Create database CONCEPTOS

CREATE TABLE PERSONAS(

IDPERSONA INT IDENTITY (1,1),

NOMBRE NVARCHAR (100),

APELLIDO NVARCHAR (100),

EDAD INT,

IDOCUPACION INT,

PRIMARY KEY (IDPERSONA),

FOREIGN KEY (IDOCUPACION) REFERENCES OCUPACIONES (IDOCUPACION)

)

CREATE TABLE OCUPACIONES (

IDOCUPACION INT IDENTITY (1,1),

NOMBRE\_DEL\_CARGO NVARCHAR (100),

ANTIGUEDAD\_AÑOS INT,

PROVINCIA NVARCHAR (100),

PRIMARY KEY (IDOCUPACION)

)

CREATE TABLE EMPRESAS (

IDEMPRESA INT IDENTITY (1,1),

NOMBRE\_EMPRESA NVARCHAR (100),

NUMERO\_EMPLEADOS INT,

IDOCUPACION INT,

PRIMARY KEY (IDEMPRESA),

FOREIGN KEY (IDOCUPACION) REFERENCES OCUPACIONES (IDOCUPACION)

)

Tipos de variable

Int enteros

Nvarchar (100) un espacio para 100 caracteres

Money tipos de moneda

numeric (18,0)

----INDICES: SIRVEN PARA ORDERNAR ALFABETICAMENTE LOS DATOS

EXECUTE sp\_helpindex 'LISTADO DE NOMBRES'

--SE EJECUTA EL COMANDO SP\_HELPINDEX (QUE VIENE POR DEFAULT) Y SE LE DA UN NOMBRE

CREATE CLUSTERED INDEX LISTADOS\_DE\_NOMBRES

--SE CREA EL NOMBRE DEL INDICE

ON PERSONAS (NOMBRE)

--SE INDICA LA TABLA Y LA COLUMNA, ¡¡¡NO FUNCIONA SI HAY PRIMARY EN LA TABLA EN CUESTION!!!

--BORRAR BASE DE DATOS/TABLA/COLUMNA

DROP DATABASE

DROP TABLE

DROP COLUMN

----ELIMINAR EL CONTENIDO DE UNA TABLA

DELETE TABLE

Para mostrar en pantalla

Print …………

INSERTAR DATOS EN UNA DE LAS TABLAS (SIEMPRE PONER LOS CAMPOS, AUNQUE NO SEA NECESARIO

--PORQUE ASI LO PIDE EL PROFESOR)

--INSERT INTO PERSONAS (NOMBRE, APELLIDO,EDAD) VALUES ('AGUSTINA','MENDEZ',38)

--INSERT INTO OCUPACIONES (NOMBRE\_DEL\_CARGO, ANTIGUEDAD\_AÑOS, PROVINCIA) VALUES ('ABOGADA',15,'MISIONES')

--ALTER: CON ESTO PODEMOS AGRAGAR O QUITAR COLUMNAS A UNA TABLA DE MANERA PARTICULAR

--ALTER TABLE EMPRESAS

--ADD

--RUBRO NVARCHAR (30)

--CON ESTO PODEMOS BORRAR UNA COLUMNA EN PARTICULAR DE UNA DETERMINADA TABLA

--ALTER TABLE [NOMBRE DE LA TABLA]

--DROP COLUMN [NOMBRE DE LA COLUMNA]

----LIKE: PERMITE BUSCAR DATOS POR MEDIO DE ESTIMACIONES

--SELECT \* FROM PERSONAS WHERE NOMBRE LIKE '%ON%'

----ACTUALIZAR DATOS EN UNA TABLA DE MANERA GENERAL

--UPDATE PERSONAS SET NOMBRE ='JOSE', APELLIDO='PEREZ', EDAD=25

----ACTUALIZAR DATOS DE MANERA PARTICULAR, INGRESANDO UNA NUEVA LINEA/ROW A LA TABLA

--UPDATE PERSONAS SET NOMBRE ='URIEL', APELLIDO='SANTARELLI',EDAD=30 WHERE IDPERSONA =1

----SELECT: SIRVE PARA MOSTRAR LOS PRIMEROS ELEMENTOS DE UNA TABLA EN CUESTION

--SELECT TOP (5)\* FROM PERSONAS

----AQUI HACEMOS LOS MISMO PERO EN LUGAR DE PEDIR UNA CANTIDAD, PEDIMOS UN PORCENTAJE

--SELECT TOP (20) PERCENT \* FROM OCUPACIONES

----BETWEEN: SIRVE ESTABLECER UN RANGO DE BUSQUEDA ENTRE 2 VALORES

--SELECT \* FROM EMPRESAS WHERE (IDEMPRESA BETWEEN 3 AND 5)

----IN: PERMITE PERZONALIZAR UNA CONSULTA DICIENDO QUE ES LO QUE ESTAMOS BUSCANDO

--SELECT \* FROM OCUPACIONES WHERE (NOMBRE\_DEL\_CARGO IN ('TECNICO PROFESIONAL'))

----CONSULTAS MULTIPLES: SE ULTILIZA PARA BUSCAR ELEMENTOS MEDIANTE VARIOS PARAMETROS

--SELECT \* FROM EMPRESAS WHERE (NUMERO\_EMPLEADOS LIKE 200000) OR (NOMBRE\_EMPRESA LIKE 'AMAZON')

--ORDENAR EN ORDERN ASCENDENTE/DESCENDENTE LA SELECCION DEL CONTENIDO DE LA TABLA SEGUN EL CRITERIO

--SELECT TOP (100) PERCENT IDEMPRESA, NUMERO\_EMPLEADOS, IDOCUPACION, NOMBRE\_EMPRESA

--FROM dbo.EMPRESAS ORDER BY IDEMPRESA ASC

--PROCEDIMIENTOS ALAMCENADOS / STORE PROCEDURES SE ENCARGA DEL CONTROL DE ERRORES MEDIANTE TRAY/CATCH

--CREATE PROCEDURE SUMAR(

--@N1 INT,

--@N2 INT,

--@RESULTADO INT OUTPUT

--)

--AS

--BEGIN

--SET @RESULTADO = @N1 + @N2

--BEGIN TRY

--DECLARE @RESULTADO INT

--EXEC SUMAR 1,2,@RESULTADO OUTPUT

--SELECT @RESULTADO

--END TRY

--BEGIN CATCH

--SET @RESULTADO = -1

--END CATCH

--END

--ALTER PROCEDURE SUMAR (

--@N1 INT,

--@N2 INT,

--@RESULTADO BIT OUTPUT,

--@RESULTADOMENSAJE NVARCHAR(100) OUTPUT

--)

--AS

--BEGIN

--DECLARE @SUMA INT

-- BEGIN TRY

-- BEGIN TRANSACTION

-- insert into PERSONAS(NOMBRE,APELLIDO,EDAD)

-- values ('prueba','prueba',15)

-- SET @SUMA = @N1 + @N2

-- SET @RESULTADO=1

-- SET @RESULTADOMENSAJE='OK'

-- SELECT @SUMA

-- COMMIT TRANSACTION

-- END TRY

-- BEGIN CATCH

-- SET @RESULTADO = 0

-- SET @RESULTADOMENSAJE=ERROR\_MESSAGE()

-- ROLLBACK TRANSACTION

-- END CATCH

--END

--DECLARE @RESULTADO bit

--DECLARE @RESULTADOMENSAJE nvarchar(100)

--EXEC SUMAR 2147483647,2000000,@RESULTADO OUTPUT,@RESULTADOMENSAJE OUTPUT

--SELECT @RESULTADO,@RESULTADOMENSAJE

--DECLARE @RESULTADO bit

--DECLARE @RESULTADOMENSAJE nvarchar(100)

--EXEC SUMAR 1,1,@RESULTADO OUTPUT,@RESULTADOMENSAJE OUTPUT

--SELECT @RESULTADO,@RESULTADOMENSAJE

--SELECT \* FROM PERSONAS

--SELECT CURRENT\_USER ESTE COMANDO NOS PERMITE SABER QUE USUARIO ESTA LOGUEADO

--TRIGGERS/DISPARADORES: PROCEDIMIENTO ALMACENADO QUE SE EJECUTA CUANDO SE INTENTA

--MODIFICAR (INSERTAR, ACTUALIZAR, ELIMINAR) LOS DATOS DE UNA TABLA (O VISTA)

--CREATE TABLE AUDITORIAS (

--IDAUDITORIA INT IDENTITY (1,1),

--IDELEMENTO NVARCHAR (20),

--TABLA NVARCHAR,

--ACCION NVARCHAR

--PRIMARY KEY (IDAUDITORIA)

--)

--DROP TABLE AUDITORIAS

--CREATE TRIGGER DIS\_AUDITORIA

--ON EMPRESAS

--FOR INSERT

--AS

--BEGIN

--INSERT INTO AUDITORIA

--SELECT IDEMPRESA, 'RUBRO','NOMBRE\_EMPRESA'

--FROM inserted

--END

--MANEJO DE EXCEPCIONES: Si ocurre un error dentro de la transacción en un bloque TRY

--inmediatamente se dirige al bloque CATCH.

--Para poder regresar la información del error tenemos nuevas funciones en SQL Server, veamos cuales son:

--BEGIN TRY

-- DECLARE @TOTAL INT;

-- SET @TOTAL = 20;

-- SELECT @TOTAL/0

--END TRY

--BEGIN CATCH

-- SELECT

-- ERROR\_NUMBER() AS Numero\_de\_Error,

-- ERROR\_SEVERITY() AS Gravedad\_del\_Error,

-- ERROR\_STATE() AS Estado\_del\_Error,

-- ERROR\_PROCEDURE() AS Procedimiento\_del\_Error,

-- ERROR\_LINE() AS Linea\_de\_Error,

-- ERROR\_MESSAGE() AS Mensaje\_de\_Error;

--END CATCH

CREATE TABLE PRODUCTOS (

IDPRODUCTO INT IDENTITY (1,1),

NOMBREdePRODUCTO NVARCHAR (100),

DESCRIPCION NVARCHAR (100),

PRECIO INT,

STOCK INT,

PRIMARY KEY (IDPRODUCTO)

)

SELECT AVG (PRECIO) FROM PRODUCTOS

SELECT AVG (STOCK) FROM PRODUCTOS

SELECT NOMBREdePRODUCTO, PRECIO FROM PRODUCTOS WHERE PRECIO >(SELECT AVG (PRECIO)FROM PRODUCTOS);

SELECT COUNT (\*) FROM PRODUCTOS

SELECT COUNT (DISTINCT NOMBREdePRODUCTO) FROM PRODUCTOS

SELECT MAX (PRECIO) FROM PRODUCTOS

SELECT NOMBREdePRODUCTO, PRECIO FROM PRODUCTOS WHERE PRECIO >(SELECT MIN(PRECIO) FROM PRODUCTOS)

SELECT NOMBREdePRODUCTO FROM PRODUCTOS ORDER BY PRECIO DESC

SELECT MIN (STOCK) FROM PRODUCTOS

SELECT SUM (PRECIO) FROM PRODUCTOS

Crear operaciones

Créate procedure SP\_operacioneslistar (crear procedimiento “listaroperaciones”

(

@id\_Operaciones int,

)

as

begin

Select \* from operaciones

end

Créate procedure SP\_operaciones\_Obtener

(

@idOperaciones int,

)

as

begin

Select \* from operaciones where IdOperacion=@IdOperacion

end

llamar a procedimiento

exec SP\_operacioneslistar

exec SP\_operaciones

para alterar operaciones usar (alter)

Acceso a procedimientos almacenados

exec [dbo].[…]

create procedure SP\_Operaciones\_Listarasbeginselect \* from Operacionesendexec SP\_Operaciones\_Obtener @IdUsuarioVendedor=,@IdOperacion= 1 , @Denominacion= 'nombre' , @Cantidad=300alter procedure SP\_Operaciones\_Obtener(@IdOperacion int=null,@Cantidad int=300,@IdUsuarioVendedor int =null,@Denominacion int =null,@DenomincionIgual bit=0)asbegin if (@DenomincionIgual=1) begin select top (@Cantidad) \* from Operaciones where ((@IdOperacion is null) OR (Operaciones.IdOperacion = @IdOperacion )) AND ((@Denominacion is null) OR (@Denominacion = '') OR (Operaciones.Denominacion=@Denominacion)) AND ((@IdUsuarioVendedor is null) OR (Operaciones.IdUsuarioVendedor = @IdUsuarioVendedor)) end else begin select top (@Cantidad) \* from Operaciones where ((@IdOperacion is null) OR (Operaciones.IdOperacion = @IdOperacion )) AND ((@Denominacion is null) OR (@Denominacion = '') OR (Operaciones.Denominacion LIKE '%' + @Denominacion + '%' )) AND ((@IdUsuarioVendedor is null) OR

From Ignacio Rodriguez to Everyone: 08:15 PM

create procedure SP\_Operaciones\_Listarasbeginselect \* from Operacionesendexec SP\_Operaciones\_Obtener @IdUsuarioVendedor=,@IdOperacion= 1 , @Denominacion= 'nombre' , @Cantidad=300alter procedure SP\_Operaciones\_Obtener(@IdOperacion int=null,@Cantidad int=300,@IdUsuarioVendedor int =null,@Denominacion int =null,@DenomincionIgual bit=0)asbegin if (@DenomincionIgual=1) begin select top (@Cantidad) \* from Operaciones where ((@IdOperacion is null) OR (Operaciones.IdOperacion = @IdOperacion )) AND ((@Denominacion is null) OR (@Denominacion = '') OR (Operaciones.Denominacion=@Denominacion)) AND ((@IdUsuarioVendedor is null) OR (Operaciones.IdUsuarioVendedor = @IdUsuarioVendedor)) end else begin select top (@Cantidad) \* from Operaciones where ((@IdOperacion is null) OR (Operaciones.IdOperacion = @IdOperacion )) AND ((@Denominacion is null) OR (@Denominacion = '') OR (Operaciones.Denominacion LIKE '%' + @Denominacion + '%' )) AND ((@IdUsuarioVendedor is null) OR