# Lab 5:

# Question No 1:

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 ('Fa16-bse-001', 'haider Ali'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-002', 'eman aziz'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-003', 'taha khokar'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-004', 'naveem'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-005', 'Tabish shah'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-006', 'zain tahir'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-007', 'alisha khan'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-008', 'sardar hasseb'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-009', 'ahmad Khan'); INSERT INTO Student (ID, Name)

VALUES ('Fa16-bse-010', 'ijaz khan'); INSERT INTO Transcript(Subject,GPA,ID) VALUES ('CAL 1','2.3','Fa16-bse-001');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('stats','3.3','Fa16-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('Database system','2.7','Fa16-bse-003'); INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('urdu','3.7','Fa16-bse-004');

INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('Islamic studies','3.0','Fa16-bse-005'); INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('PAK STUDIES','2.0','Fa16-bse-006');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('math','2.7','Fa16-bse-007');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('OOP','2.3','Fa18-bse-008');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('CAL 2','2.3','Fa18-bse-009');

INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('REPORT WRITING','2.7','Fa18-bse-010');

# Update and delete;

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 ('Fa18-bse-001', 'haider Ali'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-002', 'eman aziz'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-003', 'taha khokar'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-004', 'naveem'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-005', 'Tabish shah'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-006', 'zain tahir'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-007', 'alisha khan'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-008', 'sardar haseeb'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-009', 'ahmad khan'); INSERT INTO Student (ID, Name)

VALUES ('Fa18-bse-010', 'ijaz khan'); INSERT INTO Transcript(Subject,GPA,ID) VALUES ('CAL 1','2.3','Fa18-bse-001');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('stats','3.3','Fa18-bse-002');

INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('Database system','2.3','Fa18-bse-003'); INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('urdu','3.7','Fa18-bse-004');

INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('Islamic studies','3.0','Fa18-bse-005'); INSERT INTO Transcript(Subject,GPA,ID)

VALUES ('PAK STUDIES','2.0','Fa18-bse-006');

INSERT INTO Transcript(Subject,GPA,ID) VALUES ('math','2.7','Fa18-bse-007');

INSERT INTO Transcript(Subject,GPA,ID) UPDATE Transcript

Set GPA='2.7'

WHERE ID='Fa18-bse-001';

UPDATE Transcript Set GPA='1.7'

WHERE ID='Fa18-bse-006';

UPDATE Transcript Set GPA='2.3'

WHERE ID='Fa18-bse-010';

DELETE FROM Transcript WHERE ID='Fa18-bse-009'; DELETE FROM Transcript WHERE ID='Fa18-bse-008';

# Lab 6:

# QUESTION NO 1:

create database HOME; USE HOME;

create table Branch (

branchNo varchar(20) NOT NULL PriMark Key, street varchar(100) NOT NULL,

city varchar(50) NOT NULL, postcode varchar(20) NOT NULL

);

create table Staff (

staffNo varchar(20)0 NOT NULL PRIMARK KEY, fName varchar(50) NOT NULL,

lName varchar(50) NOT NULL, position varchar(50) NOT NULL,

sex varchar(1) NOT NULL, DOB DateTime NOT NULL, salary DECIMAL NOT NULL,

branchNo varchar(20) NOT NULL References Branch(branchNo)

);

create table Client (

clientNo varchar(20) NOT NULL PRIMARK KEY, fName varchar(50) NOT NULL,

lName varchar(50) NOT NULL, telNo varchar(20) NOT NULL, prefType varchar(50) NOT NULL, maxRent DECIMAL NOT NULL

);

create table PrivateOwner (

ownerNo varchar(20) NOT NULL PRIMARK KEY, fName varchar(50) NOT NULL,

lName varchar(50) NOT NULL, address varchar(50) NOT NULL, telNo varchar(20) NOT NULL

);

create table PropertyForRent (

propertyNo varchar(20) NOT NULL PRIMARK KEY, street varchar(100) NOT NULL,

city varchar(50) NOT NULL, postcode varchar(20) NOT NULL, type varchar(10) NOT NULL, rooms int NOT NULL,

rent DECIMAL NOT NULL,

ownerNo varchar(20) References PrivateOwner(ownerNo), staffNo varchar(20) NOT NULL References Staff(staffNo), branchNo varchar(20) NOT NULL References Branch(branchNo)

);

create table Viewing (

clientNo varchar(20) NOT NULL References Client(clientNo),

propertyNo varchar(20) NOT NULL References PropertyForRent(propertyNo), viewDate DateTime NOT NULL,

comment varchar(200) NOT NULL

);

create table Registration (

clientNo varchar(20) NOT NULL References Client(clientNo), branchNo varchar(20) NOT NULL References Branch(branchNo), staffNo varchar(20) NOT NULL References Staff(staffNo), dateJoined DateTime NOT NULL

);

INSERT into Branch (

branchNo, street, city, postcode

) VALUES

(N'B100',N'H#7 I-10/2', N’LHR’, N'52000'),

(N'B021',N'H#78 Supply', N'ABT', N'53000'),

(N’B012’,N'H#79 I-10/2', N’LHR’, N'52000'),

(N'B011',N'H#78 Mandian', N'ABT', N'53000');

insert into Staff (

staffNo, fName, lName, position, sex,

DOB,

salary, branchNo

) VALUES (

N'SA9', N'Mark', N'Luther', N'Assistant', N'F',CAST(0x0001341000000000 AS DateTime), CAST(9000 AS Decimal(18, 0)),N'B021'

),

(N'SG14', N'David', N'Malan', N'Supervisor', N'M', CAST(0x0001231200000000 AS DateTime), CAST(18000 AS Decimal(18,0)), N’B012’),

(N'SG37', N'Alex', N'Beech', N'Assistant', N'F',

CAST(0x000126D400000000 AS DateTime), CAST(12000 AS Decimal(18,0)), N’B012’),

(N'SG5',N'Susan', N'Bhatti', N'Manager', N'F',

CAST(0x0000C85800000000 AS DateTime), CAST(24000 AS Decimal(18,0)), N’B012’),

(N'SL21', N’Robert’, NBlack’, N'Manager', N'M', CAST(0x0000CFF200000000 AS DateTime), CAST(30000 AS Decimal(18,0)),

N'B011'),

(N'SL41', N’Charlie’, N’Japlin’, N'Assistant', N'F', CAST(0x00012D6000000000 AS DateTime), CAST(9000 AS Decimal(18, 0)), N'B021');

insert into Client (

clientNo, fName, lName, telNo, prefType, maxRent

)

values (

'B1100',’zeeshan’,’naveed’,'030148601','yes',1000.0

), (

'B1021','Ali','Khan','030124621','yes',2000.0

), (

'B1064','Kaleem','Shahid','030456601','no',1500.0

), (

'B1011',’Ali’,’Khan’,'0306446641','yes',8800.0

), (

'B1012',’Ghafoor’,’Riyaz’,'0354654401','noo',800.0

), (

'B1013','Usama','javad','030144541','yes',4000.0

);

insert into PrivateOwner (

ownerNo, fName, lName, [address], telNo

)

values (

'B1',’Group’,’Leader’,'F18-4A',’03218323263’

), (

'B2',’Usman’,’kaleem’,'F17-4A',’03234823261’

), (

'B3','Saman','Khan','F14-7A','03239354264'

), (

'B4','Muhammad','Kaleem','F88-4A',’03234523265’

), (

'B5',’Ali’,’Khan’,'F11-3A',’031234567886’

), (

'B6',’Ali’,’Riyaz’,'F19-5A',’03105023263’

);

insert into PropertyForRent (

propertyNo,street,city,postcode,[type],rooms,rent,ownerNo,staffNo,branchNo

)

values (

'BF2','H2-h2','ABT','22017','large',8,'30000','B2','SG14','B021'

), (

'BF3','H3-h3',’LHR’,'62017','medium',6,'20000','B3','SG37',’B012’

),

(

'BF4','H4-h4',’LHR’,'62017','small',4,'10000','B4','SG5','B011'

);

insert into Viewing (

clientNo, propertyNo, viewDate, comment

)

values (

'B1021','BF2','2020-2-10','No, i am not interested’’

), (

'B1064','BF3','2020-3-10','No, i am not interested’’

), (

'B1011','BF4','2020-4-10','No, i am not interested’’

);

insert into Viewing (

clientNo, propertyNo, viewDate, comment

)

values (

'B1021','BF2','2020-2-10','No, i am not interested’’

), (

'B1064','BF3','2020-3-10','No, i am not interested’’

), (

'B1011','BF4','2020-4-10','No, i am not interested’’

);

# Question no 2:

select \* from Branch update Branch set city='ABT' where city=‘LHR’;

# QUESTION N0 3:

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 ('B1016','X#01 Y-11/1', 'ABBOTTABAD', '22500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0210','X#91 Y-22/2', 'MANSERA', '23400');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0211','X#89 Y-33/3', 'PESHAWER', '24500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0212','X#76 Y-44/4', 'MARDAN', '26500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0213','X#74 Y-55/5', 'SUKKAR', '27500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0214','X#90 Y-66/6', 'JHANG', '28500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0215','X#99 Y-77/7', 'LARKANA', '29500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0216','X#12 Y-88/8', 'ISLAMABAD', '31500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0217','X#99 Y-99/9', 'RAWALPINDI', '32500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0218','X#10 Y-12/1', 'LAHORE', '33500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0219','X#66 Y-13/2', 'KARACHI', '34500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0640','X#18 Y-14/3', 'MULTAN', '35500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0641','X#19 Y-15/4', 'HYDERABAD', '36500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0642','X#20 Y-16/5', 'DERA GHAZI KHAN', '37500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0643','X#94 Y-12/1', 'KARACHI', '8000');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0644','X#66 Y-13/2', 'KARACHI', '9000');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0645','X#90 Y-14/3', 'SARGODHA', '11000');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0646','X#12 Y-15/4', 'MARDAN', '15000');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0647','X#19 Y-16/5', 'QUETTA', '34500');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0648','X#35 Y-12/1', 'LAHORE', '1000');

INSERT into Branch (branchNo, street, city, postcode) VALUES ('B0649','X#18 Y-13/2', 'MULTAN', '19000');

# Lab 7:

# QUESTION N0 1:

Select distinct(postcode) from Branch;

# QUESTION N0 2:

Select distinct(fName) from Staff;

# QUESTION N0 3:

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;

# QUESTION N0 4:

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

# QUESTION N0 5:

Select salary from Staff where salary >10000;

# QUESTION N0 6:

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

# LAB 8:

# QUESTION No 1:

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

# QUESTION N0 2:

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

# QUESTION N0 3:

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

# QUESTION N0:

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

# QUESTION N0 5:

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

# QUESTION N0 6:

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

# QUESTION N0 7:

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

# QUESTION N0 8:

select \*from Staff where salary> any(select salary from Staff where branchNo='B021')

# QUESTION N0 9:

select \*from Staff where salary> all(select salary from Staff where branchNo='B021')

# QUESTION N0 10:

use DreamHome;

SELECT staffNo, fName, lName, salary FROM Staff ORDER BY salary DESC; SELECT propertyNo, type, rooms, rent FROM PropertyForRent ORDER BY type;

SELECT propertyNo, type, rooms, rent FROM PropertyForRent ORDER BY type, rent DESC; SELECT COUNT(DISTINCT propertyNo) AS myCount FROM Viewing WHERE viewDate BETWEEN '1-May- 04' AND '31-May-04';

SELECT COUNT(staffNo) AS myCount, SUM(salary) AS mySum FROM Staff WHERE position = 'Manager';

SELECT MIN(salary) AS myMin, MAX(salary) AS myMax, AVG(salary) AS myAvg FROM Staff;

--SELECT staffNo, COUNT(salary) FROM Staff;

--shwoing error that no aggregate or group clause

SELECT staffNo, fName, lName, position, salary FROM Staff WHERE salary > SOME (SELECT salary FROM Staff WHERE branchNo = 'B064');

SELECT staffNo, fName, lName, position, salary FROM Staff WHERE salary > ALL (SELECT salary FROM Staff WHERE branchNo = 'B064');

-- For DreamHome case study write at least 3 examples of each category for sorting, grouping and aggregate operations.

--sorting

SELECT branchNo FROM Branch ORDER BY postcode ASC;

SELECT fName , lName , maxRent FROM Client ORDER BY maxRent DESC; SELECT fName, salary FROM Staff ORDER BY salary DESC;

--grouping

SELECT clientNo, fName, maxRent FROM Client WHERE maxRent < 180000 AND maxRent > 10000; SELECT fName, lName FROM Staff WHERE (SELECT AVG(salary) FROM Staff) < salary;

SELECT fName, salary FROM Staff WHERE sex = 'M' AND position = 'Clerk';

--aggregate clauses

SELECT COUNT(propertyNo) AS TotalProperty FROM Viewing; SELECT avg(salary) AS totalsalary FROM Staff ;

SELECT max(salary) AS totalsalary FROM Staff ;

# LAB 9:

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');

# QUESTION N0 1:

SELECT \* FROM employees;

# QUESTION N0 2:

SELECT FIRST\_NAME, LAST\_NAME, SALARY

FROM employees WHERE SALARY >

(SELECT salary FROM employees WHERE last\_name = 'Bull');

# QUESTION N0 3:

SELECT first\_name, last\_name FROM employees

WHERE department\_id

IN (SELECT department\_id FROM departments WHERE department\_name='IT');

# LAB 10:

# QUESTION N0 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')));

# QUESTION N0 2:

SELECT first\_name, last\_name FROM employees

WHERE (employee\_id IN (SELECT manager\_id FROM employees));

# QUESTION N0 3:

SELECT first\_name, last\_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

# QUESTION N0 4:

SELECT first\_name, last\_name, salary FROM employees

WHERE employees.salary = (SELECT min\_salary FROM jobs

WHERE employees.job\_id = jobs.job\_id);

# QUESTION N0 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);

# QUESTION N0 6:

SELECT first\_name, last\_name, salary FROM employees

WHERE salary >

(SELECT salary FROM employees WHERE last\_name = 'Bell') ORDER BY first\_name;

# QUESTION N0 7:

SELECT \* FROM employees

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

# QUESTION N0 8:

SELECT \* FROM employees WHERE salary >

ALL(SELECT avg(salary)FROM employees GROUP BY department\_id);

# QUESTION N0 9:

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;

# QUESTION N0 10:

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);

# QUESTION N0 11:

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;

# QUESTION N0 12:

SELECT employee\_id, first\_name FROM employees AS A

WHERE salary >

(SELECT AVG(salary) FROM employees WHERE department\_id = A.department\_id);

# QUESTION N0 13:

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;

# QUESTION N0 14:

SELECT DISTINCT salary FROM employees e1

WHERE 5 = (SELECT COUNT(DISTINCT salary)

FROM employees e2

WHERE e2.salary >= e1.salary);

# QUESTION N0 15:

SELECT DISTINCT salary FROM employees e1

WHERE 4 = (SELECT COUNT(DISTINCT salary)

FROM employees e2

WHERE e2.salary <= e1.salary);

# QUESTION N0 16:

SELECT \* FROM (

SELECT \* FROM employees ORDER BY employee\_id DESC LIMIT 10) sub

ORDER BY employee\_id ASC;

# QUESTION N0 17:

SELECT \* FROM departments WHERE department\_id

NOT IN (select department\_id FROM employees);

# QUESTION N0 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;

# QUESTION N0 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;

# QUESTION N0 20:

SELECT \*

FROM employees emp1 WHERE (1) = (

SELECT COUNT(DISTINCT(emp2.salary))

FROM employees emp2

WHERE emp2.salary > emp1.salary);

# LAB 11:

# QUESTION N0 1:

create table stringOperations(FName varchar(50) NOT NULL,familyName varchar (50) NOT NULL);

insert into stringOperations(FName, familyName) values ('ali ', 'irtaza'),('sanaullah', 'usman'),('maqsood', 'muzamil');

--CONCATINATION

SELECT CONCAT(Fname, familyName) AS NCString FROM stringOperations;

--extra string funtions

SELECT upper(familyName) FROM stringOperations; SELECT lower(FName) FROM stringOperations; SELECT REPLACE('ali ', 'A', '3');

SELECT SPACE(10);

SELECT RIGHT (familyName, 5),familyName FROM stringOperations; SELECT LEFT(familyName, 5),familyName FROM stringOperations; SELECT ASCII(FName) FROM stringOperations;

--Maths functionSELECT COT(6); ALTER TABLE stringOperations ADD Amount float;

UPDATE stringOperations set Amount = (30.5); SELECT \* FROM stringOperations;

SELECT COS(Amount) FROM stringOperations; SELECT LOG(Amount) FROM stringOperations; SELECT SQUARE(Amount) FROM stringOperations;

SELECT COUNT(Fname) AS NumberOfNames FROM Staff s ; SELECT AVG(maxRent) AS 'avg' FROM client;

SELECT MAX(maxRent) AS 'Largest' FROM client; SELECT MIN(maxRent) AS 'Smallest' FROM client; SELECT FLOOR(Amount) FROM stringOperations; SELECT CEILING(Amount) FROM stringOperations;

# QUESTION N0 2:

Use dreamhome;

create table Name (FName varchar(50) NOT NULL,familyName varchar (50) NOT NULL); insert into Name

values ('ali', 'irtaza'),('muzamil', 'khan'),('usman ', 'maqsood');

-- CONCATINATION

SELECT CONCAT("ghafoor ", "qadir") AS full String;

-- extra string funtions

SELECT LENGTH(Fname) AS LengthOfString from Name; select LOCATE("i", "muzamil");

SELECT upper("irtaza"); SELECT lower("hadi");

SELECT REPEAT(familyName, 3) from Name; SELECT STRCMP("muhammad", "hadi");

SELECT SUBSTR("muzamil", 4) AS ExtractString; SELECT LEFT("tanveer", 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:

# QUESTION N0 1:

SELECT country\_name, COUNT(Country\_code)

SELECT Sum(Urdu+English+pashto)AS total FROM country\_language

# QUESTION N0 2:

SELECT sum([DISTINCT] expression) From ‘Country’

# QUESTION N0 3:

SELECT count(\*) as total record ‘Country’

# QUESTION N0 4:

SELECT countrylanguage FROM (

SELECT countrylanguage, COUNT(\*) AS cnt FROM mytable

WHERE language IN ('urdu','hindko','punjabi','english') GROUP BY countrylanguage

)

# LAB 13:

SELECT

c.fName, c.telNo, v.propertyNo

FROM Client c INNER JOIN Viewing v ON c.clientNo = v.clientNo

-- Display order details for products. Use inner join.

SELECT Order\_t.OrderID, OrderedQuantity, ProductDescription, ProductStandardPrice FROM FROM Order\_t inner join Product\_t on Order\_t.OrderID = Product\_t.OrderID

-- Using right outer join for productline display products. SELECT Product\_t.ProductName

FROM Product\_l

RIGHT JOIN Product\_t

ON Product\_t.id = Product\_l.id

-- Select customers name and order he made for id =103. Use AND with inner join. SELECT Order\_t.OrderID, Customer\_t.name FROM Customer

FROM Order\_t inner join Product\_t on Customer\_t.customerid = Order\_t.customerid AND Order\_t