Business School
UNIVERSITY OF COLORADO **DENVER**

Information Systems Program

Module 2
Multidimensional data representation and manipulation

Lesson 4: Microsoft MDX Statements

Lesson Objectives

- Explain simple MDX statements
- Compare and contrast MDX and SQL
- Gain insight into MDX complexity

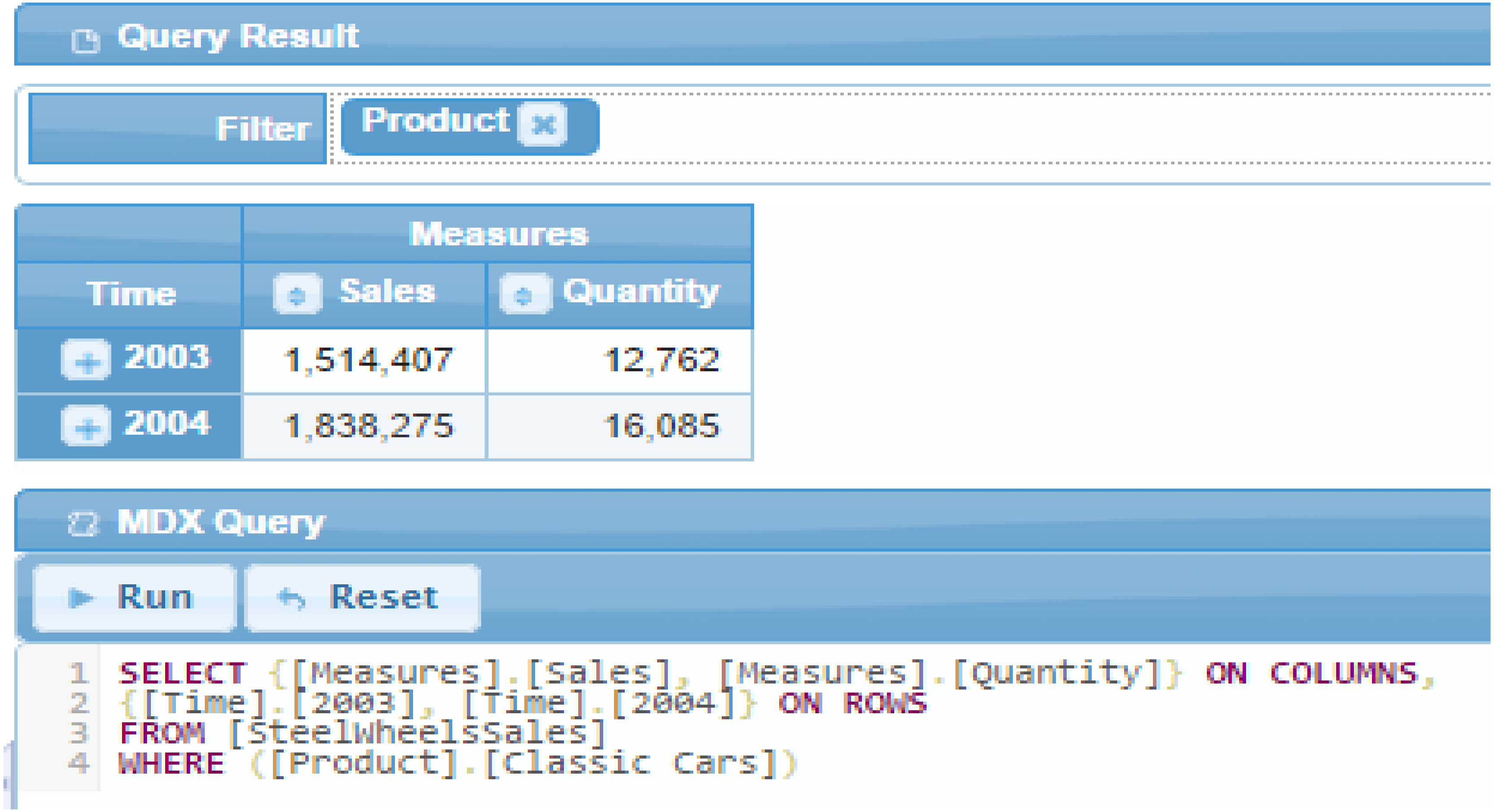
SQL Versus MDX

- Table result for SQL SELECT statement
- Data cube result for MDX SELECT statement
- Different mathematical approaches for manipulating tables and data cubes

Comparison of Clauses

	Language	
Clause	**SQL**	**MDX**
SELECT	List of columns	List of axis dimensions (source cube cells)
FROM	List of tables	Cube name
WHERE	Conditions restricting rows	Restriction to a combination of dimension members (result cube cells)

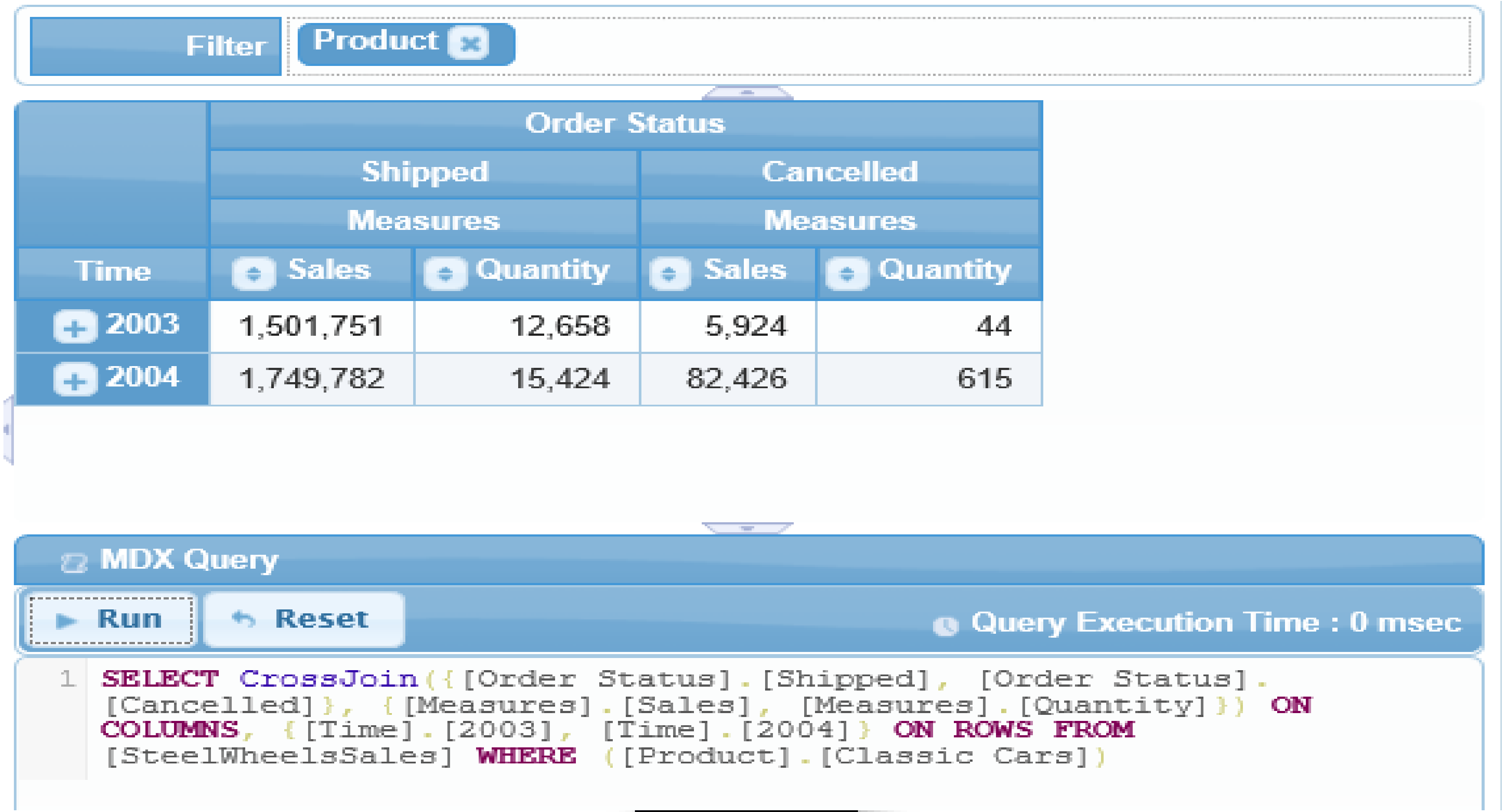
Example MDX Statement and Result



	Measures	
Time	Sales	Quantity
2003	1,514,407	12,762
2004	1,838,275	16,085

```
SELECT {[Measures].[Sales], [Measures].[Quantity]} ON COLUMNS,
{[Time].[2003], [Time].[2004]} ON ROWS
FROM [SteelWheelsSales]
WHERE ([Product].[Classic Cars])
```

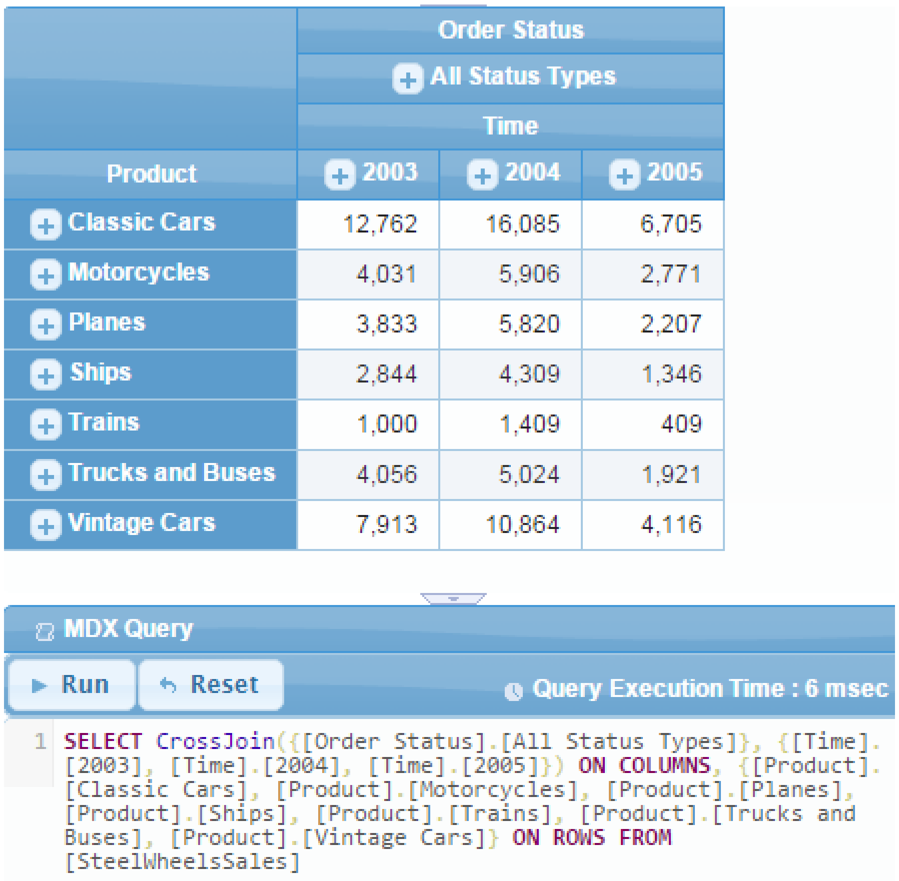
CrossJoin Operation



	Order Status			
	Shipped		Cancelled	
	Measures		Measures	
Time	Sales	Quantity	Sales	Quantity
2003	1,501,751	12,658	5,924	44
2004	1,749,782	15,424	82,426	615

```
SELECT CrossJoin({[Order Status].[Shipped], [Order Status].
[Cancelled]}, {[Measures].[Sales], [Measures].[Quantity]}) ON
COLUMNS, {[Time].[2003], [Time].[2004]} ON ROWS FROM
[SteelWheelsSales] WHERE ([Product].[Classic Cars])
```

Slicer Comparison Examples

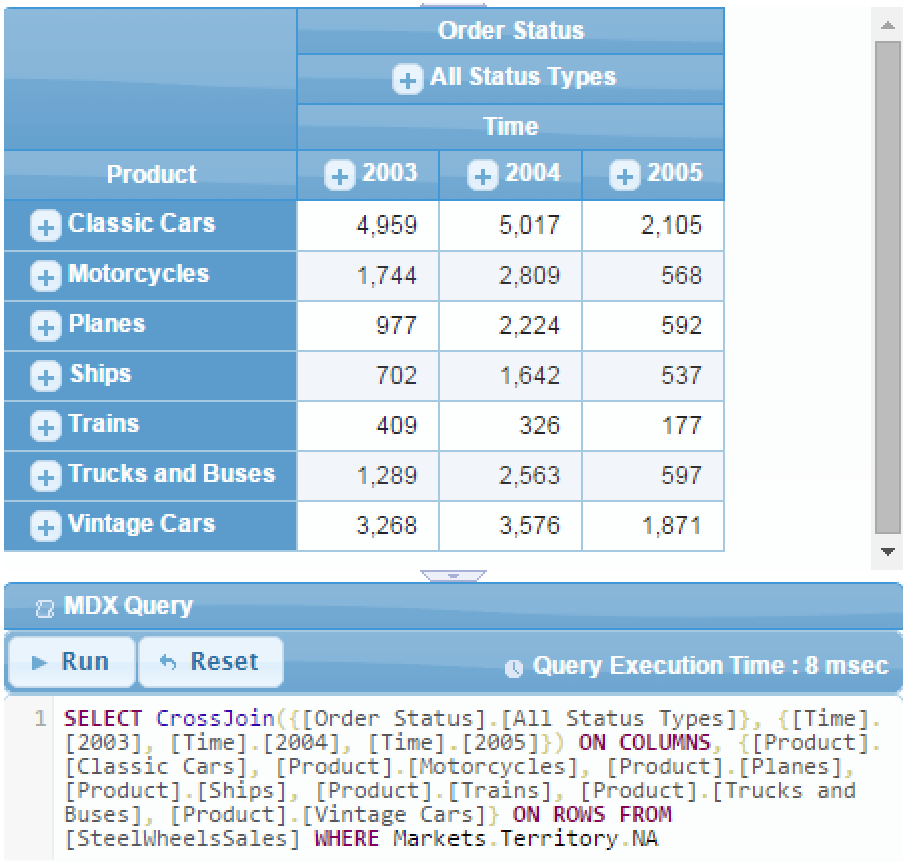


	Order Status		
	All Status Types		
	Time		
Product	2003	2004	2005
Classic Cars	12,762	16,085	6,705
Motorcycles	4,031	5,906	2,771
Planes	3,833	5,820	2,207
Ships	2,844	4,309	1,346
Trains	1,000	1,409	409
Trucks and Buses	4,056	5,024	1,921
Vintage Cars	7,913	10,864	4,116

MDX Query

Query Execution Time : 6 msec

```
SELECT CrossJoin({[Order Status].[All Status Types]}, {[Time].
[2003], [Time].[2004], [Time].[2005]}) ON COLUMNS, {[Product].
[Classic Cars], [Product].[Motorcycles], [Product].[Planes],
[Product].[Ships], [Product].[Trains], [Product].[Trucks and
Buses], [Product].[Vintage Cars]} ON ROWS FROM
[SteelWheelsSales]
```



	Order Status		
	All Status Types		
	Time		
Product	2003	2004	2005
Classic Cars	4,959	5,017	2,105
Motorcycles	1,744	2,809	568
Planes	977	2,224	592
Ships	702	1,642	537
Trains	409	326	177
Trucks and Buses	1,289	2,563	597
Vintage Cars	3,268	3,576	1,871

MDX Query

Query Execution Time : 8 msec

```
SELECT CrossJoin({[Order Status].[All Status Types]}, {[Time].
[2003], [Time].[2004], [Time].[2005]}) ON COLUMNS, {[Product].
[Classic Cars], [Product].[Motorcycles], [Product].[Planes],
[Product].[Ships], [Product].[Trains], [Product].[Trucks and
Buses], [Product].[Vintage Cars]} ON ROWS FROM
[SteelWheelsSales] WHERE Markets.Territory.NA
```

Summary

- Similar syntax as SQL SELECT statement
- Axes specified in SELECT clause
- Crossjoin operator to combine dimensions on axis
- Slicer conditions specified in the WHERE clause
- Tedious and complex language