

# Fundamentos de Bases de Datos.

## Práctica 6.

Profesor: M.I. Gerardo Avilés Rosas  
*gar@ciencias.unam.mx*  
Laboratorio: Luis Eduardo Castro Omaña  
*lalo\_castro@ciencias.unam.mx*

14 de marzo de 2017

Se dan a conocer especificaciones de entrega para la practica 6.

### 1. DDL

Un esquema de base de datos se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (DDL)<sup>1</sup>.

Un diccionario de datos contiene metadatos, es decir, datos acerca de los datos. El esquema de una tabla es un ejemplo de metadatos. Un sistema de base de datos consulta el diccionario de datos antes de leer o modificar los datos reales.

Especificamos el almacenamiento y los métodos de acceso usados por el sistema de bases de datos por un conjunto de instrucciones en un tipo especial de DDL denominado lenguaje de almacenamiento y definición de datos. Estas instrucciones definen los detalles de implementación de los esquemas de base de datos, que se ocultan usualmente a los usuarios.

### 2. sqlcmd

La utilidad sqlcmd es una herramienta de línea de comandos para la ejecución ad hoc e interactiva de instrucciones y scripts de Transact-SQL y para la automatización de tareas de creación de scripts de Transact-SQL. Es posible utilizar la herramienta de las siguientes formas:

---

<sup>1</sup>Siglas en inglés de Data Definition Language

- Los usuarios escriben instrucciones Transact-SQL interactivamente de una forma similar al modo en que trabajan con el símbolo del sistema. Los resultados se muestran en el símbolo del sistema.
- Los usuarios envían un trabajo sqlcmd especificando la ejecución de una instrucción Transact-SQL individual o dirigiendo la utilidad hacia un archivo de texto que contiene las instrucciones Transact-SQL que se van a ejecutar. El resultado se dirige normalmente hacia un archivo de texto, aunque también se puede mostrar en el símbolo del sistema.

Algunos de los comandos mas utilizados en sqlcmd son:

- La opción del servidor (-S) que identifica la instancia de Microsoft SQL Server a la que se conecta sqlcmd.

```
sqlcmd -S <nombre-servidor>\<nombre-instancia>
```

- Las opciones de autenticación (-E, -U y -P) que especifican las credenciales que usa sqlcmd para conectarse a la instancia de SQL Server. Conectarse a la instancia con nombre mediante la Autenticación de Windows para ejecutar de forma interactiva instrucciones Transact-SQL:

```
sqlcmd -U <nombreUsuario> -S <nombre-servidor>\<nombre-instancia>
```

- Las opciones de entrada (-Q, -q e -i) que identifican la ubicación de la entrada a sqlcmd. Conectarse a una instancia con nombre mediante la Autenticación de Windows y especificar los archivos de entrada:

```
sqlcmd -S <nombre-servidor>\<nombre-instancia> -i <script.sql>
```

Conectarse a la instancia predeterminada del equipo local mediante la Autenticación de Windows, ejecutar una consulta y mantener la ejecución de sqlcmd después de la finalización de la consulta:

```
sqlcmd -q "<Instruccion Transact-SQL>"
```

- La opción de salida (-o) que especifica el archivo en el que se guardará la salida de sqlcmd.

```
sqlcmd -S <nombre-servidor>\<nombre-instancia> -i <script.sql> -o <MyOutput.rpt>
```

- Para ver una lista de opciones admitidas por la utilidad sqlcmd, ejecuta:

```
sqlcmd -?
```

- Para ver la lista de esquemas:

```
SELECT * FROM sysdatabases
```

- Utilizar un esquema:

```
USE <nombre_base>
```

- Ver las tablas de un esquema

```
SELECT TABLE_NAME FROM <nombre_esquema>.INFORMATION_SCHEMA.[TABLES]
```

Para mas información sobre sqlcmd puedes revisar la documentación en:  
<https://msdn.microsoft.com/es-es/library/ms162773.aspx>

### 3. Instrucciones DDL

Instrucción	Función
CREATE DATABASE	Crea una nueva base de datos y el archivo usado para almacenarla.
CREATE TABLE	Crea una nueva tabla.
ALTER TABLE	Modifica la definición de una tabla alterando, agregando o eliminando columnas y restricciones.
CREATE INDEX	Crea un índice en una tabla determinada.
DROP TABLE	Quita la definición de una tabla y todos sus datos, índices y restricciones.

#### 3.1. Crear Base De Datos

Crea una nueva base de datos y los archivos que se usan para almacenar la base de datos, una instantánea de base de datos, o adjunta una base de datos a partir de los archivos separados de una base de datos creada anteriormente.

```
CREATE DATABASE database_name  
[ COLLATE collation_name ]  
[ WITH <option> [,...n ] ]  
[;]
```

<option> ::=

```
{  
    FILESTREAM ( <filestream_option> [,...n ] )  
    | DEFAULT_FULLTEXT_LANGUAGE = { lcid | language_name | language_alias }  
    | DEFAULT_LANGUAGE = { lcid | language_name | language_alias }  
    | NESTED_TRIGGERS = { OFF | ON }  
    | TRANSFORM_NOISE_WORDS = { OFF | ON }  
    | TWO_DIGIT_YEAR_CUTOFF = <two_digit_year_cutoff>  
    | DB_CHAINING { OFF | ON }
```

```

    | TRUSTWORTHY { OFF | ON }
}

```

database\_name: Es el nombre de la nueva base de datos. Nombres de base de datos deben ser únicos dentro de una instancia de SQL Server y cumplir las reglas de identificadores.

database\_name puede tener un máximo de 128 caracteres, no es necesario especificar un nombre lógico del archivo de registro. Si no se especifica un nombre de archivo de registro lógico, SQL Server genera el nombre\_de\_archivo\_lógico y nombre\_de\_archivo\_de\_sistema\_operativo para el registro, anexando un sufijo a database\_name. Esto limita database\_name a 123 caracteres para que el nombre de archivo lógico generado es no más de 128 caracteres.

### 3.2. Crear Esquemas

Crea un esquema en la base de datos actual.

```
CREATE SCHEMA schema_name_clause [ <schema_element> [ ...n ] ]
```

```

<schema_name_clause> ::=
{
    schema_name
    | AUTHORIZATION owner_name
    | schema_name AUTHORIZATION owner_name
}

```

```

<schema_element> ::=
{
    table_definition | view_definition | grant_statement |
    revoke_statement | deny_statement
}

```

schema\_name: Es el nombre por el que se identifica al esquema en esta base de datos.

AUTHORIZATION owner\_name: Especifica el nombre de la entidad de seguridad de la base de datos que poseerá el esquema. Es posible que esta entidad de seguridad posea otros esquemas y no utilice el esquema actual como predefinido.

table\_definition: Especifica una instrucción CREATE TABLE que crea una tabla en el esquema. La entidad de seguridad que ejecuta esta instrucción debe tener el permiso CREATE TABLE en la base de datos actual.

**VIEW\_DEFINITION:** Especifica una instrucción **CREATE VIEW** que crea una vista en el esquema. La entidad de seguridad que ejecuta esta instrucción debe tener el permiso **CREATE VIEW** en la base de datos actual.

**grant\_statement:** Especifica una instrucción **GRANT** que otorga permisos sobre cualquier elemento protegible, excepto el esquema nuevo.

**revoke\_statement:** Especifica una instrucción **REVOKE** que revoca permisos sobre cualquier elemento protegible, excepto el esquema nuevo.

**deny\_statement:** Especifica una instrucción **DENY** que deniega permisos sobre cualquier elemento protegible, excepto el esquema nuevo.

### 3.3. Crear Tablas

Crea una nueva tabla en SQL Server

```
CREATE TABLE
    [ database_name . [ schema_name ] . | schema_name . ] table_name
    ( { <column_definition> } [ ,...n ] )
[ ; ]
```

**Database\_name:** Es el nombre de la base de datos en la que se crea la tabla. **database\_name** debe especificar el nombre de una base de datos existente. Si no se especifica, **database\_name** el valor predeterminado es la base de datos actual. El inicio de sesión para la conexión actual debe estar asociado con un identificador de usuario existente en la base de datos especificada por **database\_name**, y el Id. de usuario debe tener permisos **CREATE TABLE**.

**schema\_name:** Es el nombre del esquema al que pertenece la nueva tabla.

**table\_name:** Es el nombre de la nueva tabla. Los nombres de tabla deben seguir las reglas para identificadores. **table\_name** puede tener un máximo de 128 caracteres, excepto para los nombres de tablas temporales locales (nombres precedidos de un solo signo de número (#)) que no puede superar los 116 caracteres.

### 3.4. Editar Tablas

Modifica una definición de tabla al alterar, agregar o quitar columnas y restricciones, reasignar y regenerar particiones, o deshabilitar o habilitar restricciones y desencadenadores.

```
ALTER TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
{
    ALTER COLUMN column_name
    {
```

```

[ type_schema_name. ] type_name
[ (
    {
        precision [ , scale ]
        | max
        | xml_schema_collection
    }
) ]
[ NULL | NOT NULL ]
}

```

### 3.5. Tipos de datos

En SQL Server, cada columna, variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de datos es un atributo que especifica el tipo de datos que el objeto puede contener: datos de enteros, datos de caracteres, datos de moneda, datos de fecha y hora, cadenas binarias, etc.

Tipos de datos de SQL Server se organizan en las siguientes categorías:

- Numéricos exactos:
  - Enteros: Tipos de datos numéricos exactos que utilizan datos enteros.

Tipo de datos	Intervalo	Storage
bigint	De $-2^{63}(-9,223,372,036,854,775,808)$ a $2^{63} - 1(9,223,372,036,854,775,807)$	8 bytes
int	De $-2^{31}(-2,147,483,648)$ a $2^{31} - 1(2,147,483,647)$	4 bytes
smallint	De $-2^{15}(-32,768)$ a $2^{15} - 1(32,767)$	2 bytes
tinyint	De 0 a 255	1 byte

- Decimal y numéricos: Tipos de datos numéricos que tienen precisión y escala fijas. Decimal y numeric son sinónimos y se pueden usar indistintamente.

decimal[ (p[,s]) ] y numeric[ (p[,s]) ] Números de precisión y escala fijas. Cuando se utiliza la precisión máxima, los valores válidos se sitúan entre  $-10^{38} + 1$  y  $10^{38} - 1$ . numérico es funcionalmente equivalente a decimal.

p (precisión): El número total máximo de dígitos decimales que almacenará, tanto a la izquierda como a la derecha del separador decimal. La precisión debe ser un valor comprendido entre 1 y la precisión

máxima de 38. La precisión predeterminada es 18.

s (escala): El número de dígitos decimales que se almacenará a la derecha del separador decimal. Este número se resta de p para determinar el número máximo de dígitos a la izquierda del separador decimal. El número máximo de dígitos decimales que se puede almacenar a la derecha del separador decimal. Escala debe ser un valor comprendido entre 0 y p. Solo es posible especificar la escala si se ha especificado la precisión. La escala predeterminada es 0.

- Caracteres: Son tipos de datos de cadena de longitud fija o de longitud variable.

char [ ( n ) ] Datos de cadena no Unicode de longitud fija. n define la longitud de cadena y debe ser un valor entre 1 y 8.000. El tamaño de almacenamiento es n bytes.

varchar [ ( n | max ) ] Datos de cadena no Unicode de longitud variable. n define la longitud de cadena y puede ser un valor entre 1 y 8.000. max indica que el tamaño máximo de almacenamiento es  $2^{31} - 1$  bytes (2 GB). El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes.

## 4. Actividad

Deberán crear las instrucciones DDL para crear el esquema de bases de datos utilizando como base el diagrama relacional que realizaron en la práctica pasada.

La práctica se dividirá en dos partes. Para esta primera entrega deberán crear todo lo relacionado con los choferes, dueños, vehículos y viajes. Deben tener cuidado de utilizar el correcto tipo de dato para cada una de sus columnas.

Deben especificar las llaves foráneas, compuestas y primarias de cada tabla; así como las restricciones con las que puede contar la tabla.

## 5. Entregables

Se debe crear un script DDL.sql el cual contendrá las instrucciones para la creación de la base de datos, sus tablas y sus restricciones.

Como todo se debe hacer por línea de comando, el archivo README.txt debe tener una explicación muy detallada de como debo correr su script, si al

seguir las instrucciones no es posible crear la base de datos no se calificará la práctica.

La entrega deberá ser el día lunes 20 de Marzo de 2017.