(T28)討論 CrossApply 和 OuterApply
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc
==========
(T28)討論 CrossApply 和 OuterApply
==========

==========

0. Summary

1.
INNER JOIN V.S. CROSS APPLY
----Table Value Function must use CROSS APPLY
```
--SELECT  d.DepartmentName ,
--       p.[Name] ,
--       p.Gender ,
--       p.Salary
--FROM    DepartmentA d
--       CROSS APPLY fn_GetPersonAByDepartmentAID(d.ID) p
--ORDER BY d.ID;
--GO -- Run the previous command and begins new batch
```
1.1.
```
--FROM    DepartmentA d
--       CROSS APPLY fn_GetPersonAByDepartmentAID(d.ID) p
```
Pass each DepartmentID into fn_GetPersonAByDepartmentAID()
This will return all the Persons who has Department.
Thus, fn_GetPersonAByDepartmentAID() CROSS APPLY DepartmentA
will return all the Persons with their DepartmentName.
1.2.
```
--TableA INNER JOIN TableB
--ON TableA.ColumnAB = TableB.ColumnAB
```
INNER JOIN is for join 2 tables.
1.3.
```
--fnTableValueFunction CROSS APPLY TableA
```
This will cause ERROR,
fnTableValueFunction must be the right hand side of CROSS APPLY
1.4.
```
--TableA CROSS APPLY fnTableValueFunction
```
fnTableValueFunction must be the right hand side of CROSS APPLY
CROSS APPLY is similar to INNER JOIN
which retrieves only the matching rows.
However,
INNER JOIN is for join 2 tables.
CROSS APPLY is join 1 table(Left Hand Side)
and fnTableValueFunction(Right Hand Side).
fnTableValueFunction can not use INNER JOIN

2.
LEFT JOIN V.S. OUTER APPLY
--SELECT d.DepartmentName ,
-- p.[Name] ,
-- p.Gender ,
-- p.Salary
--FROM DepartmentA d
-- OUTER APPLY fn_GetPersonAByDepartmentAID(d.ID) p
--ORDER BY d.ID;
--GO -- Run the previous command and begins new batch
2.1.
--FROM DepartmentA d
-- OUTER APPLY fn_GetPersonAByDepartmentAID(d.ID) p
Pass each DepartmentID into fn_GetPersonAByDepartmentAID()
This will return all the Persons who has Department.
DepartmentA d is in Left Hand Side of OUTER APPLY.
Thus, the query will return
all the Persons with their DepartmentName
plus all departments name which has no persons.
2.2.
--TableA LEFT JOIN TableB
--ON TableA.ColumnAB = TableB.ColumnAB
LEFT JOIN is for join 2 tables.
2.3.
--fnTableValueFunction OUTER APPLY TableA
This will cause ERROR,
fnTableValueFunction must be the right hand side of OUTER APPLY
2.4.
--TableA OUTER APPLY fnTableValueFunction
fnTableValueFunction must be the right hand side of OUTER APPLY
OUTER APPLY is similar to LEFT JOIN
which retrieves only the matching rows + Left Hand Side un-matching rows
However,
LEFT JOIN is for join 2 tables.
OUTER APPLY is join 1 table(Left Hand Side)
and fnTableValueFunction(Right Hand Side).
fnTableValueFunction can not use LEFT JOIN

==========

1. Create Sample Data

```
--=============================================================================
--T028_01_Create Sample Data
--=============================================================================
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'PersonA' ) )
    BEGIN
        TRUNCATE TABLE dbo.PersonA;
        DROP TABLE PersonA;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'DepartmentA' ) )
    BEGIN
        TRUNCATE TABLE dbo.DepartmentA;
        DROP TABLE DepartmentA;
    END;
```

```
GO -- Run the previous command and begins new batch
CREATE TABLE DepartmentA
    (
       ID INT PRIMARY KEY ,
       DepartmentName NVARCHAR(50)
    );
GO -- Run the previous command and begins new batch
INSERT  INTO DepartmentA
VALUES  ( 1, 'Department01' );
INSERT  INTO DepartmentA
VALUES  ( 2, 'Department02' );
INSERT  INTO DepartmentA
VALUES  ( 3, 'Department03' );
INSERT  INTO DepartmentA
VALUES  ( 4, 'Department04' );
INSERT  INTO DepartmentA
VALUES  ( 5, 'Department05' );
GO -- Run the previous command and begins new batch
CREATE TABLE PersonA
    (
       ID INT PRIMARY KEY ,
       [Name] NVARCHAR(50) ,
       Gender NVARCHAR(10) ,
       Salary MONEY ,
       DepartmentID INT FOREIGN KEY REFERENCES DepartmentA ( ID )
    );
GO -- Run the previous command and begins new batch
INSERT  INTO PersonA
VALUES  ( 1, 'Name01', 'Male', 41000, 1 );
INSERT  INTO PersonA
VALUES  ( 2, 'Name02', 'Female', 75000, 3 );
INSERT  INTO PersonA
VALUES  ( 3, 'Name03', 'Female', 65000, 2 );
INSERT  INTO PersonA
VALUES  ( 4, 'Name04', 'Female', 44000, 3 );
INSERT  INTO PersonA
VALUES  ( 5, 'Name05', 'Male', 38000, 1 );
GO -- Run the previous command and begins new batch
SELECT  *
FROM    dbo.PersonA;
SELECT  *
FROM    dbo.DepartmentA;
GO -- Run the previous command and begins new batch
```

	ID	Name	Gender	Salary	DepartmentID
1	1	Name01	Male	41000.00	1
2	2	Name02	Female	75000.00	3
3	3	Name03	Female	65000.00	2
4	4	Name04	Female	44000.00	3
5	5	Name05	Male	38000.00	1

	ID	DepartmentName
1	1	Department01
2	2	Department02
3	3	Department03
4	4	Department04
5	5	Department05

==========

2. InnerJoin V.S. LeftJoin

```
--============================================================================
--T028_02_InnerJoin V.S. LeftJoin
--============================================================================
--============================================================================
--T028_02_01
--Department INNER JOIN Person
SELECT  d.DepartmentName ,
        p.Name ,
        p.Gender ,
        p.Salary
FROM    DepartmentA d
        INNER JOIN dbo.PersonA p ON p.DepartmentID = d.ID
ORDER BY d.ID;
GO -- Run the previous command and begins new batch
```

	DepartmentName	Name	Gender	Salary
1	Department01	Name01	Male	41000.00
2	Department01	Name05	Male	38000.00
3	Department02	Name03	Female	65000.00
4	Department03	Name04	Female	44000.00
5	Department03	Name02	Female	75000.00

```
--============================================================================
--T028_02_02
--Department LEFT JOIN Person
SELECT  d.DepartmentName ,
        p.Name ,
        p.Gender ,
        p.Salary
FROM    DepartmentA d
        LEFT JOIN dbo.PersonA p ON p.DepartmentID = d.ID
ORDER BY d.ID;
GO -- Run the previous command and begins new batch
```

	DepartmentName	Name	Gender	Salary
1	Department01	Name01	Male	41000.00
2	Department01	Name05	Male	38000.00
3	Department02	Name03	Female	65000.00
4	Department03	Name02	Female	75000.00
5	Department03	Name04	Female	44000.00
6	Department04	NULL	NULL	NULL
7	Department05	NULL	NULL	NULL

==========

3. TableValueFunction need Cross Apply and Outter Apply instead of Join

```
--============================================================================
--T028_03_TableValueFunction need Cross Apply and Outter Apply instead of Join
--============================================================================
```

3.1. Create Table Value Function, fn_GetPersonAByDepartmentAID

```
--============================================================================
--T028_03_01
--Create Table Value Function, fn_GetPersonAByDepartmentAID
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'FUNCTION'
                        AND LEFT(ROUTINE_NAME, 2) NOT IN ( '@@' )
                        AND SPECIFIC_NAME = 'fn_GetPersonAByDepartmentAID' ) )
    BEGIN
        DROP FUNCTION fn_GetPersonAByDepartmentAID;
    END;
GO -- Run the previous command and begins new batch
CREATE FUNCTION fn_GetPersonAByDepartmentAID ( @DepartmentAID int )
RETURNS TABLE
AS
RETURN
    ( SELECT    p.ID ,
                p.[Name] ,
                p.Gender ,
                p.Salary ,
                p.DepartmentID
      FROM      dbo.PersonA p
      WHERE     DepartmentID = @DepartmentAID
    );
GO -- Run the previous command and begins new batch
SELECT  *
FROM    fn_GetPersonAByDepartmentAID(1);
GO -- Run the previous command and begins new batch
```

	ID	Name	Gender	Salary	DepartmentID
1	1	Name01	Male	41000.00	1
2	5	Name05	Male	38000.00	1

3.2. Table Value Function can not use INNER JOIN

```
--=============================================================================
--T028_03_02
--Table Value Function can not use INNER JOIN
SELECT  d.DepartmentName ,
        p.[Name] ,
        p.Gender ,
        p.Salary
FROM    DepartmentA d
       INNER JOIN fn_GetPersonAByDepartmentAID(d.ID) p ON d.ID = p.DepartmentID;
GO -- Run the previous command and begins new batch
/*
Error Message
--Msg 4104, Level 16, State 1, Line 133
--The multi-part identifier "d.ID" could not be bound.
Table Value Function can not use INNER JOIN
*/
```

Messages

```
Msg 4104, Level 16, State 1, Line 217
The multi-part identifier "d.ID" could not be bound.
```

3.3. Table Value Function must use CROSS APPLY

```
--=============================================================================
--T028_03_03
--Table Value Function must use CROSS APPLY
SELECT  d.DepartmentName ,
        p.[Name] ,
        p.Gender ,
        p.Salary
FROM    DepartmentA d
       CROSS APPLY fn_GetPersonAByDepartmentAID(d.ID) p
ORDER BY d.ID;
GO -- Run the previous command and begins new batch
/*
1.
--FROM    DepartmentA d
--        CROSS APPLY fn_GetPersonAByDepartmentAID(d.ID) p
Pass each DepartmentID into fn_GetPersonAByDepartmentAID()
This will return all the Persons who has Department.
Thus, fn_GetPersonAByDepartmentAID() CROSS APPLY DepartmentA
will return all the Persons with their DepartmentName.
1.1.
--TableA INNER JOIN TableB
--ON TableA.ColumnAB = TableB.ColumnAB
INNER JOIN is for join 2 tables.
1.2.
--fnTableValueFunction CROSS APPLY TableA
This will cause ERROR,
fnTableValueFunction must be the right hand side of CROSS APPLY
1.3.
--TableA CROSS APPLY fnTableValueFunction
fnTableValueFunction must be the right hand side of CROSS APPLY
CROSS APPLY is similar to INNER JOIN
which retrieves only the matching rows.
However,
INNER JOIN is for join 2 tables.
CROSS APPLY is join 1 table(Left Hand Side)
and fnTableValueFunction(Right Hand Side).
fnTableValueFunction can not use INNER JOIN
*/
```

	DepartmentName	Name	Gender	Salary
1	Department01	Name01	Male	41000.00
2	Department01	Name05	Male	38000.00
3	Department02	Name03	Female	65000.00
4	Department03	Name04	Female	44000.00
5	Department03	Name02	Female	75000.00

3.4. fnTableValueFunction must be the right hand side of CROSS APPLY

```
--=============================================================================
--T028_03_04
--ERROR: fnTableValueFunction must be the right hand side of CROSS APPLY
SELECT  d.DepartmentName ,
        p.[Name] ,
        p.Gender ,
        p.Salary
FROM    fn_GetPersonAByDepartmentAID(d.ID) p
       CROSS APPLY DepartmentA d
ORDER BY d.ID;
GO -- Run the previous command and begins new batch
/*
1.
--fnTableValueFunction CROSS APPLY TableA
This will cause ERROR,
fnTableValueFunction must be the right hand side of CROSS APPLY
2.
Output
--Msg 4104, Level 16, State 1, Line 186
--The multi-part identifier "d.ID" could not be bound.
*/
```

Messages

```
Msg 4104, Level 16, State 1, Line 272
The multi-part identifier "d.ID" could not be bound.
```

3.5. Table Value Function must use OUTER APPLY

```
--=============================================================================
--T028_03_05
--Table Value Function must use OUTER APPLY
SELECT  d.DepartmentName ,
        p.[Name] ,
        p.Gender ,
        p.Salary
FROM    DepartmentA d
       OUTER APPLY fn_GetPersonAByDepartmentAID(d.ID) p
ORDER BY d.ID;
GO -- Run the previous command and begins new batch
```

	DepartmentName	Name	Gender	Salary
1	Department01	Name01	Male	41000.00
2	Department01	Name05	Male	38000.00
3	Department02	Name03	Female	65000.00
4	Department03	Name02	Female	75000.00
5	Department03	Name04	Female	44000.00
6	Department04	NULL	NULL	NULL
7	Department05	NULL	NULL	NULL

```
/*
1.
--FROM    DepartmentA d
--        OUTER APPLY fn_GetPersonAByDepartmentAID(d.ID) p
Pass each DepartmentID into fn_GetPersonAByDepartmentAID()
This will return all the Persons who has Department.
DepartmentA d is in Left Hand Side of OUTER APPLY.
Thus, the query will return
all the Persons with their DepartmentName
plus all departments name which has no persons.
1.1.
--TableA LEFT JOIN TableB
--ON TableA.ColumnAB = TableB.ColumnAB
LEFT JOIN is for join 2 tables.
1.2.
--fnTableValueFunction OUTER APPLY TableA
This will cause ERROR,
fnTableValueFunction must be the right hand side of OUTER APPLY
1.3.
--TableA OUTER APPLY fnTableValueFunction
fnTableValueFunction must be the right hand side of OUTER APPLY
OUTER APPLY is similar to LEFT JOIN
which retrieves only the matching rows + Left Hand Side un-matching rows
However,
LEFT JOIN is for join 2 tables.
OUTER APPLY is join 1 table(Left Hand Side)
and fnTableValueFunction(Right Hand Side).
fnTableValueFunction can not use LEFT JOIN
*/

==================================================
```

4. Clean up

```
--=====================================================================
--T028_04_Clean up
--=====================================================================
--Drop Function if it exists.
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'FUNCTION'
                        AND LEFT(ROUTINE_NAME, 2) NOT IN ( '@@' )
                        AND SPECIFIC_NAME = 'fn_GetPersonAByDepartmentAID' ) )
    BEGIN
        DROP FUNCTION fn_GetPersonAByDepartmentAID;
    END;
GO -- Run the previous command and begins new batch
---------------------------
--Drop Table if it exists.
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
```

```
            WHERE      TABLE_NAME = 'PersonA' ) )
    BEGIN
        TRUNCATE TABLE dbo.PersonA;
        DROP TABLE PersonA;
    END;
GO -- Run the previous command and begins new batch
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE      TABLE_NAME = 'DepartmentA' ) )
    BEGIN
        TRUNCATE TABLE dbo.DepartmentA;
        DROP TABLE DepartmentA;
    END;
GO -- Run the previous command and begins new batch
```