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");
```

```
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
```

// below work is by Nauman ali

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

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

INSERT INTO Std(Identity, Name)
VALUES ("fa17-bse-000", "ali");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-004", "nauman");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-006", "shan");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-008", "sardar");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-010", "qasim");
```

```
INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Urdu","2.3","fa17-bse-000");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("English","3.3","fa17-bse-004");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Cal","2.7","fa17-bse-006");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Ca","3.7","fa17-bse-008")


INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Ps","3.0","fa17-bse-010");
```

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")
```

```sql
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

  //NAUMAN ALI


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

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

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-000", "ali");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-004", "nauman")
```

```
INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-006", "sardar");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-008", "shan");

INSERT INTO Std (Identity, Name)
VALUES ("fa17-bse-010", "qasim");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Urdu","2.3","fa17-bse-000");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("English","3.3","fa17-bse-004");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Cal","2.7","fa17-bse-006");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Ca","3.7","fa17-bse-008");

INSERT INTO Transcript(Subj,GPA,Identity)
VALUES ("Ps","3.0","fa17-bse-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');
```

```
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
```

```sql
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");
```

```sql
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
```

//NAUMAN

```
create database StudentTranscriptDB
use StudentTranscriptDB


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

  insert into Transcript (Course,GPA,StdId) values
  ('Database Systems','2.3','fa17-bse-000')
insert into Transcript (Course,GPA,StdId) values
  ('Database Systems','3.3','fa17-bse-004')
 insert into Transcript (Course,GPA,StdId) values
  ('Database System','3.7','fa17-bse-006')
insert into Transcript (Course,GPA,StdId) values
  ('Database Systems','3.3','fa17-bse-008')
insert into Transcript (Course,GPA,StdId) values
  ('Database System','3.0','fa17-bse-010')
```

DELETE FROM Transcript WHERE StudentId='fa17-bse-000';
select * from Transcript

UPDATE Transcript set GPA='3.3'WHERE StudentId='fa17-bse-004';
select * from Transcript

DELETE FROM Transcript WHERE StudentId='fa17-bse-006';
select * from Transcript

UPDATE Transcript set GPA='3.3'WHERE StudentId='fa17-bse-008';
select * from Transcript

DELETE FROM Transcript WHERE StudentId='fa17-bse-010';
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 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");

/* 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");
```

// NAUMAN ALI

```
select * from branch;
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR000', 'H#0 A-1/1', 'ABBOTTABAD', '00000');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR001', 'H#1 B-1/2', 'LAHORE', '01000');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR002', 'H#2 C-3/3', 'MULTAN', '02000');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR003', 'H#3 D-3/4', 'QUETTA', '03000');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR004', 'H#4 E-4/5', 'SAWAT', '04000');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR005', 'H#5 F-1/6', 'SAWABI', '05000');

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 ('BR000','H#6 G-7/7', 'ABBOTTABAD', '20000');
```

```
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR001','H#7 H-8/8', 'LAHORE', '20001');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR002','H#8 I-9/9', 'MULTAN', '20002');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR003','H#9 J-10/10', 'QUETTA', '20003');
INSERT into Branch (branchNo, street, city, postcode) VALUES ('BR004', 'H#10 K-11/11', 'SAWAT', '20004');
```

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'

//NAUMAN

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'

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')

//NAUMAN
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='B002')
Qno9:-.....
Ans:select *from Staff where salary> all(select salary from Staff where branchNo='B002')

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');

// NAUMAN

Q1:.....

Ans;SELECT * FROM employees;

Q2:.....

Ans;SELECT FIRST_NAME, LAST_NAME, SALARY
FROM employees
WHERE SALARY >
(SELECT salary FROM employees WHERE last_name = 'xxxx');
Q3:.....
Ans;SELECT first_name, last_name
FROM employees
WHERE department_id
IN (SELECT department_id FROM departments WHERE department_name='cs');

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);
```

//Nauman

Q1:.....

Ans: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:.....

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

Q3:.....

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

Q4:.....

Ans: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:.....
```
Ans:SELECT first_name, last_name, salary
FROM employees
WHERE salary >
(SELECT salary FROM employees WHERE last_name = 'Bell') ORDER BY first_name;
```
Q7:.....
```
Ans: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.
```
Ans: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?
```
Ans: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.
```
Ans: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.
```
Ans: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.
```
Ans: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.

```
Ans: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.

Ans: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.

Ans: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.

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

Q18: Write a query to get 3 maximum salaries.

Ans: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.

Ans: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

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

LAB 11 Common Solution:

Zahir Ayub Khan:

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;
```

```sql
-- 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;
-- Maths function
SELECT COT(6);
SELECT COS(2);
SELECT LOG(2);

SELECT SQUARE(64);
SELECT COUNT(Fname) AS NumberOfNames FROM Name;
SELECT AVG(maxRent) AS AveragePrice FROM client;
SELECT MAX(maxRent) AS LargestPrice FROM client;
SELECT MIN(maxRent) AS SmallestPrice FROM client;
SELECT FLOOR(25.75) AS FloorValue;
 SELECT CEILING(25.75) AS CeilValue;
```

LAB 12 Common Solution:

Zahir Ayub Khan:

Jamal Khan:

Q1:

```sql
SELECT country_name, COUNT(Country_code)
SELECT Sum(Urdu+English+pashto)AS total FROM country_language
```

Q2:

```sql
SELECT sum([DISTINCT] expression) From 'Country'
```

Q3:

```sql
SELECT count(*) as total record 'Country'
```

Q4:

```sql
SELECT countrylanguage
FROM (
   SELECT countrylanguage, COUNT(*) AS cnt
   FROM mytable
   WHERE language IN ('urdu','german','french','english')
   GROUP BY countrylanguage
)
```

//Nauman

Q1:.....

Ans:SELECT country_name, COUNT(Country_code)

SELECT Sum(Urdu+English+pashto)AS total FROM country_language

Q2:.....

Ans:SELECT sum([DISTINCT] expression) From 'Country'

Q3:.....

Ans:SELECT count(*) as total record 'Country'

Q4:.....

Ans:SELECT countrylanguage

```
FROM (
    SELECT countrylanguage, COUNT(*) AS cnt
    FROM mytable
    WHERE language IN ('urdu','german','french','english')
    GROUP BY countrylanguage
)
```