

Cis 111 Chapter 3 Student Lecture Notes

Topic #1: SQL Select Statement

Once data is stored into a database a developer can design, code, test and implement an application program that will **retrieve data** from the database and display information to an end user. The SQL language contains the **Select** statement in order to accomplish this task. Here is an example of a simplified syntax of the Select statement:

The simplified syntax of the SELECT statement

```
SELECT select_list
FROM table_source
[WHERE search_condition]
[ORDER BY order_by_list]
```

The four clauses of the SELECT statement

- SELECT
- FROM
- WHERE
- ORDER BY

A **Select** statement **minimally** has to contain the **Select** and **From** clause in order for it to be a valid Select statement. That is why the Where and Order by clauses are contained in square brackets []

Topic #2: Select clause

The **Select clause** contains a **select list** which can be anyone of the following items:

Field of a table, string expression, numeric expression, SQL Server intrinsic function, and/or a constant

Select clause example	Meaning
Select *	Return all columns from a specified table. The column names will be the field names.
Select InvoiceNumber, InvoiceDate as InvDate, InvoiceTotal	Return 3 columns. The second column uses an alias which means that will be the column name in the query result set.
Select InvoiceTotal, CreditTotal + Payment as TotalCredits	Return 2 columns. The second column is a calculated numeric expression .
Select VendorContactFName + ' ' + VendorContactLName as FullName	Return 1 column. It is a string expression using the concatenation operator (+)
Select 'This is a test' as Col1, 100 as Col2	Returns 2 columns. The first column is a string constant while the second column is a numeric constant.
Select Left(VendorContactFName,1) + Left(VendorContactLName,1) as Initials	Returns 1 column. This is a string expression that uses a SQL Server intrinsic function

Cis 111 Chapter 3 Student Lecture Notes

Select Statement with query results...

Example 1:

```
SELECT *  
FROM Invoices;
```

	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	
1	1	122	989319-457	2015-12-08 00:00:00	3813.33	3813.33	0.00	^
2	2	123	263253241	2015-12-10 00:00:00	40.20	40.20	0.00	
3	3	123	963253234	2015-12-13 00:00:00	138.75	138.75	0.00	
4	4	123	2-000-2993	2015-12-16 00:00:00	144.70	144.70	0.00	v
< >								

(114 rows)

Example 2:

Two ways to name the columns in a result set

Using the AS keyword (the preferred technique)

```
SELECT InvoiceNumber AS [Invoice Number],  
       InvoiceDate AS Date, InvoiceTotal AS Total  
FROM Invoices;
```

Using the equal operator (an older technique)

```
SELECT [Invoice Number] = InvoiceNumber, Date =  
       InvoiceDate,  
       Total = InvoiceTotal  
FROM Invoices;
```

The result set for both SELECT statements

	Invoice Number	Date	Total	
1	989319-457	2015-12-08 00:00:00	3813.33	^
2	263253241	2015-12-10 00:00:00	40.20	
3	963253234	2015-12-13 00:00:00	138.75	
4	2-000-2993	2015-12-16 00:00:00	144.70	
5	963253251	2015-12-16 00:00:00	15.50	v

Cis 111 Chapter 3 Student Lecture Notes

Why the square brackets for the alias name of [Invoice Number]? If you code an identifier for a column name or a table name it will either be a valid **regular identifier** or a valid **delimited identifier**.

Regular Identifier

- ✓ The **first character** must be a letter, an underscore (_), an at sign (@), or a number sign (#)
- ✓ All characters **after the first** must be a letter, a number, an at sign, a dollar sign (\$), a number sign or an underscore
- ✓ An identifier cannot be a Transact-SQL reserved keyword. Transact-SQL is the SQL syntax defined by Microsoft's DBMS.
- ✓ An identifier cannot contain spaces or special characters other than those listed above.

Delimited Identifier

- ✓ A delimited identifier does not have to follow any of the above rules but must have either the square brackets [] or quotation characters as delimiters

Valid Regular Identifiers	Valid Delimited Identifiers
Employees	[%Increase]
#PaidInvoices	"Invoice Line Items"
@Total_Due	[Invoice Number]

Example 3:

How to include apostrophes in literal values

```
SELECT VendorName + ' 's Address: ',  
       VendorCity + ', ' + VendorState + ' ' + VendorZipCode  
FROM Vendors;
```

	(No column name)	(No column name)
1	US Postal Service's Address:	Madison, WI 53707
2	National Information Data Ctr's Address:	Washington, DC 20090
3	Register of Copyrights's Address:	Washington, DC 20559
4	Jobtrak's Address:	Los Angeles, CA 90025
5	Newbrige Book Clubs's Address:	Washington, NJ 07882
6	California Chamber Of Commerce's Ad...	Sacramento, CA 95827

Most of the time you will want to avoid **(No column name)** as a column heading.

Cis 111 Chapter 3 Student Lecture Notes

Example 4:

A SELECT statement that computes the age of an invoice

```
SELECT InvoiceDate,  
       GETDATE() AS 'Today's Date',  
       DATEDIFF(day, InvoiceDate, GETDATE()) AS Age  
FROM Invoices;
```

	InvoiceDate	Today's Date	Age
1	2016-04-02 00:00:00	2016-05-01	29
2	2016-04-01 00:00:00	2016-05-01	30
3	2016-03-31 00:00:00	2016-05-01	31

What do you need to know about SQL Server Intrinsic functions?

1. **Name** of function.
2. What **arguments** are needed between the parenthesis ()?
3. What does it **return**? Integer, decimal, string, datetime, etc...

Some of the examples you have reviewed in a select list used a numeric expression. What arithmetic operators are available in SQL?

Order Precedence – Use parenthesis to change the order.

1. * / %
2. + -

Operator	Meaning	Examples – Using constants instead of field names	Result
*	Multiplication	Select 3 + 5 * 3 as Expr1	
/	Division	Select 18 / 5 % 2 as Expr2	
%	Modulo – Integer remainder	Select 20 - 5 * 2 / 4.0 as Expr3	
+	Addition	Select 20 - 5 * 2 / 4 as Expr4	
-	Subtraction	Select (3 + 1) * (20 % 6 * 2) as Expr5	

Note: If the division operator (/) is dividing two integer values then the result is an integer (decimal value is dropped). If either value is a decimal then the result is a decimal value.

Cis 111 Chapter 3 Student Lecture Notes

Topic #3: From clause

The **From** clause indicates where is the source of data located? For now, it will only be one table. This becomes more complicated when two or more tables have to be used in a SQL Select statement. This will be discussed starting in chapter 4.

From clause example	Meaning
From Invoices	The source of data will come from the Invoices table.
From Invoices as Inv	You can refer to a table name using an alias. This is useful when subqueries are coded or a developer wants to abbreviate a long table name to a shorter name in their SQL code.

Topic #4: Where clause

A SQL **Where** clause allows you to filter records by coding a conditional expression. If a record meets the conditional expression (True) it will be included in the query result set. Remember, it does not matter how long or complicated the Where clause is coded, it will always return a **True** or **False** Boolean value.

What **comparison operators** can be used?

Comparison Operator	Meaning
=	Equals (it must be exact)
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
< >	Not equal to

Where clause examples	Meaning
Where VendorState= 'MI'	Return a query result set of Vendors from the state of Michigan.
Where VendorName < 'D'	Return a query result set of Vendors with a vendor name that begins with a, b or c.
Where InvoiceTotal – PaymentTotal – CreditTotal > 0	Return a query result set of invoices with a balance due greater than 0.
Where InvoiceDate < > '2016-05-31'	Return a query result set of all invoices that do not have an invoice date of 5-31-2016.

Cis 111 Chapter 3 Student Lecture Notes

What is the **IN** phrase of a Where clause?

If you have a scenario where you want to check if a **certain field** has **one or more specific values** then the IN phrase is very useful.

Where clause examples	Meaning
Where VendorState In('IN', 'OH', 'MI')	Return a query result set of Vendors from the state of Indiana, Ohio, or Michigan.
Where InvoiceDate In('1/13/2016', '1/14/2016', '1/15/2016')	Return a query result set of Invoices whose invoice date is on the 13 th , 14 th or 15 th of January 2016.
Where TermsID In(1,2,3)	Return records where terms id is either 1, 2 or 3.

What is the **Between** phrase of a Where clause?

If you have a scenario where you want to check if a **certain field** has **a range of values** then the Between phrase is very useful.

Where clause examples	Meaning
Where InvoiceTotal Between 1000 and 9000	Return a query result set of invoices where the invoice total is between and including 1000 to 9000.
Where InvoiceDate Between '1/1/2016' and '3/31/2016'	Return a query result set of Invoices for the 1 st quarter of 2016.
Where VendorName Between 'a' and 'c'	Return vendors whose vendor name starts with the letter 'a' or the letter 'b'.
Where InvoiceTotal-PaymentTotal-CreditTotal Between 200 and 300	Return all invoices with a balance due between and including 200 to 300.
Where InvoiceDueDate Between GetDate() and GetDate() + 30	Return all invoices whose due date is between today's date and 30 days from the current date.

Warning about date comparisons

- All columns that have the datetime data type include both a date and time, and so does the value returned by the GetDate function.
- When you code a date literal without a time, the time defaults to 12:00 AM (midnight). As a result, a date comparison may not yield the results you expect.

Assume a date field contains the following 2016-03-28 11:23:20.000. If a query where clause contains a date only value such as **Where InvoiceDate= '3/28/2016'** then a match would not occur since the time value used in the comparison would be 00:00:00.000

Cis 111 Chapter 3 Student Lecture Notes

What is the **Like** phrase of a Where clause?

If you have a scenario where you want to check if a **certain string field** has a **string pattern (or mask)** then the Like phrase can be coded.

Wildcard Symbols

Symbol	Description
%	Matches any string of zero or more characters.
_ (underscore character)	Matches any single character.
[]	Matches a single character listed within the brackets.
[-]	Matches a single character within a given range.
[^]	Matches a single character not listed after the caret.

Where clause examples	Meaning
Where VendorName Like 'A%'	Return vendors whose name starts with the letter "a".
Where VendorName Like '%a'	Return vendors whose name ends with the letter "a".
Where VendorName Like '%a%'	Return vendors whose name contains the letter "a".
Where VendorName Like 'a_b[fg]%'	Return vendors whose name starts with the letter "a", the third letter contains a "b", and the fourth contains a "f" or "g".
Where InvoiceDate Like '%2015%'	Return invoices for the year 2015.
Where VendorName Like '[a-d]%'	Return vendor names which start with "a" or "b" or "c" or "d".
Where VendorName Like '[^a-d]%'	Return vendor names which do not start with "a" or "b" or "c" or "d".
Where VendorName Like 'C'	No wildcard symbols are used, therefore, this will be interpreted as VendorName = 'C'.

Cis 111 Chapter 3 Student Lecture Notes

What about the **Not** operator?

The Not operator exists for the SQL Where clause which has an opposite meaning to an expression.

Where clause examples	Meaning
Where VendorState Not In('OH', 'MI')	Return all vendors except for the state of Ohio or Michigan.
Where VendorName Not Like 'A%'	Return vendors whose name does not start with the letter "A".
Where InvoiceTotal Not Between 1000 and 9000	Return invoice totals that are less than 1000 or greater than 9000.

Each example above can be coded to have the Not operator right after the Where clause:

Where Not VendorState In('OH', 'MI')
Where Not VendorName Like 'A%'
Where Not InvoiceTotal Between 1000 and 9000

What about using the **IsNull** clause in the Where clause?

The following is a definition found in Wikipedia for a **Null** value:

Null is a **special marker** used in Structured Query Language (SQL) to indicate that a **data value does not exist** in the database. Introduced by the creator of the relational database model, E. F. Codd, SQL Null serves to fulfill the requirement that all true relational database management systems (RDBMS) support a representation of "missing information and inapplicable information".

When a field is defined for a table it can be set up to accept a Null value or not. Data that is optional such as Address Line 2 would be an example of allowing a field to accept a null value.

Where clause examples	Meaning
Where PaymentDate Is Null	Return all invoices that have not been paid.
Where PaymentDate Not Is Null	This will not work . You will receive a syntax error.
Where Not PaymentDate Is Null	Return all invoices where the PaymentDate contains no data (Null value)

Cis 111 Chapter 3 Student Lecture Notes

What about coding a **Compound Condition** in the Where clause?

SQL allows you to code Where clauses using compound conditions. The **logical operators** that exist are listed in **order of precedence**: **NOT, AND, OR**

Examples of queries using logical operators

The AND operator

```
WHERE VendorState = 'NJ' AND YTDPurchases > 200
```

The OR operator

```
WHERE VendorState = 'NJ' OR YTDPurchases > 200
```

The NOT operator

```
WHERE NOT (InvoiceTotal >= 5000 OR  
          NOT InvoiceDate <= '2016-07-01')
```

The same condition without the NOT operator

```
WHERE InvoiceTotal < 5000 AND InvoiceDate <= '2016-07-01'
```

A compound condition without parentheses

```
WHERE InvoiceDate > '01/01/2016'  
      OR InvoiceTotal > 500  
      AND InvoiceTotal - PaymentTotal - CreditTotal > 0
```

	InvoiceNumber	InvoiceDate	InvoiceTotal	BalanceDue
1	P02-88D77S7	2016-01-03 00:00:00	856.92	0.00
2	21-4748363	2016-01-03 00:00:00	9.95	0.00
3	4-321-2596	2016-01-05 00:00:00	10.00	0.00
4	963253242	2016-01-06 00:00:00	104.00	0.00

(100 rows)

The same compound condition with parentheses

```
WHERE (InvoiceDate > '01/01/2016'  
      OR InvoiceTotal > 500)  
      AND InvoiceTotal - PaymentTotal - CreditTotal > 0
```

	InvoiceNumber	InvoiceDate	InvoiceTotal	BalanceDue
1	39104	2016-03-10 00:00:00	85.31	85.31
2	963253264	2016-03-18 00:00:00	52.25	52.25
3	31361833	2016-03-21 00:00:00	579.42	579.42
4	263253268	2016-03-21 00:00:00	59.97	59.97

(11 rows)

Cis 111 Chapter 3 Student Lecture Notes

Topic #5: Order by clause

The Order by clause allows you to sort the query result set by one or more fields in either ascending or descending order.

Order by clause examples	Meaning
Order by VendorName	Sort the query result set by VendorName in ascending order (A to Z)
Order by VendorName Desc	Sort the query result set by VendorName in descending order (Z to A)
Order by VendorState, VendorCity, VendorName	Sort the query result using three sort fields: 1. VendorState 2. VendorCity 3. VendorName All in ascending order
Order by Address	You can sort the query result set using an alias. The assumption in this example is that Address is an alias column name.
Order by VendorContactLName + VendorContactFName	You can sort the query result set by a string expression, numeric expression or by using a SQL Server intrinsic function.
Order by 2, 1	You can sort the query result set by a column ordinal position. The primary sort key is column number 2 while the secondary sort key is column number 1.

Can the **Order by** clause retrieve a **range of rows**?

Please review the following examples...

An ORDER BY clause that retrieves the first five rows

```
SELECT VendorID, InvoiceTotal
FROM Invoices
ORDER BY InvoiceTotal DESC
OFFSET 0 ROWS
FETCH FIRST 5 ROWS ONLY;
```

Start with the first row

	VendorID	InvoiceTotal
1	110	37966.19
2	110	26881.40
3	110	23517.58
4	72	21842.00
5	110	20551.18

Retrieve the first 5 rows from the offset.

An ORDER BY clause that retrieves rows 11 through 20

```
SELECT VendorName, VendorCity, VendorState, VendorZipCode
FROM Vendors
WHERE VendorState = 'CA'
ORDER BY VendorCity
      OFFSET 10 ROWS
      FETCH NEXT 10 ROWS ONLY;
```

	VendorName	VendorCity	VendorState	VendorZipCode
1	Robbins Mobile Lock And Key	Fresno	CA	93726
2	BFI Industries	Fresno	CA	93792
3	California Data Marketing	Fresno	CA	93721
4	Yale Industrial Trucks-Fresno	Fresno	CA	93706
5	Costco	Fresno	CA	93711
6	Graylift	Fresno	CA	93745
7	Shields Design	Fresno	CA	93728
8	Fresno County Tax Collector	Fresno	CA	93715
9	Gary McKeighan Insurance	Fresno	CA	93711
10	Ph Photographic Services	Fresno	CA	93726

How can the **Distinct** keyword and the **Top** clause be used in the Select statement?

Even though there are many vendors located in the same state and city the following query statement will display all combinations of city and state only once...

A SELECT statement that eliminates duplicate rows

```
SELECT DISTINCT VendorCity, VendorState
FROM Vendors;
```

	VendorCity	VendorState
1	Anaheim	CA
2	Ann Arbor	MI
3	Auburn Hills	MI
4	Boston	MA
5	Brea	CA
6	Carol Stream	IL
7	Charlotte	NC
8	Chicago	IL

(53 rows)

Cis 111 Chapter 3 Student Lecture Notes

The next query example allows you to return a portion, or subset of records from a query result set.

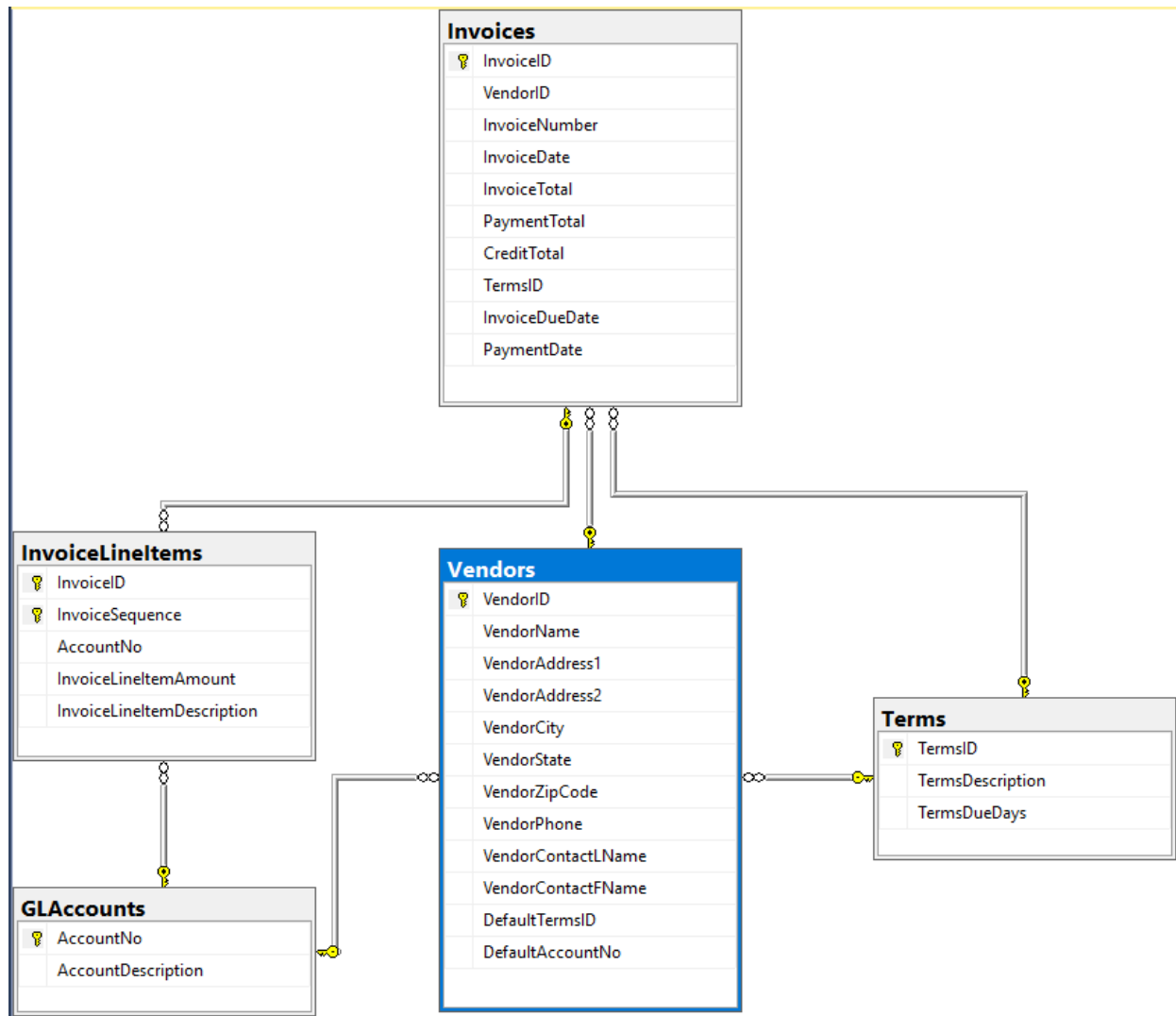
A SELECT statement with a TOP clause

```
SELECT TOP 5 VendorID, InvoiceTotal  
FROM Invoices  
ORDER BY InvoiceTotal DESC;
```

	VendorID	InvoiceTotal
1	110	37966.19
2	110	26881.40
3	110	23517.58
4	72	21842.00
5	110	20551.18

Cis 111 Chapter 3 Student Lecture Notes

Accounts Payable (AP) Database:



Lecture Lab Activity

You will now work on problems from chapter 3 on page 123.

Save the query script file as Lecture_Lab_Chap03.sql