|  |
| --- |
| National University of Computer and Emerging Sciences |
| Lab Manual 4  “Data Retrieval Select-from-where Join Order by” |
|  |
| Database Systems |
| Spring 2018 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Table of Contents

[1. Objective 2](#_Toc473723517)

[2. Pre-requisites 2](#_Toc473723518)

[3. Task Distribution 2](#_Toc473723519)

[4. SELECT-FROM-WHERE 3](#_Toc473723520)

[Most Basic Select: 3](#_Toc473723521)

[Retrieving certain Columns from Select 4](#_Toc473723522)

[Retrieving certain Rows from Select- WHERE CLAUSE 4](#_Toc473723523)

[Renaming Resulting Column 4](#_Toc473723524)

[5. Order by Clause 5](#_Toc473723525)

[TOP Clause 5](#_Toc473723526)

[6. Join Operation 6](#_Toc473723527)

[Inner Join: 6](#_Toc473723528)

[Left/Right/Full Outer Join 7](#_Toc473723529)

[Cross Join 8](#_Toc473723530)

[Joining More than two tables 8](#_Toc473723531)

[References 9](#_Toc473723532)

# Objective

The purpose of this manual is to get stared with data retrieval queries, starting from Simple Select-From-Where, going towards Join operation, and then covering Order by clause.

# Pre-requisites

* Lab 2 manual, on how to get started with MS-SQL server.
* How Select From Where clause work.
* How Joining and all its type work.
* How Order by clause works.
* Chapter 1, Lesson 1 and Lesson 2 from Microsoft SQL Server 2008 Database Development.

# Task Distribution

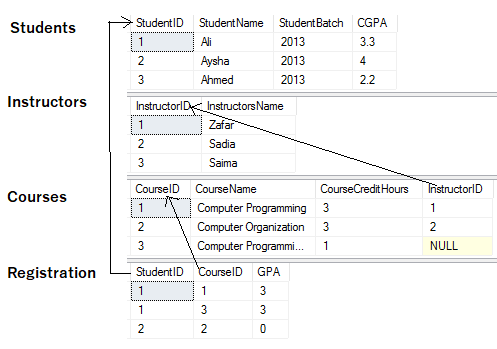
|  |  |
| --- | --- |
| Total Time | 170 Minutes |
| Select from where | 15 min |
| Order by | 15 Minutes |
| Joining | 30 Minutes |
| Exercise | 90 Minutes |
| Evaluation | Last 20 Minutes |

# 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 Lab3ExampleSchema.sql file



## Most Basic Select:

SELECT \*

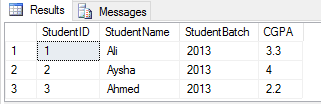
FROM <tableName>

\* after select means that all columns will be retrieved

Try this



Results



## Retrieving certain Columns from Select

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

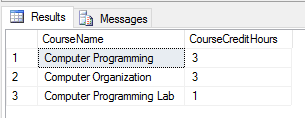
SELECT ColumnX, ColumnY, ColumnZ

FROM <tableName>

Try this



Results



## 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 \*

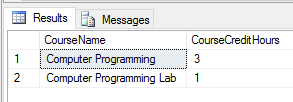
FROM <tableName>

where <conditions>

Try this



Results



## 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.

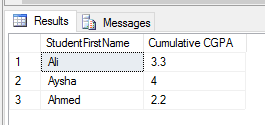
SELECT ColumnX as X , ColumnY as Y, ColumnZ

FROM <tableName> as Table1

Try this



Results



# Order by Clause

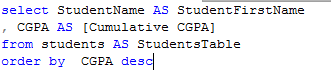
Order by clause is used to arrange the rows in ascending or descending order of one or more columns

SELECT ColumnX as X , ColumnY as Y, ColumnZ

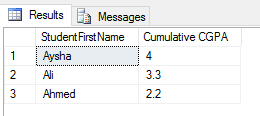
FROM <tableName> as Table1

ORDER BY ColumnX asc/desc, ColumnZ asc/desc

Try this



Results



## TOP Clause

Top n clause will give you first n rows from result instead of all the rows.

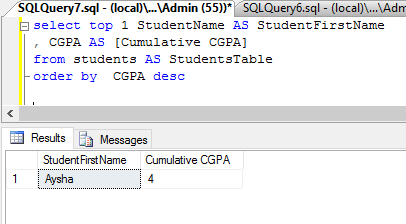
SELECT TOP <n> \*

FROM <tableName>

where <conditions>

Order by <column Name> asc/desc

Try this



# Join Operation

This section gives sytax details of different types of joining, for futher details read Chapter 1 Lesson 2 Microsoft SQL Server 2008 Database Development.

We will use the following tables in examples

## Inner Join:

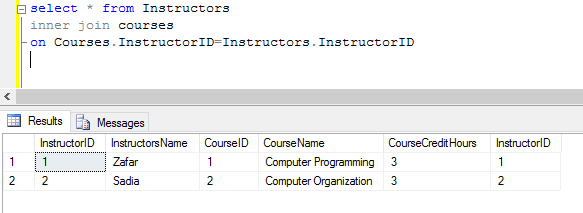
Returns only those rows that match in both tables.

SELECT \*

FROM <table1> inner join <table2>

ON <Joining Condition>

Try this



## Left/Right/Full Outer Join

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

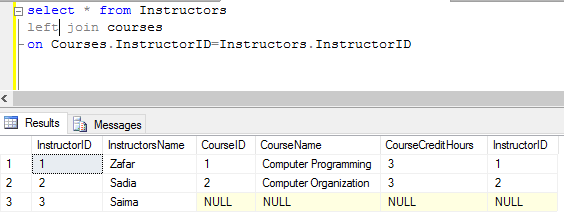
Right Join: Returns all the rows of Right table with corresponding row or null row of Left table

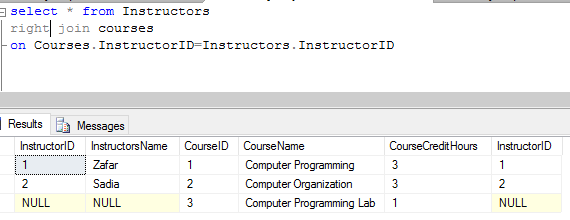
Full Join: Union of Left and Right Outer join

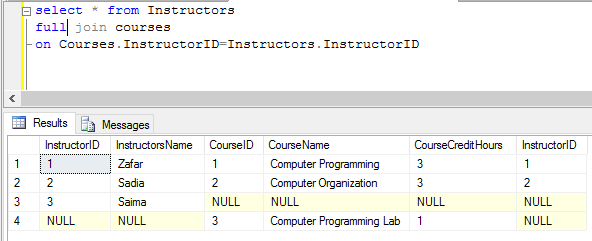
SELECT \* FROM <table1> Left/Right/Full join <table2>

ON <Joining Condition>

Try these







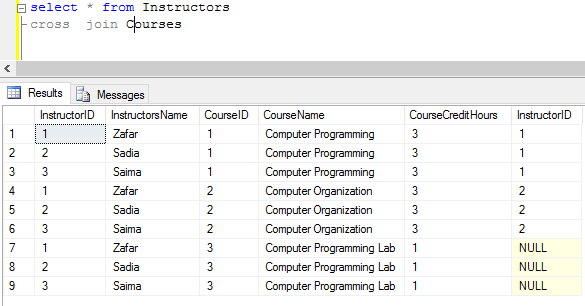
## Cross Join

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

SELECT \*

FROM <table1> cross Join <table2>

Try this



## Joining More than two tables

SELECT \*

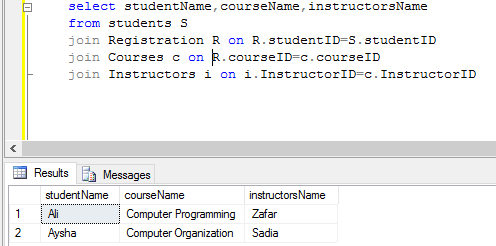
FROM <table1>

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



# References

Microsoft SQL Server 2008 Database Development Chapter 1, Lesson 1 and Lesson 2

