/\*An introduction to the creation of tables will clarify how joins work when we use SQL joins.

When we create a table with various columns, we also create a column with increasing values such as an integer.

This column is then set as a primary key for that whole table; and the purpose of that primary key is to keep each row of data unique.

We then create that primary key column in another referencing table and this then becomes the foreign in the new table.

This relationship between the primary key and the foreign key allows us to extract data from both tables.\*/

/\*

SQL joins are used to combine data from two or more tables based on a common field between them (Usually the Primary key / Foreign Key)

The relationship between the primary key and the foreign key allows us to extract data from both tables.

There are several types of joins; the common ones are:

Inner join

Left outer join

Right outer join

Full outer join

A combination of inner and outer join

An introduction to the creation of tables will clarify how joins work when we use SQL joins.

When we create a table with various columns, we also create a column with increasing values such as an integer.

This column is then set as a primary key for that whole table; and the purpose of that primary key is to keep each row of data unique.

We then create that primary key column in another referencing table and this then becomes the foreign in the new table.

This relationship between the primary key and the foreign key allows us to extract data from both tables.

\*/

USE TSQL

GO

--Create a table

CREATE TABLE Names3

( NameID int NOT NULL IDENTITY (1,1) PRIMARY KEY, --<< Note: This is the Primary Key for this table.

Fname varchar (20) NULL,

Lname varchar (20) NULL,

Gender char (1) NULL)

--Insert data into table names

INSERT INTO Names3

VALUES

('Tom','Jones','M'), --1

('Bob','Smith','M'), --2

('Henry','Book','M'), --3

('Mary','Lamp','F'), --4

('Susan','Keys','F'),--5

('Danella','Hortom','F')--6

SELECT \* FROM Names3

--Now when we create another table, and if has relationship on this table, then we need to create a table with a common column to both tables –

--in this case, the NameID.

--Create a second table

CREATE TABLE Department

(DepartmentID int NOT NULL IDENTITY (1,1) PRIMARY KEY, --<<Note: This is the Primary Key for this table

Department varchar (20),

BuildingCode varchar (20),

NameID int FOREIGN KEY REFERENCES Names3 (NameID) NULL) --<<Note: This is the Foreign Key in this table that has a relationship with Names table

--Insert data into table Department

INSERT INTO Department

VALUES

('Finance','abc',1),

('Business','xyz',1),

('Accouting','123',2),

('Business','xyz',2),

('Taxes','xyz',NULL),

('Taxes','xyz',NULL),

('Taxes','xyz',NULL)

SELECT \* FROM Department

--Question: Which department does each person work for?

--Example of INNER JOIN

--(GIVES US ALL THE DATA FROM THE FIRST TABLE (NAMES) WHEN THERE IS A MATCH IN THE SECOND TABLE(DEPARTMENT))

SELECT \* FROM Names3

SELECT \* FROM Department

SELECT

Names3.Fname,

Names3.Lname,

Department.Department

FROM

Names3 --FIRST TABLE

INNER JOIN

Department --SECOND TABLE

ON

Names3.NameID = Department.NameID --<<Note: The relationship between these two columns (Keys) determine the join

--Example of LEFT OUTER JOIN

--(GIVES US ALL THE DATA FROM THE FIRST TABLE (NAMES) EVEN WHEN THERE IS NO MATCH IN THE SECOND TABLE(DEPARTMENT))

SELECT \* FROM Names3

SELECT \* FROM Department

SELECT

Names3.Fname,

Names3.Lname,

Department.Department

FROM

Names3 --FIRST TABLE

LEFT JOIN

Department --SECOND TABLE

ON

Names3.NameID = Department.NameID --<<Note: The relationship between these two columns (Keys) determine the join

--Example of RIGHT OUTER JOIN

--(GIVES US ALL THE DATA FROM THE SECOND TABLE (NAMES) EVEN WHEN THERE IS NO MATCH IN THE FIRST TABLE (DEPARTMENT))

SELECT \* FROM Names3

SELECT \* FROM Department

SELECT

Department.Department,

Names3.Fname,

Names3.Lname

FROM

Department --FIRST TABLE

RIGHT JOIN

Names3 --SECOND TABLE

ON

Department.NameID = Names3.NameID --<<Note: The relationship between these two columns (Keys) determine the join

--Example of FULL OUTER JOIN

--(GIVES US ALL THE DATA FROM BOTH TABLES EVEN IF THEY DONT MATCH)

SELECT \* FROM Names3

SELECT \* FROM Department

SELECT

Names3.Fname,

Names3.Lname,

Department.Department

FROM

Names3

FULL JOIN

Department

ON

Names3.NameID = Department.NameID --<<Note: The relationship between these two columns (Keys) determine the join

SELECT

Names3.Fname,

Names3.Lname,

Department.Department

FROM

Names3

FULL JOIN

Department

ON

Names3.NameID = Department.NameID

--Write an Full Join that eliminates Nulls. Hint: use the where clause to filter nulls

SELECT

Names3.Fname,

Names3.Lname,

Department.Department

FROM

Names3

FULL JOIN

Department

ON

Names3.NameID = Department.NameID

Where Fname IS NOT NULL AND Lname IS NOT NULL AND Department IS NOT NULL

Drop Table Names3

Drop Table Department

CreateDatabaseJoins

Go

USEjoins

GO

--Create a table

CREATETABLENames

(NameIDintNOTNULLIDENTITY (1,1)PRIMARYKEY,--<< Note: This is the Primary Key for this table.

Fnamevarchar(20)NULL,

Lnamevarchar(20)NULL,

Genderchar (1)NULL)

--Insert data into table names

INSERTINTONames

VALUES

('Tom','Jones','M'),--1

('Bob','Smith','M'),--2

('Henry','Book','M'),--3

('Mary','Lamp','F'),--4

('Susan','Keys','F'),--5

('Danella','Hortom','F')--6

SELECT\*FROMNames

--Now when we create another table, and if has relationship on this table, then we need to create a table with a common column to both tables –

--in this case, the NameID.

--Create a second table

CREATETABLEDepartment

(DepartmentIDintNOTNULLIDENTITY (1,1)PRIMARYKEY,--<<Note: This is the Primary Key for this table

Departmentvarchar(20),

BuildingCodevarchar(20),

NameIDintFOREIGNKEYREFERENCESNames(NameID)NULL)--<<Note: This is the Foreign Key in this table that has a relationship with Names table

--Insert data into table Department

INSERTINTODepartment

VALUES

('Finance','abc',1),

('Business','xyz',1),

('Accouting','123',2),

('Business','xyz',2),

('Taxes','xyz',NULL),

('Taxes','xyz',NULL),

('Taxes','xyz',NULL)

SELECT\*FROMDepartment

--Question: Which department does each person work for?

--Example of INNER JOIN

--(GIVES US ALL THE DATA FROM THE FIRST TABLE (NAMES) WHEN THERE IS A MATCH IN THE SECOND TABLE(DEPARTMENT))

SELECT

Names.Fname,

Names.Lname,

Department.Department

FROM

Names--FIRST TABLE

INNERJOIN

Department--SECOND TABLE

ON

Names.NameID=Department.NameID--<<Note: The relationship between these two columns (Keys) determine the join

--Example of LEFT OUTER JOIN

--(GIVES US ALL THE DATA FROM THE FIRST TABLE (NAMES) EVEN WHEN THERE IS NO MATCH IN THE SECOND TABLE(DEPARTMENT))

SELECT

Names.Fname,

Names.Lname,

Department.Department

FROM

Names--FIRTS TABLE

LEFTJOIN

Department--SECOND TABLE

ON

Names.NameID=Department.NameID--<<Note: The relationship between these two columns (Keys) determine the join

--Example of RIGHT OUTER JOIN

--(GIVES US ALL THE DATA FROM THE SECOND TABLE (NAMES) EVEN WHEN THERE IS NO MATCH IN THE FIRST TABLE (DEPARTMENT))

SELECT

Department.Department,

Names.Fname,

Names.Lname

FROM

Department--FIRST TABLE

RIGHTJOIN

Names--SECOND TABLE

ON

Department.NameID=Names.NameID--<<Note: The relationship between these two columns (Keys) determine the join

--Example of FULL OUTER JOIN

--(GIVES US ALL THE DATA FROM BOTH TABLES EVEN IF THEY DONT NATCH)

SELECT

Names.Fname,

Names.Lname,

Department.Department

FROM

Names

FULLJOIN

Department

ON

Names.NameID=Department.NameID--<<Note: The relationship between these two columns (Keys) determine the join