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

==========

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

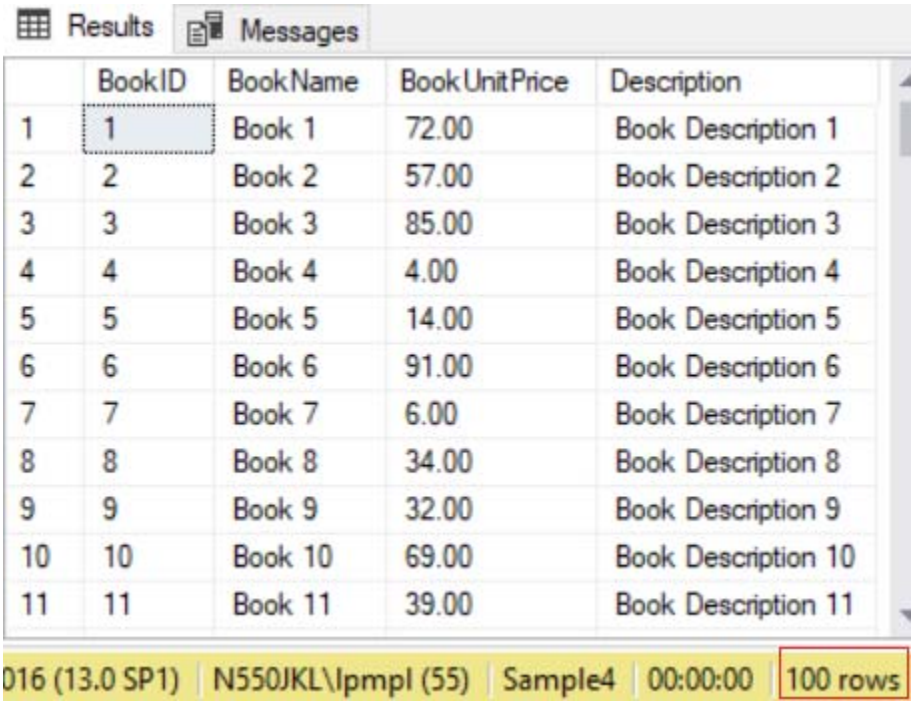
```
-- T036_OffsetN1RowsFetchNextN2RowsOnly ------------------------------------
/*
1.
OffsetFetchNext Syntax:
--SELECT  *
--FROM    TableName
--ORDER BY C1,C2,...
--        OFFSET RowsToSkip ROWS
--FETCH NEXT RowsToFetch ROWS ONLY
ORDER BY clause is compulsory.
OffsetFetchNext is normally used in
returning 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
```



Results | Messages

	BookID	BookName	BookUnitPrice	Description
1	1	Book 1	72.00	Book Description 1
2	2	Book 2	57.00	Book Description 2
3	3	Book 3	85.00	Book Description 3
4	4	Book 4	4.00	Book Description 4
5	5	Book 5	14.00	Book Description 5
6	6	Book 6	91.00	Book Description 6
7	7	Book 7	6.00	Book Description 7
8	8	Book 8	34.00	Book Description 8
9	9	Book 9	32.00	Book Description 9
10	10	Book 10	69.00	Book Description 10
11	11	Book 11	39.00	Book Description 11

016 (13.0 SP1) | N550JKL\Ipmpl (55) | Sample4 | 00:00:00 | 100 rows

```
--=========================================================================
--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
*/
```

	BookID	BookName	BookUnitPrice	Description
1	21	Book 21	77.00	Book Description 21
2	22	Book 22	32.00	Book Description 22
3	23	Book 23	57.00	Book Description 23
4	24	Book 24	2.00	Book Description 24
5	25	Book 25	69.00	Book Description 25
6	26	Book 26	27.00	Book Description 26
7	27	Book 27	67.00	Book Description 27
8	28	Book 28	13.00	Book Description 28
9	29	Book 29	91.00	Book Description 29
10	30	Book 30	73.00	Book Description 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.
*/
```

	BookID	BookName	BookUnitPrice	Description
1	31	Book 31	84.00	Book Description 31
2	32	Book 32	66.00	Book Description 32
3	33	Book 33	19.00	Book Description 33
4	34	Book 34	37.00	Book Description 34
5	35	Book 35	67.00	Book Description 35
6	36	Book 36	12.00	Book Description 36
7	37	Book 37	13.00	Book Description 37
8	38	Book 38	95.00	Book Description 38
9	39	Book 39	53.00	Book Description 39
10	40	Book 40	20.00	Book Description 40