

Lab 06 Manual

Database Systems.



## Objective:

To explore concepts i.e., Wild Card Search, SQL Functions and its types, Views, Top-N Analysis, working with Date and Number and Joins.

## Scope:

The student shall know the following:

- Workaround SQL Server.
- SQL Practice.
- Hands-on experience on the above-mentioned concepts.

## Discussion:

### 1. Wild Card Search

A wildcard character is used to substitute one or more characters in a string. Wildcard characters are used with the LIKE operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

Wildcard Characters in SQL Server can be found in following table

Like Operator	Description
WHERE <i>columnName</i> LIKE 'a%'	Find any value that starts with "a".
WHERE <i>columnName</i> LIKE '%a'	Find any value that ends with "a".
WHERE <i>columnName</i> LIKE '%or%'	Find any value that has "or" in any position.
WHERE <i>columnName</i> LIKE '_a%'	Find any value that starts with any character but shall have second character as 'a'.
WHERE <i>columnName</i> LIKE 'a_%'	Find any value that starts with "a" but shall have at least 3 characters in length.
WHERE <i>columnName</i> LIKE 'a%o'	Find any value that starts with "a" and ends with "o".

### Exercise:

1. Display all employees who have four figure salary?

2. Display information of employees whose name start with 'S' and shall have length 4?

Solution:

1. `SELECT *FROM dbo.EMP WHERE SAL LIKE '____.00%';`
2. `SELECT *FROM dbo.EMP WHERE ENAME LIKE 'S____';`

## 2. Functions:

Functions are a very powerful feature of SQL. They can be used to do the following:

- Perform calculations on data
- Modify individual data items
- Manipulate output for groups of rows
- Format dates and numbers for display
- Convert column data types
- SQL functions sometimes take arguments and always return a value.

### Two Types of SQL Functions

There are two types of functions:

- Single-row functions
- Multiple-row functions

### Single-Row Functions

These functions operate on single rows only and return one result per row. There are different types of single-row functions.

Single-row functions do the following:

- Manipulate data items
- Accept arguments and return one value
- Act on each row that is returned
- Return one result per row
- May modify the data type and accepts arguments that can be a column or an expression.

### Syntax:

*functionName*(arg1, arg2, arg3,.....argN)

In the syntax:

**functionName** is the name of the function

**arg1, arg2, arg3** is any argument to be used by the function. This can be represented by a column name or expression.

This lab covers the following single-row functions

- **Character functions:** accepts character input and can return both character and number values.

Function	Result
CONCAT('Hello', 'World')	HelloWorld
SUBSTRING('HelloWorld', 1, 5)	Hello
LEN('HelloWorld')	10
CHARINDEX('HelloWorld', 'W')	6
REPLACE('JACK and JUE', 'J', 'BL')	BLACK and BLUE

### Exercise:

1. `SELECT CONCAT(ENAME, ' is ' JOB) AS INTRO from dbo.EMP;`
2. `SELECT SUBSTRING(ENAME, 1, 3) from dbo.EMP;`

```

3. SELECT LEN(ENAME) from dbo.EMP;
4. SELECT CHARINDEX('S', ENAME) from dbo.EMP WHERE EMPNO = 1234;
5. SELECT REPLACE(ENAME, 'S', '*') from dbo.EMP;

```

- **Numeric functions:** accepts numeric input and return numeric values.

Function	Result
ROUND(124.25, 1)	124.30
ABS(-385)	385
AVG(columnName)	Returns average

The list of numeric function includes SQRT(), SIN(), RAND() and POWER().

**Exercise:**

```

1. SELECT ROUND(124.255, 2, 0);
2. SELECT ABS(-850);
3. Also try other functions aswell.

```

- **Date functions:** operates on values of the DATE data type (All date functions return a value of the DATE data type except the MONTHS\_BETWEEN function, which returns a number.)

```

1. SELECT GETDATE();
2. SELECT DATENAME(YEAR, GETDATE()) AS 'Year';
3. SELECT DATENAME(QUARTER, GETDATE()) AS 'Quarter';
4. SELECT DATENAME(MONTH, GETDATE()) AS 'Month Name';
5. SELECT DATENAME(DAYOFYEAR, GETDATE()) AS 'DayOfYear';
6. SELECT DATENAME(DAY, GETDATE()) AS 'Day';
7. SELECT DATENAME(WEEK, GETDATE()) AS 'Week';
8. SELECT DATENAME(WEEKDAY, GETDATE()) AS 'Day of the Week';
9. SELECT DATENAME(HOUR, GETDATE()) AS 'Hour';
10. SELECT DATENAME(MINUTE, GETDATE()) AS 'Minute';
11. SELECT DATENAME(SECOND, GETDATE()) AS 'Second';
12. SELECT DATEPART(YYYY, GETDATE());
13. SELECT DATEPART(mm, GETDATE());
14. SELECT DATEPART(dd, GETDATE());

```

Also explore functions like DATEADD(), DATEIFF(), CONVERT().

- **Case-conversion functions:** converts the case of the string i.e., lowercase, uppercase.

```
1. SELECT CONVERT (VARCHAR(30), GETDATE(), 23);  
2. SELECT CAST (25.65 AS varchar);  
3. SELECT CAST ('2017-08-25' AS datetime);  
4. SELECT LOWER (ENAME) AS lower_case, UPPER (ENAME) AS higher_case from  
    dbo.EMP;
```

## General Functions:

### NULL Function in SQL Server

```
1. SELECT ISNULL (COMM, 0) from dbo.EMP;
```

## Multiple-Row Functions

Functions can manipulate groups of rows to give one result per group of rows. These functions are also known as group functions.

```
1. SELECT MAX(SAL) AS MaximumSalary from dbo.EMP;  
2. SELECT MIN(SAL) AS MinimumSalary from dbo.EMP;  
3. SELECT AVG(SAL) AS AverageSalary from dbo.EMP;
```

Note: Explore COUNT(), count\_big() as well

### Exercise:

1. Display name of employees along with JOB title?
2. Display hire date of employees in US-English format?
3. Display names of employees in lower case?
4. Replace employee name with last name?

### Solution:

```
1. SELECT CONCAT (ENAME, ' is ', JOB) FROM dbo.EMP;  
2. SELECT FORMAT (GETDATE(), 'd') AS 'US ENGLISH Format';  
3. SELECT LOWER (ENAME) FROM dbo.EMP;  
4. SELECT REPLACE (ENAME, 'SAMI', 'ULLAH') AS 'Last Name' FROM dbo.EMP WHERE ENAME  
    = 'SAMI';
```

## 3. Usage of Views in SQL Server

Views are virtual tables that hold data from one or more tables. It is stored in the database. A view does not contain any data itself, it is a set of queries that are applied to one or more tables that are stored within the database as an object. Views are used for security purposes in databases. Views restrict the user from viewing certain columns and rows. In other words, using a view we can apply the restriction on accessing specific rows and columns for a specific user. A view can be created using the tables of the same database or different databases. It is used to implement the security mechanism in the SQL Server.

Similarly, Views are used to implement the security mechanism in SQL Server. Views are generally used to restrict the user from viewing certain columns and rows. Views display only the data specified in the query, so it shows only the data that is returned by the query defined during the creation of the view. The rest of the data is totally abstract from the end user.

### Creating SQL View in SQL Server

```
1. CREATE VIEW EMP_NAME AS
   SELECT ENAME, JOB, HIREDATE
   FROM dbo.EMP;
```

### Retrieve Data from View in SQL Server

```
1. SELECT *FROM EMP_NAME;
```

Further, you can do multiple operations on views i.e., dropping a view, renaming the view and do alteration in the view.

## 4. Top-N Analysis

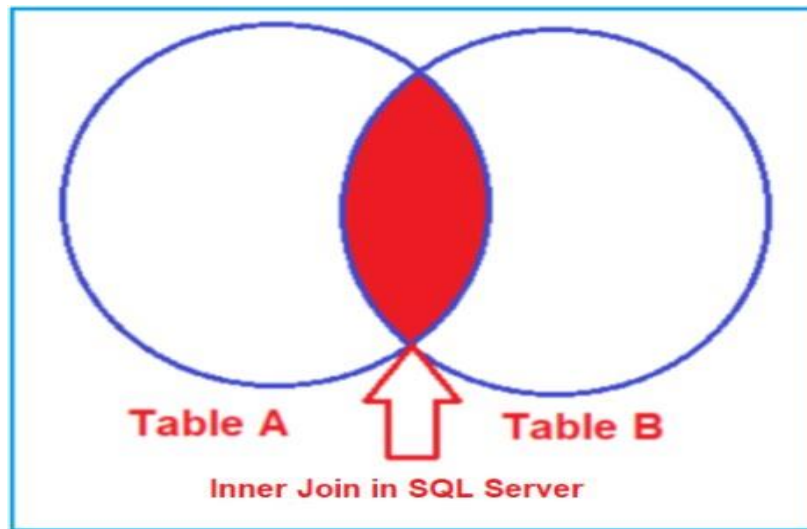
Sometimes we want to retrieve a specific number of records from the SQL table rather than retrieving all table records. We can use the WHERE clause in a SELECT statement to limit the number of rows. Suppose you have a customer table and you want to retrieve records for customers belonging to a particular country. There might be many records satisfying the condition. We require top 'N' customer records to satisfy the conditions. For this requirement, we can use TOP clauses.

```
1. SELECT TOP (10) *FROM dbo.EMP;
```

## 5. SQL JOINS:

The SQL Server Joins are used to retrieve the data from two or more related tables. In general, tables are related to each other using the primary key and foreign key relationship but it is not mandatory. The tables involved in the joins must have a common field. And based on that common field the SQL Server JOINS retrieves the records. There are different types of *Joins* available in SQL Server i.e., Inner, Outer, Left Outer, Right Outer and Full Outer Join.

### Inner Join:

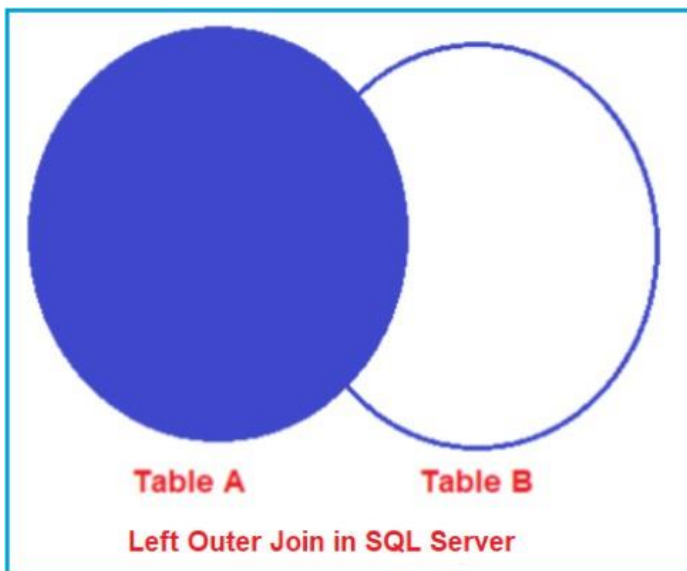


```
SELECT ENAME, JOB, DNAME FROM dbo.EMP INNER JOIN dbo.DEPT ON dbo.EMP.DEPTNO =  
dbo.DEPT.DEPTNO;
```

### Outer Joins:

There are different varieties of outer joins which are discussed one by one in detail

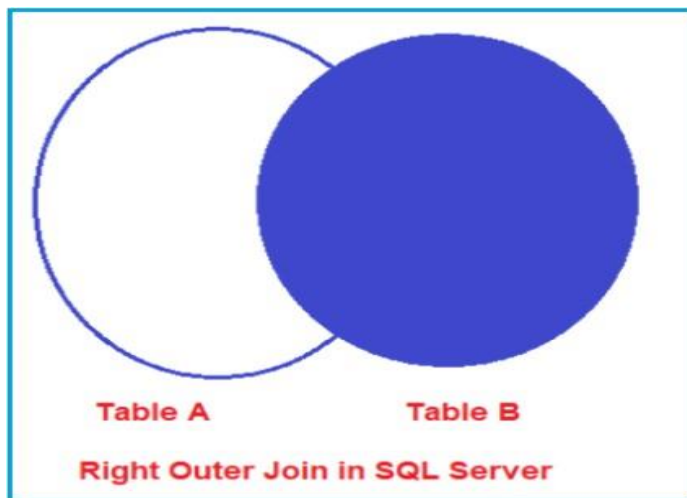
#### Left Outer Join



```
SELECT ENAME, JOB, DNAME FROM dbo.EMP LEFT OUTER JOIN dbo.DEPT ON dbo.EMP.DEPTNO =  
dbo.DEPT.DEPTNO;
```



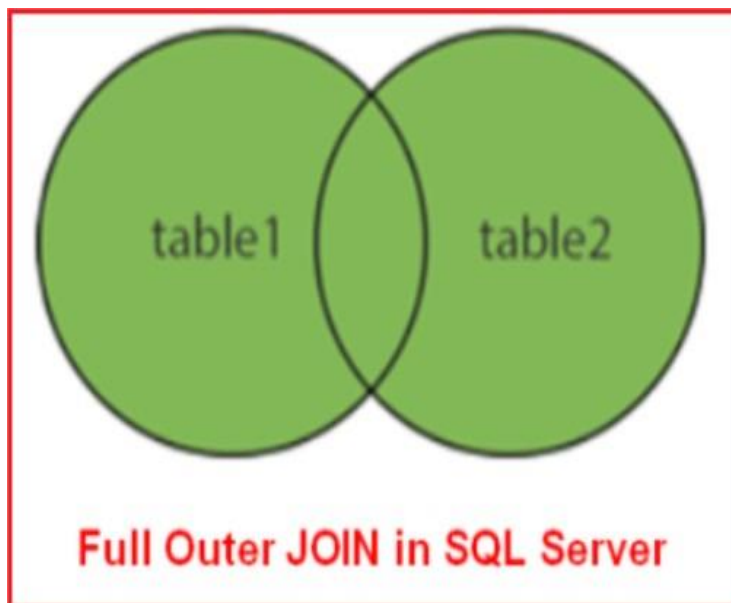
## Right Outer Join



```
SELECT ENAME, JOB, DNAME FROM dbo.EMP RIGHT OUTER JOIN dbo.DEPT ON dbo.EMP.DEPTNO =  
dbo.DEPT.DEPTNO;
```

## Full Outer Join

```
SELECT ENAME, JOB, DNAME FROM dbo.EMP FULL JOIN dbo.DEPT ON dbo.EMP.DEPTNO =  
dbo.DEPT.DEPTNO;
```



**Lab Task:**

1. The HR department wants to determine the names of all the employees who were hired after Smith. Create a query to display the name and hire date of any employee hired after employee Smith.
2. The HR department needs to find the names and hire dates of all the employees who were hired before their managers, along with their managers' names and hire dates.
3. The HR department needs a report of all employees. Write a query to display the name, department number, and department name for all the employees sorted according to increasing order of department number.
5. Create a report to display employees' name and employee number along with their manager's name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.
6. Retrieve the names of managers (MANAGER) and their respective department names.