(T36)討論Pagging。討論OffsetN1RowsFetchNextN2RowsOnly  
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc  
=======================================================================  
(T36)討論Pagging。討論OffsetN1RowsFetchNextN2RowsOnly  
=======================================================================  
0. Summary

1. Offset N1 Rows Fetch Next N2 Rows Only  
=======================================================================

0. Summary

OFFSET n1 ROWS

FETCH NEXT n2 ROWS ONLY

1.

OffsetFetchNext Syntax:

--SELECT  \*

--FROM    TableName

--ORDER BY C1,C2,...

--        OFFSET RowsToSkip ROWS

--FETCH NEXT RowsToFetch ROWS ONLY

ORDER BY clause is compulsory.

OffsetFetchNext is normally used in

returning a page/sub-set of results.

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

1.2.

E.g.

--SELECT  \*

--FROM    Book

--ORDER BY BookID

--        OFFSET 20 ROWS

--FETCH NEXT 10 ROWS ONLY

The 1st BookID is 1.

Offset 20 rows from BookID=1  will be BookID=21.

Start from BookID=21, fetch next 10 rows.

Thus, this will return from from ID=21 to ID=30

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

1.2.

spGetRowsByPageNumberAndSize receive

the PAGE NUMBER and the PAGE SIZE to get a page of rows.

E.g.

--IF ( EXISTS ( SELECT    \*

--              FROM      INFORMATION\_SCHEMA.ROUTINES

--              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

--                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

--                        AND SPECIFIC\_NAME = 'spGetRowsByPageNumberAndSize' ) )

--    BEGIN

--        DROP PROCEDURE spGetRowsByPageNumberAndSize;

--    END;

--GO -- Run the previous command and begins new batch

--CREATE PROCEDURE spGetRowsByPageNumberAndSize

--    (

--      @PageNumber INT ,

--      @PageSize INT

--    )

--AS

--    BEGIN

--        SELECT  \*

--        FROM    Book

--        ORDER BY BookID

--                OFFSET ( @PageNumber - 1 ) \* @PageSize ROWS

--    FETCH NEXT @PageSize ROWS ONLY;

--    END;

--GO -- Run the previous command and begins new batch

----Test it

--EXECUTE spGetRowsByPageNumberAndSize 4, 10;

--GO -- Run the previous command and begins new batch

=====================================================

1. Offset N1 Rows Fetch Next N2 Rows Only

-- T036\_OffsetN1RowsFetchNextN2RowsOnly -------------------------------------

/\*

1.

OffsetFetchNext Syntax:

--SELECT  \*

--FROM    TableName

--ORDER BY C1,C2,...

--        OFFSET RowsToSkip ROWS

--FETCH NEXT RowsToFetch ROWS ONLY

ORDER BY clause is compulsory.

OffsetFetchNext is normally used in

returning a page/sub-set of results.

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

1.2.

E.g.

--SELECT  \*

--FROM    Book

--ORDER BY BookID

--        OFFSET 20 ROWS

--FETCH NEXT 10 ROWS ONLY

The 1st BookID is 1.

Offset 20 rows from BookID=1  will be BookID=21.

Start from BookID=21, fetch next 10 rows.

Thus, this will return from from ID=21 to ID=30

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

1.2.

spGetRowsByPageNumberAndSize receive

the PAGE NUMBER and the PAGE SIZE to get a page of rows.

E.g.

--IF ( EXISTS ( SELECT    \*

--              FROM      INFORMATION\_SCHEMA.ROUTINES

--              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

--                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

--                        AND SPECIFIC\_NAME = 'spGetRowsByPageNumberAndSize' ) )

--    BEGIN

--        DROP PROCEDURE spGetRowsByPageNumberAndSize;

--    END;

--GO -- Run the previous command and begins new batch

--CREATE PROCEDURE spGetRowsByPageNumberAndSize

--    (

--      @PageNumber INT ,

--      @PageSize INT

--    )

--AS

--    BEGIN

--        SELECT  \*

--        FROM    Book

--        ORDER BY BookID

--                OFFSET ( @PageNumber - 1 ) \* @PageSize ROWS

--    FETCH NEXT @PageSize ROWS ONLY;

--    END;

--GO -- Run the previous command and begins new batch

----Test it

--EXECUTE spGetRowsByPageNumberAndSize 4, 10;

--GO -- Run the previous command and begins new batch

\*/

--=====================================================================

--T036\_01

--Create Sample Data

--Revise Ch61\_PerformanceTesting - Create large amount of test data

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.TABLES

              WHERE     TABLE\_NAME = 'Book' ) )

    BEGIN

        DROP TABLE Book;

    END;

GO -- Run the previous command and begins new batch

CREATE TABLE Book

    (

      BookID INT PRIMARY KEY

                 IDENTITY(1, 1)

                 NOT NULL ,

      BookName NVARCHAR(100) NULL ,

      BookUnitPrice MONEY NULL ,

      [Description] NVARCHAR(1000) NULL,

    )

ON  [PRIMARY];

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

--Insert sample data to Book table

--Book Counter

DECLARE @TotalBookRows INT = 100;

DECLARE @BookCount INT = 1;

-- random UnitPrice between 1 and 100

DECLARE @RandomUnitPrice MONEY;

DECLARE @BookUnitPrice\_Max INT = 100;

DECLARE @BookUnitPrice\_Min INT = 1;

--Loop

WHILE ( @BookCount <= @TotalBookRows )

    BEGIN

        SELECT  @RandomUnitPrice = FLOOR(RAND() \* ( @BookUnitPrice\_Max

                                                    - @BookUnitPrice\_Min )

                                         + @BookUnitPrice\_Min);

        INSERT  INTO Book

        VALUES  ( 'Book ' + CAST(@BookCount AS NVARCHAR(20)), @RandomUnitPrice,

                  'Book Description ' + CAST(@BookCount AS NVARCHAR(20)) );

        PRINT @BookCount;

        SET @BookCount += 1;

    END;

GO -- Run the previous command and begins new batch

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

SELECT  \*

FROM    Book;

GO -- Run the previous command and begins new batch

Table

Description automatically generated

--=====================================================================

--T036\_02

--OFFSET n1 ROWS

--FETCH NEXT n2 ROWS ONLY

--Return from ID=21 to ID=30

SELECT  \*

FROM    Book

ORDER BY BookID

        OFFSET 20 ROWS

FETCH NEXT 10 ROWS ONLY;

GO -- Run the previous command and begins new batch

/\*

1.

OffsetFetchNext Syntax:

--SELECT  \*

--FROM    TableName

--ORDER BY C1,C2,...

--        OFFSET RowsToSkip ROWS

--FETCH NEXT RowsToFetch ROWS ONLY

ORDER BY clause is compulsory.

OffsetFetchNext is normally used in

returning a page/sub-set of results.

1.2.

E.g.

--SELECT  \*

--FROM    Book

--ORDER BY BookID

--        OFFSET 20 ROWS

--FETCH NEXT 10 ROWS ONLY

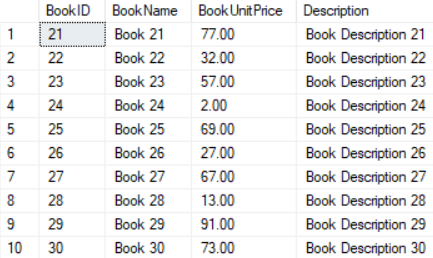
The 1st BookID is 1.

Offset 20 rows from BookID=1  will be BookID=21.

Start from BookID=21, fetch next 10 rows.

Thus, this will return from from ID=21 to ID=30

\*/



--=====================================================================

--T036\_03

--OFFSET n1 ROWS

--FETCH NEXT n2 ROWS ONLY

--Drop Store Procedure exists then DROP it

IF ( EXISTS ( SELECT    \*

              FROM      INFORMATION\_SCHEMA.ROUTINES

              WHERE     ROUTINE\_TYPE = 'PROCEDURE'

                        AND LEFT(ROUTINE\_NAME, 3) NOT IN ( 'sp\_', 'xp\_', 'ms\_' )

                        AND SPECIFIC\_NAME = 'spGetRowsByPageNumberAndSize' ) )

    BEGIN

        DROP PROCEDURE spGetRowsByPageNumberAndSize;

    END;

GO -- Run the previous command and begins new batch

CREATE PROCEDURE spGetRowsByPageNumberAndSize

    (

      @PageNumber INT ,

      @PageSize INT

    )

AS

    BEGIN

        SELECT  \*

        FROM    Book

        ORDER BY BookID

                OFFSET ( @PageNumber - 1 ) \* @PageSize ROWS

    FETCH NEXT @PageSize ROWS ONLY;

    END;

GO -- Run the previous command and begins new batch

--Test it

EXECUTE spGetRowsByPageNumberAndSize 4, 10;

GO -- Run the previous command and begins new batch

/\*

1.

--ORDER BY BookID

--OFFSET ( @PageNumber - 1 ) \* @PageSize ROWS

--FETCH NEXT @PageSize ROWS ONLY;

spGetRowsByPageNumberAndSize receive

the PAGE NUMBER and the PAGE SIZE to get a page of rows.

The table has 100 rows and Id is from 1 to 100.

1.1.

If @PageNumber=1, @PageSize=10,

then Page 1 will show the first top 10 rows, ID=1 to ID=10,

which means OFFSET 0 FETCH NEXT 10.

1.2.

If @PageNumber=2, @PageSize=10,

then Page 2 will show the second top 10 rows, ID=11 to ID=20

which means OFFSET 10 FETCH NEXT 10.

1.3.

If @PageNumber=3, @PageSize=10,

then Page 3 will show the third top 10 rows, ID=21 to ID=30

which means OFFSET 20 FETCH NEXT 10.

\*/

