DDL and DML Statements

Authored and presented by : Sushant B.

Cybage®
Delivering Value. Scientifically.

Agenda

- Introduction to T-SQL
- Working With Databases
- Working With Tables
- Working With Data
- Defining Data Integrity

Cybage® Delivering Value. Scientifically.

T-SQL Statements

- Transact SQL(T-SQL) is Microsoft's implementation of SQL
- Supports following operations

- DDL – Data Definition Language
  - CREATE, ALTER, DROP Statements

- DML – Data Manipulation Language
  - SELECT, INSERT, UPDATE, DELETE Statements

- DCL – Data Control Language
  - GRANT, REVOKE, DENY Statements

Working With Databases

System Databases

- master
  - Records all system level information
- model
  - A template for all new databases
- msdb
  - SQL Server Agent schedules jobs and alerts
- tempdb
  - Holds temporary objects or intermediate results

Cybage®
Delivering Value. Scientifically.

Creating Database

- Creating database without optional parameters
- A copy of model database is created.

```
CREATE DATABASE TestDB
```

Creating Database

- Creating database with optional parameters

```
CREATE DATABASE TestDB2
ON PRIMARY
(
NAME=MyDB,
FILENAME='C:\Demo\TestDB2.mdf',
SIZE=5MB,
MAXSIZE=1GB,
FILEGROWTH=10%
)
LOG ON
(
NAME=MyDBLog,
FILENAME='C:\Demo\TestDB2.ldf',
SIZE=2MB,
MAXSIZE=50MB,
FILEGROWTH=1MB
)
```

Cybage® Delivering Value. Scientifically.

Modifying Database

- Changing name of the database

```
ALTER DATABASE TestDB2
Modify Name = TestDB3
```

- Modifying file size

```
ALTER DATABASE TestDB3
modify file
(name=MyDB, size=10MB, maxsize=100MB, filegrowth=0)
```

- Deleting a database

```
DROP DATABASE TestDB3
```

Displaying Database Details

- Displaying single database details

```
execute sp_helpdb TestDB
```

- Displaying all database details

```
execute sp_databases
```

- Querying sysdatabases

```
SELECT * FROM sys.sysdatabases
```

Cybage®
Delivering Value. Scientifically.

Demo

Working With Tables

Creating Tables

- Connecting query editor to a database

```
USE TestDB
```

- Creating table in above database

```
CREATE TABLE dbo.Department
(
    DepartmentID int PRIMARY KEY NOT NULL,
    Name char(30) NOT NULL,
    Location char(30) NULL
)
```

Cybage®
Delivering Value. Scientifically.

Displaying Table Details

```
--Returns details of a table
execute sp_help 'dbo.Department'

--Returns all tables in current database
SELECT * FROM sys.tables
```

Cybage® Delivering Value. Scientifically.

Modifying Table

```
--Adding new column
ALTER TABLE dbo.Department
ADD DepartmentHead CHAR(30) NULL

--Modifying data type and size of a column
ALTER TABLE dbo.Department
ALTER COLUMN DepartmentHead VARCHAR(20) NULL

--Modifying name of a column
--Passing parameters(old name, new name, object type)
EXEC sp_rename 'dbo.Department.DepartmentHead', 'HOD', 'COLUMN'
```

- Can’t change order of columns

Cybage® Delivering Value. Scientifically.

Removing Columns

```
--Removing a column
ALTER TABLE dbo.Department
DROP COLUMN HOD

--Removing more than one columns
ALTER TABLE dbo.Department
DROP COLUMN Location, HOD
```

Cybage®
Delivering Value. Scientifically.

Demo

Working With Data

Cybage® Delivering Value. Scientifically.

Inserting Data

```
--inserting data without column list
INSERT INTO dbo.Department
VALUES(1, 'Process', 'Pune')

--inserting data with column list
--allows changing column sequence
INSERT INTO dbo.Department(Name, DepartmentID, Location)
VALUES('HR', 2, 'GNR')

--inserting data with null values
INSERT INTO dbo.Department(Name, DepartmentID, Location)
VALUES('Finance', 3, null)
```

Cybage® Delivering Value. Scientifically.

Updating Data

```
--updating single column values
UPDATE dbo.Department
SET Name = 'Marketing'
WHERE DepartmentID = 3

--updating multiple column value
UPDATE dbo.Department
SET Name = 'Finance', Location = 'HYD'
WHERE DepartmentID = 3
```

Reading Data

```
--Returns all columns and all rows
--Expected result changes in case new column added
SELECT * FROM Department

--Returns all rows from specified column list
SELECT DepartmentId, Name, Location
FROM Department

--Returns limited rows after filtering
SELECT DepartmentId, Name, Location
FROM Department
WHERE Location = 'Pune'
```

Cybage® Delivering Value. Scientifically.

Delete Data

```
--Delete a specific row
DELETE FROM dbo.Department
WHERE DepartmentID = 3

--Delete a range of rows
DELETE FROM dbo.Department
WHERE DepartmentID > 2 AND DepartmentID <= 4

--Deleting all rows
DELETE FROM dbo.Department

--Deleting all rows with truncate
TRUNCATE TABLE dbo.Department
```

Cybage®
Delivering Value. Scientifically.

Demo

Defining Data Integrity

Data Integrity

- Entity integrity
  - PRIMARY KEY and UNIQUE constraints, UNIQUE indexes
- Domain integrity
  - CHECK, DEFAULT, FOREIGN KEY and NOT NULL constraints
- Referential integrity
  - FOREIGN KEY constraint
- User-defined integrity

Enforcing Data Integrity

- Enforcing data integrity by applying constraints

- PRIMARY KEY
- FOREIGN KEY
- UNIQUE
- CHECK
- DEFAULT
- NOT NULL

Creating Primary and Foreign Keys

```
--Creating Primary Key
CREATE TABLE dbo.Department
(
    DepartmentID INT PRIMARY KEY NOT NULL,
    Name char(30) NOT NULL,
    Location char(30) NULL
)

--Creating Foreign Key
CREATE TABLE dbo.Employee
(
    EmployeeID INT PRIMARY KEY NOT NULL,
    Name CHAR(30) NOT NULL,
    City CHAR(30) NULL,
    DepartmentId INT REFERENCES dbo.Department(DepartmentId)
    ON DELETE CASCADE ON UPDATE CASCADE
)
```

Creating Unique and Check Constraints

```
--Creating table with unique,check and default
CREATE TABLE dbo.Employee
(
    EmployeeID INT PRIMARY KEY NOT NULL,
    Name CHAR(30) NOT NULL,
    City CHAR(30) NULL DEFAULT('PUNE'),
    Phone BIGINT NULL,
    DepartmentId INT REFERENCES dbo.Department(DepartmentId)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT chk_city CHECK(City in('PUNE','HYD','GNR')),
    CONSTRAINT unq_phone UNIQUE(Phone)
)
```

Adding Constraint to Existing Tables

```
--Adding primary key
ALTER TABLE dbo.Employee
ADD CONSTRAINT pk_prodid PRIMARY KEY(EmployeeID)

--Adding foreign key
ALTER TABLE Employee
ADD CONSTRAINT fk_deptid foreign key(DepartmentID)
REFERENCES Department(DepartmentID)

--Adding check constraint
ALTER TABLE emp2 WITH NOCHECK
ADD CONSTRAINT chk_city CHECK(City in('PUNE','HYD','GNR'))
```

Cybage®
Delivering Value. Scientifically.

Demo

Summary

- DDL Statements
- DML Statements
- Data Integrity
- Constraints

Bibliography, Important Links

- https://msdn.microsoft.com/en-us/library/bb510741.aspx
- https://msdn.microsoft.com/en-us/library/ms178028(v=sql.120).aspx
- https://msdn.microsoft.com/en-us/library/ms365340.aspx
- https://msdn.microsoft.com/en-us/library/ms176061.aspx
- https://msdn.microsoft.com/en-US/library/ms174269.aspx
- https://msdn.microsoft.com/en-us/library/ms178613.aspx
- https://msdn.microsoft.com/en-in/library/ms178568.aspx
- https://msdn.microsoft.com/en-us/library/ms365315(v=sql.120).aspx
- https://msdn.microsoft.com/en-us/library/ms189084(v=sql.120).aspx
- https://msdn.microsoft.com/en-IN/library/ms190273.aspx
- https://msdn.microsoft.com/en-us/library/ms365309(v=sql.120).aspx
- https://msdn.microsoft.com/en-us/library/ms365310(v=sql.120).aspx
- https://msdn.microsoft.com/en-in/library/ms189835.aspx
- https://msdn.microsoft.com/en-US/library/ms179610(v=sql.120).aspx
- https://msdn.microsoft.com/en-US/library/ms190024(v=sql.120).aspx

Any Questions?

Email : sushantba@cybage.com
Extn : 7221

Thank you!