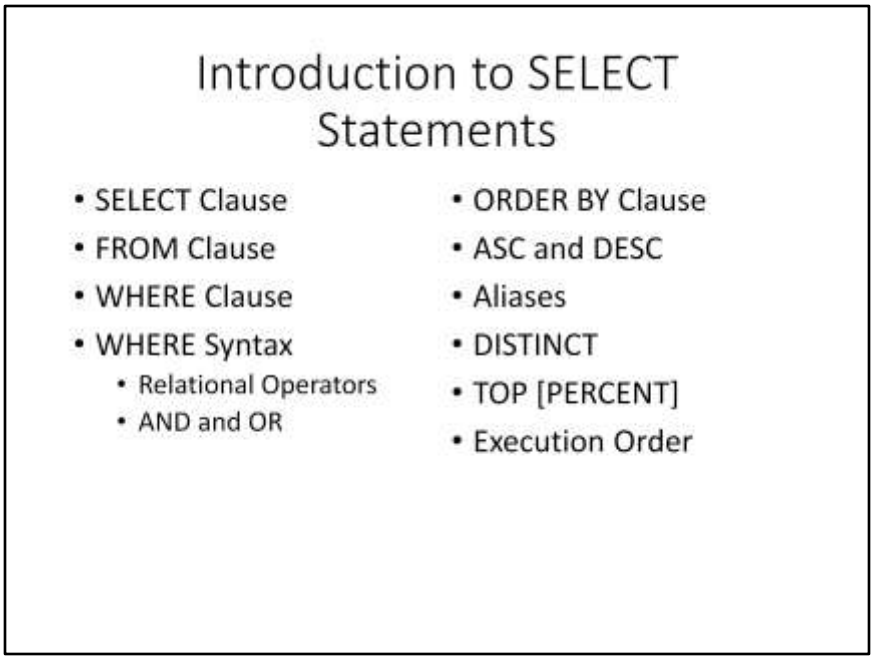
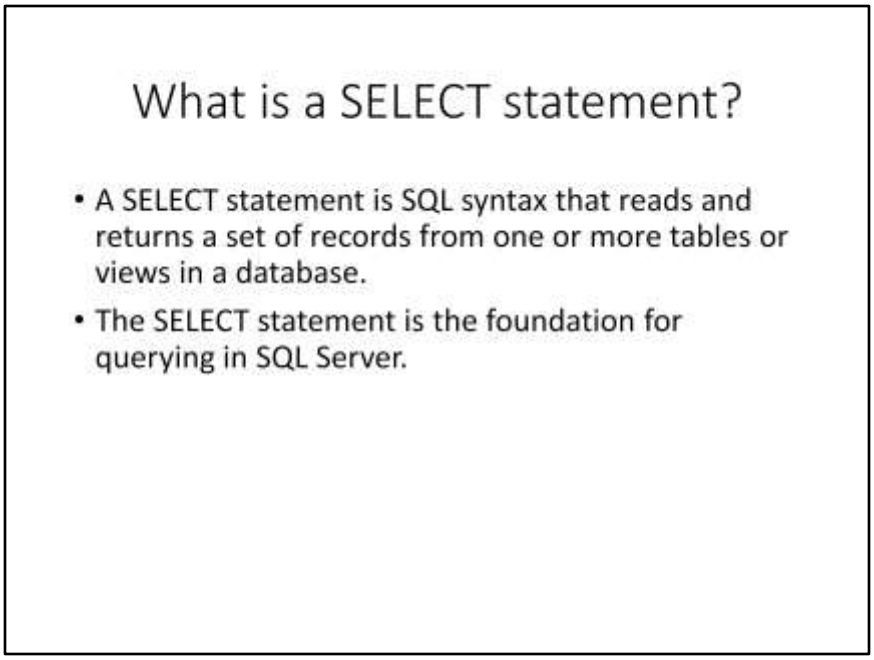
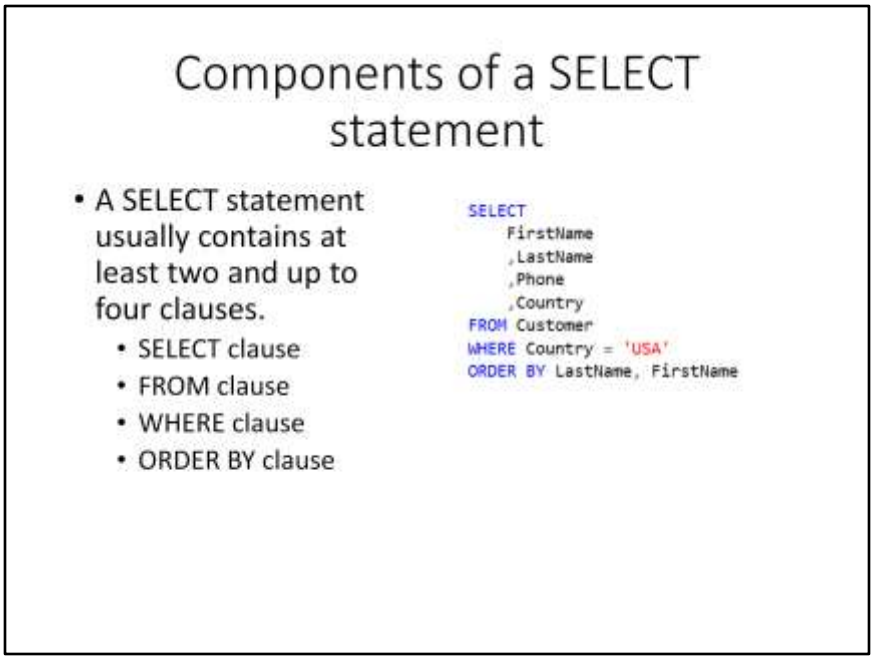


This presentation will be on the SQL Select statement. I will talk about the clauses that make up a select statement, select statement syntax, and some of the popular key words used with select statements.



Select statements are the foundation for querying or reading data out of a database. A select statement tells the SQL server from which tables within a database it wants to read as well as which columns to display. A select statement also defines how the displayed data is sorted and filtered.



There are four clauses that make up a select SQL statement . These are the SELECT clause, the FROM clause, the WHERE clause, and the ORDER BY clause. You can see an example of a select statement on this slide. Notice that the clauses are highlighted in blue. SQL Server Management Studio automatically highlights keywords for you.

SELECT clause

- When querying from a table the SELECT clause determines what columns will be displayed in the result set
- The order you write the columns will be the order in which they will be displayed

```
SELECT
    FirstName
    ,LastName
    ,Phone
    ,Country
FROM Customer
WHERE Country = 'USA'
ORDER BY LastName, FirstName
```

	FirstName	LastName	Phone	Country
1	Julia	Barnett	+1 (801) 531-7272	USA
2	Michelle	Brooks	+1 (212) 221-3546	USA
3	Kathy	Chase	+1 (775) 223-7665	USA
4	Richard	Cunningham	+1 (817) 924-7272	USA
5	John	Gordon	+1 (617) 522-1333	USA
6	Tim	Goyer	+1 (408) 996-1010	USA

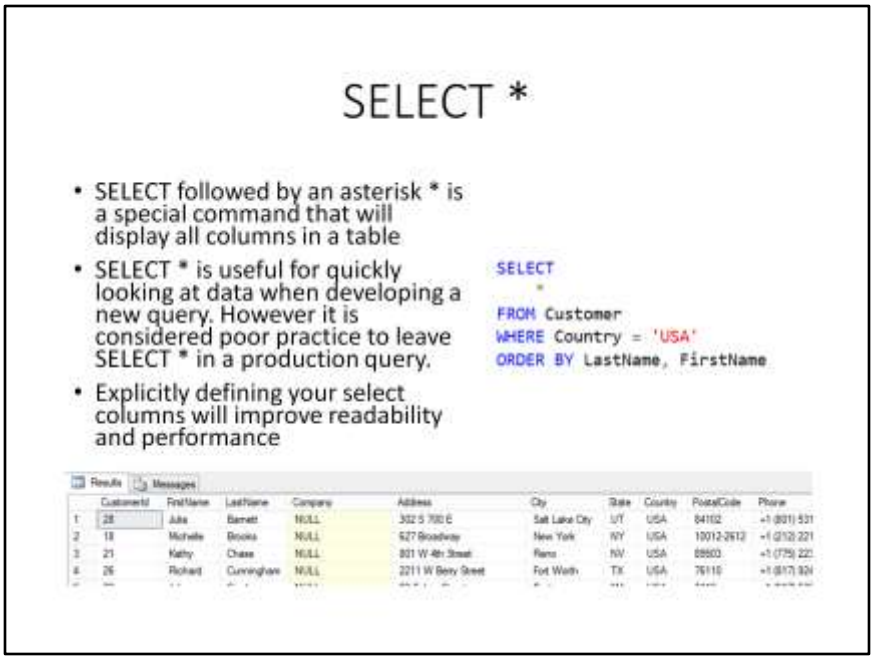
The select clause of a select statement is used to identify which columns will be displayed from the table or record set you are querying against. The order in which you type the columns will be the order in which they are displayed in the result set. A select clause is required for all select statements.

Column Syntax

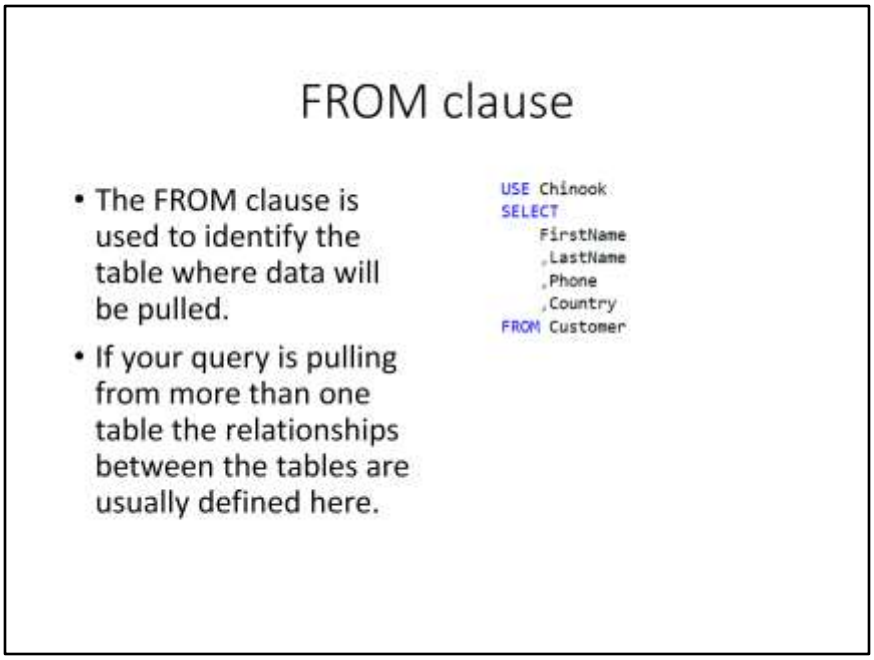
- You must type the column name as it appears in the table
- Columns with a space in the name must be enclosed in square brackets [ ] or quotes " ".
  - e.g. [First Name] or "First Name"
- Multiple columns in the SELECT statement must be separated by a comma.
- For readability it is common practice to put each column on a separate line.
- indenting column lines and placing commas at the beginning of the line is also a common readability practice

```
SELECT
    SystemInformationID
    ,[Database Version]
    ,VersionDate
    ,ModifiedDate
FROM AdventureWorks2014.dbo.AWBuildVersion
```

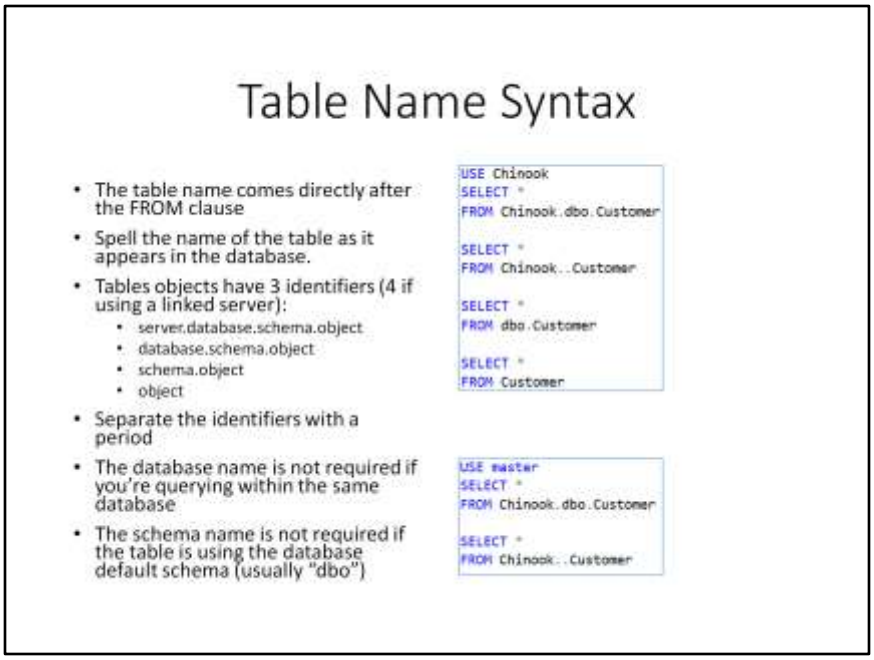
When writing the select clause there are a couple syntax requirements you need to follow. First all columns in the select clause need to be separated by a comma. Spacing before or after commas is not required but it can help with readability. Second if the column names in your table have spaces then you will need to enclose the column names in brackets or double quotes. Brackets are the preferred method in Microsoft SQL Server. It is common practice to indent and place each column on its own line. Many developers also prefer placing the commas at the beginning of each line. Again this is for readability purposes, but it is also makes editing column lines a little easier by not having to move a comma at the end of the line.



The asterisk is a special command used to return all the columns in your table or record set. It is very useful when you want to quickly look at all available columns and their values. It is fine to use during query development, but it is usually considered bad practice to keep select star in any production queries. It is better to write out the column names for the sake of readability and performance. Also, any queries using select star might fail if columns are added to the underlying tables at a later date.



The from clause is used to identify from which table or tables you will be pulling data. The from clause is also where the relationships between tables are usually defined. I will discuss relationships between multiple tables in a later presentation. In Microsoft SQL server the from clause is technically optional, but you cannot read data from a database without the from clause.



All tables on a single SQL Server have a three part name. The name of the database in which it resides, the schema in which is resides, and the object name itself. When you run query inside a database it is not required to include the database identifier in the name. Additionally if your table is using the default schema, then you don't need to include the schema identifier either. The default schema for a database is the "dbo" schema. In the first example on this slide, all four versions of the table name will execute successfully. This is because the queries are being executed within the Chinook database. In the second example only two of the four table names will work because we are calling the query from within the master database. This means we need the database identifier in order to map to Chinook database. The USE keyword in the examples tells SQL Server on which database to run the queries.

WHERE clause

- The WHERE clause of a query is used to filter the rows of a result set.
- The WHERE clause can only filter against objects that exist in the FROM clause
- A column does not have to be present in the SELECT clause in order to filter against it in the WHERE clause

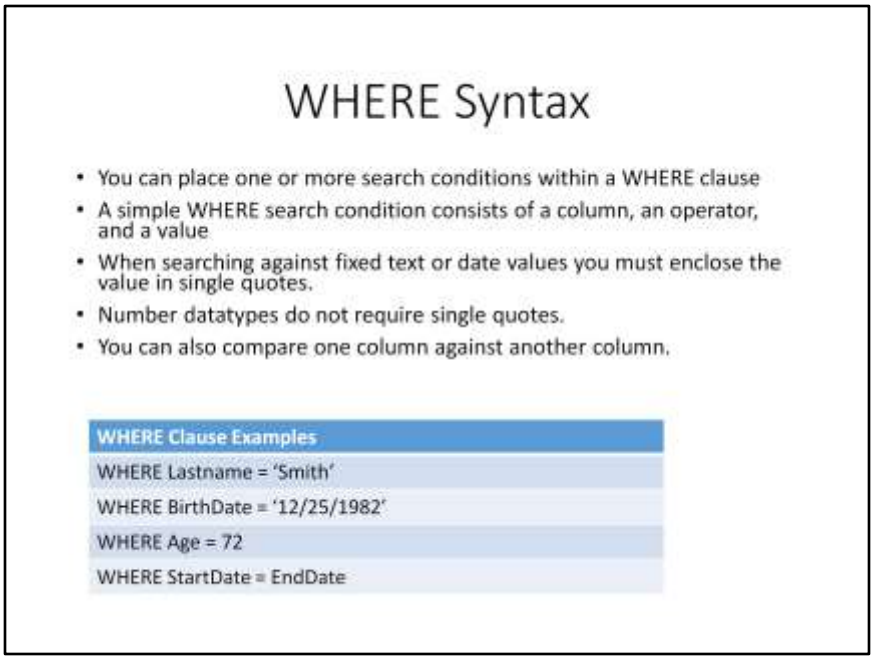
```
SELECT *
FROM Album
WHERE ArtistId = 90
```

00 %

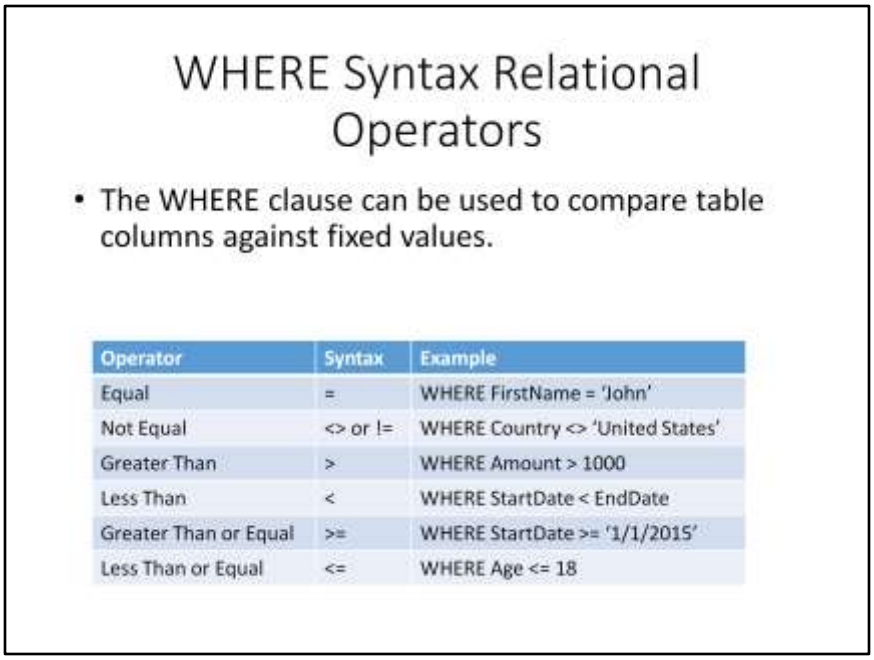
Results | Messages

	AlbumId	Title	ArtistId
1	94	A Matter of Life and Death	90
2	95	A Real Dead One	90
3	96	A Real Live One	90
4	97	Brave New World	90
5	98	Dance Of Death	90
6	99	Fear Of The Dark	90
7	100	Iron Maiden	90

The where clause is used to filter rows within a record set. A where clause can reference any of the columns present in a table in the from clause. It is not required that the filtering column in the where clause be present in the select clause. The where clause is optional within a select statement.



You can place one or more search conditions within a where clause. The syntax of where clause consists of the column on which you're searching, the operator such as an equal sign, and the value for which you are searching. When searching on text or date related columns, you must enclose the value within single quotes. Single quotes are not required when searching columns based on a numeric datatype. It is possible to compare two columns to one another within a where clause.



This slide shows the syntax of the relational operators available within a where clause. You should be familiar with most of these operators based on your school math days. Note that the not equal operator has two possible syntaxes. The greater than and less than signs together are the ANSI SQL standard. However Microsoft SQL Server also accepts the exclamation point followed by an equal sign for the not equal operator. This syntax is popular in other programming languages.

WHERE Logical Operators

- You can use the AND and OR operators to string together multiple search conditions
- Use parenthesis prioritize the order search conditions are executed. Search conditions in parenthesis are executed first

```
SELECT
    FirstName
    ,LastName
    ,City
    ,State
    ,Country
FROM Chinook.dbo.Customer
WHERE Country = 'Brazil'
    OR (Country = 'USA' AND State = 'CA')
```

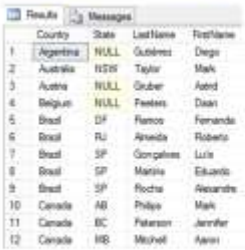
	FirstName	LastName	City	State	Country
1	Luis	Gonçalves	São José dos Campos	SP	Brazil
2	Eduardo	Martins	São Paulo	SP	Brazil
3	Alexandre	Rocha	São Paulo	SP	Brazil
4	Roberto	Almeida	Rio de Janeiro	RJ	Brazil
5	Fernanda	Ramos	Brasília	DF	Brazil
6	Frank	Harris	Mountain View	CA	USA
7	Tim	Goyer	Cupertino	CA	USA
8	Dan	Miller	Mountain View	CA	USA

It is possible to string together multiple search conditions within a single where clause. You can do this using the AND and OR operators. When building the where clause it is important to remember that the AND operator gets resolved before the OR operator. If you wish to change the default execution order of search conditions, you need to use parenthesis. Conditions within parenthesis will resolve first. Parenthesis can also help with the readability of complex where clauses.

ORDER BY Clause

- The ORDER BY clause is used to sort the order of rows in the record set.
- Separate multiple order by clauses with a comma.

```
SELECT
    Country
    ,State
    ,LastName
    ,FirstName
FROM Customer
ORDER BY Country, State, LastName
```



	Country	State	LastName	FirstName
1	Argentina	NULL	Gutiérrez	Diego
2	Australia	NSW	Taylor	Mark
3	Austria	NULL	Gruber	Astrid
4	Belgium	NULL	Peeters	Daan
5	Brazil	DF	Ramos	Fernanda
6	Brazil	RJ	Almeida	Roberto
7	Brazil	SP	Gonçalves	Luís
8	Brazil	SP	Martins	Eduardo
9	Brazil	SP	Rocha	Alexandre
10	Canada	AB	Philips	Mark
11	Canada	BC	Paterson	Jennifer
12	Canada	MB	Mitchell	Aaron

The order by clause is used to sort the order of rows present within a record set. It is possible to sort by more than one column by adding additional columns each separated by a comma. The first column will be the first to be sorted followed by the second column then the third, and so on and so on. Columns will be sorted in ascending order by default. The order by clause is optional within a select statement.

ORDER BY ASC and DESC

- The ASC and DESC keywords can be added after each sort condition to sort in ascending or descending order.
- If no keyword is included the sorting will be done in ascending order by default.

```
SELECT
    Country
    ,State
    ,LastName
    ,FirstName
FROM Customer
ORDER BY
    Country DESC, State DESC, LastName ASC
```

48	Canada	MB	Mitchell	Aaron
49	Canada	BC	Peterson	Jennifer
50	Canada	AB	Philips	Mark
51	Brazil	SP	Gonçalves	Luís
52	Brazil	SP	Martins	Eduardo
53	Brazil	SP	Rocha	Alexandre
54	Brazil	RJ	Almeida	Roberto
55	Brazil	DF	Ramos	Fernanda
56	Belgium	NULL	Peeters	Daan
57	Austria	NULL	Gruber	Astrid

To specify whether a column is sorted in ascending or descending order, you can add the ASC or DESC keyword after the column name. If no keyword is added, the column will sort in ascending order by default.

Column Aliases

- It is possible to rename column headers as they appear in the result set.
- Add the AS keyword and the new name to the end of the column being renamed.
- The AS keyword can be omitted but it helps with readability.
- Enclose aliases with a space in brackets or double quotes

```
SELECT
    FirstName AS [First Name]
    ,LastName AS [Last Name]
    ,Phone AS [Phone Number]
    ,PostalCode AS Zip
FROM Customer
```

OR

```
SELECT
    FirstName [First Name]
    ,LastName [Last Name]
    ,Phone [Phone Number]
    ,PostalCode Zip
FROM Customer
```

When working with select statements you may be interested in changing what name is displayed for a column name within the result set. This is known as aliasing a column. You can alias a column by typing the AS keyword after the column name followed by the alias name. If you choose to include a space in your alias, you must enclose the name in brackets or double quotes. Note that the AS keyword is not required for making a column alias, however it does help with readability.

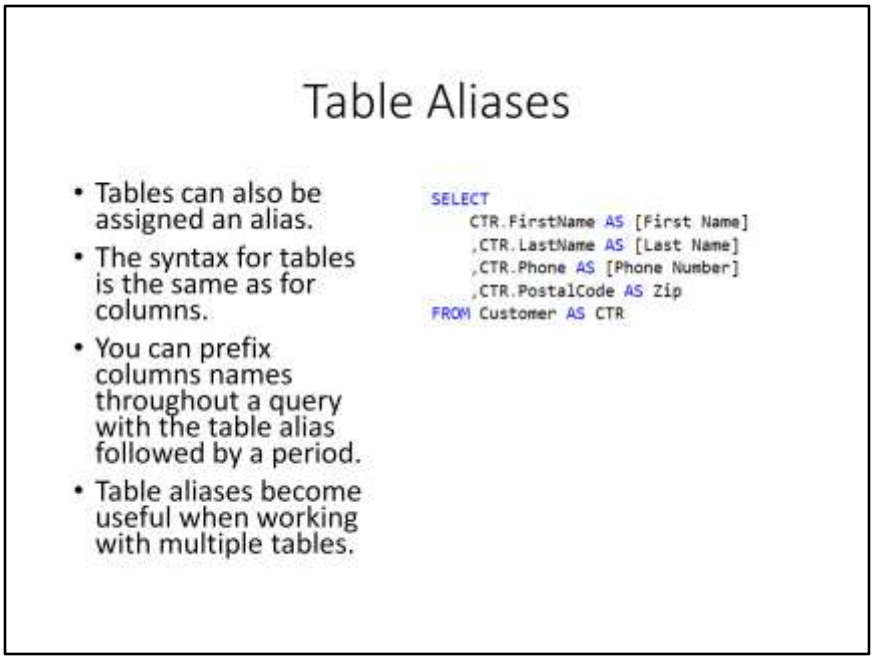
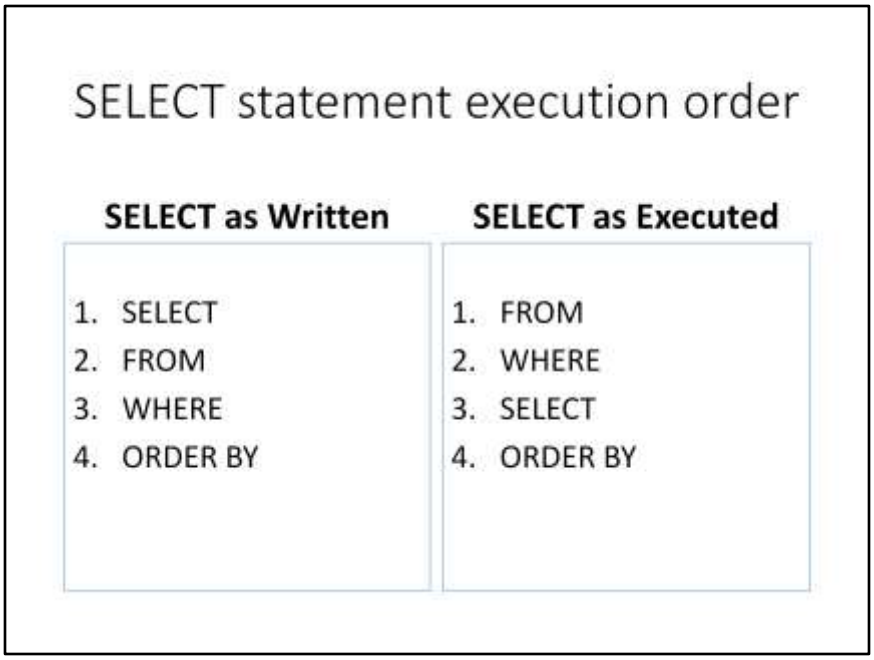


Table names can also be assigned an alias. The syntax for creating a table alias is the same as that for a column alias. You can use a table alias, or the full table name for that matter, to identify which table a column comes from. You do this by prefixing the column name with the table alias name followed by a period. Table aliases come into use when querying against multiple tables. We will discuss table aliases more later in the course.



It is important to note that the order in which a select statement is written, is not the same order in which it is executed. You write a select statement starting with the select clause followed by the from, where, and order by clauses. However in SQL the from clause in executed first, then the where clause, followed by the select and order by clauses.

Why Execution Order is Important

- Intellisence for the SELECT clause will not pick up column names until there is a from clause to reference
- You can use alias names in the ORDER BY clause
- You *cannot* use alias names in the WHERE clause (where is executed before select)

There are a couple reasons why knowing select statement execution order is important. First if you are writing a query using SQL server management studio, you cannot take advantage of intellisence for the select clause columns until you have already defined the from clause. Second when writing an order by clause, you can use the alias name within the order by clause. This is possible because the select clause resolves before the order by clause so SQL server knows the aliases exist. However you are not allowed to use column aliases in the where clause. This is because the where clause resolves before the select clause.

SELECT DISTINCT

- Add the DISTINCT keyword immediately after the SELECT clause if you wish to remove duplicate values from the result set

With Distinct

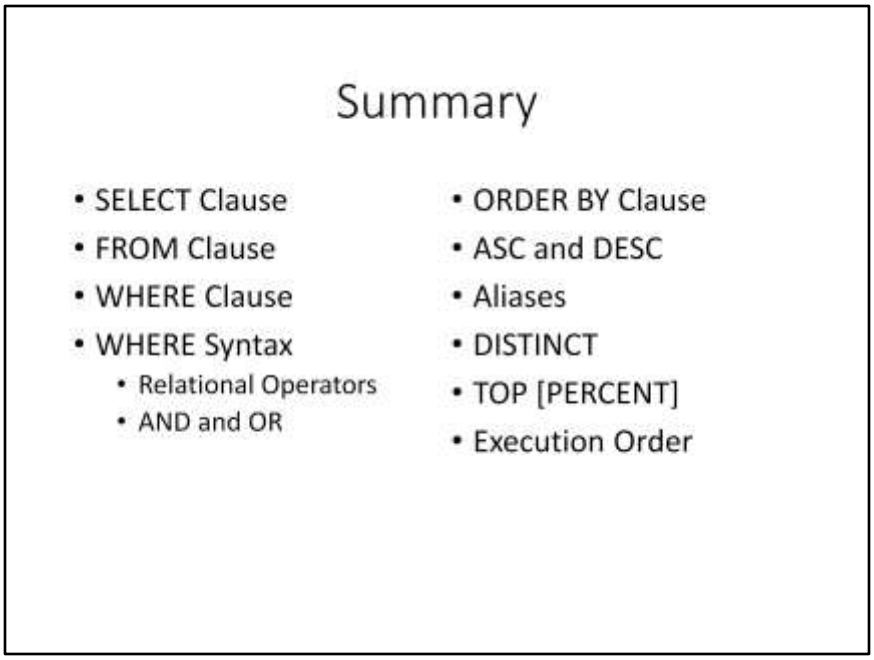
Without Distinct

A keyword you can add to a select clause is the distinct keyword. The distinct keyword will remove duplicate values from your result set. This can come in handy when you have a table with thousands of rows and you want to only see the unique values populated in a particular column, or columns.

SELECT TOP

- Use the TOP keyword and a number after SELECT to limit the number of rows returned
- Add the PERCENT keyword if you want to limit by percentage instead of a fixed number
- TOP is useful when dealing with very large result sets

The top keyword is used in a select clause to only return a specified number of records. The syntax is the top keyword followed by an integer. The integer specifies the number of rows to return. An optional parameter for the top keyword is the percent keyword. If you add the percent keyword after the integer it returns a percentage of the total result set, where the integer represents a percentage between 1 and 100. Top is useful for speeding up result output when working with large data sets, as well as finding the top values…as the name implies.



This completes the presentation on the introduction to select statements in SQL Server.