CIS560

Table Expressions – Part 2

Table Expressions - Review

- Named query expressions that represent a valid relational table.
- They are not materialized – they are virtual.
  References to a table expression are internally translated to a query against underlying objects.
- Benefits are for logical aspects of your code.
  - Code structure/readability.
  - *Possibly* code re-use.
  - Rarely used for optimization!

KANSAS STATE UNIVERSITY | Computer Science

Views & Inline TVFs

- Derived tables and CTEs have single-statement scope.
- Views & Inline Table-Valued Functions (TVFs) have broader scope.
  - Stored as database objects
  - Available until explicitly dropped
- You can think of an Inline TVF as a parameterized view.
- Remember Views and Inline TVFs are logical constructs only.

APPLY Operator

- Nonstandard table operator in SQL Server
  Standard defines LATERAL joins
- Operates on two tables, the second of which may be a table expression
- Allows correlated derived table expressions
  Joins only allow self-contained

APPLY Operator

- Logical processing phases
  - CROSS APPLY
    1) Applies the right table expression to each row from the left table
  - OUTER APPLY
    1) Applies the right table expression to each row from the left table
    2) Adds outer rows from the left table for which an empty set resulted from the expression on the right
- APPLY with TOP provides two benefits over other subquery approaches:
  - Multiple rows (TOP N where N > 1) are needed for each row in the outer query.
  - Need multiple columns even if only a single row is needed. Otherwise a scalar subquery can be used in the SELECT clause.

Syntax

Views

```
CREATE [ OR ALTER ] VIEW schema_name.view_name [ (column [ ,...n ] ) ]
AS
<select_statement>;
```

Inline Table-Valued Function

```
CREATE [ OR ALTER ] FUNCTION schema_name.function_name
(
   [ { @parameter_name [ AS ] [ type_schema_name. ] data_type [ = default ] [ READONLY ] }
   [ ,...n ] ]
)
RETURNS TABLE
[ AS ]

RETURN
   (
      select_statement
   );
```

KANSAS STATE UNIVERSITY | Computer Science

Syntax

APPLY Operator

```
<table_source> { CROSS | OUTER } APPLY <right_table_source>

<table_source>:=
   table | table_expression

<right_table_source>::=
   table | table_expression | correlated_table_expression
```