

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# SQL Server2005 中的T-SQL增强

宋立桓  
微软最有价值专家  
[slhhacker@21cn.com](mailto:slhhacker@21cn.com)

# 议程

- Richer Data Types( varchar(max)、 xml data type)
- ERROR HANDLING
- SNAPSHOT
- TOP
- OUTPUT
- New functions: ROW\_NUMBER, RANK,DENSE\_RANK
- CTE
- PIVOT/UNPIVOT
- APPLY
- DLL TRIGGER

# Richer Data Types

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **varchar(max).nvarchar(max) 和 varbinary(max)** 数据类型最多可保存 2 GB 的数据，可取代 **text**、**ntext** 或 **image** 数据类型。

- Uses MAX size specifier

```
CREATE TABLE myTable  
(id INT,  
content VARCHAR(MAX))
```

- **XML 数据类型**
  - **XML** 数据类型允许用户在 **SQL Server** 数据库中保存 **XML** 片断或文档。

# Error Handling 错误处理

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 新的异常处理结构
- 可以捕获和处理过去会导致批处理终止的错误。
  - 前提是这些错误不会导致连接中断（通常是严重度为 21 及以上的错误，例如，表或数据库完整性可疑、硬件错误等等）。

- **TRY/CATCH construct**

```
SET XACT_ABORT ON
```

```
BEGIN TRY
```

```
    <core logic>
```

```
END TRY
```

```
BEGIN CATCH TRAN_ABORT
```

```
    <exception handling logic>
```

```
END CATCH
```

- @@error may be queried as first statement in CATCH block



您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## Error Handling

# Snapshot Isolation 快照隔离

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 写入程序不会阻碍读取程序
- **Snapshot isolation must be enabled for DB**
  - Alter database 数据库 set allow\_snapshot\_isolation on
- **Snapshot isolation must be set for connection**
  - Set transaction isolation level snapshot
- **UPDATE transactions keep old versions of data in a linked list**
- 新的隔离级别提供了以下优点：
  - 1 提高了只读应用程序的数据可用性
  - 2 允许在 **OLTP** 环境中执行非阻止读取操作
  - 3 可对写入事务进行自动的强制冲突检测

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## SNAPSHOT ISOLATION

# TOP增强功能

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **TOP 增强:**
  - 可以指定一个数字表达式，以返回要通过查询影响的行数或百分比，还可以根据情况使用变量和子查询。
  - 可以在 **DELETE**、**UPDATE** 和 **INSERT** 查询中使用 **TOP** 选项。
- 更好地替代**SET ROWCOUNT**选项，使之更为高效



# OUTPUT

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- SQL Server 2005 引入了一个新的 **OUTPUT** 子句，以使您可以从修改语句（**INSERT**、**UPDATE**、**DELETE**）中将数据返回到表变量中。
- 新的 **OUTPUT** 子句的语法为：
  - **OUTPUT <dml\_select\_list> INTO @table\_variable**  
可以通过引用插入的表和删除的表来访问被修改的行的旧/新映像，其方式与访问触发器类似。在 **INSERT** 语句中，只能访问插入的表。在 **DELETE** 语句中，只能访问删除的表。在 **UPDATE** 语句中，可以访问插入的表和删除的表。

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## OUTPUT、TOP



**MSDN Webcasts**

# Ranking Functions

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- SQL Server 2005 引入了几个新的排序函数：如 **ROW NUMBER**、**RANK**、**DENSE RANK** 等。这些新函数使您可以有效地分析数据以及向查询的结果行提供排序值。
- 排序函数都遵循类似的语法模式：
  - 排序函数  
( ) **OVER**  
( [ **PARTITION BY** ]  
**ORDER BY** )该函数只能在查询的两个子句中指定——在 **SELECT** 子句或 **ORDER BY** 子句中。以下详细讨论不同的函数。



# ROW\_NUMBER

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

```
SELECT orderid, qty,  
       ROW_NUMBER() OVER(ORDER BY qty) AS rownum,  
       RANK()      OVER(ORDER BY qty) AS rnk,  
       DENSE_RANK() OVER(ORDER BY qty) AS densernk  
  
FROM dbo.Orders  
ORDER BY qty
```

- **ROW\_NUMBER** assigns values starting with 1, and incremented by one for each row according to the specified sort

orderid	qty	rownum	rnk	densernk
30001	10	1	1	1
10001	10	2	1	1
10006	10	3	1	1
40005	10	4	1	1
30003	15	5	5	2
30004	20	6	6	3
20002	20	7	6	3
20001	20	8	6	3
10005	30	9	9	4
30007	30	10	9	4
40001	40	11	11	5



# RANK

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

```
SELECT orderid, qty,  
       ROW_NUMBER() OVER(ORDER BY qty) AS rownum,  
       RANK() OVER(ORDER BY qty) AS rnk,  
       DENSE_RANK() OVER(ORDER BY qty) AS densernk  
FROM dbo.Orders  
ORDER BY qty
```

- RANK assigns values that indicate how many rows have lower values in order\_by\_list + 1
- Duplicates get same rank values
- Result rank values may have gaps

orderid	qty	rownum	rnk	densernk
30001	10	1	1	1
10001	10	2	1	1
10006	10	3	1	1
40005	10	4	1	1
30003	15	5	5	2
30004	20	6	6	3
20002	20	7	6	3
20001	20	8	6	3
10005	30	9	9	4
30007	30	10	9	4
40001	40	11	11	5

# DENSE\_RANK

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

```
SELECT orderid, qty,  
       ROW_NUMBER() OVER(ORDER BY qty) AS rownum,  
       RANK()      OVER(ORDER BY qty) AS rnk,  
       DENSE_RANK() OVER(ORDER BY qty) AS densernk  
  
FROM dbo.Orders  
ORDER BY qty
```

- DENSE\_RANK assigns values that indicate how many distinct rows have lower values in order\_by\_list + 1
- Duplicates get same dense rank values
- Similar to RANK with no gaps

orderid	qty	rownum	rnk	densernk
30001	10	1	1	1
10001	10	2	1	1
10006	10	3	1	1
40005	10	4	1	1
30003	15	5	5	2
30004	20	6	6	3
20002	20	7	6	3
20001	20	8	6	3
10005	30	9	9	4
30007	30	10	9	4
40001	40	11	11	5

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# demo

## Ranking Functions



# Common Table Expressions

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- 通用表表达式 (CTE) 是一个可以由定义语句引用的临时命名的结果集。在它们的简单形式中，您可以将 CTE 视为类似于视图和派生表混合功能的改进版本。在查询的 FROM 子句中引用 CTE 的方式类似于引用派生表和视图的方式。只须定义 CTE 一次，即可在查询中多次引用它。在 CTE 的定义中，可以引用在同一批处理中定义的变量。但是，CTE 的真正威力在于它们的递归功能，即 CTE 可以包含对它们自身的引用。



# 视图、派生表和 CTE 内部的查询的一般形式

- 视图

```
CREATE VIEW <view_name>(<column_aliases>) AS  
<view_query>
```

- 派生表

```
SELECT * FROM (<derived_table_query>) AS  
<derived_table_alias>(<column_aliases>)
```

- CTE

```
WITH <cte_alias>(<column_aliases>)  
AS  
(  
    <cte_query>  
)
```

```
SELECT * FROM <cte_alias>
```

在关键字 **WITH** 之后，为 **CTE** 提供一个别名，并且为它的结果列提供一个可选的别名列表；编写 **CTE** 的主体；然后从外部查询中引用它。

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## Simple CTEs



**MSDN Webcasts**

# Recursive CTEs

- 递归的 **CTE** 是根据至少两个查询（或者称为成员）构建的。一个是非递归查询，也称为固定成员 (**AM**)，只能调用一次。另一个是递归查询，也称为递归成员 (**RM**)，可以反复调用，直到查询不再返回行。查询由 **UNION ALL** 运算符连接为一个单独的 **CTE**。

```
WITH RecursiveCTE()  
AS  
(  
-- Anchor Member:  
-- SELECT query that does not refer to RecursiveCTE  
SELECT ... FROM ...  
UNION ALL  
-- Recursive Member  
-- SELECT query that refers to RecursiveCTE  
SELECT ... FROM JOIN RecursiveCTE...  
)  
-- Outer query  
SELECT ... FROM RecursiveCTE ...
```

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## ◆ Using CTEs Recursively



# 新的关系运算符

## PIVOT/UNPIVOT/APPLY

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **PIVOT**

- **PIVOT** 运算符将行旋转为列，并且可能同时执行聚合。使用**PIVOT**运算符时要注意的重要一点是，需要为它提供一个查询表达式，表达式使用视图、派生表或者**CTE**只返回所关注的列。

- **UNPIVOT**

- **UNPIVOT** 运算符执行与 **PIVOT** 运算符相反的操作；它将列旋转为行。

- **APPLY**

- **APPLY** 关系运算符允许您对外部表的每个行调用指定的表值函数一次。您可以在查询的 **FROM** 子句中指定 **APPLY**，其方式与使用 **JOIN** 关系运算符类似。**APPLY** 具有两种形式：**CROSS APPLY** 和 **OUTER APPLY**。

# New Relational Operators

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

## PIVOT 例子

Make	Year	Sales
Honda	1990	2000
Honda	1990	1000
Acura	1990	500
Honda	1991	3000
Acura	1991	300
Acura	1991	600
Acura	1992	800

SELECT \* FROM CarSales  
PIVOT(SUM(Sales) FOR  
Year IN  
([1990],[1991])) t

Make	1990	1991
Honda	3000	3000
Acura	500	900

# New Relational Operators

您的潜力，我们的动力

Microsoft®  
微软(中国)有限公司

## UNPIVOT 例子

Make	Year	Sales
Acura	1990	500
Acura	1991	900
Honda	1990	3000
Honda	1991	3000

select Make,t.Year,t.Sales  
from CarSalesPivot

UNPIVOT

(Sales for Year in ([1990],  
[1991])) t

Make	1990	1991
Honda	3000	3000
Acura	500	900

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## PIVOT/UNPIVOT/APPLY



# DDL Triggers

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

- **SQL Server 2005** 可以就整个服务器或数据库的某个范围为 **DDL** 事件定义触发器。也可以为单个 **DDL** 语句（例如，**CREATE\_TABLE**、**DROP\_TABLE**等）或者为一组语句（例如，指定 **DDL\_DATABASE\_LEVEL\_EVENTS** 想要触发器触发数据库所有 **DDL** 事件）定义 **DDL** 触发器。
  - 在 **DDL** 触发器内部，可以通过访问 `eventdata()` 函数获得与激发该触发器的事件有关的数据。该 `eventdata()` 函数返回有关事件的 **XML** 数据。
- **DDL** 触发器特别有用的方案包括 **DDL** 更改的完整性检查、审核方案以及其他方案。

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# demo

## DLL TRIGGER



**MSDN Webcasts**

# Summary

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司


- **SQL Server 2005 中的 Transact-SQL 增强功能** 提高了用户在编写查询时的表达能力，使用户可以改善代码的性能，并且扩充了错误管理能力。
- **SQL Server 2005 在 Transact-SQL 上所做的改进** 反映了其更好地满足了 **ANSI-99 SQL** 规范的要求以及客户的需求。
- **在 Transact-SQL 和 托管代码之间的选择。**

# Q&A


您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)



# 感谢大家的参与!

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

- 非常感谢大家的参与
- 您的**反馈**对我们来说是非常重要的



您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

**Microsoft**<sup>®</sup>

您的潜力，我们的动力！

**Microsoft**

msdn  


MSDN Webcasts