Trabajo 2 de Base De Datos 1

Gustavo Ángel Ortega Pérez

Hennys Sebastian Tous De Avila

Universidad de Cartagena

**Texto, Carta

Descripción generada automáticamente**

Programa: Ingeniería de Software

Asignatura: Estructura de Datos

Tutor: Jhon Arrieta Arrieta

Cartagena de Indias – agosto 2 del año 2022

# Tabla de Contenido

1. Introducción.
2. Objetivo.
3. Justificación.
4. Resultados
5. Síntesis y argumentos individuales.
6. Bibliografía

# Introducción

En el presente documento vamos a explorar el manejo adecuado de un Motor de Bases de datos por medio del CMD, haciendo uso de sentencias DDL para crear el esquema de nuestra base de datos a partir del modelo relacional creado en la actividad anterior.

# Objetivos

* Aprender a manejar un Motor de Base de datos mediante la Caja de Comandos de Windows.
* Interpretar los modelos relacionales para convertirlos a sentencias DDL.
* Dominar sentencias DDL para la creación de esquemas de Bases de Datos.
* Aplicar restricciones de integridad a las tablas creadas mediante sentencias DDL.

# Justificación

Principalmente hemos decidido tomar riendas de este documento para ahondar nuestros conocimientos dentro del uso de motores para bases Bases de Datos, teniendo en cuenta sentencias DDL y comandos por medio de consola.

Esto con la finalidad de aprender a como manejar motores de bases de datos y como usar sentencias DDL en los mismos

# Resultados

1. **Entrar al Motor de Bases** de datos usando la **Terminal (consola)** de Línea de Comandos

sqlcmd -S HENNYS\SQLEXPRESS -E



Se hace una conexión mediante sqlcmd en la CMD de Windows o Powershell de Windows. -S especifica el nombre del servidor SQL y -E especifica que es una conexión segura

1. **Crear una base de datos** (tablespace en oracle) llamada **bd\_grupo\_xyz**, donde xyz debe reemplazarse por el número de la carpeta con los ejercicios escogidos por ustedes

CREATE DATABASE bd\_grupo\_8

go



Se coloca CREATE DATABASE y se especifica el nombre de la base de datos. Luego se coloca go para realizar la accion

1. **Crear** un nuevo **usuario** para poder entrar al Motor de BD, los datos del nuevo usuario deben ser: }

Nombre: **la\_primera\_inicial\_de\_tu\_nombre\_tu\_primer\_apellido**.

Clave: **AbcdeUdeC**

El nuevo usuario podrá conectarse desde localhost o cualquier PC.

CREATE LOGIN htous WITH PASSWORD = 'AbcdeUdeC'

go

CREATE USER htous FOR LOGIN htous

go

Texto

Descripción generada automáticamente

Hacemos un login y luego especificamos su contraseña.

Luego creamos un usuario para dicho login.

1. Al nuevo usuario creado anteriormente, se le deben **asignar Privilegios de Super Usuario** sobre la base de datos **bd\_grupo\_xyz**

Sp\_addsrvrolemember htous, sys\_admin;

ALTER ROLE [db\_owner] add member htous



Primero asignamos roles de admin del sistema al login htous por medio de sp\_addsrvrolemember, y luego asignamos permisos de db\_owner al usuario htous por medio de ALTER ROLE.

1. **Mostrar los usuarios** creados para poder usar el Motor de BD

select \* from master.sys.server\_principals where (type =’S’ or type = ‘U’)

go

Texto

Descripción generada automáticamente

Usamos master.sys.server\_principals la cual es un catalogo del sistema y nosotros podemos llamar fácilmente a los logins creados para el server, pero, especificamos que solo queremos logins con tipo S o tipo U

1. **Salir de la terminal** o consola del Motor de BD

quit



Usamos quit para salirnos de sqlcmd.

1. Volver a **entrar** la Motor de Bases de datos **desde la terminal o consola** pero ahora usando las credenciales del nuevo usuario creado anteriormente-

sqlcmd -S HENNYS\SQLEXPRESS -U htous -P AbcdeUdeC

Texto

Descripción generada automáticamente

Le decimos a sqlcmd que queremos conectarnos al server HENNYS mediante el usuario htous con contraseña AbcdeUdeC

Las siguientes operaciones SQL deben ser realizadas utilizando el nuevo usuario **bd\_grupo\_xyz**.

1. **Crear** una **nueva BD** llamada **bd\_temporal.**

CREATE DATABASE bd\_temporal



Una vez hayamos iniciado la sesión como dicho usuario, podemos crear la base de datos bd temporal

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_grupo\_xyz**

select \* from sys.databases WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb');



Interfaz de usuario gráfica, Aplicación

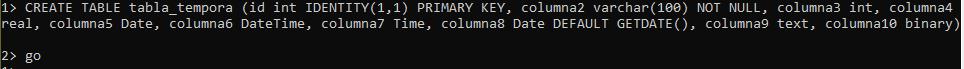
Descripción generada automáticamente

Sencillamente seleccionamos todas las bases de datos que se encuentran en sys.databases y excluimos a todas las bases de datos que sean propias del sistema para solo mostrar las creadas por el usuario

1. **Crear una tabla** llamada **tabla\_tempora** con las siguiente estructura

**tabla\_tempora**(**id** auto\_incremental y clave primera, **columna2** Texto no nula y con longitud de 100, **columna3** Entero, **columna4** Real, **columna5** Fecha, **columna6** FechaHora, **columna7** hora, **columna8** FechaHora automatica, **columna9** Texto con longitud variable, **columna10** archivo binario)

CREATE TABLE tabla\_tempora (id int IDENTITY(1,1) PRIMARY KEY, columna2 varchar(100) NOT NULL, columna3 int, columna4 real, columna5 Date, columna6 DateTime, columna7 Time, columna8 Date DEFAULT GETDATE(), columna9 text, columna10 binary)



Creamos la tabla especificando cada campo y su tipo de dato.

Cabe destacar que DEFAULT GETDATE() es para obtener la fecha actual del sistema y IDENTITY es la palabra clave para que un campo se incremente automáticamente

1. **Mostrar las tablas** de la **bd\_temporal**

Use bd\_temporal

Sp\_tables

Go

Texto

Descripción generada automáticamente

Primero cambiamos el contexto de la base de datos a bd\_temporal usando use. Luego mostramos todas las tablas (incluyendo las del sistema) con sp\_tables.

1. **Describir** la estructura de la **tabla\_tempora**

Use bd\_temporal

Sp\_help ‘dbo.tabla\_tempora’

Pantalla de computadora con fondo negro

Descripción generada automáticamente con confianza media

Primero aseguramos que estamos en la bd\_temporal.  
 Luego ejecutamos sp.help, especificamos el schema y el nombre de la tabla.

1. **Cambiar** el **nombre** de la **tabla\_tempora** por  **tabla\_tempora\_v2**

Sp\_rename ‘tabla\_tempora’, ‘tabla\_tempora\_v2’



Ejecutamos sp\_rename para asignar el nuevo nombre a la tabla tabla\_tempora

1. **Mostrar** las **tablas** de la **bd\_temporal**

Sp\_tables

go

Texto

Descripción generada automáticamente con confianza media

Mostramos todas las tablas (incluyendo las del sistema) con sp\_tables

1. En la tabla **tabla\_tempora\_v2 cambiar el nombre** de la **culumna4** por **culumnaIV**

sp\_rename 'tabla\_tempora\_v2.columna4', 'columnaIV' , 'COLUMN';



Usamos el comando sp\_rename en donde especificamos la tabla y el nombre de la columna, luego colocamos el nuevo nombre y especificamos que es tipo COLUMN

1. En la tabla **tabla\_tempora\_v2 cambiar el tipo** de dato de **columna7** por **FechaHora**

ALTER TABLE [tabla\_tempora\_v2] ALTER COLUMN [columna7] [DATETIME]



Usamos ALTER TABLE y ALTER COLUMN para cambiar el tipo de datos de la columna7



1. En la tabla **tabla\_tempora\_v2 agregar una columna** nueva llamada **email** de tipo Texto, que sea única y con valor por defecto **‘ABC’**

ALTER TABLE tabla\_tempora\_v2 add email VARCHAR(30) NULL default 'ABC'

ALTER TABLE tabla\_tempora\_v2 ADD CONSTRAINT uc\_tablatempora UNIQUE (email)

Texto

Descripción generada automáticamente

Primero añadimos la columna nueva usando ALTER TABLE y especificamos su nombre, su tipo de dato, si es null o no y su valor por defecto.

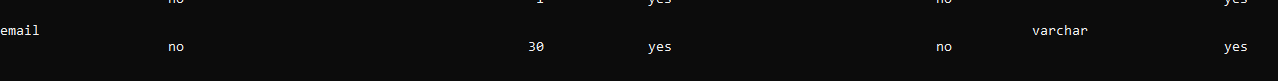
Luego usamos ALTER TABLE para añadir un constraint UNIQUE a la columna email.

1. **Describir** la Estructura de la **tabla\_tempora\_v2** para observar los cambios

Sp\_help 'dbo.tabla\_tempora\_v2';

go





Primero aseguramos que estamos en la bd\_temporal.  
Luego ejecutamos sp.help, especificamos el schema y el nombre de la tabla.

Y podemos observar que si se creó la columna email.

1. **Eliminar** la **columna6**

ALTER TABLE tabla\_tempora\_v2 DROP COLUMN columna6



Usamos ALTER TABLE tabla\_tempora\_v2, luego usamos DROP COLUMN y el nombre de la columna que queremos eliminar

1. **Describir** la Estructura de la **tabla\_tempora\_v2** para observar los cambios

Sp\_help 'dbo.tabla\_tempora\_v2';

go

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

Primero aseguramos que estamos en la bd\_temporal.  
Luego ejecutamos sp.help, especificamos el schema y el nombre de la tabla.

Y podemos observar que si se elimino la columna6.

1. **Eliminar** la **tabla\_tempora\_v2**

DROP TABLE tabla\_tempora\_v2



Usamos DROP TABLE y especificamos el nombre de la tabla a eliminar.

1. **Mostrar** las **tablas** de la **bd\_temporal**

Sp\_tables

go

Texto

Descripción generada automáticamente

Mostramos todas las tablas (incluyendo las del sistema) con sp\_tables y podemos apreciar que ya no se encuentra la tabla\_tempora\_v2

1. **Cambiarle** el **nombre** de la **base de datos** **bd\_temporal** por **bd\_temporal\_v2**

ALTER DATABASE [bd\_temporal] MODIFY NAME = [bd\_temporal\_v2]

go

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Usamos ALTER DATABASE, especificamos el nombre de la base de datos, luego le decimos que queremos modificar el nombre y por ultimo especificamos el nuevo nombre de la base de datos

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_grupo\_xyz**

select \* from sys.databases WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb');

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Sencillamente seleccionamos todas las bases de datos que se encuentran en sys.databases y excluimos a todas las bases de datos que sean propias del sistema para solo mostrar las creadas por el usuario.  
Claramente podemos ver que se ha cambiado el nombre de la bd.

1. **Entrar** la base de datos **bd\_temporal\_v2**

use bd\_temporal\_v2



Usamos use para especificar en que bd queremos entrar

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_grupo\_xyz**

select \* from sys.databases WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb');

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Sencillamente seleccionamos todas las bases de datos que se encuentran en sys.databases y excluimos a todas las bases de datos que sean propias del sistema para solo mostrar las creadas por el usuario.  
Claramente podemos ver que se ha cambiado el nombre de la bd.

Las siguientes operaciones SQL se deben realizar tomando como base **Modelo Relacional** del ejercicio seleccionado por ustedes:

1. **Entrar** o usar la base de datos **bd\_grupo\_xyz** creada anteriormente.

use bd\_grupo\_8

Texto

Descripción generada automáticamente

Usamos use para especificar en que bd queremos entrar

1. **Mostrar** las **tablas** de la **bd\_grupo\_xyz**

Sp\_tables

go

Texto

Descripción generada automáticamente

Usamos sp\_tables para mostrar todas las tablas en la bd deseada (incluyendo las del sistema).

Podemos observar que solo hay tablas del sistema.

1. **Crear** en la **base de datos** **bd\_grupo\_xyz** cada una de sus tablas de acuerdo a lo reflejado en el Modelo Relacional del ejercicio que han elegido.

CREATE TABLE Alumno

(

Matricula VARCHAR(10) NOT NULL,

Nombre VARCHAR(30),

Grupo INT,

PRIMARY KEY (Matricula)

);

CREATE TABLE Examen

(

Fecha DATE,

Npreguntas INT,

N INT NOT NULL,

PRIMARY KEY (N)

);

CREATE TABLE Practica

(

Dificultad VARCHAR(10),

Titulo VARCHAR(10),

P INT NOT NULL,

PRIMARY KEY (P)

);

CREATE TABLE Profesor

(

CC INT NOT NULL,

Nombre VARCHAR(30),

PRIMARY KEY (CC)

);

CREATE TABLE AlumnoExamen

(

Nota FLOAT,

Matricula VARCHAR(10) NOT NULL,

N INT NOT NULL,

PRIMARY KEY (Matricula, N),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (N) REFERENCES Examen(N) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE AlumnoPractica

(

Nota FLOAT,

Fecha DATE,

Matricula VARCHAR(10) NOT NULL,

P INT NOT NULL,

PRIMARY KEY (Matricula, P),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE ProfesorPractica

(

Fecha DATE,

P INT NOT NULL,

CC INT NOT NULL,

PRIMARY KEY (P, CC),

FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE,

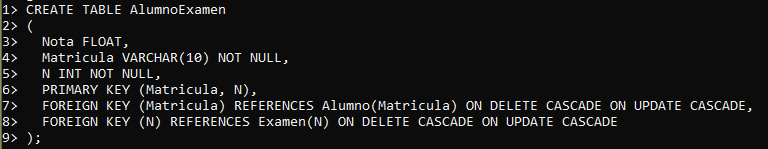
FOREIGN KEY (CC) REFERENCES Profesor(CC) ON DELETE CASCADE ON UPDATE CASCADE

); 









Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. **Describir** cada una de las **tablas** creadas en **bd\_grupo\_xyz**

Sp\_help 'dbo.Alumno'

Sp\_help 'dbo.AlumnoExamen'

Sp\_help 'dbo.AlumnoPractica'

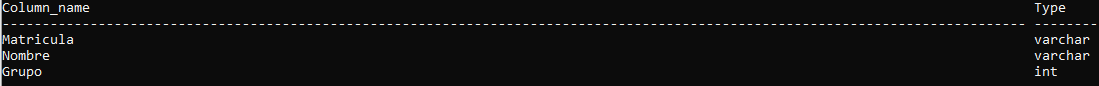
Sp\_help 'dbo.Examen'

Sp\_help 'dbo.Practica'

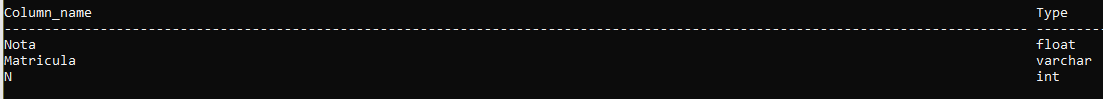
Sp\_help 'dbo.Profesor'

Sp\_help 'dbo.ProfesorPractica'

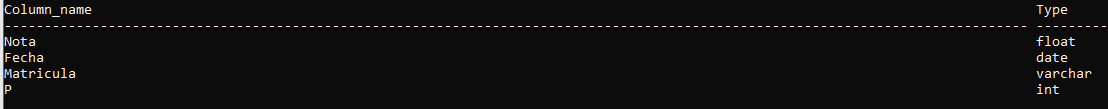
Alumno



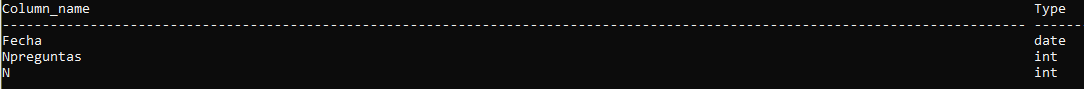
AlumnoExamen



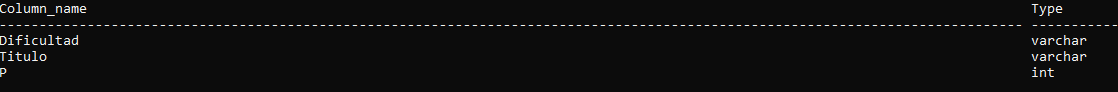
AlumnoPractica



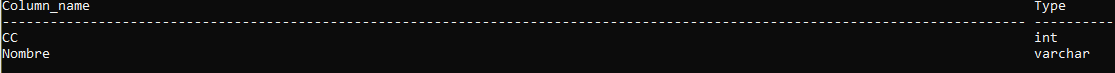
Examen



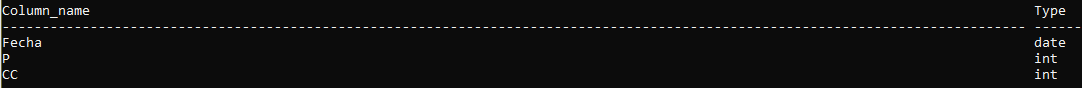
Practica



Profesor



ProfesorPractica



1. Si las tablas aún no tienen las columnas que deben actuar como llaves foráneas, entonces deben agregar dichas columnas a en respectivas tablas.

CREATE TABLE AlumnoExamen

(

Nota FLOAT,

Matricula VARCHAR(10) NOT NULL,

N INT NOT NULL,

PRIMARY KEY (Matricula, N),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (N) REFERENCES Examen(N) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE AlumnoPractica

(

Nota FLOAT,

Fecha DATE,

Matricula VARCHAR(10) NOT NULL,

P INT NOT NULL,

PRIMARY KEY (Matricula, P),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE ProfesorPractica

(

Fecha DATE,

P INT NOT NULL,

CC INT NOT NULL,

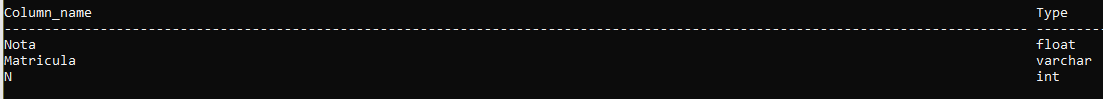
PRIMARY KEY (P, CC),

FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE,

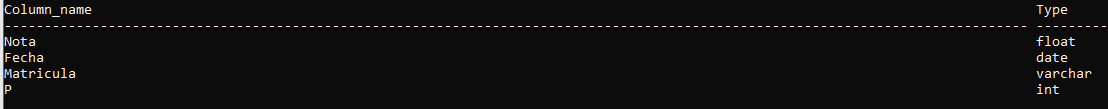
FOREIGN KEY (CC) REFERENCES Profesor(CC) ON DELETE CASCADE ON UPDATE CASCADE

);

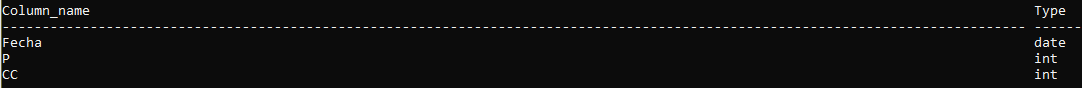
AlumnoExamen



AlumnoPractica



ProfesorPractica



En el punto anterior ya habíamos creado las columnas de las llaves foráneas que irán en las tablas intermedias.

1. **Agregar** los **índices** en las columnas que actuarán como llaves foráneas

CREATE TABLE AlumnoExamen

(

Nota FLOAT,

Matricula VARCHAR(10) NOT NULL,

N INT NOT NULL,

PRIMARY KEY (Matricula, N),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (N) REFERENCES Examen(N) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE AlumnoPractica

(

Nota FLOAT,

Fecha DATE,

Matricula VARCHAR(10) NOT NULL,

P INT NOT NULL,

PRIMARY KEY (Matricula, P),

FOREIGN KEY (Matricula) REFERENCES Alumno(Matricula) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE ProfesorPractica

(

Fecha DATE,

P INT NOT NULL,

CC INT NOT NULL,

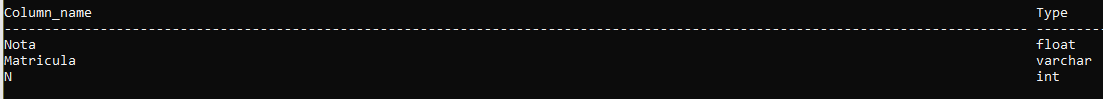
PRIMARY KEY (P, CC),

FOREIGN KEY (P) REFERENCES Practica(P) ON DELETE CASCADE ON UPDATE CASCADE,

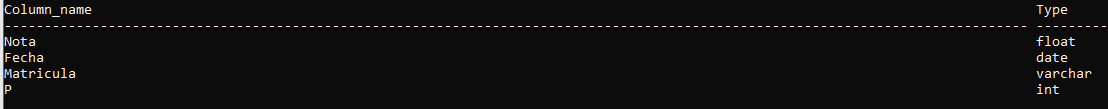
FOREIGN KEY (CC) REFERENCES Profesor(CC) ON DELETE CASCADE ON UPDATE CASCADE

);

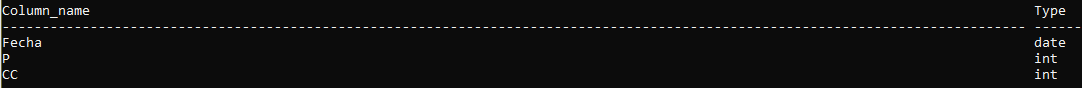
AlumnoExamen



AlumnoPractica



ProfesorPractica



1. **Agregar** las respectivas **llaves foraneas** en cada una de las tablas a las que haya lugar según el **Modelo Relacional**, cada llave foránea debe aplicar restricción de integridad referencial Restrictiva para operaciones de Eliminar y Cascada para Operaciones de Actualizar.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Debido a que SQL Server no posee la restricción RESTRICT, he tenido que colocar CASCADE en las operaciones de ON DELETE y ON UPDATE

1. Utilizar alguna herramienta (ideal que sea la herramienta por defecto de cada motor de bd) para visualizar el Modelo Relacional a partir de la BD creada.

Diagrama

Descripción generada automáticamente

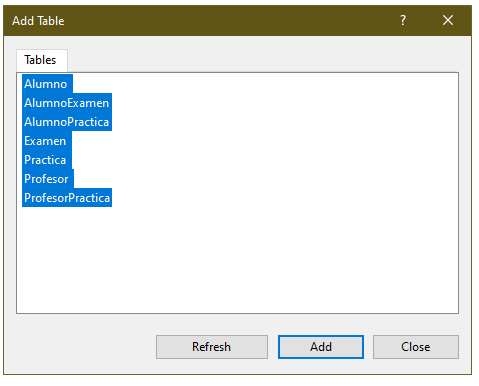
Para crear este diagrama debemos seguir los siguientes pasos:

* Vamos a SQL Server Management Studio
* Nos dirigimos al explorador de Objetos
* Nos dirigimos al lugar donde están las bases de datos
* Buscamos la base de datos deseada y le damos al símbolo +
* En las carpetas que salen una vez se expande la bd deseada, debe salir una carpeta llamada Database Diagrams, le damos Click derecho y seleccionamos New Database Diagram

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

* Luego nos saldrá una ventana en donde debemos añadir las tablas que queremos en el diagrama, shift para seleccionar todas y le damos en Add



1. Crear un **Backup** **de la estructura** de la **bd\_grupo\_xyz** y guardar el resultado en un archivo llamado **backup\_bd\_grupo\_xyz.sql**

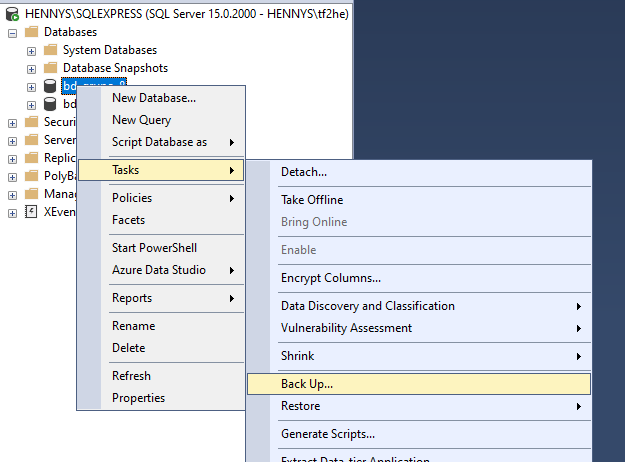
Para generar un backup de nuestra base de datos debemos hacer lo siguiente:

* Buscamos la base de datos que queremos hacer backup

Texto, Aplicación

Descripción generada automáticamente con confianza media

* Le damos click derecho y nos vamos a Tasks, luego damos click en donde dice Back Up…



* Luego nos saldrá esta ventana en donde seleccionamos la bd que queremos hacer backup y si queremos añadir una ubicación personalizada le damos al botón Add

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

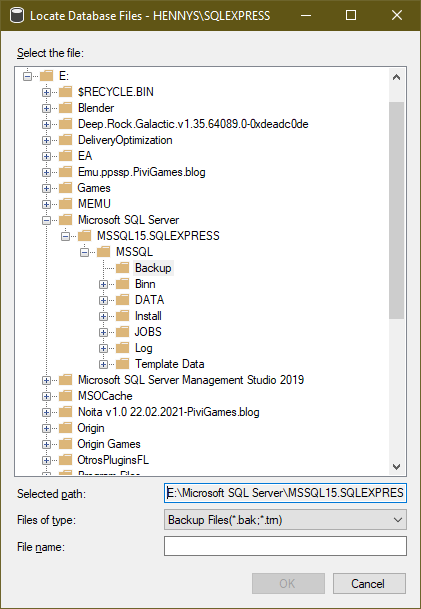
Descripción generada automáticamente

* Luego nos preguntara si queremos guardar el backup en un archivo o como url. Seleccionamos archivo y le damos al botón …

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Una vez hayamos dado click al botón … podremos navegar por nuestro ordenador y elegir donde queremos guardar el archivo. En la parte de debajo de la ventana podemos asignar el nombre que queremos al archivo (en el campo File name)



* Le damos Ok y nos regresa a la ventana inicial. Solamente le debemos dar OK y empezara a hacer el backup de la base de datos. Finalmente nos saldrá esta ventana:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

1. Eliminar la Base de datos **bd\_grupo\_xyz**

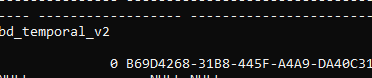
DROP DATABASE bd\_grupo\_8



Usamos la instrucción DROP DATABASE y luego escribimos el nombre de la base de datos a borrar

1. Mostrar las bases de datos del usuario **bd\_grupo\_xyz**

select \* from sys.databases WHERE name NOT IN ('master', 'tempdb', 'model', 'msdb');



Sencillamente seleccionamos todas las bases de datos que se encuentran en sys.databases y excluimos a todas las bases de datos que sean propias del sistema para solo mostrar las creadas por el usuario.  
Claramente podemos ver que se ha borrado la base de datos y solo queda la bd\_temporal\_v2

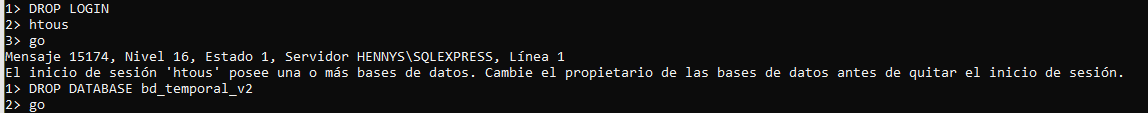
1. Eliminar el Usuario **bd\_grupo\_xyz**

sqlcmd -S HENNYS\SQLEXPRESS -E

DROP DATABASE bd\_temporal\_v2

DROP LOGIN htous







Inicialmente debemos acceder al motor de BD como el login predeterminado del sistema. Luego eliminamos la base de datos bd\_temporal\_v2 ya que no nos permite borrar el login cuando esta existe.  
Finalmente usamos DROP LOGIN y luego especificamos el nombre del login que queremos eliminar.

# Síntesis

Podemos identificar dos principales conceptos dentro de este documento.

## Motor de Base de Datos

Un motor de base de datos es una herramienta interna que habilita o facilita ciertas operaciones sobre las tablas y sus datos.

Pueden realizar toda la lógica que se puede aplicar a las tablas y sus datos, manteniendo su integridad. Ojo, no podemos confundir Motor de Base de Datos con Gestor de base de datos; Este último es un sistema de gestión de bases de datos es un software que tiene características y funciones propias que facilitan la conexión con el motor de la base de datos.

## Sentencias DDL

Las sentencias DDL (Data Definition Language) son sentencias que nos permiten definir, alterar, modificar objetos dentro de mi base de datos, estas sentencias son:

* CREATE: Esta sentencia permite crear objetos dentro de la base de datos.
* ALTER: Esta sentencia permite modificar objetos de la base de datos.
* DROP: Esta sentencia permite eliminar objetos de la base de datos.

# Conclusión

En el presente documento, pudimos comprender como funcionan los motores de bases de datos y como estos hacen uso de las sentencias DDL para la creación de esquemas de Bases de Datos.

# Referencias

*Diferencia entre gestor de base de datos y motor de base de datos*. (s. f.). EDTeam. https://ed.team/comunidad/cual-es-la-diferencia-entre-sistema-gestor-de-base-de-datos-y-motor-de-base-de-datos

Grijalva, I. (2020, 30 octubre). *▷ SQL - Las Sentencias DDL y DML*. Sentencias DDL y DML. https://codigosql.com/blog/sentencias-ddl-y-dml/

W3 Schools. (s. f.). *SQL tutorial*. <https://www.w3schools.com:443/sql/>

M. (2022, 11 enero). *Transact-SQL Reference (Database Engine) - SQL Server*. Microsoft Docs. https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver16