

II. Comandos de Administración de Base de datos

II. Comandos de Administración de Base de datos

- Lenguaje de descripción de datos (DDL): Definir tipos datos, estructuras y restricciones de datos
- Lenguaje de manipulación de datos(DML): Permite insertar, actualizar, eliminar y obtener datos, proporcionando servicios generales de consulta a través de SQL

DDL - Data Definition Language

SQL se puede usar para crear y mantener bases de datos y sus tablas

Nombre de la base de datos

CREATE DATABASE _____

Create database BD_Construcciones

BD_construcciones

DDL - Data Definition Language

Se puede usar para eliminar bases de datos completas y sus tablas

Nombre de la base de datos

DROP DATABASE _____

drop database BD_Construcciones

BD_Construcciones



DDL - Data Definition Language

Crear tablas

Nombre de la tabla

CREATE TABLE _____

```
create table tbl_casas  
(  
  Columna1 TipoDeDato,  
  Columna2 TipoDeDato  
)
```

```
Create table tbl_Casas  
(  
  IdCasa int,  
  Descripcion varchar(200)  
  Estatus bit  
)
```

DDL - Data Definition Language

Data types	Description	Others Data types	
Bit	True/False 0/1,	BigInt	Geometry
Decimal(18,0)	No + Decimales,	Binary	Hierarchyid
Float	Valor aproximado,	Bit	Money
Datetime	Fecha,	Char	Nchar(10)
Int	Entero,	Date	Numeric(18,0)
Timestamp	Fecha Numerica,	Money	Nvarchar(50)
Varchar(Max)	Char exacto,	Datetime2(7)	Real
UniqueIdentifier	NewID,	DatetimeOffset	Smalldatetime
Xml	Xml,	Image	SmallInt
		Geography	text

DDL - Data Definition Language

Crear tablas

```
CREATE TABLE IOT_DEVICES (  
  id  INTEGER PRIMARY KEY,  
  name TEXT  
);
```

Nombre de la tabla

```
CREATE TABLE Caracteristicas (  
  caid  INTEGER,  
  Descripcion Varchar(200),  
  IOT_DEVICE INTEGER,  
  FOREIGN KEY(IOT_DEVICE) REFERENCES  
  IOT_DEVICES(id)  
);
```

Campos

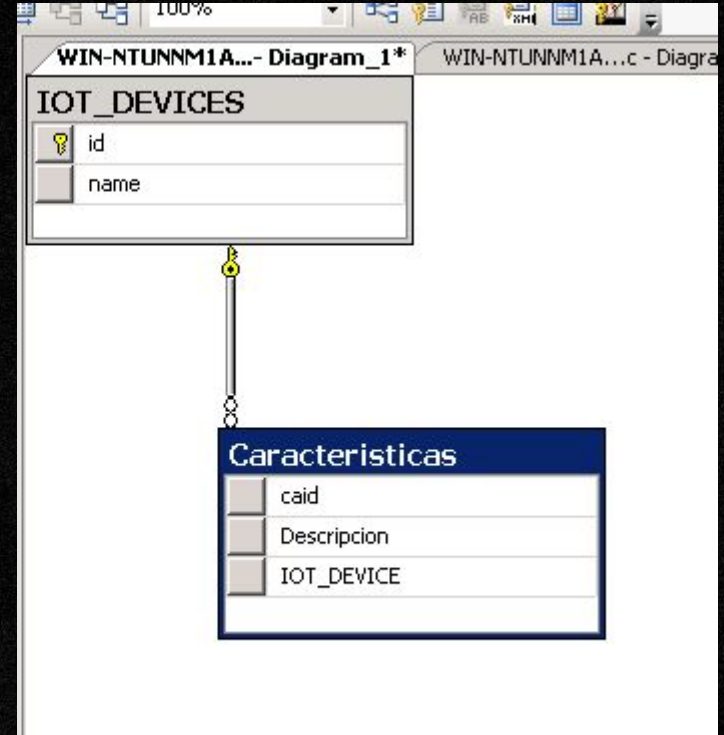
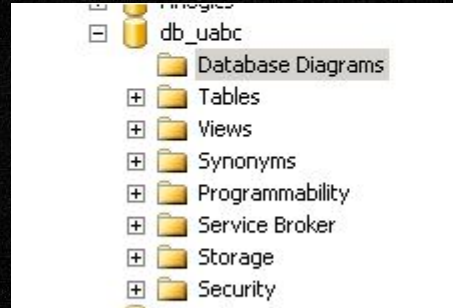
Constrains

DDL - Data Definition Language

Diagramas SQL

En la base de datos:

- 1- Seleccione Database Diagrams
- 2- Seleccione las tablas



DDL - Data Definition Language

Permite eliminar tablas

Nombre de la tabla

DROP TABLE _____

drop table tbl_casas

* Ejercicio Borrar todas las tablas del Sistema
Meetup



Tbl_Casas

A diagram showing a white rectangular box with the text 'Tbl_Casas' inside. A thick red diagonal line crosses the box from the bottom-left to the top-right, indicating that the table is being deleted or is no longer valid.

DDL - Data Definition Language

Agregar y eliminar columnas a tablas

ALTER TABLE _____

Nombre de la tabla



```
alter table tbl_casas  
ADD Arquitecto INT
```

```
alter table tbl_casas  
Alter COLUMN Arquitecto INT NOT NULL
```

```
alter table tbl_casas  
drop COLUMN Arquitecto INT
```


DML - Data Manipulation Language

Permite borrar registros

Nombre columnas a consultar

SELECT _____ FROM _____ WHERE (_____)

Nombre de la tabla

Condición

DML - Data Manipulation Language

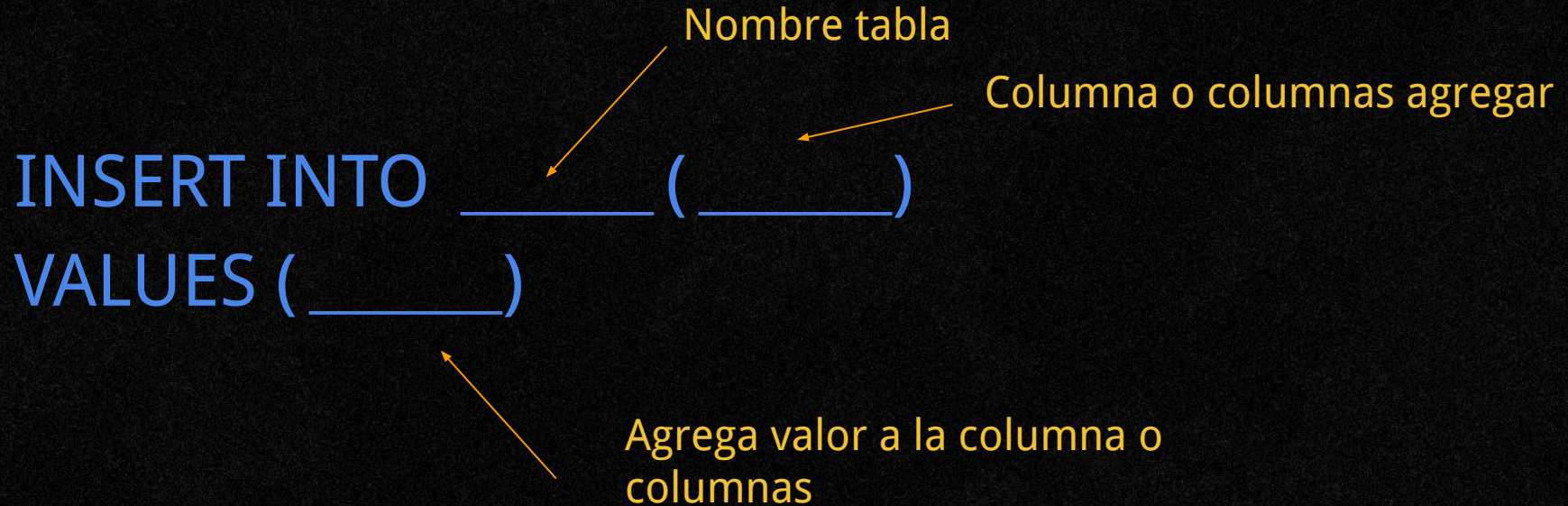
Permite Insertar registros

INSERT INTO _____ (_____)
VALUES (_____)

Nombre tabla

Columna o columnas agregar

Agrega valor a la columna o columnas

The diagram illustrates the syntax of the INSERT INTO statement. It shows the keywords 'INSERT INTO' followed by a blank line for the table name, then an opening parenthesis followed by a blank line for the column names, and finally a closing parenthesis. Below this, the keyword 'VALUES' is followed by an opening parenthesis and a blank line for the values. Three yellow arrows point from text labels to the corresponding parts of the syntax: 'Nombre tabla' points to the blank line after 'INSERT INTO'; 'Columna o columnas agregar' points to the blank line inside the parentheses after 'INSERT INTO'; and 'Agrega valor a la columna o columnas' points to the blank line inside the parentheses after 'VALUES'.

DML - Data Manipulation Language

Permite Actualizar registros

UPDATE _____ SET _____ = _____
WHERE (_____)

Nombre tabla

Columna a modificar

Valor asignar

Condición para actualizar

The diagram illustrates the SQL UPDATE syntax. It shows the keywords 'UPDATE' and 'WHERE' in blue, followed by four blue underscores representing placeholders. The keyword 'SET' is also in blue, followed by a blue equals sign and another blue underscore. Yellow arrows point from text labels to these placeholders: 'Nombre tabla' points to the first underscore after 'UPDATE'; 'Columna a modificar' points to the underscore after 'SET'; 'Valor asignar' points to the underscore after the equals sign; and 'Condición para actualizar' points to the underscore inside the parentheses after 'WHERE'.

DML - Data Manipulation Language

Permite borrar registros

Nombre tabla

DELETE _____ WHERE (_____)

Condición

DDL - Data Definition Language

Permite crear procedimientos almacenados

CREATE PROCEDURE Nombre procedimiento


Ejemplo

```
Create procedure dbo.insertarCasas
```

```
@param1 TipoDeDato,
```

```
@param2 TipoDeDato
```

```
AS
```

```
BEGIN
```

```
Insert into casas (columna1, columna2) values (@param1, @param2)
```

```
End
```

DDL - Data Definition Language

Procedimientos almacenado

Ejemplo

```
Create procedure dbo.insertarCasas
```

```
@direccion varchar(50),
```

```
@Color varchar(20)
```

```
AS
```

```
BEGIN
```

```
Insert into casas (columna1, columna2) values (@direccion, @vvarchar)
```

```
End
```


Ejecutar Procedimiento Almacenado

Para ejecutar Procedimiento desde cualquier script

```
Exec InsertarCasas @param1, @param2
```

DDL - Data Definition Language

Permite crear procedimientos almacenados

Nombre vista

CREATE VIEW _____ AS



Ejemplo

Create view ConsultaDatosEmail as

Select tbl1.Usuario, tbl2.Mensaje, tbl2.FechaMensaje from Usuarios as tbl1

Join Mensajes as tbl2 on tbl2.idUsuario = tbl1.idUsuario

Select * from ConsultaDatosEmail

//Aplicar diferentes Where a la vista

Consultar Vista

Importante: Ver cómo se realiza la consulta sin crear nuevamente la unión y ver que no se aplicó WHERE al crear la vista.

Ejemplo

Se solicita de la vista Creada, consultar por Usuario, otra consulta por FechaMensaje o por Mensaje

```
Select * from ConsultaDatosEmail where Usuario = 'UsuarioAconsultar'
```

```
Select * from ConsultaDatosEmail where FechaMensaje = '2020.02.20'
```

```
Select * from ConsultaDatosEmail where Mensaje like '%Checando%'
```

Triggers

Se disparan después de un evento de INSERT, DELETE, UPDATE sobre las tablas asociadas y se guarda la acción en una tabla en este caso Historial.

Create table tablaPrincipal

```
( ID INT IDENTITY (1,1) PRIMARY KEY,  
  Emp_First_name VARCHAR (50)  
)
```

Create table Historial

```
( ID INT IDENTITY (1,1) PRIMARY KEY,  
  Emp_First_name VARCHAR (50),  
)
```


Puntos a considerar con triggers

1.- Los triggers no se pueden ver de modo visual en SQL, por lo que debe ejecutar la siguiente instrucción para Consultar: `select * from sys.triggers` y las instrucciones de los siguientes slides para Crear/ modificar o eliminar

2- Después de la Instrucción que vas a visualizar `AS` va la instrucción de insertar tabla. **Los triggers funcionan para auditar la manipulación de la estructura de la base de datos como de la información.**

Ejemplo:

```
Create Trigger EjemplodeTrigger AS Insert into Historial(id, descripcion)  
Values(valor1, 'valor2')
```

Triggers en Manipulación de Datos (DML)

Se disparan después de un evento de INSERT, DELETE, UPDATE sobre las tablas asociadas (Es decir manipular registros)

Nombre del TRIGGER Tipo operación Insert, Update, Delete

CREATE TRIGGER _____ ON _____ AFTER _____ AS

Tabla a la que se hará una operación

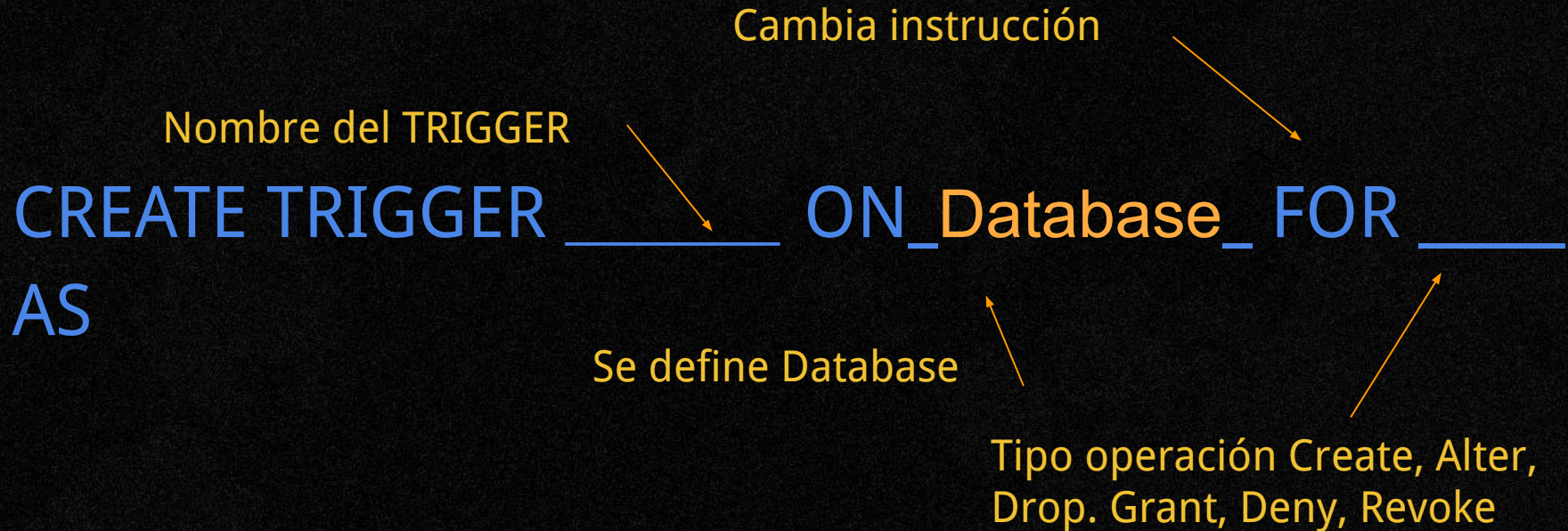
```
Create trigger DespuesDeInsertTrigger ON tablaPrincipal after INSERT as  
Insert into Historial Values ('Insert', (select top 1 ID from tablaPrincipal order by ID desc) )
```

Consulta --> select * from sys.triggers

select* from sys.all_sql_modules WHERE definition LIKE '%Trigge%'

Triggers on Database (DDL)

Se disparan después de un evento de crear/ modificar/ eliminar un objeto de la estructura de base de datos o de la definición de datos. (ejemplo tablas, etc)



Ejemplos

DDL

```
create trigger CrearTabla on database for create_table as begin print 'Creacion Tabla' end
```

```
create trigger BorrarTabla on database for drop_table as begin print 'Borrar Tabla' end
```

```
create trigger Crearprocedimiento on database for create_procedure as begin print 'se creo  
procedimiento' end
```

DML

```
CREATE TRIGGER InsertRegistro ON Probando AFTER Insert AS  
insert into probando2 (idp,descripcion) values(1,'inserto')
```

```
CREATE TRIGGER BorrarRegistro ON Probando AFTER Delete AS  
insert into probando2 (idp,descripcion) values(2,'Borrar')
```


DCL - Data Control Language

Permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.

- Grant
- Revoke

Grant select on TABLA to USUARIO
Revoke select on deep from public

Comandos para respaldo y recuperación

Permiten respaldar la base de datos

```
BACKUP DATABASE MiBaseDatos  
TO DISK = 'C:\visual\test\miBD.BAK'  
GO
```

```
RESTORE DATABASE MiBaseDatos  
FROM DISK = 'C:\visual\test\miBD.BAK' ;
```