



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®

**Instituto Tecnológico de Nuevo Laredo**

***INGENIERIA EN SISTEMAS COMPUTACIONALES***

## ***Taller de Base de Datos***

**Unidad 1.**

**Integrantes:**

***Ivan Rosales Cancino (20100259)***

***Guillermo Del Bosque García (20100184)***

***Docente: Ing. Humberto Peña Valle***



# INDICE

1. Introducción
2. Manejadores de tipos de datos en SQL Server
  - 2.1 Tipos de datos numéricos.
  - 2.2 Tipos de datos de cadenas de texto.
  - 2.3 Tipo de datos booleano.
  - 2.4 Tipo de dato de fecha y hora.
  - 2.5 Tipos de datos binarios.
3. Restricciones de datos
  - 3.1 Primary Key (Clave Primaria)
  - 3.2 Foreign Key (Clave Foránea)
  - 3.3 Check (Restricción de Comprobación)
  - 3.4 Default (Restricción de valor predeterminado)
  - 3.5 Constraint (Restricción)
4. Índices
5. Reporte de la practica en clase
6. Conclusión
7. Fuentes bibliográficas

# 1. INTRODUCCION

En esta unidad se comprenderá sobre el mundo de las bases de datos, el manejo eficiente de los datos es fundamental para garantizar un rendimiento óptimo y una gestión eficaz de la información, al igual de los lenguajes de definición de datos, consultas y lenguaje de manipulación de datos. Los manejadores de tipos de datos, las restricciones de datos y los índices son herramientas esenciales en SQL (Structured Query Lenguaje) que permiten definir, organizar y acceder a los datos de manera efectiva. En esta introducción, exploraremos cada uno de estos temas y su importancia en el contexto de las bases de datos relacionales.

Tenga en cuenta que para poder comprender con detalle la información que se recopilara en este reporte, tiene que obtener los conocimientos que se otorgaron en la materia de Fundamentos de base de datos, esto con el fin de poder brindarle mejor la información.

## 2. Manejadores de tipos de datos en SQL Server

Los manejadores de tipos de datos en SQL son responsables de definir y gestionar los diferentes tipos de datos que pueden almacenarse en una base de datos. Estos tipos de datos incluyen números, cadenas de texto, fechas, booleanos y más. Cada tipo de dato tiene sus propias características y restricciones, como el tamaño de almacenamiento y el rango de valores permitidos. El uso adecuado de los tipos de datos es crucial para garantizar la integridad y la consistencia de los datos en la base de datos.

A continuación, se dará una explicación y ejemplos sobre el uso de tipos de datos numéricos, cadenas y booleanas:

### 2.1 Tipos de datos numéricos:

- **TINYINT:** Almacenan numéricos enteros pequeños. Tiene una precisión de 8 bits y puede almacenar valores en un rango de 0 a 255 o de -128 a 127 si se utiliza el signo.
- **SMALLINT:** Almacena números enteros más grandes que 'TINYINT'. Tiene una precisión de 16 bits y puede almacenar valores en un rango de -32,768 a 32,767.
- **INT:** Almacena enteros aún más grandes que 'SMALLINT'. Tiene una precisión de 32 bits y puede almacenar valores en un rango de aproximadamente -2,147,483,648 a 2,147,483,647.
- **BIGINT:** Almacena enteros muy grandes y por el momento es el dato numérico más grande de todos. Tiene una precisión de 64 bits y puede almacenar valores en un rango de aproximadamente -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807.

- **SMALLMONEY:** Almacena valores monetarios en un rango más limitado, aproximadamente de -214,748.3648 a 214,748.3647.
- **MONEY:** se utiliza para almacenar valores de tipo monetario. Tiene una precisión fija de 19 dígitos, con 4 dígitos a la derecha del punto decimal. Esto permite almacenar valores monetarios en un rango de aproximadamente -922,337,203,685,477.5808 a 922,337,203,685,477.5807.

## 2.2 Tipos de datos de cadena de texto:

- **CHAR:** Almacena cadenas de texto de longitud fija con una longitud máxima de n caracteres.
- **VARCHAR:** Almacena cadenas de texto de longitud variable con una longitud máxima de n caracteres.

## 2.3 Tipo de datos booleano:

- **BOOLEAN:** Almacena valores de verdad, es decir, verdadero o falso. A menudo representado como 1 para verdadero y 0 para falso.

## 2.4 Tipos de datos de fecha y hora:

- **DATE:** Almacena valores de fecha en formato 'YYYY-MM-DD'.
- **TIME:** Almacena valores de tiempo en formato 'HH:MM:SS'.
- **DATETIME:** Almacena valores de fecha y hora combinados en formato 'YYYY-MM-DD HH:MM:SS'.
- **TIMESTAMP:** Almacena marcas de tiempo, generalmente utilizadas para registrar la fecha y hora en que se inserta o actualiza un registro.

## 2.5 Tipos de datos binarios:

- **BLOB:** Almacena datos binarios de longitud variable, como imágenes, audio o vídeo.
- **BINARY(n):** Almacena datos binarios de longitud fija con una longitud máxima de n bytes.

### 3. Restricciones de datos

Las restricciones de datos en SQL son reglas que se aplican a las columnas de una tabla para limitar los tipos de valores que pueden almacenarse en ellas. Estas restricciones pueden incluir restricciones de clave primaria, restricciones de clave foránea, restricciones de unicidad, restricciones de valores predeterminados y más. Al aplicar restricciones de datos, se asegura la integridad de los datos y se previenen errores e inconsistencias en la base de datos.

A continuación, se describirá cada restricción de dato:

- **3.1 Primary Key (Clave Primaria):** La restricción de clave primaria se utiliza para identificar de forma única cada fila en una tabla. Garantiza que no haya valores duplicados en la columna o conjunto de columnas especificado como clave primaria.
- **3.2 Foreign Key (Clave Foránea):** La restricción de clave foránea o externa, se utiliza para mantener la integridad referencial entre dos tablas. Requiere que los valores en una columna (o conjunto de columnas) de una tabla (clave foránea) coincidan con los valores en la clave primaria de otra tabla (clave primaria).
- **3.3 Check (Restricción de Comprobación):** La restricción de comprobación se utiliza para asegurar que los valores en una columna cumplan con una condición específica. Puede especificar una expresión booleana que debe evaluarse como verdadera para que los valores sean aceptados.
- **3.4 Default (Restricción de valor predeterminado):** La restricción de valor predeterminado se utiliza para establecer un valor predeterminado para una columna cuando no se proporciona ningún valor durante la inserción de datos. Esto evita que la columna tenga un valor NULL si no se especifica un valor explícito.
- **3.5 Constraint (Restricción):** Es una regla o condición que se aplica a los datos almacenados en una tabla. Estas restricciones se utilizan para garantizar la integridad de los datos, es decir, para asegurar de que los datos cumplan con ciertas reglas o condiciones específicas definidas por el usuario.

## 4. Índices

Los índices en SQL son estructuras que se utilizan para mejorar el rendimiento de las consultas al acelerar la búsqueda y recuperación de datos. Los índices se crean en columnas específicas de una tabla y almacenan referencias ordenadas a los datos, lo que permite una búsqueda más eficiente basada en esos valores. Los índices son especialmente útiles para tablas grandes o para columnas que se utilizan con frecuencia en las consultas. Sin embargo, el uso excesivo de índices puede afectar negativamente el rendimiento de las operaciones de inserción y actualización.

## 5. Reporte de la practica en clase

```
CREATE DATABASE CRUZROJAA
GO

USE CRUZROJAA

CREATE TABLE TIPODONADOR
(
    IDTIPODONADOR TINYINT NOT NULL IDENTITY(1,1),
    DESCRIPTIPO VARCHAR(20) NOT NULL,
    CONSTRAINT PK_TIPODONADOR PRIMARY KEY(IDTIPODONADOR)
)

CREATE TABLE ACTIVIDAD
(
    IDACTIVIDAD TINYINT NOT NULL IDENTITY(1,1),
    DESCRIPACTIVIDAD VARCHAR(20) NOT NULL,
    MONTOAPORTADO SMALLMONEY NOT NULL CONSTRAINT DF_MONTOAPORTADO DEFAULT 0
)
```

```

CREATE TABLE DONADOR
(
  IDDONADOR INT NOT NULL IDENTITY(1,1),
  APPATERNO VARCHAR(20) NOT NULL CONSTRAINT DF_APPATERNO DEFAULT '',
  APMATERNO VARCHAR(20) NOT NULL CONSTRAINT DF_APMATERNO DEFAULT '',
  NOMBRE VARCHAR(20) NOT NULL CONSTRAINT DF_NOMBRE DEFAULT '',
  NOMDONADOR AS (APPATERNO+' '+APMATERNO+' '+NOMBRE),
  CALLE VARCHAR(30) NOT NULL CONSTRAINT DF_CALLE DEFAULT '',
  NUMEXT VARCHAR(10) NOT NULL CONSTRAINT DF_NUMEXT DEFAULT '',
  NUMINT VARCHAR(5) NOT NULL CONSTRAINT DF_NUMINT DEFAULT '',
  COLONIA VARCHAR(30) NOT NULL CONSTRAINT DF_COLONIA DEFAULT '',
  CP CHAR(5) NOT NULL CONSTRAINT DF_CP DEFAULT '',
  DIRDONADOR AS (CALLE+' '+NUMEXT+' '+NUMINT+' '+COLONIA+' '+CP),
  RFC CHAR(15),
  IDTIPODONADOR TINYINT NOT NULL,
  IDACTIVIDAD TINYINT NOT NULL,
  CONSTRAINT PK_DONADOR PRIMARY KEY(IDDONADOR),
  CONSTRAINT FK_DONADOR_TIPODONADOR FOREIGN KEY(IDTIPODONADOR)
  REFERENCES TIPODONADOR (IDTIPODONADOR)
  ON DELETE NO ACTION ON UPDATE NO ACTION
)

CREATE TABLE MEDIO
(
  IDMEDIO TINYINT NOT NULL IDENTITY(1, 1),
  DESCRIPMEDIO VARCHAR(20) NOT NULL,
  CONSTRAINT PK_MEDIO PRIMARY KEY (IDMEDIO)
)

CREATE TABLE ESTADO
(
  IDESTADO TINYINT NOT NULL,
  NOMEESTADO VARCHAR(25) NOT NULL,
  CONSTRAINT PK_ESTADO PRIMARY KEY (IDESTADO),
  CONSTRAINT CK_IDESTADO CHECK(IDESTADO <=32)
)

CREATE TABLE ESTADOMEDIO
(
  IDESTADO TINYINT NOT NULL,
  IDMEDIO TINYINT NOT NULL,
  CONSTRAINT PK_ESTADOMEDIO PRIMARY KEY (IDESTADO, IDMEDIO),
  CONSTRAINT FK_ESTADOMEDIO_ESTADO FOREIGN KEY (IDESTADO)
  REFERENCES ESTADO (IDESTADO)
  ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT FK_ESTADOMEDIO_MEDIO FOREIGN KEY (IDMEDIO)
  REFERENCES MEDIO (IDMEDIO)
  ON DELETE NO ACTION ON UPDATE NO ACTION
)

```

```

CREATE TABLE DONACION
(
  IDDONACION INT NOT NULL IDENTITY(1,1),
  FECHADONACION DATE NOT NULL DEFAULT GETDATE(),
  IDDONADOR INT NOT NULL,
  IDESTADO TINYINT NOT NULL,
  IDMEDIO TINYINT NOT NULL,
  CONSTRAINT PK_DONACION PRIMARY KEY (IDDONACION),
  CONSTRAINT CK_FECHADONACION CHECK (FECHADONACION =CAST(GETDATE() AS DATE)),
  CONSTRAINT FK_DONACION_DONADOR FOREIGN KEY (IDDONADOR)
REFERENCES DONADOR (IDDONADOR)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT FK_DONACION_ESTADOMEDIO FOREIGN KEY (IDESTADO, IDMEDIO)
REFERENCES ESTADOMEDIO (IDESTADO, IDMEDIO)
ON DELETE NO ACTION ON UPDATE CASCADE
)

ALTER TABLE ACTIVIDAD ADD CONSTRAINT PK_ACTIVIDAD PRIMARY KEY(IDACTIVIDAD)

ALTER TABLE DONADOR ADD CONSTRAINT FK_DONADOR_ACTIVIDAD FOREIGN KEY(IDACTIVIDAD)
REFERENCES ACTIVIDAD (IDACTIVIDAD)
ON DELETE NO ACTION ON UPDATE NO ACTION

ALTER TABLE ACTIVIDAD ADD CONSTRAINT CK_MONTOAPORTADO CHECK(MONTOAPORTADO >= 0)

ALTER TABLE DONACION ADD MONTO MONEY NOT NULL
ALTER TABLE DONACION ADD CONSTRAINT CK_MONTO CHECK (MONTO >0)

ALTER TABLE DONACION DROP COLUMN MONTO

ALTER TABLE DONACION DROP CK_MONTO
ALTER TABLE DONACION ALTER COLUMN MONTO MONEY NULL
ALTER TABLE DONACION ALTER COLUMN MONTO MONEY NULL
ALTER TABLE DONACION DROP COLUMN MONTO

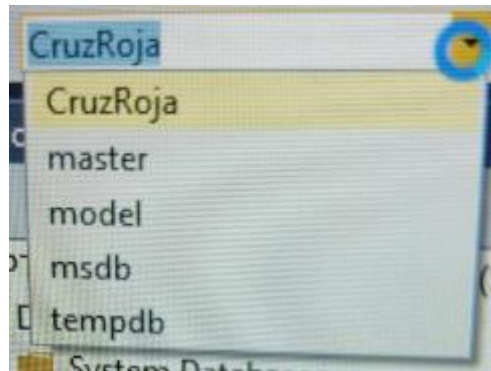
CREATE INDEX IDX_FECHADONACION ON DONACION (FECHADONACION)
DROP INDEX DONACION.IDX_FECHADONACION

```

Para la creación de esta actividad nos basamos en un ejemplo de unas donaciones de una cruz roja, por lo que tuvimos que analizar detalladamente los puntos más importantes para crear las tablas que correspondían, por lo que terminamos creando 7 tablas que son la actividad, el donador, donaciones, El tipo donador, Estado, EstadoMedio y Medio.

Una vez obtenido la información de cada tabla, es hora de implementarlo en la base de datos por lo que comenzamos a crear la base de datos y una vez creada, tenemos que tener en cuenta que debemos de cambiar la base de datos correspondiente, ya que si no lo hacemos podríamos perder datos por haberlo creado en otra base de datos, Para eso irán a la parte superior izquierda y encontraran lo siguiente:





Ahí cambiamos la base de datos que en este caso es la Cruz Roja.

Después, creamos la primera tabla que es la de actividades porque esta tabla es una entidad fuerte ya que no depende de otra tabla, después creamos la tabla Tipo donador y estas dos tablas estarán enlazadas con otra tabla llamada "Donador", esta tabla si se considera como una entidad débil porque depende de las tablas de Actividad y Tipo Donador.

Ahora continuamos creando la tabla de Medio que este nos ayudara a registrar los medios de comunicación que recibe la cruz roja para hacer donaciones, solo nos tomara su ID y la descripción, después, seguirá la tabla del Estado que se refiere al estado del país de México donde nos están dando la donación, ya que esta cruz roja recibe donaciones en todas partes de México, por lo que tiene que considerar los estados, ahora como ambos son entidades que reciben de muchos a muchos se creara una tabla intermedia que se llamara "EstadoMedio" solo nos ayudara a mantener un control en los sistemas de datos que vaya recibiendo.

Por último, solo queda a crear la tabla de Donación, Esta tabla se crea al último porque depende de las tablas EstadoMedio y Donador, si se crea antes que las otras tablas serían más difíciles y ocasionaría error en la base de datos por falta de atributos.

## 6. Conclusión

En resumen, los manejadores de datos, son muy necesarios y hay que reconocer cada tipo de datos como sus funciones y su capacidad, ya que dependiendo el tipo de datos que vayamos a manejar, ahí tenemos la decisión del tipo de dato que sea adecuado para un atributo, no importa que sea de cadena de texto, numérico, fecha y entre otros, aquí hacemos el conocimiento de las principales herramientas de las bases de datos, como son las creaciones de base de datos, tablas, índices, también el manejo de las restricciones de datos que nos permite hacer llaves primarias, llaves foráneas que son las conexiones entre una tabla de entidad fuerte o débil.

Mi recomendación es que por experiencia propia, he estudiado por mi cuenta todos estos datos para el uso de base de datos viendo tutoriales o metiéndome a cursos del mismo tema, pero la verdad, a como me lo explico el maestro, fue más detallado y más comprensible con cada dato y herramientas, en ningún video o tutorial podrían explicarte tan afondo.

## 7. Fuentes bibliográficas

- <https://learn.microsoft.com/es-es/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>
- <https://docs.microsoft.com/es-es/sql/relational-databases/data-handling/data-handling?view=sql-server-ver15>
- <https://docs.microsoft.com/es-es/sql/relational-databases/data-constraints/data-constraints?view=sql-server-ver15>
- <https://docs.microsoft.com/es-es/sql/relational-databases/data-definition/ddl-sql-server?view=sql-server-ver15>
- <https://docs.microsoft.com/es-es/sql/relational-databases/data-language-reference/queries-data-language-reference?view=sql-server-ver15>
- <https://docs.microsoft.com/es-es/sql/relational-databases/data-manipulation/dml?view=sql-server-ver15>