**Material de Recursos Adicionales**

Asignatura de Ingeniería de Datos

SQL Server

**Contenido**

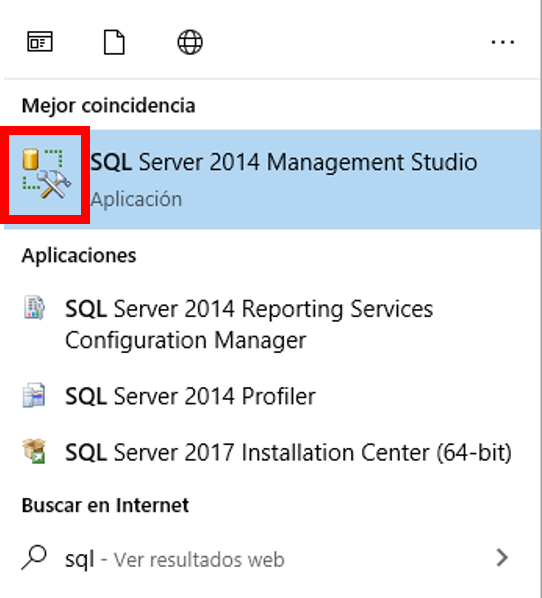
1. **Introducción a SQL**
   1. Abrir Microsoft SQL Server
   2. Crear Diagramas de Bases de Datos
2. **DDL: Data Definition Language**
   1. Crear y Eliminar Base de Datos
   2. Crear Tablas
   3. Modificar y Eliminar Tablas
   4. Generar Diagramas de la Base de Datos
3. **DML: Data Manipulation Language**
   1. Insertar Datos
   2. Select y Select Distinct
   3. Update
   4. Delete
4. **Consultas Condicionales**
   1. Join (From)
   2. Where y Join (Where)
   3. Operadores Lógicos (And, Or, Not)
   4. Operadores Numéricos
   5. Operadores de Relación (Between, Like, In, >, <, >=, <=)
5. **Consultas Básicas**
   1. Order By
   2. All
   3. Top
6. Consultas Anidadas
7. **Consultas Agrupadas**
   1. Group By
   2. Having
8. **Funciones de Librería**
   1. Funciones de Cadenas
   2. Funciones Matemáticas
   3. Funciones de Fecha

Anexo

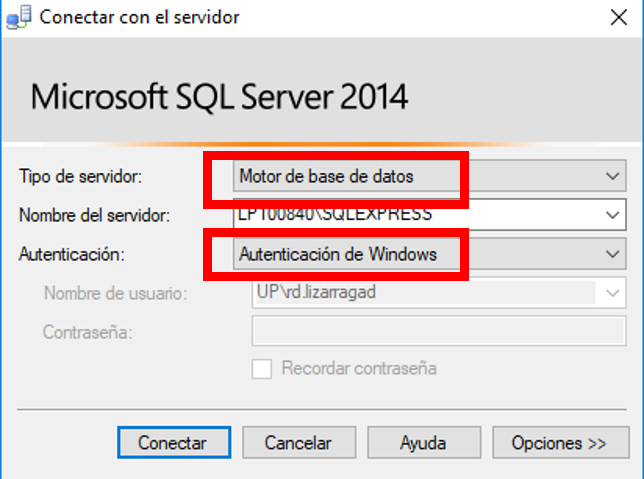
**Introducción a SQL**

**1. Abrir Microsoft SQL Server**

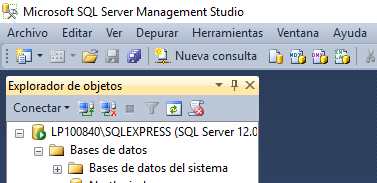
* En el buscador ingresamos **“SQL Server 2014 Management Studio”**



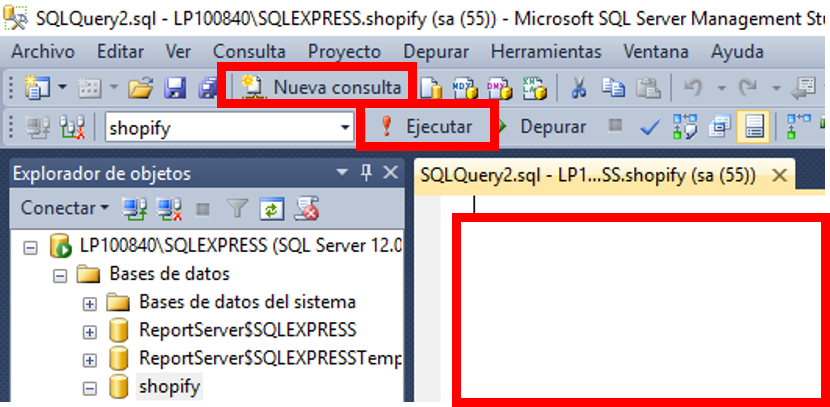
* Verificamos que en la ventana emergente estén seleccionadas las opciones **“Motor de base de datos”** y **“Autenticación de Windows”.** No debemos ingresar ni usuario ni clave. Damos click en **“Conectar”.**



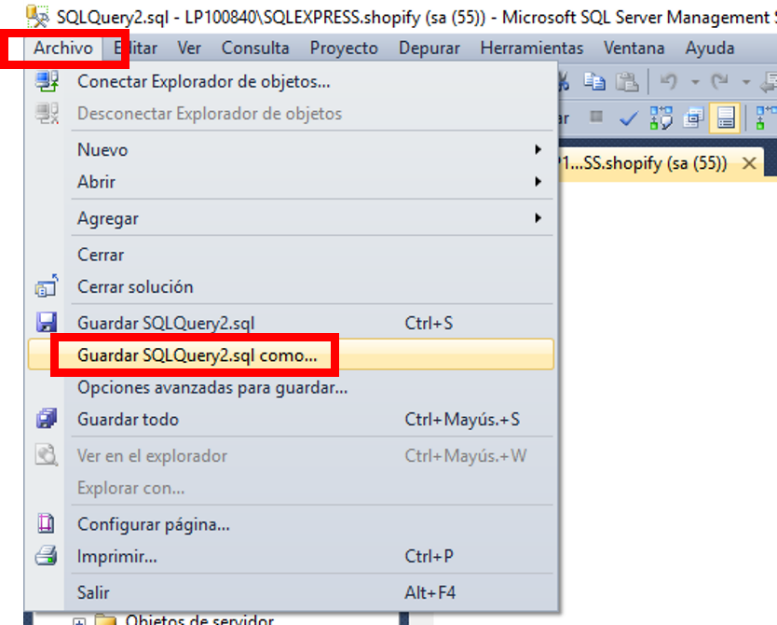
* Si procedimos de manera correcta, debería verse de la siguiente manera. Si detectamos algún problema, pasar al acápite ***“En caso de error”,*** al final de esta sección.



* De ahora en adelante, cada vez que trabajemos en SQL, debemos abrir el espacio de trabajo **“Nueva consulta”**. Cuando queramos ejecutar las sentencias, damos click a **“Ejecutar”.** ***Nota: Este botón ejecuta todas las sentencias, para ejecutar solo una parte de las sentencias se deberá seleccionar las sentencias debidas.***

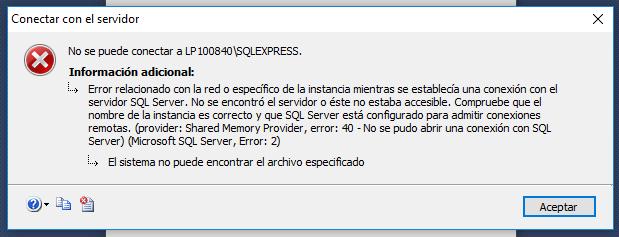


* Para guardar las consultas, nos dirigimos a ***Archivo*** *>* ***Guardar [nombre].sql como…***

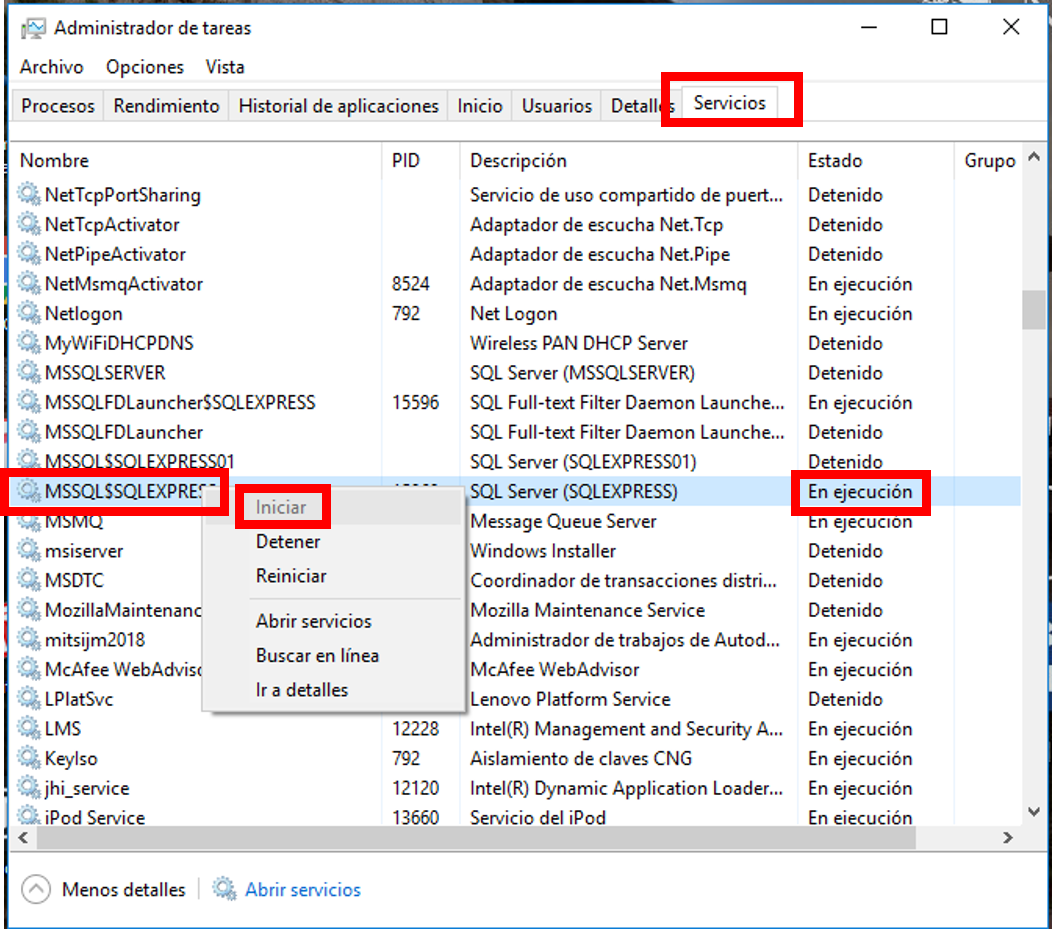


***En caso de error:***

* *Puede que se haya procedido de manera correcta y aun así aparezca un mensaje de error como el siguiente.*

**

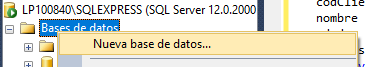
* *Para solucionarlo, nos dirigimos al* ***Administrador de Tareas (Ctrl+Alt+Supr)****, vamos a la pestaña* ***“Servicios”****, damos click derecho en MSSQL$SQLEXPRESS o el nombre con el que no se pudo ingresar, damos click a* ***“Iniciar”.*** *Una vez que el estado sea* ***“En ejecución”,*** *abrimos de nuevo SQL, siguiendo los pasos anteriores.*

**

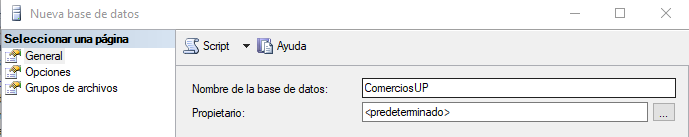
**Introducción a SQL**

**2. Crear Diagramas de Bases de Datos**

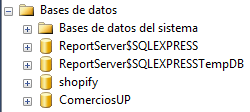
* Damos click derecho en **“Base de Datos”**, seleccionamos **“Nueva base de datos…”**



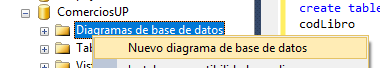
* Asignamos un nombre a la base de datos, damos click a **“Aceptar”.**



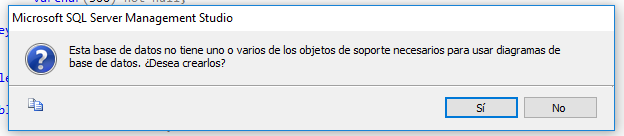
* La base de datos se actualiza automáticamente, y muestra la base de datos creada. De no ser así, damos click derecho en **“Base de datos”** y seleccionamos **“Actualizar”.**



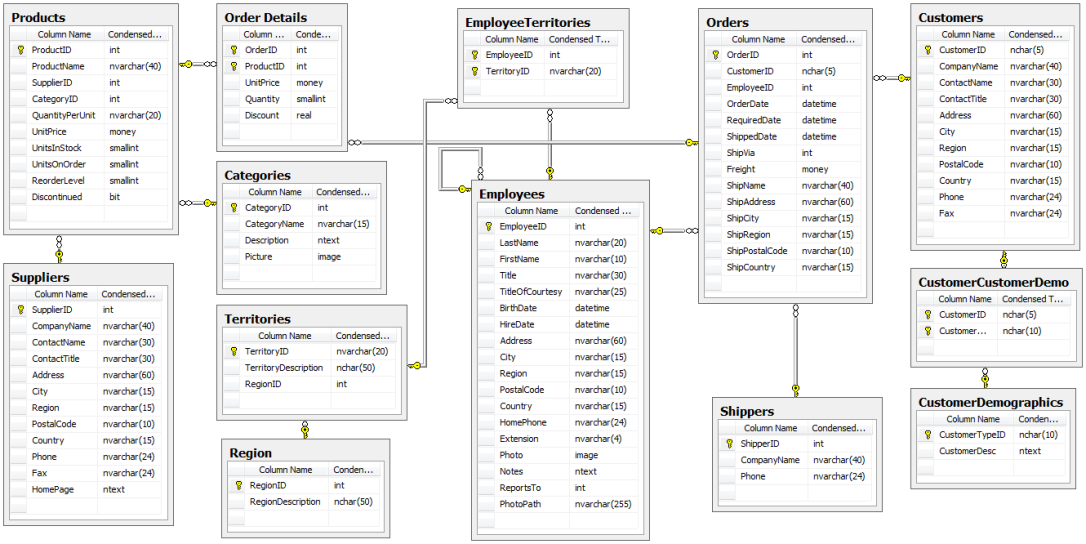
* Nos ubicamos en nuestra base de datos, expandimos y damos click derecho en **“Diagramas de base de datos”**, seleccionamos **“Nuevo diagrama de base de datos”.**



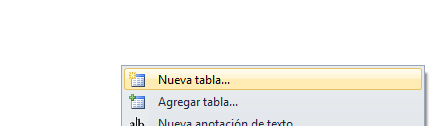
* Nos aparecerá una ventana emergente de consulta, aceptamos. Posteriormente nos aparecerá un recuadro **“Agregar tabla”,** no necesitamos modificar nada allí, lo cerramos. Comenzamos a trabajar en el espacio en blanco.



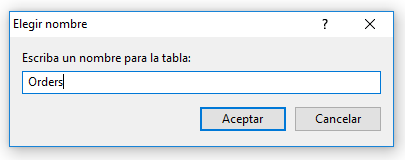
* **Caso:** Replicaremos el diagrama de la base de datos de Northwind, que requiere de 13 tablas *(Employees, EmployeeTerritories, Territories, Region, Orders, Shippers, Customers, CustomerCustomerDemo, CustomerDemographics, OrderDetails, Products, Categories, Suppliers),* y presenta relaciones 1:M, 1:M, N:M, entre otros tópicos aprendidos en clase.

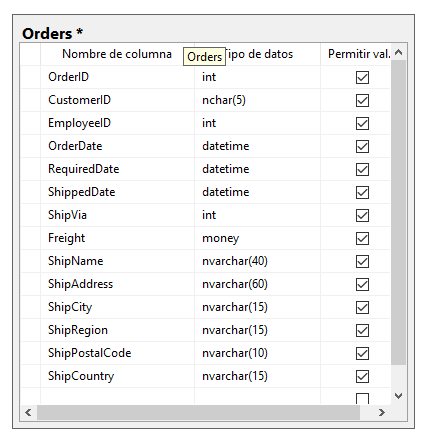


* En el espacio de trabajo, dar click derecho y seleccionar **“Nueva tabla…”**

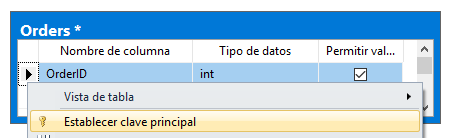


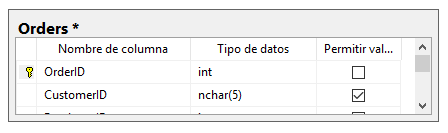
* Comenzamos creando la tabla Orders, para lo cual ingresamos el nombre y aceptamos. Luego, completamos con todos los atributos y con los tipos de datos respectivos. ***Nota: Los tipos serán útiles para las sentencias (punto 4), para entenerlos ver Anexo 1.***



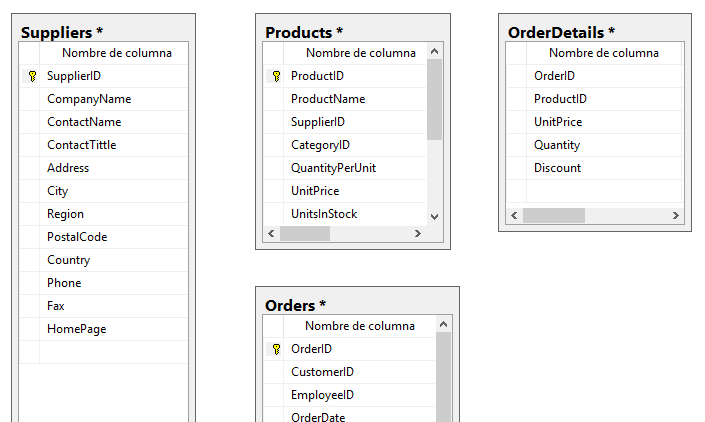


* Para asignar la clave principal, nos ubicamos en el atributo respectivo y damos click derecho. Seleccionamos **“Establecer clave principal”.** Luego, verificamos que haya el símbolo de una llave dorada vertical.

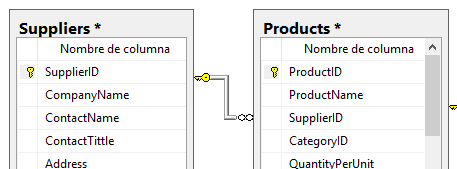




* Procedemos de la misma manera que en los 3 puntos anteriores, y creamos las tablas Suppliers, Products y Order Details.



* Para asignar una clave foránea, damos click sobre la llave (símbolo) de la clave principal de la tabla respectiva y arrastramos sin soltar hasta la clave foránea de la tabla respectiva. En este caso, arrastraremos desde **“SupplierID”** (clave principal) de la tabla **“Suppliers”** hasta **“SupplierID”** (clave foránea) de la tabla Products. Debe quedar como en la imagen, será necesario corroborar los pasos que se encuentran a continuación:

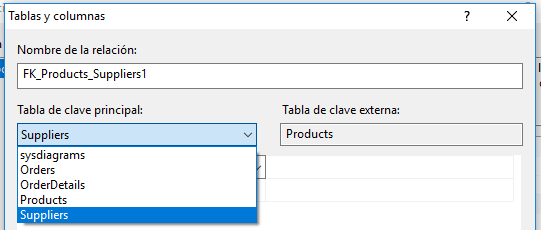


c

c

c

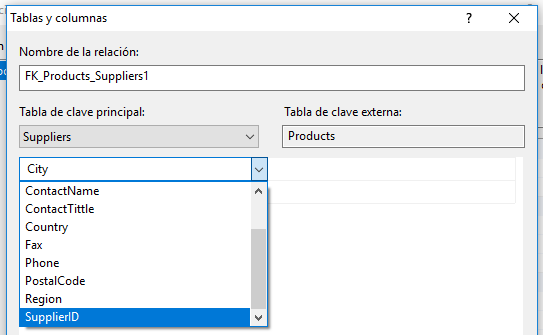
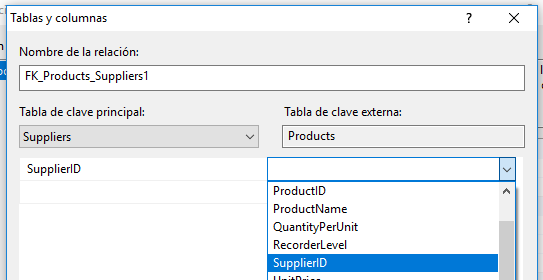
* **Paso 1:** Escoger las tablas que serán relacionadas, tanto la que contiene la clave como principal y la que la contiene como foránea.



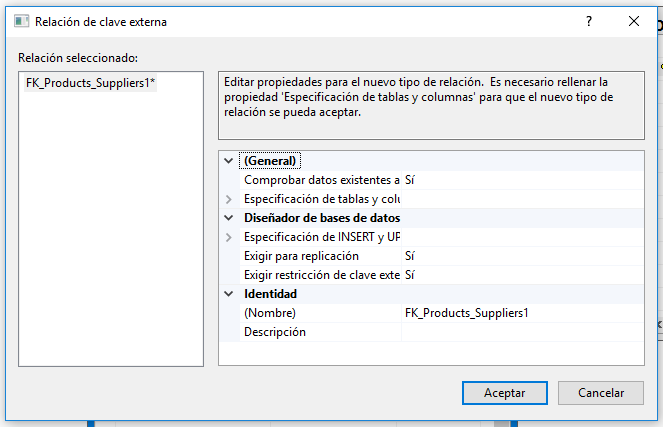
c

c

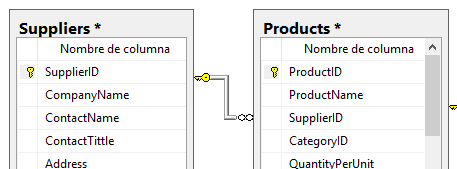
* **Paso 2:** Escoger las claves principal y foránea, según corresponda la tabla.

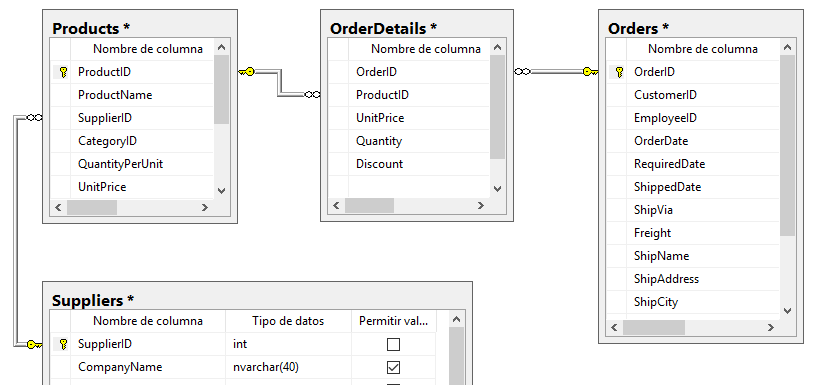
 

* **Paso 3:** Con los 2 pasos anteriores ya creamos la relación, solo nos queda dar click en **“Aceptar”** en la ventana de confirmación de la relación creada.

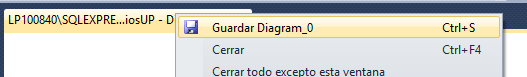


* El resultado final debe ser el de la imagen. Notemos que al final de la relación tenemos 2 símbolos, una llave y un infinito, ellos simbolizan la relación 1:M. Después, creamos las relaciones restantes.

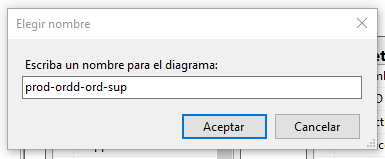




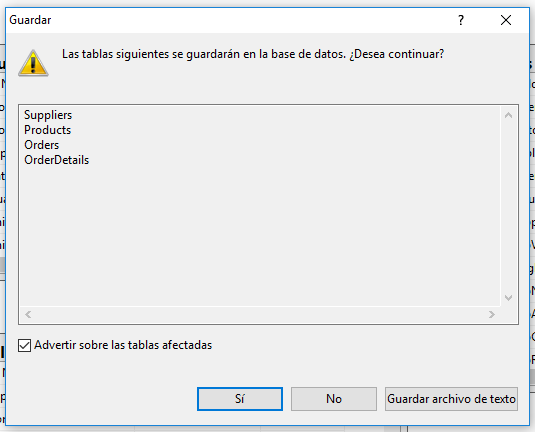
* Con las tablas de nuestra base de datos completas, ya se encuentra lista para guardarla en SQL. Nos posicionamos sobre el nombre de nuestra hoja de trabajo y damos click derecho, seleccionamos **“Guardar…”**.



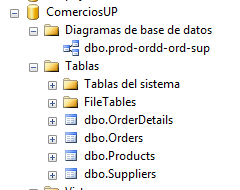
* Escogemos un nombre que sugiera la relación del diagrama, así cuando tengamos varios nos podremos dirigir fácilmente a él. Aceptamos.



* Damos click en **“Sí”**, para que se procedan a guardar las tablas que creamos dentro de nuestra base de datos.



* Nos dirigimos a nuestra base de datos, en este caso es **“ComerciosUP”.** Desplegamos la carpeta **“Diagramas de base de datos”.** Encontraremos nuestro diagrama con el nombre que escogimos en el punto anterior. Ahora, desplegamos la carpeta **“Tablas”** y verificamos que se hayan creado las tablas del diagrama.

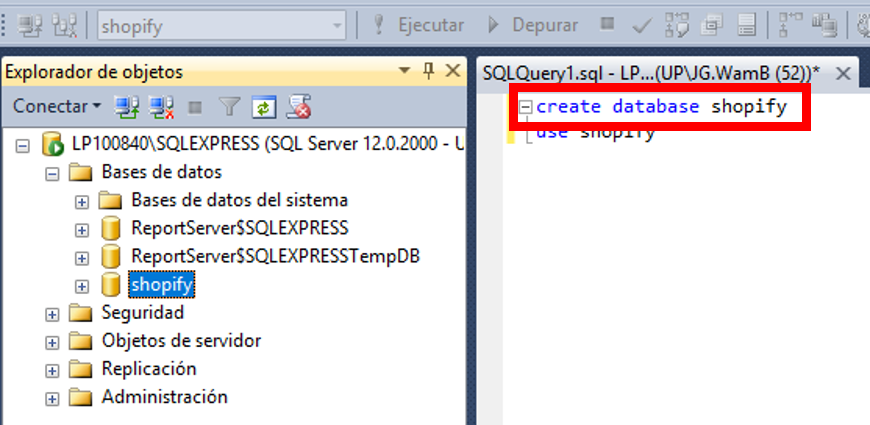


**DDL: Data Definition Language**

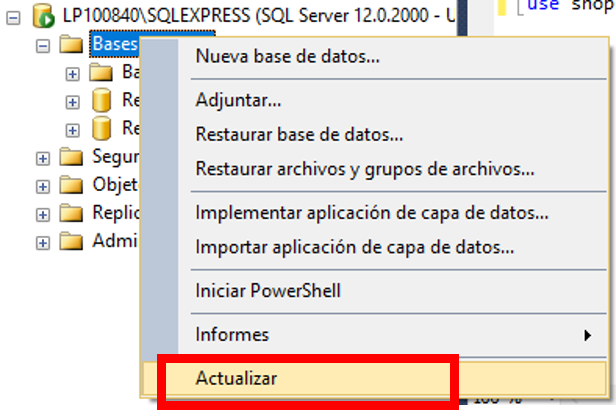
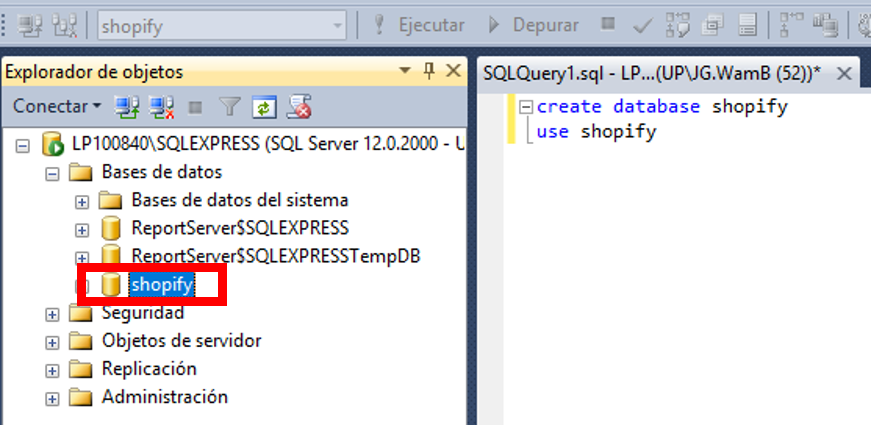
Sirve para definir las estructuras que almacenarán los datos así como los procedimientos o funciones que permitan consultarlos: **create**, **alter** y **drop**.

**3. Crear y Eliminar Bases de Datos**

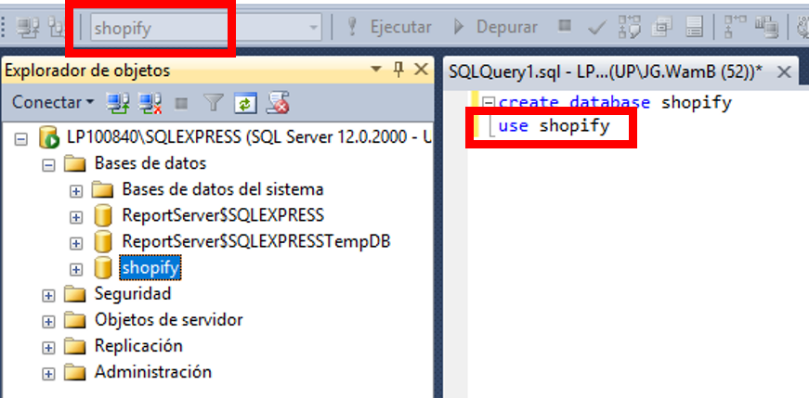
* Ingresamos **create database [nombre de la base de datos]** y ejecutamos.



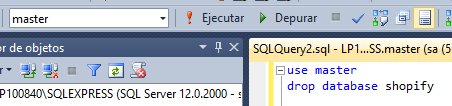
* Para actualizar la base de datos, damos click derecho en **“Base de Datos”** y click a **“Actualizar”.** Ahora podemos visualizar la nueva base de datos.

* Para poder realizar modificaciones en la base de datos, se debe trabajar dentro de ella. Ingresar la sentencia **use [nombre de la base de datos]** y **ejecutamos**. Para confirmar, verificar que en la parte superior se encuentra el nombre de la base de datos.



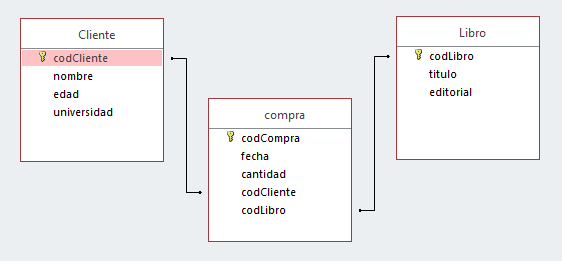
* De haber algún inconveniente y de ser necesario eliminar la base de datos hacemos lo siguiente: Primero, nos ubicamos en otra base de datos con el comando anterior (por lo general **use master**). Segundo verificamos que ya salimos. Tercero, digitamos la sentencia **drop database [nombre de la base de datos]**. Por último, actualizamos **“Bases de datos”** como en el paso 2, donde ya no deberíamos visualizarla.



**DDL: Data Definition Language**

**4. Crear Tablas**

* **Caso:** La librería UP tiene las tablas mostradas debajo. Se sabe que un cliente puede comprar varios libros, y un libro puede ser comprado por varios clientes, por lo que la relación es de muchos a muchos entre ambas tablas (N:M). La tabla entre ambas es la de compra, la cual nos servirá para poder mostrar la relación e ingresar los datos.



* **Tablas Entidades:**

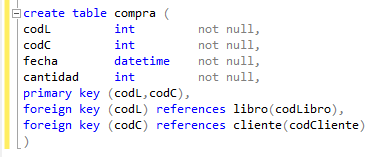
En una nueva consulta ingresamos las siguientes sentencias, para crear las tablas que necesitaremos.

|  |  |
| --- | --- |
|  | **create table [nombreTabla]**  **[atributo][tipo] null/not null** Las sentencias internas contienen:   * Nombre del atributo * El tipo de dato ***(Nota: Ir al Anexo 1 para conocer todos los tipos de datos)*** * *null*: Se permite valor nulo * *not null*: No se permite valor nulo   **primary key [clavePrimaria]**  Después de haber ingresado todos los atributos, se define la clave primaria. |

* **Tabla Relación:**

La clave primaria de la relación se originará de ambas claves primarias de las tablas. Para que se pueda establecer la relación, las claves foráneas también son atributos.

**foreign key ([claveForánea])** (de la tabla relación) **references [tabla]([clavePrimaria])** (de la tabla entidad)

****

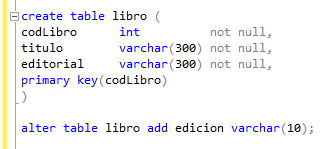
**DDL: Data Definition Language**

**5. Modificar y Eliminar Tablas**

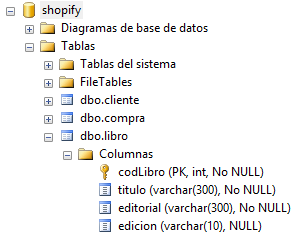
* Del caso anterior, sabemos que algunos libros son actualizados cada cierto período y se publican nuevas ediciones. Cada vez que requeramos añadir un nuevo atributo a una tabla, usaremos el comando:

**alter table [nombreTabla] add [nombreAtributo] [tipoDato];**

Añadiremos esta sentencia para agregar el atributo **“edición”** fuera de las sentencias de la tabla **“Libro”.**



* Más adelante conoceremos un comando que emplearemos todo el tiempo, solo por esta ocasión verificaremos de la siguiente manera que se agregó exitosamente el atributo: Nos dirigimos a **“Shopify” > Tablas > dbo.libro** (o el nombre de tabla asignado) **> Columnas**. Al final de esta ruta está el nuevo atributo.



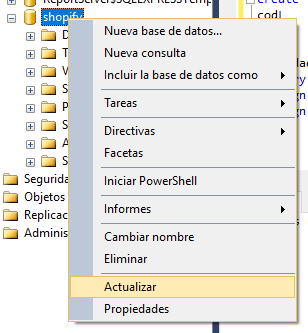
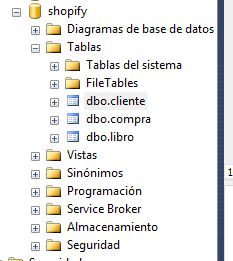
* De haber algún inconveniente y de ser necesario eliminar alguna de las tablas, digitamos el comando **drop table [nombreTabla];.** Supongamos que creamos la tabla **“Librería”,** entonces procederemos a eliminarla.

|  |  |
| --- | --- |
|  |  |

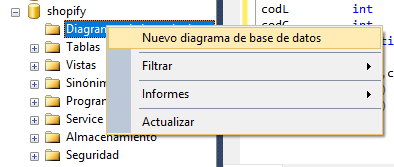
**DDL: Data Definition Language**

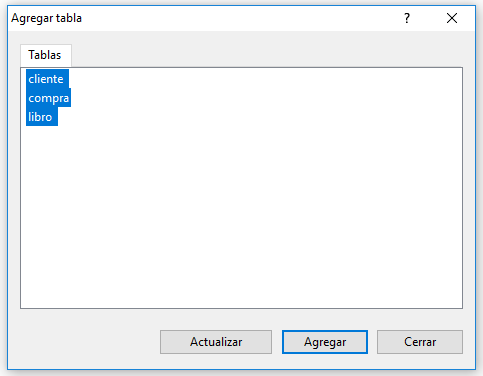
**6. Generar Diagramas de la Base de Datos**

* Actualicemos la base de datos. Damos click derecho en **“Base de Datos”,** damos click a **“Actualizar”.** Abrimos las **“Tablas”.** Ahora podemos visualizar las tablas creadas.

** **

* Si los atributos como las llaves fueron descritos correctamente, al generar el diagrama de las bases de datos, las relaciones se generarán automáticamente. Dentro de nuestra base de datos, damos click derecho en **“Diagrama de base de datos”**, luego a **“Nuevo diagrama de base de datos”.** En la ventana emergente, seleccionamos todas las tablas creadas para agregarlas con un solo click (esto será más útil cuando tengamos muchas tablas).

****

****

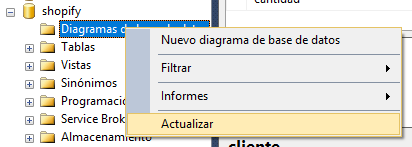
* Si deseamos agregar todas las tablas, tendremos la ventana emergente vacía; si deseamos agregar solo algunas, verifiquemos que fuera se hallen las que requerimos. En este caso, agregaremos todas. Al cerrar, obtendremos el diagrama con las relaciones que indicamos en las sentencias. ***Nota: Si no logramos el objetivo, entonces eliminamos la hoja y corregimos las llaves de las sentencias en la consulta.***

|  |  |
| --- | --- |
|  |  |

* Es muy importante tener diagramas de cada relación y el diagrama general de la base de datos. Para guardarlos, damos click derecho en la hoja donde se ubica nuestro diagrama, damos click a **“Guardar…”**. Escogemos un nombre que sugiera la relación del diagrama, así cuando tengamos varios nos podremos dirigir fácilmente a él.

|  |  |
| --- | --- |
|  |  |

* Actualicemos **“Diagramas de bases de datos”** dando click derecho y click a **“Actualizar”.** Esperamos a que nos aparezca el botón **“+”** de expansión y abrimos **“Diagrama de base de datos”,** donde se hallará guardado el diagrama de la relación **“cliente-compra-libro”.**

****

****

**DML: Data Manipulation Language**

Permite introducir datos para realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos: **insert**, **select**, **update** y **delete**.

**1. Insertar valores**

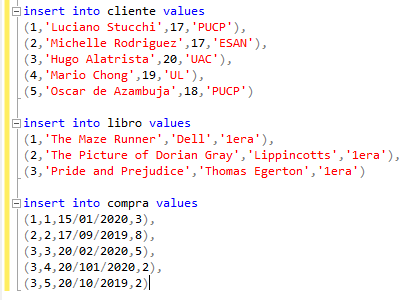
* Los datos que ingresemos deben mantener la estructura en que fueron ordenados los atributos (en las sentencias de las tablas), y deben obedecer a los tipos de datos indicados. La sentencia correspondiente para añadir una tupla es la que sigue:

**insert into [nombreTabla] values (valor1,valor2,valor3,…);**

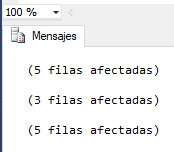
* Si requerimos añadir varias tuplas, podemos repetir la misma sentencia.



* También podemos separar una tupla de otra con comas:

****

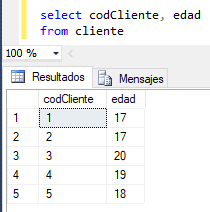
* Si se efectuó con éxito entonces veremos este mensaje en la consola:

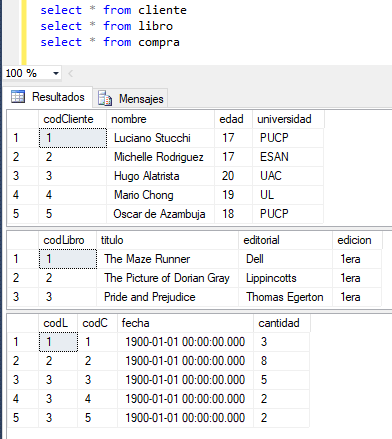
****

**DML: Data Manipulation Language**

**2. Select**

* Para mostrar todos los registros, con datos que corresponden a una columna o varias, de una determinada tabla: **select [nombreColumna1], [nombreColumna2], …**

**from [nombreTabla]  
  
  
  
select [\* = todas las columnas]  
from [nombre Tabla]**

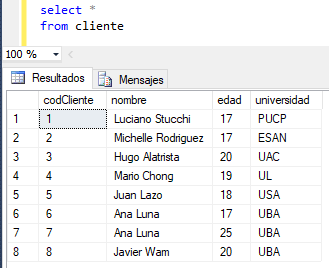
****

**2. Select Distinct**

* Respecto al caso, si quisiéramos saber cuáles son todas las universidades cuyos alumnos han comprado en la librería, necesitamos quitar duplicados de los registros, pues varios alumnos pueden provenir de la misma universidad. A continuación, agregaremos 3 registros más, cuyos estudiantes sean de la ‘UBA’.

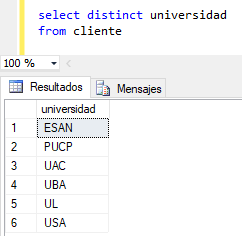


* Verificamos que se hayan agregado los nuevos registrados son **select.**



* Para mostrar todos los registros sin duplicados, con datos que corresponden a una columna o varias, de una determinada tabla: **select distinct [nombreColumna1], [nombreColumna2], …**

**from [nombreTabla]**



**DML: Data Manipulation Language**

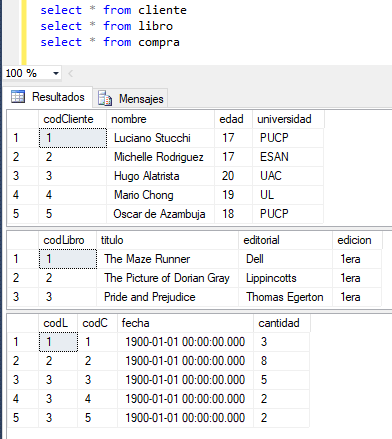
**3. Update**

* Para modificar datos ya existentes en una tabla: **update [nombreTabla],**

**set [nombreAtributo1]=[valor1], [nombreAtributo2]=[valor2], …**

**where [condición]**

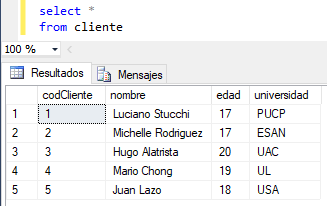
* En este caso, modificaremos el registro 5.

****

* Modificaremos el nombre, la edad y la universidad del codCliente 5, y ejecutamos.



* Consultamos con **select** los registros de la tabla **cliente**, y vemos que se actualizó exitosamente.



**DML: Data Manipulation Language**

**4. Delete**

Para modificar datos ya

**Anexo**

**Anexo 1**

**SQL Server Data Types**

*(Recuperado de* [*https://www.w3schools.com/sql/sql\_datatypes.asp*](https://www.w3schools.com/sql/sql_datatypes.asp)*)*

* **String Data Types**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| CHAR(size) | A FIXED length string (can contain letters, numbers, and special characters). The *size* parameter specifies the column length in characters - can be from 0 to 255. Default is 1 |
| VARCHAR(size) | A VARIABLE length string (can contain letters, numbers, and special characters). The *size* parameter specifies the maximum column length in characters - can be from 0 to 65535 |
| BINARY(size) | Equal to CHAR(), but stores binary byte strings. The *size* parameter specifies the column length in bytes. Default is 1 |
| VARBINARY(size) | Equal to VARCHAR(), but stores binary byte strings. The *size* parameter specifies the maximum column length in bytes. |
| TINYBLOB | For BLOBs (Binary Large OBjects). Max length: 255 bytes |
| TINYTEXT | Holds a string with a maximum length of 255 characters |
| TEXT(size) | Holds a string with a maximum length of 65,535 bytes |
| BLOB(size) | For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data |
| MEDIUMTEXT | Holds a string with a maximum length of 16,777,215 characters |
| MEDIUMBLOB | For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data |
| LONGTEXT | Holds a string with a maximum length of 4,294,967,295 characters |
| LONGBLOB | For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data |
| ENUM(val1, val2, val3, ...) | A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them |
| SET(val1, val2, val3, ...) | A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list |

* **Numeric Data Types**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| BIT(*size*) | A bit-value type. The number of bits per value is specified in *size*. The *size* parameter can hold a value from 1 to 64. The default value for *size* is 1. |
| TINYINT(*size*) | A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The *size* parameter specifies the maximum display width (which is 255) |
| BOOL | Zero is considered as false, nonzero values are considered as true. |
| BOOLEAN | Equal to BOOL |
| SMALLINT(*size*) | A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The *size* parameter specifies the maximum display width (which is 255) |
| MEDIUMINT(*size*) | A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The *size* parameter specifies the maximum display width (which is 255) |
| INT(*size*) | A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The *size* parameter specifies the maximum display width (which is 255) |
| INTEGER(*size*) | Equal to INT(size) |
| BIGINT(*size*) | A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The *size* parameter specifies the maximum display width (which is 255) |
| FLOAT(*size*, *d*) | A floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions |
| FLOAT(*p*) | A floating point number. MySQL uses the *p* value to determine whether to use FLOAT or DOUBLE for the resulting data type. If *p* is from 0 to 24, the data type becomes FLOAT(). If *p* is from 25 to 53, the data type becomes DOUBLE() |
| DOUBLE(*size*, *d*) | A normal-size floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter |
| DOUBLE PRECISION(*size*, *d*) |  |
| DECIMAL(*size*, *d*) | An exact fixed-point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. The maximum number for *size* is 65. The maximum number for *d* is 30. The default value for *size* is 10. The default value for *d* is 0. |
| DEC(*size*, *d*) | Equal to DECIMAL(size,d) |

**Note:** All the numeric data types may have an extra option: UNSIGNED or ZEROFILL. If you add the UNSIGNED option, MySQL disallows negative values for the column. If you add the ZEROFILL option, MySQL automatically also adds the UNSIGNED attribute to the column.

* **Date and Time Data Types**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| DATE | A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31' |
| DATETIME(*fsp*) | A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time |
| TIMESTAMP(*fsp*) | A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT\_TIMESTAMP and ON UPDATE CURRENT\_TIMESTAMP in the column definition |
| TIME(*fsp*) | A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59' |
| YEAR | A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format. |