|  |
| --- |
| **National University of Computer and Emerging Sciences** |
| Lab Manual 3  **“Data Retrieval & Set Operations”** |
|  |
| Database Systems |
| **SPRING 2023** |

**Department of Computer Science**

**FAST-NU, Lahore, Pakistan**

Table of Contents

[1. Objective 2](#_Toc51880536)

[2. Pre-requisites 2](#_Toc51880537)

[3. SELECT-FROM-WHERE 3](#_Toc51880538)

[Most Basic Select: 3](#_Toc51880539)

[Retrieving Certain Columns from Select 4](#_Toc51880540)

[Retrieving Certain Rows from SELECT - WHERE CLAUSE 4](#_Toc51880541)

[Like Operator Scenarios 5](#_Toc51880542)

[Renaming Resulting Column 5](#_Toc51880543)

[SQL Server Built-in Functions 5](#_Toc51880544)

[4. Order by Clause 6](#_Toc51880545)

[TOP Clause 6](#_Toc51880546)

[5. Arithmetic Operations 7](#_Toc51880547)

[6. Set Operations 8](#_Toc51880548)

# Objective

The purpose of this manual is to get stared with data retrieval queries, starting from simple Select-From-Where, Order by clause, arithmetic operations and finally covering set operations.

# Pre-requisites

* Lab 2 manual, on how to get started with MS-SQL server
* How Select-From-Where clause works
* How Order by clause works
* How arithmetic operations like +, -, \*, /, % works
* How set Operations like Union, Intersect, Except work

**Task Distribution**

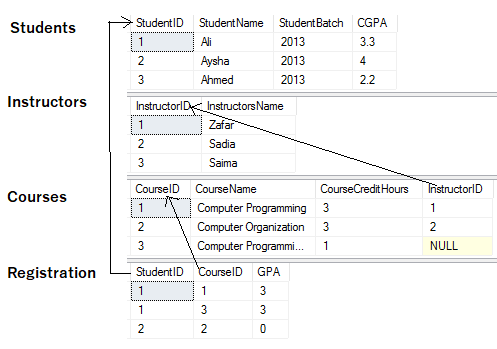
|  |  |
| --- | --- |
| Total Time | 170 Minutes |
| Select-From-Where | 25 Minutes |
| Order By | 10 Minutes |
| Arithmetic Operations | 10 Minutes |
| Set Operations | 15 Minutes |
| Exercise | 90 Minutes |
| Evaluation | Last 20 Minutes |

# 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. Find the queries for this database in InLab3Practice.sql and start practicing.



## Most Basic Select:

Retrieve data from table. Operator \* after select means that all columns will be retrieved.

**Syntax:**

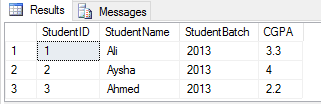
SELECT \*

FROM <tableName>

Try this



Results



## Retrieving Certain Columns from Select

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

**Syntax:**

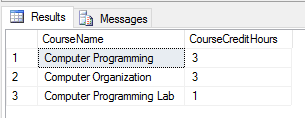
SELECT ColumnX, ColumnY, ColumnZ

FROM <tableName>

Try this



Results



## Retrieving Certain Rows from SELECT - WHERE CLAUSE

Rows can be filtered 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 in from clause. Also, observe the use of Like operator in where clause.

**Syntax:**

SELECT \*

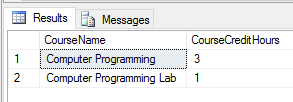
FROM <tableName>

where <conditions>

Try this



Results



## Like Operator Scenarios

|  |  |
| --- | --- |
| WHERE CourseName LIKE 'C%' | Finds any values that start with "C" |
| WHERE CourseName LIKE '%C' | Finds any values that end with "C" |
| WHERE CourseName LIKE '%Co%' | Finds any values that have "Co" in any position |
| WHERE CourseName LIKE '\_r%' | Finds any values that have "r" in the second position |
| WHERE CourseName LIKE 'C\_%' | Finds any values that start with "C" and are at least 2 characters in length |
| WHERE CourseName LIKE 'C\_\_% | Finds any values that start with "C" and are at least 3 characters in length |
| WHERE CourseName LIKE 'C%r' | Finds any values that start with "C" and ends with "r" |

**NOTE: %** is referred to as **wildcard**.

## 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.

**Syntax:**

SELECT ColumnX as X , ColumnY as Y, ColumnZ

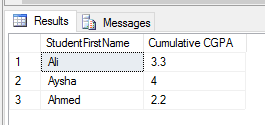
FROM <tableName> as Table1

## **SQL** Server Built-in Functions

Try this



Results



Sql Server has many built-in functions which can be used for different purposes.

For example:

1) GETDATE Returns the current database system date and time

2) CURRENT\_TIMESTAMP Returns the current date and time

3) SUBSTRING Extracts some characters from a string

**Syntax:**

1) SELECT GETDATE();

3) SELECT CURRENT\_TIMESTAMP;

2) SELECT SUBSTRING(columnName, startposition, substringlength) AS alias FROM <tableName>;

Try to explore as many string and data functions through this link: https://www.w3schools.com/sql/sql\_ref\_sqlserver.asp

# Order by Clause

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

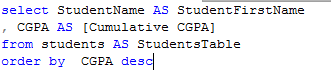
**Syntax:**

SELECT ColumnX as X, ColumnY as Y, ColumnZ

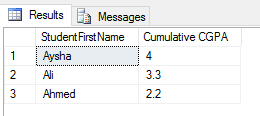
FROM <tableName> as Table1

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.

**Syntax:**

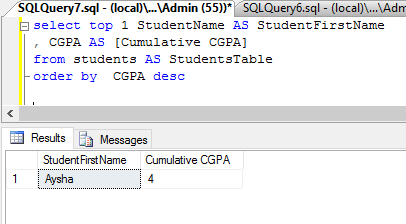
SELECT TOP <n> \*

FROM <tableName>

WHERE <conditions>

ORDER BY <column Name> asc/desc

Try this



# Arithmetic Operations

Sql arithmetic operators are:

* + Addition
* - Subtraction
* / Division
* \* Multiplication
* % Modulus

All operations can be performed on either single column or multiple columns

**Syntax:**

1. Apply operation on single columns

SELECT ColumnX, ColumnY + 100

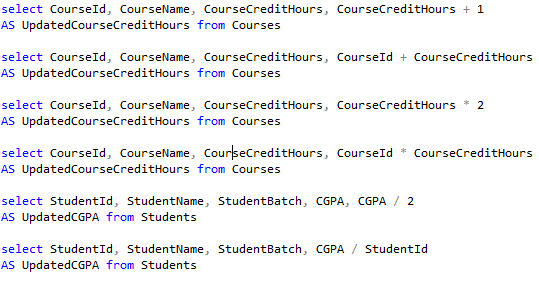
FROM <tableName>

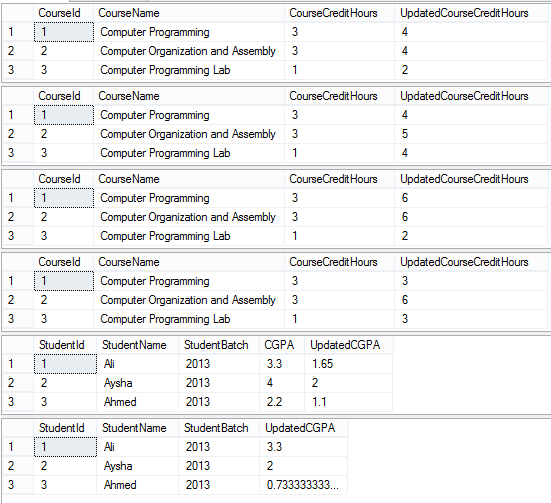
2. Apply operation on multiple columns

SELECT ColumnX, ColumnY + ColumnZ

FROM <tableName>

Replace + with other operators and try them out yourself.





# Set Operations

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

**Syntax:**

SELECT ColumnX, ColumnY

FROM <tableName>

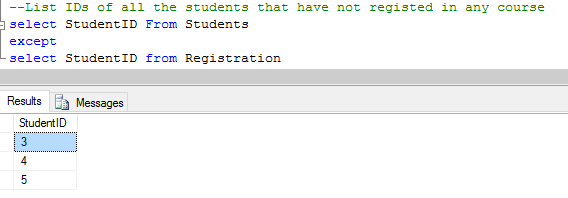
Union/Intersect/Except

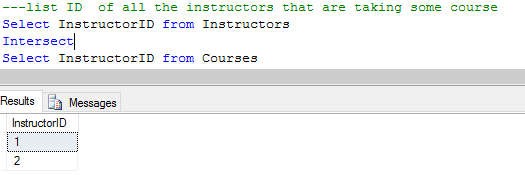
SELECT ColumnX, ColumnY

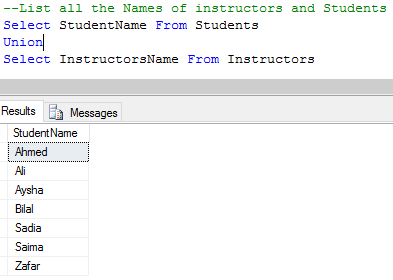
FROM <tableName>

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

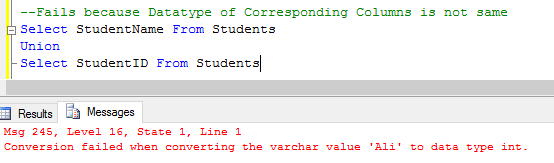
Try this –Set operations

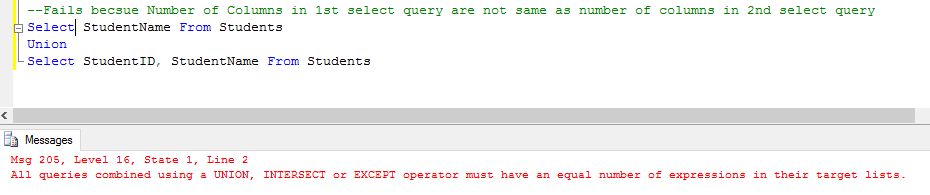






Try this - error to look out for in set operations





# Join Operation

## Inner Join:

Returns only those rows that match in both tables.

SELECT \*

FROM <table1> inner join <table2>

ON <Joining Condition>

