Juan Pablo Romero Londoño

Juan David Giraldo Marín

Luis Felipe Torres López

Frameworks de Persistencia

Bases de datos SQL y bases de datos No SQL

Taller Final

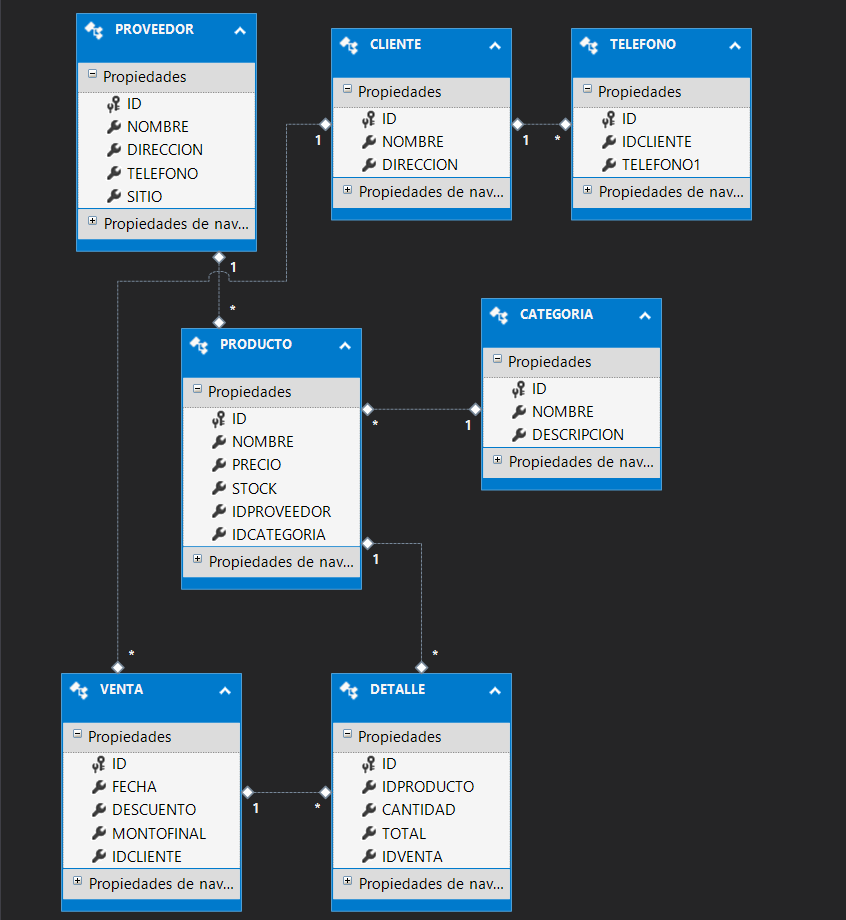
Tópicos Avanzados en Bases de Datos

# EJERCICIOS

1. (30%) Se debe crear el modelo Entidad – Relación para dar solución al problema planteado de ventas (sólo ventas), teniendo en cuenta que todas las tablas del modelo deben estar en tercera forma normal.

Como resultado final de la realización de este punto se requiere:

* Diagrama.



* Scripts de creación de las tablas del modelo y de inserción de datos semilla (maestros).
  + Scripts de creación de base de datos y tablas

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

CREATE DATABASE BDVentasOnline

New query en BDVentasOnline

CREATE TABLE CLIENTE

(

ID bigint NOT NULL,

NOMBRE nvarchar(255) NULL,

DIRECCION nvarchar(255) NULL,

CONTRASENA nvarchar(255) NULL,

CONSTRAINT PK\_CLIENTE PRIMARY KEY(ID)

)

CREATE TABLE CATEGORIA

(

ID bigint NOT NULL,

NOMBRE nvarchar(255) NULL,

DESCRIPCION nvarchar(255) NULL,

CONSTRAINT PK\_CATEGORIA PRIMARY KEY(ID)

)

CREATE TABLE DETALLE

(

ID bigint IDENTITY(1,1) NOT NULL,

CANTIDAD bigint NULL,

TOTAL bigint NULL,

IDVENTA bigint NOT NULL,

IDPRODUCTO bigint NOT NULL,

CONSTRAINT PK\_DETALLE PRIMARY KEY(ID)

)

CREATE TABLE VENTA

(

ID bigint NOT NULL,

FECHA nvarchar(255) NULL,

DESCUENTO float NULL,

IDCLIENTE bigint NOT NULL,

CONSTRAINT PK\_VENTA PRIMARY KEY(ID)

)

CREATE TABLE PRODUCTO

(

ID bigint NOT NULL,

NOMBRE nvarchar(255) NULL,

PRECIO bigint NULL,

STOCK nvarchar(255) NULL,

IDPROVEEDOR bigint NOT NULL,

IDCATEGORIA bigint NOT NULL,

CONSTRAINT PK\_PRODUCTO PRIMARY KEY(ID)

)

CREATE TABLE PROVEEDOR

(

ID bigint NOT NULL,

NOMBRE nvarchar(255) NULL,

DIRECCION nvarchar(255) NULL,

TELEFONO nvarchar(255) NULL,

SITIO nvarchar(255) NULL,

CONSTRAINT PK\_PROVEEDOR PRIMARY KEY(ID)

)

CREATE TABLE TELEFONO

(

ID bigint NOT NULL,

TELEFONO nvarchar(255) NULL,

IDCLIENTE bigint NOT NULL,

CONSTRAINT PK\_TELEFONO PRIMARY KEY(ID)

)

ALTER TABLE DETALLE

ADD CONSTRAINT FK\_DETALLE\_VENTA

FOREIGN KEY(IDVENTA) REFERENCES VENTA(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE DETALLE

ADD CONSTRAINT FK\_DETALLE\_PRODUCTO

FOREIGN KEY(IDPRODUCTO) REFERENCES PRODUCTO(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE VENTA

ADD CONSTRAINT FK\_VENTA\_CLIENTE

FOREIGN KEY(IDCLIENTE) REFERENCES CLIENTE(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE PRODUCTO

ADD CONSTRAINT FK\_PRODUCTO\_PROVEEDOR

FOREIGN KEY(IDPROVEEDOR) REFERENCES PROVEEDOR(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE PRODUCTO

ADD CONSTRAINT FK\_PRODUCTO\_CATEGORIA

FOREIGN KEY(IDCATEGORIA) REFERENCES CATEGORIA(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE TELEFONO

ADD CONSTRAINT FK\_TELEFONO\_CLIENTE

FOREIGN KEY(IDCLIENTE) REFERENCES CLIENTE(ID)

ON DELETE CASCADE

ON UPDATE CASCADE

* + Scripts de inserción de datos semilla (maestros)

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

INSERT INTO dbo.CLIENTE (ID, NOMBRE, DIRECCION, CONTRASENA) VALUES (1017259022, 'Juan Pablo Romero Londoño', 'Calle 83A', 'ElRomero123')

INSERT INTO dbo.PROVEEDOR (ID, NOMBRE, DIRECCION, TELEFONO, SITIO) VALUES (1, 'Sebastián Becerra', 'Calle 53', '849146', 'www.becerras.com')

INSERT INTO dbo.CATEGORIA (ID, NOMBRE, DESCRIPCION) VALUES (1, 'Camisetas deportivas', 'Camisetas deportivas con nombres de equipos')

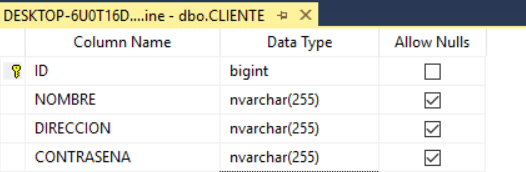
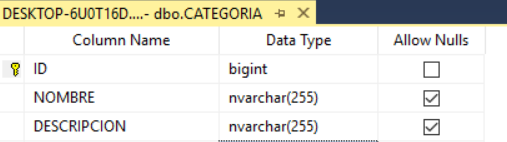
INSERT INTO dbo.PRODUCTO (ID, NOMBRE, PRECIO, STOCK, IDCATEGORIA, IDPROVEEDOR) VALUES (1, 'Camiseta firmada', 350400, 'N/A', 1, 1)

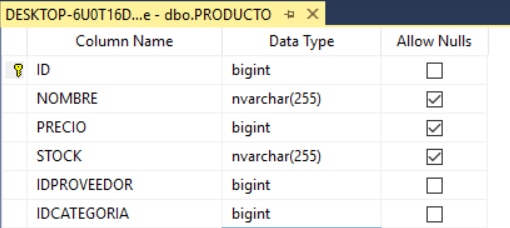
INSERT INTO dbo.CATEGORIA (ID, NOMBRE, DESCRIPCION) VALUES (2, 'Accesorios deportivos', 'Gafas, relojes, collares, etc. ')

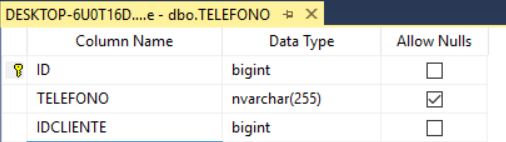
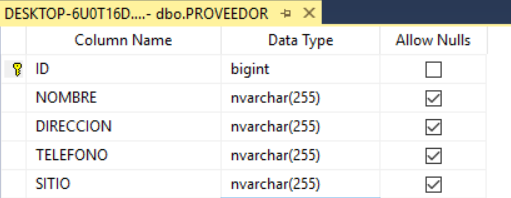
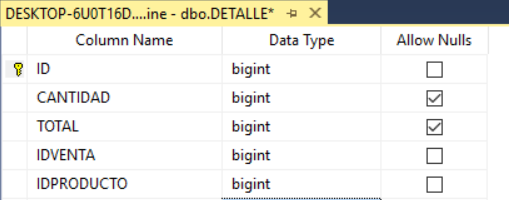
INSERT INTO dbo.PRODUCTO (ID, NOMBRE, PRECIO, STOCK, IDCATEGORIA, IDPROVEEDOR) VALUES (2, 'Gafas de playa', 87400, 'N/A', 2, 1)

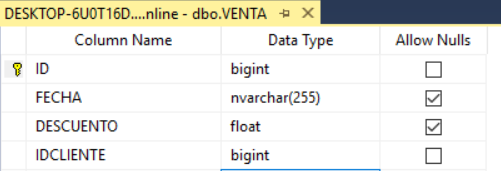
- Diccionario de datos.

A continuación, se muestra un diccionario de datos proporcionado por SQL server.









2. (35%) Se debe construir una aplicación web que permita la compra de artículos por parte de los usuarios registrados, la consulta de artículos disponibles, el historial de compra de los usuarios del sistema (no es necesario crear interfaces para la administración de usuarios del sistema ni para la gestión de maestros) y debe existir un módulo adicional donde se puede observar el comportamiento de los usuarios (con su puntuación, listas de deseos, búsquedas y comentarios). Toda la capa de acceso a datos debe realizarse a través de un Framework de persistencia (para el módulo de ventas).

Como resultado final de la realización de este punto se requiere:

- Código fuente de la aplicación.

Se encuentra anexo en el proyecto.

3. (35%) Es necesario construir una base de datos noSQL para la administración de la información de puntuación de productos, comentarios, listas de deseos y búsqueda de productos. Esta información es necesaria para el análisis de perfil de compra de los usuarios.

Como resultado final de la realización de este punto se requiere:

- Artefactos necesarios para la instalación de la base de datos noSQL.

Primero que todo se debe crear una base de datos NOSQL en Mlab. Plan: gratuito.

To connect using the mongo shell:

mongo ds111496.mlab.com:11496/db\_final -u <dbuser> -p <dbpassword>

To connect using a driver via the standard MongoDB URI (what's this?): mongodb://<dbuser>:<dbpassword>@ds111496.mlab.com:11496/db\_final

Servidor: ds111496.mlab.com

Puerto: 11496

Nombre de la base de datos: db\_final

Usuario: final

Contraseña: 123456

En RoboMongo

Tipo: conexión directa.

Nombre conexión = db\_final

Después de estar conectados con RoboMongo, se creará un documento

db.createCollection("Clientes");

Insertar la primera información.

db.createCollection("Clientes");

db.Clientes.insert(

[

{

\_id: 287960,

Nombre: "Juan Pablo",

Apellido: "Romero Londoño",

Comentarios:

[

[1,"Camiseta firmada", 4, "Interesante!"]

],

Deseos:

[

[2 ,"Gafas de playa"]

],

Busquedas:

[

[2 ,"Gafas de playa"]

]

}

]

);

db.Clientes.update({\_id:1}, {$push: { "Comentarios": [3,"Gafas azules", 4, "Interesante"] } } );