



Basic SQL Server

Authored by : Sushant Banerjee
Email : sushantba@cybage.com

Presented by : Sushant Banerjee
Extn. 7210

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only. This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

Agenda

- Query Fundamentals
- Joins
- Subqueries
- Built-in Functions
- String Functions
- DateTime Functions



Query Fundamentals

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only. This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

What is a Query?

- Requests data from database
- Mostly SELECT statement is used
- Server returns data in Result Sets
- Manipulating result sets does not change actual table data

Identifiers

- Name of any object
 - Server name
 - Database name
 - Table name
 - Column name etc.

Classes of Identifiers

- Regular Identifiers
 - Follows the rules of identifiers
- Delimited Identifiers
 - Doesn't follow the rules of identifiers
 - Uses either quotation marks “ ” or brackets []

Rules for Regular Identifier

- The first character must use
 - Characters from A through Z or
 - The underscore (_), at sign (@) or number sign (#)
- The subsequent characters may include
 - Characters from A through Z
 - Dollar sign (\$) underscore (_), at sign (@) or number sign (#)
- The identifier must not use T-SQL reserved words.
- Any embedded space or special characters are not allowed.

The SELECT Statement

SELECT < Column List >
From < Table Name >
Where < Conditions >
Group By < Column Name >
Having < Condition >
Order By < Column Name >

Retrieving Data

- All columns and rows from a table
- Specific columns from a table
- Hide original column name using column alias
- Format result set using character string constant

Arithmetic Operators

- Arithmetic operators can be used in expressions.
 - Addition (+)
 - Subtraction (-)
 - Division (/)
 - Multiplication (*)
 - Modulo (%)

DISTINCT Keyword

- Eliminates duplicate rows
- NULL values are also considered

The TOP Clause

- The TOP clause is used to limit number of rows that are returned in the result set.
- You can specify either numeric expression as number of rows or percentage followed by the TOP clause.

The FROM Clause

- Used to specify table names in every SELECT statement.
- FROM clause is not required when the SELECT statement returns data from local or global variables and T-SQL functions.

The WHERE Clause

- The WHERE clause is used in the SELECT statement just like a filter.
- You can specify your search conditions in the WHERE clause and only those rows that meet the specified condition will return.
- A search condition may include following:
 - Comparison Operators (= , < , > etc.)
 - Ranges (BETWEEN and NOT BETWEEN)
 - Lists (IN and NOT IN)
 - Pattern Matches (LIKE and NOT LIKE)
 - Null Values (IS NULL and IS NOT NULL)
 - Combination of the conditions (AND, OR, NOT)

Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
< >	Not equal to (ISO compatible)
!>	Not greater than
!<	Not less than
!=	Not equal to

Range and Lists

- Used to specify certain range – BETWEEN, NOT BETWEEN
- Used to specify list of values – IN, NOT IN

Pattern Matches

- The LIKE keyword is used for pattern matching.
- The pattern may contain character string along with 4 wildcards.

Wildcards	Meaning
%	Any string of zero or more characters.
_	Any single character
[]	Any single character within the specified range
[^]	Any single character not within the specified range

NULL Values

- NULL means data value for the column is unknown or not available.
- NULL does not mean a numeric zero, zero length character string or blank.
- SQL Server automatically enters NULL value if no data is entered and there is no DEFAULT constraint defined for that column.
- IS NULL, IS NOT NULL can be used in the WHERE clause.

Logical Operators

- There are three logical operators such as AND, OR, and NOT.
- AND and OR are used to combine search conditions,
- NOT is used reverse the result of a search condition.
- Logical Operator Precedence - NOT is evaluated first then AND and finally OR.

Aggregate Functions

- COUNT
- AVG
- MIN
- MAX
- SUM

Grouping and Sorting

- ORDER BY (default ASC) Clause
- GROUP BY Clause
- HAVING clause is used with the GROUP BY clause to filter groups in the result set.

UNION and UNION ALL Operators

- The UNION operator will combine two result sets to produce a single result set.
- The UNION will select only distinct values, to allow duplicate values using ALL keyword with UNION operator.
- The restrictions to use UNION :
 - Both the SELECT statements must have same number and ORDER of columns.
 - The columns must have the similar data types.

EXCEPT and INTERSECT

- The EXCEPT will compare the result sets of two queries and returns distinct values.
- EXCEPT will return any distinct values from the left query that are not found on the right query.
- INTERSECT returns any distinct values that are returned by both the queries on the left and right sides of the INTERSECT operand.
- The basic rules for combining the result sets of two queries that use EXCEPT or INTERSECT are the following:
 - The number and the order of the columns must be the same in all queries.
 - The data types must be compatible.



Joins and Subqueries

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only. This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

JOINS

- You can use JOINS to retrieve data from two or more tables based on logical relationships between the tables.
- The JOIN condition needs following :
 - The column from each table used for the JOIN
 - A logical operator for comparing values from the columns.

Resolving Ambiguity

- When a same column name is used in more than one table in a join operation, the reference of the column becomes ambiguous.
- TableAlias.ColumnName
- **Fully qualified column names(4 parts name)**
- DatabaseName.SchemaName.TableName.ColumnName

JOINS (contd...)

- INNER JOIN
- OUTER JOIN
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL JOIN or FULL OUTER JOIN
- CROSS JOINS
- SELF JOINS

Subquery

- A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery.
- Up to 32 levels of nesting is allowed by SQL Server.
- A subquery can be used anywhere an expression is allowed.
- A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.
- Many Transact-SQL statements that include subqueries can be alternatively use joins

Types of Subquery

- Using Alias
- Using IN or NOT IN
- In UPDATE, DELETE and INSERT statements
- Using Comparison Operators
- Using ANY, ALL
- Using EXISTS or NOT EXISTS
- In place of an expression

Correlated Subquery

- A correlated subquery depends on the outer query for its evaluation.
- The subquery executes repeatedly, once for each row selected by the outer query.
- For this reason a correlated subquery is also called a repeating subquery.
- A correlated subquery can not evaluate itself independently of the outer query.

Subquery Restrictions

- The select list of a subquery introduced with a comparison operator can include only one expression or column.
- If the WHERE clause of an outer query includes a column name, it must be join-compatible with the column in the subquery select list.
- The **ntext**, **text**, and **image** data types cannot be used in the select list of subqueries.
- Subqueries introduced by an unmodified comparison operator (one not followed by the keyword ANY or ALL) cannot include GROUP BY and HAVING clauses.

Subquery Restrictions (contd...)

- ORDER BY can only be specified when TOP is also specified.
- A view created by using a subquery cannot be updated.
- A subquery introduced with EXISTS creates an existence test and returns TRUE or FALSE, instead of data.
- The DISTINCT keyword cannot be used with subqueries that include GROUP BY.



Built-in Functions

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only. This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

Built-in Functions

- In SQL Server, built-in functions are basically classified as:
- **Deterministic**
 - When a function always returns the same results for a specific set of input values.
- **Nondeterministic**
 - When a function may return different results when it is called repeatedly with the same set of input values, for example GETDATE.

Built-in Functions

- SQL Server provides many different types of built-in functions, out of those following are some of the useful types of functions :
 - System Functions
 - String Functions
 - Date and Time Functions
 - Conversions Functions
 - Aggregate Functions

System Functions

- System functions operate on or report on various system level options and objects.
- Following are some of the useful system functions:
 - DB_NAME and DB_ID
 - HOST_ID and HOST_NAME
 - OBJECT_ID and OBJECT_NAME
 - SUSER_ID and SUSER_NAME
 - USER_ID and USER_NAME

String Functions

- Following are some commonly used string functions:
- ASCII(), CHAR() and CHARINDEX()
- LEFT() and RIGHT()
- SUBSTRING()
- LEN()
- LOWER() and UPPER()
- LTRIM() and RTRIM()
- REPLACE(), REPLICATE() and REVERSE()
- SPACE() and STUFF()

Date and Time Functions

- GETDATE()
- DAY()
- MONTH()
- YEAR()
- DATEPART()
- DATENAME()
- DATEADD()
- DATEDIFF()

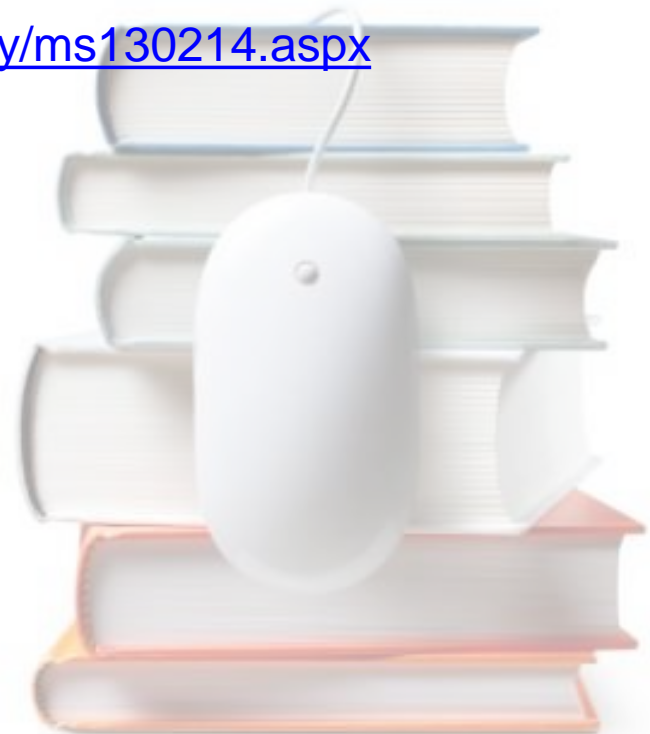
Conversion Functions

- Use these functions to convert expression from one data type to another data type.
- CAST()
- CONVERT()

Bibliography, Important Links

[WWW.MSDN.COM](http://www.msdn.com) (SQL SERVER 2012 BOOKS ONLINE)

<http://msdn.microsoft.com/en-us/library/ms130214.aspx>



Any Questions?



Thank you!