FUNDAMENTOS DE BASE DE DATOS.

Tema Nº7:Identificación del lenguaje de definición de datos DDL.

Indicador de logro Nº7:

Crea base de datos optimizada, tablas, esquemas e índices, usando lenguaje de Definición de Datos (DDL), garantizando disponibilidad y rendimiento en SQL Server".

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº7:**

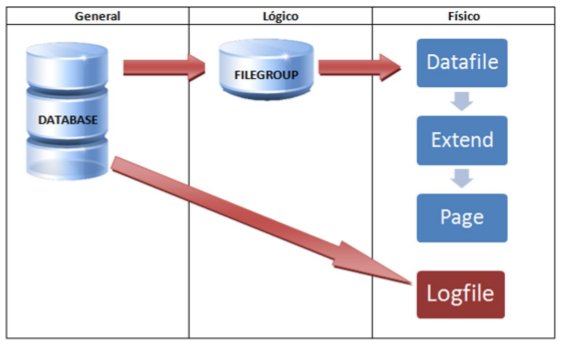
Identificación del lenguaje de definición de datos DDL.

**Subtema 7.1:**

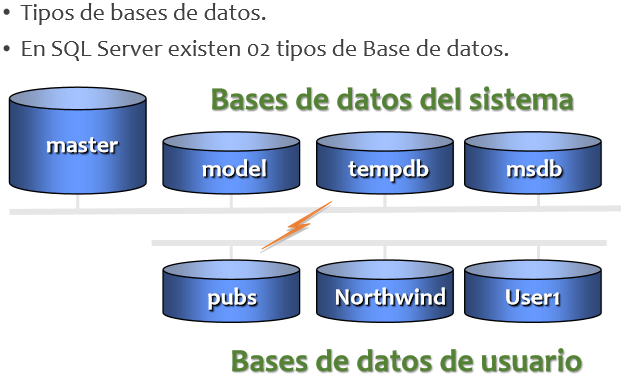
Características CRUD: CREATE, ALTER, DROP y diccionario de datos.

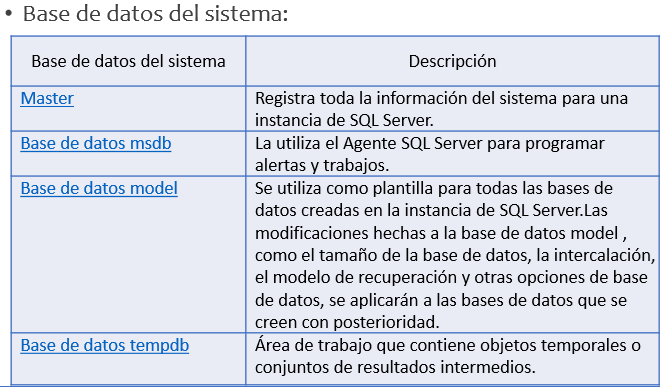
**INTRODUCCIÓN A SQL SERVER.**

SQL Server 2016 es un sistema gestor de bases de datos relacionales de Microsoft Corporation orientado a sistemas medianos y grandes, aunque también puede rodar en ordenadores personales. SQL Server Management Studio (SSMS) es la herramienta de SQL Server que permite definir y gestionar todas las bases de datos almacenadas en el servidor SQL Server 2016.

**ESTRUCTURA**:

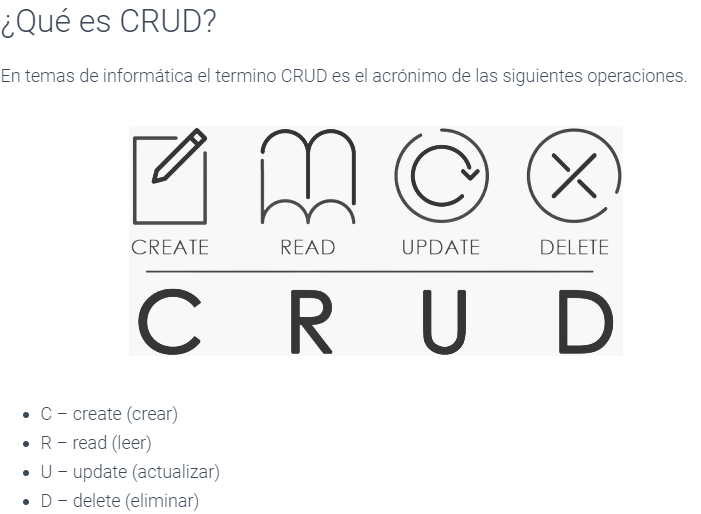
**TIPOS DE BASES DE DATOS EN SQL SERVER.**



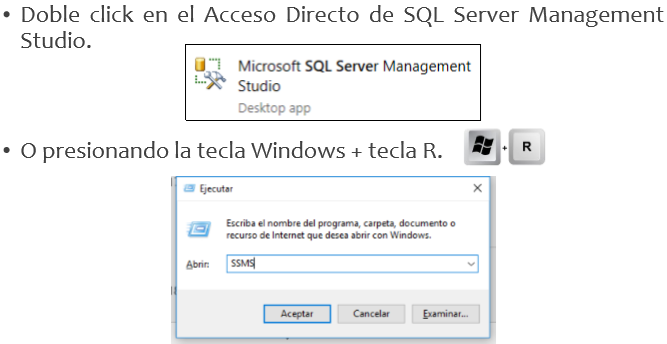


**CARACTERÍSTICAS CRUD.**

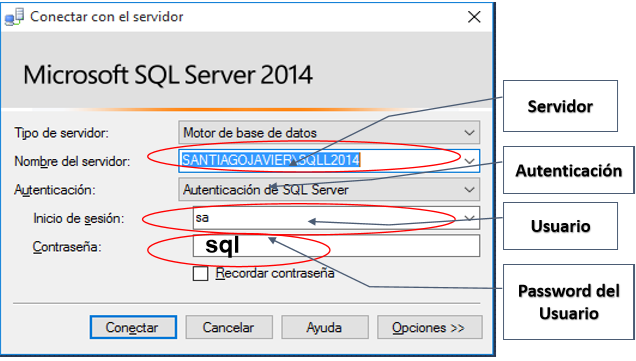
Estas operaciones hacen referencia a la interacción de la aplicación con la base de datos o la capa de almacenamiento de información.



**COMO CONECTARNOS CON UN SERVIDOR DE BASE DE DATOS.**



**AUTENTICANDO LA CONEXIÓN:**



**IMPORTANCIA:**

De esta manera podemos entender la importancia de lo que significa este término, prácticamente la base de una aplicación.

**CREATE**

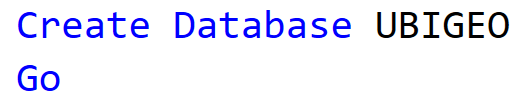
Creación de una Base de Datos SQL.

**Existen 2 Formas:**

1. **Definición por Defecto:** Con valores de propiedades predeterminadas (No recomendado).
2. **Definición Personalizada:** Con valores de propiedades definidas por el DBA según el plan, tales como nombre de los archivos, ubicación, tamaño inicial, tamaño máximo y factor de crecimiento

**CREANDO UNA BASE DE DATOS EN SQL SERVER 2016**

1. **Definición por defecto**



1. **Definición Personalizadas.**



Cuando se crea una base de datos en SQL SERVER, se crean archivos los llamados MDF y LDF, con características tales como:

**NAME:** Nombre identificativo y único de la base de datos, si se desea realizar alguna modificación sobre este archivo, se deberá usar este nombre.

**FILENAME:** representa la ruta y el nombre del archivo a crear, si es archivo físico MDF o archivo lógico LDF.

**SIZE:** representa el tamaño inicial de la base de datos MB ya sea físico o lógico, este será reservado por el sistema apenas sea creado.

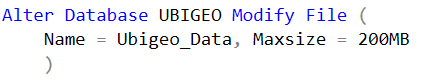
**MAXSIZE**: indica el tamaño máximo del archivo en MB, ya sea físico o lógico.

**FILEGROWTH:** este parámetro indica si los archivos de base de datos MDF o LDF son crecientes o no, se representa en MB.

**ALTER**

Actualización de los archivos de bases de datos SQL SERVER.

En el siguiente ejemplo, actualice los valores del archivo **Ubigeo\_Data**, de la base de datos **UBIGEO**, incrementando el valor de la propiedad **MAXSIZE** a 200 MB.



**Recuerde:** que al darle un nuevo valor a los parámetros **Size** y **MaxSize** no podrá reconfigurar para reducir tamaños, en cambio el parámetro **Filegrowth** puede reconfigurar para aumentar o reducir.

**Comprobar las propiedades de una base de datos:**



**DROP**

Eliminar una base de datos SQL SERVER.

En el siguiente ejemplo, se elimina la base de datos UBIGEO.

**DROP DATABASE** Ubigeo

**Cuidado:** Cuando elimine a una base de datos SQL SERVER, antes esta deberá dejar de ser usada, en caso contrario provoca un error.

**Subtema 1.2:**

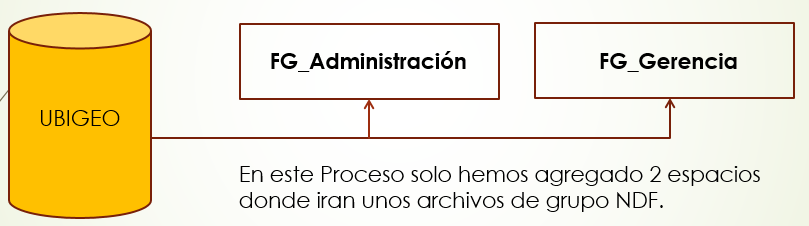
Creación de **Filegroup**, agregar y actualizar archivos secundarios a una base de datos.

**AGREGAR ARCHIVOS A UNA BASE DE DATOS**

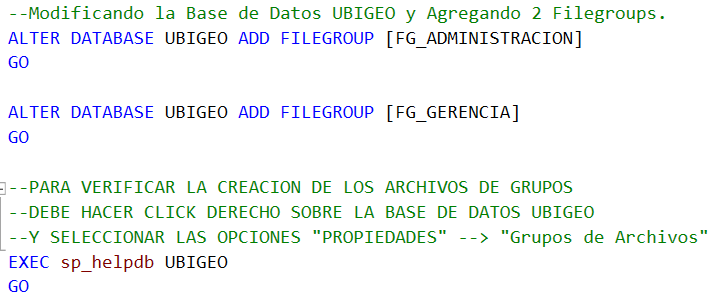
El proceso consiste, en dividir a una base de datos en grupos de archivos denominados **File Group**. Mejorando la concurrencia de los usuarios.

**Por Ejemplo:**

En la base de datos **UBIGEO** agregar 2 grupos de archivos llamados: **FG\_ADMINISTRACION** y **FG\_GERENCIA**.



**COMANDOS A EJECUTAR EN EL PROCESO DE CREACION DE LOS FILEGROUP:**



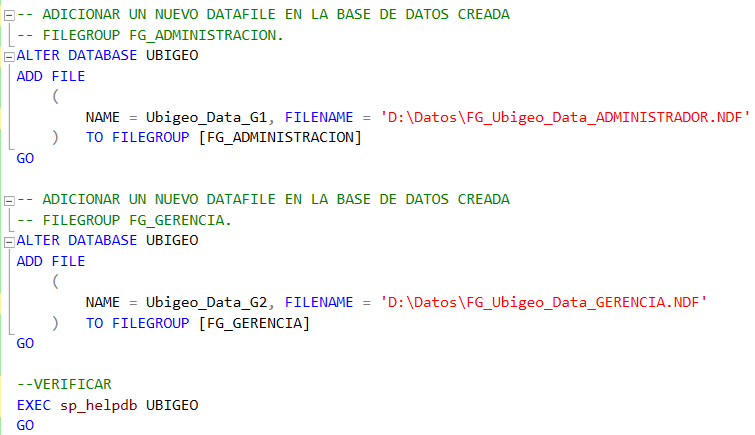
**NOTA:** hasta el momento solo se han creado los espacios necesarios para los archivos de grupo. [**FG\_ADMINISTRACION**] y [**FG\_GERENCIA**]

**Ahora a cada espacio se le asignará un archivo de grupo que corresponda.**



Como observa cada espacio tiene un pequeño cilindro pequeño, el cual representa a los archivos de grupo NDF al mismo tiempo llamados **DATAFILE**.

**CREACION DE DATAFILE.**



* **Cada archivo de grupo debe llevar un nombre (NAME) identificativo propio, Ubigeo\_Data\_G1 y Ubigeo\_Data\_G2, tal como se muestra en la imagen.**
* **La ruta del archivo también debe ser único, C:\FG\_Ubigeo\_Data\_GERENCIA.NDF C:\FG\_Ubigeo\_Data\_ADMINISTRADOR.NDF, recomienda que estos 2 DATAFILE se encuentren en la misma ruta que el archivo de base de datos MDF y LDF.**

**Subtema 1.3:**

Creación de Esquemas & Tablas, manejo de Restricciones – CONSTRAINTS, manejo de Índices, diagramas de base de datos.

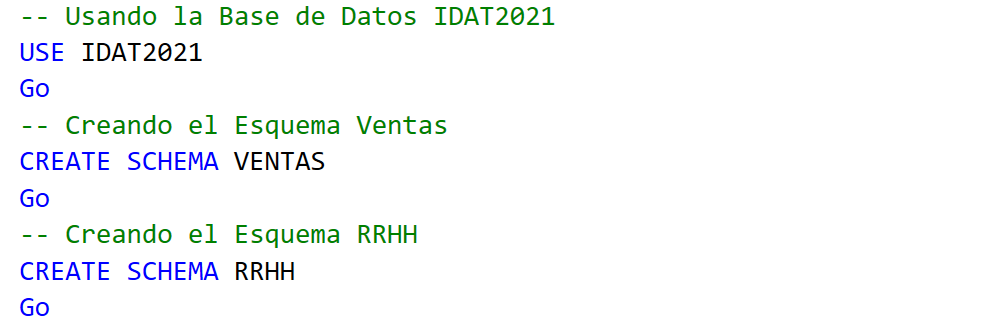
**ESQUEMAS:**

* Un esquema es un namespace que permite organizar objetos
* Contenedor de objetos. • Permite gestionar seguridad.
* El schema predeterminado es DBO.

**CREACIÓN DE SCHEMA**

* Se utiliza el comando **CREATE SCHEMA**.
* Sintaxis: **CREATE SCHEMA** <schema\_name>
* **Ejemplo:**

Crear en la Base de Datos **IDAT2021**, los schemas **VENTAS** y **RRHH**.



**VISUALIZAR LOS SCHEMAS CREADOS EN LA BASE DE DATOS**

* Se utiliza el catálogo SYS.SCHEMAS.
* Ejemplo: Ver los schemas creados.



**MODIFICAR SCHEMA:**

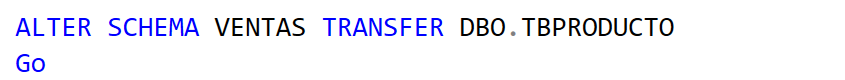
Transferir objetos de un esquema a otro

* Se utiliza el comando **ALTER SCHEMA**
* Sintaxis:

**ALTER SCHEMA** schema\_name **TRANSFER** schema\_name. Object\_name

* Ejemplo:

Transferir el objeto tabla dbo. **TBProducto** hacia el esquema **Ventas**.

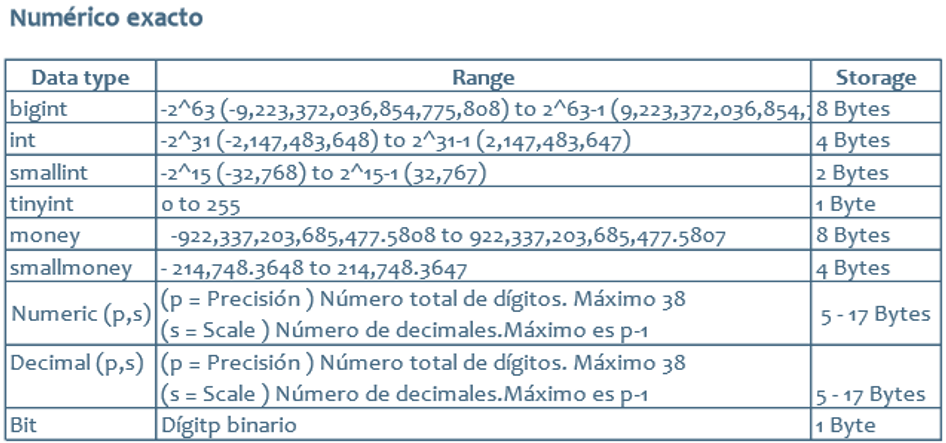


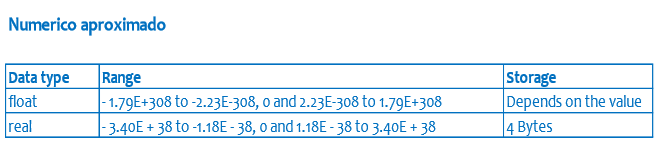
**Nota:**

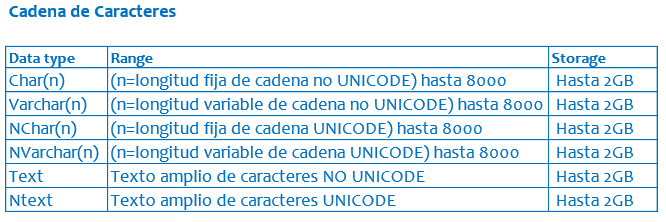
Cuando elimine un esquema, el schema a eliminar debe estar sin objetos.

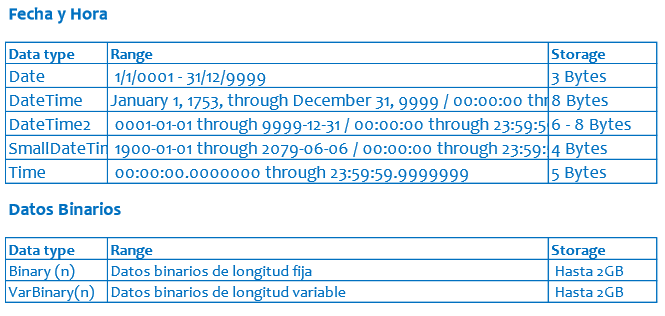
**TIPO DE DATO**

* Antes de crear una tabla, debe definir los tipos de datos para la tabla.
* Los tipos de datos especifica el tipo de información (caracteres, números o fechas) que una columna puede contener, además de cómo se almacenan los datos.
* Microsoft SQL Server proporciona varios tipos de datos del sistema. SQL Server también permite que el usuario define los tipos de datos que se basan en datos del sistema de tipos.









**CREACIÓN DE TABLAS.**

**¿Qué es una TABLA?**

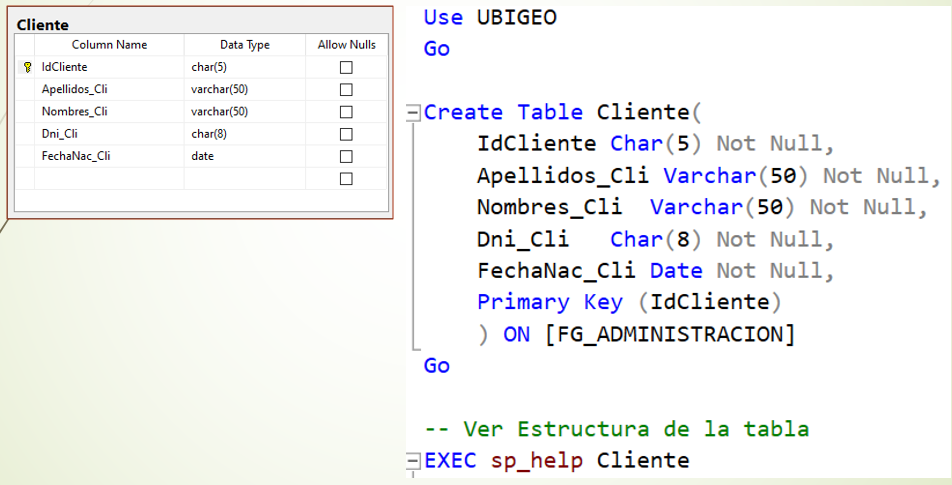
* Colección de datos sobre una entidad (Persona, Lugar, Cosa) específica, que tiene un número discreto de atributos designados (por ejemplo, cantidad o tipo).
* Las tablas están en el corazón de SQL Server y del modelo relacional en general.

**SINTAXIS:**



**Ejemplo**:

Crear la Tabla **CLIENTE** y almacenarla en el archivo de grupo **[FG\_ADMINISTRACION**].

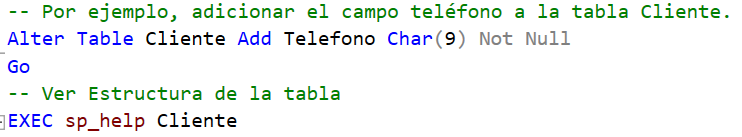


**MODIFICAR LA ESTRUCTURA DE UNA TABLA**

Comando: **ALTER TABLE**.

**PARA AGREGAR** una o varias columnas a una tabla se ejecuta el comando. Por ejemplo, adicionar el campo **teléfono** a la tabla **Cliente**.

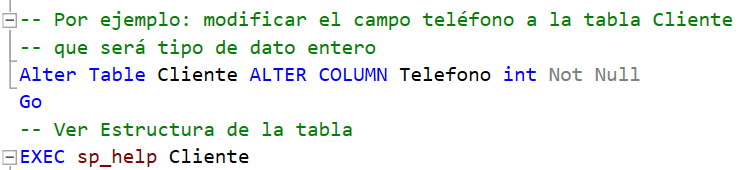
**SOLUCIÓN:**



**PARA MODIFICAR** las propiedades de un campo de una tabla se ejecuta el comando **ALTER TABLE** con la opción **ALTER COLUMN**.

Por ejemplo: modificar el campo **teléfono** a la tabla **Cliente** que será tipo de dato entero.

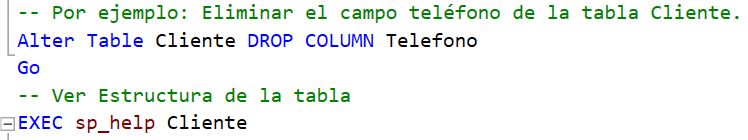
**SOLUCIÓN:**



**PARA ELIMINAR** un campo de una tabla se ejecuta el comando **ALTER TABLE** con la opción **DROP COLUMN**.

**Por ejemplo**: Eliminar el campo teléfono de la tabla Cliente.

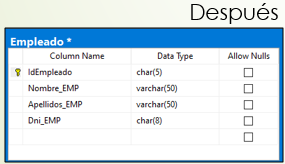
**SOLUCIÓN:**

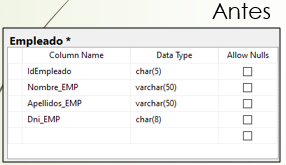


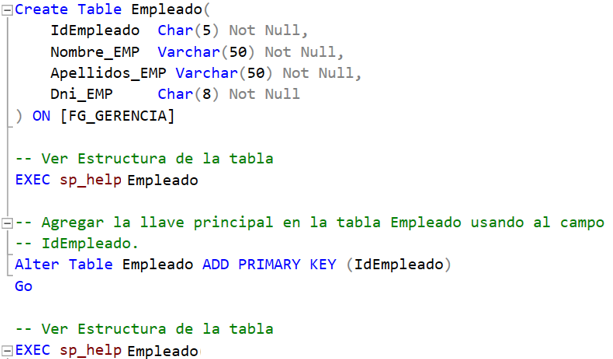
**PARA QUE UN CAMPO DE UNA TABLA SEA PRIMARY (LLAVE)** ejecuta el comando **ADD PRIMARY KEY** y entre paréntesis colocar el, o los campos que desea que sean claves.

**Por ejemplo:** Crear la tabla **Empleado** en el Archivo de Grupo **[FG\_GERENCIA]**, sin llave, luego asignar la llave.

**SOLUCIÓN:**



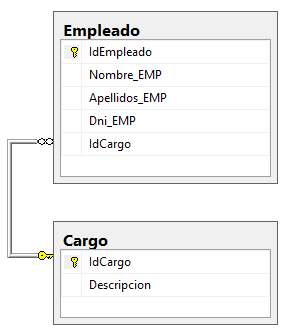


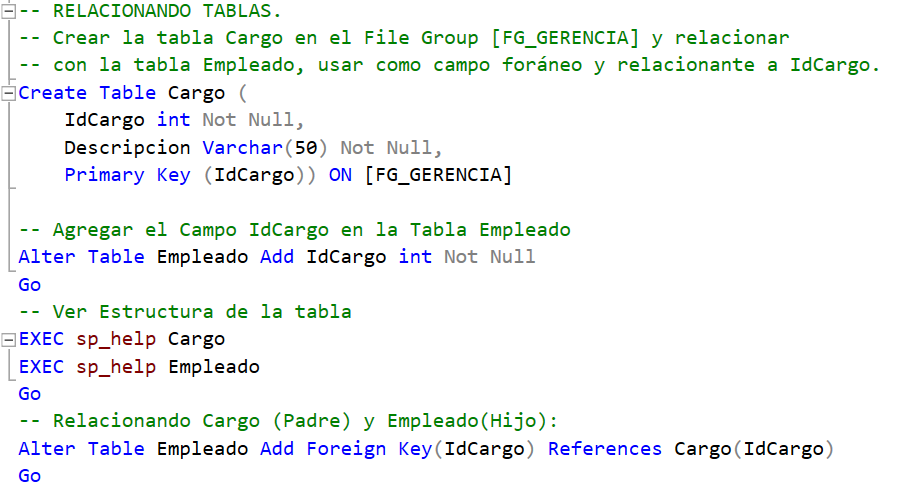


**PARA QUE SE PUEDA CREA UNA LLAVE FORÁNEA Y RELACIONAR**, se usa el comando **ADD FEREIGN KEY**

**Por ejemplo**: crear la tabla **CARGO** y relacionar con la tabla **EMPLEADO**, usar como campo foráneo a **IdCargo**.

**SOLUCIÓN:**

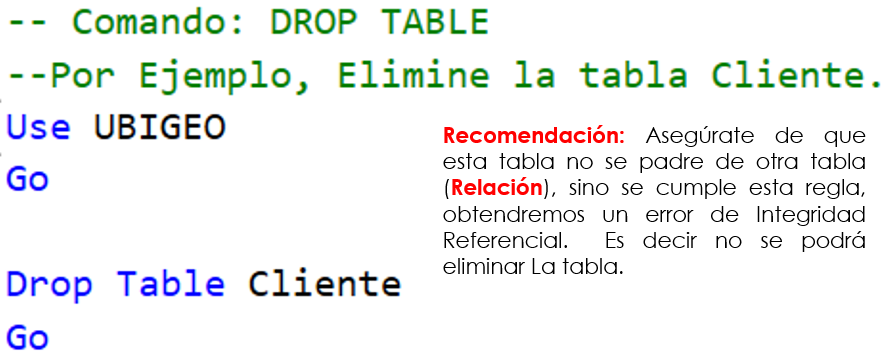




**ELIMINACIÓN DE TABLAS**

Comando **DROP TABLE**.

Por Ejemplo, Elimine la tabla Cliente.



**CONSTRUCTORES.**

**¿QUÉ ES UN CONSTRUCTOR?**

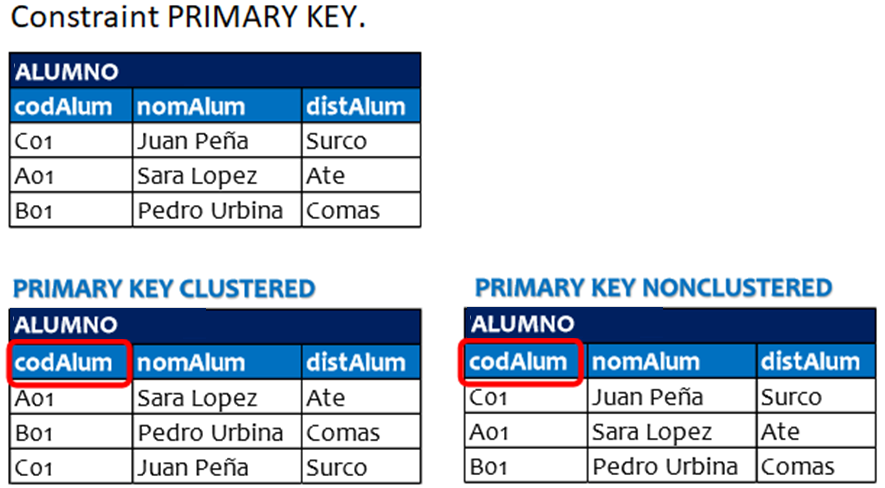
* Es una restricción a un campo de la tabla.
* Logra la integridad de datos.
* Permite implementar las reglas de negocio.
* Toda CONSTRAINT debe tener un nombre definido por el diseñador de la base de datos

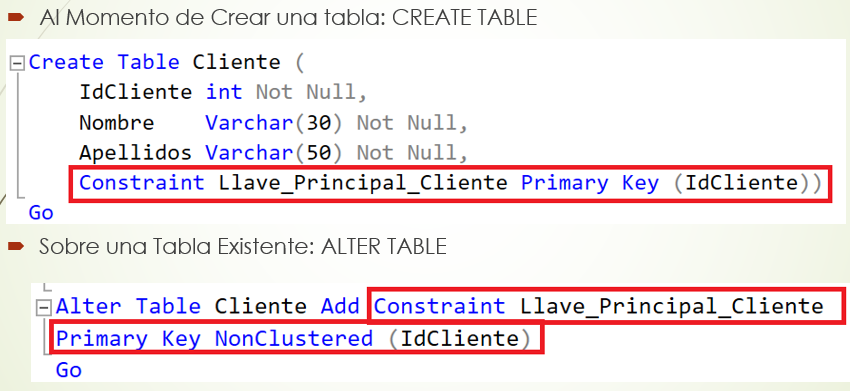
**LOS CONTRUCTORES SE CLASIFICAN EN:**

* Integridad de Entidad.
* Integridad Referencial.
* Integridad de Datos.

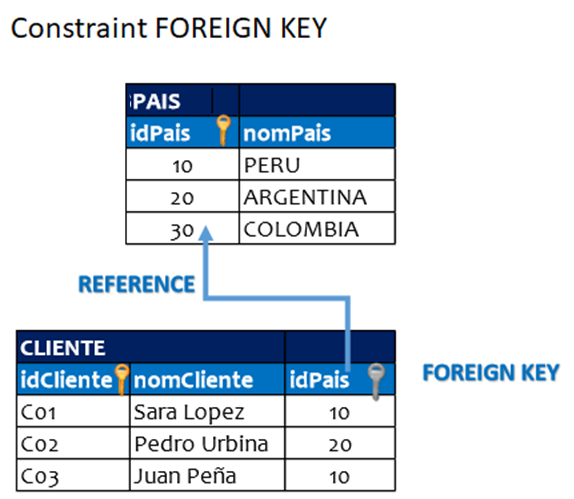
**INTEGRIDAD DE ENTIDAD.**

* Constraint **PRIMARY KEY**
* El valor puede ser el de un campo (**simple**) o la combinación de varios campos (**compuesta**)
* Los campos que identifican a PRIMARY KEY deben ser **Not Null**.
* Restricción:
  + No permite valores nulos.
  + No permite valores duplicados.
* Se define un Primary Key por TABLA
* Se puede definir para una o más COLUMNAS
* Tipo **CLUSTERED**: La tabla queda ordenada físicamente por su índice.

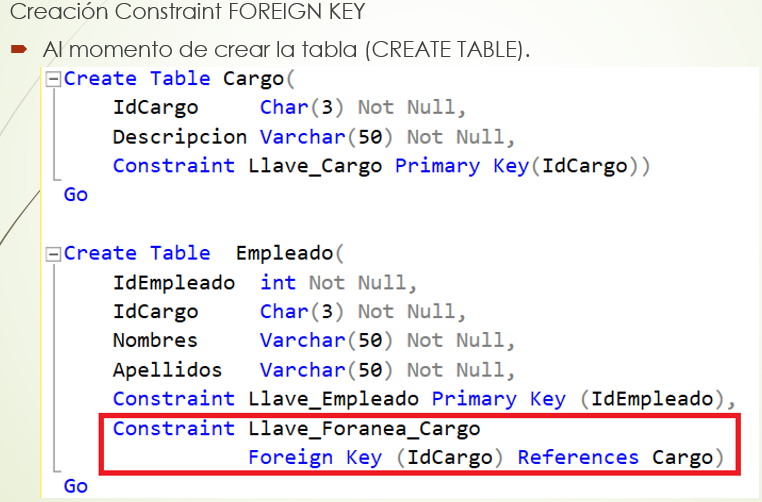


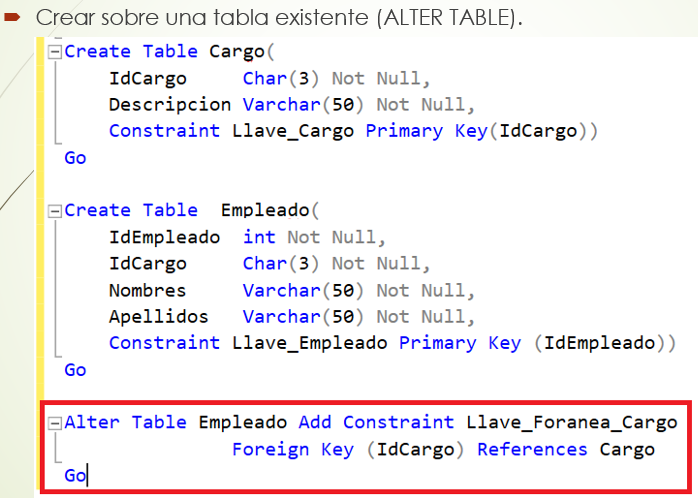


**INTEGRIDAD REFERENCIAL**

* Constraint **FOREIGN KEY**
* Un Foreign Key permite definir una clave externa, que es una columna o combinación de columnas, que se utiliza para establecer y exigir un vínculo entre los datos de dos tablas.
* Evidencia una relación física entre dos entidades
* En una relación “Uno a Muchos” la llave del lado “Uno” es referenciada desde el lado “Muchos”

**CREACIÓN DE CONSTRAINT FOREIGN KEY**

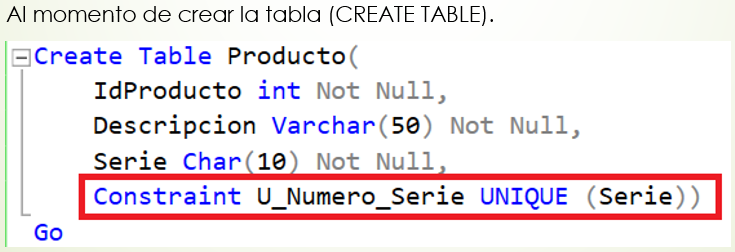




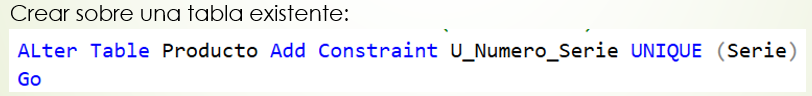
**INTEGRIDAD DE DATOS.**

**CONSTRAINT UNIQUE.**

* Garantiza que no se escriban valores duplicados en columnas específicas que no forman parte de una clave principal.
* En una tabla se pueden definir varias restricciones **UNIQUE**.
* **UNIQUE** admiten valores **NULL**. Sin embargo, de la misma forma que cualquier valor incluido en una restricción **UNIQUE**, sólo se admite un valor **NULL** por columna.

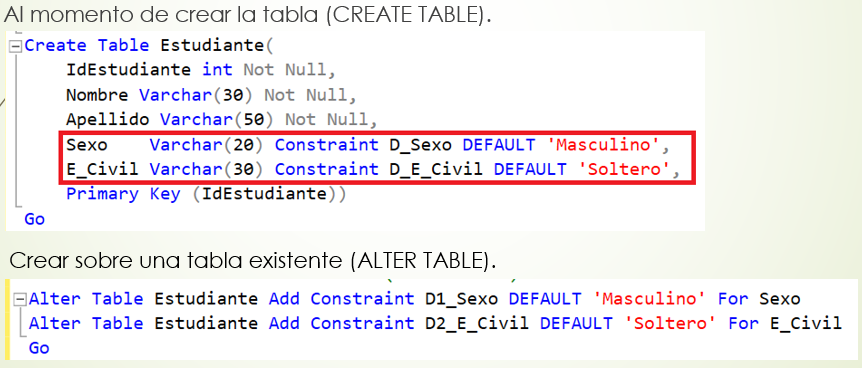


|



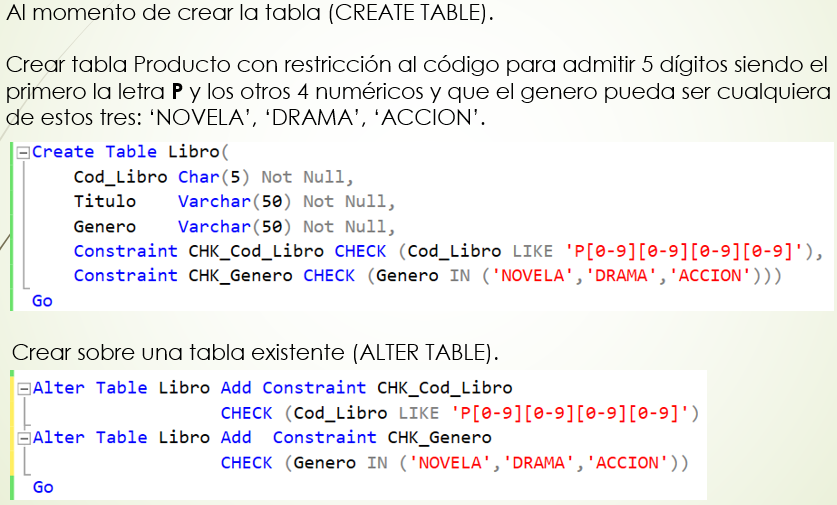
**CONSTRAINT DEFAULT**

* Se asigna un valor predeterminado a la columna.
* En caso de dejarse vacía al momento de la creación de la fila se registra dicho valor.

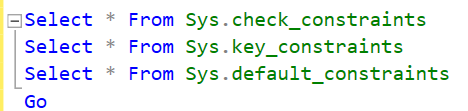


**CONSTRUCTOR CHECK.**

* **CHECK** exige la integridad del dominio mediante la limitación de los valores que puede aceptar una columna a partir de una expresión lógica que no se basa en datos de otra columna.
* Este rango puede ser un conjunto de valores estáticos: (‘MASCULINO’,’FEMENINO’)
* Este rango puede ser una condición de validez: =, <, >, >=, <=.
* Este rango puede tener relación con otras entidades.



**VISUALIZAR LOS CONSTRUCTORES CREADOS, PARA ELLOS USAREMOS EL DICCIONARIO DE DATOS O LLAMADOS CATALOGOS.**



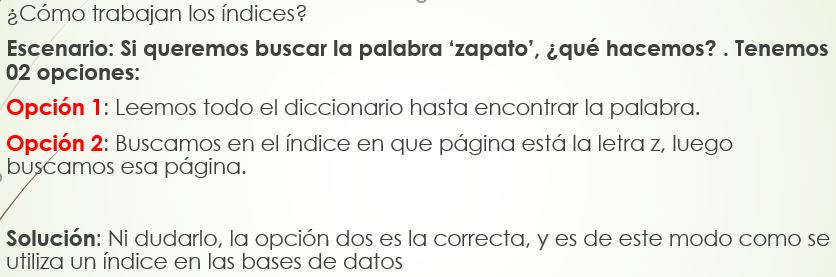
**INDICES.**

**¿Qué es un índice?**

Un índice es una estructura de datos que permite acceder a diferentes filas de una misma tabla a través de un campo (o campos clave).

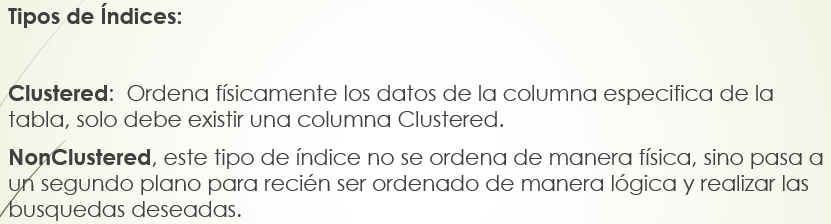
* Un índice permite un acceso mucho más rápido a los datos.
* Cada tabla se divide en páginas de datos, o bien en páginas a las que podemos acceder rápidamente a través de un índice.
* Está idea es la que se aplica en el mundo de las bases de datos, la información esta guardada en una tabla (el libro) que tiene muchas hojas de datos (las páginas del libro), con un índice en el que podemos buscar la información que nos interesa.

**GESTION DE INDICES**



Video Referencial: <https://www.youtube.com/watch?v=XWX1YvS5Kec>

**TIPOS DE INDICES.**

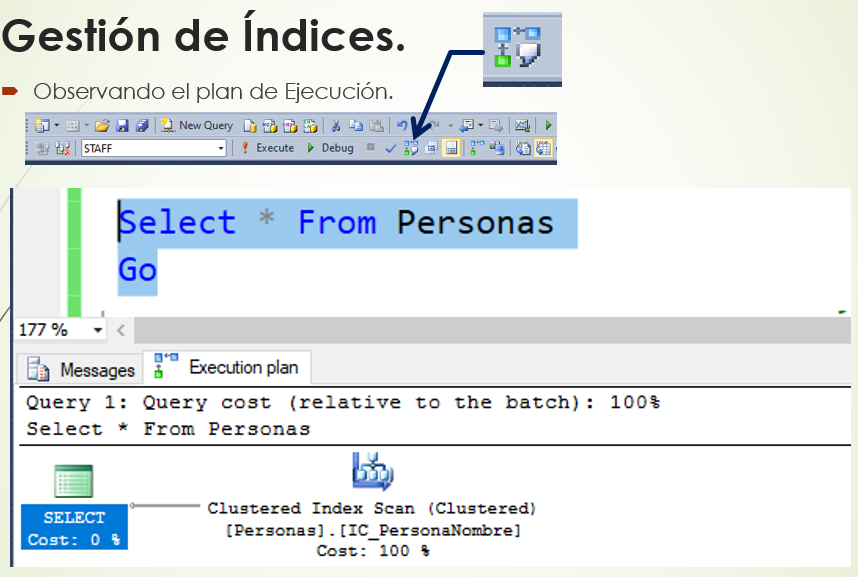


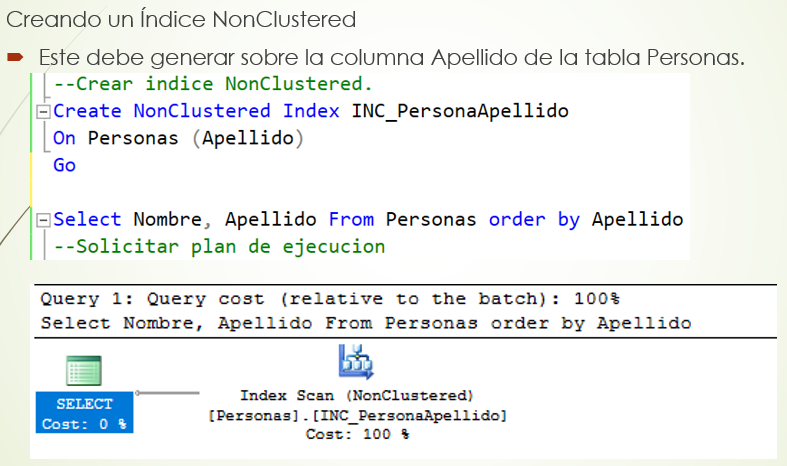
**EJEMPLO DE INDICES CLUSTERED y NON CLUSTERED**

Se tiene la siguiente tabla de datos. **PERSONAS**.









**Videos Referencial:**

<https://www.youtube.com/watch?v=XWX1YvS5Kec>

<https://www.youtube.com/watch?v=y1TxR53RlYU>

**CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA**

* Crear la base datos utilizando el SQL Server 2016
* Comprender sus relaciones con sentencias y restricciones a través de la plataforma T-SQL y los lenguajes en que se divide.
* Las tablas están en el core de SQL Server y del modelo relacional en general.
* Creación de restricciones de tablas.
* Las tablas relacionadas y separadas en grupos ayudan a optimizar el rendimiento obteniendo mejores tiempos de respuestas de las consultas.

**ACTIVIDAD VIRTUAL**

Analizar y revisar los siguientes enlaces; luego responde las preguntas propuestas.

<https://www.youtube.com/watch?v=9ZI8Bm5hYHg>

<https://www.youtube.com/watch?v=y1TxR53RlYU>

<https://www.youtube.com/watch?v=lUilQHMbToI>

<https://www.youtube.com/watch?v=l9Ijy1g5Doc>

<https://www.youtube.com/watch?v=wdbXJJH_JOU>

* ¿Cuál es la diferencia entre el DDL y DML?
* Facilite un ejemplo de archivos de grupo.
* ¿Qué es un constructor?
* ¿Qué es un índice?
* Tipos de índices y utilidad
* ¿En qué se diferencia realizar la sentencia DROP con TRUNCATE?
* ¿Qué instrucciones podemos realizar con la sentencia WHERE?