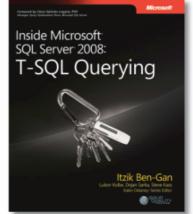
## LOGICAL QUERY PROCESSING Start **Entering FROM** First Table Operator Exist? Operator Type? APPLY UNPIVOT JOIN **PIVOT** 1-P1 artesian Produc Generate Copies OUTER 1-P2 ON Filter Add Outer Rows **Extract Element** Spread No 1-P3 1-J3 Add Outer Rows Remove NULLs Aggregate Another Yes Table Operator Exists? No Yes WHERE Exists? • (1) FROM The FROM phase identifies the query's source tables and processes table operators. Each table operator applies a series of WHERE subphases. For example, the phases involved in a join are: (1-J1) Cartesian Product, (1-J2) ON Filter, (1-J3) Add Outer Rows. The FROM phase No generates virtual table VT1. Yes • (1-J1) Cartesian Product This phase performs a Cartesian product (cross **GROUP BY** join) between the two tables involved in the table operator, generating Exists? **GROUP BY** • (1-J2) ON Filter This phase filters the rows from VT1-J1 based on the No predicate that appears in the ON clause (<on\_predicate>). Only rows for which the predicate evaluates to TRUE are inserted into VT1-J2. • (1-J3) Add Outer Rows If OUTER JOIN is specified (as opposed to CROSS **HAVING Exists?** JOIN or INNER JOIN), rows from the preserved table or tables for which a HAVING match was not found are added to the rows from VT1-J2 as outer rows, generating VT1-J3. No • (2) WHERE This phase filters the rows from VT1 based on the predicate that appears in the WHERE clause (<where\_predicate>). Only rows for which **Entering SELECT** the predicate evaluates to TRUE are inserted into VT2. • (3) GROUP BY This phase arranges the rows from VT2 in groups based on the column list specified in the GROUP BY clause, generating VT3. 5-1 Ultimately, there will be one result row per each group. **Evaluate Expressions** • (4) HAVING This phase filters the groups from VT3 based on the predicate that appears in the HAVING clause (< having\_predicate>). Only groups for Yes which the predicate evaluates to TRUE are inserted into VT4. • (5) SELECT This phase processes the elements in the SELECT clause, **DISTINCT Exists?** 5-2 generating VT5. DISTINCT • (5-1) Evaluate Expressions This phase evaluates the expressions in No the SELECT list, generating VT5-1 Available in bookstores now Yes • (5-2) DISTINCT This phase TOP Exists? removes duplicate rows from VT5-1, 5-3 generating VT5-2. TOP Inside Microsoft • (5-3) TOP This phase filters the No SQL Server 2008: specified top number or percentage of rows from VT5-2 based on the Yes logical ordering defined by the ORDER BY ORDER BY clause, generating the Exists? table VT5-3. ORDER BY • (6) ORDER BY This phase sorts the No rows from VT5-3 according to the column list specified in the ORDER Set Curso

1-U1

1-U2

1-U3



BY clause, generating the cursor

VC6.

End