SESIÓN
05

INSTRUCCIONES DE MANTENIMIENTO DE DATOS

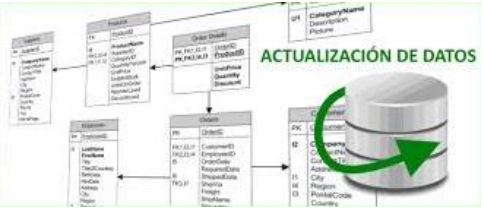
- Ingreso de registros instrucción insert
- Actualización de datos instrucción udpate
- Eliminación de registros instrucción delete

MANTENIMIENTO DE DATOS DE LAS TABLAS

Uso de comandos INSERT, UPDATE y DELETE en base de datos de SQL Server

Ingresar datos

Actualizar datos



Eliminar datos

Resultados

DELETE FROM <nombre de tabla>
WHERE <condición lógica>

INGRESAR REGISTROS A UNA TABLA

Aplicación cliente

Aplicación servidor en la ventana Servicios de Windows

ACTUALIZAR DATOS DE UNA TABLA

Aplicación cliente

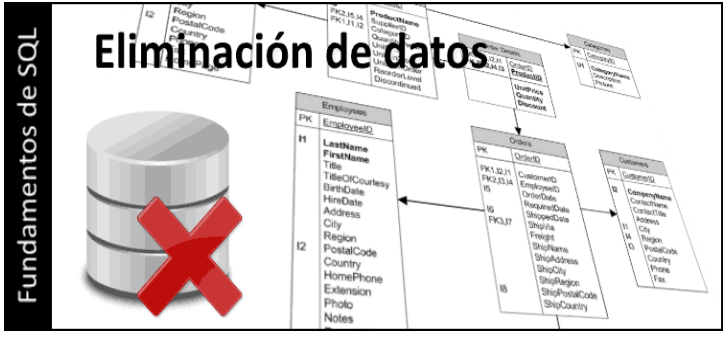
```
1  UPDATE customers
2     SET forename = 'Sean',
3         surname = 'Longs'
4   WHERE forename = 'John';
```

Script Output | Query Result | Task completed in

1 rows updated.

Aplicación servidor en la ventana Servicios de Windows

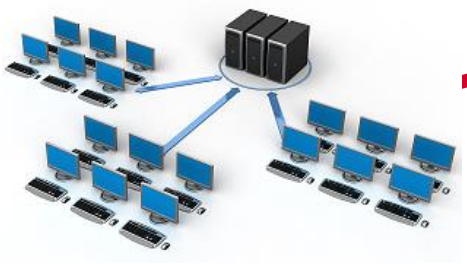
ELIMINAR REGISTROS DE UNA TABLA



Aplicación cliente

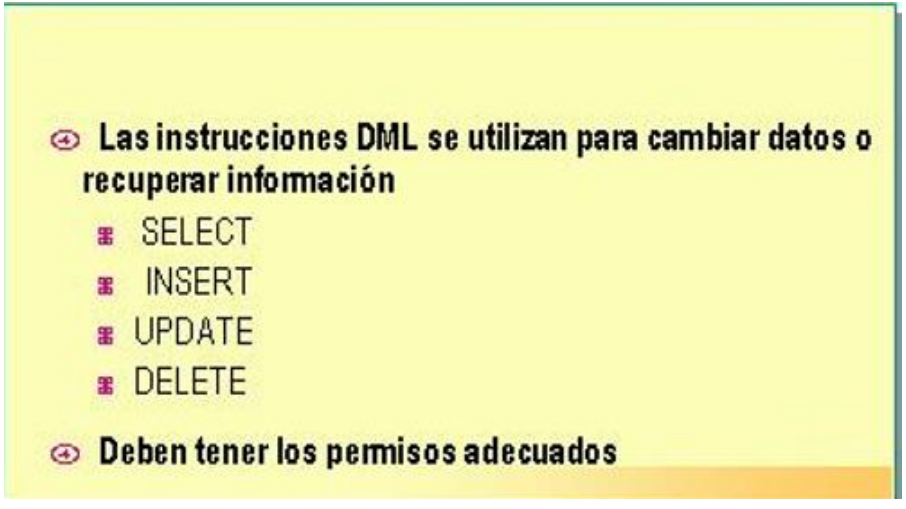
MANTENIMIENTO DE DATOS (CRUD)

C reate, R ead, U pdate , D elete



/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS



Lenguaje de manipulación de datos (DML). Seleccionar, insertar, actualizar y eliminar datos.

INGRESO DE DATOS

La Instrucción INSERT

El proceso de inserción de filas consiste en añadir a una tabla una o más filas y en cada fila todos o parte de sus campos.

INGRESO DE DATOS

SINTAXIS 1

```
INSERT [INTO] nombre_tabla[ ( lista_de_columnas ) ]
              VALUES( lista_de_valores )
```

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

```
EXEC SP_HELP TB_PROVEEDOR
GO
```

159 %

Results | Messages

	Name	Owner	Type	Created_datetime
1	Tb_Proveedor	dbo	user table	2021-04-02 23:14:20.160

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Cod_prv	nchar	no	8			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
2	Raz_soc_prv	nvarchar	no	100			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
3	Dir_prv	nvarchar	no	100			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
4	Tel_prv	nvarchar	no	20			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
5	Ruc_Prv	nchar	no	22			no	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
6	Rep_ven	nvarchar	no	60			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
7	Id_Ubigeo	nchar	no	12			yes	(n/a)	(n/a)	SQL_Latin1_General_CP1_CI_AS
8	Fec_Registro	datetime	no	8			yes	(n/a)	(n/a)	NULL

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	PK__Tb_Proveedor__37A5467C	clustered, unique, primary key located on PRIMARY	Cod_prv
2	UQ_Ruc_prv	nonclustered, unique, unique key located on PRIMA...	Ruc_Prv

	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

```
--INSERTAMOS EL REGISTRO COMPLETO
INSERT INTO TB_PROVEEDOR
VALUES  ('V207','Distribuidora ACME SCRL','Jr. Las Amapolas 344','4457611','00998812310',
'Carla Gutierrez','140110',Getdate(),'jleon',Null,Null,1)
GO
-- COMPROBAMOS
SELECT * FROM TB_PROVEEDOR
GO
```

Messages

```
(1 row(s) affected)
```

COMPROBANDO…

206	V206	ELIZABET...	NULL		10080791807	NULL	120103	2020-09-10 14:29:17.937	jleon	NULL	NULL	1
207	V207	Distribuidor...	Jr. Las Amapolas 344	4457611	00998812310	Carla Gutierrez	140110	2022-05-15 15:09:09.023	jleon	NULL	NULL	1

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

```
-- Insertando varios registros
Insert into Tb_UnidadMedida
values ('CAJA POR 100')
Insert into Tb_UnidadMedida
values ('CAJA POR 200')
Insert into Tb_UnidadMedida
values ('CAJA POR 500')
go
--Comprobemos
SELECT * FROM Tb_UnidadMedida
GO
```

159 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2022-05-15T15:37:19.5806844-05:00
```

	Id_UM	Des_UM
10	10	CAJA POR 6
11	11	CAJA POR 15
12	12	CAJA POR 20
13	13	CAJA POR 24
14	14	BOLSA POR 100
15	15	BOLSA POR 200
16	16	CAJA POR 100
17	17	CAJA POR 200
18	18	CAJA POR 500

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

- Ingresando varios registros a la vez

```
-- Desde la version SQL 2008 se pueden insertar varios registros
-- con un solo Insert
Insert into Tb_UnidadMedida
values ('BOLSA POR 20'),
       ('BOLSA POR 50'),
       ('EMPAQUE POR 100')
go
--Comprobemos
SELECT * FROM Tb_UnidadMedida
GO
```

```
Messages

(3 rows affected)

Completion time: 2022-05-15T15:43:59.5431234-05:00
```

	Id_UM	Des_UM
10	10	CAJA POR 6
11	11	CAJA POR 15
12	12	CAJA POR 20
13	13	CAJA POR 24
14	14	BOLSA POR 100
15	15	BOLSA POR 200
16	16	CAJA POR 100
17	17	CAJA POR 200
18	18	CAJA POR 500
19	19	BOLSA POR 20
20	20	BOLSA POR 50
21	21	EMPAQUE POR 100

INGRESO DE DATOS

SINTAXIS 2

```
INSERT [INTO] nombre_tabla_destino
SELECT [ ( lista_de_columnas ) ]
                FROM ( nombre_tabla_origen )
```

Podemos insertar registros tomando valores que provienen de un Select. Para implementar esta forma de inserción debemos asegurarnos que la tabla de destino esté creada.

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

- Ingresando registros desde un resultado Select

```
Paso  1: Crear una tabla llamada TopVendedor con los campos
codigo, nombres  apellidos , sueldo , fecha de ingreso de
vendedores y Tipo de vendedor de la tabla Tb_Vendedor*/
CREATE TABLE TOPVENDEDOR (
    Codigo VARCHAR(3) PRIMARY KEY ,
    Nombres VARCHAR(50) NULL,
    Apellidos VARCHAR(50) NULL,
    Sueldo REAL NULL,
    FechaIngreso DATETIME NULL,
    Tipo Int NULL )
GO
```

```
/*Paso 2 :
  Insertar a la tabla TopVendedor, todos los vendedores
  de la tabla Tb_Vendedor que perciban un sueldo mayor de 1500*/

INSERT INTO TOPVENDEDOR
SELECT Cod_ven,Nom_ven,Ape_ven, Sue_ven, Fec_Ing ,
Tip_ven  FROM TB_VENDEDOR
WHERE SUE_VEN > 1500
GO

/* Paso 3: Comprobamos*/
SELECT *
FROM  TOPVENDEDOR
GO
```

	Codigo	Nombres	Apellidos	Sueldo	FechaIngreso	Tipo
1	V01	Juana	Masias	3200	2003-12-17 00:00:00.000	1
2	V02	Eva M...	Abad L...	3455.5	2005-04-18 00:00:00.000	1
3	V03	Eduardo	Zavala	3200	2004-11-02 00:00:00.000	1
4	V04	Renato	Salas	3200	2004-11-16 00:00:00.000	1
5	V05	Julio	Vega	3200	1997-05-14 00:00:00.000	1
6	V06	Hugo	Ruiz	3200	2004-11-08 00:00:00.000	1
7	V07	Jose	Palacios	3200	1997-02-22 00:00:00.000	1
8	V08	Juan	Salazar	3200	1996-08-15 00:00:00.000	1
9	V09	Felipe	Martinez	3200	2004-11-08 00:00:00.000	1
10	V10	Carmen	Montoya	3200	2004-11-15 00:00:00.000	2
11	V12	Feman...	Salazar	2500	2004-11-03 13:50:10.000	2
12	V16	Juan	Perez ...	1701	2020-07-05 16:51:08.130	2
13	V17	Cesar	Huaman	1714	2020-07-05 17:10:12.583	2
14	V18	Ruth	Allca G...	1648	2020-07-05 17:13:21.597	2
15	V19	Juana	Castillo	1506	2020-07-05 17:00:18.157	2

INGRESO DE DATOS

Empleando Select

```
SELECT [ ( lista_de_columnas ) ]
                FROM ( nombre_tabla_origen )
        [INTO] nombre_tabla_destino
```

Podemos insertar registros (sin emplear la sentencia Insert) derivando el resultado de un Select a una tabla destino. En esta forma, no se requiere crear la tabla destino previamente, dado que la ejecución del Select construye la tabla destino.
NOTA: Se sugiere emplear esta forma de generación de tablas para crear tablas Backup previas a cualquier actualización o eliminación de registros en una tabla principal.

/ INGRESO DE REGISTROS INSTRUCCIÓN INSERT

INGRESO DE DATOS

- Derivando el resultado de un Select a una tabla destino

```
Empleando Select Into crear la tabla  Tb_ClientesTop
con los campos Cod_cli,raz_soc_cli,dir_cli y antiguedad,
de aquellos clientes con mas de 10 años de haberse registrado.*/

--Generamos la tabla y la llenamos con los registros pedidos
SELECT COD_CLI,RAZ_SOC_CLI,RUC_CLI,DIR_CLI,
DATEDIFF(YEAR,FEC_REG,GETDATE()) AS ANTIGUEDAD
INTO TB_CLIENTESTOP
FROM TB_CLIENTE
WHERE DATEDIFF(YEAR,FEC_REG,GETDATE())>10
GO
--Comprobando
SELECT * FROM TB_CLIENTESTOP
GO
```

	COD_CLI	RAZ_SOC_CLI	RUC_CLI	DIR_CLI	ANTIGUEDAD
1	C001	FINSETH	48632081112	Av. Los Viñedos 150	31
2	C002	ORBI	57031642221	Av. Emilio Cavene...	32
3	C003	SERVIEMSA	75012403231	Jr. Collagate 522	27
4	C004	ISSA	46720159214	Calle Los Aviadore...	30
5	C005	MASS	83175942998	Av. Tomas Marsan...	30
6	C006	BERKER	54890124200	Av. Los Proceres 5...	33
7	C007	FIDENZA	16204790012	Jr. El Niquel 282	31
8	C008	INTECH	34021824991	Av. San Luis 2619 ...	25
9	C009	PROMINENT	43233519100	Jr. Iquique 132	29
10	C010	LANDU	30405261192	Av.Nicolas de Ayll...	28
11	C011	FILAUR	70345201233	Av. El Santuario 1...	32
12	C012	SUCERTE	62014503344	Jr. Grito de Huaura...	33
13	C013	HAYASHI	42847990366	Jr. Ayacucho 359	32

/ ACTUALIZACIÓN DE DATOS INSTRUCCIÓN UDPATE

/ ACTUALIZACIÓN DE DATOS INSTRUCCIÓN UDPATE

INSTRUCCIÓN UPDATE

- Actualizar datos de una Tabla (Cambiar datos)

INSTRUCCIÓN UPDATE

Sintaxis

```
UPDATE nombre_tabla
        SET   columnaX = expresiónX ,
              columnaP = expresiónP , …
[ WHERE (condición_de_las_filas_a_actualizar)]
```

Nota: A pesar que la clausula WHERE es opcional, el desarrollador debe considerar su empleo, dado que si no se incluye, el alcance de la actualización será a todos los registros de la tabla, sin lugar a la opción “deshacer”

/ ACTUALIZACIÓN DE DATOS INSTRUCCIÓN UDPATE

INSTRUCCIÓN UPDATE

```
/*2. ACTUALIZAR LOS DATOS NOMBRE, APELLIDO Y FECHA DE INGRESO DEL VENDEDOR
-- V02 CON LOS DATOS 'CECILIA','ZEVALLOS', Y LA FECHA ACTUAL RESPECTIVAMENTE */
-- Verificamos el estado actual..
Select * from Tb_Vendedor
go
-- Actualizamos
Update Tb_Vendedor set nom_ven='Cecilia' , ape_ven ='Zevallos' , fec_ing=GETDATE ()
where cod_ven='V02'
GO
--Comprobamos...
Select * from Tb_Vendedor
go
```

	Cod_ven	Nom_ven	Ape_ven	Sue_ven	Fec_ing	Tip_ven	Dni_ven	Email_ven	Cod_Supervisor	Fec_Registro	Usu_Registro	Fec_Ult_Mod
1	V01	Juana	Masias	3200.00	2003-12-17 00:00:00.000	1	12345678	jmasiasv01@leoncorp.com	V01	2020-09-08 11:15:28.353	jleon	NULL
2	V02	Eva Maria	Abad Lopez	3455.50	2005-04-18 00:00:00.000	1	12354579	eabadv02@leoncorp.com	V06	2020-09-08 11:15:28.353	jleon	2020-10-09 19
3	V03	Eduardo	Zavala	3200.00	2004-11-02 00:00:00.000	1	15459879	ezavalav03@leoncorp.com	V03	2020-09-08 11:15:28.353	jleon	NULL
4	V04	Renato	Salas	3200.00	2004-11-16 00:00:00.000	1	25458797	rsalasv04@leoncorp.com	V04	2020-09-08 11:15:28.353	jleon	NULL
5	V05	Julio	Vega	3200.00	1997-05-14 00:00:00.000	1	32165478	jvegav05@leoncorp.com	V05	2020-09-08 11:15:28.353	jleon	NULL
6	V06	Hugo	Ruiz	3200.00	2004-11-08 00:00:00.000	1	32564897	hruizv06@leoncorp.com	V06	2020-09-08 11:15:28.353	jleon	NULL
7	V07	Jose	Palacios	3200.00	1997-02-22 00:00:00.000	1	44565423	jpalaciosv07@leoncorp.com	V07	2020-09-08 11:15:28.353	jleon	NULL

Messages

```
(1 row affected)

Completion time: 2022-05-15T16:52:53.7230786-05:00
```

	Cod_ven	Nom_ven	Ape_ven	Sue_ven	Fec_ing	Tip_ven	Dni_ven	Email_ven	Cod_Supervisor	Fec_Registro	Usu_Registro	Fec_Ult_Mod
1	V01	Juana	Masias	3200.00	2003-12-17 00:00:00.000	1	12345678	jmasiasv01@leoncorp.com	V01	2020-09-08 11:15:28.353	jleon	NULL
2	V02	Cecilia	Zevallos	3455.50	2022-05-15 16:52:53.710	1	12354579	eabadv02@leoncorp.com	V06	2020-09-08 11:15:28.353	jleon	2020-10-09 19:28:3
3	V03	Eduardo	Zavala	3200.00	2004-11-02 00:00:00.000	1	15459879	ezavalav03@leoncorp.com	V03	2020-09-08 11:15:28.353	jleon	NULL
4	V04	Renato	Salas	3200.00	2004-11-16 00:00:00.000	1	25458797	rsalasv04@leoncorp.com	V04	2020-09-08 11:15:28.353	jleon	NULL

/ ELIMINACIÓN DE REGISTROS INSTRUCCIÓN DELETE

/ ELIMINACIÓN DE REGISTROS INSTRUCCIÓN DELETE

INSTRUCCIÓN DELETE

- Borrando registros

INSTRUCCIÓN DELETE

Sintaxis

```
DELETE [FROM] nombre_tabla
[WHERE condición_de_las_filas_a_eliminar]
GO
```

Nota: Al igual que con UPDATE la clausula WHERE es opcional, pero el desarrollador debe considerar su empleo en el comando DELETE, dado que si no se incluye, puede eliminar todos los registros de la tabla involuntariamente, sin lugar a la opción “deshacer”

/ ELIMINACIÓN DE REGISTROS INSTRUCCIÓN DELETE

INSTRUCCIÓN DELETE

```
--PARA NO ALTERAR LA INFORMACION DE NUESTRAS TABLAS ORIGINALES,
-- CREAREMOS COPIAS DE LAS MISMAS Y EN ELLAS APLICAREMOS EJEMPLOS DE ELIMINACION


--COPIA DE LA TABLA CLIENTES
SELECT *
  INTO COPIACLIENTE
  FROM TB_CLIENTE
GO
-- COMPROBAMOS
SELECT * FROM COPIACLIENTE
GO
```

```
/*  ELIMINE DE LA TABLA COPIA CLIENTE A AQUELLOS CLIENTES
CON MENOS DE 2 AÑOS DE ANTIGUEDAD*/


Delete from COPIACLIENTE
where  datediff(d,fec_reg,getdate())/365 <2
go


-- COMPROBAMOS
SELECT * FROM COPIACLIENTE
GO
```

/ ELIMINACIÓN DE REGISTROS INSTRUCCIÓN DELETE

NOTA IMPORTANTE:

En ambientes de produccion es poco probable que un registro insertado en una tabla maestra se elimine. Por ejemplo, si en un sistema de ventas se inserta un nuevo cliente , este no puede ser eliminado (bajo escenarios normales) por mas que sea calificado como cliente “mal pagador” o sea un cliente que se ha registrado pero no efectua compra alguna. Si por alguna regla de negocio o requerimiento en particular se necesita no contar mas con ese cliente , lo que se sugiere es cambiar su estado de activo (1) a inactivo (0) haciendo un Update sobre dicho campo. Por ello es importante considerar los campos de estado dentro de las tablas maestras de nuestras bases de datos.

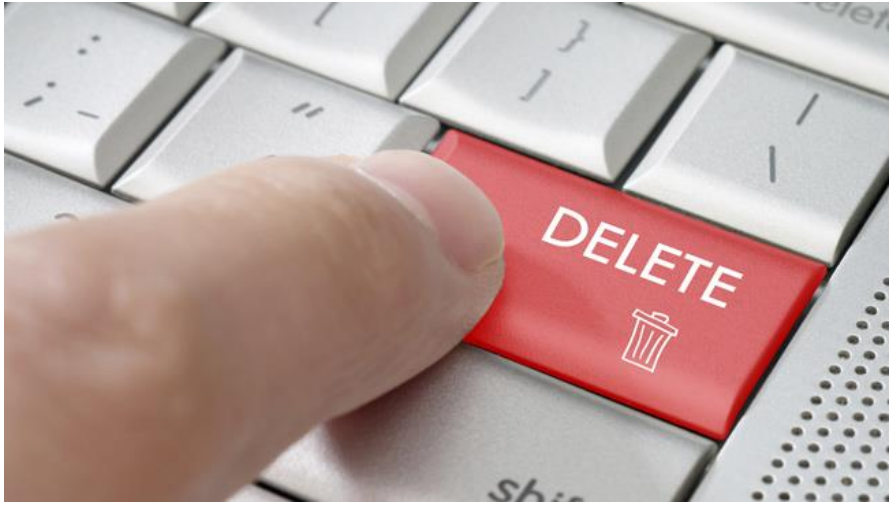
```
-- Inactivamos el  cliente C009
Update Tb_Cliente set est_cli=0 where cod_cli ='C009'
go

-- Activamos el cliente C456
Update Tb_Cliente set est_cli=1 where cod_cli='C456'
go

-- Seleccionamos el codigo  y razon social
-- de clientes activos ordenados por razon social
Select cod_cli,raz_soc_cli from Tb_Cliente where cod_cli=1
order by raz_soc_cli
go
```

/ ELIMINACIÓN DE REGISTROS INSTRUCCIÓN DELETE

DIFERENCIA ENTRE EL TRUNCATE TABLE Y EL DELETE

- Comando de Eliminación de información, las 2 sentencias se pueden deshacer con un ROLLBACK. TRUNCATE reiniciará el contador para una tabla que contenga una columna IDENTITY, mientras que el DELETE mantendrá el contador de la tabla para una columna IDENTITY. TRUNCATE es un comando DDL(lenguaje de definición de datos) mientras que DELETE es un DML(lenguaje de manipulación de datos)

```
Truncate table Names
Insert into Names (Name) values('Narendra')
Insert into Names (Name) values('Rahul')
Insert into Names (Name) values('Harish')
Insert into Names (Name) values('Vijay')
Insert into Names (Name) values('Shubham')
Select * from Names
```



/ ACTIVIDAD

OBJETIVO

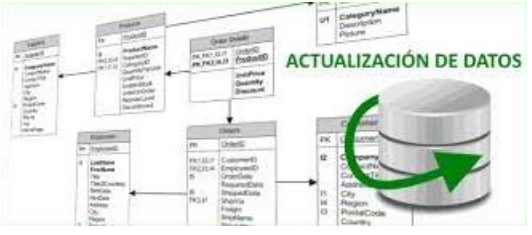
- Reforzar el procedimiento para realizar tareas de mantenimiento de datos en un servidor SQL Server utilizando el cliente SQL Server Management Studio.
- Reforzar el uso de las Instrucciones INSERT, UPDATE Y DELETE.
- Emplear los Scripts 5.1 , 5.2 y 5.3 del material de la sesión.

/ RESUMEN

/ RESUMEN

- Se logro conocer y aplicar las tareas de Mantenimiento de datos de las Tablas

Uso de comandos INSERT, UPDATE y DELETE en bases de datos de SQL Server



DELETE FROM <nombre de tabla>
WHERE <condición lógica>

/ RESUMEN

- Esta clara ahora la competencia de realizar tareas de Mantenimiento de datos.

Mantenimiento de datos

/ RESUMEN

- Finalmente ya estamos capacitados para Ingresar, Actualizar y Eliminar registros de las Tablas de una Base de Datos.