



Práctica 06: Introducción a SQL

Fundamentos de Bases de Datos

Integrantes:

- Aceves Higareda Mayra Guadalupe
ahigareda.mayra@ciencias.unam.mx
310646246
- Badillo Lora Carlos
carlosb@ciencias.unam.mx
415083504
- Moreno Ruíz Jesús Fernando
chuchini@ciencias.unam.mx
414001967
- Servín Mote Edson
edson@ciencias.unam.mx
308292503

Profesor: Gerardo Avilés Rosas

Laboratorio: Erick Orlando Matla Cruz

1. Tablas de la base de datos “pastoreenchilado”

Se presenta una lista de las 26 tablas que corresponden al diagrama del caso de uso del “Pastor Enchilado”. Cada tabla tiene una lista con sus atributos y al final se especifica el tiempo invertido en la creación de cada tabla (el formato del tiempo está en hh:mm:ss).

1. Promocion:

- **nIdPromocion** → INT IDENTITY, este entero funciona como llave primaria e identifica una promoción de manera única.
- **sPromocion** → VARCHAR(32), hicimos un cambio de integer(10) que es como estaba especificado en el diagrama, ya que el atributo indica el nombre de la de promoción, y en todo caso nIdPromocion ya lleva el registro del número de promoción que estamos usando.
- **sDia** → VARCHAR(10), este atributo indica únicamente el día en que se llevará a cabo una promoción (un día entre lunes a domingo), por lo que podemos reducirlo de 32 a 10.
- **nIdSucursalProducto** → INT, este atributo pertenece a la tabla Sucursal_Producto, por lo que mantenemos el mismo tipo de dato.



Tiempo: 00:05:40

2. Proveedor:

- **nIdProveedor** → INT IDENTITY, este entero funciona como llave primaria e identifica a un proveedor de manera única.
- **sProveedor** → VARCHAR(32), este atributo sirve para almacenar el nombre del proveedor.
- **sRFC** → CHAR(13), queremos que el RFC sea exactamente de esa longitud y con caracteres alfanuméricos, por lo que usamos el tipo de dato “char”.
- **nIdDireccion** → INT, este atributo pertenece a la tabla Direccion, por lo que mantenemos el mismo tipo de dato.



Tiempo: 00:05:55

3. Proveedor_Ingrediente:

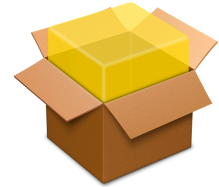
- **nIdProveedor** → INT, este atributo es una llave foránea, pertenece a la tabla Proveedor, por lo que mantenemos el mismo tipo de dato.
- **nIdIngrediente** → INT, este atributo es una llave foránea, pertenece a la tabla Ingrediente, por lo que mantenemos el mismo tipo de dato.
- **dFecha** → DATE, este atributo indica la fecha exacta (AAAA-MM-DD) en la que un proveedor surte un ingrediente.
- **dCosto** → MONEY, este atributo indica el costo de los productos que surte el proveedor. Cambiamos el tipo de dato double(10) en el diagrama por “money”, ya que queremos trabajar con datos monetarios.

-
-
- **nCantidad** → INT, este atributo indica la cantidad de productos que va a surtir el proveedor, por lo que el tipo debe ser entero.
 - **nIdSucursal** → INT, este atributo pertenece a la tabla Sucursal, por lo que mantenemos el mismo tipo de dato.
 - **sNombre** → VARCHAR(16), este atributo pertenece a la tabla Sucursal, por lo que mantenemos el mismo tipo de dato.

Tiempo: 00:09:35

4. Producto:

- **nIdProducto** → INT IDENTITY, este entero funciona como llave primaria e identifica un producto de manera única.
- **sProducto** → VARCHAR(64), el atributo indica el nombre del producto. Lo dejamos de ese tamaño, ya que consideramos que es suficiente.



Tiempo: 00:02:10

5. Ingrediente_Producto:

- **nIdIngrediente** → INT, este atributo es una llave foránea, pertenece a la tabla Ingrediente, por lo que mantenemos el mismo tipo de dato.
- **nIdProducto** → INT, este atributo es una llave foránea, pertenece a la tabla Producto, por lo que mantenemos el mismo tipo de dato.
- **sPorcion** → VARCHAR(64), el atributo indica la información respecto a las porciones que se manejan, por lo que conservamos el tipo de dato "varchar".

Tiempo: 00:03:29

6. Ingrediente:

- **nIdIngrediente** → INT IDENTITY, este entero funciona como llave primaria e identifica un ingrediente de manera única.
- **sIngrediente** → VARCHAR(64), el atributo indica la información sobre el nombre de un ingrediente. Conservamos el tipo de dato "varchar".



Tiempo: 00:02:00

7. Cliente:

- **nIdCliente** → INT IDENTITY, es el atributo que identificará de manera única a la tabla "Cliente". Es de tipo INT ya que actúa como "llave primaria" y así es especificado en el diagrama.
- **sMonedero** → VARCHAR(32), es el atributo utilizado para guardar el número de la tarjeta monedero, por lo tanto es del tipo varchar.



-
-
- **sSaldo** → FLOAT(10), identifica el saldo del monedero. Es de tipo float ya que es un valor numérico aproximado.
 - **nldPersona** → INT, actúa como una referencia a la tabla Persona, es de tipo INT ya que debe de llevar el mismo valor utilizado por el atributo “nldPersona” que es la “llave primaria” de la tabla antes mencionada.

Tiempo: 01:44:00

8. Direccion_Persona:

- **nldDireccion** → INT, es el atributo que hace referencia a la tabla “Direccion”, por lo tanto debe ser del mismo tipo.
- **nldPersona** → INT, es el atributo que hace referencia a la tabla “Persona”, por lo tanto debe ser del mismo tipo.

Tiempo: 00:02:43

9. Persona_Telefono:

- **nldPersona** → INT, es el atributo que hace referencia a la tabla “Persona”, igual que en los casos anteriores debe de ser del mismo tipo.
- **nldTelefono** → INT, hace referencia a la tabla “Telefono”. Es INT para que corresponda al mismo tipo de dato que **nldTelefono** que es el atributo que funge como “llave primaria” de dicha tabla.
- **nldTipoTelefono** → INT hace referencia a la tabla “CTipoTelefono”. Es INT por la misma razón que ya se dijo antes.

Tiempo: 00:01:35

10. Sucursal:

- **nldSucursal** → INT IDENTITY, es el atributo que servirá para identificar de manera única a tal tabla. Es INT para que sea un entero como es referido en el esquema.
- **sNombre** → VARCHAR(16), es el atributo que se utiliza para almacenar el nombre de la sucursal y es del tipo descrito antes porque almacena caracteres de longitud 16. Hay que aclarar que en la tabla forma parte del conjunto que identifica de manera única a la sucursal, sin embargo se mantiene como una “llave candidata” (aunque no usemos llaves como tal) y en nuestro caso elegimos **nldSucursal** para tal propósito.
- **nldDireccion** → INT hace referencia a la tabla “Direccion” y es del tipo int debido a que debe de cumplir con el mismo tipo de dato al que hace nos generó la secuencia en la referencia (nldDireccion).
- **nldTelefono** → INT hace referencia a la tabla “Telefono” y es del tipo antes mencionado por lo mismo que en el atributo anterior.



Tiempo: 00:02:22

11. CTipoTelefono:

- **nIdTipoTelefono** → INT IDENTITY, es el atributo que sirve para identificar a la tabla. Es de tal tipo debido a que en el esquema es un INTEGER. Representa la “llave primaria” de la tabla.
- **sTipoTelefono** → VARCHAR(32) es el atributo que sirve para saber a qué tipo de teléfono nos referimos (móvil o fijo) y es varchar de longitud 32 para respetar lo mencionado en el esquema.

Tiempo: 00:01:58

12. Empleado:

- **nIdEmpleado** → INT IDENTITY, sirve para identificar de manera única a los empleados, es decir, es la “llave primaria” de la tabla.
- **dFechaContratacion** → DATE, es el atributo que nos indica la fecha en que el empleado fue contratado y es del tipo antes mencionado ya que nos facilita manejar las fechas y horas.
- **sRFC** → VARCHAR(13) representa el RFC del empleado y por lo tanto es un conjunto de caracteres de longitud 13.
- **nIdPersona** → INT hace referencia a la tabla “Persona” y es del tipo INT por las mismas razones que y se han explicado antes para esta clase de atributos.
- **nIdSucursal** → INT hace referencia a la tabla “Sucursal”. Aplica lo mismo que en el atributo anterior.
- **sNombre** → VARCHAR(16) es un atributo que en nuestro esquema forma parte del conjunto que identifica de manera única a la tabla “Sucursal”. Sin embargo lo interpretamos como una “llave candidata”. Lo añadimos para mantener la igualdad con el modelo relacional.



Tiempo: 00:03:58

13. Sucursal_Producto:

- **nIdSucursalProducto** → INT IDENTITY, es el atributo que sirve para identificar de manera única a dicha tabla. Es de tipo int por su descripción en la tabla correspondiente en el diagrama.
- **dPrecio** → FLOAT(10), indica el precio de los productos de la sucursal. Es de tipo float ya que se trata de un número aproximado.
- **dFechaActualizacion** → DATE, indica la fecha en que un producto fue actualizado y se utiliza el tipo de dato date para facilitar el manejo de las fechas.
- **bActivo** → BIT, usa el tipo de dato anterior para saber si el atributo es falso o verdadero (0 ó 1 respectivamente).
- **nIdSucursal** → INT, hace referencia a la tabla “Sucursal” y aplican las mismas reglas que ya hemos mencionado.
- **nIdProducto** → INT, hace referencia a la tabla “Producto” y sucede lo mismo que con el atributo anterior.
- **sNombre** → VARCHAR(16), es un atributo que forma parte del conjunto que identifica de manera única la tabla “Sucursal” y, debido a que no podemos

hacer uso de IDENTITY más de una vez en las tablas lo tomamos como una “llave candidata” y entonces sólo respetamos la referencia y el mismo tipo de dato que conlleva.

Tiempo: 00:03:30

14. Horario:

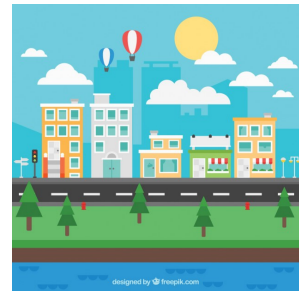
- **nIdHorario** → INT IDENTITY, es el atributo que sirve para identificar de manera única a la tabla horario, por lo que su tipo será un entero.
- **sDia** → VARCHAR(16), dado que es un atributo que guarda el nombre del día de la semana se usa como tipo cadena.
- **tHoraApertura** → TIME(7), el tipo de este atributo es time, el cual tiene un formato que sirve para almacenar una hora de apertura específica.
- **tHoraCierre** → TIME(7), de la misma manera que el atributo anterior el tipo requerido en este atributo es time, para almacenar la hora de cierre del horario.
- **nIdSucursal** → INT, hace referencia a la tabla sucursal, que sirve para asociar un horario específico a una cierta sucursal y el tipo de dato permanece como entero para su identificación.
- **sNombre** → VARCHAR(16), el nombre como tal es de tipo cadena.



Tiempo: 00:03:47

15. Direccion:

- **nIdDireccion** → INT IDENTITY, es el atributo que identifica cierta dirección, por lo que usaremos un entero como tipo de dato. Identifica de manera única a la tabla “Direccion”.
- **sCalle** → VARCHAR(100), para el nombre de la calle usaremos una cadena como tipo de dato. Reducimos el tamaño de 128 a 100, ya que se considera suficiente.
- **sColonia** → VARCHAR(64), de la misma manera que el atributo anterior se requiere usar un tipo de dato cadena para el atributo colonia.
- **sCp** → INT, para el código postal se requirió cambiar el tipo de dato de cadena a un entero, ya que el código postal requiere sólo de números para su identificación.
- **sObservaciones** → VARCHAR(256), para las observaciones, se necesita como tipo de dato varchar, ya que es una cadena de las observaciones y opiniones.
- **nIdMunicipio** → INT, este atributo hace referencia a la tabla del municipio (es una llave foránea), por lo que prevalece el tipo de dato tal cual.



Tiempo: 00:03:33

16. CMunicipio:

- **nIdMunicipio** → INT IDENTITY, atributo que sirve para identificar de manera única al municipio, se usa de tipo de dato entero para que sea mejor su uso.
- **sMunicipio** → VARCHAR(64), en sí este atributo es el nombre del municipio por lo que su tipo de dato necesita ser una cadena.
- **nIdEstado** → INT, este atributo hace referencia a la tabla estado, por lo que su tipo de dato debe ser del mismo tipo del atributo original.



Tiempo: 00:02:47

17. Estado:

- **nIdEstado** → INT IDENTITY, este atributo es el identificador de la tabla estado por lo que su tipo de dato es un entero para facilitar la manipulación.
- **sEstado** → VARCHAR(30), el nombre del estado necesita ser una cadena por eso este atributo tiene de tipo de dato "varchar". Reducimos el tamaño de 64 a 30, ya que consideramos que es suficiente.



Tiempo: 00:01:05

18. Proveedor_Telefono:

- **nIdProveedor** → INT, atributo que sirve para identificar de manera única al proveedor por lo que lo usamos como entero el tipo de dato. Este atributo es una llave foránea, por lo que mantenemos el mismo tipo de dato.
- **nIdTelefono** → INT, este atributo hace referencia a la tabla "Telefono" por lo que su tipo de dato es un entero para la identificación del mismo. Este atributo es una llave foránea, por lo que mantenemos el mismo tipo de dato.
- **nIdTipoTelefono** → INT, es un atributo el cual también hace referencia a tipo teléfono, por lo que es un entero su tipo de dato.

Tiempo: 00:02:11

19. Teléfono:

- **nIdTelefono** → INT IDENTITY, para el identificador del teléfono se usa un entero por las razones anteriores. Este atributo es una llave primaria, puesto que queremos que sea único.
- **sTelefono** → VARCHAR(16), un número puede estar construido además de por números, por caracteres (como un guión), por lo que usamos cadenas como su tipo de dato.



Tiempo: 00:01:20

20. Persona:

- **nIdPersona** → INT IDENTITY, atributo que identifica a una persona de manera única, por lo que su tipo de dato es un entero para facilitar la manipulación.
- **sApp** → VARCHAR(32), para el apellido paterno se requiere una secuencia de caracteres por lo que su tipo de dato debe ser una cadena.
- **sApm** → VARCHAR(32), para el apellido materno se requiere una secuencia de caracteres por lo que su tipo de dato debe ser una cadena.
- **sNombre** → VARCHAR(64), para el nombre se requiere una secuencia de caracteres por lo que su tipo de dato debe ser una cadena.
- **dFechaNacimiento** → DATE, para la fecha de nacimiento de una persona requiere que su tipo de dato sea date, lo cual facilita su comprensión.
- **sCorreo** → VARCHAR(126), el correo puede estar compuesto por una serie de caracteres por lo que el tipo de dato debe ser una cadena.



Tiempo: 00:03:30

21. Salsa_Producto:

- **nIdSalsa** → INT, este entero es una llave foránea que identifica el tipo de salsa de manera única.
- **nIdProducto** → INT, este atributo es una llave foránea que identifica el producto del menú.

Tiempo: 00:03:45

22. DetalleOrden:

- **nIdDetalleOrden** → INT IDENTITY, este entero funciona como llave primaria e identifica el detalle de una orden de manera única.
- **nCantidad** → INT, el atributo indica la cantidad del producto que se ordenara.
- **bDomicilio** → BIT, este booleano identifica si la orden es para entregar a domicilio o no.
- **nIdSucursalProducto** → INT, este atributo es una llave foránea que identifica la sucursal.
- **nIdOrden** → INT, este atributo es una llave foránea que identifica la orden.
- **nIdCliente** → INT, el entero es una llave foránea que identifica al cliente.



Tiempo: 00:05:51

23. Orden:

- **nldOrden** → INT IDENTITY, este entero funciona como llave primaria e identifica una orden de manera única..
- **dTotal** → FLOAT, este flotante indica el costo total de la orden.
- **sDia** → VARCHAR(10), la cadena muestra el día en el que se hizo la orden. Lo reducimos de tamaño de 16 a 10, ya que lo consideramos suficiente.
- **dfecha** → DATE, el atributo es de tipo fecha e indica la fecha exacta en la que se hizo la orden.
- **nldEmpleado** → INT, el atributo es una llave foránea que identifica al empleado.



Tiempo: 00:04:36

24. Salsa_Ingrediente:

- **nldSalsa** → INT, este atributo es una llave foránea, pertenece a la tabla Salsa, por lo que mantenemos el mismo tipo de dato.
- **nldIngrediente** → INT, este atributo es una llave foránea, pertenece a la tabla Ingrediente, por lo que mantenemos el mismo tipo de dato.
- **sPorcion** → VARCHAR(64) esta cadena indica la porción de la salsa.

Tiempo: 00:02:16

25. Salsa:

- **nldSalsa** → INT IDENTITY, este atributo sirve como llave primaria e identifica una salsa de manera única.
- **sSalsa** → VARCHAR(10), indica el nombre de la salsa.
- **nldPicor** → INT, indica el nivel de picor. Este atributo pertenece a la tabla CPicor, por lo que mantenemos el mismo tipo de dato.



Tiempo: 00:03:42

26. CPicor:

- **nldPicor** → INT, este atributo sirve como llave primaria e identifica el picor de una salsa de manera única.
- **sPicor** → VARCHAR(32), este atributo corresponde al nombre del picante, por lo que decidimos cambiarlo del tipo “integer” al tipo “varchar”.



Tiempo: 00:02:07

Aproximadamente, la creación de las tablas en conjunto nos tomó el siguiente tiempo:

Tiempo Total: 01:27:09

2. NoSQL

En el DBMS MongoDB no se usan tablas. En vez de tablas tenemos “Colecciones”. En las “Colecciones” se guardan “documentos”. A diferencia de las filas en una tabla, los documentos en una colección no necesitan compartir los mismos atributos.

Para crear una colección basta con insertar un documento en la colección. A continuación se muestran 5 ejemplos:

1)

```
db.DetalleOrden.insert({
  nIdDetalleOrden: 432432,
  nCantidad: 3,
  bDomicilio: 0,
  Sucursal_ProductonIdSucursalProducto: 6534654,
  OrdennIdOrden: 231321,
  ClientenIdCliente: 43543243,
})
```

2)

```
db.Orden.insert({
  nIdOrdenn: 4536452,
  dTotal: 43.50,
  sDia: Lunes,
  dFecha: 10-09-1935,
  EmpleadoIdEmpleado: 65433,
})
```

3)

```
db.Proveedor_Ingrediente.insert({
  nIdProveedor: 53432,
  nIdIngrediente: 42450,
  dFecha: 11-11-2032,
  dCosto: 43,
  nCantidad: 33,
  nIdSucursal: 8543,
  sNombre: "Vegetales Selectos"
})
```

4)

```
db.Empleado.insert({
  nIdEmpleado : 7432,
  dFechaContratacion : 12-01-2015,
  sRFC : GAFS02031985,
  nIdPersona : 843234,
  nIdSucursal : 01342,
  nIdSucursal: 8543,
  sNombre : "Salvador Garcia Fuentes"
})
```

5)

```
db.Sucursal_Producto.insert({
  nIdSucursalProducto : 9432,
  dPrecio : 123.01,
  dFechaActualizacion : 01-03-1953,
  bActivo : 1,
  nIdSucursal : 47454,
  nIdProducto : 8753,
  sNombre : "Sucursal La Cruz"
})
```