



International Journal of Allied Practice, Research and Review

Website: www.ijaprr.com (ISSN 2350-1294)

Database Programming Skills

Junaida Shafi

M.Tech Student, Ganpati Institute of Management & Technology, Krukshetra University, India
littlechapjuni@gmail.com

Abstract— Databases play a very important role in application development and from decades we are using different database technologies to manage data efficiently like relational database technology. From the development perspective there are number of programming skills required to design a very efficient database for data management.

Keywords— Database; RDBMS; SQL; Batches; Go; Constructs

I. INTRODUCTION

Database plays a vital role in day to day life .where ever the concept of data came, the database word emerges. Whenever we want to store the data dynamically the concept of database is used. Various data warehouses are being used to store the enormous amount of data and various algorithms have been used in mining out the data from data warehouses especially when we talk about the concept of programming in database. Various programming skills are being used which make it easier to handle the data. The various programming skills like use of variables, conditional constructs, iterative constructs, exceptions and have been used in SQL server to make the data handling easy. Retrieval of data is a prime operation in database from database administrator's point of view.

II. BATCHES

Batches are group of statements send to SQL server for execution. The SQL server executes this group of statement as a single executable unit called Execution Plan.

GO keyword is used to end the group of statements. We use PRINT keyword also to display the user defined messages in case of batches. Like

<T-SQL Statements>

<T-SQL Statements>

<T-SQL Statements>

....
GO

Insert into student values (1,'junaid','mohammad shafi','m-tech')

Insert into student values (2,'farzana','peerzada mehraj-ud-din','b-tech')

Insert into student values (3,'saniya','zahoor Ahmed','m-tech')

GO

When a batch is submitted to SQL Server it is compiled to create an execution plan. The execution plan is not created if a compilation error occurs such as syntax error it means none of the statements will be executed.

III.USING VARIABLES

While using Batches you need to store some values temporarily during the execution time for that purpose we use Variables. Variables are used to hold the value. Variables are declared by using the keyword DECLARE. The name of variable is prefixed by @ sign, and followed by the data type. Batches also allow us to perform particular actions based on multiple conditions like.

```
DECLARE @variable_name data type...
```

```
Like declare @num int.
```

```
SELECT @num=SELECT sum (marks) from student.
```

```
PRINT @num.
```

```
GO
```

IV.CONSTRUCTS

SQL Server allows you to use programming constructs in batches for conditional execution of statements. Whenever you need data or an output based on some particular condition we use constructs to deal with them. We use following types of constructs in SQL Server like:

IF...ELSE statement

CASE statement

WHILE statement

IF...ELSE is the basic type of conditional construct which is executed on the basis of a condition. If the condition is true it is followed by group of statements .if the condition is not true, it will be followed by the else block statements. It means if else statements always works on Boolean expression. The syntax is:

```
IF Boolean_expression {SQL_statement |statement_block}  
ELSE  
{SQL_statement |statement_block}
```

FOR EXAMPLE:

```
DECLARE @Marks int  
SELECT @Marks=Marks from student where reg_id=12  
IF @Marks<25  
PRINT 'Review of the Marks is required'  
ELSE  
BEGIN  
PRINT 'Marks is not required to review'  
PRINT @Marks  
END  
GO
```

V.USING CASE STATEMENT

We use case statements where we have to evaluate multiple conditions simultaneously. The case statements check multiple conditions and return the one of the possible output. If all the conditions that needed to be checked are false then the control directly follows the default conditions. The syntax of case statement is;

```
CASE input_expression  
WHEN when_expression THEN result_expression  
[WHEN when_expression THEN result_expression] [.....]  
ELSE  
Else_result_expression  
END
```

Consider the following situation where a CASE construct is concluded in the SELECT statement to display the grades of student in class.

```
SELECT Reg_id, Name, 'Grade of Student'=case Marks  
When Marks >45 then 'Distinction'  
When Marks >35 then 'grade A'  
When Marks > 25 then 'Grade B'  
ELSE 'failed'  
END  
From Student  
GO
```

VI. USING WHILE STATEMENTS

While statement is a type of iterative construct where a set of SQL statements are to be executed repeatedly as long as a given condition holds true. The syntax of while statement is as

```
While boolean_expression  
{SQL_statement | block_statement}  
[BREAK]  
{SQL_statement | block_statement}  
CONTINUE
```

SQL Server provides the BREAK and CONTINUE statements to control the statements within the while loop. The BREAK statements causes an exit from the while loop. The CONTINUE statements cause the while loop to restart skipping any statement after this statement inside the loop.

```
WHILE (SELECTAVG (Marks) from Student) <40  
BEGIN  
UPDATE Student set Marks=Marks+10 from Student  
If SELECT (MAX (Marks) FromStudent)>40  
Break  
ELSE  
CONTINUE  
END
```

VII. ERRORS AND EXCEPTIONS

Exceptions are run time errors that can be handled in programming as well as in database. Unlike syntax errors and logical errors when we execute a query, it is parsed for syntactical errors before execution. If the syntax is correct, it is compiled and executed. For example, there is a primary key constraint applied on reg. id attribute of the Student table. When you try to insert a reg. id which already exists in the table, the error occurs while executing the insert statement. Exceptions can be handled in SQL server in following ways:

- A. By using the TRY...CATCH statement.
- B. By using the RAISEERROR statement.

1) USING TRY CATCH

A TRY...CATCH construct includes a TRY block followed by a CATCH block. A TRY block is a group of SQL statements enclosed in a batch, stored procedure, a trigger and a function. If an error occurs in a statement of the TRY block, the control is passed on to the block of statement followed by it called CATCH block. A CATCH block contains SQL statement that performs action when an error occurs.

TRY

```

<SQL statement>
.....
CATCH
<SQL statements>
.....
END CATCH

```

The TRY ...CATCH construct can be nested. Either a TRY block or a CATCH block can contain nested TRY.....CATCH statements.

```

BEGIN TRY
INSERT INTO STUDENT VALUES (1,'JUNAID SHAFI','NAID KADAL',
GETDATE (), 49)
INSERT INTO STUDENT VALUES (1,'NASEER GANIE','CHADOORA,
GETDATE (), 49)
END TRY
BEGIN CATCH
SELECT 'THERE WAS AN ERROR '
END CATCH
GO

```

2) USING RAISE ERROR

A RAISE ERROR Statement is used to return messages to the business applications when executing the SQL statements. This statement uses the same format as the system error or warning messages generated by database engine of the SQL server component. Consider an application that is executing the BATCH. While executing the BATCHE an error message will be generated and send to the application.

You can return user-defined messages by using the RAISEERROR statement .the syntax of RAISE ERROR statement is

```
RAISEERROR ('Message', Severity, State)
```

Where

Message is the text you want to display.

Severity is the user defined severity level associated with the message. It represents how severe the error is. Severity level can range from 0 to 25.the levels from 0 to 18 can be specified by any user.

State is an integer value from 0 to 255.

For example,

```

BEGIN TRY
DECLARE @Start datetime
DECLARE End datetime
DECALARE @Date_diff
SELECT @Start='1987-04-10 04:05:32:00', @End=GETDATE ()
SELECT @Date_diff=datediff (hh, @Start, @End)
IF (@Date_diff! =8)
RAISEERROR ('ERROR RAISED', 16, 1)

```

```
ELSE
BEGIN
UPDATE HumanResources.Shift
Set StartTime=@Start,EndTime=@End where ShiftID=3
END
END TRY
BEGIN CATCH
PRINT 'the difference between the start and end time should be 8 hours '
END CATCH
END
GO
```

VIII . ACKNOWLEDGMENT

I wish to acknowledge all the people who have worked in Database Technology and whose work inspired me to write this research paper in Database programming. I also would like to acknowledge my parents and my colleagues who supported me to write my first research paper and put my ideas and research on paper as a contribution to coming generation.

IX. REFERENCES

- [1]Leonard G.Lobel, Andrew J.Brust "Programming SQL Server 2012".
- [2] Grant Fritchey "SQL Server Execution Plans".
- [3]Ross Mistry, Stacia Misner "Introducing MS SQL SERVER 2012".
- [4] Patrick Leblane" MS SQL server 2012 step by step".
- [5]Paul Atkinson, Robert Vieira" Beginning MS SQL Server 2102 Programming".
- [6] Grega jerkic, Matija Lah, Dejon Sarka" Implementing a Data Warehouse with SQL Server 2012".
- [7]Brian knights, Devin Knight, Davis Wayne "KNIGH'S MICROSOFT SQL SERVER 2012 INTEGRATION SERVICE 24_HOUR TRAINER".
- [8] Itzik Ben_Gan"Microsoft SQL Server 2012 High Performance T-Sql".
- [9] William R Stanek"Microsoft SQL Server 2012 Pocket Consultant"