

28-11-2024

## T/SQL Programming

=====

=====

> T/SQL stands for "Transact Structure Query Language. It is an extension of SQL language. This T/SQL is same as PL/SQL in oracle.

> In SQL we can execute single line statement only where as in T/SQL we can execute block of statements at a time.

> SQL does not support conditional and looping statements like IF-Else and While loop. But we can implement these conditional and looping statements in T/SQL.

> SQL language will not provide reusability facilities where as T/SQL language will provide reusability facilities by defining objects such as Procedures and Functions.

> T/SQL commands can be embedded inside the programs where program is a block of code.

> T/SQL Program blocks can be divided into two types. Those are

1. Anonymous Blocks
2. Sub-Program Blocks

Anonymous blocks VS Sub blocks:

=====

Anonymous blocks

=====

1. these are unnamed blocks.

2. these blocks are not saved the code in database.

3. it is a temporary block.

4. we cannot reusable code.

5. every time compilation of the code.

6. these blocks are used in "DB testing".  
Developement".

Sub blocks

=====

1. these are named blocks.

2. these are saved the code in database automatically.

3. it is a permanent block.

4. we can reusable code.

5. it is a pre-compiled code.  
(i.e it compiled first time only)

6. these blocks are used in "Application

( DB applications / .NET applications / JAVA applications)

Working with variables in T/SQL:

=====

- When we want to work with variables in T/SQL then we follow the following three steps are,

step1: Declare variables:

=====

syntax:

=====

DECLARE @<variable name1> <datatype>[size],.....;

step2: Assigning values to the variables:

=====

syntax:

=====

SET @<variable name> = <value>;

step3: Print variables values:

=====

syntax:

=====

PRINT @<variable name>;

EX:

DECLARE @X INT,@S VARCHAR(10),@M MONEY

SET @X=1021 SET @S='SMITH' SET @M=25000

PRINT @X

PRINT @S

PRINT @M

EX:

DECLARE @STID INT,@SNAME VARCHAR(10),@S1 INT,@S2 INT,@S3 INT,@TM INT

SET @STID=1021 SET @SNAME='JONES' SET @S1=98 SET @S2=85 SET @S3=73

SET @TM=@S1+@S2+@S3

PRINT 'STUDENT ID:-'+CAST(@STID AS VARCHAR)

PRINT 'STUDENT NAME:-'+@SNAME

PRINT 'TOTAL MARKS:-'+CAST(@TM AS VARCHAR)

OUTPUT:

=====

STUDENT ID:-1021

STUDENT NAME:-JONES

TOTAL MARKS:-256

29-11-2024:

=====

How to store columns values into variables in T/SQL:

=====

syntax:

=====

```
SELECT @<variable name1> =<column name1>,@<variable name2>=<column name2>,...
FROM <TABLE NAME> WHERE <CONDITION>;
```

EX:

write a t/sql program to input EMPNO and display that ENAME,SALARY details from emp table?

```
DECLARE @EMPNO INT,@ENAME VARCHAR(10),@SAL MONEY
```

```
SET @EMPNO=7788
```

```
SELECT @ENAME=ENAME,@SAL=SAL FROM EMP WHERE EMPNO=@EMPNO
```

```
PRINT 'EMPLOYEE NAME IS:-'+@ENAME
```

```
PRINT 'EMPLOYEE SALARY IS:-'+CAST(@SAL AS VARCHAR)
```

OUTPUT:

=====

```
EMPLOYEE NAME IS:-SCOTT
```

```
EMPLOYEE SALARY IS:-3000.00
```

Conditional statements:

=====

i) Simple If statement :

=====

syntax:

=====

```
if(condition)
    statements;----- true block
```

EX:

```
DECLARE @X INT
```

```
SET @X=3
```

```
IF((@X%2)=0)
```

```
PRINT 'EVEN NUMBER'; ----- true block statement only
```

ii) If - else statement:

=====

syntax:

=====

```
if(condition)
```

```
        statements; -----true block
else
        statements; -----false block
```

EX:

```
DECLARE @X INT
SET @X=4
IF((@X%2)=0)
PRINT 'EVEN NUMBER' -----> true block statement
ELSE
PRINT 'ODD NUMBER'-----> false block statement
```

iii) If - else - If statement:

=====

syntax:

=====

```
    if(condition)
        statements;
    else if(condition)
        statements;
    else
        statements;
```

EX:

```
DECLARE @X INT,@Y INT
SET @X=40 SET @Y=30
IF (@X>@Y)
PRINT 'X IS A BIG NUMBER'
ELSE IF(@X=@Y)
PRINT 'BOTH ARE EQUAL'
ELSE
PRINT 'Y IS A BIG NUMBER'
```

While loop statement:

=====

syntax:

=====

```
    while <condition>
    begin
        <statements>;
        <increment / decrement>;
    end
```

EX on increment:

=====

```
DECLARE @X INT
SET @X=1
WHILE(@X<=10)
BEGIN
PRINT @X;
SET @X=@X+1
END
```

EX on decrement:

=====

```
DECLARE @X INT
SET @X=10
WHILE(@X>=1)
BEGIN
PRINT @X;
SET @X=@X-1
END
```