

Stored Procedure

```
CREATE OR ALTER PROCEDURE dbo.usp_Name(@param)
AS
BEGIN
    SET NOCOUNT ON;
    SET XACT_ABORT ON;
    BEGIN TRANSACTION
    COMMIT
    SET XACT_ABORT OFF;
    SET NOCOUNT OFF;
END
```

Add not null column to existing table

```
IF NOT EXISTS(SELECT 1 FROM sys.columns WHERE Name =
N'MyNewColumn' AND Object_ID = Object_ID(N'[dbo].[MyTable]'))
BEGIN
ALTER TABLE MyTable
    ADD MyNewColumn INT NOT NULL DEFAULT 0
END
```

drop column

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

Detect duplicate

```
SELECT *, ROW_NUMBER() OVER(PARTITION BY TenantId
ORDER BY Id) from dbo.[Test] WHERE [Name] is null
```

Check Exists

```
IF [NOT] EXISTS ( SELECT 1 FROM MyTable WHERE ... )
    <do smth>
IF NOT EXISTS ( SELECT 1 FROM Users WHERE FirstName =
'John' AND LastName = 'Smith' )
BEGIN
    INSERT INTO Users (FirstName, LastName) VALUES
('John', 'Smith')
END
```

Drop temp table

```
DROP TABLE IF EXISTS #Results
```

If else

```
IF Boolean_expression
    { sql_statement | statement_block }
[ ELSE
    { sql_statement | statement_block } ]
```

Merge

```
MERGE <target_table> AS tgt
USING <table_source> AS src
ON <merge_search_condition>
WHEN MATCHED THEN <merge_matched>
WHEN NOT MATCHED [ BY TARGET ]
THEN <merge_not_matched>
WHEN NOT MATCHED BY SOURCE THEN
<merge_matched>
MERGE Production.UnitMeasure AS
tgt
USING (SELECT @UnitMeasureCode,
@Name) as src (UnitMeasureCode,
Name)
ON (tgt.UnitMeasureCode =
src.UnitMeasureCode)
WHEN MATCHED THEN
UPDATE SET Name = src.Name
WHEN NOT MATCHED THEN
INSERT (UnitMeasureCode, Name)
VALUES (src.UnitMeasureCode,
src.Name)
```

Multi-Statement Table-Valued Function

```
CREATE [ OR ALTER ] FUNCTION
<function_name> ([ parameter ])
RETURNS @return_variable TABLE
<table_type_definition>
BEGIN
    <function_body>
RETURN
END
```

Inline Table-Valued Function

Check if table is exists

```
IF OBJECT_ID(N'db_name.table_name')
IS NOT NULL
BEGIN
    PRINT 1
END
```

add foreign key

```
CREATE TABLE <schema_name>.<table_name>
(
    <column_name> <data_type>
    , CONSTRAINT <constraint_name>
FOREIGN KEY (<column_name>)
REFERENCES <reference_table_name> (<reference_column_name>)
[ON DELETE <delete_action>]
[ON UPDATE <update_action>]
)
```

add primary key

```
CREATE TABLE <schema_name>.<table_name>
(
    <column_name> <data_type> NOT NULL
    , CONSTRAINT <constraint_name>
PRIMARY KEY (<column_name>)
)
```

create unique constraint

```
CREATE TABLE <table_name>
(
    <column_name> <data_type>
    CONSTRAINT UC_<constraint_name>
UNIQUE (<column_name>)
)
```

create index

```
CREATE TABLE <schema_name>.<table_name>
(
    <column_name> <data_type>
    INDEX IX_<table_name>_<column_name>
NONCLUSTERED (<column_name>)
)
```

```
CREATE [ OR ALTER ] FUNCTION
<function_name> (<parameters>)
RETURNS TABLE
AS
RETURN
(
    <select_statement>
);
```



By **giangpdh**

cheatography.com/giangpdh/

Published 3rd August, 2021.

Last updated 23rd December, 2021.

Page 1 of 2.

Sponsored by **Readab**

Measure your website

<https://readable.com>