BIIT Serie

Lab Manual

Database Management System

A simplified practical work book of Database management System course for Computer Science, Information Technology and Software Engineering students

Student Name:

_____

Registration No:

_____

Section / Semester:

_____

Submission Date:

_____

Student Signature:

_____

Dr. Munir Ahmed

Lab Manual Database Management System: First Edition
(Publisher:???)
Author(s): Prof. Dr. Munir Ahmed
ISBN: ????
Publication Date: ????
A publication of BIIT Series

Learning Objectives: By the end of the lab work, students should have following skills;

Sr.	Learning Objective	LAB No.
1	SQL introduction and basic SELECT Clause	
2	Arithmetic expressions and column alias	
3	Use of concatenation operator and distinct	
4	Order by and Where clause	
5	Logical operators (AND, OR, NOT)	
6	Comparison operators	
7	Use of between, in operator and top clause	
8	Selection with like operator	
9	Use of aggregate functions	
10	Group by and having clause	
11	SQL Create, Drop,Alter statement Add and drop constraints Primary key, Foreign key, Unique, Check	
12	SQL insert, update and delete statements	
13	Create update and drop a view	
14	SQL Joins	
15	SQL Union operator	
16	SQL Cross Join	

Table of Contents

Lab # 01

Introduction to SQL

SQL is a standard language for accessing/retrieving and manipulating databases.

What is SQL?

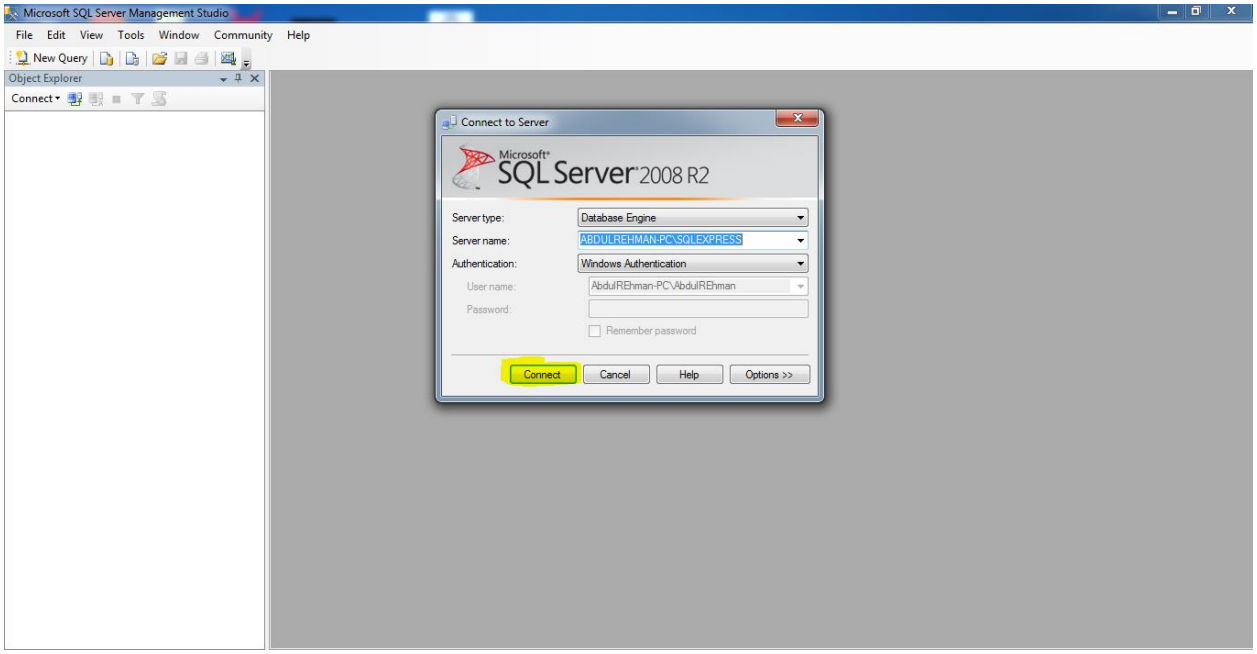
- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL is an ANSI (American National Standards Institute) standard

What Can SQL do?

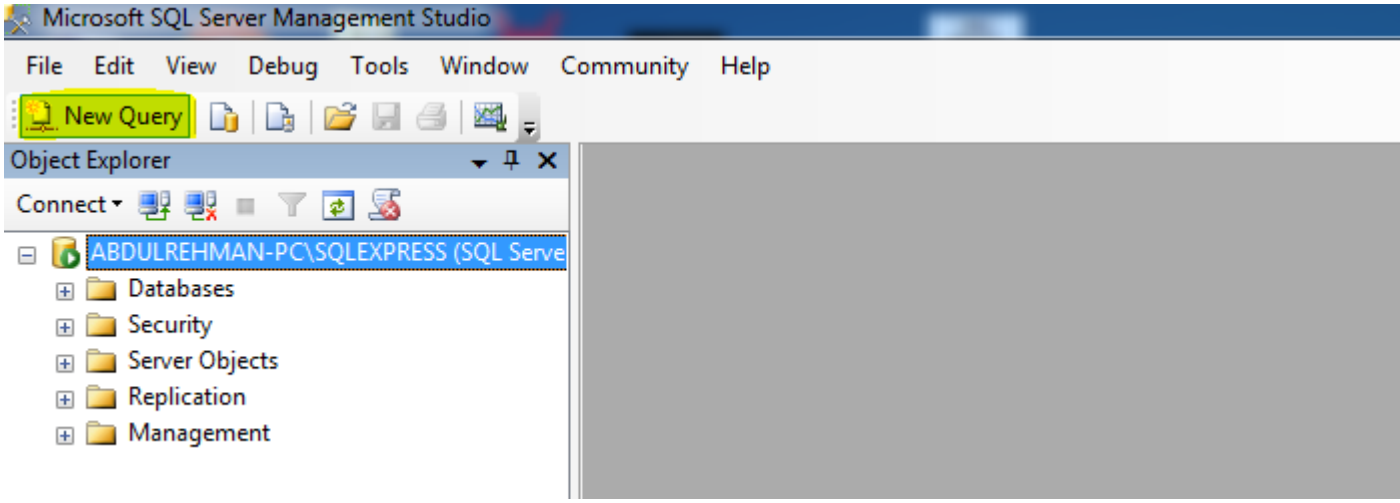
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

How can we use SQL Server (on PC)?

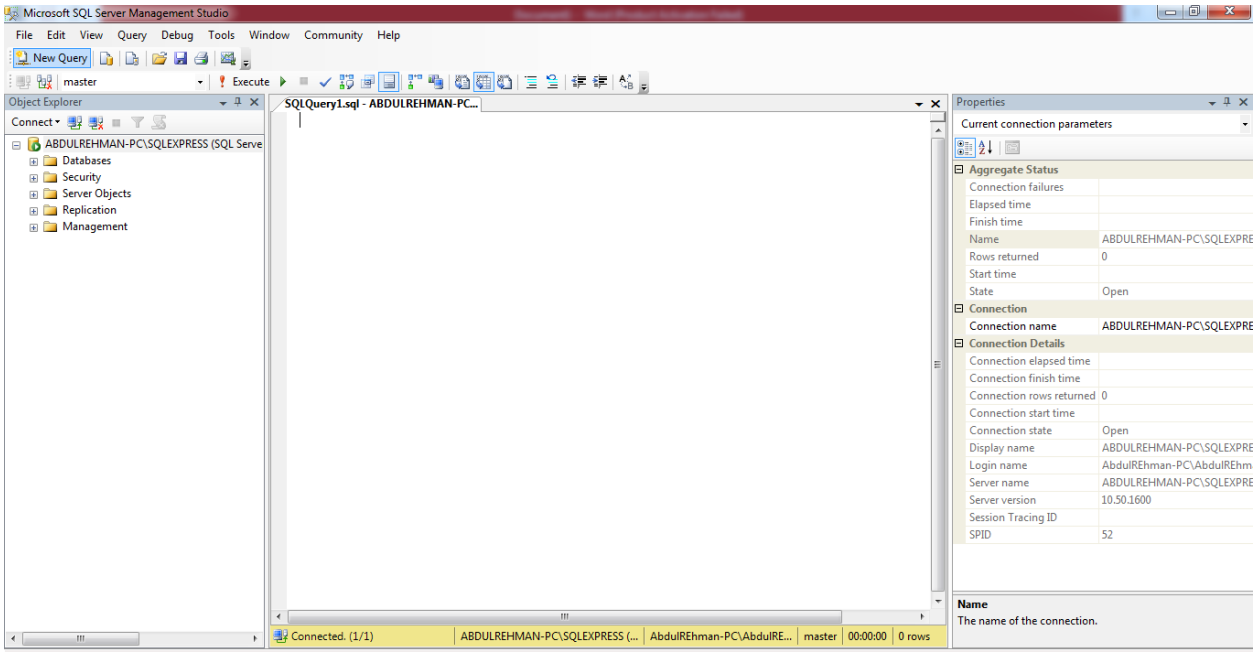
First go to start button file of your system and write SQL Server Management Studio in search bar or explore all program. Click on SQL Server Management Studio then its open a new window like this.



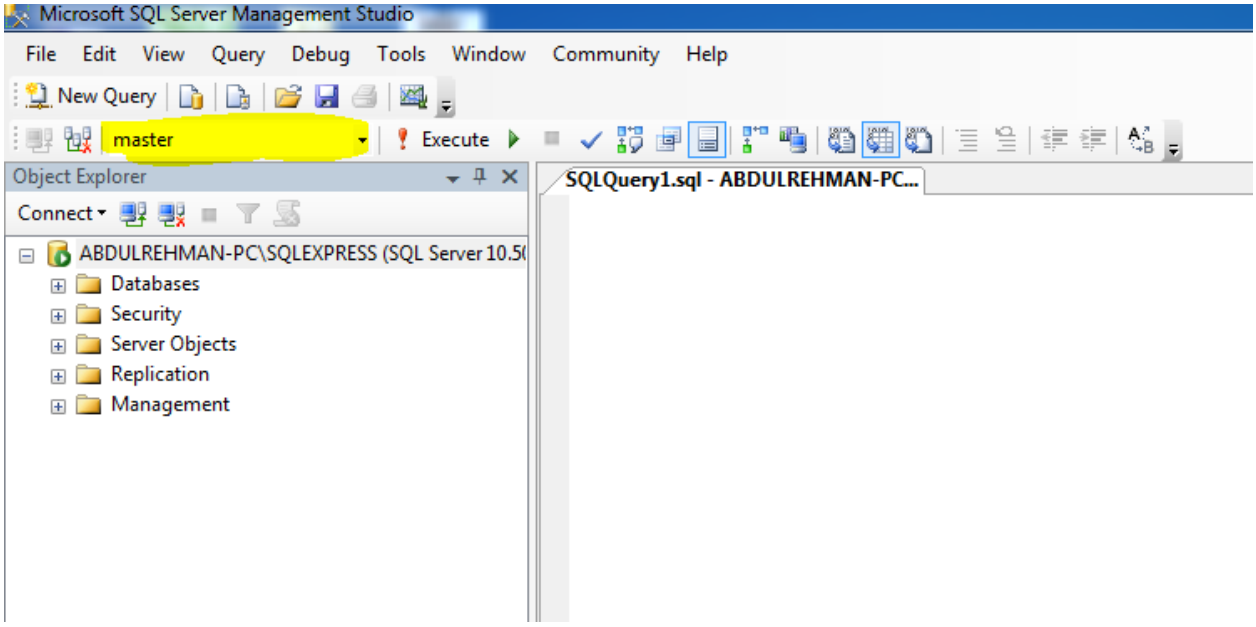
Clicking on connect button its open new Screen like this.



Now click on New Query Button its open new Screen like this to write queries.

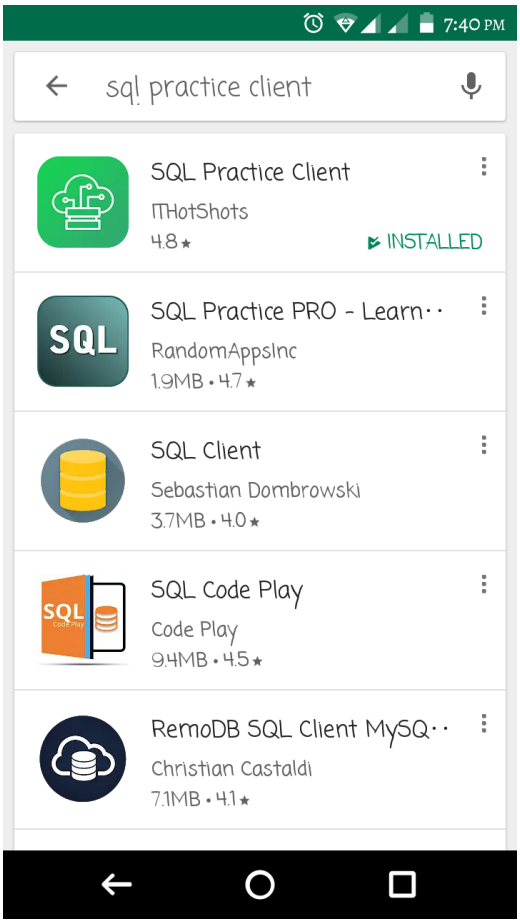


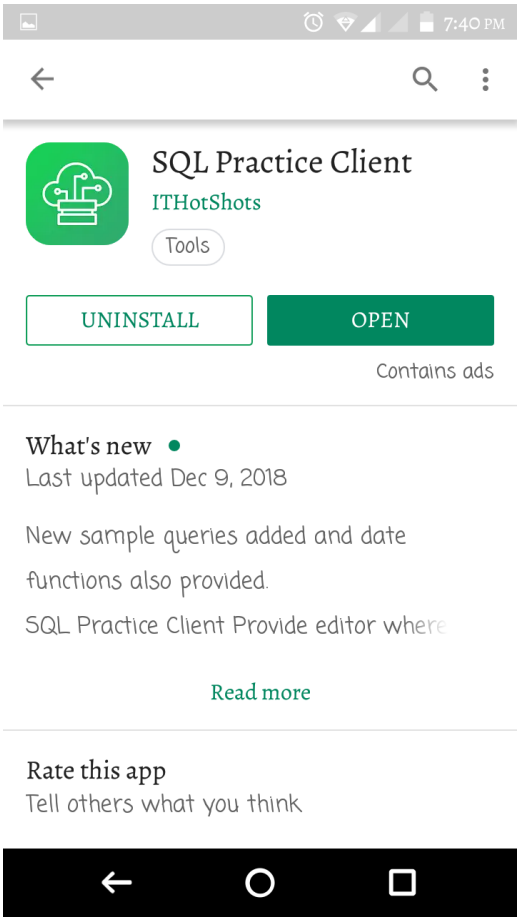
Now you can select your database or create new database. For selecting database from available database dropdown list that is underline below screen shots.



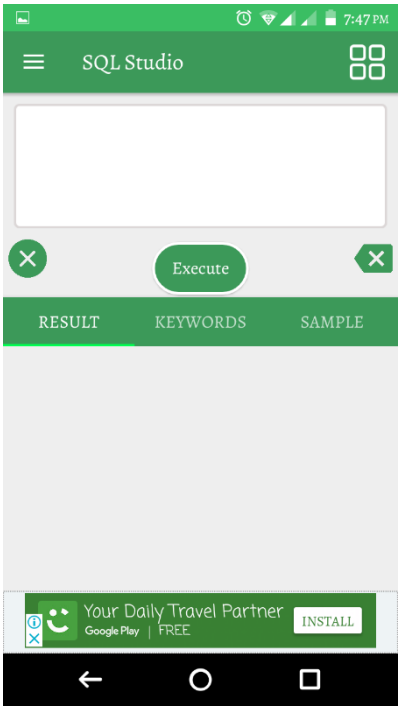
For Mobile (Android only)

To use SQL on mobile, go to Play Store and download SQL practice client and open the app.



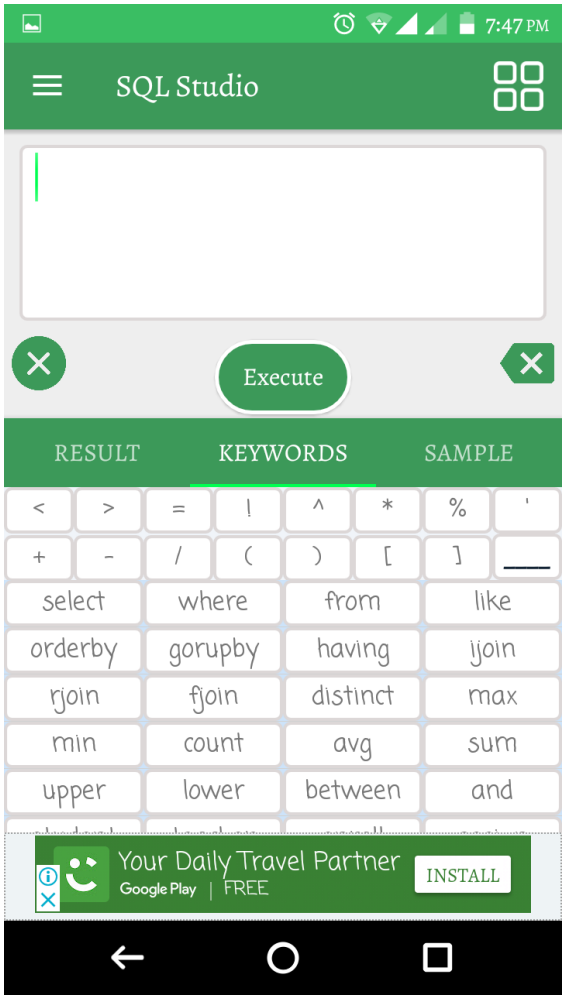


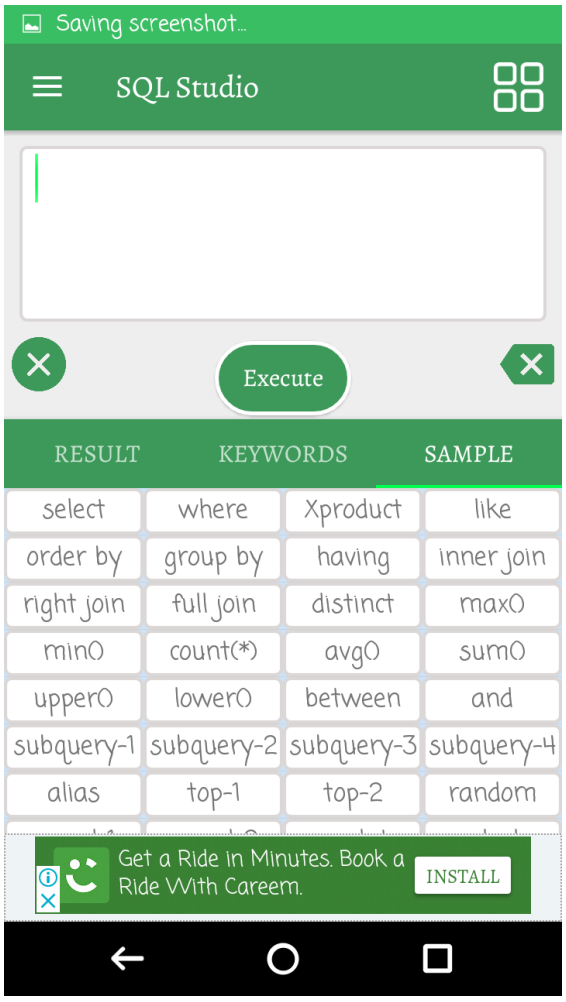
After opening the app you will get an interface like this



Benefits of using SQL Practice Client

You can use the keywords that are already added in the app. This will reduce the time required to type a query.

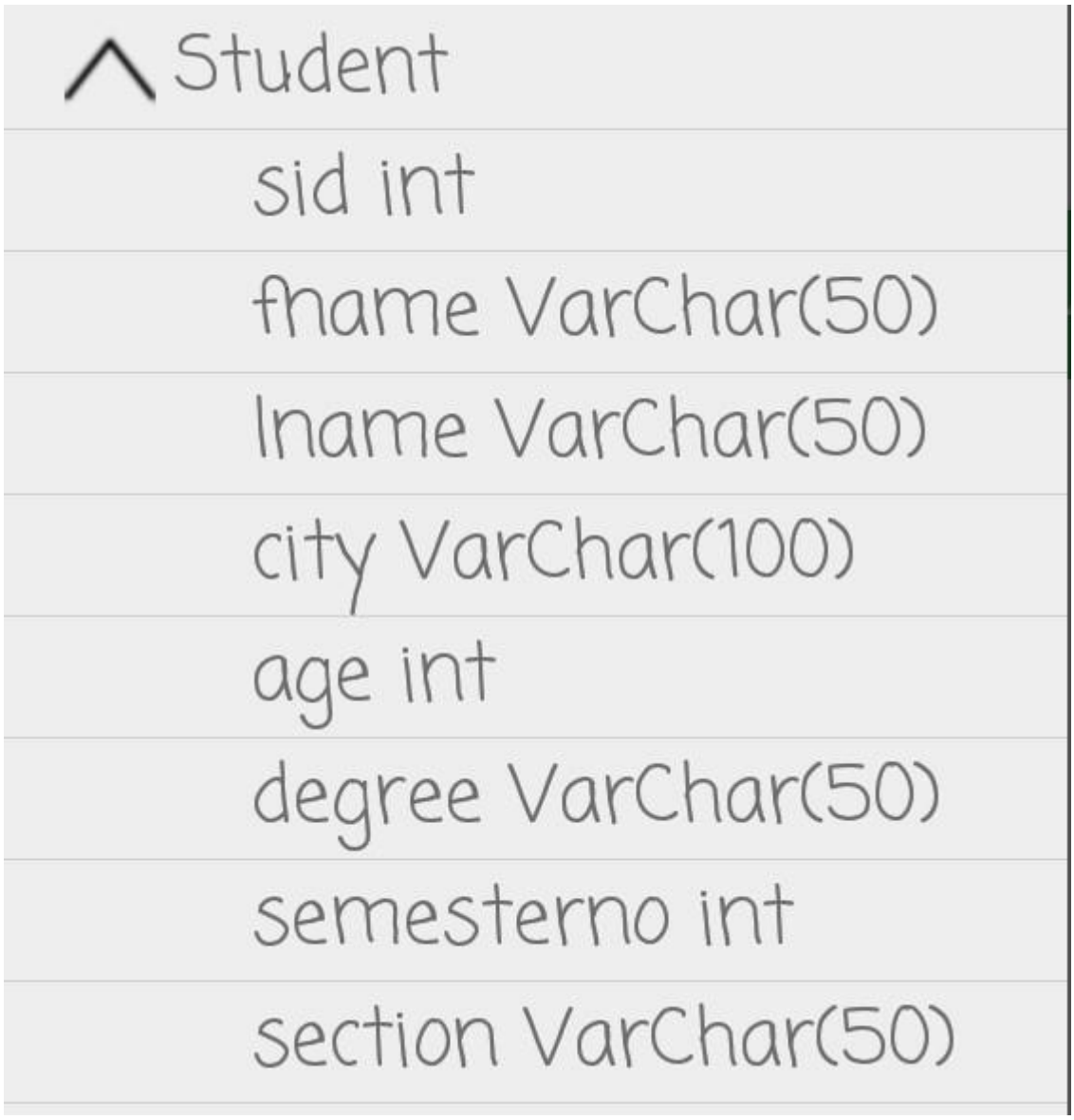




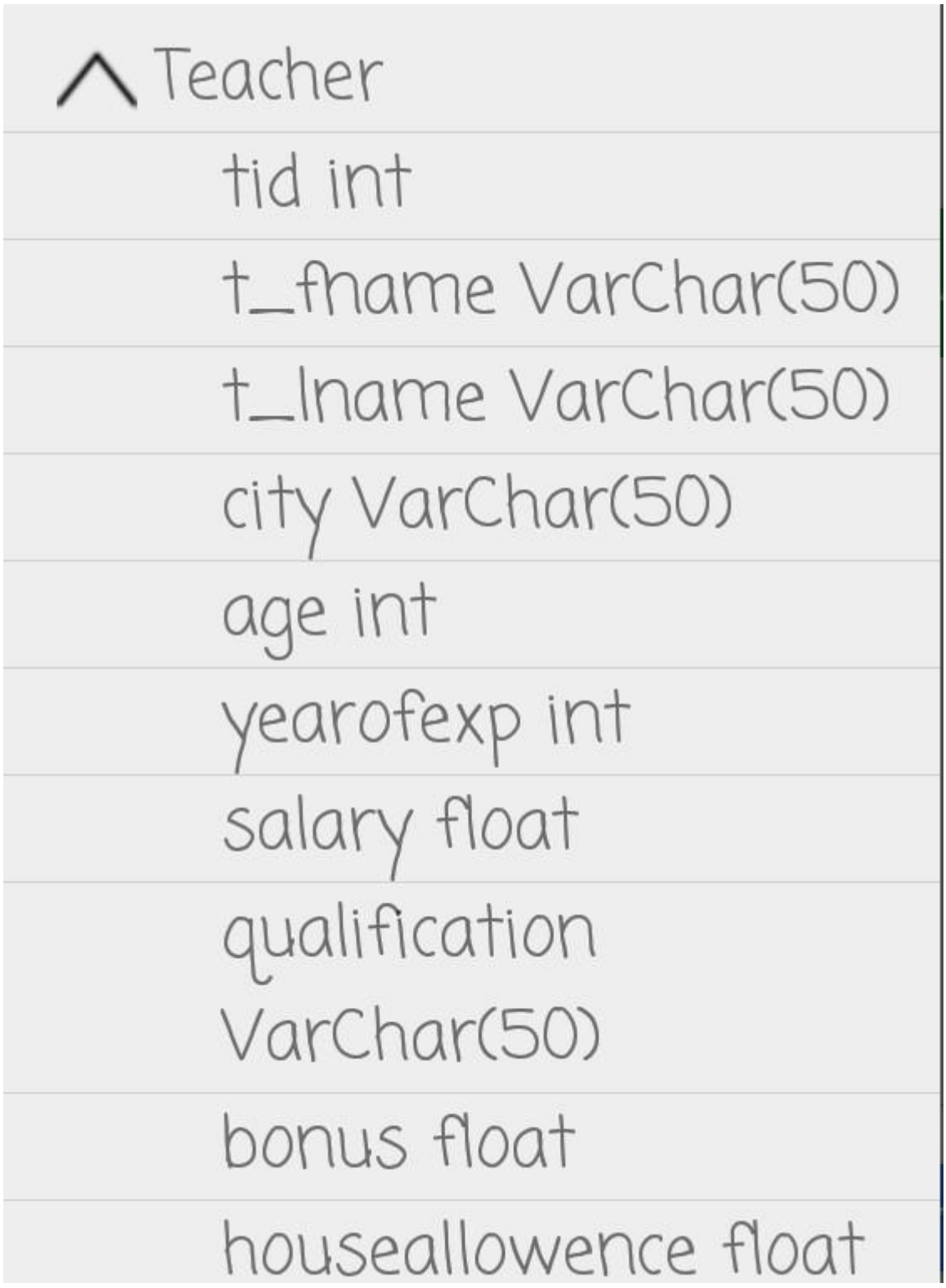
You can also use the sample queries. For this you will need to go to the sample tab.

Schema/ Meta-data

For Student

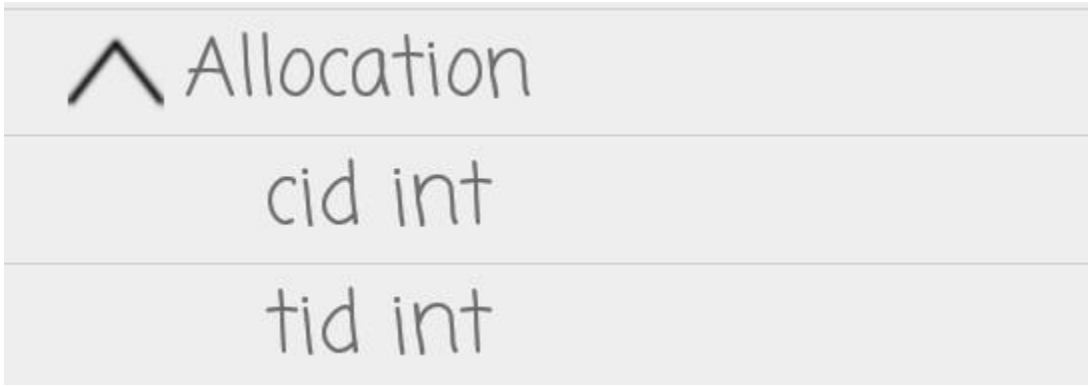


For Teacher


Teacher
tid int
t_fname VarChar(50)
t_lname VarChar(50)
city VarChar(50)
age int
yearofexp int
salary float
qualification VarChar(50)
bonus float
houseallowence float

For Course

For Allocation



For Enrollment

Writing Basic SQL Select Statement (Selecting All or Specific Columns

In this section we will learn about the SELECT and the SELECT * statements.

The SQL SELECT Statement

The SELECT statement is used to select data from a database.

The result is stored in a result table, called the result-set.

SQL SELECT Syntax

```
SELECT column_name(s) FROM table_name
```

And

```
SELECT * FROM table_name
```

Note: SQL is not case sensitive. SELECT is the same as select.

In the syntax:

1. SELECT is keyword.
    - selects all columns
2. Column_name (s) selects the named column.
3. FROM table specifies the table containing the columns.

An SQL SELECT Example

The "Student" table:

Person table stores the following data of person object; Persons S_id, fname, lname, city, age, degree, semesterno, section

Example of (*)
Select * From Student

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Selecting Specific columns:

Example:

Selecting Student's FName

```
Select fname from student
```

	fname
1	Ali
2	Asad
3	Jamal
4	Kaleem
5	Nazeer
6	Basit
7	Hassa
8	Kareem
9	Sulta
10	Nabeel

Example:

Selecting Multiple columns like Student's FName, LName, City.

```
Select fname, lname, city from student
```

	fname	lname	city
1	Ali	Ahmad	Rwp
2	Asad	Ahmad	Isb
3	Jamal	Khan	Multa
4	Kaleem	Iqbal	Rwp
5	Nazeer	Ahmad	Rwp
6	Basit	Ali	Isb
7	Hassa	Farooq	Multa
8	Kareem	Abid	Karachi
9	Sulta	Abdullah	Isb
10	Nabeel	Sohail	Rwp

Example:

Selecting Student sid, city, age

```
Select sid, city, age from student
```

	sid	city	age
1	1	Rwp	18
2	2	Isb	23
3	3	Multa	19
4	4	Rwp	20
5	5	Rwp	19
6	6	Isb	21
7	7	Multa	22
8	8	Karachi	23
9	9	Isb	18
10	10	Rwp	19

Example:

Selecting fname, age, cgpa, semesterno from student

```
Select fname, age, cgpa, semesterno from student
```

	fname	age	cgpa	semesterno
1	Ali	18	3.4	3
2	Asad	23	1.4	3
3	Jamal	19	2.7	7
4	Kaleem	20	3.9	3
5	Nazeer	19	3.4	3
6	Basit	21	3.4	3
7	Hassa	22	3.4	4
8	Kareem	23	3.4	3

Exercise 1

Student table stores the following data of student object;

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

1) Write a query that will display record of all student.

2) Write a query that will display only age of student.

3) Write a query that will display Roll_No and Name of student.

4) Write a query that will retrieve Discipline, Section and CGPA of a Students.

5) Write a query that will retrieve city, age and Name of a students.

6) Write a Sql Statement that will retrieve all record of Employees.

7) Write a Sql Statement that will retrieve name and Salary of Employees.

8) Write a Sql Statement that will retrieve Employee_id and Employees.

9) Write a Sql Statement that retrieve Quantity from Sales table.

10) Write a Sql Statement that will retrieve all record of sales.

Lab # 02

Arithmetic Expressions and Operators and Column Aliasing

Arithmetic Expression
Arithmetic expressions allow you to modify the way in which data is displayed, perform
Calculations, or look at what-if scenarios.
Here, we have a Teacher Database named Teacher (Teacher Management System). In this
Manual, we will use Teacher table with the attributes listed below.

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

Simple addition to a column value

Now Suppose we want to show salary by adding 2000 in salary of Teacher

```
Select salaray+2000 as [salary] from Teacher
```

The result would be like this.

	salary
1	112000
2	92000
3	132000
4	72000
5	752000
6	152000
7	173000
8	142000
9	112000
10	132000

Simple subtraction to a column value

Now Suppose we want to show Salary by Subtracting 200 in Salary of Teacher

```
Select salaray-200 as [salary] from Teacher
```

The result would like this.

	salary
1	109800
2	89800
3	129800
4	69800
5	749800
6	149800
7	170800
8	139800
9	109800
10	129800

Simple Multiplication to a column value

Now Suppose we want to show t_fname and bonus multiply by 2 of Teachers

```
Select t_fname, bonus*2 as [Bonus] from teacher
```

The result would like this.

	t_fname	Bonus
1	Salma	22000
2	Kamal	18000
3	Kaleem	28000
4	Jamal	34000
5	Nabeel	38000
6	Razaq	62000
7	Saad	82000
8	Saleem	23000
9	Tariq	22000
10	Ahmad	34000

Simple Division to a column value

Now Suppose we want to show t_lname, bonus by divided 2 of Teacher

```
Select t_lname, bonus/2 as [bonus] from teacher
```

The result would like this.

	t_lname	bonus
1	Iqbal	5500
2	Ahmad	4500
3	Rasheed	7000
4	Farooq	8500
5	Ahmad	9500
6	Ahmad	15500
7	Abid	20500
8	Mushtaq	5750
9	Sohail	5500
10	Iqbal	8500

Addition, Multiplication, Subtraction and Division to a column value

Now Suppose we want to show t_fname, t_lname ,salary, salary + 200, bonus – 9000, salary * 4, bonus /3

Example

```
Select t_fname, t_lname , salary + 200 as [salary + 200] ,
bonus-9000 as [bonus - 9000], salary * 4 as [salary * 4],
bonus/3 as [Bonus / 3] form teacher
```

The result would like this.

	t_fname	t_lname	salary	salary + 200	bonus - 9000	salary * 4	Bonus / 3
1	Salma	Iqbal	110000	110200	2000	440000	3666.66666666667
2	Kamal	Ahmad	90000	90200	0	360000	3000
3	Kaleem	Rasheed	130000	130200	5000	520000	4666.66666666667
4	Jamal	Farooq	70000	70200	8000	280000	5666.66666666667
5	Nabeel	Ahmad	750000	750200	10000	3000000	6333.33333333333
6	Razaq	Ahmad	150000	150200	22000	600000	10333.3333333333
7	Saad	Abid	171000	171200	32000	684000	13666.6666666667
8	Saleem	Mushtaq	140000	140200	2500	560000	3833.33333333333
9	Tariq	Sohail	110000	110200	2000	440000	3666.66666666667
10	Ahmad	Iqbal	130000	130200	8000	520000	5666.66666666667

Defining a Column Alias

Alias

Aliases can be used to create a temporary name for columns or tables.

- Renames a column heading.
- Is useful with calculations.
- Immediately follows column name; optional AS keyword between column name and Alias.
- Requires single quotation marks if it contains spaces or special characters or is case sensitive.

Syntax:

Column_name [As] Alias_name

Example:

Now suppose we want to show t_fName of teacher as Name
In Above Teacher Table

```
Select t_fName AS [Name] FROM teacher
```

	name
1	Salma
2	Kamal
3	Kaleem
4	Jamal
5	Nabeel
6	Razaq
7	Saad
8	Saleem
9	Tariq
10	Ahmad

```
Select tid AS [Teacher id] FROM Teacher
```

	Teacher id
1	5001
2	5002
3	5003
4	5004
5	5005
6	5006
7	5007
8	5008
9	5009
10	5010

Task 2

a. Write a Sql Statement to retrieve all column from Teacher Table.

b. Write a Sql Statement to retrieve any two column from Teacher table.

c. Write a Sql Statement to retrieve more than three column from Teacher table.

d. Write a Sql Statement to retrieve Age with addition of 5 from Teacher table.

e. Write a Sql Statement to retrieve t_fname column as Teacher Name from Teacher table.

Exercise 2

Student table stores the following data of student object

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

1. Write a query that will retrieve lname and age of a students. Age must be display/retrieve addition by 5.

2. Write a query that will retrieve sid and Cgpa of a student. Cgpa must be display/retrieve subtraction by 1.0.

3. Write a query that will retrieve age, degree and Section of a student. Age must be display multiply by 2.

4. Write a query that will retrieve fname, City and Cgpa of a student. Cgpa must be display division by 2.0.

5. Write a query to show name, age by Addition 10 in age, age by Subtract 5 in age, age by Multiply 2 in age and age by division 2 in age of Students.

6. Write a query to show fName of a students as a S_Name of a Student.

7. Write a query to display city of a student as a Student_City and age of a students as a Student_Age.

8. Write a Sql Statement that will display quantaty as Sales_Quantaty from Sales Table.

9. Write a Sql Statement will retrive quantaty with addation of 7 and Subtraction of 5 from Sales table.

10. Write a Sql Statement that will retrieve Sales_item, Sales_order, and quantaty must be division by 2 from sales table.

Lab # 03

Concatenation Operator, Distinct & Duplicate Record Retrieval

Concatenation Operator (+)

- You can link columns to other columns, arithmetic expressions, or constant values to create a character expression by using the concatenation operator (+).
- Columns on either side of the operator are combined to make a single output column.

Using Concatenation Operator

Consider the Student table as shown with attributes listed below:

- **Sid**
- **Fname**
- **Lname**
- **City**
- **Age**
- **Cgpa**
- **Degree**
- **Semesterno**
- **section**

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Since, the purpose of concatenation operator is to combine two columns into a single column in the output only; we can concatenate Person First Name with Last Name and display the
Full name in the output.

Example

The query for concatenation of First Name with Last Name of student is as follows:

```
Select [FName] + ' ' +[LName] as [Student Name] from Student;
```

RESULT

	Student Name
1	Ali Ahmad
2	Asad Ahmad
3	Jamal Khan
4	Kaleem Iqbal
5	Nazeer Ahmad
6	Basit Ali
7	Hassa Farooq
8	Kareem Abid
9	Sulta Abdullah
10	Nabeel Sohail

Example

The query for concatenation of Degree with section of student is as follows:

```
Select Degree + '  ' +Section as [Degree & Section] from
Student;
```

RESULT

	degree & section
1	BSIT A
2	BSIT B
3	BSCS A
4	BSIT A
5	BSIT B
6	BSIT A
7	BSIT A
8	MCS A
9	BSIT A
10	MCS A

Eliminating Duplicate Rows

Duplicate Rows

The default display of queries is all rows, including duplicate rows. Consider the table shown in Figure 1. In this table, when we retrieve all the records, we will
receive repeated values of degree since more than one student studies the same degree

Select * from Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Figure 1

If we wish to eliminate these duplicate rows, we can do this using DISTINCT keyword in the SELECT clause of SQL SELECT

```
Select distinct degree from student
```

	degree
1	BSCS
2	BSIT
3	MCS

Example distinct

Now we want to select only the distinct values from the column named "City" from the table above.

We use the following SELECT statement:

```
SELECT DISTINCT City FROM student
```

The result-set will look like this:

	city
1	Isb
2	Karachi
3	Multan
4	Rwp

Task 3

1 Write a Sql Statement to merged fname and lname as Full Name from Student table.

2 Write a Sql Statement to retrieve distinct lname from student.

3 Write a Sql Statement to retrieve all those student whose age is not duplicate/repeated.

4 Write a Sql Statement to retrieve sid as student id from student table.

5 Write a Sql Statement to retrieve all those semesterno whose value is not duplicate.

Exercise 3

Teacher table stores the following data of teacher object;

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

1 Write a query to concatenate t_fname and t_lname with city of a teacher as a Name City.

2 Write a query to merged city with qualification of a teacher as a city qualification.

3 Write a query to display different name of a teacher from teacher table.

4 Write a query to retrieve different city from teacher table.

5 Write a Sql Statement that will retrieve all those teachers whose age is distinct from teacher table.

6 Write a Sql Statement that will merge any two columns from teacher Table.

7 Write a Sql Statement that will merge three or more columns from teacher table.

8 Write a Sql Statement that will retrieve tid, city and yearsofexp as Experience from teacher Table.

9 Write a Sql Statement that will retrieve all those teachers whose Salary is not duplicate.

10 Write a Sql Statement that will retrieve all those teachers whose Salary and bonus is not duplicate.

Lab # 04

Sorting of Data, Selection and Projection

Sorting Data (Order by Keyword)

The ORDER BY keyword is used to sort the result-set by a specified column.

The ORDER BY keyword sort the records in ascending order by default.

If you want to sort the records in a descending order, you can use the DESC keyword.

SQL ORDER BY Syntax

```
SELECT column_name(s) FROM table_name ORDER BY column_name(s)
ASC|DESC
```

ORDER BY Example

The "student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to select all the student from the table above, however, we want to sort the students by their last name.

We use the following SELECT statement:

```
Select * from student order by lname
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
2	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
3	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
4	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
9	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example 2

Now we want to select all the students from the table above, however, we want to sort the students by their Age

We use the following SELECT statement:

```
Select * from student order by age
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
3	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A
4	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
7	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
8	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
9	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
10	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B

ORDER BY DESC Example 1

Now we want to select all the students from the table above, however, we want to sort the persons descending by their first name.

We use the following SELECT statement:

```
Select * from student order by fname desc
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
2	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
3	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A
4	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
5	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
6	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
9	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
10	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A

ORDER BY DESC Example 2

Now we want to select all the student from the table above, however, we want to sort the student descending order by their Age.

We use the following SELECT statement:

```
Select * from student order by age desc
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
2	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
3	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
4	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
5	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
6	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
7	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
8	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A
9	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
10	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A

Selection (Where Clause)

Selection: You can use the selection capability in SQL to choose the rows in a table that
you want returned by a query. You can use various criteria to selectively restrict the rows that
you see using Where clause in SQL Select statement.

The **WHERE clause** is used to filter records.

The WHERE clause is used to extract only those records that fulfill a specified criterion.

SQL WHERE Syntax

```
SELECT column_name(s)FROM table_name WHERE column_name
operator value
```

Example: 1

The "students" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to select only those student living in the city "Rwp" from the table above.

We use the following SELECT statement:

```
SELECT * FROM student WHERE City='Rwp'
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example: 2

Now we want to select only those students who have age equal to 18 in student table.

```
Select * from student where age=18
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A

Example: 3

Now we want to select only those students who have M_Income cgpa = 3.4 from the student table.

```
Select * from student where cgpa = 3.4
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
3	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
6	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
7	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Quotes around Text Fields

SQL uses single quotes around text values (most database systems will also accept double quotes).

Although, numeric values should not be enclosed in quotes.

For text values:

```
This is correct:
SELECT * FROM student WHERE fname='shahid'
This is wrong:
SELECT * FROM student WHERE fname=shahid
```

For numeric values:

```
This is correct:
SELECT * FROM student WHERE age=18
This is also correct:
SELECT * FROM student WHERE age='18'
```

Task 4

Teacher table stores the following data of teacher object;

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

1 Write a query that will age of teachers order by age in descending order descending.

2 Write a query that will retrieve data of teachers whose bonus is 11000.

3 Write a query that will display yearsofexp of those teachers whose houseallowence is 22700

4 Write a query that will display all teachers whose tid is 5009.

5 Write a query that will retrieve t_fname and t_fname as full name of teachers whose qualification is Ph D

Exercise 4

1 Write a query that will retrieve data of all teachers order by t_fname in ascending order.

2 Write a query that will retrieve data of all teacher order by age in descending order.

3 Write a query that will display all teacher who live in 'Rwp'.

4 Write a query that will retrieve all teacher whose t_lname is 'Iqbal'.

5 Write a query that will retrieve all Teacher whose qualification is ‘Ph D’.

6 Write a Sql Statement that will not retrieve duplicate city from teacher order by city Descending.

7 Write a Sql Statement that will not retrieve duplicate Salary from all teacher order by name Ascending.

8 Write a Sql Statement that will merge yearsofexp and bonus from teachers where bonus is equal to 9000

9 Write a Sql Statement that will merge Name and Age as a “Name Age” from teacher table order by tid descending.

10 Write a Sql Statement that will retrieve distinct name and city from student table order by name ascending.

Lab #05

Logical Operators and WHERE Clause

Logical Operators

Logical operators are used in SQL Statement's Where clause.

Logical AND

- The AND operator displays a record if both the first condition and the second condition is true.
- You can restrict the records displayed in the output by supplying more than one conditions in the where clause.
- All the records meeting the conditions (Where AND) is displayed in the output.

Consider the Student table as shown with attributes listed below:

- **Sid**
- **Fname**
- **Lname**
- **City**
- **Age**
- **Cgpa**
- **Degree**
- **Semesterno**
- **section**

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

AND Syntax

```
SELECT column1, column2, FROM table_name
WHERE condition1 AND condition2 AND condition3
```

Example: 1

Now we want to select only those students with the first name equal to "Ali" AND the last name equal to "ahmad":

We use the following SELECT statement:

```
SELECT * FROM student WHERE fname='ali' AND lname='ahmad'
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A

Example: 2

Now we want to select only those students with the first name equal to "Saad" AND the last name equal to "Amir":

```
SELECT * FROM student WHERE FName='Saad' AND LName='Khan'
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section

Logical OR

The OR operator displays a record if either the first condition or the second condition is true.

OR Syntax

```
SELECT column1, column2 FROM table_name
WHERE condition1 OR condition2 OR condition3
```

Example: 1

Now we want to select only those students with the first name equal to "ali" OR the last name equal to "ahmad":

We use the following SELECT statement:

```
SELECT * FROM student WHERE FName='ali' OR LName='ahmad'
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B

Example: 2

Now we want to select only those persons with the last name equal to 'ahmad' or cgpa=3.4

```
SELECT * FROM student
WHERE lName='ahmad' OR cgpa=3.4
```

The result-set will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
5	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
6	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
7	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
8	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Logical NOT

- The NOT operator displays a record if the condition(s) is NOT TRUE.
- Logical NOT means exclude the records where the condition is true and display other records.

NOT Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

Example: 1

The following SQL statement selects all fields from students where city is NOT 'rwp';

```
Select * from student where not city='rwp'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
6	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A

Example: 2

Now we want to select all those students where cgpa not equal to 3.4 from the student table.

```
Select * from student where not cgpa=3.4
```

The result will look like this

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A

Combining AND, OR and NOT

You can also combine the AND, OR and NOT operators.

The following SQL statement selects all fields from "Student" where city is "rwp" AND age must be 19 OR 21 (use parenthesis to form complex expressions):

```
Select * from student where city='rwp' and (age=23 or age=24)
```

The result will look like this

sid	fname	lname	city	age	cgpa	degree	semesterno	section

Task 5

The tables and the data in the tables from which the queries are written are given below:

Course

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Scenario: select + from + where

Query 1: Write a query that will retrieve all data from course where category =cs.

Query 2: Write a query that will retrieve title of course where cr_hours is equal to 10000.

Query 3: write a query that will retrieve cr_hours of course where category is either cs or maths.

Query 4: write a query that will retrieve all the data from coursewhere cid is equal to 1010 and category is not cs.

Query 5: retrieve id of all those courses that belong to title 'networks'.

Exercise 5

Student table stores the following data of student object

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Assume the following Schema to write following queries that's given below.

1 Write a query that will display all students whose cgpa is equal to "1.4" and Section is equal to "A".

2 Write a query that will display name as a Student Name of all students whose age is equal 23 and order by name descending.

3 Write a query that will retrieve sid, age and name as Student Name from students who live in “ISB” or “Rwp” order by city Ascending.

4 write a query that will retrieve sid, Name, age, city and degree where degree not equal to “MCS” order by age descending.

5 Write a query that will retrieve data of all student whose Discipline is equal to “BSCS” and city must be “Rwp” or “Multan”.

6: write a query that will retrieve all students where age is equal to 10.

7: write a query that will retrieve the cgpa from students where age is either 18 or 20.

8: Write a Sql Statement to retrieve fname and lname of all students where age is equal to 18.

9: Write a Sql Statement to retrieve all students where cgpa not equal to 2.4 and 3.4

10: Retrieve all those students whose lname is ali.

Lab # 06

Comparison Operators

Comparison Operators

- A comparison (or relational) operator is a mathematical symbol which is used to compare two values.
- Comparison operators are used in conditions that compares one expression with another. The result of a comparison can be TRUE, FALSE, or UNKNOWN (an operator that has one or two NULL expressions returns UNKNOWN).

The following table describes different types of comparison operators –

Operator	Description
=	Equal to.
>	Greater than.
<	Less than.
>=	Greater than equal to.
<=	Less than equal to.
<>	Not equal to.

Example Equal Operator: (=)

The "student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to select only those students living in the city "Rwp" from the table above.

We use the following SELECT statement:

```
SELECT * FROM student where city='rwp'
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example Greater than: (>)

Now we want to select all those Students whose age is greater than 20.

```
Select * from student where age>20
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
2	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
3	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
4	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A

Example Less than: (<)

Now we want to select all those students whose cgpa is less than 2.5

```
Select * from student where cgpa<2.5
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B

Example Greater than equal to: (>=)

Now we want to select all those students whose age is greater than or equal to 20.

```
Select * from student where age>=20
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A

Example Less than equal to: (<=)

Now we want to select all those Students whose cgpa is less than or equal to 3.4.

```
Select * from student where cgpa <= 3.4
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
5	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
6	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
7	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
8	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A
9	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example Not equal: (!=, <>)

Example: 1(!=)

Now we want to select all those Students whose cgpa is not equal to 3.4.

```
Select * from student where cgpa != 3.4
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A

Example: 2(<>)

Now we want to select all those Students whose city is not equal to rwp.

```
Select * from student where city<>'rwp'
```

The result will look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
6	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A

Task 6

The tables and the data in the tables from which the queries are written are given below:

Course

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

1 Write a query that will retrieve all data from course where category=CS.

2 Write a query that will retrieve title of courses where credit hours are equal to 4.

3 Write a query that will retrieve category of employees where title is equal to calculus.

4 Write a query that will retrieve all the data from course where category is CS and cr_hours is not 3.

5 Retrieve id of all those courses where cid is less than 1005.

Exercise 6

Student table stores the following data of student object

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

1 Write a query that will retrieve data of all students whose enrolled in Degree “BSCS” and Cgpa greater than 3.3.

2 Write a query that will retrieve sid, Name, Age, degree and Cgpa of a Students who’s Cgpa less than equal to 3.5 order by age descending.

3 Write a query that will retrieve Name as "Full Name" and degree, Section as a "Discipline Section" of Students who's Cgpa greater than equal to 3.2 order by section descending.

4 Write a query that will display all those students whose age not equal to 23 and also order by name descending.

5 Write a query that will retrieve data of all students whose age greater than equal to 24 not living in "Isb" and "Multan".

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

6: Write a query that will retrieve all Teacher where age is greater than 30

7: Write a query that will retrieve the qualification where bonus is either 9000 or 11000

8: Retrieve Name and tid of all teachers whose yearsofexp is greater than equal to 10

9: Retrieve all Teachers where house allowance is greater than 20000 and less than 50000

10: Retrieve all those teachers whose qualification is MS

Lab # 07

Logical Operators (BETWEEN, IN, NOT) and Top Clause

Logical Operators

Logical operators test for the truth of some condition. Logical operators, like comparison operators, return a Boolean data type with a value of TRUE, FALSE, or UNKNOWN.

Comparison Operators	Description
IN	Compare a value to a list of literal values that have been specified.
BETWEEN	Search for values that are within a set of values, given the minimum value and the maximum value.
IS NULL	Compare a value with a NULL value.
NOT	Reverses the meaning of the logical operator with which it is used. E.g: NOT BETWEEN, NOT IN, etc.

The BETWEEN Operator

The BETWEEN operator selects a range of data between two values. The values can be numbers, text, or dates.

SQL BETWEEN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

BETWEEN Operator Example: 1

The Student table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to select the Student with a sid between 3 and 6 from the table above.

We use the following SELECT statement:

```
Select * from student where sid between 3 and 6
```

The result will be like this

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	6	Basit	Ali	Isb	21	3.4	BSIT	3	A

BETWEEN Operator Example: 2

Now we want to select the Student with Age between 20 and 21 from the table above.

We use the following SELECT statement:

```
Select * from student where age between 20 and 21
```

The result will be like this

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
2	6	Basit	Ali	Isb	21	3.4	BSIT	3	A

SQL IN Operator

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

SQL IN Syntax

```
SELECT column_name(s)FROM table_name

WHERE column_name IN (value1,value2,...)
```

IN Operator Example

The student table

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to select the student fName equal to "Ali" or "Nabeel" from the table above.

We use the following SELECT statement:

```
Select * from student where fname in('ali','nabeel')
```

The result will be like this

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example: 2

Now we want to select the student with city equal to "Rwp" or "isb" from the table above.

We use the following SELECT statement:

```
Select * from student where city in('rwp','isb')
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
4	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
5	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
6	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
7	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Not Null Operator

Now we want to select the students with city is equal to **NOT NULL** from the table student.

We use the following SELECT statement:

```
Select * from student where city is not null
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

NOT BETWEEN

Example

Now we want to select the students whose age is not between 19 and 22.

```
Select * from student where age not between 19 and 22
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
4	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A

Example 2

Now we want to select the students with cgpa not between 3.4 and 4.5 from the table student.

```
Select * from student where cgpa not between 3.4 and 4.5
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A

Not In Example

Now we want to select the students with sid not in 3,6,8 from student table

```
Select * from student where sid not in (3,6,8)
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
3	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
4	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
5	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
6	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A
7	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

BETWEEN with IN

Example

Now we want to select the student with Age BETWEEN 19 and 20. In addition; do not show student with city in karachi and multan

```
Select * from student where (age between 19 and 20) and city
not in('karachi','multan')
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

SQL TOP CLAUSE

The TOP clause is used to specify the number of records to return.

The TOP clause can be very useful on large tables with thousands of records. Returning a large number of records can impact on performance.

Note: Not all database systems support the TOP clause.

SQL Server Syntax

```
SELECT TOP number|percent column_name(s)
FROM table_name
```

SQL TOP Example

The "teacher" table:

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

Now we want to select only 50% of the records from teacher table.

We use the following SELECT statement:

```
Select top 50 percent * from teacher
```

The Result will be like this:

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000

Example 2

Now we want to select only top 3 of the records from teacher table.

We use the following SELECT statement:

```
Select top 3 * from teacher
```

The Result will be like this:

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000

Task 7

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

Use of In

Scenario: select + from + where + in ()

Query 1: Retrieve distinct states where tid is 5001, 5005, 5006.

Query 2: Retrieve distinct salary from teacher where city is Rawalpindi and Mianwali.

Query 3: Retrieve salary and state where bonus is 11000, 9000 order by salary.

Query 4: Display distinct Full name from teacher where t_lname is abid, ahmad order by name descending.

Query 5: Retrieve all those teachers whose house allowance is 18000, 20000, 50000

Use of Between

Scenario: select + from + where + between

Query 1: retrieve tid, salary from teacher where salary greater than 36000 and less than 10000.

Query 2: retrieve t_fname from teacher where salary greater than 35000 and less than 6000.

Query 3: retrieve salary, city where tid less than 5003 and greater than 5007.

Query 4: retrieve salary, city from teacher where tid greater than 5005 and less than 5009.

Query 5: retrieve salary, bonus, t_fname from teacher where salary greater than 108000 and less than 120000.

Exercise 7

Employee table stores the following data of student object

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

1 Write a query to display the name (fName, lName) and age for all students whose age is not in the range of 19 and 21

2 Write a query to display all the information of students whose cgpa is in the range of 3.5 to 4.0 and age is not equal to 19.

3 Write a query to display all the information of employees whose city is not in rwp and isb and age is greater than equal to 20.

4 Write a query to display the full name (first and last name), and city for all students whose section is not A.

5 Write a query to select all record from students where last name in 'ali', 'ahmed'

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

6 Write a query to display the full name (first and last) name, and salary, for all teachers whose salary is out of the range 100000 and 125000 and make the result set in descending order by the full name.

7 Write a query to display the full name (first and last), house allowance and yearsofexp for those teachers who was doing Ph D and make the result set in ascending order by the age.

8 Write a query that will retrieve top two record from Teachers order by Salary Ascending.

9 Write a query that will top 50 percent record from teachers table order by age descending.

10 Write a Sql Statement that will retrieve top 70 percent record from Teachers whose salary is greater than equal to 120000.

Lab #08

Comparison Operators (LIKE, NOT LIKE)

The SQL LIKE Operator

There are two wildcards used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiple characters
- _ - The underscore represents a single character

SQL LIKE Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern
```

Here are some examples showing different LIKE operators with '%' and '_'
Wildcards:

WHERE E_fname LIKE 'a%'	Finds any values that starts with "a"
WHERE E_fname LIKE '%a'	Finds any values that ends with "a"
WHERE E_fname LIKE '%a%'	Finds any values that have "a" in any position
WHERE E_fname LIKE 'al%'	Finds any values that have "al" at start
WHERE E_fname LIKE '_a%'	Finds any values that have "a" in the second position
WHERE E_fname LIKE 'a%_%_'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE E_fname LIKE 'a%d'	Finds any values that starts with "a" and ends with "d"
WHERE E_fname LIKE 'a%d_'	Finds any values that starts with d and have "d" in the second last position

LIKE Operator Example

The "Student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example 1(like start)

Now we want to select the student living in a city that starts with "R" from the above table.

We use the following SELECT statement:

```
Select * from student where city like 'r%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

- The "%" sign can be used to define wildcards (missing letters in the pattern) both before and after the pattern.

Example 2(like end)

Now we want to select the student living in a city that End with "b" from the above table.

We use the following SELECT statement:

```
Select * from student where city like '%b'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
3	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A

Example 2(like any position)

Now we want to select the students having fname containing "s" at any position the above table.

We use the following SELECT statement:

```
Select * from student where fname like '%s%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
3	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
4	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A

Example (2nd position)

SQL statement selects all students with lname that have "a" in the second position:

```
Select * from student where lname like '_a%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A

Example (Like Not Start)

The following SQL statement selects all students with city that does NOT start with "r":

```
Select * from student where city not like 'r%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	lsb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	6	Basit	Ali	lsb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
6	9	Sulta	Abdullah	lsb	18	3.4	BSIT	3	A

Example (Like Start charlist)

The following SQL statement selects all student with fname that start with "r",”m”,”a” from the persons table.

```
Select * from student where fname like '[rma]%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B

Example (Like Not Start charlist)

The following SQL statement selects all student with city that Not start with "r",”i”,”a” from the persons table.

```
Select * from student where city not like '[ria]%'

OR

Select * from student where city like '[^ria]%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
2	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
3	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A

Example (like start and at least 4 character)

The following SQL statement selects all student with a FName that starts with "b" and are at least 4 characters in length:

```
Select * from student where fname like 'b_%_%_%'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	6	Basit	Ali	Isb	21	3.4	BSIT	3	A

Example (like start and end character)

The following SQL statement selects all student with a LName that starts with "a" and ends with "d":

```
Select * from student where lname like 'a%d'
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
4	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A

Task 8

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

Use of Like

Scenario: select+from+where+like

Query 1: retrieve all those teachers where t_lname first letter is I.

Query 2: retrieve all those teachers where fname starts with T.

Query 3: retrieve full name from teacher whose last letter of t_lname is q.

Query 4: retrieve salary from teachers where 2nd letter of t_fname is a.

Query 5: retrieve all those teachers where t_lname contains ee.

Exercise 8

1 Write a query to display the First Name of all teacher who have both "b" and "c" in their first name

2 Write a query to display all Teachers whose Last_Name start with “j” and end with “d”.

3 Write a query to display the First_Name, Last_Name as Full_Name of all teachers who’s First_Name Start with “a” or “b” or “c’ order by Last_Name Descending.

4 Write a query to display the last name of teachers having 'e' as the third character.

5 Write a query to display the last name of teachers whose names have exactly 6 characters.

6 Write a query to display the Name (First_Name, Last_Name) and salary for all teachers whose experience is greater than 8.

7 Write a Sql Statement that will retrieve all those teahcers whose city contain “a” at any position.

8 Write a Sql Statement that will retrieve name of teacher whose t_lname contain pattern “A”, ”b”, ”d” at any position.

9 Write a Sql Statement that will retrieve name of teacher whose t_lname does not contain pattern “A”, ”b”, ”d” at any position

10 Write a Sql Statement that will retrieve all teachers whose city not start with “a” and not end with “d”.

Lab # 09

Aggregate function (min, max, avg, count, sum)

SQL Aggregate Functions

SQL aggregate functions return a single value, calculated from values in a column.

Useful aggregate functions:

- AVG() - Returns the average value
- COUNT() - Returns the number of rows
- MAX() - Returns the largest value
- MIN() - Returns the smallest value
- SUM() - Returns the sum

SQL AVG() Function

The AVG () Function

The AVG () function returns the average value of a numeric column.

SQL AVG() Syntax

```
SELECT AVG(column_name) FROM table_name
```

SQL AVG () Example

We have the following "student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to find the average value of the "cgpa" fields.

We use the following SQL statement:

```
SELECT AVG(cgpa) AS [CGPA Average] FROM Student
```

The result-set will look like this:

	CGPA Average
1	3.18

Example 2

Now we want to find the average age of students

We use the following SQL statement:

```
Select avg (age) as [average age] from student
```

The result-set will look like this:

	Average age
1	20

SQL COUNT () Function

SQL has many built-in functions for performing calculations on data.

SQL COUNT () Function

The COUNT () function returns the number of rows that matches a specified criteria.

SQL COUNT (column_name) Syntax

```
SELECT COUNT(COLUMN_NAME) FROM TABLE_NAME
```

SQL COUNT (*) Syntax

The COUNT(*) function returns the number of records in a table:

```
SELECT COUNT(*) FROM TABLE_NAME
```

SQL COUNT (DISTINCT column_name) Syntax

The COUNT (DISTINCT column_name) function returns the number of distinct values of the specified column:

```
SELECT COUNT(DISTINCT COLUMN_NAME) FROM TABLE_NAME
```

Note: COUNT(DISTINCT) works with ORACLE and Microsoft SQL Server, but not with Microsoft Access.

SQL COUNT (column_name) Example

We have the following "course" table:

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Example 1

Now we want to count the number of titles having cr_hours =3.

We use the following SQL statement:

```
SELECT COUNT(TITLE) AS [TOTAL TITLES]FROM COURSE WHERE
CR_HOURS=3
```

The Result will be like this:

	Total titles
1	9

SQL COUNT (*) Example

If we omit the WHERE clause, like this:

```
SELECT COUNT(*) AS [Total Course] FROM Course
```

The result-set will look like this:

	Total Course
1	10

SQL COUNT (DISTINCT column_name) Example

Now we want to count the number of unique categories in the "course" table.

We use the following SQL statement:

```
SELECT COUNT(DISTINCT CATEGORY) AS [TOTAL CATEGORIES] FROM
COURSE
```

The result-set will look like this:

	total categories
1	3

The SQL MAX () Functions

The MAX () function returns the largest value of the selected column.

SQL MAX () Syntax

```
SELECT MAX(COLUMN_NAME) FROM TABLE_NAME
```

SQL MAX () Example

We have the following "Course" table:

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Now we want to find the largest value of the "cid" column.

We use the following SQL statement:

```
SELECT MAX(CID) AS [MAX COURSE ID] FROM COURSE
```

The Result will be like this:

	Max Course ID
1	1010

SQL MIN () Function

The MIN () Function

The MIN () function returns the smallest value of the selected column.

SQL MIN () Syntax

```
SELECT MIN(column_name) FROM table_name
```

SQL MIN () Example

We have the following "Course" table:

Now we want to find the smallest value of the "cr_hours" column.

We use the following SQL statement:

```
SELECT MIN(cr_hours) AS [Min Credit Hours] FROM course
```

The result-set will look like this:

	Min Credit Hours
1	3

SQL SUM () Function

The SUM () Function

The SUM () function returns the total sum of a numeric column.

SQL SUM () Syntax

```
SELECT SUM(column_name) FROM table_name
```

SQL SUM () Example

Now we want to find the sum of all "Cr_hours" field.

We use the following SQL statement:

```
SELECT SUM(CR_HOURS) AS [TOTAL HOURS] FROM COURSE
```

The result-set will look like this:

	TOTAL HOURS
1	31

Task 9

Teacher Table

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

“Functions”
1:“MAX”

Scenario: select+max () +from

Query 1: retrieve max salary from Teacher.

Query 2: retrieve max Bonus from teacher where t_fname starts with S.

Query 3: retrieve max yearsofexp from Teachers where age is in 30 and 35.

Query 4: retrieve max house allowance from teacher where qualification is MS

Query 5: retrieve max salary where age is greater than 25 and less than 30.

2:"MIN"

Scenario: select + min ()+from

Query 1: retrieve min salary from Teacher.

Query 2: retrieve min Bonus from teacher where t_fname starts with S.

Query 3: retrieve min yearsofexp from Teachers where age is in 30 and 35.

Query 4: retrieve min house allowance from teacher where qualification is MS

Query 5: retrieve min salary where age is greater than 25 and less than 30.

3:"Count"

Scenario: select + count () +from

Query 1: retrieve total number of distinct yearsofexp from teacher.

Query 2: retrieve total number of salaries whose qualification is MS.

Query 3: retrieve total number of distinct t_name from teacher.

Query 4: retrieve total number of salaries from sales where age is greater than 28.

Query 5: retrieve total number of distinct qualifications from teacher

4: “Sum”

Scenario: select + sum () + from

Query 1: show total salary from teachers.

Query 2: retrieve total years of experience from teachers.

Query 3: retrieve total Bonus from Teacher where tid is 5001 and 5007.

Query 4: retrieve total House Allowance where age is greater than 30.

Query 5: retrieve sum of all bonus where t_lname is Ahmad.

5:“Avg”

Scenario: select +avg () + from

Query 1: show Average salary from teachers.

Query 2: retrieve Average years of experience from teachers.

Query 3: retrieve Average Bonus from Teacher where tid is 5001 and 5007.

Query 4: retrieve Average House Allowance where age is greater than 30.

Query 5: retrieve Average of all bonus where t_lname is Ahmad.

Exercise 9

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

1 Write a query to display Maximum cgpa of Students.

2 Write a query to display the highest First_Name alphabetically from student table.

3 Write a query to display Minimum cgpa of Students.

4 Write a query to Display the lowest Last_Name alphabetically from student table.

5 Write a query to display “average Age” of all students.

6 Write a query to display ‘total cgpa” of all Students.

7 Write a query to display the highest, lowest and average age from student table.

8 Write a query to display total Number of students

9 Write a query to display total number of students whose city is distinct.

10 Write a query to display total number of Students whose Last_Name is different.

Lab # 10

Aggregate Functions (GROUP Clause and HAVING Clause)

The SQL GROUP BY Statement

- The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.
- The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

GROUP BY Syntax

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
```

OR

```
SELECT column_name(s) FROM table_name WHERE condition GROUP
BY column_name(s) ORDER BY column_name(s)
```

SQL GROUP BY Example

We have the following "Course" table:

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Now we want to find the total Categories of each Course.
We will have to use the GROUP BY statement to group the customers.
We use the following SQL statement:

```
Select category, count(category) as [total] from course group
by category
```

The Result will be like this:

	category	total
1	CS	3
2	Management	3
3	Math	4

Explanation of why the above SELECT statement cannot be used: The SELECT statement above has two columns specified (Category and Count (category). The "count(category)" returns a single value (that is the total number of the "category" column), while "category" returns 6 values (one value for each row in the "Orders" table). This will therefore not give us the correct result. However, you have seen that the GROUP BY statement solves this problem.

GROUP BY More Than One Column

We can also use the GROUP BY statement on more than one column, like this:

```
select category, count(cr_hours)as[Credit Hours] from course
group by category,cr_hours
```

The Result will be like this:

	category	Credit Hours
1	CS	2
2	Management	3
3	Math	4
4	CS	1

SQL HAVING Clause

The HAVING Clause

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

SQL HAVING Syntax

```
SELECT column_name, aggregate_function(column_name) FROM
table_name

WHERE column_name operator value

GROUP BY column_name

HAVING aggregate_function(column_name) operator value
```

SQL HAVING Example

We have the following "Course" table:

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Now we want to find if any of the category have a total credit of less than 11.

We use the following SQL statement:

```
Select category,sum(cr_hours) as[total Credit Hours] from
course group by Category having sum(cr_hours)<11
```

	category	total Credit Hours
1	CS	10
2	Management	9

Example 2

Now we want to find if the Category "CS" or "Math" have a total Cr_hours of more than 10.

We add an ordinary WHERE clause to the SQL statement:

```
Select category,sum(cr_hours) as [total Hours] from Course
where category in('cs','math') group by category having
sum(cr_hours)>10
```

The Result will be like this:

	category	total Hours
1	Math	12

Task 10

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Query 1: Write a Sql Statement to retrieve no of Students

Query 2: Write a Sql Statement to retrieve fname, minimum age from student group by section.

Query 3: Write a Sql Statement to retrieve lname, maximum cgpa from student group by city.

Query 4: Write a Sql Statement to retrieve sid, total age from student group by degree having total age less than 50

Query 5: Write a Sql Statement to retrieve Name, average age from student group by section having average age greater than 50

Exercise 10

Teacher

	tid	t_fname	t_lname	city	age	yearofexp	salary	qualification	bonus	houseallowence
1	5001	Salma	Iqbal	Rawalpindi	30	7	110000	Ph D in progress	11000	22000
2	5002	Kamal	Ahmad	Muzafar Garh	35	14	90000	Ph D	9000	18000
3	5003	Kaleem	Rasheed	Gojar Kha	32	7	130000	MS	14000	25000
4	5004	Jamal	Farooq	Mianwali	28	4	70000	MS	17000	12000
5	5005	Nabeel	Ahmad	Mianwali	28	4	750000	MS	19000	20000
6	5006	Razaq	Ahmad	M B Di	35	10	150000	Ph D	31000	52000
7	5007	Saad	Abid	Rawalpindi	34	13	171000	MS	41000	62000
8	5008	Saleem	Mushtaq	Islamabad	30	5	140000	MS	11500	22700
9	5009	Tariq	Sohail	Rawalpindi	33	7	110000	MS	11000	22000
10	5010	Ahmad	Iqbal	Rawalpindi	30	7	130000	MS	17000	24000

"Group By"

Scenario: select + from +where +group by

Query 1: retrieve no. of Teachers in each City

Query 2: retrieve Qualification, minimum salary from teacher in each city

Query 3: retrieve city, maximum bonus from teacher in each qualification

Query 4: retrieve City, total salary from teacher in each city

Query 5: retrieve qualification, average house allowance from teacher group by qualification

"Having"

Scenario: Select + from + group by + having

Query 1: retrieve no. of teacher in each city where city is Rawalpindi

Query 2: retrieve qualification, minimum salary from teacher in from distinct qualification where minimum salary greater than 120500

Query 3: retrieve city maximum salary from teacher in each city where maximum salary less than 120000

Query 4: retrieve city, total salary from teacher in each city where total salary is less than 1900000

Query 5: retrieve city, number of teachers from teacher in each city where number of teacher is greater than 4

Lab # 11

Create Database, Create Table, Alter Table, Adding Constraints

Create Database

The CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a database.

SQL CREATE DATABASE Syntax

```
CREATE DATABASE database_name
```

CREATE DATABASE Example

Now we want to create a database called "testDB".

We use the following CREATE DATABASE statement:

```
CREATE DATABASE myDBS
```

To use your database. Write a simple query that given below.

```
Use myDBS
```

SQL CREATE TABLE Statement

The CREATE TABLE Statement

The CREATE TABLE statement is used to create a table in a database.

SQL CREATE TABLE Syntax

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
)
```

CREATE TABLE Example

Now we want to create a table called "Persons" that contains five columns: C_Id, L_Name, F_Name, Age, M_Income and City.

We use the following CREATE TABLE statement:

```
CREATE TABLE student
(
   Sid int, fname varchar(50), lname varchar(50),
   City varchar(100), age int, degree(50), semesterno int,
   Section varchar(50),primary key(sid)
)
```

The sid and age column is of type int and will hold a number. The LName, FName and City columns are of type varchar with a maximum length of 50 characters. The empty "student" table will now look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section

SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype;
```

Example:-

```
Alter table student add DateOfBirth int
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section	DateOfBirth

Change Data Type Example

Now we want to change the data type of the column named "DateOfBirth" in the "Student" table.

We use the following SQL statement:

```
ALTER TABLE Persons
ALTER COLUMN DateOfBirth year;
```

Notice that the "DateOfBirth" column is now of type year and is going to hold a year in a two- or four-digit format.

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

DROP COLUMN Example

Next, we want to delete the column named "DateOfBirth" in the "Student" table.

We use the following SQL statement:

```
ALTER TABLE student
DROP COLUMN DateOfBirth;
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section

ALTER TABLE - ALTER/MODIFY COLUMN

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype;
```

Change Data Type Example

Now we want to change the data type of the column named "cgpa" in the "student" table.

We use the following SQL statement:

```
ALTER TABLE Student
ALTER COLUMN semesterno varchar(2);
```

Rename column in table

You cannot use the ALTER TABLE statement in SQL Server to rename a column in a
table. However, you can use sp_rename, though Microsoft recommends that you drop
and recreate the table so that scripts and stored procedures are not broken.

Syntax

The syntax to rename a column in an existing table in SQL Server is:
sp_rename 'table_name.old_column_name', 'new_column_name', 'COLUMN';

Example:

```
Sp_rename‘student.fname’,’first name’
```

Rename table

You cannot use the ALTER TABLE statement in SQL Server to rename a table.
However, you can use sp_rename, though Microsoft recommends that you drop and
recreate the table so that scripts and stored procedures are not broken.

Syntax

The syntax to rename a table in SQL Server is:
sp_rename 'old_table_name', 'new_table_name';

Example:-

This SQL Server example will use sp_rename to rename the Students table to Student 1

```
Sp_rename 'student','student 1'
```

Check Constraint

SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

The following constraints are commonly used in SQL:

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Uniquely identifies a row/record in another table
- **CHECK** - Ensures that all values in a column satisfies a specific condition

SQL NOT NULL Constraint

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

The following SQL ensures that the "ID", "Last_Name", and "First_Name" columns will NOT accept NULL values:

Example

```
CREATE TABLE Student (
    ID int NOT NULL,
    Last_Name varchar (255) NOT NULL,
    First_Name varchar (255) NOT NULL,
    Age int
);
```

SQL UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

SQL UNIQUE Constraint on CREATE TABLE

The following SQL creates a UNIQUE constraint on the "ID" column when the "student" table is created:

Example

```
CREATE TABLE Student (
    ID int NOT NULL UNIQUE,
    Last_Name varchar (255) NOT NULL,
    First_Name varchar (255),
    Age int
);
```

OR

```
CREATE TABLE student (
    ID int NOT NULL,
    Last_Name varchar (255) NOT NULL,
    First_Name varchar (255),
    Age int,
    UNIQUE (ID)
);
```

UNIQUE constraint on multiple columns

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE Student (
    ID int NOT NULL,
    Last_Name varchar(255) NOT NULL,
    First_Name varchar(255),
    Age int,
    CONSTRAINT Uc_PersonID UNIQUE (ID, Last_Name)
);
```

SQL CHECK Constraint on CREATE TABLE

To create a UNIQUE constraint on the "ID" column when the table is already created, use the following SQL:

```
ALTER TABLE Student ADD UNIQUE (ID);
```

SQL PRIMARY KEY Constraint

- The PRIMARY KEY constraint uniquely identifies each record in a database table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values.
- A table can have only one primary key, which may consist of single or multiple fields.

SQL PRIMARY KEY on CREATE TABLE

The following SQL creates a PRIMARY KEY on the "ID" column when the "Student" table is created:

```
CREATE TABLE student (
    ID int NOT NULL PRIMARY KEY,
    Last_Name varchar (255) NOT NULL,
    First_Name varchar (255),
    Age int
);
```

To allow naming of a PRIMARY KEY constraint, and for defining a **PRIMARY KEY constraint on multiple columns**, use the given below Syntax.

SQL syntax:

```
CREATE TABLE Student (
    ID int NOT NULL,
    Last_Name varchar (255) NOT NULL,
    First_Name varchar (255),
    Age int,
    PRIMARY KEY (ID, Last_Name)
);
```

SQL PRIMARY KEY on ALTER TABLE

To create a PRIMARY KEY constraint on the "ID" column when the table is already created, use the following SQL:

Syntax

```
ALTER TABLE Student ADD PRIMARY KEY (ID);
```

PRIMARY KEY constraint on multiple columns on Alter Table

To allow naming of a PRIMARY KEY constraint, and for defining a PRIMARY KEY constraint on multiple columns, use the following SQL syntax

Syntax:

```
ALTER TABLE Student
ADD CONSTRAINT Pk PRIMARY KEY (ID, Last_Name);
```

Note: If you use the ALTER TABLE statement to add a primary key, the primary key column(s) must already have been declared to not contain NULL values (when the table was first created).

DROP a PRIMARY KEY Constraint

To drop a PRIMARY KEY constraint, use the following SQL:

```
ALTER TABLE Student DROP PRIMARY KEY;

                                OR

ALTER TABLE Persons DROP CONSTRAINT PK_Person;
```

SQL FOREIGN KEY Constraint

A FOREIGN KEY is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

Look at the following two tables:

"Student" table:

	sid	fname	lname	city	age	cgpa	degree	semestemo	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

The enrollment Table

	sid	cid
1	3	1001
2	2	1004
3	4	1004
4	7	1004
5	8	1004
6	4	1005
7	5	1007
8	4	1008
9	8	1008
10	10	1009

Notice that the "sid" column in the "student" table points to the "sid" column in the "enrollment" table.

The "sid" column in the "student" table is the PRIMARY KEY in the

The "sid" column in the "enrollment" table is a FOREIGN KEY

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

SQL FOREIGN KEY on CREATE TABLE

The following SQL creates a FOREIGN KEY on the "sid" column when the "enrollment" table is created:

Example:-

```
CREATE TABLE enrollment (

      Sid int foreign key references student(sid),

      cid int foreign key references course (cid)

);
```

FOREIGN KEY constraint on multiple columns

To allow naming of a FOREIGN KEY constraint, and for defining a FOREIGN KEY constraint on multiple columns, use the following SQL syntax:

Example:-

```
CREATE TABLE enrollment (

     Sid int foreign key references student(sid),

     cid int foreign key references course (cid)

);
```

SQL FOREIGN KEY on ALTER TABLE

To create a FOREIGN KEY constraint on the "sid" column when the "enrollment" table is already created, use the following SQL:

Example:

```
ALTER TABLE enrollment
ADD FOREIGN KEY (sid) REFERENCES student(sid);
```

-

FOREIGN KEY constraint on multiple columns

To allow naming of a FOREIGN KEY constraint, and for defining a FOREIGN KEY constraint on multiple columns, use the following SQL syntax:

Example:-

```
ALTER TABLE enrollment
ADD CONSTRAINT FK FOREIGN KEY (sid) REFERENCES student(sid);
```

DROP a FOREIGN KEY Constraint

To drop a FOREIGN KEY constraint, use the following SQL:

Example Drop FOREIGN KEY

```
ALTER TABLE Orders
DROP FOREIGN KEY FK_PersonOrder;
```

Example Drop FOREIGN KEY constraint

```
ALTER TABLE Orders
DROP CONSTRAINT FK_PersonOrder;
```

SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

SQL CHECK on CREATE TABLE

The following SQL creates a CHECK constraint on the "Age" column when the "student" table is created. The CHECK constraint ensures that you can not have any person below 18 years:

Sql Check Example

```
CREATE TABLE student (
    ID int NOT NULL,
    Last_Name varchar(255) NOT NULL,
    First_Name varchar(255),
    Age int,
    CHECK (Age>=18)
);
OR
CREATE TABLE Student (
    ID int NOT NULL,
    Last_Name varchar(255) NOT NULL,
    First_Name varchar(255),
    Age int CHECK (Age>=18)
);
```

CHECK constraint on multiple columns

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

SQL syntax:

```
CREATE TABLE Student (
    ID int NOT NULL,
    Last_Name varchar(255) NOT NULL,
    First_Name varchar(255),
    Age int,
    City varchar(255),
    CONSTRAINT CHK CHECK (Age>=18 AND City='rwp')
);
```

SQL CHECK on ALTER TABLE

To create a CHECK constraint on the "Age" column when the table is already created, use the following SQL:

Example:-

```
ALTER TABLE Student
ADD CHECK (Age>=18);
```

Alter CHECK constraint on multiple columns

To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

SQL syntax:

```
ALTER TABLE Student
ADD CONSTRAINT CHK_Age CHECK (Age>=18 AND City='Rwp');
```

DROP a CHECK Constraint

To drop a CHECK constraint, use the following SQL:

Example DROP a CHECK Constraint

```
ALTER TABLE Student
DROP CONSTRAINT CHK_Age;

                              OR

ALTER TABLE Student
DROP CHECK CHK_Age;
```

Task 11

1: Write a Sql Statement to create new database and write Sql statement to use new database.

2: Write a Sql Statement to create table Student

3 Write a Sql Statement to Add column in Student table.

4 Write a Sql Statement to add constraint primary key in Student table.

5 Write a Sql Statement to Drop a check Constraint.

Exercise 11

1 Write a SQL statement to create a simple table teacher including columns tid, name and city.

2 Write a Sql Statement to add column in Teacher table and also write statement to drop column from Teacher table.

3 Write a Sql Statement to add constraint in Teacher table and also write statement to drop constraint from Teacher table.

4 Write a Sql Statement to rename any column in Teacher table and write a Sql statement to rename Teacher table to Professor.

5 Write a Sql Statement for you forget to create primary key in Teacher table. Now you have to create composite key (tid,name). tid and name both are null.

6 Write a SQL statement to create a table named course including columns cid, crs_name and cr_hrs and make sure that the cid column will be a key field which will not contain any duplicate data at the time of insertion.

7 Write a SQL statement to create a table named course including columns cid, crs_name and cr_hrs and make sure that no names except DBS, OOP, MVC, WDD will be entered in the table.

8 Write a SQL statement to create a table named student including columns sid, Name, Age, Cgpa and check whether the cgpa exceeding the upper limit 4.

9 Write a SQL statement to create a table named Student including columns sid, name, age, CGPA, and make sure that, the default value for name is blank and age is 18 and cgpa is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns.

10 Write a SQL statement to create a table Student including columns sid, name, age and make sure that the column Sid will be unique and store an auto incremented value.

Lab #12

Insert, Update, Delete, Drop

SQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

The INSERT INTO Statement

The INSERT INTO statement is used to insert a new row in a table.

SQL INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two forms.

The first form doesn't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
```

The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

SQL INSERT INTO Example

We have the following "Student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to insert a new row in the "Student" table.

We use the following SQL statement:

```
INSERT INTO Student
VALUES (11,'Muhammad', 'Abbas', 'ISB', '19',3.4,'MCS',5,'B')
```

The "Student" table will now look like this:

	sid	First Name	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A
11	11	Muhammad	Abbas	ISB	19	3.4	MCS	5	B

Insert Data Only in Specified Columns

It is also possible to only add data in specific columns.

The following SQL statement will add a new row, but only add data in the "sid", "fname" and the "city" columns:

Example

```
INSERT INTO student (sid, fname, city)
VALUES (12, 'Ajmal', 'ISB')
```

The "Student" table will now look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A
11	11	Muhammad	Abbas	ISB	19	3.4	MCS	5	B
12	12	Ajmal	NULL	ISB	NULL	NULL	NULL	NULL	NULL

SQL UPDATE Statement

The UPDATE statement is used to update records in a table.

The UPDATE Statement

The UPDATE statement is used to update existing records in a table.

SQL UPDATE Syntax

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

Note: Notice the WHERE clause in the UPDATE syntax. The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

SQL UPDATE Example

The "student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to update the person "ali" in the "student" table.

We use the following SQL statement:

```
UPDATE student
SET City='Islamabad'
WHERE FName='Ali'
```

The "Student" table will now look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Islamabad	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

SQL DELETE Statement

The DELETE statement is used to delete records in a table.

The DELETE Statement

The DELETE statement is used to delete rows in a table.

SQL DELETE Syntax

```
DELETE FROM table_name
WHERE some_column=some_value
```

Note: Notice the WHERE clause in the DELETE syntax. The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

SQL DELETE Example

The "Student" table:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Now we want to delete the person "Ajmal, Nazir" in the "Persons" table.

We use the following SQL statement:

```
DELETE FROM Student
WHERE LName='ahmad'
```

The "Student" table will now look like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
2	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
3	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
4	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
5	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
6	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
7	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Delete All Rows

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

Syntax

```
DELETE FROM table_name
or
DELETE * FROM table_name
```

Note: Be very careful when deleting records. You cannot undo this statement!

The DROP DATABASE Statement

The DROP DATABASE statement is used to drop an existing SQL database.

Syntax

```
DROP DATABASE databasename;
```

Note: Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!

Example

The following SQL statement drops the existing database "myDBS":

```
DROP DATABASE myDBS;
```

Tip: Make sure you have admin privilege before dropping any database. Once a database is dropped, you can check it in the list of databases with the following SQL command: SHOW DATABASES;

The SQL DROP TABLE Statement

The DROP TABLE statement is used to drop an existing table in a database.

Syntax

```
DROP TABLE table_name;
```

Note: Be careful before dropping a table. Deleting a table will result in loss of complete information stored in the table!

SQL DROP TABLE Example

The following SQL statement drops the existing table "Students":

```
DROP TABLE Students;
```

SQL TRUNCATE TABLE

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

Syntax

```
TRUNCATE TABLE table_name;
```

Example

```
TRUNCATE TABLE Student;
```

Task 12

1 Write a Sql statement to insert data in Student table.

2 Write a Sql Statement to insert data in Course table.

4 Write a Sql Statement to update data from Student table where lname is equal to 'Ahmad'.

5 Write a Sql statement to delete a record from Student where cgpa is equal to 3.7.

Exercise 12

1 Write a Sql statement to insert data in Student objects.

2 Write a Sql statement to insert data in Teacher objects.

3 Write a Sql statement to update data from Student where cgpa>3.5

4 Write a Sql statement to delete data from Customer Where name equal to "Ali".

5 Write a Sql statement to delete all those students who are living in Rwp.

6 Write a Sql statement to update lname of Student whose fname start with ‘a’.

7 Write a Sql statement to update Student Age to 21 whose degree is MCS.

8 Write a Sql statement to delete Student whose age is greater than 21.

9 Write a Sql Statement to delete only data from Student not the Student table.

10 Write a Sql Statement to delete all those Students whose age not in range of 20 and 22.

Lab #13

Sql Views and Sql Subqueries

SQL Views

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note: A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

SQL CREATE VIEW Examples

If you have the Northwind database you can see that it has several views installed by default.

The view "Salesmans_View" lists all active products (Salesman's that have commission 0.12) from the "Salesman's" table. The view is created with the following SQL:

```
CREATE VIEW [Salesmans_View] AS
SELECT salesman_id, Name
FROM Salesman's
WHERE commission = 0.12;
```

Then, we can query the view as follows:

```
SELECT * FROM [Salesmans_View];
```

Another view in the Northwind sample database selects every product in the "Products" table with a unit price higher than the average unit price:

```
CREATE VIEW [Products Above Average Price] AS
SELECT ProductName, UnitPrice
FROM Products
WHERE UnitPrice > (SELECT AVG (UnitPrice) FROM Products);
```

We can query the view above as follows:

```
SELECT * FROM [Products Above Average Price];
```

Another view in the Northwind database calculates the total sale for each category in 1997. Note that this view selects its data from another view called "Product Sales for 1997":

```
CREATE VIEW [Category Sales For 1997] AS
SELECT DISTINCT CategoryName, Sum (ProductSales) AS CategorySales
FROM [Product Sales for 1997]
GROUP BY CategoryName;
```

We can query the view above as follows:

```
SELECT * FROM [Category Sales For 1997];
```

We can also add a condition to the query. Let's see the total sale only for the category "Beverages":

```
SELECT * FROM [Category Sales For 1997]
WHERE CategoryName = 'Beverages';
```

SQL Updating a View

You can update a view by using the following syntax:

SQL CREATE OR REPLACE VIEW Syntax

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2,…
FROM table_name
WHERE condition;
```

Example

Now we want to add the "Category" column to the "Current Product List" view. We will update the view with the following SQL:

```
CREATE OR REPLACE VIEW [Current Product List] AS
SELECT ProductID, ProductName, Category
FROM Products
WHERE Discontinued = No;
```

SQL Dropping a View

You can delete a view with the DROP VIEW command.

SQL DROP VIEW Syntax

```
DROP VIEW view_name;
```

SQL Subqueries

- A **subquery** is a SQL statement that has another SQL query embedded in the **Where** or the **having** clause
- A subquery is a SQL query within a query.
- Subqueries are nested queries that provide data to the enclosing query.
- Subqueries can return individual values or a list of records
- Subqueries must be enclosed with parenthesis

Syntax

The syntax for a subquery when the embedded SQL statement is part of the **WHERE** condition is as follows:

```
SELECT "column_name1"
FROM "table_name1"
WHERE "column_name2" [Comparison Operator]
(SELECT "column_name3"
FROM "table_name2"
WHERE "condition");
```

[Comparison Operator] could be equality operators such as =, >, <, >=, <=. It can also be a text operator such as "LIKE". The portion in **red** is considered as the "inner query," while the portion in **Green** is considered as the "outer query."

Examples

We use the following tables for our examples.

Student

	sid	fname	lname	city	age	cgpa	degree	semestemo	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Example 1: Simple subquery

To use a subquery to find the data of youngest student

```
Select * from student where age in(select min(age) from
student)
```

Result:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A

This type of subquery is called a **simple subquery**.

Example 2:

Consider 2 tables with student table

Course

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Enrollment

	sid	cid
1	3	1001
2	2	1004
3	4	1004
4	7	1004
5	8	1004
6	4	1005
7	5	1007
8	4	1008
9	8	1008
10	10	1009

Now we want to show those students who have enrolled course with 3 credit hours

```
Select * from student where sid in(select sid from enrollment
where cid in(select cid from course where cr_hours=3))
```

The Result will be like this:

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
2	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
3	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
4	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
5	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
6	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
7	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Task 13

Student

	sid	fname	lname	city	age	cgpa	degree	semesterno	section
1	1	Ali	Ahmad	Rwp	18	3.4	BSIT	3	A
2	2	Asad	Ahmad	Isb	23	1.4	BSIT	3	B
3	3	Jamal	Khan	Multa	19	2.7	BSCS	7	A
4	4	Kaleem	Iqbal	Rwp	20	3.9	BSIT	3	A
5	5	Nazeer	Ahmad	Rwp	19	3.4	BSIT	3	B
6	6	Basit	Ali	Isb	21	3.4	BSIT	3	A
7	7	Hassa	Farooq	Multa	22	3.4	BSIT	4	A
8	8	Kareem	Abid	Karachi	23	3.4	MCS	3	A
9	9	Sulta	Abdullah	Isb	18	3.4	BSIT	3	A
10	10	Nabeel	Sohail	Rwp	19	3.4	MCS	3	A

Course

	cid	title	category	cr_hours
1	1001	Database System	CS	3
2	1002	Programming Fundamentals	CS	4
3	1003	Discrete Structures	Math	3
4	1004	Networks	CS	3
5	1005	Calculus	Math	3
6	1006	Differential Math	Math	3
7	1007	Linear Algebra	Math	3
8	1008	Markeeting	Management	3
9	1009	Intro to Management	Management	3
10	1010	Financial Accounting	Management	3

Enrollment

	sid	cid
1	3	1001
2	2	1004
3	4	1004
4	7	1004
5	8	1004
6	4	1005
7	5	1007
8	4	1008
9	8	1008
10	10	1009

1. Write a Sql Statement to retrieve youngest student.

2. Write a Sql Statement to retrieve name of student who registered CS

3. Write a Sql Statement to Display student who registered courses of category MATH.

4. Write a Sql Statement to Display Name of Student who live in Rawalpindi and studying Management.

5. Write a Sql Statement to display title of all courses registered by ALI.

6. Write a Sql Statement to Display title of courses which are registered by either Ali or Basit

7. Write a query to create a view for those salesmen belongs to the city New York.

8. Write a query to create a view for all salesmen with columns salesman_id, name, and city.

9. Write a query to find the salesmen of the city New York who achieved the commission more than 13%.

Lab # 14

SQL Joins

SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.

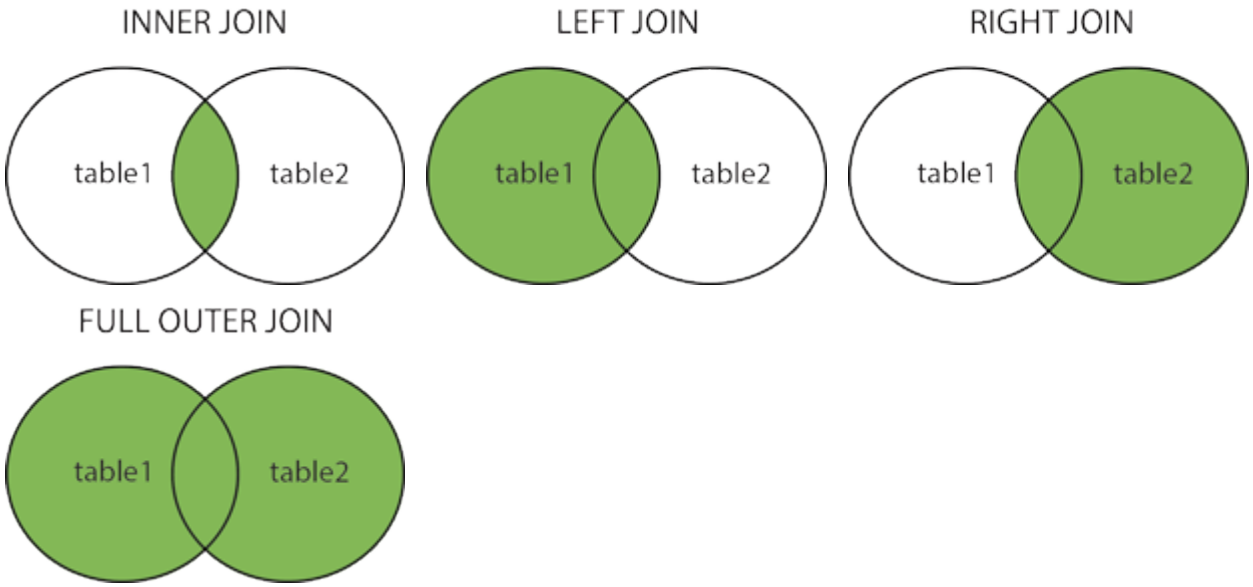
Tables in a database are often related to each other with keys.

A primary key is a column (or a combination of columns) with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Return all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Return all records when there is a match in either left or right table



SQL INNER JOIN Keyword

The INNER JOIN keyword return rows when there is at least one match in both tables.

SQL INNER JOIN Syntax

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

PS: INNER JOIN is the same as JOIN.

SQL INNER JOIN Example

The "Person" table:

	p_id	F_Name	L_Name	Address	City
1	1	Abdul	Rehman	PD/276 Nazim abad pindora Rawalpindi	Rawalpindi
2	2	Sohail	Ashiq	sadiqabad rawalpindi	Rawalpindi
3	3	Muhammad	Ramzan	fasilline 82/A block satellite town rawalpindi	Rawalpindi
4	4	Ajmal	khan	Basti deen purMuzaffargarh, T/D Muzaffargarh, P/...	Muzaffargarh
5	5	Saad	Raiz	D.G khan	D.G Khan
6	6	Muhammad	Raza	basti toly wala din pur Muzaffargarh	Muzaffargarh
7	7	Mahar	Awais	Hakeem colony	Rajan pur
8	8	Muhammad	Abbas	Muzaffargarh 2	Muzaffargarh

The "Orders" table:

	O_Id	OrderNo	P_Id
1	1	77895	3
2	2	44678	3
3	3	22456	2
4	4	24562	1

Example

Now we want to list all the persons with any orders.

We use the following SELECT statement:

```
SELECT Person.F_Name, Person.L_Name, Orders.OrderNo
FROM Person
INNER JOIN Orders
ON Person.p_id=Orders.p_id
ORDER BY Person.F_Name
```

The result-set will look like this:

	F_Name	L_Name	OrderNo
1	Abdul	Rehman	24562
2	Muhammad	Ramzan	77895
3	Muhammad	Ramzan	44678
4	Sohail	Ashiq	22456

The INNER JOIN keyword return rows when there is at least one match in both tables. If there are rows in "Persons" that do not have matches in "Orders", those rows will NOT be listed.

SQL LEFT JOIN Keyword

The LEFT JOIN keyword returns all rows from the left table (table_name1), even if there are no matches in the right table (table_name2).

SQL LEFT JOIN Syntax

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

PS: In some databases LEFT JOIN is called LEFT OUTER JOIN.

SQL LEFT JOIN Example

The "Person" table:

	p_id	F_Name	L_Name	Address	City
1	1	Abdul	Rehman	PD/276 Nazim abad pindora Rawalpindi	Rawalpindi
2	2	Sohail	Ashiq	sadiqabad rawalpindi	Rawalpindi
3	3	Muhammad	Ramzan	fasilline 82/A block satellite town rawalpindi	Rawalpindi
4	4	Ajmal	khan	Basti deen purMuzaffargarh, T/D Muzaffargarh, P/...	Muzaffargarh
5	5	Saad	Raiz	D.G khan	D.G Khan
6	6	Muhammad	Raza	basti toly wala din pur Muzaffargarh	Muzaffargarh
7	7	Mahar	Awais	Hakeem colony	Rajan pur
8	8	Muhammad	Abbas	Muzaffargarh 2	Muzaffargarh

The "Orders" table:

	O_Id	OrderNo	P_Id
1	1	77895	3
2	2	44678	3
3	3	22456	2
4	4	24562	1

Now we want to list all the persons and their orders - if any, from the tables above.

We use the following SELECT statement:

```
SELECT Person.F_Name, Person.L_Name, Orders.OrderNo
FROM Person
LEFT JOIN Orders
ON Person.p_id=Orders.p_id
ORDER BY Person.F_Name
```

The result-set will look like this:

	F_Name	L_Name	OrderNo
1	Abdul	Rehman	24562
2	Ajmal	khan	NULL
3	Mahar	Awais	NULL
4	Muhammad	Abbas	NULL
5	Muhammad	Ramzan	77895
6	Muhammad	Ramzan	44678
7	Muhammad	Raza	NULL
8	Saad	Raiz	NULL
9	Sohail	Ashiq	22456

The LEFT JOIN keyword returns all the rows from the left table (Persons), even if there are no matches in the right table (Orders).

SQL RIGHT JOIN Keyword

The RIGHT JOIN keyword Return all rows from the right table (table_name2), even if there are no matches in the left table (table_name1).

SQL RIGHT JOIN Syntax

```
SELECT column_name(s)
FROM table_name1
RIGHT JOIN table_name2

ON table_name1.column_name=table_name2.column_name
```

PS: In some databases RIGHT JOIN is called RIGHT OUTER JOIN.

SQL RIGHT JOIN Example

The "Persons" table:

	p_id	F_Name	L_Name	Address	City
1	1	Abdul	Rehman	PD/276 Nazim abad pindora Rawalpindi	Rawalpindi
2	2	Sohail	Ashiq	sadiqabad rawalpindi	Rawalpindi
3	3	Muhammad	Ramzan	fasilline 82/A block satellite town rawalpindi	Rawalpindi
4	4	Ajmal	khan	Basti deen purMuzaffargarh, T/D Muzaffargarh, P/...	Muzaffargarh
5	5	Saad	Raiz	D.G khan	D.G Khan
6	6	Muhammad	Raza	basti toly wala din pur Muzaffargarh	Muzaffargarh
7	7	Mahar	Awais	Hakeem colony	Rajan pur
8	8	Muhammad	Abbas	Muzaffargarh 2	Muzaffargarh

The "Orders" table:

	O_Id	OrderNo	P_Id
1	1	77895	3
2	2	44678	3
3	3	22456	2
4	4	24562	1

Now we want to list all the orders with containing persons - if any, from the tables above.

We use the following SELECT statement:

```
SELECT Person.F_Name, Person.L_Name, Orders.OrderNo
FROM Person
RIGHT JOIN Orders
ON Person.p_id=Orders.p_id
ORDER BY Person.F_Name
```

The result-set will look like this:

	F_Name	L_Name	OrderNo
1	Abdul	Rehman	24562
2	Muhammad	Ramzan	77895
3	Muhammad	Ramzan	44678
4	Sohail	Ashiq	22456

The RIGHT JOIN keyword returns all the rows from the right table (Orders), even if there are no matches in the left table (Persons).

SQL FULL JOIN Keyword

The FULL JOIN keyword return rows when there is a match in one of the tables.

SQL FULL JOIN Syntax

```
SELECT column_name(s)
FROM table_name1
FULL JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

SQL FULL JOIN Example

The "Person" table:

	p_id	F_Name	L_Name	Address	City
1	1	Abdul	Rehman	PD/276 Nazim abad pindora Rawalpindi	Rawalpindi
2	2	Sohail	Ashiq	sadiqabad rawalpindi	Rawalpindi
3	3	Muhammad	Ramzan	fasilline 82/A block satellite town rawalpindi	Rawalpindi
4	4	Ajmal	khan	Basti deen purMuzaffargarh, T/D Muzaffargarh, P/...	Muzaffargarh
5	5	Saad	Raiz	D.G khan	D.G Khan
6	6	Muhammad	Raza	basti toly wala din pur Muzaffargarh	Muzaffargarh
7	7	Mahar	Awais	Hakeem colony	Rajan pur
8	8	Muhammad	Abbas	Muzaffargarh 2	Muzaffargarh

The "Orders" table:

	O_Id	OrderNo	P_Id
1	1	77895	3
2	2	44678	3
3	3	22456	2
4	4	24562	1

Now we want to list all the persons and their orders, and all the orders with their persons.

We use the following SELECT statement:

```
SELECT Person.F_Name, Person.L_Name, Orders.OrderNo
FROM Person
FULL JOIN Orders
ON Person.p_id=Orders.p_id
ORDER BY Person.F_Name
```

The result-set will look like this:

	F_Name	L_Name	OrderNo
1	Abdul	Rehman	24562
2	Ajmal	khan	NULL
3	Mahar	Awais	NULL
4	Muhammad	Abbas	NULL
5	Muhammad	Ramzan	77895
6	Muhammad	Ramzan	44678
7	Muhammad	Raza	NULL
8	Saad	Raiz	NULL
9	Sohail	Ashiq	22456

The FULL JOIN keyword returns all the rows from the left table (Persons), and all the rows from the right table (Orders). If there are rows in "Persons" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Persons", those rows will be listed as well.

Task 14

Student

Sid	SNAME	Sage	Cgpa
1	Ali	23	3.4
2	Aslam	17	2.4
3	Rahi	21	2.7
4	Munir	29	1.4
5	Akram	28	3.7
6	Eimal	22	2.9
7	Elahi	24	1.3
8	Jamal	16	3.2
9	Adnan	20	3.8
10	Aimal	28	1.6
11	Zaheer	18	3.7
12	Zameer	24	3.9

Reg

Sid	cid
1	1
9	2
1	3
8	7
2	4
3	5
6	6
3	4
5	4
4	1
7	3
4	2
10	2
10	4

Asg

Tid	Cid
1	7
2	1
3	2
4	5
5	3
6	4
7	6
7	7
7	8
8	9
8	10

Course

cid	title	Category
1	DBS	CS
2	PF	CS
3	CCN	CS
4	DM	Math
5	DL	Math
6	FA	Management
7	ITM	Management
8	OOP	CS
9	DDS	CS
10	CALI	Math

Teacher

Tid	TName	Tage	Sal
1	Ali	43	30000
2	Aslam	27	50000
3	Ikram	21	25000
4	Amir	49	40000
5	Munir	28	50000
6	Jamal	22	25000
7	Azhar	24	15000
8	Muneeb	66	18000
9	Elahi	40	37000

1 Write a query that will display all those students who have registered a course which is taught by a teacher having name Aslam.

2 write a query that will display tname and course title where category is CS.

3 write a query that will display name of teacher who are teaching a course which is registered by a student having Cgpa greater than 3.0.

4 Write a query that will display all those Courses which is assign to Students having name Adnan.

5 Write a query that will display all those course which are registered by student having name Ali.

Exercise 14

We have data of Doctor and patients stored in the following tables.

Doctor Table:

Results | Messages

	Did	Name	City	Age	gender	Specialization	salary
1	1	Noman Ejaz	Rawalpindi	40	Male	cardiologist	30000
2	2	Farhan Syed	Islamabad	50	Male	Urology	45000
3	3	Uzma Shah	Rawalpindi	35	Female	Audilogist	38000
4	4	Fatima Gul	Islamabad	42	Female	Ophthalmologist	34000
5	5	Zoe Viccaji	Rawalpindi	46	Female	Dentist	48000
6	6	Shahzad Abbasi	Islamabad	37	Male	Orthodontists	39500
7	7	Shahrukh Amir	Islamabad	32	Male	Otolaryngology	36500
8	8	Junaid Alam	Karachi	34	Male	Neurologist	50000
9	9	Ayesha Khan	Lahore	36	Female	Cardiologist	35000

Patient Table:

Results Messages

	P_ID	name	City	Age	gender	Did
1	1	Nouman Malik	Rawalpindi	32	Male	1
2	2	Osama Zubair	Islamabad	65	Male	2
3	3	Mohsin	Islamabad	35	Male	3
4	4	Hina Akhtar	Islamabad	53	Fmale	4
5	5	Farhana	Rawalpindi	25	female	4

Ward Table:

Results Messages

	Ward_ID	Type	Bed	nid
1	1	ICU	2	1
2	2	Children	4	2
3	3	Adults	4	3
4	4	Adults	4	4
5	5	Heart Ward	4	5

Nurse Table:

Results Messages

	Nid	Nurse Name	Address	Salary
1	1	Debbie Johnson	California, US	500
2	2	Jessica Aries	Virginia, US	850
3	3	Marry Albert	Montana, US	600
4	4	Farhana Masood	Islamabad, Pakistan	750
5	5	Maria Ashiq	Lahore, Pakistan	800

1 Write a query to show records of those patients who are being treated by Female Doctors.

2 Write a query to show City and ages of those patients who are treated by doctors who age is between 30 to 40 years old.

3 Write a query to show nurse ID and address of those nurses whose are managing ward where ward type is adult.

4 Write a query to all records of wards which are managed by nurses whose salary is between 600 to 800.

5 Write a query to show name, age, gender of those doctors who are assigned to projects whose duration of project is maximum.

6 Write a query to show patient ID, name of the patient(s) who is being treated by the doctor whose salary is the lowest.

7 Write a query to show name, duration of those projects who are assigned to doctors whose age lies between 35 to 45.

8 Write a query to show name of those doctors who are treating patients who reside in Islamabad.

9 Write a query to show name and salary of those nurses whose nurse are managing ward with 4 beds.

10 Write a query to show name of those projects who are assigned to doctors whose gender is female.

LAB #15

Relational Algebraic Operators

SQL UNION Operator

The SQL UNION operator combines two or more SELECT statements.

The UNION operator is used to combine the result-set of two or more SELECT statements.

Notice that each SELECT statement within the UNION must have the same number of columns. The columns must also have similar data types. Also, the columns in each SELECT statement must be in the same order.

SQL UNION Syntax

```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2
```

Note: The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL.

SQL UNION ALL Syntax

```
SELECT column_name(s) FROM table_name1
UNION ALL
SELECT column_name(s) FROM table_name2
```

PS: The column names in the result-set of a UNION are always equal to the column names in the first SELECT statement in the UNION.

SQL UNION Example

Look at the following tables:

"Employees_Rwp":

E_ID	E_Name
01	Ashiq Hussain
02	Ahmed Raza
03	Akhter Rehmani
04	Muhammad Arshad

"Employees_Isb":

E_ID	E_Name
01	Tanveer Abbas
02	Awais Qadir
03	Akhter Rehmani
04	Azeem Javed

Now we want to list **all the different** employees in Rwp and Isb.

We use the following SELECT statement:

```
SELECT E_Name FROM Employees_Rwp
UNION
SELECT E_Name FROM Employees_Isb
```

The result-set will look like this:

E_Name
Ashiq Hussain
Ahmed Raza
Akhter Rehmani
Muhammad Arshad
Tanveer Abbas
Awais Qadir
Azeem Javed

Note: This command cannot be used to list all employees in Rwp and Isb. In the example above we have two employees with equal names, and only one of them will be listed. The UNION command selects only distinct values.

SQL UNION ALL Example

Now we want to list **all** employees in Rwp and Isb:

```
SELECT E_Name FROM Employees_Rwp
```

```
UNION ALL
SELECT E_Name FROM Employees_Isb
```

Result

E_Name
Ashiq Hussain
Ahmed Raza
Akhter Rehmani
Muhammad Arshad
Tanveer Abbas
Awais Qadir
Akhter Rehmani
Azeem Javed

Exercise 15

Salesman Table

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

Customer Table

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		5005
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	5003
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moscow	200	5007

Order Table

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005

70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

1: Write a query to display all salesmen and customer located in London.

2: Write a query to display distinct salesman and their cities.

3: Write a query to display all the salesmen and customer involved in this inventory management system.

4: Write a query to make a report of which salesman produce the largest and smallest orders on each date.

5: Write a query to make a report of which salesman produce the largest and smallest orders on each date and arranged the orders number in smallest to the largest number.

6: Write a query to list all the salesmen, and indicate those who do not have customers in their cities.

7: Write a query to that appends strings to the selected fields, indicating whether or not a specified salesman was matched to a customer in his city.

8: Create a union of two queries that shows the names, cities, and ratings of all customers. Those with a rating of 200 or greater will also have the words "High Rating", while the others will have the words "Low Rating".

9: Write a command that produces the name and number of each salesman and each customer with more than one current order. Put the results in alphabetical order.

Lab # 16

SQL Cross Join

What is Cross Join in SQL?

The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN. This kind of result is called as Cartesian Product.

If WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN.

An alternative way of achieving the same result is to use column names separated by commas after SELECT and mentioning the table names involved, after a FROM clause.

Syntax:

```
SELECT *
FROM table1
CROSS JOIN table2;
```

Pictorial Presentation of Cross Join syn

Example:

Here is an example of cross join in SQL between two tables.

Sample table: foods

```
+---------+-------------+-----------+------------+
| ITEM_ID | ITEM_NAME    | ITEM_UNIT | COMPANY_ID |
+---------+-------------+-----------+------------+
| 1       | Chex Mix     | Pcs       | 16         |
| 6       | Cheez-It     | Pcs       | 15         |
| 2       | BN Biscuit   | Pcs       | 15         |
| 3       | Mighty Munch | Pcs       | 17         |
| 4       | Pot Rice     | Pcs       | 15         |
| 5       | Jaffa Cakes  | Pcs       | 18         |
| 7       | Salt n Shake | Pcs       |            |
+---------+-------------+-----------+------------+
```

Sample table: company

```
+------------+---------------+--------------+
| COMPANY_ID | COMPANY_NAME  | COMPANY_CITY |
+------------+---------------+--------------+
| 18         | Order All     | Boston       |
| 15         | Jack Hill Ltd | London       |
| 16         | Akas Foods    | Delhi        |
| 17         | Foodies.      | London       |
| 19         | sip-n-Bite.   | New York     |
+------------+---------------+--------------+
```

To get item name and item unit columns from foods table and company name, company city columns from company table, after a CROSS JOINING with these mentioned tables, the following SQL statement can be used:

SQL Code:

```
SELECT foods.item_name, foods.item_unit,

company.company_name, company.company_city

FROM foods

CROSS JOIN Company;
```

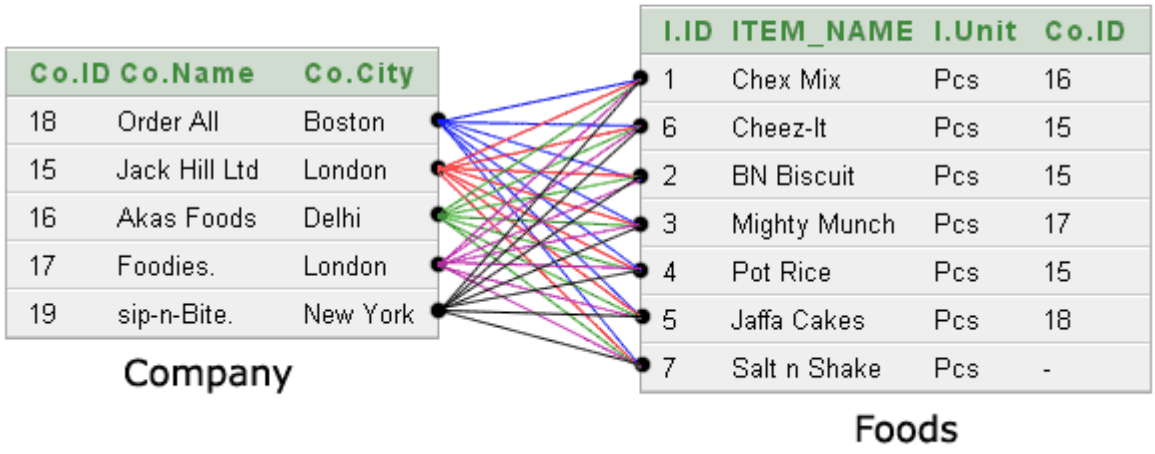
or

SQL Code:

```
SELECT foods.item_name,foods.item_unit,
company.company_name,company.company_city
FROM foods,company;
```

How cross joining happend into two tables

```
SELECT foods.item_name,foods.item_unit,
company.company_name,company.company_city
FROM foods
CROSS JOIN company;
```



Output:

```
ITEM_NAME       ITEM_UNIT  COMPANY_NAME    COMPANY_CITY
--------------- ---------- --------------- ---------------
Chex Mix        Pcs        Order All       Boston
Cheez-It        Pcs        Order All       Boston
BN Biscuit      Pcs        Order All       Boston
Mighty Munch    Pcs        Order All       Boston
Pot Rice        Pcs        Order All       Boston
Jaffa Cakes     Pcs        Order All       Boston
Salt n Shake    Pcs        Order All       Boston
Chex Mix        Pcs        Jack Hill Ltd   London
Cheez-It        Pcs        Jack Hill Ltd   London
BN Biscuit      Pcs        Jack Hill Ltd   London
Mighty Munch    Pcs        Jack Hill Ltd   London
Pot Rice        Pcs        Jack Hill Ltd   London
Jaffa Cakes     Pcs        Jack Hill Ltd   London
Salt n Shake    Pcs        Jack Hill Ltd   London
Chex Mix        Pcs        Akas Foods      Delhi
Cheez-It        Pcs        Akas Foods      Delhi
BN Biscuit      Pcs        Akas Foods      Delhi
Mighty Munch    Pcs        Akas Foods      Delhi
Pot Rice        Pcs        Akas Foods      Delhi
Jaffa Cakes     Pcs        Akas Foods      Delhi
Salt n Shake    Pcs        Akas Foods      Delhi
Chex Mix        Pcs        Foodies.        London
.........
.........
```

More presentaion of the said output:

Exercise 16

Salesman Table

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

Customer Table

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		5005
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001

3009	Geoff Camero	Berlin	100	5003
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moscow	200	5007

Order Table

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

1: Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

2: Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.

3: Write a SQL statement to know which salesman are working for which customer.

4: Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

5: Write a SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in the same city where their customer lives, and gets a commission is above 12%.

6: Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

7: Write a SQL statement to make a join on the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.

8: Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

9: Write a SQL statement to make a list in ascending order for the customer who holds a grade less than 300 and works either through a salesman or by own.

10: Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.