*SELECT clause*

*(FROM not required in Microsoft SQL)*

SELECT 1+1 AS Result

SELECT 4\*4 AS [Result 2]

SELECT 1+1 AS Result,

2+2 AS Result2

*FROM clause*

SELECT \*

FROM sys.objects

SELECT name, object\_id, type\_desc

FROM sys.objects

SELECT [name], [object\_id], [type\_desc]

FROM sys.objects

SELECT [name] AS Name\_Of\_Object, [object\_id], [type\_desc]

FROM sys.objects

*WHERE clause*

SELECT \*

FROM sys.objects

WHERE schema\_id = 1

SELECT \*

FROM sys.objects

WHERE schema\_id <> 1

SELECT \*

FROM sys.objects

WHERE schema\_id != 1

SELECT \*

FROM sys.objects

WHERE object\_id > 4 AND object\_id < 10

SELECT \*

FROM sys.objects

WHERE object\_id >= 4 AND object\_id <= 10

SELECT \*

FROM sys.objects

WHERE object\_id between 4 AND 10

SELECT \*

FROM sys.objects

WHERE object\_id = 5 OR object\_id = 9

SELECT \*

FROM sys.objects

WHERE object\_id IN (5, 9)

SELECT \*

FROM sys.objects

WHERE type\_desc = 'SYSTEM\_TABLE'

SELECT \*

FROM sys.objects

WHERE name LIKE 'sysr%'

SELECT \*

FROM sys.objects

WHERE name NOT LIKE 'sysr%'

SELECT \*

FROM sys.objects

WHERE name LIKE '%a%'

SELECT \*

FROM sys.objects

WHERE create\_date < '2019-01-01'

*GROUP BY clause*

SELECT schema\_id, count(schema\_id) AS NumberOfRows

FROM sys.objects

GROUP BY schema\_id

SELECT principal\_id, count(\*) AS NumberOfRows

FROM sys.objects

GROUP BY principal\_id

SELECT principal\_id

FROM sys.objects

GROUP BY principal\_id

SELECT DISTINCT principal\_id

FROM sys.objects

SELECT TOP 10 \*

FROM sys.objects

SELECT schema\_id, name, count(\*) AS NumberOfRows, sum(schema\_id)

FROM sys.objects

GROUP BY schema\_id, name

SELECT object\_id AS ObjectID, name AS ObjectName, column\_id AS ColumnID

FROM sys.columns

WHERE name LIKE 'N%'

SELECT object\_id AS ObjectID, count(\*) AS NumbersOfRows

FROM sys.columns

GROUP BY object\_id

*HAVING clause*

*(Like ‘where’, but after ‘group by’)*

USE msdb

GO

SELECT name, count(\*) as NumberOfRows

FROM sys.objects

GROUP BY name

HAVING count(\*) > 1

SELECT name

FROM sys.objects

WHERE name LIKE 'f%'

GROUP BY name

HAVING count(\*) > 1

SELECT name, count(\*) AS NumberOfRows

FROM sys.objects

WHERE name LIKE 'f%'

GROUP BY name

HAVING count(\*) > 1

*ORDER BY clause*

SELECT name, count(\*) AS NumberOfRows

FROM sys.objects

WHERE name LIKE 'f%'

GROUP BY name

HAVING count(\*) > 1

ORDER BY name DESC

SELECT name, count(\*) AS NumberOfRows

FROM sys.objects

WHERE name LIKE 'f%'

GROUP BY name

ORDER BY NumberOfRows DESC, name ASC

SELECT object\_id AS objectID, count(\*) as NumberOfRows

FROM sys.columns

GROUP BY object\_id

HAVING count(\*) >= 10

ORDER BY NumberOfRows DESC, ObjectID ASC

*VARIABLES*

DECLARE @variable date = '2024-03-02 12:34:56.1234567'

SELECT @variable AS Answer

DECLARE @variable char(5) = 'Some text goes here'

SELECT @variable AS Answer

DECLARE @variable varchar(25) = 'Some text goes here'

SELECT @variable AS Answer

DECLARE @variable nchar(25) = N'Some text goes here Қ'

SELECT @variable AS Answer

*DATABASE*

USE DatabaseFundamentals

GO

CREATE TABLE EmployeeTable (

EmployeeNumber int NOT NULL,

EmployeeFirstName varchar(50) NOT NULL,

EmployeeLastName varchar(50) NOT NULL,

EmployeeMiddleName varchar(50) NULL,

EmployeeGovernmentID char(10) NULL,

DateOfBirth date NOT NULL,

Department varchar(50) NULL,

Manager int NULL)

INSERT INTO [dbo].[EmployeeTable]

([EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName],

[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager])

VALUES(1, 'Jane', 'Doe',NULL, 'AB123456G', '1994-12-30', 'Customer Relations',NULL)

CREATE TABLE TransactionTable(

Amount smallmoney NOT NULL,

DateOfTransaction date NOT NULL,

EmployeeNumber int NOT NULL)

*JOINS*

SELECT E.\*, Amount, DateOfTransaction

FROM [dbo].[EmployeeTable] AS E

JOIN [dbo].[TransactionTable] AS T

ON E.EmployeeNumber = T.EmployeeNumber

SELECT E.\*, Amount, DateOfTransaction

FROM [dbo].[EmployeeTable] AS E

LEFT JOIN [dbo].[TransactionTable] AS T

ON E.EmployeeNumber = T.EmployeeNumber

SELECT N.StampID, StampName, StampYear, PurchasePrice

FROM tblStampNames AS N

LEFT JOIN tblStampPurchases as P

ON N.StampID = P.StampID

WHERE PurchasePrice IS NULL

SELECT N.StampID, StampName, StampYear, PurchasePrice

FROM tblStampNames AS N

FULL JOIN tblStampPurchases as P

ON N.StampID = P.StampID

WHERE PurchasePrice IS NULL

*ADDING, ALTERING & REMOVING COLUMNS*

CREATE TABLE tblTest(

EmployeeNumber tinyint NOT NULL)

ALTER TABLE tblTest

ALTER COLUMN EmployeeNumber smallint NOT NULL

ALTER TABLE tblTest

ADD [Transaction] smallmoney NOT NULL

ALTER TABLE tblTest

DROP COLUMN [Transaction]

DROP TABLE tblTest

TRUNCATE TABLE tblTest

ALTER TABLE [dbo].[tblStampNames]

ADD Rarity VARCHAR(20) NULL

ALTER TABLE [dbo].[tblStampNames]

ALTER COLUMN Rarity VARCHAR(30) NULL

ALTER TABLE [dbo].[tblStampNames]

DROP COLUMN Rarity

*PRIMARY KEYS*

ALTER TABLE EmployeeTable

ADD CONSTRAINT PK\_EmployeeTable\_EmployeeNumber PRIMARY KEY ([EmployeeNumber])

ALTER TABLE EmployeeTable

DROP CONSTRAINT PK\_EmployeeTable\_EmployeeNumber

CREATE TABLE tblTest(

EmployeeNumber int NOT NULL,

CONSTRAINT PK\_EmployeeTable\_EmployeeNumber PRIMARY KEY ([EmployeeNumber]) )

CREATE TABLE tblTest(

EmployeeNumber int NOT NULL PRIMARY KEY)

ALTER TABLE tblTest

ADD EmployeeID INT NOT NULL PRIMARY KEY

ALTER TABLE [dbo].TransactionTable

ADD CONSTRAINT PK\_TransactionTable\_EmployeeNumber\_DateTransaction

PRIMARY KEY (EmployeeNumber, DateOfTransaction)

ALTER TABLE [dbo].TransactionTable

DROP CONSTRAINT PK\_TransactionTable\_EmployeeNumber\_DateTransaction

ALTER TABLE [dbo].TransactionTable

ADD TransactionID int IDENTITY(1,1) NOT NULL

ALTER TABLE [dbo].TransactionTable

ADD CONSTRAINT PK\_TransactionTable\_TransactionID PRIMARY KEY (TransactionID)

SET IDENTITY\_INSERT [dbo].TransactionTable ON

INSERT INTO [dbo].TransactionTable(

[Amount], [DateOfTransaction], [EmployeeNumber], [TransactionID])

VALUES (1, '2030-01-01', 11, 70)

SET IDENTITY\_INSERT [dbo].TransactionTable OFF

ALTER TABLE [dbo].tblStampPurchases

ADD PurchaseID INT PRIMARY KEY IDENTITY(1,1)

ALTER TABLE [dbo].tblStampPurchases

ADD PurchaseID INT IDENTITY(1,1)

ALTER TABLE [dbo].tblStampPurchases

ADD CONSTRAINT PK\_tblStampPurchases\_PurchaseID PRIMARY KEY (PurchaseID)

*FOREIGN KEYS*

ALTER TABLE [dbo].TransactionTable WITH NOCHECK

ADD CONSTRAINT FK\_TransactionTable\_EmployeeNumber

FOREIGN KEY ([EmployeeNumber])

REFERENCES [dbo].EmployeeTable([EmployeeNumber])

ALTER TABLE [dbo].TransactionTable

DROP CONSTRAINT FK\_TransactionTable\_EmployeeNumber

ALTER TABLE [dbo].TransactionTable WITH NOCHECK

ADD CONSTRAINT FK\_TransactionTable\_EmployeeNumber

FOREIGN KEY ([EmployeeNumber])

REFERENCES [dbo].EmployeeTable([EmployeeNumber])

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE [dbo].TransactionTable WITH NOCHECK

ADD CONSTRAINT FK\_TransactionTable\_EmployeeNumber

FOREIGN KEY ([EmployeeNumber])

REFERENCES [dbo].EmployeeTable([EmployeeNumber])

ON DELETE NO ACTION

ON UPDATE

ALTER TABLE [dbo].TransactionTable WITH NOCHECK

ADD CONSTRAINT FK\_TransactionTable\_EmployeeNumber

FOREIGN KEY ([EmployeeNumber])

REFERENCES [dbo].EmployeeTable([EmployeeNumber])

ON DELETE SET NULL

ON UPDATE SET NULL

ALTER TABLE [dbo].TransactionTable WITH NOCHECK

ADD CONSTRAINT FK\_TransactionTable\_EmployeeNumber

FOREIGN KEY ([EmployeeNumber])

REFERENCES [dbo].EmployeeTable([EmployeeNumber])

ON DELETE SET DEFAULT

ON UPDATE SET DEFAULT

ALTER TABLE [dbo].[tblStampNames]

ADD CONSTRAINT PK\_tblStampNames\_StampID PRIMARY KEY ([StampID])

ALTER TABLE [dbo].[tblStampPurchases] WITH NOCHECK

ADD CONSTRAINT FK\_tblStampPurchases\_StampID FOREIGN KEY ([StampID])

REFERENCES [dbo].[tblStampNames]([StampID])

*CONSTRAINTS*

ALTER TABLE [dbo].TransactionTable

ADD CONSTRAINT DF\_TransactionTable\_Amount

DEFAULT 0 FOR [Amount]

CREATE TABLE tblTest(

EmployeeNumber INT CONSTRAINT DEFAULT 0)

CREATE TABLE tblTest(

EmployeeNumber INT CONSTRAINT DF\_tblTest\_Employee DEFAULT 0)

ALTER TABLE [dbo].EmployeeTable

ADD CONSTRAINT UQ\_EmployeeTable\_EmployeeGovernmentID

UNIQUE ([EmployeeGovernmentID])

ALTER TABLE [dbo].TransactionTable

ADD CONSTRAINT CK\_TransactionTable\_Amount

CHECK ((Amount >= -1000 and Amount <= 1000) or EmployeeNumber = 2)

ALTER TABLE [dbo].tblStampNames

ADD CONSTRAINT DF\_tblStampNames\_StampCountry

DEFAULT 'Unknown' FOR [StampCountry]

ALTER TABLE [dbo].tblStampNames

ADD CONSTRAINT UQ\_tblStampNames\_StampNameStampYear

UNIQUE ([StampName], [StampYear])

ALTER TABLE [dbo].tblStampNames

ADD CONSTRAINT CK\_tblStampNames\_StampYear

CHECK (StampYear >= 1800)

*DATA MANIPULATION LANGUAGE (DML)*

INSERT INTO [dbo].EmployeeTable([EmployeeNumber], [EmployeeFirstName], [EmployeeLastName])

VALUES (1, 'Jane', 'Doe'),

(2, 'John', 'Doe')

INSERT INTO [dbo].tblTest2

SELECT \*

FROM [dbo].tblTest

WHERE EmployeeNumber IN (2, 3, 6)

INSERT INTO [dbo].[tblStampNames]

([StampID], [StampName], [StampCountry], [StampYear])

VALUES (10, 'Norwegian Red', 'Norway', 1899)

SELECT StampCountry, PurchasePrice

FROM [dbo].[tblStampNames] AS N

JOIN [dbo].[tblStampPurchases] AS P

ON N.StampID = P.StampID

SELECT StampCountry, PurchasePrice

INTO tblStampAnalysis

FROM [dbo].[tblStampNames] AS N

JOIN [dbo].[tblStampPurchases] AS P

ON N.StampID = P.StampID

BEGIN TRAN

UPDATE [dbo].[EmployeeTable]

SET [EmployeeGovernmentID] = 'ABC', EmployeeFirstName = 'Terry'

WHERE EmployeeNumber = 6

SELECT \* FROM EmployeeTable

ROLLBACK TRAN

CREATE TABLE tblTestChanges(

NewEmployeeNumber INT, NewEmployeeGovernmentID CHAR(10))

BEGIN TRAN

UPDATE [dbo].[tblTest]

SET [EmployeeGovernmentID]

FROM [dbo].[tblTestChanges]

WHERE EmployeeNumber = [NewEmployeeNumber]

SELECT \* FROM tblTest

ROLLBACK TRAN

CREATE TABLE tblTestChanges(

EmployeeNumber INT, EmployeeGovernmentID CHAR(10))

BEGIN TRAN

UPDATE [dbo].[tblTest]

SET [EmployeeGovernmentID = B.[EmployeeGovernmentID]

FROM [dbo].[tblTest] AS A

JOIN [dbo].[tblTestChanges] AS B

ON A.EmployeeNumber = B.[EmployeeNumber]

SELECT \* FROM tblTest

ROLLBACK TRAN

BEGIN TRAN

DELETE

FROM [dbo].[EmployeeTable]

WHERE EmployeeNumber = 9

SELECT \* FROM [dbo].[EmployeeTable]

ROLLBACK TRAN

BEGIN TRAN

DELETE FROM [dbo].[TransactionTable]

FROM [dbo].[TransactionTable] AS T

LEFT JOIN [dbo].[EmployeeTable] AS E

ON T.EmployeeNumber = E.EmployeeNumber

WHERE E.EmplyeeNumber IS NULL

SELECT \* FROM [dbo].[TransactionTable]

ROLLBACK TRAN

CREATE TABLE tblStampNamesUpdate(

StampID tinyint NOT NULL,

StampName VARCHAR(17) NOT NULL)

BEGIN TRAN

UPDATE [dbo].[tblStampNames]

SET StampName = StampName

SELECT \*

FROM [dbo].[tblStampNamesUpdate] AS U

JOIN [dbo].[tblStampNames] AS N

ON U.StampID = N.StampID

SELECT \* FROM [dbo].tblStampNames

ROLLBACK TRAN

CREATE TABLE tblStampNamesUpdate(

StampID tinyint NOT NULL,

StampName VARCHAR(17) NOT NULL)

INSERT INTO tblStampNamesUpdate

VALUES(2,'John Quincy Adams'),

(8, 'William III')

UPDATE [dbo].[tblStampNames]

SET StampName = U.[StampName]

SELECT \*

FROM [dbo].[tblStampNamesUpdate] AS U

JOIN [dbo].[tblStampNames] AS N

ON U.StampID = N.StampID

DELETE

FROM tblStampNamesUpdate

WHERE StampName LIKE '%a%'

*DATA DEFINITION LANGUAGE (DDL)*

CREATE VIEW MyViewSQL AS

SELECT \*

FROM [dbo].[EmployeeTable]

WHERE EmployeeNumber < 10

SELECT \*

FROM [dbo].[MyViewSQL]

GO

BEGIN TRAN

UPDATE [dbo].[MyViewSQL]

SET EmployeeGovernmentID = 'ABC123'

WHERE EmployeeNumber = 6

SELECT \* FROM [dbo].[MyViewSQL]

SELECT \* FROM [dbo].[EmployeeTable]

ROLLBACK TRAN

CREATE VIEW EmployeeTransaction AS

SELECT E.\*, T.Amount, T.DateOfTransaction

FROM EmployeeTable AS E

LEFT JOIN TransactionTable AS T

ON E.EmployeeNumber = T.EmployeeNumber

CREATE VIEW vw\_Stamps AS

SELECT N.\*, PurchaseDate, PurchasePrice

FROM [dbo].[tblStampNames] AS N

LEFT JOIN [dbo].[tblStampPurchases] AS P

ON N.STAMPID = P.StampID

WHERE N.StampID BETWEEN 1 AND 5

SELECT \*

FROM vw\_Stamps

WHERE StampID IN (2, 4)

ALTER VIEW vw\_Stamps AS

SELECT N.\*, PurchaseDate, PurchasePrice

FROM [dbo].[tblStampNames] AS N

LEFT JOIN [dbo].[tblStampPurchases] AS P

ON N.STAMPID = P.StampID

WHERE N.StampID BETWEEN 1 AND 5

WITH CHECK OPTION

GO

UPDATE vw\_Stamps

SET StampID = StampID + 10

WHERE StampID = 2

ALTER VIEW vw\_Stamps WITH SCHEMABINDING AS

SELECT N.StampID, N.StampName, N.StampCountry, N.StampYear, PurchaseDate, PurchasePrice

FROM [dbo].[tblStampNames] AS N

LEFT JOIN [dbo].[tblStampPurchases] AS P

ON N.STAMPID = P.StampID

WHERE N.StampID BETWEEN 1 AND 5

WITH CHECK OPTION

GO

CREATE PROC proc\_EmployeeTransaction AS

SELECT \*

FROM EmployeeTransaction

SELECT \*

FROM [dbo].[TransactionTable]

EXEC [dbo].[proc\_EmployeeTransaction]

ALTER PROC proc\_EmployeeTransaction (@EmployeeNumber INT) AS

SELECT \*

FROM EmployeeTransaction

WHERE EmployeeNumber = @EmployeeNumber

SELECT \*

FROM [dbo].[TransactionTable]

WHERE EmployeeNumber = @EmployeeNumber

EXEC [dbo].[proc\_EmployeeTransaction] 2

ALTER PROC proc\_EmployeeTransaction (@EmployeeNumber INT, @NumOfTransactions INT OUTPUT) AS

SELECT \*

FROM EmployeeTransaction

WHERE EmployeeNumber = @EmployeeNumber

SELECT @NumOfTransactions = COUNT(\*)

FROM [dbo].[TransactionTable]

WHERE EmployeeNumber = @EmployeeNumber

DECLARE @Transactions INT

EXEC [dbo].[proc\_EmployeeTransaction] 2, @Transactions OUTPUT

SELECT @Transactions AS Result

CREATE FUNCTION fn\_TransactionCount (@EmployeeNumber INT)

RETURNS INT

AS

BEGIN

DECLARE @NumOfTransactions INT

SELECT @NumOfTransactions = COUNT(\*)

FROM [dbo].[TransactionTable]

WHERE EmployeeNumber = @EmployeeNumber

RETURN (@NumOfTransactions)

END

GO

SELECT dbo.fn\_TransactionCount(2)

SELECT \*, dbo.fn\_TransactionCount(EmployeeNumber) AS NumberOfTransactions

FROM [dbo].[EmployeeTable]

ALTER TABLE [dbo].[EmployeeTable2]

ADD TypeOfChange VARCHAR(12)

ALTER TABLE [dbo].[EmployeeTable2]

ADD TimeOfChange DATETIME2

CREATE TRIGGER trg\_EmployeeTable\_After

ON [dbo].[EmployeeTable]

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

INSERT INTO [dbo].[EmployeeTable2]

([EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager], TypeOfChange, TimeOfChange)

SELECT [EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager], 'Inserted', GETDATE()

FROM inserted

INSERT INTO [dbo].[EmployeeTable2]

([EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager] , TypeOfChange, TimeOfChange)

SELECT [EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager], 'Deleted', GETDATE()

FROM inserted

END

SELECT \* FROM [dbo].[EmployeeTable]

WHERE EmployeeNumber in (1, 12)

GO

CREATE TRIGGER tr\_EmployeeTransaction ON [dbo].[EmployeeTransaction]

INSTEAD OF INSERT

AS

BEGIN

INSERT INTO [dbo].[EmployeeTable]

([EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager])

SELECT [EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager]

FROM inserted

INSERT INTO [dbo].[TransactionTable]

([Amount],[DateOfTransaction],[EmployeeNumber])

SELECT [Amount],[DateOfTransaction],[EmployeeNumber]

FROM inserted

END

BEGIN TRAN

INSERT INTO [dbo].[EmployeeTransaction]

([EmployeeNumber],[EmployeeFirstName],[EmployeeLastName],[EmployeeMiddleName]

,[EmployeeGovernmentID],[DateOfBirth],[Department],[Manager],[Amount],[DateOfTransaction])

VALUES (12,'Dylan','A','Word','SU416128X','1999-11-26','Customer Relations','5',1.23,'2025-01-01')

SELECT \* FROM [dbo].[EmployeeTable]

WHERE EmployeeNumber in (1, 12)

ROLLBACK TRAN

CREATE FUNCTION fn\_StampPurchasePrice (@StampID tinyint)

RETURNS INT

AS

BEGIN

DECLARE @TotalPrice INT

SELECT @TotalPrice = SUM(PurchasePrice)

FROM [dbo].[tblStampPurchases]

WHERE StampID = @StampID

RETURN @TotalPrice

END

GO

SELECT \*, dbo.fn\_StampPurchasePrice(StampID)

FROM tblStampNames

GO

CREATE PROCEDURE proc\_StampPurchasePrice(@StampIDFrom tinyint, @StampIDTo tinyint)

AS

SELECT \*, dbo.fn\_StampPurchasePrice(StampID) AS TotalPurchasePrice

FROM tblStampNames

WHERE StampID BETWEEN @StampIDFrom AND @StampIDTo

GO

EXEC proc\_StampPurchasePrice

GO

CREATE TABLE [dbo].[tblStampPurchaseAudit](

[StampID] tinyint NOT NULL,

[PurchaseDate] date NOT NULL,

[PurchasePrice] int NOT NULL,

[PurchaseID] int NOT NULL,

TypeOfChange varchar(20) NOT NULL)

GO

CREATE TRIGGER tgl\_tblStampPurchases

ON [dbo].[tblStampPurchases]

AFTER DELETE, INSERT, UPDATE

AS

BEGIN

INSERT INTO [dbo].[tblStampPurchaseAudit]([StampID],[PurchaseDate],[PurchasePrice],[PurchaseID],[TypeOfChange])

SELECT [StampID],[PurchaseDate],[PurchasePrice],[PurchaseID],'Inserted'

FROM inserted

INSERT INTO [dbo].[tblStampPurchaseAudit]([StampID],[PurchaseDate],[PurchasePrice],[PurchaseID],[TypeOfChange])

SELECT [StampID],[PurchaseDate],[PurchasePrice],[PurchaseID], 'Deleted'

FROM deleted

END

GO

INSERT INTO tblStampPurchases([StampID],[PurchaseDate],[PurchasePrice])

VALUES(7,'2030-01-01', 1234)

UPDATE tblStampPurchases

SET PurchasePrice = 1235

WHERE StampID = 7

DELETE FROM tblStampPurchases

WHERE StampID = 7

SELECT \* FROM [dbo].[tblStampPurchaseAudit]

*SET OPERATORS*

SELECT \*

FROM tblTest1

UNION

SELECT \*

FROM tblTest2

SELECT \*

FROM tblTest1

UNION ALL

SELECT \*

FROM tblTest2

SELECT \*

FROM tblTest1

INTERSECT

SELECT \*

FROM tblTest2

SELECT \*

FROM tblTest1

EXCEPT

SELECT \*

FROM tblTest2

*OTHER TOPICS*

CREATE UNIQUE CLUSTERED INDEX Ind\_EmployeeTable\_EmployeeGovernmentID

ON [dbo].[EmployeeTable]([EmployeeGovernmentID])

CREATE CLUSTERED INDEX Ind\_EmployeeTable\_EmployeeGovernmentID

ON [dbo].[EmployeeTable]([EmployeeGovernmentID])

CREATE NONCLUSTERED INDEX Ind\_EmployeeTable\_EmployeeNumber

ON [dbo].[EmployeeTable]([EmployeeNumber])