



UNAH

Universidad Nacional Autónoma de Honduras

Proyecto Final

Clase: IS501-Base de Datos I

Sección: 0700

Pertenece A: David Antonio Aguirre Sierra

Profesor: Ing. Cindy Euceda.

Cuenta: 20151020035

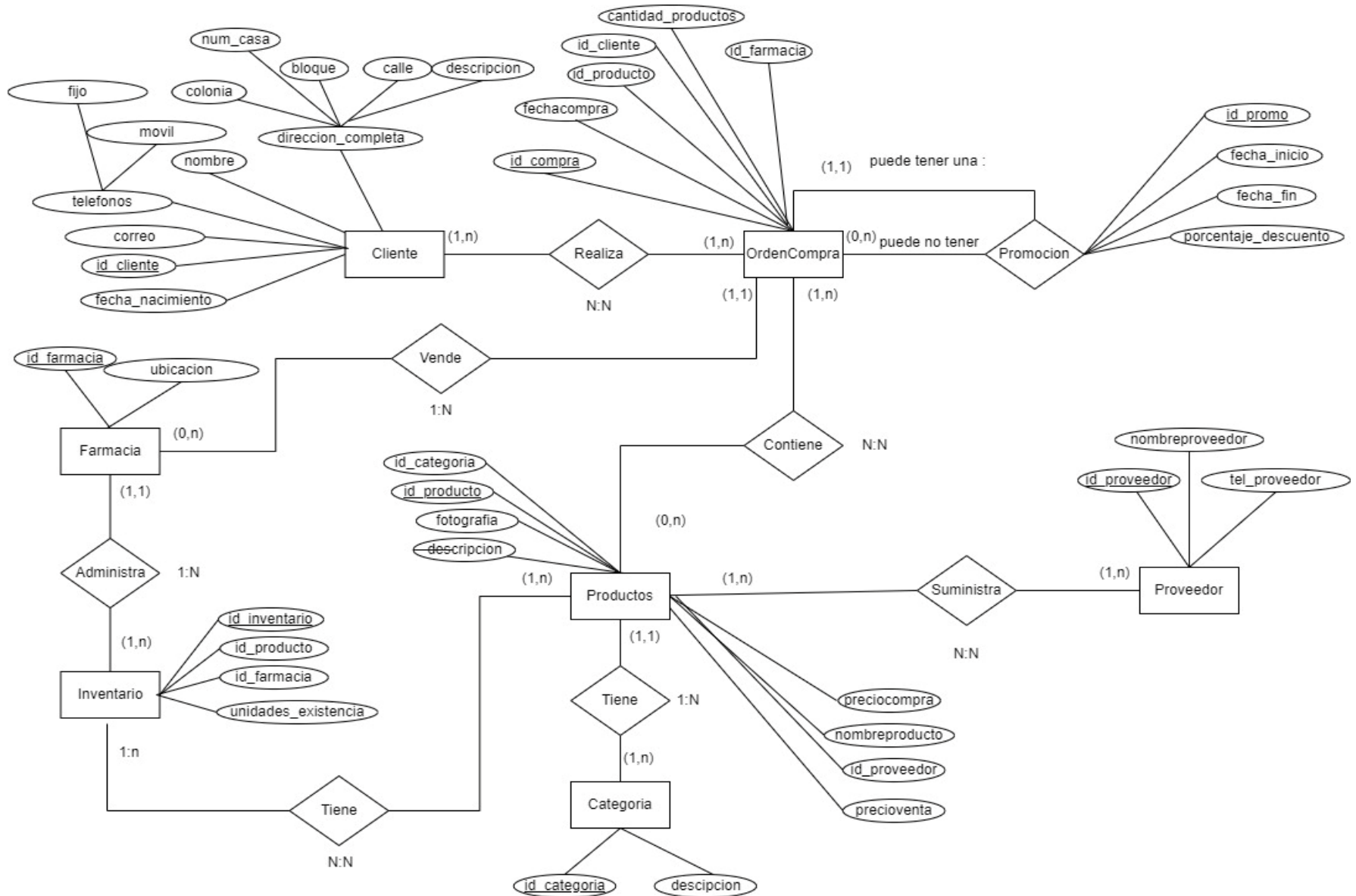
Tegucigalpa M.D.C. 30 de Abril de 2022.

Introducción

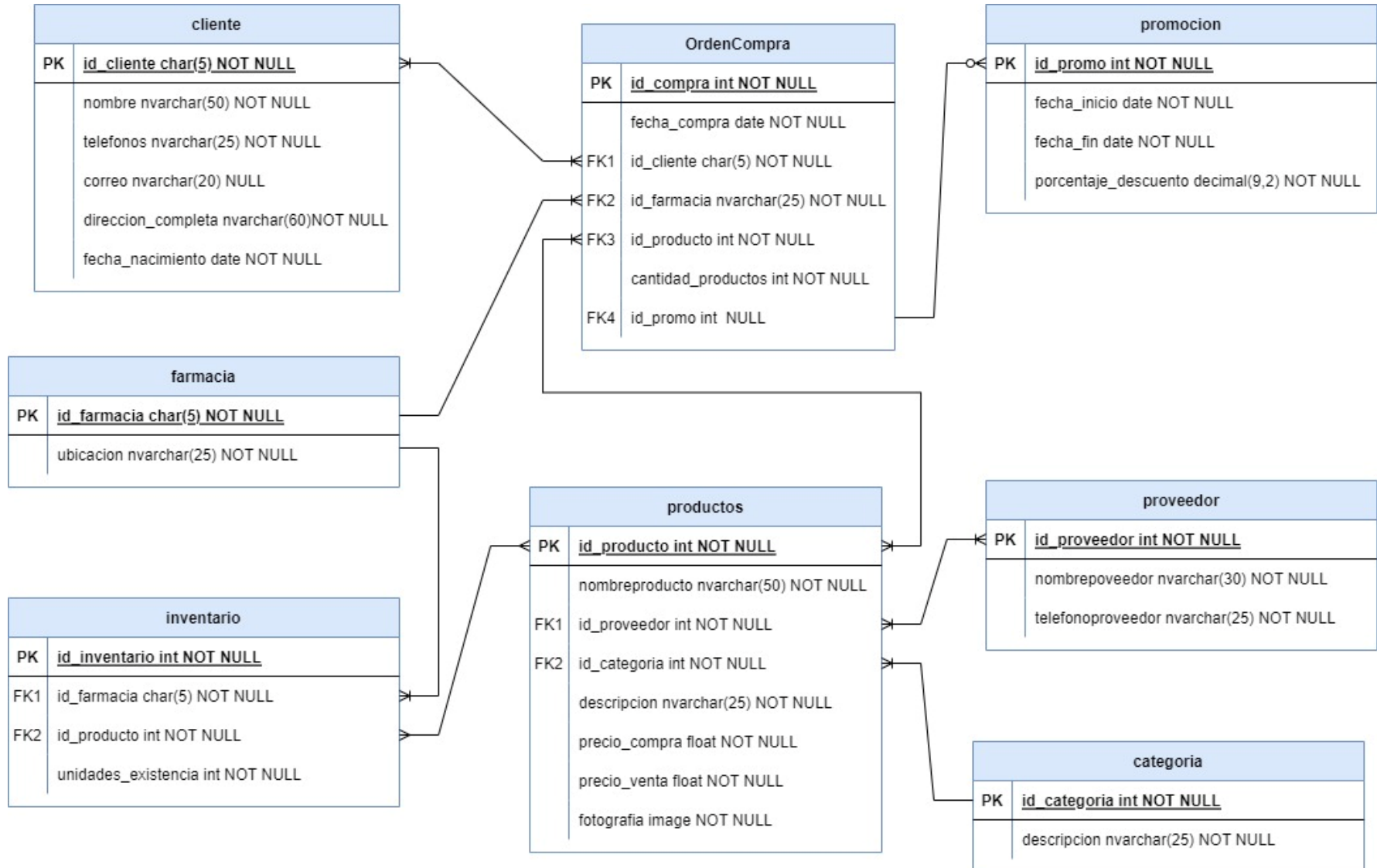
El presente trabajo se desarrolla ya que la mayoría de nosotros nos encontramos con actividades que requieren algún tipo de interacción con una base de datos, ingresos bancarios, una entrada al cine, solicitud de una suscripción, compras de productos, etc. Estas interacciones son ejemplos bases de datos, datos que permiten a diario generar reportes, datos históricos que serán de utilidad para los usuarios que interactúen con esta información.

En el presente proyecto se pueden observar los datos que están relacionados entre si conforme al esquema entidad relación, en el cual se detalla en los diagramas realizados previo a la construcción de las tablas en la herramienta MS-SQL, También se adjuntan los scripts realizados para la creación, consultas, vistas y procedimientos almacenados, necesarios en el problema dado para desarrollo de este proyecto.

I. Diagrama entidad – relación



II. Diagrama Relacional:



III. Script de creación de la base de datos

```
CREATE DATABASE Farmacia;
```

```
USE Farmacia;
```

```
CREATE TABLE cliente(  
    id_cliente char(5) primary key NOT NULL,  
    nombre nvarchar(50) NOT NULL,  
    telefonos nvarchar(25) NOT NULL,  
    correo nvarchar(20) NULL,  
    direccion_completa nvarchar(60) NOT NULL,  
    fecha_nacimiento date NOT NULL  
);
```

```
CREATE TABLE farmacia(  
    id_farmacia char(5) primary key NOT NULL,  
    ubicacion nvarchar(25) NOT NULL  
);
```

```
CREATE TABLE categoria(  
    id_categoria int identity(1,1) primary key NOT NULL,  
    descripcion nvarchar(25) NOT NULL  
);
```

```
CREATE TABLE proveedor(  
    id_proveedor int identity(1,1) primary key NOT NULL,  
    nombrepoveedor nvarchar(30) NOT NULL,  
    telefonoproveedor nvarchar(25) NOT NULL  
);
```

```
CREATE TABLE promociones(  
    id_promo int identity(1,1) primary key NOT NULL,  
    fecha_inicio date NOT NULL,  
    fecha_fin date NOT NULL,  
    porcentaje_descuento decimal (9, 2) NOT NULL  
);
```

```

CREATE TABLE productos(
    id_producto int primary key identity(1,1) NOT NULL,
    nombreproducto nvarchar(50) NOT NULL,
    id_proveedor int foreign key references proveedor(id_proveedor)NOT NULL,
    id_categoria int foreign key references categoria(id_categoria)NOT NULL,
    descripcion nvarchar(25) NOT NULL,
    precio_compra float NOT NULL,
    precio_venta float NOT NULL,
    fotografia varbinary(max) NOT NULL
);

CREATE TABLE inventario(
    id_inventario int identity(1,1) primary key NOT NULL,
    id_farmacia char(5) foreign key references farmacia(id_farmacia) NOT NULL,
    id_producto int foreign key references productos(id_producto)NOT NULL,
    unidades_existencia int NOT NULL
);

CREATE TABLE ordencompra(
    id_compra int primary key identity(1,1) NOT NULL,
    fecha_compra date NOT NULL,
    id_cliente char(5) foreign key references cliente(id_cliente)NOT NULL,
    id_farmacia char(5) foreign key references farmacia(id_farmacia) NOT NULL,
    id_producto int foreign key references productos(id_producto)NOT NULL,
    cantidad_productos int NOT NULL,
    id_promo int foreign key references promociones(id_promo) NULL,
);

```

Script de creación de la base de datos.sql - L-65J1B2OL\SQLEXPRESS.Farmacia (L-65J1B2OL\d_jk7 (52)) - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Project Tools Window Help

compras

Execute

Object Explorer

Connect

- EJERCICIO6
- EJERCICIO7
- Farmacia
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.categoria
 - dbo.cliente
 - dbo.farmacia
 - dbo.inventario
 - dbo.ordencompra
 - dbo.productos
 - dbo.promociones
 - dbo.proveedor
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Northwind
 - TAREA4
 - UNAH1
 - VENTAS
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler

Script de creación...J1B2OL\d_jk7 (52)

```

CREATE DATABASE Farmacia;
USE Farmacia;

CREATE TABLE cliente(
    id_cliente char(5) primary key NOT NULL,
    nombre nvarchar(50) NOT NULL,
    telefonos nvarchar(25) NOT NULL,
    correo nvarchar(20) NULL,
    direccion_completa nvarchar(60) NOT NULL,
    fecha_nacimiento date NOT NULL
);

CREATE TABLE farmacia(
    id_farmacia char(5) primary key NOT NULL,
    ubicacion nvarchar(25) NOT NULL
);

CREATE TABLE categoria(
    id_categoria int identity(1,1) primary key NOT NULL,
    descripcion nvarchar(25) NOT NULL
);

CREATE TABLE proveedor(
    id_proveedor int identity(1,1) primary key NOT NULL,
    nombrepoveedor nvarchar(30) NOT NULL,
    telefonoproveedor nvarchar(25) NOT NULL
);

CREATE TABLE promociones(
    id_promo int identity(1,1) primary key NOT NULL,
    fecha_inicio date NOT NULL,
    fecha_fin date NOT NULL,
    porcentaje_descuento decimal (9, 2) NOT NULL
);
  
```

100 %

Connected. (1/1)

L-65J1B2OL\SQLEXPRESS (15.0... | L-65J1B2OL\d_jk7 (52) | Farmacia | 00:00:00 | 0 rows

IV. Reportes

a. Consultas

```
USE Farmacia;

--i. Edad promedio de los clientes
select AVG(CASE WHEN DATEADD(yy,DATEDIFF(YEAR,fecha_nacimiento,GETDATE()),fecha_nacimiento)<GETDATE() then
DATEDIFF(YEAR,fecha_nacimiento,GETDATE())
      else DATEDIFF(YEAR,fecha_nacimiento,GETDATE())-1
      END) as Edad_Promedio from cliente

--ii. 10 productos más vendidos
select top(10) id_producto, sum(cantidad_productos)as Cantidad from ordencompra
group by id_producto order by sum(cantidad_productos) desc

--iii. Cantidad de productos por categoría
SELECT C.descripcion, SUM(B.cantidad)as Cantidad_por_Categoría from categoria C
left join
(
    select p.id_categoria,i.Cantidad from productos p
    left join(
        Select id_producto,sum(unidades_existencia)as Cantidad from inventario group by id_producto
    ) i
    on i.id_producto=p.id_producto
) B
on C.id_categoria=B.id_categoria
Group by C.descripcion

--iv. Listar los productos más bajos en ventas
select p.nombreproducto, sum(C.cantidad_productos)as Cantidad from productos p
left join
    ordencompra C
on C.id_producto=P.id_producto
group by p.nombreproducto order by sum(cantidad_productos) asc

--v. Listado de productos agrupados por categoría*/
SELECT C.id_categoria,C.descripcion, P.id_producto,p.nombreproducto FROM productos P
FULL OUTER join
    categoria c
ON p.id_categoria=c.id_categoria
```


Consultas.sql - L-65J1B2OL\SQLEXPRESS.Farmacia (L-65J1B2OL\d_jk7 (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

compras

Execute

Object Explorer

Connect

- EJERCICIO6
- EJERCICIO7
- Farmacia
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.categoria
 - dbo.cliente
 - dbo.farmacia
 - dbo.inventario
 - dbo.ordencompra
 - dbo.productos
 - dbo.promociones
 - dbo.proveedor
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- Northwind
- TAREA4
- UNAH1
- VENTAS
- Security
- Server Objects
- Replication
- PolyBase
- Management
- XEvent Profiler

Consultas.sql - L-65J1B2OL\d_jk7 (52)

```

/*A) Consultas*/
USE Farmacia;
--i. Edad promedio de los clientes
select AVG(CASE WHEN DATEADD(yy,DATEDIFF(YEAR,fecha_nacimiento,GETDATE()),fecha_nacimiento)<GETDATE() then DATEDIFF(YEAR,fecha_nacimiento,GETDATE())
else DATEDIFF(YEAR,fecha_nacimiento,GETDATE())-1
END) as Edad_Promedio from cliente

--ii. 10 productos más vendidos
select top(10) id_producto, sum(cantidad_productos)as Cantidad from ordencompra
group by id_producto order by sum(cantidad_productos) desc

--iii. Cantidad de productos por categoría
SELECT C.descripcion, SUM(B.cantidad)as Cantidad_por_Categoría from categoria C
left join
(
select p.id_categoria,i.Cantidad from productos p
left join(
Select id_producto,sum(unidades_existencia)as Cantidad from inventario group by id_producto
) i
on i.id_producto=p.id_producto
) B
on C.id_categoria=B.id_categoria
Group by C.descripcion

--iv. Listar los productos más bajos en ventas
select p.nombreproducto, sum(C.cantidad_productos)as Cantidad from productos p
left join
ordencompra C
on C.id_producto=P.id_producto
group by p.nombreproducto order by sum(cantidad_productos) asc

--v. Listado de productos agrupados por categoría*/
SELECT C.id_categoria,C.descripcion, P.id_producto,p.nombreproducto FROM productos P
FULL OUTER join
categoria c
ON p.id_categoria=c.id_categoria
  
```

100 %

Connected: (1/1)

L-65J1B2OL\SQLEXPRESS (15.0... | L-65J1B2OL\d_jk7 (52) | Farmacia | 00:00:00 | 0 rows

Ready

Ln 1 Col 1 Ch 1 INS

b. Vistas

--i. Reporte mensual de ventas por sucursal

```
CREATE VIEW Reporte_MensualVentasSucursal as
SELECT DATENAME (MONTH,C.fecha_compra)as Mes,C.id_farmacia as 'Sucursal',sum((C.cantidad_productos*P.precio_venta))as 'Total
Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by DATENAME (MONTH,C.fecha_compra),C.id_farmacia
```

--ii. Reporte de ventas mensual por producto

```
CREATE VIEW Reporte_MensualVentasProducto as
SELECT DATENAME (MONTH,C.fecha_compra)as Mes,C.id_producto as 'Producto',sum((C.cantidad_productos*P.precio_venta))as 'Total
Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by DATENAME (MONTH,C.fecha_compra),C.id_producto
```

--iii. Total ventas por categoría

```
CREATE VIEW Reporte_VentasCategoria as
SELECT P.id_categoria as 'Categoría',sum((C.cantidad_productos*P.precio_venta))as 'Total Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by P.id_categoria
```

--iv. Productos en el inventario con cantidad menor que 10 unidades por sucursal

```
CREATE VIEW Reporte_InventarioMenorDiez as
SELECT id_farmacia,id_producto,unidades_existencia FROM inventario
WHERE unidades_existencia <10
```

--v. Listado de clientes que no han realizado ningún pedido durante el mes

```
CREATE VIEW Reporte_ClientesSinPedidos as
SELECT id_cliente,nombre,telefonos FROM cliente
WHERE id_cliente NOT IN (SELECT id_cliente FROM ordencompra WHERE MONTH(fecha_compra)=MONTH(GETDATE()))
```

Vistas.sql - L-65J1B2OL\SQLEXPRESS.Farmacia (L-65J1B2OL\d_jk7 (52)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

compras

Execute

Object Explorer

Connect

- EJERCICIO2_2
- EJERCICIO2_3
- EJERCICIO2_4
- Ejercicio3
- EJERCICIO4
- EJERCICIO5
- EJERCICIO6
- EJERCICIO7
- Farmacia
 - Database Diagrams
 - Tables
 - Views
 - System Views
 - dbo.Reporte_ClientesSinPedidos
 - dbo.Reporte_InventarioMenorDiez
 - dbo.Reporte_MensualVentas
 - dbo.Reporte_MensualVentasProducto
 - dbo.Reporte_VentasCategoria
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Northwind
 - TAREA4
 - UNAH1
 - VENTAS
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEvent Profiler

Vistas.sql - L-65J1B2OL\d_jk7 (52)

```

/*b) Vistas*/
--i. Reporte mensual de ventas por sucursal
CREATE VIEW Reporte_MensualVentasSucursal AS
SELECT DATENAME (MONTH,C.fecha_compra)as Mes,C.id_farmacia as 'Sucursal',sum((C.cantidad_productos*P.precio_venta))as 'Total Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by DATENAME (MONTH,C.fecha_compra),C.id_farmacia

--ii. Reporte de ventas mensual por producto
CREATE VIEW Reporte_MensualVentasProducto AS
SELECT DATENAME (MONTH,C.fecha_compra)as Mes,C.id_producto as 'Producto',sum((C.cantidad_productos*P.precio_venta))as 'Total Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by DATENAME (MONTH,C.fecha_compra),C.id_producto

--iii. Total ventas por categoría
CREATE VIEW Reporte_VentasCategoria AS
SELECT P.id_categoria as 'Categoria',sum((C.cantidad_productos*P.precio_venta))as 'Total Ventas' FROM ordencompra C
left join productos P
on P.id_producto=C.id_producto
Group by P.id_categoria

--iv. Productos en el inventario con cantidad menor que 10 unidades por sucursal
CREATE VIEW Reporte_InventarioMenorDiez AS
SELECT id_farmacia,id_producto,unidades_existencia FROM inventario
WHERE unidades_existencia <10

--v. Listado de clientes que no han realizado ningún pedido durante el mes
CREATE VIEW Reporte_ClientesSinPedidos AS
SELECT id_cliente,nombre,telefonos FROM cliente
WHERE id_cliente NOT IN (SELECT id_cliente FROM ordencompra WHERE MONTH(fecha_compra)=MONTH(GETDATE()))
  
```

100 %

Connected: (1/1)

L-65J1B2OL\SQLEXPRESS (15.0... | L-65J1B2OL\d_jk7 (52) | Farmacia | 00:00:00 | 0 rows

Ready

c. Procedimientos Almacenados

```
USE Farmacia;
--i. Función para búsqueda de clientes por nombre
create procedure getCliente @nombre nvarchar(50)
as
select id_cliente,nombre,telefonos from cliente where nombre like '%' + @nombre + '%'

execute getCliente @nombre='Ana'

--ii. Eliminar clientes por id
create procedure Eliminaclientes @id char(5)
as
delete cliente where id_cliente=@id

execute Eliminaclientes @id='p'

--iii. Actualizar clientes por id

create procedure Actualizaclientes @id char(5),@nombre nvarchar(50),@telefonos nvarchar(25),@correo nvarchar(20),@direccion
nvarchar(60)
as
update cliente
set nombre=@nombre,
    telefonos=@telefonos,
    correo=@correo,
    direccion_completa=@direccion
where id_cliente=@id

execute Actualizaclientes @id='WOLZA',@nombre='Zbyszek Piestrzeniewicz',@telefonos='(26) 642-
7012',@correo='zp455@gmail.com',@direccion='ul. Filtrowa 68'

SELECT * from cliente

--iv. Listado de todos los pedidos realizados por un cliente
create procedure PedidosRealizados @id char(5)
as
SELECT * from ordencompra where id_cliente=@id
```

```
execute PedidosRealizados @id=ANATR
```

```
--v. Total de ventas de un producto en una fecha determinada
```

```
create procedure VentasProductoFecha @idproducto int, @fecha date
```

```
as
```

```
SELECT * from ordencompra where id_producto=@idproducto and fecha_compra=@fecha
```

```
execute VentasProductoFecha @idproducto=15, @fecha='2022-03-27'
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the 'Stored Procedures' folder expanded under the 'Farmacia' database. The main editor pane shows a SQL script with the following content:

```

/*c) Procedimientos almacenados*/
USE Farmacia;
--i. Función para búsqueda de clientes por nombre
create procedure getCliente @nombre nvarchar(50)
as
select id_cliente,nombre,telefonos from cliente where nombre like '%' + @nombre + '%'

execute getCliente @nombre='Ana'

--ii. Eliminar clientes por id
create procedure Eliminaclientes @id char(5)
as
delete cliente where id_cliente=@id

execute Eliminaclientes @id='p'

--iii. Actualizar clientes por id
create procedure Actualizaclientes @id char(5),@nombre nvarchar(50),@telefonos nvarchar(25),@correo nvarchar(20),@direccion nvarchar(60)
as
update cliente
set nombre=@nombre,
    telefonos=@telefonos,
    correo=@correo,
    direccion_completa=@direccion
where id_cliente=@id

execute Actualizaclientes @id='WOLZA',@nombre='Zbyszek Piestrzeniewicz',@telefonos='(26) 642-7012',@correo='zp455@gmail.com',@direccion='ul. Filtrowa 68'

SELECT * from cliente

--iv. Listado de todos los pedidos realizados por un cliente
create procedure PedidosRealizados @id char(5)
as
SELECT * from ordencompra where id_cliente=@id
  
```

The status bar at the bottom indicates the connection is successful and shows the current database as 'Farmacia'.

Procedimientos Almacenados.sql - L-65J1B2OL\SQLEXPRESS.Farmacia (L-65J1B2OL\d_jk7 (52)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

compras

Execute

Object Explorer

Connect

- EJERCICIO2_1
- EJERCICIO2_2
- EJERCICIO2_3
- EJERCICIO2_4
- Ejercicio3
- EJERCICIO4
- EJERCICIO5
- EJERCICIO6
- EJERCICIO7
- Farmacia
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Stored Procedures
 - System Stored Procedures
 - dbo.Actualizaclientes
 - dbo.Eliminacientes
 - dbo.getCliente
 - dbo.PedidosRealizados
 - dbo.VentasProductoFecha
 - Functions
 - Database Triggers
 - Assemblies
 - Types
 - Rules
 - Defaults
 - Sequences
 - Service Broker
 - Storage
 - Security
 - Northwind
 - TAREA1

Procedimientos Alm...J1B2OL\d_jk7 (52)

```
--iv. Listado de todos los pedidos realizados por un cliente
create procedure PedidosRealizados @id char(5)
as
SELECT * from ordencompra where id_cliente=@id

execute PedidosRealizados @id=ANATR

--v. Total de ventas de un producto en una fecha determinada
create procedure VentasProductoFecha @idproducto int, @fecha date
as
SELECT * from ordencompra where id_producto=@idproducto and fecha_compra=@fecha

execute VentasProductoFecha @idproducto=15, @fecha='2022-03-27'
```

100 %

Connected. (1/1)

L-65J1B2OL\SQLEXPRESS (15.0... | L-65J1B2OL\d_jk7 (52) | Farmacia | 00:00:00 | 0 rows

Ready