National University of Computer and Emerging Sciences

Lab Manual 4

"Data Retrieval Select-from-where, Joins"

Database Systems Lab

Spring 2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Contents

1. Objective

- This manual aims to get started with data retrieval queries, starting from Simple Select-From-Where, going towards Join operations.

2. Pre-requisites

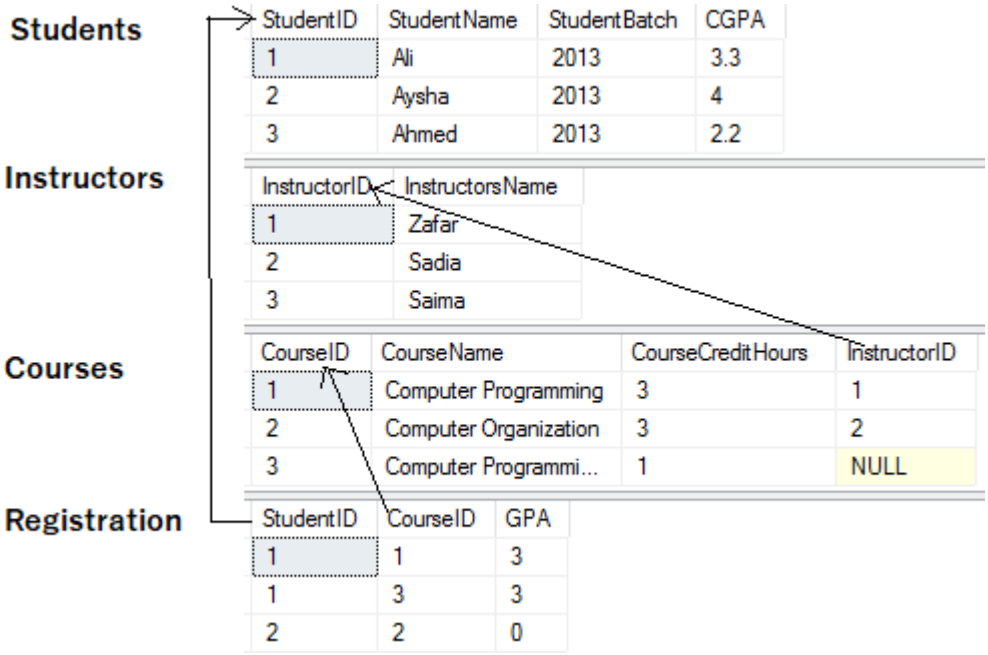
- Lab 2,3 manual, on how to get started with MS-SQL server
- How Select From Where clause work

Task Distribution

Total Time	170 Minutes
Select from where	15 Minutes
Order by	15 Minutes
Joining	15 Minutes
Exercise	125 Minutes

3. 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 Lab4Manual.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

4. 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

5. Join Operation

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>

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

Results Messages

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

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>

Try this

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

Self Join

A self join is a regular join, but the table is joined with itself.

```
SELECT column_name(s)
FROM table1 as T1, table1 as T2
WHERE condition
```