Universidad Tecnológica de Puebla

Base de Datos II

Garrido Hernández Jorge

3° Cuatrimestre

Mayo-Agosto

Grupo: “H”

Uso de Comandos: SQL Server

Profesor: José Francisco Espinosa Garita

Tabla de Contenidos :v

[Introducción 3](#_Toc483417975)

[Check 3](#_Toc483417976)

[Unique 4](#_Toc483417977)

[Default 5](#_Toc483417978)

[PK (Primary Key) 6](#_Toc483417979)

[FK (Foreign Key) 7](#_Toc483417980)

[Dominio 7](#_Toc483417981)

[Autoincremento 8](#_Toc483417982)

# Introducción

El presente reporte, que sirve a manera de repaso, nos ayudara a comprender un poco mejor las restricciones que se tienen al manejar una base de datos, enfocado principalmente al SGBD que se utilizara en este cuatrimestre (SQL Server)

Las restricciones le permiten definir la manera en que Motor de base de datos exigirá automáticamente la integridad de una base de datos. Las restricciones definen reglas relativas a los valores permitidos en las columnas y constituyen el mecanismo estándar para exigir la integridad.

**Uso de Comandos en SQL Server**

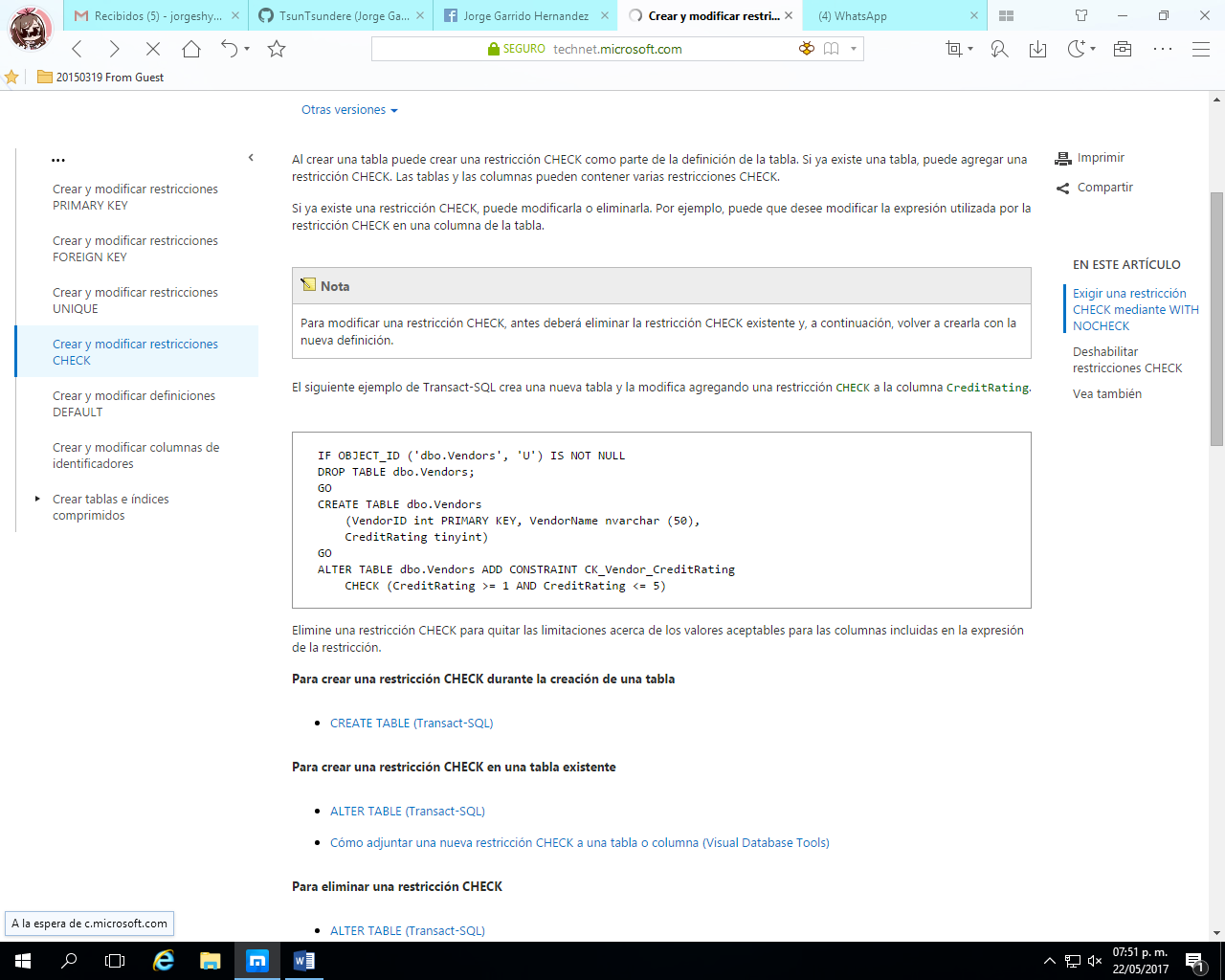
# Check

Las restricciones CHECK exigen la integridad del dominio mediante la limitación de los valores que puede aceptar una columna. Son similares a las restricciones FOREIGN KEY porque controlan los valores que se colocan en una columna. La diferencia reside en la forma en que determinan qué valores son válidos: las restricciones FOREIGN KEY obtienen la lista de valores válidos de otra tabla, mientras que las restricciones CHECK determinan los valores válidos a partir de una expresión lógica que no se basa en datos de otra columna.

Puede crear una restricción CHECK con cualquier expresión lógica (booleana) que devuelva TRUE (verdadero) o FALSE (falso) basándose en operadores lógicos.

Al crear una tabla puede crear una restricción CHECK como parte de la definición de la tabla. Si ya existe una tabla, puede agregar una restricción CHECK. Las tablas y las columnas pueden contener varias restricciones CHECK.

Si ya existe una restricción CHECK, puede modificarla o eliminarla. Por ejemplo, puede que desee modificar la expresión utilizada por la restricción CHECK en una columna de la tabla.



# Unique

Puede utilizar restricciones UNIQUE para garantizar que no se escriben valores duplicados en columnas específicas que no forman parte de una clave principal. Tanto la restricción UNIQUE como la restricción PRIMARY KEY exigen la unicidad; sin embargo, debe utilizar la restricción UNIQUE y no PRIMARY KEY si desea exigir la unicidad de una columna o una combinación de columnas que no forman la clave principal.

En una tabla se pueden definir varias restricciones UNIQUE, pero sólo una restricción PRIMARY KEY.

Además, a diferencia de las restricciones PRIMARY KEY, las restricciones 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.

Es posible hacer referencia a una restricción UNIQUE con una restricción FOREIGN KEY.

En el siguiente fragmento de una instrucción CREATE TABLE se muestra la definición de la columna Name, en la que se especifica una restricción UNIQUE para garantizar que los valores son únicos.

Name nvarchar(100) NOT NULL

UNIQUE NONCLUSTERED

Para quitar el requisito de unicidad de los valores de una columna o de una combinación de columnas incluida en la restricción, elimine una restricción UNIQUE. No puede eliminar una restricción UNIQUE si la columna asociada se utiliza como clave de texto de la tabla.

# Default

Cada columna de un registro debe contener un valor, aunque sea un valor NULL. Puede haber situaciones en las que deba cargar una fila de datos en una tabla, pero no conozca el valor de una columna o el valor ya no exista. Si la columna acepta valores NULL, puede cargar la fila con un valor NULL. Pero, dado que puede no resultar conveniente utilizar columnas que acepten valores NULL, una mejor solución podría ser establecer una definición DEFAULT para la columna siempre que sea necesario. Por ejemplo, es habitual especificar el valor cero como valor predeterminado para las columnas numéricas, o N/D (no disponible) como valor predeterminado para las columnas de cadenas cuando no se especifica ningún valor.

Al cargar una fila en una tabla con una definición DEFAULT para una columna, se indica implícitamente a Motor de base de datos que cargue un valor predeterminado en la columna en la que no se haya especificado ningún valor.

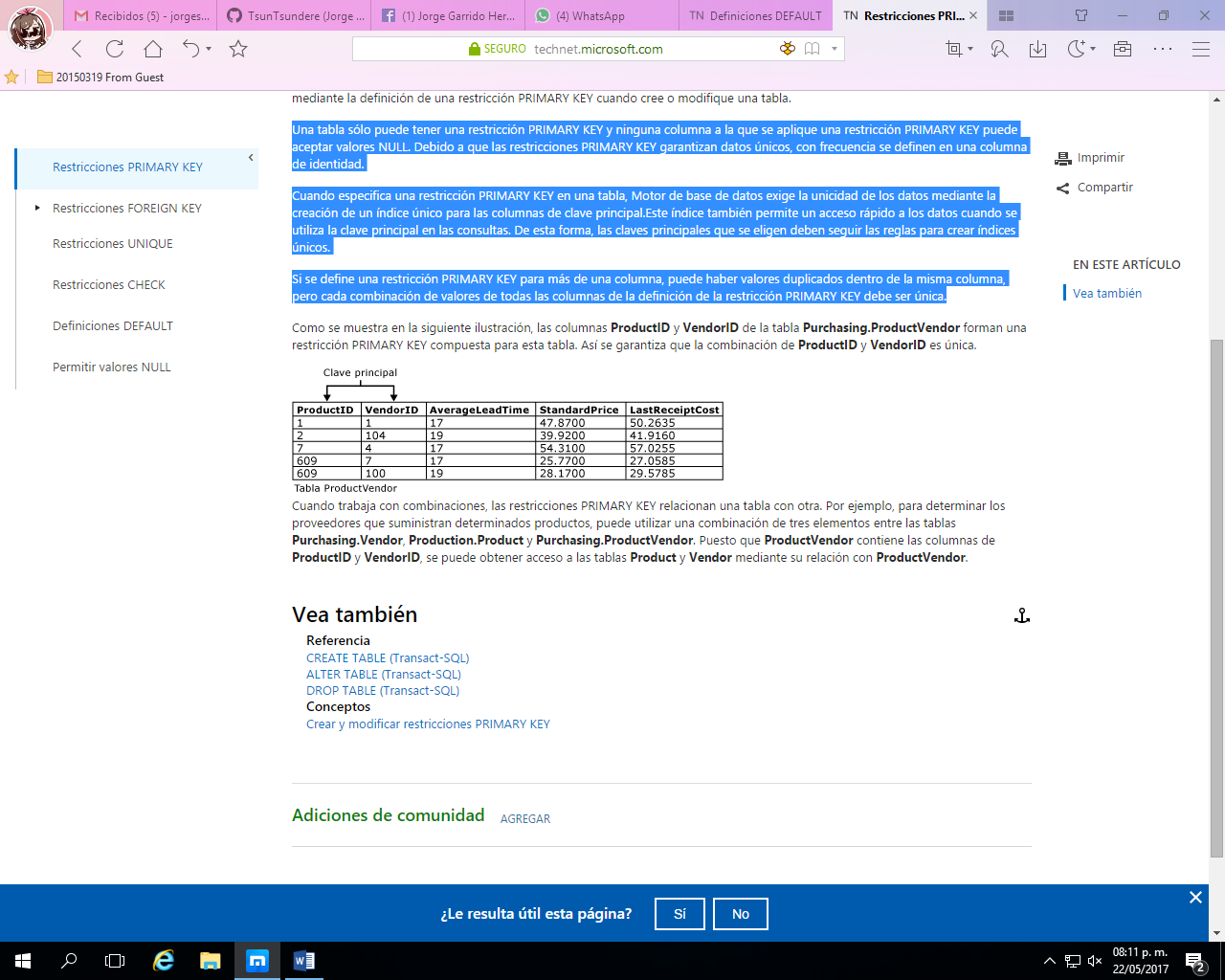
|  |  |  |  |
| --- | --- | --- | --- |
| Definición de columna | Ninguna entrada,  ninguna definición DEFAULT | Ninguna entrada,  definición DEFAULT | Especifique un valor NULL |
| Permite valores NULL | NULL | Valor predeterminado | NULL |
| No permite valores NULL | Error | Valor predeterminado | Error |

# PK (Primary Key)

Una tabla sólo puede tener una restricción PRIMARY KEY y ninguna columna a la que se aplique una restricción PRIMARY KEY puede aceptar valores NULL. Debido a que las restricciones PRIMARY KEY garantizan datos únicos, con frecuencia se definen en una columna de identidad.

Cuando especifica una restricción PRIMARY KEY en una tabla, Motor de base de datos exige la unicidad de los datos mediante la creación de un índice único para las columnas de clave principal. Este índice también permite un acceso rápido a los datos cuando se utiliza la clave principal en las consultas. De esta forma, las claves principales que se eligen deben seguir las reglas para crear índices únicos.

Si se define una restricción PRIMARY KEY para más de una columna, puede haber valores duplicados dentro de la misma columna, pero cada combinación de valores de todas las columnas de la definición de la restricción PRIMARY KEY debe ser única.



# FK (Foreign Key)

* **Restringida:** El borrado de las tuplas de la relación que contiene la clave referenciada solo se permite si no existen tuplas con ese valor en la relación que contiene la clave ajena.
* **Cascada:** El borrado de tuplas de la relación que contiene la clave candidata referenciada, lleva consigo el borrar en cascada las tuplas de la relación que contienen la clave ajena.
* **Null:** El borrado de tuplas de la relación que contiene la clave candidata referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia. Esta opción solo es posible cuando el atributo que es clave ajena admite valores nulos.
* **Default:** El borrar tuplas de la relación que contiene la clave candidata referenciada lleva consigo poner el valor por defecto a la clave ajena de la relación que referencia, valor por defecto que habría sido definido al crear la tabla correspondiente.

# Dominio

Un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben de ser elementos del conjunto especificado".

Hay dos maneras de crear un nuevo dominio. La primera es durante el paso de asignación de la actividad de detección de conocimiento, cuando se analiza una muestra de los datos para agregar conocimiento a una base de conocimiento nueva o a una ya existente. La segunda es durante la actividad de administración de dominios, cuando se crea un nuevo dominio en lugar de modificar uno ya existente.

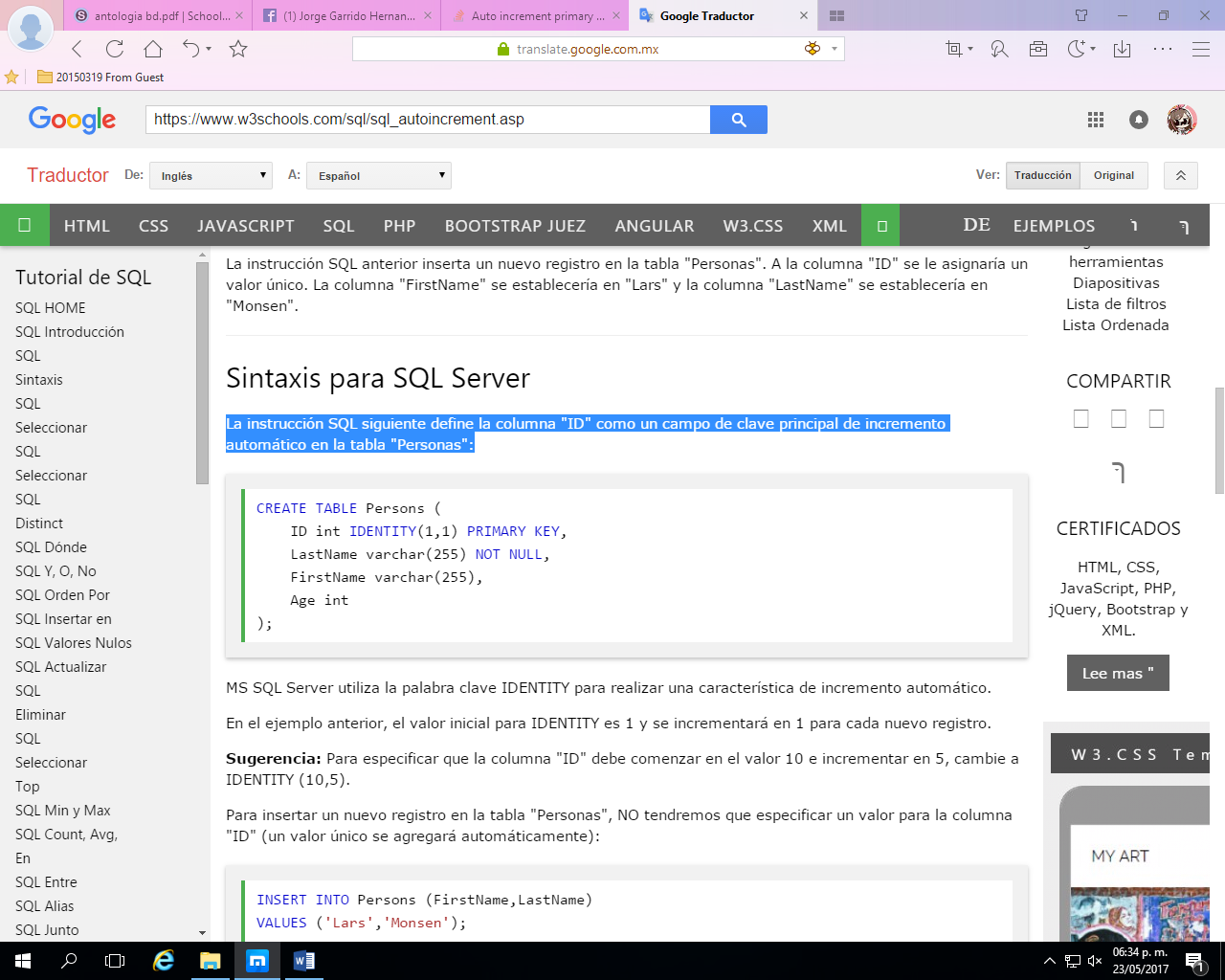
* **CHAR (p):** Cadena de caracteres de longitud fija p (máxima longitud p)
* **VARCHAR (p):** Cadena de caracteres de longitud variable. Maxima longitud p.
* **NUMERICOS:** Valor numérico de los siguientes tipos: enteros, precisión o punto flotante.
* **DATE:** Fechas válidas.
* **BINARY:** Binarios utilizados para almacenar cadenas de bytes (BLOB).

# Autoincremento

El incremento automático permite generar automáticamente un número único cuando se inserta un nuevo registro en una tabla.

A menudo, este es el campo de clave principal que nos gustaría que se creara automáticamente cada vez que se inserta un nuevo registro.

La instrucción SQL siguiente define la columna "ID" como un campo de clave principal de incremento automático en la tabla "Personas":



MS SQL Server utiliza la palabra clave IDENTITY para realizar una característica de incremento automático. En el ejemplo anterior, el valor inicial para IDENTITY es 1 y se incrementará en 1 para cada nuevo registro.

# Conclusión

Como pudimos observar en la extensión de este documento, pudimos comprobar la importancia de las restricciones dentro de las bases de datos, esto con el objetivo de aplicarlas y de ese modo la base de datos tenga un correcto funcionamiento y podamos evitar errores futuros.

# Bibliografía

https://technet.microsoft.com/es-es/library/ms190765(v=sql.105).aspx

Antologia BD

Universidad Tecnologica de Puebla

Pags: 51-54

https://msdn.microsoft.com/es-es/library/ms187550.aspx