4:

select office_code from offices where office_code=5 union select office_code from employees where office_code=5;

Zahir Ayub Khan:

Q1 update employees set firstName='office number 6' where employeeNumber=1002;

Q2

update customers set customerName= 10124 where customerNumber=112;

Q3

```
 select orderNumber, status from orders union select
orderNumber,productCode from orderdetails;
```

Q4

```
 select officecode from offices where officecode=5 union select
officecode from employees where officecode=5;
```

Jamal Khan:

Q1:

```
update employees set first_Name='office number 6' where
employee_Number=1002;
```

Q2:

```
update customers set customer_Name= 10124 where
customer_Number=112;
```

Q3:

```
select order_Number, status from orders union select
order_Number,product_Code from order_details;
```

Q4:

```
select office_code from offices where office_code=5 union select
office_code from employees where office_code=5;
```

Lab 15 Common Solution:

Q1

```
update country set Name='Canada' where Code='ABW';
```

Q2

```
 delete from city where id='2831' ;
```

 Q3

```
 update orders set comments='no longer null' where orderNumber=10100;
```

Q5

```
 insert into customers
values(90,'AliKhan','ahmad','Rana',03775645891,'E_11','jinanabad','Islamab
ad','Punjab',4400,'Pak istan',1000,15000.00);
```

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

Q1:

```
update country set Name='Dubai' where Code='ABC';
```

Q2:

```
delete from city where id='2211' ;
```

Q3:

```
update orders set comments='no longer null' where order_Number=1122;
```

Q4:

```
 insert into customers
values(11,'AftabAhmad','Khan','Ali',03558945685,'F_10','Islamabad','Abbott
abad','Sindh',4455,'Pakistan',1000,10000.00);
```

Zahir Ayub Khan:

Q1

```
update country set Name='Canada' where Code='ABW';
```

Q2

```
 delete from city where id='2831' ;
```

 Q3

```
 update orders set comments='no longer null' where orderNumber=10100;
```

Q5

```
 insert into customers
values(90,'AliKhan','ahmad','Rana',03775645891,'E_11','jinanabad','Islamab
ad','Punjab',4400,'Pak istan',1000,15000.00);
```

Jamal Khan:

Q1:

```
update country set Name='Dubai' where Code='ABC';
```

Q2:

```
delete from city where id='2211' ;
```

Q3:

```
update orders set comments='no longer null' where order_Number=1122;
```

Q4:

```
 insert into customers
values(11,'AftabAhmad','Khan','Ali',03558945685,'F_10','Islamabad','Abbott
abad','Sindh',4455,'Pakistan',1000,10000.00);
```

Lab 16 Common Solution:

```
create database suppliers
 Create table suppliers (supplier_id int(10) not null,
 supplier_name varchar(50),
 address varchar(50),
 primary key (supplier_id));
 Create table Item
 (itemname varchar(50) ,
 supplier_id int(10),
 itemprice int(10),
 FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id) );
```

```
/* Above work is Zahir Ayub Khan. Work Below Is Jamal khan*/
```

```
Create database Supplier
Create table supplier (supplier_id int(10) not null,
Supplier_name varchar(50),
Address varchar(50),
Primary key (supplier_id));
Create table Item
(itemname varchar(50) ,
supplier_id int(10),
itemprice int(10),
FOREIGN KEY (Supplier_id) REFERENCES Supplier(Supplier_id) );
```

Zahir Ayub Khan:

```
create database suppliers
 Create table suppliers (supplier_id int(10) not null,
 supplier_name varchar(50),
 address varchar(50),
 primary key (supplier_id));
 Create table Item
 (itemname varchar(50) ,
 supplier_id int(10),
 itemprice int(10),
 FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id) );
```

Jamal Khan:

```
Create database Supplier
Create table supplier (supplier_id int(10) not null,
Supplier_name varchar(50),
Address varchar(50),
Primary key (supplier_id));
Create table Item
(itemname varchar(50) ,
supplier_id int(10),
itemprice int(10),
FOREIGN KEY (Supplier_id) REFERENCES Supplier(Supplier_id) );
```

Lab 18 Common Solution:

Q1

```
create view crust as
select customerNumber,customerName from customers;
update crust
set customerName='zahir khan'
where customerNumber=103;
```

Q2

```
create view duckgo as
select customerNumber,amount from payments;
select avg(amount) from duckgo;
select avg(amount) from duckgo where customerNumber <151 ;
```

Zahir Ayub Khan:

Q1

```
create view crust as
select customerNumber,customerName from customers;
update crust
set customerName='zahir khan'
where customerNumber=103;
```

Q2

```
create view duckgo as
select customerNumber,amount from payments;
select avg(amount) from duckgo;
select avg(amount) from duckgo where customerNumber <151 ;
```

Jamal Khan: