SQL Script

Pemrograman SQL II

PMI 1417

4(2/2)

Eko Win Kenali, S.Kom., M.Cs.

1 SQL Script

Script SQL adalah koleksi statement SQL yang ditulis dan akan dieksekusi oleh compiler SQL dalam sekali operasi.

Misalnya statement SQL dasar seperti ***insert, select, delete, update, where***, dll.

Pendefinisian dan penggunaan variables dalam script SQL. Penggunaan perintah untuk kendali proses seperti IF-ELSE, WHILE, dll.

Baris komentar juga dapat digunakan didalam script SQL untuk memudahkan pembacaan dan memahami script.

1. Comment
2. Variables
3. Global Variabels
4. Flow Control

Comment | Komentar

Penggunaan komentar di dalam script SQL sangat penting untuk kemudahan membaca dan memahami script SQL, khususnya untuk script SQL yang panjang. Dan baris komentar tidak akan diekseskusi (diabaikan) oleh compiler.

2 (dua) jenis komentar dalam SQL Script, yaitu :

1. Single-Line Comment
2. Multiple-Line Comment

Comment | Komentar

Single-Line Comment

Komentar dalam satu baris perintah, menggunakan simbol ‘--’ diawal teks komentar. Sintak perintah sebagai berikut :

```
-- text_of_comment
```

Contoh :

```
-- input data mahasiswa
```

Comment | Komentar

Multiple-Line Comment

Komentar dengan banyak baris perintah. Diawali dengan simbol ‘ /* ’ dan diakhiri dengan simbol ‘ */ ‘. Sintak perintah sebagai berikut :

```
/*
text_of_comment_1
text_of_comment_2
*/
```

Contoh :

```
/*
input data mahasiswa
Jika program studi mahasiswa tersebut adalah manajemen informatika
*/
```

Variable

Salah satu fitur M. SQL-Server adalah Variabel. Kata kunci DECLARE wajib digunakan untuk mendefinisikan variabel. Variabel lokal harus memiliki simbol “@” sebagai awalan penulisan nama variabel. Tipe data untuk variabel (int, varchar(x), dll.) juga perlu didefinisikan.

Sintak perintah deklarasi variabel :

```
declare @local_variable data_type
```

```
declare
@local_variable_1 data_type,
@local_variable_2 data_type,
...
```

1 SQL Script

Variabel

Setelah variabel dideklarasikan, sebuah variabel dapat diberi nilai. Pemberian nilai pada variabel melibatkan keyword ‘ **set** ’. Kemudian nilai dalam variabel dapat dicetak/ditampilkan, dengan keyword ‘ **print** ’.

Sintak perintah deklarasi variabel, pemberian dan mencetak nilai variabel :

declare @*local_variable* data_type

set @*local_variable = value*

print @*local_variable*

Variabel

Contoh penggunaan variabel dalam statement SELECT :

```
declare @find varchar(30)
set @find = 'J%'
select * from CUSTOMER where LastName LIKE @find
```

Global Variable (@@IDENTITY)

Setelah perintah INSERT INTO, SELECT, berhasil dilakukan, jika sebuah tabel memiliki kolom yang memiliki nilai constraint IDENTITY, maka variabel **@@IDENTITY** akan memiliki nilai terakhir yang telah ditambahkan otomastis. @@IDENTITY disebut sebagai variabel global yang dapat digunakan pada proses selanjutnya.

Contoh :

school

	SchoolId	SchoolName	Description	Address	Phone	PostCode	PostAddress
1	1	TUC	NULL	NULL	NULL	NULL	NULL
2	2	NTNU	NULL	NULL	NULL	NULL	NULL

sourse

	CourseId	CourseName	SchoolId	Description
1	1	SCE2006	1	NULL
2	2	SCE1106	1	NULL
3	3	SCE4206	1	NULL
4	4	SCE4106	1	NULL

Global Variabel

Contoh penggunaan variabel global :

```
declare @SchoolId int
-- Insert Data into SCHOOL table
insert into SCHOOL(SchoolName) values ('MIT')
select @SchoolId = @@IDENTITY
-- Insert Courses for the specific School above in the COURSE table
insert into COURSE(SchoolId,CourseName) values (@SchoolId, 'MIT-101')
insert into COURSE(SchoolId,CourseName) values (@SchoolId, 'MIT-201')
```

	SchoolId	SchoolName	Description	Address	Phone	PostCode	PostAddress
1	1	TUC	NULL	NULL	NULL	NULL	NULL
2	2	NTNU	NULL	NULL	NULL	NULL	NULL
3	16	MIT	NULL	NULL	NULL	NULL	NULL

	CourseId	CourseName	SchoolId	Description
1	1	SCE2006	1	NULL
2	2	SCE1106	1	NULL
3	3	SCE4206	1	NULL
4	4	SCE4106	1	NULL
5	5	MIT-101	16	NULL
6	6	MIT-201	16	NULL

Flow Control

IF_ELSE

```
declare @customerNumber int
select @customerNumber=CustomerNumber from CUSTOMER
where CustomerId=2
if @customerNumber > 1000
        print 'The Customer Number is larger than 1000'
else
        print 'The Customer Number is not larger than 1000'
```

FLOW CONTROL (IF_ELSE)

Penggunaan Blog dengan BEGIN dan END;

```
select @customerNumber=CustomerNumber from CUSTOMER where CustomerId=2
if @customerNumber > 1000
BEGIN
        print 'The Customer Number is larger than 1000'
        update CUSTOMER set AreaCode=46 where CustomerId=2
END
else
        print 'The Customer Number is not larger than 1000'
```

Flow Control

WHILE

```
while (select AreaCode from CUSTOMER where CustomerId=1) < 20
begin
      update CUSTOMER set AreaCode = AreaCode + 1
end
select * from CUSTOMER
```

Flow Control

IF_ELSE

```
Select GradeId, StudentId, CourseId,
case Grade
when 5 then 'A'
when 4 then 'B'
when 3 then 'C'
when 2 then 'D'
when 1 then 'E'
when 0 then 'F'
else '-'
end as Grade
from GRADE
```

Transact SQL

(T-SQL)

2 Transact SQL

Transact-SQL adalah bahasa pemrograman prosedural database, digunakan dalam SQL Server.

Bahasa prosedural dirancang untuk memperluas kemampuan SQL sekaligus dapat berintegrasi dengan baik dengan SQL.

Beberapa fitur seperti variabel lokal dan pemrosesan string / data ditambahkan. Fitur-fitur ini membuat bahasa Turing-lengkap, juga digunakan untuk menulis prosedur tersimpan: potongan kode yang berada di server untuk mengelola business-rule yang rumit atau tidak mungkin untuk dikelola dengan operasi berbasis set murni.

*A **Turing Complete** system means a system in which a program can be written that will find an answer (although with no guarantees regarding runtime or memory)*

T-SQL is organized by each block of statement. A block of statement can embrace another block of statement in it. A block of statement starts by BEGIN and finishes by END. There are many statements in the block, and statements is separated from each other by a semicolon (;).

```
BEGIN
   -- Declare variables
   -- T-SQL Statements
END;
```

2 Transact SQL

```
Begin
   -- Declaring a variable
  Declare @v_Result Int;
  -- Declaring a variable with a value of 50
  Declare @v_a Int = 50;
  -- Declaring a variable with a value of 100
  Declare @v_b Int = 100;

  -- Print out Console (For developer).
  -- Using Cast to convert Int to String
  -- Using + operator to concatenate 2 string
  Print 'v_a= ' + Cast(@v_a as varchar(15));
   -- Print out Console
  Print 'v_b= ' + Cast(@v_b as varchar(15));
   -- Sum
  Set @v_Result = @v_a + @v_b;
   -- Print out Console
  Print 'v_Result= ' + Cast(@v_Result as varchar(15));
End;
```

```
v_a = 50
v_b = 100
v_Result = 150
```

Selesai