



Introduction to SELECT Statements

- SELECT Clause
- FROM Clause
- WHERE Clause
- WHERE Syntax
 - Relational Operators
 - AND and OR
- ORDER BY Clause
- ASC and DESC
- Aliases
- DISTINCT
- TOP [PERCENT] [WITH TIES]
- Execution Order



What is a SELECT statement?

- A SELECT statement is SQL syntax that reads and returns a set of records from one or more tables or views in a database.
- The SELECT statement is the foundation for querying in SQL Server.



Components of a SELECT statement

- A SELECT statement usually contains at least two and up to four clauses.
 - SELECT clause
 - FROM clause
 - WHERE clause
 - ORDER BY clause

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

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
```

Results		Messages		
	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

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

Results		Messages		
	SystemInformationID	Database Version	VersionDate	ModifiedDate
1	1	12.0.1800	2014-02-20 ...	2014-07-08 ...

SELECT *

- `SELECT` followed by an asterisk `*` is a special command that will display all columns in a table
- `SELECT *` is useful for quickly looking at data when developing a new query. However it is considered poor practice to leave `SELECT *` in a production query.
- Explicitly defining your select columns will improve readability and performance

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

	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone
1	28	Julia	Barnett	NULL	302 S 700 E	Salt Lake City	UT	USA	84102	+1 (801) 531
2	18	Michelle	Brooks	NULL	627 Broadway	New York	NY	USA	10012-2612	+1 (212) 221
3	21	Kathy	Chase	NULL	801 W 4th Street	Reno	NV	USA	89503	+1 (775) 223
4	26	Richard	Cunningham	NULL	2211 W Berry Street	Fort Worth	TX	USA	76110	+1 (817) 924
5	22	John	Smith	NULL	2211 W Berry Street	Fort Worth	TX	USA	76110	+1 (817) 924



FROM clause

- The FROM clause is used to identify the table where data will be pulled.
- If your query is pulling from more than one table the relationships between the tables are usually defined here.

```
USE Chinook
SELECT
    FirstName
    , LastName
    , Phone
    , Country
FROM Customer
```



Table Name Syntax

- The table name comes directly after the FROM clause
- Spell the name of the table as it appears in the database.
- Tables objects have 3 identifiers (4 if using a linked server):
 - server.database.schema.object
 - database.schema.object
 - schema.object
 - object
- Separate the identifiers with a period
- The database name is not required if you're querying within the same database
- The schema name is not required if the table is using the database default schema (usually "dbo")

```
USE Chinook
SELECT *
FROM Chinook.dbo.Customer

SELECT *
FROM Chinook..Customer

SELECT *
FROM dbo.Customer

SELECT *
FROM Customer
```

```
USE master
SELECT *
FROM Chinook.dbo.Customer

SELECT *
FROM Chinook..Customer
```


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



WHERE Syntax

- You can place one or more search conditions within a WHERE clause
- A simple WHERE search condition consists of a column, an operator, and a value
- When searching against fixed text or date values you must enclose the value in single quotes.
- Number datatypes do not require single quotes.
- You can also compare one column against another column.

WHERE Clause Examples

WHERE Lastname = 'Smith'

WHERE BirthDate = '12/25/1982'

WHERE Age = 72

WHERE StartDate = EndDate



WHERE Syntax Relational Operators

- The WHERE clause can be used to compare table columns against fixed values.

Operator	Syntax	Example
Equal	=	WHERE FirstName = 'John'
Not Equal	<> or !=	WHERE Country <> 'United States'
Greater Than	>	WHERE Amount > 1000
Less Than	<	WHERE StartDate < EndDate
Greater Than or Equal	>=	WHERE StartDate >= '1/1/2015'
Less Than or Equal	<=	WHERE Age <= 18

WHERE Logical Operators

- You can use the AND and OR operators to string together multiple search conditions
- Use parenthesis to 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')
```

Results

Messages

	FirstName	LastName	City	State	Country
1	Luís	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

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	Peterson	Jennifer
12	Canada	MB	Mitchell	Aaron

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

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
```

Results		Messages		
	First Name	Last Name	Phone Number	Zip
1	Luis	Gonçalves	+55 (12) 3923-5555	12227-000
2	Leonie	Köhler	+49 0711 2842222	70174



Table Aliases

- Tables can also be assigned an alias.
- The syntax for tables is the same as for columns.
- You can prefix columns names throughout a query with the table alias followed by a period.
- Table aliases become useful when working with multiple tables.

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




SELECT statement execution order

SELECT as Written

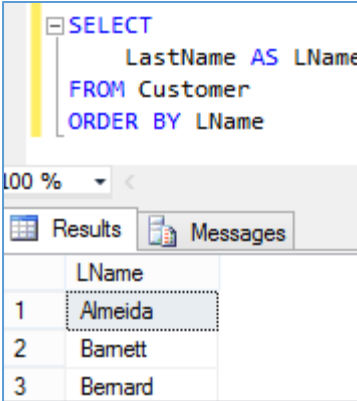
1. SELECT
2. FROM
3. WHERE
4. ORDER BY

SELECT as Executed

1. FROM
2. WHERE
3. SELECT
4. ORDER BY

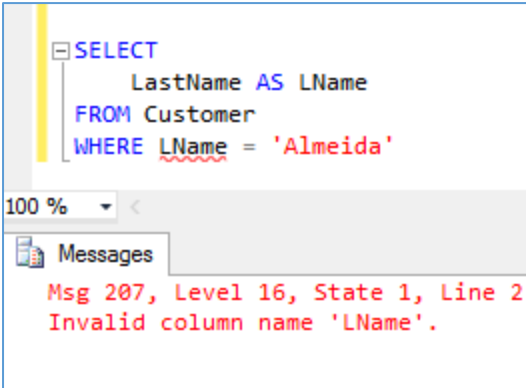
Why Execution Order is Important

- Intellisense 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)



```
SELECT
    LastName AS LName
FROM Customer
ORDER BY LName
```

	LName
1	Almeida
2	Barnett
3	Bernard



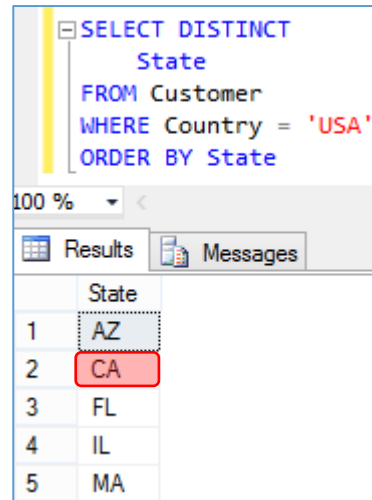
```
SELECT
    LastName AS LName
FROM Customer
WHERE LName = 'Almeida'
```

Msg 207, Level 16, State 1, Line 2
Invalid column name 'LName'.

SELECT DISTINCT

- Add the DISTINCT keyword immediately after the SELECT clause if you wish to remove duplicate values from the result set
- If there are multiple columns in your result set, the entire row must be a duplicate for it to be removed

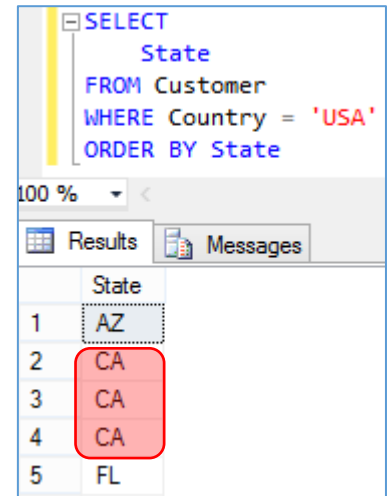
With Distinct



The screenshot shows a SQL query window with the following text: `SELECT DISTINCT State FROM Customer WHERE Country = 'USA' ORDER BY State`. Below the query, the 'Results' tab is active, displaying a table with one column 'State' and five rows. The values are AZ, CA, FL, IL, and MA. The row containing 'CA' is highlighted with a red border.

	State
1	AZ
2	CA
3	FL
4	IL
5	MA

Without Distinct



The screenshot shows a SQL query window with the following text: `SELECT State FROM Customer WHERE Country = 'USA' ORDER BY State`. Below the query, the 'Results' tab is active, displaying a table with one column 'State' and five rows. The values are AZ, CA, CA, CA, and FL. The rows containing 'CA' (rows 2, 3, and 4) are highlighted with a red border, indicating that duplicates are not removed.

	State
1	AZ
2	CA
3	CA
4	CA
5	FL

SELECT TOP

- Use the TOP keyword and a number after SELECT to limit the number of rows returned
- The ORDER BY clause determines the order records are displayed when using the TOP keyword
- 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

```
SELECT
    State
FROM Customer
WHERE Country = 'USA'
ORDER BY State
```

100 % < >

Results Messages

	State
1	AZ
2	CA

\\Eric (52) | Chinook | 00:00:00 13 rows

```
SELECT TOP 3
    State
FROM Customer
WHERE Country = 'USA'
ORDER BY State
```

100 % < >

Results Messages

	State
1	AZ
2	CA

\\Eric (52) | Chinook | 00:00:00 3 rows

```
SELECT TOP 10 PERCENT
    State
FROM Customer
WHERE Country = 'USA'
ORDER BY State
```

100 % < >

Results Messages

	State
1	AZ
2	CA

\\Eric (52) | Chinook | 00:00:00 2 rows

SELECT TOP... WITH TIES

- Use the WITH TIES keywords to return all records if there is a tie for the last spot
- Ties are determined by the ORDER BY clause
- You can add an additional columns or expressions to the ORDER BY clause to break potential ties

```
SELECT TOP 5  
    Country, CustomerId  
FROM Customer  
ORDER BY Country
```

	Country	CustomerId
1	Argentina	56
2	Australia	55
3	Austria	7
4	Belgium	8
5	Brazil	1

```
SELECT TOP 5 WITH TIES  
    Country, CustomerId  
FROM Customer  
ORDER BY Country
```

	Country	CustomerId
1	Argentina	56
2	Australia	55
3	Austria	7
4	Belgium	8
5	Brazil	1
6	Brazil	10
7	Brazil	11
8	Brazil	12
9	Brazil	13

```
SELECT TOP 5 WITH TIES  
    Country, CustomerId  
FROM Customer  
ORDER BY Country, CustomerId DESC
```

	Country	CustomerId
1	Argentina	56
2	Australia	55
3	Austria	7
4	Belgium	8
5	Brazil	13



Summary

- SELECT Clause
- FROM Clause
- WHERE Clause
- WHERE Syntax
 - Relational Operators
 - AND and OR
- ORDER BY Clause
- ASC and DESC
- Aliases
- DISTINCT
- TOP [PERCENT] [WITH TIES]
- Execution Order