

National University of Computer and Emerging Sciences



Lab Manual 3

“Introduction to SQL Retrieval, Set Operations, Joins”

Database Systems

Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Table of Contents

Objective	3
Task Distribution	3
1. SELECT-FROM-WHERE	4
a) Most Basic Select:	4
b) Retrieving certain Columns from Select	5
c) Retrieving certain Rows from Select- WHERE CLAUSE	5
d) Between condition	6
e) Like operator	6
f) Renaming Resulting Column	7
2. Set operations	7
3. Join Operation	9
a) Inner Join:	9
b) Left/Right/Full Outer Join	9
c) Cross Join	11
d) Joining More than two tables	11
Appendix	12

Objective

The purpose of this lab activity is to familiarize you with the concept of SQL Set Operations and Joins.

Task Distribution

Total Time	150 Minutes
Select From Where	15 Minutes
Set Operations	5 Minutes
Joins	10 Minutes
Miscellaneous Functions	10 Minutes
Exercise	110 Minutes

1.

SELECT-FROM-WHERE

Select from where is equivalent to projection and selection in Relational Algebra, it will give output in form of a table.

The most basic select statement includes `Select` and `from` clause, and it will retrieve all columns and rows from the table.

We will use the following schema and database for the examples. Script to create this schema is given in Lab4ExampleSchema.sql file

Students

StudentID	StudentName	StudentBatch	CGPA
1	Ali	2013	3.3
2	Aysha	2013	4
3	Ahmed	2013	2.2

Instructors

InstructorID	InstructorsName
1	Zafar
2	Sadia
3	Saima

Courses

CourseID	CourseName	CourseCreditHours	InstructorID
1	Computer Programming	3	1
2	Computer Organization	3	2
3	Computer Programmi...	1	NULL

Registration

StudentID	CourseID	GPA
1	1	3
1	3	3
2	2	0

a) Most Basic Select:

```
SELECT *
FROM <table Name>
```

* after select means that all columns will be retrieved

Try this

```
select * from students
```

Results

	StudentID	StudentName	StudentBatch	CGPA
1	1	Ali	2013	3.3
2	2	Aysha	2013	4
3	3	Ahmed	2013	2.2

b) Retrieving certain Columns from Select

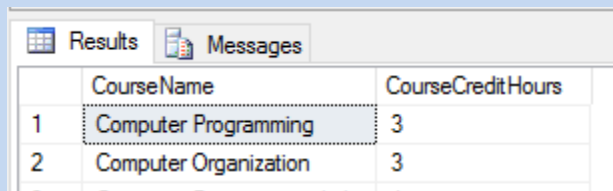
To retrieve only certain columns give a comma separated list of those columns after Select keyword

```
SELECT ColumnX, ColumnY, ColumnZ  
FROM <tableName>
```

Try this

```
Select Course Name, CourseCreditHours  
from courses
```

Results



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'CourseName' and 'CourseCreditHours'. The table contains two rows of data.

	CourseName	CourseCreditHours
1	Computer Programming	3
2	Computer Organization	3

c) Retrieving certain Rows from Select- WHERE CLAUSE

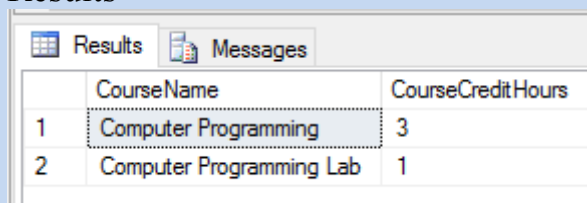
Like Selection in RA, rows are filter in SQL using WHERE clause, rows that fulfill where clause conditions will be projected in result. Where clause can put condition on original columns of tables mentioned on from clause, or derived columns.

```
SELECT *  
FROM <table Name>  
where <conditions>
```

Try this

```
Select CourseName, CourseCreditHours  
from courses  
where CourseName like '%Programming%' and CourseCreditHours >= 1
```

Results



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'CourseName' and 'CourseCreditHours'. The table contains two rows of data.

	CourseName	CourseCreditHours
1	Computer Programming	3
2	Computer Programming Lab	1

d) Between condition

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates. (inclusive boundary values)

```
SELECT *  
FROM student  
WHERE cgpa BETWEEN 3 AND 4;
```

You can also check the range for date data types

```
SELECT * FROM Person  
WHERE DateOfBirth BETWEEN '1996-07-01' AND '1996-07-31';
```

e) Like operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

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

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_ %'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

f) Renaming Resulting Column

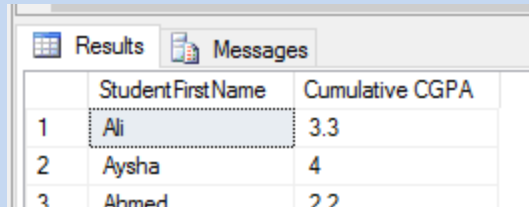
You can rename a column in result by using AS keyword also called Alias. The scope of this renaming is only to that select query, this is useful in joining where more than one table have same column names.

```
SELECT ColumnX as X , ColumnY as Y, ColumnZ  
FROM <tableName> as Table1
```

Try this

```
select StudentName AS StudentFirstName  
, CGPA AS [Cumulative CGPA]  
from students AS StudentsTable
```

Results



	StudentFirst Name	Cumulative CGPA
1	Ali	3.3
2	Aysha	4
3	Ahmed	2.2

1. Set operations

Result of two (or more) select queries can be combined using Set operations such as UNION, INTERSECT, EXCEPT.

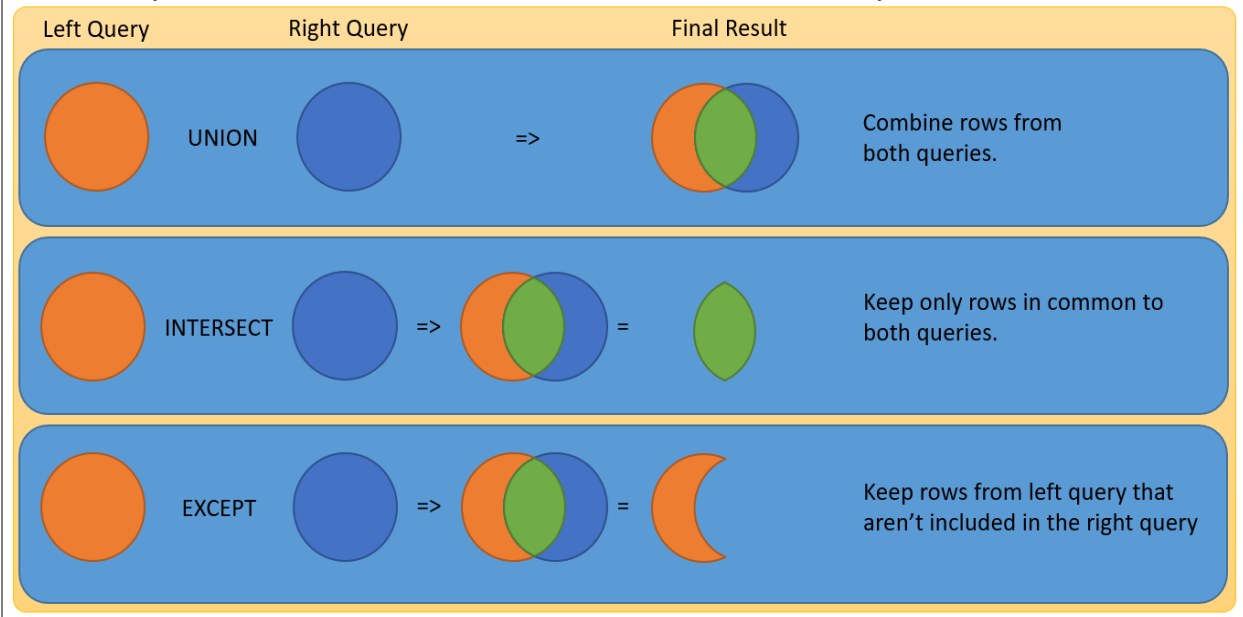
Syntax

```
Select ColumnX, ColumnY  
From Table1
```

Union/Intersect/Except

```
Select ColumnA, ColumnB  
From Table2
```

Visual Explanation of UNION, INTERSECT, and EXCEPT operators



NOTE: The output of first select query should have same number and type of column as of second select query.

Try this- error to look out for in set operations

```
--Fails because Datatype of Corresponding Columns is not same
Select StudentName From Students
Union
Select StudentID From Students
```

Results Messages

Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'Ali' to data type int.

```
--Fails becscue Number of Columns in 1st select query are not same as number of columns in 2nd select query
Select StudentName From Students
Union
Select StudentID, StudentName From Students
```

Messages

Msg 205, Level 16, State 1, Line 2
All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of expressions in their target lists.

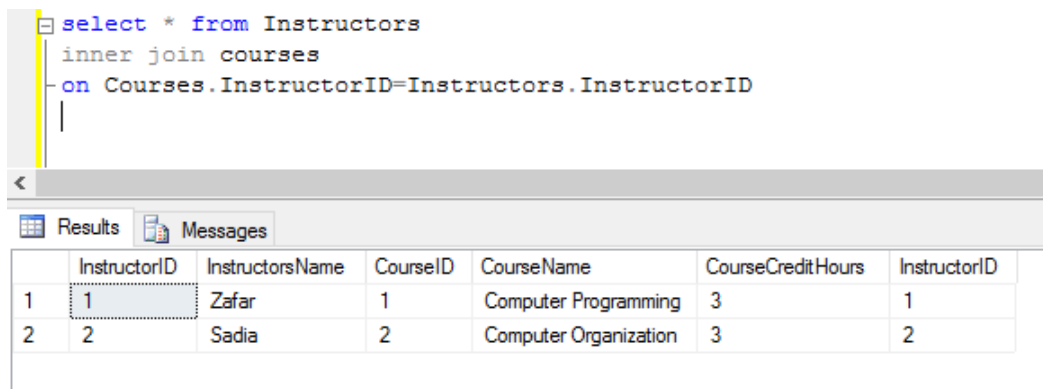
2. Join Operation

We will use the following tables in examples

a) Inner Join:

Returns only those rows that match in both tables.

```
SELECT *  
FROM <table1> inner join <table2>  
ON <Joining Condition>
```



The screenshot shows a SQL query editor with the following query:

```
select * from Instructors  
inner join courses  
on Courses.InstructorID=Instructors.InstructorID
```

Below the query editor is a results window with two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with the following data:

	InstructorID	InstructorsName	CourseID	CourseName	CourseCreditHours	InstructorID
1	1	Zafar	1	Computer Programming	3	1
2	2	Sadia	2	Computer Organization	3	2

g) Left/Right/Full Outer Join

Left Join: Returns all the rows of Left table with corresponding row or null row of right table

Right Join: Returns all the rows of Right table with corresponding row or null row of Left table

Full Join: Union of Left and Right Outer join

```
SELECT * FROM <table1> Left/Right/Full join <table2> ON <Joining  
Condition>
```

Try these

```

select * from Instructors
left join courses
on Courses.InstructorID=Instructors.InstructorID

```

	InstructorID	InstructorsName	CourseID	CourseName	CourseCreditHours	InstructorID
1	1	Zafar	1	Computer Programming	3	1
2	2	Sadia	2	Computer Organization	3	2
3	3	Saima	NULL	NULL	NULL	NULL

```

select * from Instructors
right join courses
on Courses.InstructorID=Instructors.InstructorID

```

InstructorID	InstructorsName	CourseID	CourseName	CourseCreditHours	InstructorID
1	Zafar	1	Computer Programming	3	1
2	Sadia	2	Computer Organization	3	2
NULL	NULL	3	Computer Programming Lab	1	NULL

```

select * from Instructors
full join courses
on Courses.InstructorID=Instructors.InstructorID

```

InstructorID	InstructorsName	CourseID	CourseName	CourseCreditHours	InstructorID
1	Zafar	1	Computer Programming	3	1
2	Sadia	2	Computer Organization	3	2
3	Saima	NULL	NULL	NULL	NULL
NULL	NULL	3	Computer Programming Lab	1	NULL

h) Cross Join

It's a cross product of two tables, no ON condition is required here

SELECT * FROM <table1> cross Join <table2>

Try this

```
select * from Instructors
cross join Courses
```

	InstructorID	InstructorsName	CourseID	CourseName	CourseCreditHours	InstructorID
1	1	Zafar	1	Computer Programming	3	1
2	2	Sadia	1	Computer Programming	3	1
3	3	Saima	1	Computer Programming	3	1
4	1	Zafar	2	Computer Organization	3	2
5	2	Sadia	2	Computer Organization	3	2
6	3	Saima	2	Computer Organization	3	2
7	1	Zafar	3	Computer Programming Lab	1	NULL
8	2	Sadia	3	Computer Programming Lab	1	NULL
9	3	Saima	3	Computer Programming Lab	1	NULL

i) Joining More than two tables

SELECT *

FROM <table1>

Left/Right/Full/Inner join <table2> ON <Joining Condition>

Left/Right/Full/Inner join <table3> ON <Joining Condition>

Left/Right/Full/Inner join <table4> ON <Joining Condition>

Try this

```
select studentName, courseName, instructorsName
from students S
join Registration R on R.studentID=S.studentID
join Courses c on R.courseID=c.courseID
join Instructors i on i.InstructorID=c.InstructorID
```

	studentName	courseName	instructorsName
1	Ali	Computer Programming	Zafar
2	Aysha	Computer Organization	Sadia

Appendix

Some Useful clause

Distinct

```
--% eliminates duplicated.  
Select Distinct Departments from students
```

Like

```
--% for Any string of zero or more characters.  
Select * from students where studentName like '%ed%'  
-- _ for Any single character.  
Select * from students where studentName like 'Ahm_d'  
--[ ] for Any single character within the specified range ([a-f]) or set  
([abcdef]).  
select * From Students where studentName like 'Ahm[ae]d'  
--[^] for Any single character not within the specified range ([a-f]) or set  
([abcdef]).  
select * From Students where studentName like 'Ahm[^a]d'
```

is null/ is not null

```
select * from Course where InstructorID is null  
select * from Course where InstructorID is not null
```

between

```
select * From Students where studentId between 1 and 10  
select * From Orders where orderDate between '2-2-2001' and '2-2-2010'  
select *, year(orderDate) as Year From Orders where year(orderDate) between  
2001 and 2010
```

Some usefull functions

```
isNull(col,value) - replces the null entry with value  
CAST ( expression AS data_type )  
CONVERT ( data_type, expression)
```

DATE FUNCTIONS

```
DATEPART(datepart, date) --returns the datepart of date  
Year(date)-- returns the Year of date  
Month(date) --returns month of date  
Day(date) --returns Day from date  
DATEDIFF ( datepart , startdate , enddate ) --returns the difference in start  
and end date in datepart (eg year,days ,months)
```

STRING FUNCTIONS

```
UPPER(String)  
LOWER(String)  
LEFT(String,7) -- returns left 7 Characters
```

```
RIGHT(String,7)
LEN(String)
LTRIM (String) -- Trim the left end of string
RTRIM(String)
SUBSTRING (String, 8, 7)
CHARINDEX ('demo', String) -will return the starting index of 'demo' in
String
REPLACE (String, 's', '$') REVERSE (String)
```