



SQL BASICS

Instructor: <Name of Instructor>

Latest updated by: HanhTT1

Agenda

- ❑ Data Definition Language (DDL)
- ❑ Data Manipulation Language (DML)
- ❑ Built-in Functions

Overview

- Data Definition Language (DDL)
 - Define data structures in SQL Server as creating, altering, and dropping tables and establishing constraints...
 - CREATE, ALTER, DROP, TRUNCATE... statements
- Data Manipulation Language (DML)
 - Retrieve and work with data in SQL Server
 - SELECT, UPDATE, INSERT, DELETE... statements
- Built-in Functions

SQL Overview

SQL Database Objects

- A SQL Server database has lot of objects like
 - Database
 - Schema
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - Rules
 - Defaults
 - Triggers

Data Definition Language Database

- SQL Server supports both scripts editor and graphic tool in order to
 - Create a database
 - Rename a database
 - Drop a database

Data Definition Language Database Demo

- Demo

See DDL_Database_Demo.docx

Data Definition Language Schema

- A Schema is a boundary within a database that enables you to logically group database objects such as tables, stored procedures and views.
 - **dbo** is default schema in every database
 - Schemas as naming boundaries
 - Schemas as security boundaries

Schemas as naming boundaries: For example, within a **Sales** database, you might want to logically separate tables that relate to internet sales from those that relate to in-store sales. To achieve this, you could create two schemas called **Internet** and **Instore**, and then create database objects in the appropriate schema. You should always use qualified names that include the schema name when referring to database objects to avoid errors and ambiguity. For example, to specify a table called **Orders** in the **Internet** schema in the **Sales** database, you should use the name **Sales.Internet.Orders**.

Schemas as security boundaries: For example, if you give the database role **SalesManagers** the SELECT permission on the **Internet** schema, they will have the SELECT permission on all of the tables in the schema and on all other objects in the schema to which the SELECT permission applies. You can enhance or override permissions granted at the schema level by setting permissions directly on schema objects. For example, to enhance permissions, you can grant a user SELECT permission on the **Internet** schema, and then grant them the UPDATE permission on the **Internet.Orders** table. To override permissions, you can use the DENY permission. DENY overrides GRANT permissions, so if you grant SELECT permission on the **Internet** schema, and then deny them the SELECT permission on the **Internet.Orders** table, they will not be able to select from the **Internet.Orders** table.

Data Definition Language Schema Demo

- Demo

Demo: DDL_Schema_Demo.docx

Data Definition Language Table

- Table is a repository for data, with items of data grouped in one or more columns
 - Data types
 - Constraints
 - Index

	EmployeeID	NationalIDNumber	ManagerID	Title	BirthDate	MaritalStatus	Gender	HireDate
1	1	14417807	16	Production Technician - WC60	1972-05-15 00:00:00.000	M	M	1996-07-31 00:00:00.000
2	2	253022876	6	Marketing Assistant	1977-06-03 00:00:00.000	S	M	1997-02-26 00:00:00.000
3	3	509647174	12	Engineering Manager	1964-12-13 00:00:00.000	M	M	1997-12-12 00:00:00.000
4	4	112457891	3	Senior Tool Designer	1965-01-23 00:00:00.000	S	M	1998-01-05 00:00:00.000
5	5	480168528	263	Tool Designer	1949-08-29 00:00:00.000	M	M	1998-01-11 00:00:00.000
6	6	24756624	109	Marketing Manager	1965-04-19 00:00:00.000	S	M	1998-01-20 00:00:00.000
7	7	309738752	21	Production Supervisor - WC60	1946-02-16 00:00:00.000	S	F	1998-01-26 00:00:00.000
8	8	690627818	185	Production Technician - WC10	1946-07-06 00:00:00.000	M	F	1998-02-06 00:00:00.000
9	9	695256908	3	Design Engineer	1942-10-29 00:00:00.000	M	F	1998-02-06 00:00:00.000

Data Definition Language Table

- Create table
- Alter table
 - Add new column
 - Change data type of existing column
 - Delete a column
 - Add or remove constraints
- Drop table
 - Remove table structure and its data.

Table Demo

- Demo
 - Create basic table
 - Alter table
 - Drop table

See “Create basic table” in DDL_Table_Demo.docx

Data Definition Language Table Constraints

- **Table Constraints:** Are used to limit the type of data that can go into a table.
- We will focus on the following constraints:
 - NOT NULL
 - CHECK
 - UNIQUE
 - PRIMARY KEY
 - DEFAULT
 - FOREIGN KEY

Data Definition Language Table Constraints (1)

- **NOT NULL:** Specifies that the column does not accept NULL values.
- **CHECK:** Enforce domain integrity by limiting the values that can be put in a column.
 - **Syntax:**
[CONSTRAINT *constraint_name*]
CHECK (*condition*)

Data Definition Language Table Constraint (2)

- **UNIQUE:** Enforce the uniqueness of the values in a set of columns
 - Syntax:
`CONSTRAINT unique_name UNIQUE (col_names)`
- **PRIMARY KEY:** Specify primary key of table.
 - Syntax:
`[CONSTRAINT PK_Name]
PRIMARY KEY [col_names]`

Data Definition Language Table Constraint (3)

- **FOREIGN KEY:** Used to define relationships between tables in the database.
 - Syntax:

```
[CONSTRAINT FK_Name]
FOREIGN KEY [(col_names)]
REFERENCES reference_table(col_names)
```
- **DEFAULT:** Defaults specify what values are used in a column if you do not specify a value for the column when you insert a row.

SQL Constraints Scope

- SQL constraints can be applied at:
 - Table level
 - Are declared independently from the column definition
 - declare table-level constraints at the end of the CREATE TABLE statement
 - Column level:
 - Are declared when define columns for the table.
 - It is applied particularly to the column where it attached to

Table constraint Demo

- Demo

See “Table Constraints” in DDL_Table_Demo.docx

Data Definition Language Table Indexes

- Index in SQL Server
 - Similar to index in books
 - Allow find data in a table without scanning the entire table
- There are 2 types of Indexes:
 - Clustered
 - Sort and store the data rows in the table based on their key value
 - Non-clustered
 - Have a structure completely separate from the data rows

Table Indexes Demo

- Demo

See “Table Indexes” in DDL_Table_Demo.docx

Data Definition Language Sequences

- This creates an auto increment for a column
- If a table has a column with sequence or auto increment, the user do not need insert data explicitly for the column

Data Definition Language Identity

- Identity has:
 - A seed
 - An increment
- Seed is the initial value
- Increment is the value by which we need to skip to fetch the next value
- For example:
 - Identity(1,2) will generate sequence numbers 1,3,5,7...

Data Definition Language Identity Demo

Demo

© FPT Software

22

See “Identity demo” in DDL_Table_Demo.docx

Data Definition Language

Truncate statement

- Removes all rows in a table.
 - Table structure and its columns, constraints, indexes, ...remain.
- Resets the identity value.
- Releases the memory used.

Data Definition Language Truncate Demo

- Demo

See “Truncate demo” in DDL_Table_Demo.docx

Data Definition Language Views Overview

- Views are logical tables
- The fields in a view are fields from one or more real tables in the database.
- A view is used to do:
 - Restrict a user to specific rows in a table
 - Restrict a user to specific columns

Data Definition Language Views Definition & Manipulation

- Create views:

CREATE VIEW viewname AS select stmt

Sample: Create view view_emp as select empid,
empname from employee

- Select from views:

SELECT * FROM viewname

Sample: Select empid,empname view_emp

- Drop views:

DROP VIEW viewname

Sample: Drop view view_emp

Data Manipulation Language

Insert statements 1/2

- Adds one or more rows to a table or a view
 - Inserting data to all columns
 - Inserting data to selected columns

```
insert into Student values(1,'Ramu')  
insert into Student(sid,sname) values(6,'Raj')
```

```
insert into Student(sid) values(2)  
insert into Student(sname) values('Seetha')
```

Data Manipulation Language Insert Demo

- Demo
 - Inserting data to selected columns
 - Inserting data to all columns with identity column
 - Insert many rows at one time

See “Insert demo” in DML_Demo.docx

Data Manipulation Language UPDATE

- Changes existing data in a table or view
- Syntax

```
UPDATE table_name  
SET Col_name = Value  
[WHERE condition]
```

Data Manipulation Language UPDATE Demo

- Demo

See “Update demo” in DML_Demo.docx

Data Manipulation Language DELETE

- Removes one or more rows from a table or view
- Syntax

**DELETE [FROM] table_Name
[WHERE condition]**

Data Manipulation Language DELETE Demo

- Demo

See “Delete demo” in DML_Demo.docx

Data Manipulation Language SELECT Statement 1/4

- Retrieves rows from the database and enables the selection of one or many rows or columns from one or many tables

- **Syntax:**

```
SELECT [ALL/DISTINCT/TOP] <Column name1>, <Column  
name2>, ...  
FROM <Table name>  
[WHERE <Search condition>]  
[GROUP BY grouping columns]  
[HAVING search condition]  
[ORDER BY sort specification]
```

To execute a statement in MS SQL, Select the statement and Click on the Execute button in the query analyser or press F5

Data Manipulation Language SELECT Statement 2/4

- Eg. Select * from table1;
 - This will fetch all rows and all columns from table1
- Eg. Select col1,col2 from table1
 - This will fetch col1 and col2 from table1 for all rows
- Eg. Select * from table1 where <>condn<>
 - This will fetch all rows from table1 that satisfies a condition
- Eg. Select col1,col2 from table1 where <>condn<>
 - This will fetch col1 and col2 of rows from table1 that satisfies a condition

Data Manipulation Language SELECT Statement 3/4

- The SELECT DISTINCT / TOP statements
 - DISTINCT: List only the different (distinct) values in a table.
 - TOP: Specify the number of records to return.
- The SELECT INTO statement selects data from one table and inserts it into a different table.
 - Syntax:

```
SELECT *
INTO new_table_name
FROM old_tablename
```

The TOP clause can be very useful on large tables with thousands of records. Returning a large number of records can impact on performance.

Data Manipulation Language SELECT Statement 4/4

- SQL Alias syntax:

- For table

```
SELECT column_name(s)  
FROM table_name AS alias_name
```

- For Column(s)

```
SELECT column_name AS alias_name  
FROM table_name
```

Data Manipulation Language SELECT Demo

- Demo

See “Select demo” in DML_Demo.docx

Data Manipulation Language SQL Operators

- Arithmetic : +, -, *, /, %
- Assignment : =
- Comparison : <, >, <=, >= <>, =, !=, !<, !>
- Logical : AND, OR, NOT, IN (set), LIKE, BETWEEN...
AND ..., ANY, ALL, EXISTS, SOME
- Set Operator: Except, INTERSECT, UNION
- String : Concatenation (+, +=)
- Unary : -, +, ~
- Bitwise: &, |, ^

Data Manipulation Language SQL Operators Demo

- Demo

See “SQL Operators” in DML_Demo.docx

Data Manipulation Language SELECT Options

- Aggregate functions :
 - Sum(col1): sum of data in the column col1
 - Max(col1): data with maximum value in col1
 - Min(col1): data with minimum value in col1
 - Avg(col1): Average of data in col1
 - Count(col1): Number of not null records in table
- Grouping – Group by col1 : Groups data by col1
- Ordering – Order by col1 : Orders the result in ascending order (default order) of col1
- Filtering – Where <<condn>> and Having <<condn>>

To select distinct rows, we need to use the distinct key word

Select distinct name from orders;

Orders

Id Name

1 Ram
2 Krish
3 Ram
4 Raj

Will fetch

Ram

Krish

Raj

Select count(name) from orders;

will yield the result as 4

40

Data Manipulation Language SELECT Options Demo

- Demo

See “Select demo” in DML_Demo.docx

Built-in Functions

- **Conversion Functions**
 - Converts an expression of one data type to another
 - CAST and CONVERT
- **Date and Time Functions**
 - GETDATE()
 - DATEPART, DATENAME
 - DAY, MONTH, YEAR
 - DATEADD, DATEDIFF
- **String Functions**
 - RTRIM, LTRIM
 - SUBSTRING
 - LEN
 - CHARINDEX, PATINDEX
 - REPLACE

Built-in Functions Demo

- Demo

See “Built-in_Function_Demo.docx”



What are different between DELETE and TRUNCATE statement?

What are different between UNION and UNION ALL?

What is DDL, DML?