--UNIT-5,6 &7:

SQL @ RAJKUMAR KARKI

---------------------------------------------------------------------------------------------

-------------------------------------------1ST DAY:------------------------------------------

---------------------------------------------------------------------------------------------

/\*Q.1.Make 3 tables in the database using sql query where tables must be your

info.table with many data types, employee tble and staff table unit 4. \*/

/\* A.1. Staff table yo xai \*/

CREATE TABLE tbl\_staff(

StaffID int Primary key,

FirstName varchar(255) ,

Addresspl varchar(255),

Contact int,

Department varchar(255)

);

/\* tabling garda kei mistake vayo vane feri drop garera delete garera feri table create garnu parx \*/

DROP TABLE tbl\_staff;

/\* yeslay xai staff table show garx\*/

SELECT \* FROM tbl\_staff;

/\*A.2. Employee table yo xai \*/

CREATE TABLE tbl\_employee(

EmpID int,

Name varchar(255),

Address varchar(255),

CodeBranch int,

Branch varchar(225)

);

/\* yeslay xai employee table show garx\*/

SELECT \* FROM tbl\_employee;

/\* tabling garda kei mistake vayo vane feri drop garera delete garera feri table create garnu parx \*/

DROP TABLE tbl\_employee

/\* yeslay xai employee table show garx after deleting\*/

SELECT \* FROM tbl\_employee;

/\*A.3. Mero Table\*/

CREATE TABLE Prince(

Name varchar(255),

Address varchar(255),

Phone int,

Gmail varchar(225)

);

/\*Display garx prince table lai\*/

SELECT \* FROM prince;

/\* tabling garda kei mistake vayo vane feri drop garera delete garera feri table create garnu parx \*/

DROP TABLE prince;

/\* yeslay xai prince table show garx after deleting\*/

SELECT \* FROM prince;

/\*Q.2.Make 3 tables in the database using sql query where tables must be your

info.table with many data types, employee tble and staff table unit 4.

--Alter teacher, datatype as integers.\*/

/\*A.1.Creating Teacher Table \*/

CREATE TABLE Teacher(

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255) NOT NULL,

Contact int,

Address varchar(255),

Age int

);

/\*A.1.1.Displaying \*/

SELECT \*FROM Teacher;

/\*A.1.2.Altering in the Teacher table by adding the Gmail column \*/

ALTER TABLE Teacher

ADD Gmail varchar(255);

/\* A.1.3.Displaying \*/

SELECT \*FROM Teacher;

/\*A.1.4.Altering in the Teacher table by deleting the Gamil column only \*/

ALTER TABLE Teacher

DROP COLUMN Gmail;

/\*A.1.5.Displaying \*/

--

SELECT \*FROM Teacher;

/\*A.1.6.Altering in theTeacher table by adding gmail column again which is delete above \*/

ALTER TABLE Teacher

ADD Gmail varchar(255);

/\*A.1.7.Display \*/

SELECT \*FROM Teacher;

/\*A.1.8.Altering in the Teacher table by altering in the gmail column (varchar to integer) \*/

ALTER TABLE Teacher

ALTER COLUMN Gmail int;

/\*A.1.9.Displaying \*/

SELECT \*FROM Teacher;

/\*A.2. Employee table \*/

CREATE TABLE tbl\_employee(

EmpID int,

Name varchar(255),

Address varchar(255),

CodeBranch int,

Branch varchar(225)

);

/\*A.2.1.Displaying \*/

SELECT \*FROM tbl\_employee;

/\*A.1.2.Altering in the Teacher table by adding the Phone column \*/

ALTER TABLE tbl\_employee

ADD Phone int;

/\* A.2.3.Displaying \*/

SELECT \*FROM tbl\_employee;

/\*A.2.4.Altering in the Teacher table by deleting the Phone column only \*/

ALTER TABLE tbl\_employee

DROP COLUMN Phone;

/\*A.2.5.Displaying \*/

SELECT \*FROM tbl\_employee;

/\*A.2.6.Altering in theTeacher table by adding gmail column again which is delete above \*/

ALTER TABLE tbl\_employee

ADD Phone int;

/\*A.2.7.Display \*/

SELECT \*FROM tbl\_employee;

/\*A.2.8.Altering in the Teacher table by altering in the Phone column (varchar to integer) \*/

ALTER TABLE tbl\_employee

ALTER COLUMN Gmail varchar(225);

/\*A.2.9.Displaying \*/

SELECT \*FROM tbl\_employee;

/\*A.3. Mero Table\*/

CREATE TABLE Prince(

ID int,

Name varchar(255),

Address varchar(255),

Phone int,

Gmail varchar(225)

);

/\*A.3.1.Displaying \*/

SELECT \*FROM Prince;

/\*A.3.2.Altering in the Teacher table by adding the Course column \*/

ALTER TABLE Prince

ADD Course varchar(255);

/\* A.3.3.Displaying \*/

SELECT \*FROM Prince;

/\*A.3.4.Altering in the Teacher table by deleting the Course column only \*/

ALTER TABLE Prince

DROP COLUMN Course;

/\*A.3.5.Displaying \*/

SELECT \*FROM Prince;

/\*A.1.6.Altering in theTeacher table by adding Course column again which is delete above \*/

ALTER TABLE Prince

ADD Course varchar(255);

/\*A.3.7.Display \*/

SELECT \*FROM Prince;

/\*A.3.8.Altering in the Teacher table by altering in the Course column (varchar to integer) \*/

ALTER TABLE Prince

ALTER COLUMN Course int;

/\*A.3.9.Displaying \*/

SELECT \*FROM Prince;

/\*Q.3.Make 3 tables in the database using sql query where tables must be your

info.table with many data types, employee tble and staff table unit 4.

--Add constraints to the columns.

--NOT NULL, UNIQUE, PK, FK\*/

/\*A.1.UNQUIE\*/

CREATE TABLE Rajkumar(

ID int NOT NULL UNIQUE,

LastName varchar(255)NOT NULL,

FirstName varchar(255),

Age int

CONSTRAINT UC\_RAJKUMAR UNIQUE(ID,LastName)

);

/\*Displaying\*/

SELECT \* FROM Rajkumar;

ALTER TABLE Rajkumar

DROP CONSTRAINT UC\_RAJKUMAR;

/\*deleting\*/

DROP TABLE Rajkumar;

/\*A.2.PRIMARY KEY \*/

CREATE TABLE Rajkumars(

ID int NOT NULL PRIMARY KEY,

LastName varchar(255)NOT NULL,

FirstName varchar(255),

Age int

);

/\*Displaying\*/

SELECT \* FROM Rajkumars;

/\* During Altering table\*/

ALTER TABLE Rajkumars

ADD PRIMARY KEY(ID);

/\*Displaying\*/

SELECT \* FROM Rajkumars;

/\* To allow naming of a PRIMARY KEY constraint, and for defining a PRIMARY KEY constraint on multiple columns\*/

CREATE TABLE Rajkumars(

ID int NOT NULL,

LastName varchar(255)NOT NULL,

FirstName varchar(255,

Age int,

CONSTRIANT PK\_RAJKUMARS PRIMARY KEY(ID,LastName)

);

/\*Displaying\*/

SELECT \* FROM Rajkumars;

ALTER TABLE Rajkumars

ADD CONSTRAINT PK\_RAJKUMARS PRIMARY KEY(ID,LastName);

/\*Displaying\*/

SELECT \* FROM Rajkumars;

ALTER TABLE Rajkumars

DROP CONSTRAINT PK\_RAJKUMARS;

/\*Displaying\*/

SELECT \* FROM Rajkumars;

/\*A.3.FOREIGN KEY\*/

CREATE TABLE Purchase(

ItemID int NOT NULL PRIMARY KEY,

ItemNumber int NOT NULL,

PersonID int

);

/\*Displaying\*/

SELECT \* FROM Purchase;

DROP TABLE Purchase;

ALTER TABLE Rajkumars

ADD FOREIGN KEY (PersonsID) REFERENCES Rajkumars(ID);

/\*FOREIGN KEY constraint\*/

CREATE TABLE Purchase(

ItemID int NOT NULL,

ItemNumber int NOT NULL,

PersonID int,

PRIMARY KEY(ItemID),

CONSTRAINT FK\_PersonPurchase FOREIGN KEY(PersonID)

REFERENCES Rajkumars(ID)

);

/\*Displaying\*/

SELECT \* FROM Purchase;

ALTER TABLE Purchase

ADD CONSTRAINT FK\_PersonPurchase

FOREIGN KEY(PersonID) REFERENCES Rajkumars(ID);

/\*Displaying\*/

SELECT \* FROM Purchase;

ALTER TABLE Purchase

DROP CONSTRAINT FK\_PersonPurchase;

/\*Displaying\*/

SELECT \* FROM Purchase;

/\*Q.4.Make 3 tables in the database using sql query where tables must be your info.table

with many data types, employee tble and staff table unit 4.

--CHECK,DEFAULT

Use both create and alter for all 3 tables\*/

DROP TABLE Prince;

CREATE TABLE Prince(

ID int,

Name varchar(255),

Address varchar(255),

Phone int,

Gmail varchar(225)

Age int CHECK (Age>=18)

);

ALTER TABLE Prince

ADD CHECK (Age>=18);

/\*--To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns\*/

ALTER TABLE Prince

ADD CONSTRAINT CHK\_PrinceAge CHECK (Age>=18); ---changing

ALTER TABLE Prince

DROP CONSTRAINT CHECK CHK\_PrinceAge; ---deleting

/\* --Default- set a default value for a column if no value is specified \*/

DROP TABLE Prince; ---delelting

CREATE TABLE Prince(

ID int NOT NULL,

Name varchar(255)NOT NULL,

Address varchar(255) DEFAULT 'Nepal',

Phone int,

Gmail varchar(225)

Age int

);

ALTER TABLE Prince

ADD CONSTRAINT df\_City

DEFAULT 'Kathmandu' FOR Address; --changing

ALTER TABLE Prince

DROP CONSTRAINT df\_Address; ---deleleting

DROP TABLE Employees; --Sabai table nai delete garx

---1.)Creating Employee table with Autoincrement primary key filed:

CREATE TABLE Employees (

EmployeeID int IDENTITY(200,10)PRIMARY KEY, --(200,10)=suru 200 bata ani 10 ko gaplay hunx

EmployeeName varchar(255),

ContactNo varchar(255),

Address varchar(255),

City varchar(255),

Country varchar(255),

Age int,

DateOfBirth varchar(255),

);

SELECT \*FROM Employees;--Purai table lai display garx

--2.) INSERT INTO statements by two ways:

--2.a.)Specify both the column names and the values to be inserted:

--Syntax:

INSERT INTO table\_name (column1, column2, column3,....)

VALUES (value1, value2, value3,....);

--Example:

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Ram', 987654321, 'Talchhikhel', 'Ktm', 'Nepal',19,'1996-07-01');

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Logan', 9873957373, 'Huston', 'WashingtonDC', 'USA',20,'1999-01-24');

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('sandeep', 8459553575, 'Haude', 'Delhi', 'India',33,'1950-10-30');

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Sita', 8575849444, 'Jaulakhel', 'Latipur', 'Nepal',32,'1896-05-12');

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Hank', 3558489378, 'Hewei', 'Hongkong', 'China',45,'1972-12-06');

INSERT INTO Employees (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Mahomat', 4749494948, 'Kasi', 'Baudi', 'Pakistan',25,'1866-03-07');

SELECT \*FROM Employees;

--2.b.)If you are adding values for all the columns of the table, you don't need to specify

--Syntax:

INSERT INTO table\_name

VALUES (value1, value2, value3,....);

--Example:

INSERT INTO Employees (EmployeeName, City, Country)

VALUES ('Ram', 'NewDelhi', 'India');

SELECT \*FROM Employees;

--3.) SELECT statement:

SELECT column1, column2, ....

FROM table\_name;

SELECT EmployeeName, City FROM Employees;

SELECT \*FROM Employees;

--4.)SELECT DISTINCT statment:

--Syntax:

SELECT DISTINCT column1, column2, ....

FROM table\_name;

--Example:

SELECT DISTINCT City, Country

FROM Employees;

--5.)SQL WHERE Clause:

--Syntax:

SELECT column1, column2, ....

FROM table\_name

WHERE condition;

--Example:

SELECT \*FROM Employees

WHERE Country='Nepal';

SELECT \*FROM Employees

WHERE EmployeeID=220;

--6.)SQL SELECT: AND, OR & NOT operators:

--6.a.)AND syntax:yeslay xai duetai condition match garesi matra display garx.

SELECT column1, column2, .....

FROM table\_name

WHERE condition1 AND condition2 AND condition3 ...;

--Example:Country & City:

SELECT \*FROM Employees

WHERE Country='USA' AND City='WashingtonDC';

--6.b)OR Syntax:yeslay xai anyone match vayesi display garx.

SELECT column1, column2, .....

FROM table\_name

WHERE condition1 OR condition2 OR condition3 ...;

--Example:City:

SELECT \*FROM Employees

WHERE City='Hongkong' OR City='Chapagau';

--6.c)NOT Syntax:yeslay xai vaye condition vayek aru sabai display garx.

SELECT column1, column2, .....

FROM table\_name

WHERE NOT condition;

--Example:Country & City:

SELECT \*FROM Employees

WHERE NOT Country='China';

--7.)Combining: (AND, OR & NOT):

--7.a.)Example:AND & OR combining:

SELECT \*FROM Employees

WHERE Country='USA' AND (City='WashingtonDC' OR City='Hongkong');

--7.b.)Example:AND & NOT combining:

SELECT \*FROM Employees

WHERE NOT Country='China' AND NOT Country='USA';

--8.)ORDER BY keyword: records lai sorts garx ascending dekhi descending ma change garera

--(DESC)keyword

SELECT column1, column2, ....

FROM table\_name

ORDER BY column1,column2,....ASC|DESC;

--Example:

SELECT \*FROM Employees

ORDER BY Country;

---------------------------------------------------------------------------------------------

-------------------------------------------2ND DAY:------------------------------------------

---------------------------------------------------------------------------------------------

--ORDER BY DESC Example:

SELECT \*FROM Employees

ORDER BY Country DESC;

--ORDER BY Several Columns Example:

SELECT \*FROM Employees

ORDER BY Country,EmployeeName;

--9.)Aggregate Functions:

--1.)MIN() Function:

--Syntax:

SELECT MIN(Column\_name)

FROM table\_name

WHERE codition;

--Example:

SELECT MIN(Age) AS YoungerAge

FROM Employees;

--2.)MAX() Function:

--Syntax:

SELECT MAX(Column\_name)

FROM table\_name

WHERE codition;

--Example:

SELECT MAX(Age) AS OlderAge

FROM Employees;

--3.)COUNT() Function:

--Syntax:

SELECT COUNT(Column\_name)

FROM table\_name

WHERE condition;

--Example:

SELECT COUNT(EmployeeID)

FROM Employees;

--4.)AVG() Function:

--Syntax:

SELECT AVG(Column\_name)

FROM table\_name

WHERE condition;

--Example:

SELECT AVG(Age)

FROM Employees;

--5.)SUM() Function:

--Syntax:

SELECT SUM(Column\_name)

FROM table\_name

WHERE condition;

--Example:

SELECT SUM(Age)

FROM Employees;

--10.)SQL Operators:

--1.)LIKE Operator:

--Syntax:

SELECT column1, column2,...

FROM table\_name

WHERE columnN LIKE pattern;

--a.)'a%' LIKE Operator: suru xai a word bata vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE 'a%';

--b.)'%a' LIKE Operator: end xai a word ma vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE '%a';

--c.)'%or%' LIKE Operator: or vako word aako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE '%or%';

--d.)'\_r%' LIKE Operator: second mai xai r word suru vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE '\_r%';

--e.)'\_or%' LIKE Operator: second mai xai or word suru vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE '\_or%';

--f.)'a\_%' LIKE Operator: second last ma xai a word suru vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE 'a\_%';

--g.)'a%o' LIKE Operator: suru a word vako ra end xai o word ma vako

--Example:

SELECT \*FROM Employees

WHERE Address LIKE 'a%o';

--h.)NOT 'a%' LIKE Operator: suru vako a word bahek aru dekhau x

--Example:

SELECT \*FROM Employees

WHERE Address LIKE 'a%';

--2.)IN Operator:

--Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (value1, value2,....);

--Or:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (SELECT STATEMENT);

--Example:IN Operator:particular select garne

SELECT \*FROM Employees

WHERE Country IN ('India', 'Pakistan', 'Nepal');

--Example:NOT IN Operator: select gareko bahek aru dekhaux

SELECT \*FROM Employees

WHERE Country NOT IN ('India', 'Pakistan', 'Nepal');

--Example:IN & SELECT Operator: sabai dekhaux

SELECT \*FROM Employees

WHERE Country IN (SELECT Country FROM Employees);

--3.)BETWEEN Operator:

--Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name BETWEEN value1 AND value2;

--Example:tokeko range dekhau x

SELECT \*FROM Employees

WHERE Age BETWEEN 20 AND 30;

--NOT BETWEEN Example:tokeko range bahek aru dekhaux

SELECT \*FROM Employees

WHERE Age NOT BETWEEN 20 AND 30;

--BETWEEN with NOT IN Example:tokeko range dekhaux tara tokeko epmloyeeID bahek dekhaux

SELECT \*FROM Employees

WHERE Age BETWEEN 20 AND 30

AND EmployeeID NOT IN (200,210,220);

--BETWEEN with IN Example:

SELECT \*FROM Employees

WHERE Age BETWEEN 20 AND 30

AND EmployeeID IN (200,210,220);

--BETWEEN Dates Example:

SELECT \*FROM Employees

WHERE DateOfBirth BETWEEN #07/01/1996# AND #07/06/1996#;

--OR,

SELECT \*FROM Employees

WHERE DateOfBirth BETWEEN '1996-07-01' AND '1996-07-06';

--11.)SQL GROUP BY Statement:

--Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

ORDER BY column\_name(s);

--Example:

SELECT COUNT(EmployeeID),Country

FROM Employees

GROUP BY Country;

--Example:descriding sort:

SELECT COUNT(EmployeeID),Country

FROM Employees

GROUP BY Country

ORDER BY COUNT(EmployeeID) DESC;

--12.)SQL UPDATE Statement:

--Syntax:

UPDATE table\_name

SET column1 = value1, column2 = value2,....

WHERE condition;

--UPDATE Table:

--Example: cahnge garnu paro vane set use garne

UPDATE Employees

SET EmployeeName='Shyam'

WHERE EmployeeID=200;

SELECT \*FROM Employees;

--UPDATE Multiple Records: sabai set garx jas country ko naam usa x

--Example:

UPDATE Employees

SET EmployeeName='Hami'

WHERE Country='USA';

SELECT \*FROM Employees;

--13.)SQL DELETE Statement: table bitra ma delete garna sakx

--a.)Deleted from column:column matra delete garx

--Syntax:

DELETE FROM table\_name WHERE condition;

--Example:

DELETE FROM Employees WHERE EmployeeName='Sita';

SELECT \*FROM Employees;

--b.)Deleted whole table:

--Syntax:

DELETE FROM table\_name;

--Example:

DELETE FROM Employees;

SELECT \*FROM Employees;

yees.EmployeeId;

---------------------------------------------------------------------------------------------

-------------------------------------------3rd DAY:------------------------------------------

---------------------------------------------------------------------------------------------

DROP TABLE Students;

DROP TABLE Course;

---14.) JOIN SQL:

--- Students Table:

CREATE TABLE Students(

StudentID int IDENTITY(1,1) PRIMARY KEY,

StudentName varchar(255) NOT NULL,

Address varchar(255),

Phone int,

Gmail varchar(225)

);

---Iserting values:

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Rupa', 'ktm', '984762224', 'haua@gmail.com');

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Raju', 'lalitpur', '347984774',' jkakljfa@gmail.com');

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Rajkumar', 'hattiban', '654656264' , 'hbwbiba@gmail.com');

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Ramesh', 'patan', '89987879', 'jgigrvvjngfa@gmail.com');

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Ramu', 'Chapagau', '798984343', 'yjhsakfahjf@gmail.com');

INSERT INTO Students (StudentName, Address, Phone, Gmail)

VALUES ('Hritik', 'lele', '892844424', 'hkjhfhuwre@gmail.com');

---Displaying:

SELECT \* FROM Students;

---Course Table:

CREATE TABLE Course(

CourseID int NOT NULL IDENTITY(1,1) PRIMARY KEY,

CourseName varchar(255) NOT NULL,

StudentID int NOT NULL,

);

---Iserting values:

INSERT INTO Course (CourseName, StudentID)

VALUES ('Science', 10);

INSERT INTO Course (CourseName, StudentID)

VALUES ('Social', 15);

INSERT INTO Course (CourseName, StudentID)

VALUES ('english' , 20);

INSERT INTO Course (CourseName, StudentID)

VALUES ( 'computer', 25);

INSERT INTO Course (CourseName, StudentID)

VALUES ('math' , 30);

INSERT INTO Course (CourseName, StudentID)

VALUES ( 'nepali', 35);

---Displaying:

SELECT \* FROM Course;

---4 Types of JOIN:

---1.) LEFT JOIN:

SELECT Students.StudentName, Course.CourseID

FROM Students

LEFT JOIN Course On Students.StudentID = Course.StudentID

ORDER BY Students.StudentName;

---2.) RIGHT JOIN:

SELECT Course.CourseID, Students.StudentName, Students.Gmail

From Course

RIGHT JOIN Students ON Course.StudentID=Students.Gmail

ORDER BY Course.CourseID;

---3.) INNER JOIN:

SELECT Course.CourseID, Students.StudentName

FROM Course

INNER JOIN Students ON Students.StudentID = Students.StudentID

---3.) FULL OUTER JOIN:

SELECT Students.StudentName, Course.CourseID

FROM Students

FULL OUTER JOIN Course ON Students.StudentID = Course.StudentID

ORDER BY Students.StudentName;

DROP TABLE Employ;

---14.)VIEW SQL:

CREATE TABLE Employ(

EmployeeID int NOT NULL IDENTITY(200,10)PRIMARY KEY, --(200,10)=suru 200 bata ani 10 ko gaplay hunx

EmployeeName varchar(255) NOT NULL,

ContactNo varchar(255),

Address varchar(255),

City varchar(255),

Country varchar(255),

Age int,

DateOfBirth varchar(255),

);

---Inserting values:

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Ram', 987654321, 'Talchhikhel', 'Ktm', 'Nepal',19,'1996-07-01');

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Logan', 9873957373, 'Huston', 'WashingtonDC', 'USA',20,'1999-01-24');

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('sandeep', 8459553575, 'Haude', 'Delhi', 'India',33,'1950-10-30');

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Sita', 8575849444, 'Jaulakhel', 'Latipur', 'Nepal',32,'1896-05-12');

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Hank', 3558489378, 'Hewei', 'Hongkong', 'China',45,'1972-12-06');

INSERT INTO Employ (EmployeeName, ContactNo, Address, City, Country,Age,DateOfBirth)

VALUES ('Mahomat', 4749494948, 'Kasi', 'Baudi', 'Pakistan',25,'1866-03-07');

---Displaying:

SELECT \*FROM Employ;

---1.) VIEW :

CREATE VIEW [Nepal Employ] AS

SELECT EmployeeName, City, ContactNo

FROM Employ

WHERE Country = 'Nepal';

---Displaying:

SELECT \* FROM [Nepal Employ];

---VIEW with Average:

CREATE VIEW [Employ Above Average Age] AS

SELECT EmployeeName, Age

From Employ

WHERE Age>(SELECT AVG(Age) FROM Employee);

---Displaying:

SELECT \* FROM [Employ Above Average Age];

---2.) UPDATING VIEW:

CREATE OR ALTER VIEW [Nepal Employ] AS

SELECT EmployeeName, City, Address

FROM Employ

WHERE Country = 'Nepal';

---Displaying:

SELECT \* FROM [Nepal Employ];

---3.) DROP VIEW:

DROP VIEW [Nepal Employ];

---Displaying:

SELECT \* FROM [Nepal Employ];

---------------------------------------------------------------------------------------------

-------------------------------------------4th DAY:------------------------------------------

---------------------------------------------------------------------------------------------

---Product Table:

CREATE TABLE Product(

ProductId int IDENTITY(100, 5) PRIMARY KEY,

ProductName varchar(255),

CompanyName varchar(255),

Manufacture\_address varchar(255),

City varchar(255),

Price int

);

---Inserting the values in the table:

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('gloves', 'panja', 'fsagsdg', 'Bhaktapur', 1500);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('Harddisk', 'seagate', 'sitapaila', 'ktm', 1000);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('Copy', 'go green', 'Gathaghar', 'Bhaktapur', 150);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('Book', 'kev', 'bagbazar', 'ktm', 450);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('Laptop', 'dell', 'waunk', 'China', 80000);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('keyboard', 'microsmart', 'tokkfajk', 'tokiyo', 500);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('Milkbars', 'chocolate', 'sgsga', 'Mumbai', 200);

INSERT INTO Product(ProductName, CompanyName, Manufacture\_address, City, Price)

Values ('coffee', 'brew', 'pashupatinagar', 'Illam', 700);

---Displaying Tabel:

SELECT \* FROM Product;

---Deleting the Table:

DROP TABLE Product;

---Displaying the table:

SELECT \* FROM Product;

---15.) GO STORED PROCEDURE SQL:

---Stored Procedure:

GO

CREATE PROCEDURE SelectAllProduct

AS

SELECT\*FROM Product

GO

---Execute the stored procedure:

EXEC SelectAllProducts;

---Deleting the all product only:

DROP PROCEDURE SelectAllProduct;

---Stored Procedure:

GO

CREATE PROCEDURE SelectAllProduct

AS

SELECT\*FROM Product

GO

---Execute the stored procedure:

EXEC SelectAllProducts;

---Stored Procedure with One Parameter:

GO

CREATE PROCEDURE SelectAllProducts2 @City nvarchar(30)

AS

SELECT \* FROM Product WHERE City= @city

GO

---Execute the stored procedure with one parameter:

EXEC SelectAllProducts2 @City = 'ktm';

---Stored Procedure with multiple parameter:

GO

CREATE PROCEDURE SelectAllProducts3 @City nvarchar(30), @price nvarchar(10)

AS

SELECT \* FROM Product WHERE City = @City AND Price = @Price

GO

---Execute the stored procedure with multiple parameter:

EXEC SelectAllProducts3 @City='china', @price = '80000';

---Delelting stored procedure with multi parameter:

DROP PROCEDURE SelectAllProducts3;

---Stored Procedure:

GO

CREATE PROCEDURE SelectAllProducts3 @City nvarchar(30), @price nvarchar(10)

AS

SELECT \* FROM Product WHERE City = @City AND Price = @Price

GO

---Execute the stored procedure:

EXEC SelectAllProducts3 @City='china', @price = '80000';

---Deleting stored procedure:

DROP PROCEDURE SelectAllProducts3;

---Stored Procedure:

GO

CREATE PROCEDURE SelectAllProducts3 @City nvarchar(30), @price nvarchar(10)

AS

SELECT \* FROM Product WHERE City = @City AND Price = @Price

GO

---Execute the stored procedure:

EXEC SelectAllProducts3 @City='japan', @price = '500';

---ALTER Stored Procedure:

GO

ALTER PROCEDURE SelectAllProducts3

AS

SELECT \* FROM Product

SELECT COUNT(1)AS[Total Count] FROM Product

GO

---Execute the stored procedure:

EXEC SelectAllProducts3;

---Stored Procedure parameter:

---1.) INPUT parameter:

GO

CREATE PROCEDURE SelectAllProcedures4

(

@ProductID int,

@Price money

)

AS

UPDATE Product

SET Price = @Price

WHERE ProductID = @productID

GO

---Execute the stored procedure:

EXEC SelectAllProcedures4 @ProductID = 4, @Price = 3000

---2.) OUTPUT parameter(OUT):

GO

CREATE PROCEDURE SelectAllProducts5

@productid int,

@price int OUTPUT

AS

SELECT @productid = ProductID

FROM Product

WHERE price = @Price

GO

---Deleting the stored procedure:

DROP PROCEDURE SelectAllProducts5;

---Passing OUTPUT parameter copy:

DECLARE @price int

EXECUTE SelectAllProducts5 @ProductID=2, @price = 5000

PRINT @price

---3.) Optional Parameter(INOUT):

GO

CREATE PROCEDURE SelectAllProducts6

(

@productid int,

@price int = 1500

)

AS

UPDATE Product

SET Price = @price

WHERE ProductID = @productid

---Deleting stored procedure:

DROP PROCEDURE SelectAllProducts6;

---Execute the stored procedure:

EXEC SelectAllProducts6;

---------------------------------------------------------------------------------------------

-------------------------------------------5th DAY:------------------------------------------

---------------------------------------------------------------------------------------------

---i). SQL Stored Procedure:

---Syntax:

CREATE PROCEDURE procedure\_name

AS

sql\_statement

GO;

---Execute a Stored Procedure:

EXEC procedure\_name;

---Example:creating table

CREATE TABLE Customer(

CustomerID int INDENTITY(100,11)PRIMARY,

CustomerName varchar(255) NOT NULL,

ContactName varchar(255),

Address varchar(255),

City varchar(255),

PostalCode int,

Country varchar(255)

);

SELECT \* FROM Customer;

DROP TABLE Customer;

---Inserting values:

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Ram', 'Ramesh', 'Talchhikhel', 'Ktm', 4006,'Nepal');

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Logan', 'Shyam', 'Huston', 'WashingtonDC', 4007,'USA');

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('sandeep', 'Rabin', 'Haude', 'Delhi', 4008,'India');

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Sita', 'Sabin', 'Jaulakhel', 'Latipur', 4009,'Nepal');

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Hank', 'Uwjal', 'Hewei', 'Hongkong', 4010,'China');

INSERT INTO Customer(CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Mahomat', 'Amita', 'Kasi', 'Baudi',4011, 'Pakistan');

---1.Selectallcustomers:

CREATE PROCEDURE SelectAllCustomer

AS

SELECT \*FROM Customer

GO;

EXEC SelectAllCustomer;

DROP PROCEDURE SelectAllCustomer;

---2.Stord Prosedure with One parameter:

CREATE PROCEDURE SelectAllCustomer @City nvarchar(30)

AS

SELECT \*FROM Customer WHERE City = @City

GO;

EXEC SelectAllCustomer @City = 'Butwal';

---3.Stord Prosedure with Multiple parameter:

CREATE PROCEDURE SelectAllCustomer @City nvarchar(30),@PostalCode nvarchar(10)

AS

SELECT \*FROM Customer WHERE City = @City AND PostalCode = @PostalCode

GO;

EXEC SelectAllCustomer @City = 'Ktm',@PostalCode ='4006';

---4. Alter Stord Prosedure:

ALTER PROC SelectAllCustomer

AS

SELECT \*FROM Customer

SELECT COUNT(1)AS [Total Count]FROM Customer

GO

EXEC SelectAllCustomer;

DROP PROCEDURE SelectAllCustomer;

---Stored Procedure Parameter:

--1). INPUT:

CREATE PROCEDURE SelectAllCustomer(

@customerId int,

@postalcode money

)

AS

UPDATE Customer

SET PostalCode = @postalcode

WHERE CustomerID = @customerId

SELECT \*FROM Customer

GO;

EXEC SelectAllCustomer @CustomerID = 4, @PostalCode = 26000 ;

--Or

EXEC SelectAllCustomer 4,26000 ;

DROP PROCEDURE SelectAllCustomer;

--2). OUTPUT:

CREATE PROCEDURE SelectAllCustomer

@customerId int,

@postalcode int OUTPUT

AS

SELECT @customerId = CustomerID

FROM Customer

WHERE PostalCode = @postalcode

GO;

DECLARE @postalcode int

EXECUTE SelectAllCustomer @customerId = 2,@postalcode = 100000

PRINT @postalcode

DROP PROCEDURE SelectAllCustomer;

--3). OPTIONAL:

CREATE PROCEDURE SelectAllCustomer(

@customerId int,

@postalcode int = 1000

)

AS

UPDATE Customer

SET PostalCode = @postalcode

WHERE CustomerID = @customerId

GO;

EXEC SelectAllCustomer 4;

DROP PROCEDURE SelectAllCustomer;

---INDEX:

---1.CREATING INDEX:

CREATE TABLE People(

PeopleID int IDENTITY(100,11) PRIMARY KEY,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

DROP TABLE People;

INSERT INTO People(LastName, FirstName, Address, City)

VALUES ('karki','Raj','hattiban','Ktm');

INSERT INTO People(LastName, FirstName, Address, City)

VALUES ('b.k','Raju','hattiban','Ktm');

INSERT INTO People(LastName, FirstName, Address, City)

VALUES ('pariyar','Rupa','hattiban','Ktm');

INSERT INTO People(LastName, FirstName, Address, City)

VALUES ('koirala','Isha','hattiban','Ktm');

SELECT \*FROM People;

---CREATING INDEX WITH LASTNAME:

CREATE INDEX idx\_pname

ON People(LastName);

---CREATING INDEX WITH COMBINATION OF LASTNAME AND FIRSTNAME:

CREATE INDEX idx\_pname

ON People(LastName,FirstName);

---DROPING INDEX:

DROP INDEX People.idx\_pname;

---TRIGGER IN SQL SERVER:

---SYNTAX:

CREATE TRIGGER schema.trigger\_name

ON table\_name

AFTER {INSERT,UPDATE,DELETE}

{NOT FOR REPLICATION}

AS

{SQL Statements}

----TRIGGER:INSERT:

CREATE TABLE Employeeee(

Id INT IDENTITY(100,11) PRIMARY KEY,

Name VARCHAR(45),

Salary INT,

Gender VARCHAR(12),

Department INT

);

---INSERITNG:

INSERT INTO Employeeee

('Raju', 82000,'Male',3),

('Rabina', 52000,'Female',2),

('Rani', 25000,'Female',1),

('Ramesh', 47000,'Male',2),

('Rani', 46000,'Female',3)

SELECT \*FROM Employeeee;

---1.TRIGGER: UPDATE:

CREATE TABLE Employeeee\_Audit\_Test

(

Id int IDENTITY,

Audit\_Action text

)

---Example:

CREATE TRIGGER trInsertEmployee

ON Employee

FOR INSERT

AS

BEGIN

Declare @Id int

SELECT @Id = Id FROM inserted

INSERT INTO Employeeee\_Audit\_Test

VALUES('New epmloyeeee with Id='+CAST(@Id AS VARCHAR(10))+'is added at'+CAST (Getdate() AS VARCHAR(22)))

END

--- ADDING :

INSERT INTO Employeeee VALUES('Rabin',62000,'Male',3)

SELECT \*FROM Employeeee\_Audit\_Test;

---3.TRIGGER:DELETE:

CREATE TRIGGER trDeleteEmployee

ON Employee

FOR DELETE

AS

BEGIN

Declare @Id int

SELECT @Id = Id FROM deleted

INSERT INTO Employeeee\_Audit\_Test

VALUES('An existing employeeee with Id='+CAST(@Id AS VARCHAR(10))+'is deleted at'+CAST (Getdate() AS VARCHAR(22)))

END

DELETE FROM Employeeee WHERE Id = 2;

SELECT \*FROM Employeeee\_Audit\_Test;

--TRIGGER: REMOVE DML:

DROP TRIGGER trInsertEmployeeee;