Department of Computer Science
University of Cyprus

EPL342 –Databases

Lab 11

SQL-DML 4

(Stored Procedures, Cursors)

Panayiotis Andreou

http://www.cs.ucy.ac.cy/courses/EPL342

Before We Begin

- Start the SQL Server Management Studio
  - Start →
    All Programs →
    Microsoft SQL Server →
    SQL Server Management Studio

  Server: APOLLO

  Authentication: SQL Server Authentication

  Username: <your username>

  Password: <your password>

Hogwarts table

ID	Name	SID
1	Albus Dumbledore	NULL
2	Argus Filch	1
3	Filius Flitwick	1
4	Rubeus Hagrid	1
5	Madam Hooch	1
6	Gilderoy Lockhart	1
7	Minerva McGonagall	1
8	Severus Snape	1
9	Cedric Diggory	5
10	Harry Potter	7
11	Ron Weasly	7
12	Hermione Granger	7
13	Any Slytherin	8
14	Draco Malfoy	8
15	Fred Weasly	3
16	George Weasly	3

Hogwarts Cursor Example

```
DECLARE @ID int
DECLARE @Name nvarchar(100)

DECLARE c CURSOR FAST_FORWARD
FOR SELECT ID, Name FROM Hogwarts

OPEN c

FETCH NEXT FROM c INTO @ID, @Name
WHILE @@FETCH_STATUS=0
BEGIN
    --YOUR CODE HERE
    FETCH NEXT FROM c INTO @ID, @Name
END
CLOSE c
DEALLOCATE c
```

DECLARE: Variables for storing intermediate results

Specifies a FORWARD_ONLY, READ_ONLY cursor with performance optimizations enabled

OPEN: Initialize cursor and execute T-SQL statement

FETCH: Move cursor to the 1st record

WHILE: more records exist

FETCH: Move cursor to the next record

CLOSE: Release the current result set

DEALLOCATE: Removes the cursor reference and all associate data structures

Hogwarts Cursor Example

```
…
--YOUR CODE HERE
PRINT CAST(@ID as nvarchar) + ' ' + @Name
…
```

Execute the statement by pressing F5

1	Albus Dumbledore
2	Argus Filch
3	Filius Flitwick
4	Rubeus Hagrid
5	Madam Hooch
6	Gilderoy Lockhart
7	Minerva McGonagall
8	Severus Snape
9	Cedric Diggory
10	Harry Potter
11	Ron Weasly
12	Hermione Granger
13	Any Slytherin
14	Draco Malfoy
15	Fred Weasly
16	George Weasly

Recursive Procedures

How can we print the structure of the Hogwarts school?

1. There is one Headmaster
   Headmaster:Albus Dumbledore
2. There are many teachers who are supervised by the headmaster
   Argus Filch, Filius Flitwick, Rubeus Hagrid,…
3. There are many students who are supervised by the teachers
   (e.g., Minerva McGonagall: Harry Potter, Ron Weasly, Hermione Granger)

Headmaster:Albus Dumbledore
- Argus Filch
- Filius Flitwick
  - Fred Weasly
  - George Weasly
- Rubeus Hagrid
- Madam Hooch
  - Cedric Diggory
- Gilderoy Lockhart
- Minerva McGonagall
  - Harry Potter
  - Ron Weasly
  - Hermione Granger
- Severus Snape
  - Any Slytherin
  - Draco Malfoy

Recursive Procedures

How can we find the IDs and names of persons that are supervised by a person with ID=A?

```
SELECT    ID, Name
FROM      Hogwarts
WHERE     SID=A
```

Let's modify our cursor example to accept the SID as parameter and print the ID and names of all persons supervised by another person.

Hogwarts Tree

1. Create procedure [Hogwarts_Tree]
   - Input Parameters: @sid int
   - Output Parameters: <nothing>
   - Modify the cursor example to print the ID and name of all persors supervised by person with ID=@sid
   - Execute the procedure with @sid=1 and @sid=7

@sid=1

2 Argus Filch
3 Filius Flitwick
4 Rubeus Hagrid
5 Madam Hooch
6 Gilderoy Lockhart
7 Minerva McGonagall
8 Severus Snape

@sid=7

10 Harry Potter
11 Ron Weasly
12 Hermione Granger

Hogwarts Tree

Question: How can we extend the Hogwarts_Tree SP to print the persons that are supervised by each printed so far?

Answer: by calling the procedure with the @id of the person at the current cursor position

Include the following statement after

```
PRINT CAST(@ID as nvarchar) + ' ' + @Name
```

```
EXEC Hogwarts_Tree @ID
```

Execute the procedure with @sid=1

Hogwarts Tree

2 Argus Filch
Msg 16915, Level 16, State 1, Procedure hog, Line 9
A cursor with the name 'c' already exists.
Msg 16905, Level 16, State 1, Procedure hog, Line 11
The cursor is already open.
3 Filius Flitwick
…

Problem: Unlike common programming languages The **<c>** cursor's scope extends to the inner calls of the stored procedure

Answer: Declare the <c> cursor as LOCAL

DECLARE c CURSOR FAST_FORWARD →

DECLARE c CURSOR **LOCAL** FAST_FORWARD

Hogwarts Tree

- Execute the procedure with @sid=1 → **@sid=1**
- Notice that the order is correct (e.g., Harry Potter, Ron Weasly and Hermiony Granger are supervised by Minerva McGonnagall)
- Albus Dumbledore is not printed ☹
- We need to include some spaces to distinguish supervisors from supervisees
- One way to do that is to print spaces according to the level of recursion (e.g., Albus Dumbledore-1, Argus Filch-2, Harry Potter-3
- We can get the level of recursion easily using the @@NESTLEVEL

2 Argus Filch
3 Filius Flitwick
15 Fred Weasly
16 George Weasly
4 Rubeus Hagrid
5 Madam Hooch
9 Cedric Diggory
6 Gilderoy Lockhart
7 Minerva McGonagall
10 Harry Potter
11 Ron Weasly
12 Hermione Granger
8 Severus Snape
13 Any Slytherin
14 Draco Malfoy

Hogwarts Tree

- To print a number of spaces we can use the SPACE(int x) function (prints x spaces)
- Modify Hogwarts_Tree

```
PRINT CAST(@ID as nvarchar) + ' ' + @Name)
```

→

```
PRINT SPACE(@@NESTLEVEL * 2) + CAST(@ID as nvarchar) + ' ' + @Name
```

Execute the procedure with @sid=1 →

@sid=1

2 Argus Filch
3 Filius Flitwick
15 Fred Weasly
16 George Weasly
4 Rubeus Hagrid
5 Madam Hooch
9 Cedric Diggory
6 Gilderoy Lockhart
7 Minerva McGonagall
10 Harry Potter
11 Ron Weasly
12 Hermione Granger
8 Severus Snape
13 Any Slytherin
14 Draco Malfoy

Hogwarts Tree

Implement the following tasks

1. Extend the Hogwarts_Tree SP to print also the name of the person from the first call of the procedure (e.g., @sid=1→print Albus Dumbledore.
2. Extend the Hogwarts_Tree SP to save the records in **an existing table T (e.g., Results)**
3. Extend the Hogwarts_Tree SP return the results of T **ONLY FROM THE INITIAL** call of the procedure (i.e., @sid=1)
4. Extend the Hogwarts_Tree SP to use a temporary table instead of an already designed table

Northwind Stored Procedures

Implement the following SPs in Northwind

1. **[sp_update_Discount]**: Northwind has decided to award its customers with a 20% discount on the top 5 products that have sold more (largest quantities). Create a procedure to update the Discount column of table [Order Details] to do that (HINT: The largest quantities (300,200,100,90,87) may appear on more than one product).

Northwind Stored Procedures

2. **[sp_create_vouchers]**: Northwind customers who are affected by **sp_update_Discount** (i.e., have orders who have discount) should be issued vouchers on their orders. Create a procedure which print the vouchers for each Customer.

Template

Customer ID Customer Full Name

Order ID Order Date Old Amount Cur. Amount Voucher

Order ID Order Date Old Amount Cur. Amount Voucher

…

Customer ID Customer Full Name

Order ID Order Date Old Amount Cur. Amount Voucher

Order ID Order Date Old Amount Cur. Amount Voucher

…

Other Information

Setting multiple parameters with one SELECT statement

```
DECLARE @id int
DECLARE @name nvarchar(100)

SET @id = (SELECT ID FROM Table WHERE SID=1)
SET @name = (SELECT Name FROM Table WHERE SID=1)
```

OR

```
SELECT @id=ID, @name=Name FROM Table WHERE SID=1
```