

Module 5:

Windowing and the OVER Clause

Windows on Data

- **OVER Clause**

- Added in SQL Server 2005, along with the ranking functions ROW_NUMBER, RANK, DENSE_RANK, NTILE

- **Can also be used with aggregate functions**

- SUM, COUNT, MIN, MAX, AVG
 - Doesn't require GROUP BY

- **Exposes a window over the result set for each row**

- SQL Server 2012 adds
 - Running aggregations
 - Sliding aggregations
 - Row lookups
 - Percentiles

OVER

AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125

AcctId	TxnDate	Amount
1	3/10/2012	500
1	3/22/2012	250
1	3/24/2012	75
1	3/26/2012	125
2	3/11/2012	500
2	3/15/2012	50
2	3/22/2012	5000
2	3/24/2012	550
2	3/27/2012	95
3	3/15/2012	600
3	3/22/2012	25
3	3/23/2012	125

Module 6:

Additional T-SQL Language Enhancements

New T-SQL Features in SQL Server 2012

- **New Functions**
- **THROW**
- **Server-Side Paging**
- **Sequences**

New T-SQL Functions

- **Logical**

- CHOOSE
- IIF

- **String**

- CONCAT
- FORMAT

- **Conversion**

- TRY_CONVERT
- PARSE
- TRY_PARSE

- **Date/Time**

- DATEFROMPARTS
- TIMEFROMPARTS
- DATETIME2FROMPARTS
- DATETIMEOFFSETFROMPARTS
- DATETIMEFROMPARTS
- SMALLDATETIMEFROMPARTS
- EOMONTH

- **Math**

- LOG (changed)

Demo

- New T-SQL Functions

The THROW Statement

- **SQL Server 2005 added TRY/CATCH**
 - Borrowed from .NET's try/catch model
 - Vast improvement over repeatedly testing @@ERROR
 - Still used RAISERROR for generating errors
- **SQL Server 2012 adds THROW**
 - Recommended alternative way to generate your own errors
- **Two usages for THROW**
 - With error code, description, and state parameters (like RAISERROR)
 - Inside a CATCH block with no parameters (re-throw)

Demo

- The THROW Statement

Server-Side Paging

- **New result paging keywords**
 - Append to ORDER BY clause
 - Limits the number of rows returned
- **OFFSET @Start ROWS**
 - The first result row to return (zero-based)
- **FETCH NEXT @Count ROWS**
 - The number of rows to return

Demo

- **Server-Side Paging**

Sequences

- **Sequential Number Generator**
 - As found in Oracle and DB2
 - Alternative to using IDENTITY for assigning new primary keys
- **Advantages over IDENTITY**
 - Can obtain next sequence value without performing an insert
 - SET IDENTITY INSERT ON/OFF not needed for inserts
- **Create a Sequence Object**
 - CREATE SEQUENCE MySequence
START WITH 1
INCREMENT 1
NO MAXVALUE
- **Retrieve Sequence Values**
 - INSERT INTO MyTable(Id, ...)
VALUES(NEXT VALUE FOR MySequence, ...)

Demo

- Sequences

Sequence Limitations

- **Cannot be used in**
 - Subqueries
 - Common Table Expressions (CTEs)
 - The TOP clause
 - CHECK CONSTRAINT definition
 - Aggregate function arguments
 - Views, computed columns, user-defined functions
- **Must drop DEFAULT constraints before dropping the sequence**

End of Module 6

Exploring SQL Server Tools and Language Enhancements

End of Course