ST. XAVIER’S COLLEGE

**Maitighar,Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**Lab Assignment #8**

**Submitted By**

Aashish Raj Shrestha

B.Sc. CSIT

Year II/IV Semester

013BSCIT002

**Submitted To**

Er. Sanjay Kumar Yadav

Lecturer,

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

September 27, 2015

4.2. Data Definition Language (DDL):

4.2.1. Domain Type in SQL:

4.2.2. Schema Definition in SQL:

4.3. Data Manipulation Language (DML):

4.3.1. The SELECT Clause:

4.3.2. The WHERE Clause:

4.3.3. The FROM Clause:

4.3.4. The RENAME Clause:

4.3.5. Tuple Variable:

4.3.6. String Operations:

SQL Server provides the following string operators. String concatenation operators can combine two or more character or binary strings, columns, or a combination of strings and column names into one expression. Wildcard string operators can matches one or more characters in a string comparison operation such as LIKE or PATINDEX.

1. **+ (String Concatenation) :**

An operator in a string expression that concatenates two or more character or binary strings, columns, or a combination of strings and column names into one expression (a string operator)[1].

**Syntax:**

expression + expression

**Arguments:**

*expression*

Is any valid expression of any one of the data types in the character and binary data type category, except the**image**, **ntext**, or **text** data types. Both expressions must be of the same data type, or one expression must be able to be implicitly converted to the data type of the other expression.

An explicit conversion to character data must be used when concatenating binary strings and any characters between the binary strings. The following example shows when CONVERT, or CAST, must be used with binary concatenation and when CONVERT, or CAST, does not have to be used.

**Example:**

The following example concatenates multiple strings to form one long string to display the last name and the first initial of the vice presidents at Adventure Works Cycles. A comma is added after the last name and a period after the first initial.

USE AdventureWorks2012;

GO

SELECT (LastName + ',' + SPACE(1) + SUBSTRING(FirstName, 1, 1) + '.') AS Name, e.JobTitle

FROM Person.Person AS p

JOIN HumanResources.Employee AS e

ON p.BusinessEntityID = e.BusinessEntityID

WHERE e.JobTitle LIKE 'Vice%'

ORDER BY LastName ASC;

GO

**Result:**

Name               Title

-------------      ---------------

Duffy, T.          Vice President of Engineering

Hamilton, J.       Vice President of Production

Welcker, B.        Vice President of Sales

(3 row(s) affected)

1. **+= (String Concatenation):**

Concatenates two strings and sets the string to the result of the operation. For example, if a variable @x equals 'Adventure', then @x += 'Works' takes the original value of @x, adds 'Works' to the string, and sets @x to that new value 'AdventureWorks'.

**Syntax:**

expression += expression

**Arguments:**

*expression*

Is any valid expression of any of the character data types.

**Example:**

The following example concatenates using the += operator.

DECLARE @v1 varchar(40);

SET @v1 = 'This is the original.';

SET @v1 += ' More text.';

PRINT @v1;

**Result:**

This is the original. More text.

1. **% (Wildcard - Character(s) to Match):**

Matches any string of zero or more characters. This wildcard character can be used as either a prefix or a suffix.

* 1. **‘ABC%’:**

All string that start with ABC. E.g: ABCD and ABCD ABC Both Satisfy the condition.

* 1. **‘%XYZ’:**

All string that end with XYZ. E.g. WXYZ, and ZZXYZ

* 1. **‘%AN%’:**

All string that contain the pattern ‘---AN---‘ anywhere. E.g. LOS ANGELES AND SAN FRANCISCO

1. **[ ] (Wildcard - Character(s) to Match):**

Matches any single character within the specified range or set that is specified between the brackets. These wildcard characters can be used in string comparisons that involve pattern matching, such as LIKE and PATINDEX.

**Examples:**

The following example uses the [] operator to find the IDs and names of all Adventure Works employees who have addresses with a four-digit postal code.

USE AdventureWorks2012;

GO

SELECT e.BusinessEntityID, p.FirstName, p.LastName, a.PostalCode

FROM HumanResources.Employee AS e

INNER JOIN Person.Person AS p ON e.BusinessEntityID = p.BusinessEntityID

INNER JOIN Person.BusinessEntityAddress AS ea ON e.BusinessEntityID = ea.BusinessEntityID

INNER JOIN Person.Address AS a ON a.AddressID = ea.AddressID

WHERE a.PostalCode LIKE '[0-9][0-9][0-9][0-9]';

**Results:**

EmployeeID FirstName LastName PostalCode

---------- --------- --------- ----------

290 Lynn Tsoflias 3000

1. **[^] (Wildcard - Character(s) Not to Match):**

Matches any single character that is not within the range or set specified between the square brackets.

**Examples:**

The following example uses the [^] operator to find all the people in the Contact table who have first names that start with Al and have a third letter that is not the letter a.

USE AdventureWorks2012;

GO

SELECT FirstName, LastName

FROM Person.Person

WHERE FirstName LIKE 'Al[^a]%'

ORDER BY FirstName;

1. **\_ (Wildcard - Match One Character):**

Matches any single character in a string comparison operation that involves pattern matching, such as LIKE and PATINDEX.

**Examples:**

The following example uses the **\_** operator to find all the people in the Person table, who have a three-letter first name that ends in an.

USE AdventureWorks2012;

GO

SELECT FirstName, LastName

FROM Person.Person

WHERE FirstName LIKE '\_an'

ORDER BY FirstName;

4.3.7. Ordering the Display of Tuples:

4.3.8. Duplicate Tuples:

References:

[1] https://msdn.microsoft.com/en-us/library/ms190301.aspx