Basic Select Queries

CMPE 2400

Databases

use Statement and go

- The use statement specifies which database to access for the execution of a query script, or portion thereof.
  - This may also be accomplished using the database dropdown box in SQL Server Management Studio
- go causes the current group of statements to be executed immediately by the query processor, rather than grouping all queries into a single batch before execution
  - Proper use of go in this course will include
    - separating questions on labs / exams
    - ensuring that create statements always appear as the first command in a batch

Absolute Naming

- AKA: **Fully Qualified Names**
- Though use statements *could* be used to navigate between databases for the execution of individual query batches, and will even be necessary at times, a preferred method is to have your query batch work no matter what database you are in
- This is accomplished by resolving the table names in your queries to the databases in which they reside.
  - Another benefit to absolute naming is that tables in more than one database may be accessed in a single query.

Absolute Naming

- The syntax is fairly straight-forward.

DatabaseName.SchemaName.TableName[.ColumnName]

- Note that for older databases, ones that predate the introduction of *schemas*, the SchemaName is always *dbo*, which stands for *database owner.*
  - Both the NorthwindTraders and Publishers databases use *dbo*.
  - *dbo* is still a valid SchemaName choice in newer databases.
- Also note that the ColumnName is an optional add-on to the absolute naming syntax, but is not a requirement unless there is a ColumnName duplicated between the tables.

select

- The purpose of the select command is to retrieve a result set.
  - This does not mean that a table in the database must be accessed for data.
  - Literal values may be selected, as well as calculations
    - Literal values may be numerical or text based

select

- The select statement is used to create a query that will return a set of data.
  - When a table is specified, it will be the source used to retrieve said data.

```
select [field] from [table]
```

```
select [field1], [field2] from [table]
```

select *

- The * operator can be used to return all of the fields in a table
  - This is basically a data dump.

```
1  --returns all fields from the employees table
2  select *
3  from NorthwindTraders.dbo.Employees
4
```

  - Avoid using * in a production RDBMS
    - Causes a table lookup for actual field names every time it is used.
    - If the structure of a table is changed, the order of the returned fields will also change, possibly causing dependencies to break.
    - If executed against a **view**, the performance implications can be much larger!

Selecting Specific Columns

- We can specify which columns we want displayed from a table.

```
1  use ClassTrak
2  go
3
4  select last_name, first_name
5  from Students
```

Specifying Database and Table Names

We can specify the database name in our select query, so we don't need a separate "Use" statement

```
select last_name, first_name
from ClassTrak.dbo.Students
```

We can also specify the table name together with the field name.

```
select Students.last_name, Students.first_name
from ClassTrak.dbo.Students
```

Column Alias

- Use when the column name is not meaningful
  - “au_n” is only obvious to someone familiar with the database
- The as keyword allows you to set an alias that can be used for a column name or table name in the query
- The alias will be displayed as the column heading to the user
  - Note that this does not change the column name in the database

- **In this course, we shall alias all columns that are not a single, easily understood word**

Aliasing Using as

```
1  select   CompanyName as Company,
2           ContactName as Contact
3  from     NorthwindTraders.dbo.Customers
```

Results

Company	Contact
Alfreds Futterkiste	Maria Anders
Ana Trujillo Emparedados y helados	Ana Trujillo
Antonio Moreno Taquería	Antonio Moreno
Around the Horn	Thomas Hardy
Berglunds snabbköp	Christina Berglund
Blauer See Delikatessen	Hanna Moos
Blondesddsl père et fils	Frédérique Citeaux
Bólido Comidas preparadas	Martín Sommer
Bon app'	Laurence Lebihan
Bottom-Dollar Markets	Elizabeth Lincoln
B's Beverages	Victoria Ashworth
Cactus Comidas para llevar	Patricio Simpson
Centro comercial Moctezuma	Francisco Chang
Chop-suey Chinese	Yang Wang
Comércio Mineiro	Pedro Afonso

127 %

Query executed successfully. | data.net.nait.ca,24680 (14.... | sdytiuk (53) | ClassTrak | 00:00:00 | 94 rows

Aliasing in MS SQL Server

- In SQL Server, you may also alias **without** the as keyword
  - Caution should be used here, as this may not work properly in another RDBMS

```
1  select   CompanyName Company,
2           ContactName Contact
3  from     NorthwindTraders.dbo.Customers
```

Aliasing

- If you wish to include spaces in your result set column name, you may do so using single quotes

```
1  select   CompanyName 'Company Name',
2           ContactName as 'Contact Name'
3  from     NorthwindTraders.dbo.Customers
```

Why Alias?

- Change the column name to something else:
  - FName becomes ‘First Name’
- Name calculated fields
  - By default, calculated fields have **no name**!
  - (Cost * Units) becomes ‘Total Cost’ instead of (no column name)

Results Ordering

- Often, the number of results returned by a query will be plentiful.
  - Result order is **never guaranteed** unless order by is used!
  - Using the order by clause, specify how the result set is ordered

Sorting with order by

- You may specify the sort order, or group of orders, for the result set returned by a select statement.

```
select      field1, field2, … , fieldN
from        table
order by    field1 [asc/desc], field2 [asc/desc], …
```

asc = sort ascending *(default)*

desc = sort descending

asc does not need to be specified as it is the default.

Sorting with order by

- Display an list of employees sorted in descending order by Hire Date and then ascending order by first name
  - Another way to say this would be to sort by FirstName in ascending order ***within*** HireDate in descending order

```
Select FirstName as 'First Name',
       LastName as 'Last Name',
       HireDate as 'Date Hired'
From NorthWindTraders.dbo.Employees
Order By HireDate Desc, FirstName Asc
```

TOP

DISTINCT

select modifiers

- View records that match:
  - “best” or “worst”
  - “greatest” or “least”
  - “most recent” or “least recent”, etc.
- top can help solve problems such as:
  - At the end of each month, present the “best” salespeople with a bonus.
  - Find the regions in which the company has experienced the “worst” gross sales to target for extra advertising.

top combined with order by

top n

- specify an exact number (n) records to be returned

```
1  --what are the last 5 orders?
2  select  top 5
3          OrderID,
4          OrderDate
5  from    NorthwindTraders.dbo.Orders
6  order by OrderDate desc
7
```

127 %

Results

OrderID	OrderDate
11077	2021-05-06 16:39:00.000
11075	2021-05-06 16:31:00.000
11074	2021-05-06 13:27:00.000
11076	2021-05-06 01:12:00.000
11071	2021-05-05 18:40:00.000

(5 rows affected)

```sql
--what are the orders ordered in the last 1% of all order dates?
select  top 1 percent
        OrderID,
        OrderDate
from    NorthwindTraders.dbo.Orders
order by OrderDate desc
```

top n percent

```
OrderID     OrderDate
----------- -----------------------
11077       2021-05-06 16:39:00.000
11075       2021-05-06 16:31:00.000
11074       2021-05-06 13:27:00.000
11076       2021-05-06 01:12:00.000
11071       2021-05-05 18:40:00.000
11070       2021-05-05 18:37:00.000
11073       2021-05-05 06:46:00.000
11072       2021-05-05 05:37:00.000
11067       2021-05-04 14:27:00.000

(9 rows affected)
```

top n [percent] with ties

Careful! Sometimes more records qualify for the top n prize, and if ties aren't included, the DBMS decides what gets included!

What are the 3 highest quantity orders from the Order Details table in NorthwindTraders?

top n [percent] with ties

- The following is the unrestricted result set.
  - As can be seen below, there are several records that qualify as having the third highest quantity.

```
select  OrderID as 'Order ID',
        Quantity
from    NorthwindTraders.dbo.[Order Details]
order by Quantity desc
```

100 %

Results

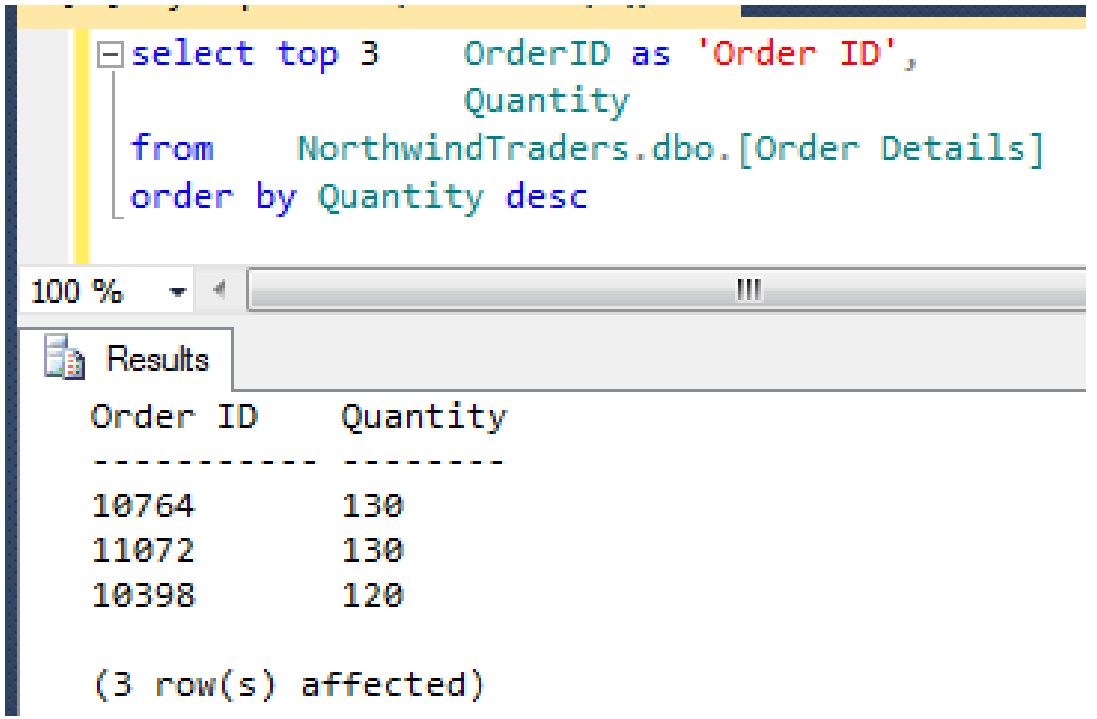
Order ID	Quantity
10764	130
11072	130
10776	120
10711	120
10894	120
10398	120
10451	120
10515	120
10595	120
10678	120
10895	110
10713	110
11017	110

100 %

bender.net.nait.ca,24680 (1... | skelemen1 (78) | NorthwindTraders | 00:00:00 | 2155 rows

top n [percent] with ties

- The database engine will do its best to follow your wishes, yielding the following, inaccurate result set.



top n [percent] with ties

- Adding the with ties clause permits all rows that meet the top 3 requirement to be included.

```
select top 3 with ties  OrderID as 'Order ID',
                        Quantity
from    NorthwindTraders.dbo.[Order Details]
order by Quantity desc
```

100 %

Results

Order ID	Quantity
10764	130
11072	130
10776	120
10711	120
10894	120
10398	120
10451	120
10515	120
10595	120
10678	120

(10 row(s) affected)

order by and top

- In the queries on the previous slides, it should be noted that the order by clause was always present.
  - The reason for this is that you must specify an ordering scheme for your result set if your results are to represent the “most” or “least” of some measureable value.
    - If the ordering is left out, then your result set will correspond to the first *n* records in the unrestricted result set.
    - As might be expected, the *with ties* portion of the modifier will break completely. Try to explain…

order by and top

- The information in the following set is not exactly useful.

```
select top 10   OrderID as 'Order ID',
                Quantity
from    NorthwindTraders.dbo.[Order Details]
```

100 %

Results

Order ID	Quantity
10248	12
10248	10
10248	5
10249	9
10249	40
10250	10
10250	35
10250	15
10251	6
10251	15

(10 row(s) affected)

order by and top

- Even worse…

```
select top 10 with ties OrderID as 'Order ID',
                        Quantity
from    NorthwindTraders.dbo.[Order Details]
```

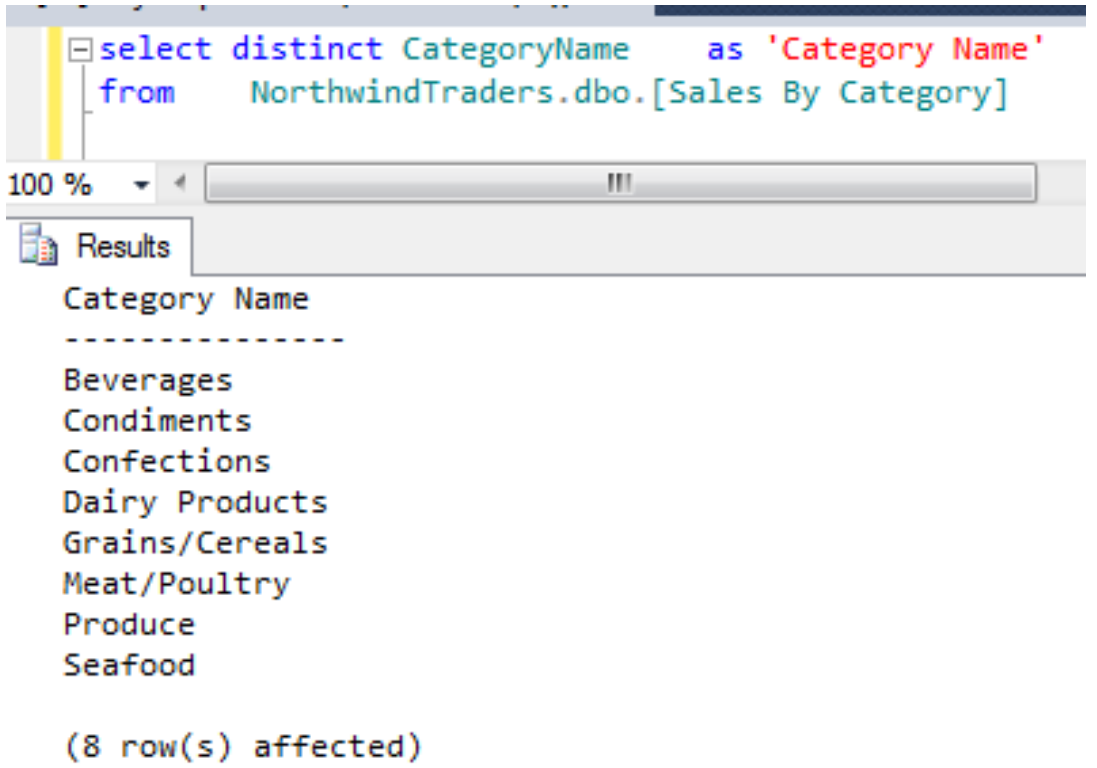
100 %

Results

```
Msg 1062, Level 15, State 1, Line 3
The TOP N WITH TIES clause is not allowed without a corresponding ORDER BY clause.
```

distinct

- The distinct modifier removes duplicate data from the returned result set
  - This will produce a result set of unique rows



distinct Operator

- If multiple columns are specified, then ***any*** difference in ***any*** column defines uniqueness.

```
select distinct CategoryName    as 'Category Name',
                ProductSales    as 'Product Sales'
from    NorthwindTraders.dbo.[Sales By Category]
where   CategoryName = 'Meat/Poultry'
```

100 %

Results

```
Category Name   Product Sales
--------------- --------------------
Meat/Poultry    3184.29
Meat/Poultry    6935.50
Meat/Poultry    7832.88
Meat/Poultry    10661.93
Meat/Poultry    17604.60
Meat/Poultry    34755.92

(6 row(s) affected)
```

Using Alias for Table Names

- We can use an alias for a table name and use this alias together with the column names for the table

```
2  select  o.OrderID,
3          o.OrderDate
4  from    NorthwindTraders.dbo.Orders o
5  order by o.OrderDate desc
```