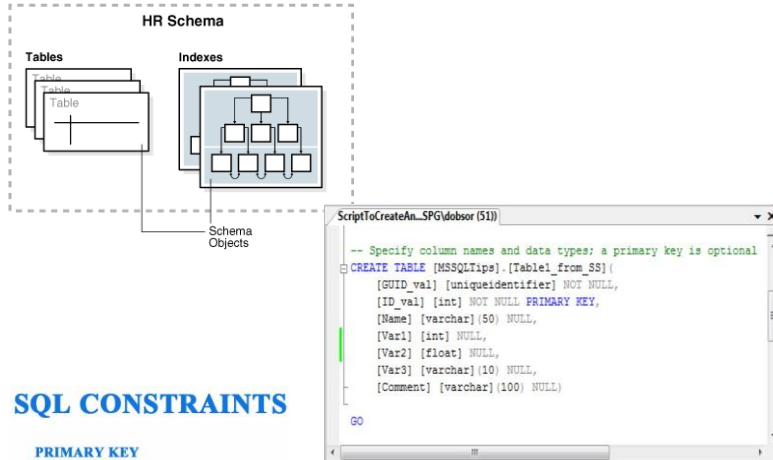


DDL STATEMENTS

Learning Goals

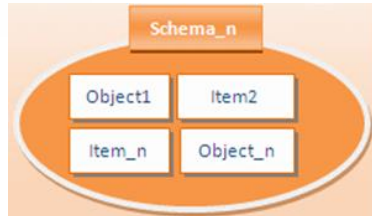
By the end of this lecture students
should be able to:

Categorize the main database objects



SQL CONSTRAINTS

PRIMARY KEY
FOREIGN KEY
UNIQUE KEY
NOT NULL
CHECK
DEFAULT



Create a simple table

Understand how constraints are created at the time of table creation

Describe how schema objects work

Understand and use to be commands create, alter, drop, truncate table

Table of contents

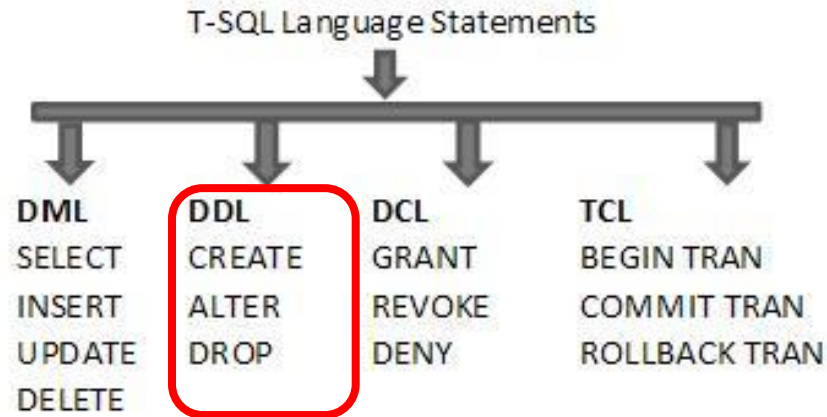
- **Introduction to DDL Statements**
- **Database Object**
- **Schema Object**
- **Table and Constraints**
- **Truncate**
- **View**

Section1

INTRODUCTION TO DDL STATEMENTS

Introduction to DDL Statements

- **DDL** stands for **D**ata **D**efinition **L**anguage
- Define data structures in SQL Server as creating, altering, and dropping tables and establishing constraints...

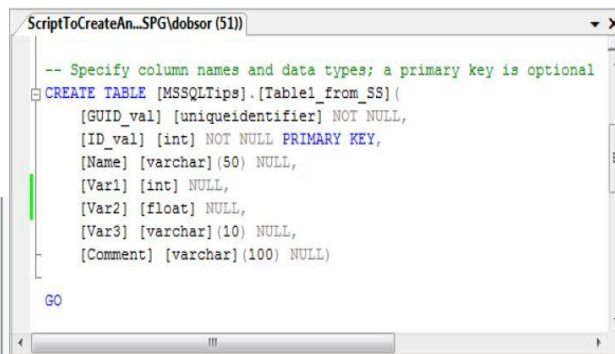
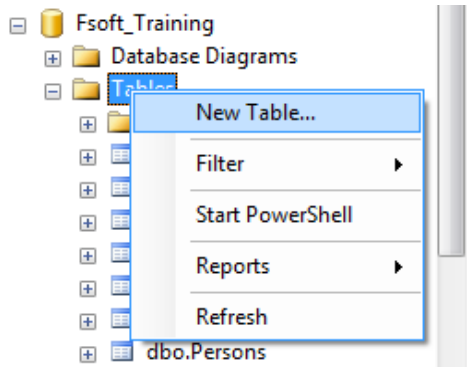


- **A SQL Server database has lot of objects like:**
 - Database
 - Schema
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - Rules
 - Defaults
 - Triggers

Section2

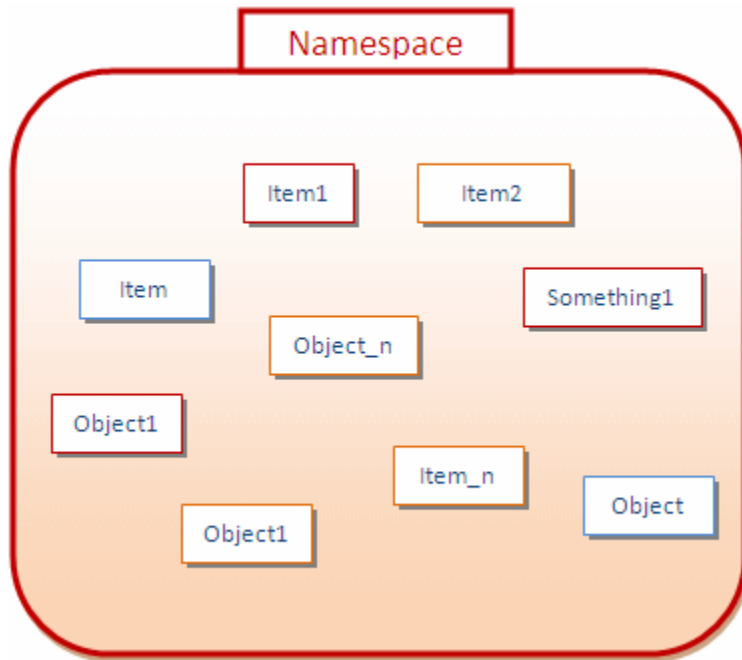
DATABASE & SCHEMA OBJECTS

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

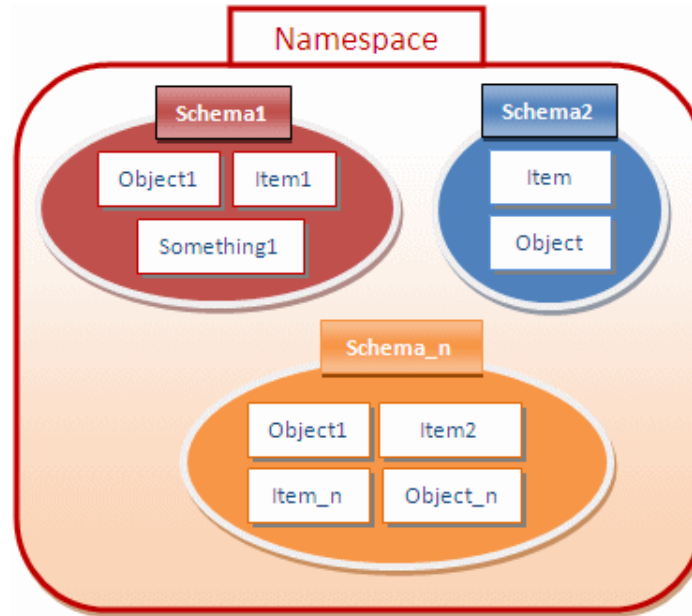


- **Scripts editor:**
 - Create a database
 - Rename a database
 - Drop a database
- **Graphic tool**
 - Create a database
 - Rename a database
 - Drop a database
- **Create database by using a template**

- A **namespace** can have objects inside



- To further control and manage the objects inside of a namespace, you can put them in *sub-groups* called **schemas**.



- **Schema default:**
 - **dbo** is default schema in every database
 - Ex: SalesOrderDetail, HumanResources.Department
 - **[linked-server].[DBName].[SchemaName].[Objectname]**
- **Schema as:**
 - naming boundaries
 - security boundaries

Section3

TABLE AND CONSTRAINTS

- 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

- **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 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

- **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*)

- **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]

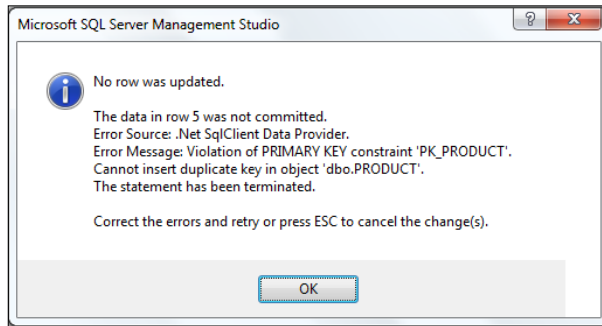
- **FOREIGN KEY:** Used to define relationships between tables in the database.
 - **Syntax:**
[CONSTRAINT *FK_Name*]
FOREIGN KEY [(*col_names*)]
REFERENCES *reference_table*(*col_names*)
- **DEFAULT:** Defaults specify what values are used in a column if you do not specify a value for the column when you insert a row.

- SQL constraints 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

Primary key

	PRODUCT_ID	PWIDTH	PLENGTH	PRICE
	1	40	50	2000.0000
	2	45	55	2000.0000
	3	40	60	3000.0000
	4	50	55	2500.0000
	4	45	50	2100
*	NULL	NULL	NULL	NULL

Error



Auto increment



	PRODUCT_ID	PWIDTH	PLENGTH	PRICE
	1	40	50	2000.0000
	2	45	55	2000.0000
	3	40	60	3000.0000
	4	50	55	2500.0000
	5	45	50	2100.0000
**	NULL	NULL	NULL	NULL

- 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...

1
2
3
4
5
...

Identity(1,1)

1
4
7
10
13
...

Identity(1,3)

1
3
5
7
9
...

Identity(1,2)

Truncate statement

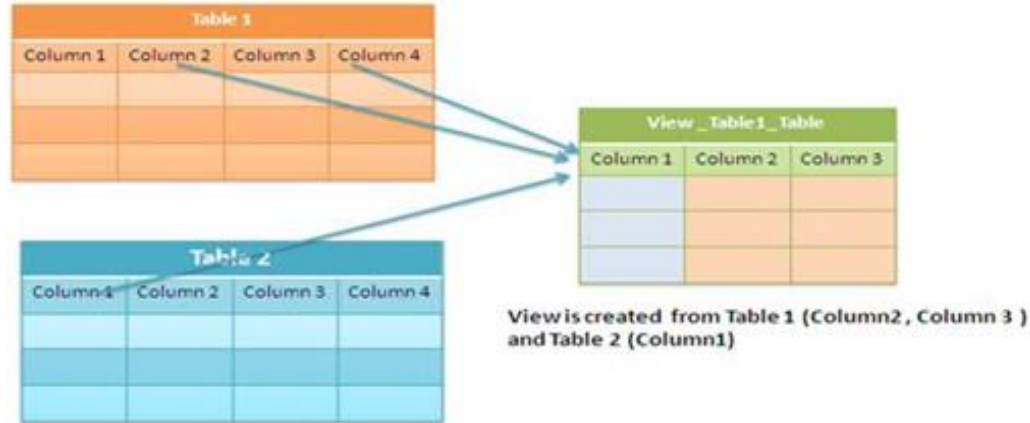
- Removes all rows in a table.
- Table structure and its columns, constraints, indexes, ...remain.
 - Resets the identity value.
 - Releases the memory used.

Section4

VIEWS

What is a view?

- A **View** is a logical or virtual table. The fields in a view are fields from one or more real tables in the database.
- There are **two major reasons** you might want to use views:
 - ✓ Views allow you to limit the data users can access
 - ✓ Views reduce complexity for end users.



Creating a view

```
CREATE VIEW View_Name [list of column names]
AS
SELECT...
```

Example:

```
CREATE VIEW view_EmployeeByDpt
AS
SELECT ID, NAME, AGE, DEPT_NAME
FROM EMP, DEPARTMENT
WHERE EMP.DEPT_ID = DEPARTMENT.DEPT_ID
```

Table: EMP

ID	NAME	AGE	DEP_ID
1	John	25	3
2	Mike	30	2
3	Parm	25	1
4	Todd	23	4
5	Sara	35	1
6	Ben	40	3

Table: DEPARTMENT

DEPT_ID	DEPT_NAME
1	IT
2	Payroll
3	HR
4	Admin

```
SELECT * FROM view_EmployeeByDpt
```

view_EmployeeByDpt

ID	NAME	AGE	DEPT_NAME
1	John	25	HR
2	Mike	30	Payroll
3	Parm	25	IT
4	Todd	23	Admin
5	Sara	35	IT
6	Ben	40	HR

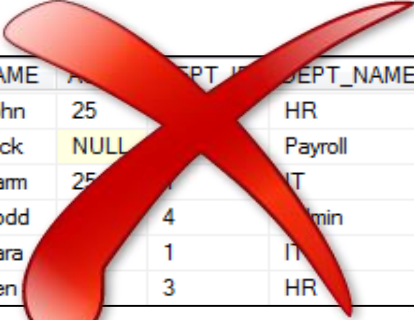
Deleting a view

- Syntax:

```
DROP VIEW View_Name
```

- Example:

```
DROP VIEW view_EmployeeByDpt
```



ID	NAME	AGE	DEPT_ID	DEPT_NAME
1	John	25	1	HR
2	Nick	NULL	2	Payroll
3	Pam	25	3	IT
4	Todd	4	4	Admin
5	Sara	1	1	IT
6	Ben	3	3	HR

view_EmployeeByDpt

✓ Introduction to DDL Statements

- ⑩ SQL Server Database Objects

✓ Database Object

- ⑩ Create, Rename, Drop a database: Graphic, Scripts, Template

✓ Schema Object

- ⑩ What is schema in database? Schema default?

✓ Table and Constraints

- ⑩ Create, Alter, Drop Table. NOT NULL, CHECK, UNIQUE, PRIMARY KEY, DEFAULT, FOREIGN KEY

✓ SQL View