

B A S E S D E
D A T O S

Proyecto Final

P R E S E N T A D O P O R

EQUIPO 7
Coria Martínez Gustavo
Oronzor Montes Manases Leonel
Pang Araizaga Ian

E N T R E G A :
1 2 / 0 6 / 2 0 2 3

Índice

<u>Introducción</u>	2
<u>Enunciado</u>	2
<u>Modelo conceptual</u>	3
<u>Modelo lógico</u>	3
<u>Diccionario de datos</u>	4
<u>Inserciones</u>	28
<u>Procedimientos almacenados</u>	34
<u>Triggers</u>	39
<u>Estadísticas</u>	42
<u>Clearbase código</u>	47
<u>Cargainicial código</u>	64
<u>DML código</u>	75
<u>Valida_trigger código</u>	88
<u>Seguridad código</u>	90
<u>Reportes código</u>	92
<u>Conclusión</u>	97

Introducción

Las bases de datos son una herramienta fundamental en el mundo de la informática y juegan un papel crucial en el almacenamiento y gestión de la información. En el ámbito académico, la materia de Bases de Datos se enfoca en enseñar los fundamentos teóricos y prácticos para diseñar, implementar y administrar sistemas de bases de datos eficientes y confiables.

En el marco de este proyecto, nos proponemos explorar y aplicar los conceptos aprendidos en la materia de Bases de Datos en un escenario real. El objetivo principal es diseñar y desarrollar una base de datos para abordar una problemática específica y ofrecer soluciones eficientes y escalables.

Para comenzar, se realizó un análisis detallado de los requisitos y necesidades del proyecto. Identificamos las entidades principales, los atributos relevantes y las relaciones entre ellas. Esto nos permitirá establecer un modelo conceptual que sirva como base para el diseño de la base de datos.

Luego, utilizaremos el modelo conceptual para crear un esquema relacional, definiendo las tablas y los campos correspondientes. También consideraremos las restricciones de integridad para garantizar la consistencia de los datos.

Una vez definido el esquema relacional, pasaremos a implementar la base de datos utilizando un sistema de gestión de bases de datos (SGBD) adecuado, se usará Microsoft SQL Server. Configuraremos la estructura de la base de datos y cargaremos los datos iniciales necesarios.

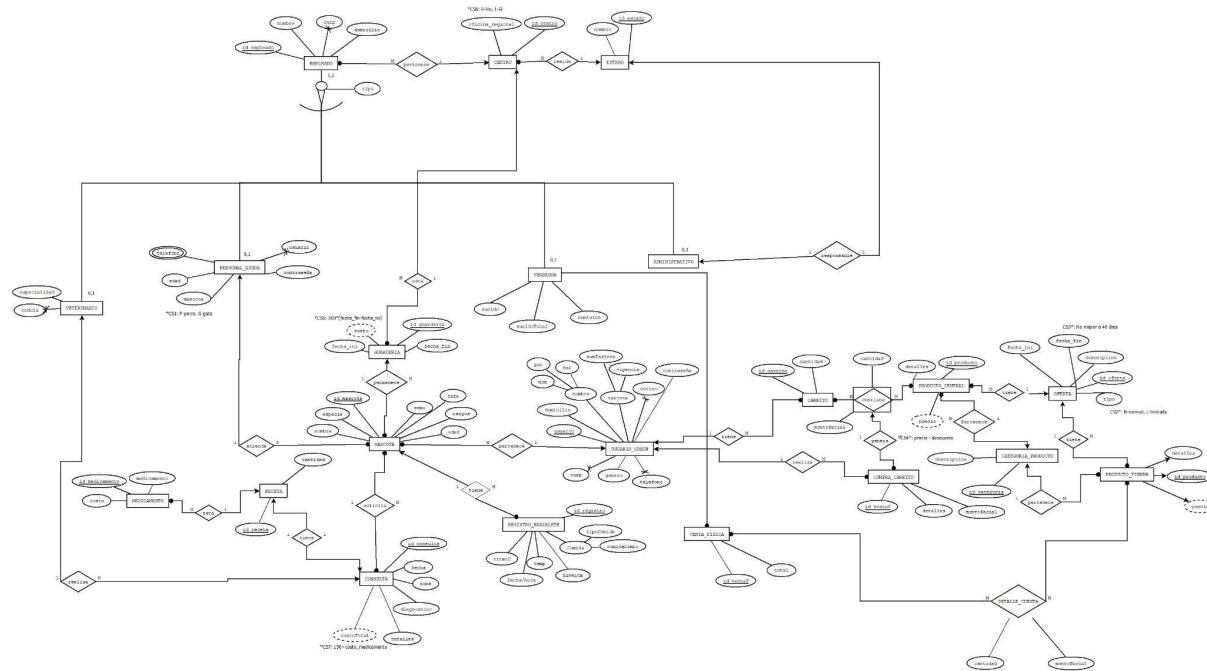
A continuación, nos enfocaremos en desarrollar consultas y operaciones que permitan obtener información relevante de la base de datos. Utilizaremos el lenguaje SQL para realizar consultas simples y complejas, generando informes y análisis que ayuden a la toma de decisiones.

Además, consideraremos la seguridad y la privacidad de los datos, implementando medidas para proteger la información sensible y asegurar el acceso adecuado a los usuarios autorizados.

Enunciado

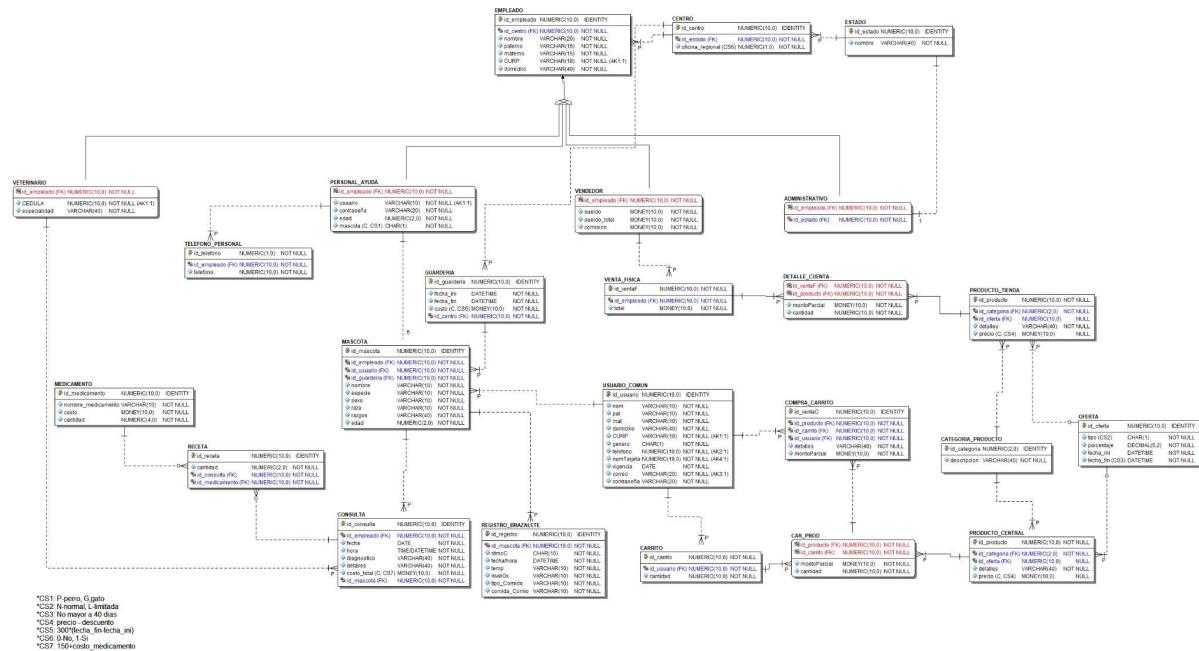
Se debe desarrollar una aplicación móvil para los usuarios de un servicio de guardería y veterinaria para perros y gatos, así como para los empleados que trabajan en los centros de cuidado.

Modelo conceptual



Modelo lógico

EQUIPO 7
Coria Martínez Gustavo
Oronoz Montes Manases Leonel
Pang Araizaga Ian



Diccionario de datos

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio® Web Report Generation interface. On the left, there is a navigation tree for a database model named P1.DM1. The tree includes sections for Title, Logical, Equipo7_PETCO, Main Model, Tables, Columns, Indexes, and Foreign Keys. Under Tables, the ADMINISTRATIVO table is selected, showing its detailed properties and relationships.

ADMINISTRATIVO Entity Properties:

Logical Entity Name	ADMINISTRATIVO
Default Table Name	ADMINISTRATIVO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Tabla la cual almacena los datos de los empleados que son administrativos.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE ADMINISTRATIVO(id_empleado numeric(10, 0) NOT NULL, id_estado numeric(10, 0) NOT NULL, CONSTRAINT PK_ADMINISTRATIVO PRIMARY KEY NONCLUSTERED (id_empleado), CONSTRAINT FK_ESTADO_ADMIN FOREIGN KEY (id_estado) REFERENCES ESTADO(id_estado), CONSTRAINT RefEMPLEADO10 FOREIGN KEY (id_empleado) REFERENCES EMPLEADO(id_empleado))</pre>
PostSQL	

ADMINISTRATIVO Columns:

ColumnName	Domain	Datatype	NULL	Definition
id_empleado		numeric(10, 0)	NO	Es la llave de la tabla que sera igual a la del empleado.
id_estado		numeric(10, 0)	NO	Es la llave foranea la cual indica en que estado se encuentra el administrativo.

ADMINISTRATIVO Primary and Unique Keys:

Key Name	Key Type	Keys
PK_ADMINISTRATIVO	PrimaryKey	id_empleado

ADMINISTRATIVO Foreign Keys:

--



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PROD
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTRAL
 - PRODUCTO_TIENDA
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

CAR_PROD

Logical Entity Name	CAR_PROD
Default Table Name	CAR_PROD
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los productos que se van a comprar, así como el carrito.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE CAR_PROD(id_producto numeric(10, 0) NOT NULL, id_carrito numeric(10, 0) NOT NULL, montoParcial money NOT NULL, cantidad numeric(10, 0) NOT NULL, CONSTRAINT PK_CAR_PROD PRIMARY KEY NONCLUSTERED (id_producto), CONSTRAINT FK_CARRITO_CAR_PROD FOREIGN KEY (id_carrito) REFERENCES CARRITO(id_carrito), CONSTRAINT FK_PRODUCTO_CENTRAL_CAR_PROD FOREIGN KEY (id_producto) REFERENCES PRODUCTO_CENTRAL(id_producto)) go</pre>
PostSQL	

CAR_PROD Columns

ColumnName	Domain	Datatype	NULL	Definition
id_producto		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
id_carrito		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
montoParcial		money	NO	Columna que va a guardar el costo parcial
cantidad		numeric(10, 0)	NO	Columna que va a guardar el numero de unidades



Embarcadero ER/Studio® Web Report Generation

CARRITO

Logical Entity Name	CARRITO
Default Table Name	CARRITO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla va a almacenar el identificador del usuario y asignara una ide
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE CARRITO(id_carrito numeric(10, 0) NOT NULL, id_usuario numeric(10, 0) NOT NULL, cantidad numeric(10, 0) NOT NULL, CONSTRAINT PK_CARRITO PRIMARY KEY NONCLUSTERED (id_carrito) CONSTRAINT FK_USUARIO_COMUN_CARRITO FOREIGN KEY (id_usuario) REFERENCES USUARIO_COMUN(id_usuario)) go</pre>
PostSQL	

CARRITO Columns

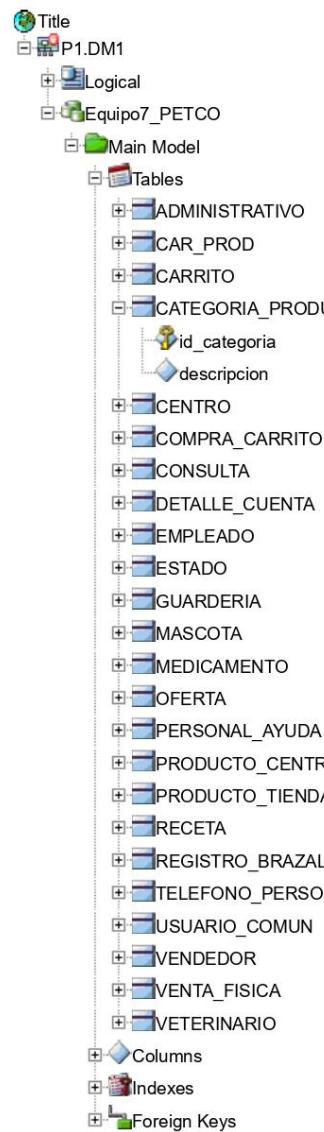
ColumnName	Domain	Datatype	NULL	Definition
id_carrito		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
id_usuario		numeric(10, 0)	NO	Es la llave foranea de la tabla que se
cantidad		numeric(10, 0)	NO	Columna que va a guardar la cantida

CARRITO Primary and Unique Keys

Key Name	Key Type	Keys
PK_CARRITO	PrimaryKey	id_carrito



Embarcadero ER/Studio® Web Report Generation



CATEGORIA_PRODUCTO	
Logical Entity Name	CATEGORIA_PRODUCTO
Default Table Name	CATEGORIA_PRODUCTO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Tabla la cual va a tener todas las distintas categorias que pueden tener los productos.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE CATEGORIA_PRODUCTO(id_categoria numeric(2, 0) IDENTITY(1,1), descripcion varchar(40) NOT NULL, CONSTRAINT PK_CATEGORIA_PRODUCTO PRIMARY KEY NONCLUSTERED (id_categoria)) go</pre>
PostSQL	

CATEGORIA_PRODUCTO Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_categoria		numeric(2, 0)	NO	Es la llave de la tabla y sera un constraint.
descripcion		varchar(40)	NO	Es la columna la cual va a almacenar el nombre de la categoria.

CATEGORIA_PRODUCTO Primary and Unique Keys		
Key Name	Key Type	Keys
PK_CATEGORIA_PRODUCTO	PrimaryKey	id_categoria

CATEGORIA_PRODUCTO Foreign Keys		
Parent Relationships:		



Embarcadero ER/Studio® Web Report Generation

CENTRO

Logical Entity Name	CENTRO
Default Table Name	CENTRO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla es un catalogo el cual va a almacenar todos los centros que existen.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE CENTRO(id_centro numeric(10, 0) IDENTITY(1,1), id_estado numeric(10, 0) NOT NULL, oficinaRegional numeric(1, 0) NOT NULL, CONSTRAINT CK_CS6 CHECK (oficinaRegional in(0,1)), CONSTRAINT PK_CENTRO PRIMARY KEY NONCLUSTERED (id_centro) CONSTRAINT FK_ESTADO_CEN FOREIGN KEY (id_estado) REFERENCES ESTADO(id_estado)) go</pre>
PostSQL	

CENTRO Columns

ColumnName	Domain	Datatype	NULL	Definition
id_centro		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint.
id_estado		numeric(10, 0)	NO	Es una llave foranea la cual va a determinar si es un centro.
oficinaRegional		numeric(1, 0)	NO	Columna que va a determinar si es un centro.

CENTRO Primary and Unique Keys

Key Name	Key Type	Keys
PK_CENTRO	PrimaryKey	id_centro



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
- Logical
- Equipo7_PETCO
- Main Model
- Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - id_ventaC
 - id_producto
 - id_carrito
 - id_usuario
 - detalles
 - montoParcial
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
- Columns
- Indexes
- Foreign Keys

COMPRA_CARRITO

Logical Entity Name	COMPRA_CARRITO
Default Table Name	COMPRA_CARRITO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Tabla la cual va a tener la informacion de los productos comprados, su definicion es que va a tener la informacion de los productos comprados.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE COMPRA_CARRITO(id_ventaC numeric(10, 0) IDENTITY(1,1), id_producto numeric(10, 0) NOT NULL, id_carrito numeric(10, 0) NOT NULL, id_usuario numeric(10, 0) NOT NULL, detalles varchar(40) NOT NULL, montoParcial money NOT NULL, CONSTRAINT PK_COMPRA_CARRITO PRIMARY KEY NONCLUSTERED REFERENCES USUARIO_COMUN(id_usuario), CONSTRAINT FK_CAR_PROD_COMPRA_CARRITO FOREIGN KEY (id_producto) REFERENCES CAR_PROD(id_producto, id_carrito)) go</pre>
PostSQL	

COMPRA_CARRITO Columns

ColumnName	Domain	Datatype	NULL	Definition
id_ventaC		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint.
id_producto		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla CAR_PROD.
id_carrito		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla CARRITO.
id_usuario		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla USUARIO_COMUN.
detalles		varchar(40)	NO	Columna de la tabla que va a mostrar la informacion de los productos comprados.

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface. On the left, the logical model tree for 'P1.DM1' is displayed, showing various tables like 'ADMINISTRATIVO', 'CAR_PROD', 'CARRITO', etc., under the 'Equipo7_PETCO' schema. On the right, two reports are shown:

- CONSULTA**: A table definition report. It includes fields for Logical Entity Name (CONSULTA), Default Table Name (CONSULTA), Physical Only (NO), Do Not Generate (False), Owner, Definition (a note about storing details from a consultation), Note, PreSQL, and DDL Code. The DDL code is as follows:


```
CREATE TABLE CONSULTA(
    id_consulta numeric(10, 0) IDENTITY(1,1),
    id_empleado numeric(10, 0) NOT NULL,
    fecha date NOT NULL,
    hora time(7) NOT NULL,
    diagnostico varchar(40) NOT NULL,
    detalles varchar(40) NOT NULL,
    costo_total money NOT NULL,
    id_mascota numeric(10, 0) NOT NULL,
    CONSTRAINT PK_CONSULTA PRIMARY KEY NONCLUSTERED (id_consulta)
    CONSTRAINT FK_VETERINARIO_CONSULTA FOREIGN KEY (id_empleado)
    REFERENCES VETERINARIO(id_empleado),
    CONSTRAINT FK_MASCOTA_CONSULTA FOREIGN KEY (id_mascota)
    REFERENCES MASCOTA(id_mascota)
)
go
```
- CONSULTA Columns**: A column definition report. It lists columns with their names, domains, datatypes, nullability, and definitions:

ColumnName	Domain	Datatype	NULL	Definition
id_consulta		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint
id_empleado		numeric(10, 0)	NO	Es la llave foranea de la tabla que se
fecha		date	NO	Columna que va a almacenar la fech
hora		time(7)	NO	Columna que va a almacenar la hor



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - id_ventaF
 - id_producto
 - montoParcial
 - cantidad
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

DETALLE CUENTA	
Logical Entity Name	DETALLE CUENTA
Default Table Name	DETALLE CUENTA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los productos que se van a comprar, así como el monto parcial y la cantidad.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE DETALLE CUENTA(id_ventaF numeric(10, 0) NOT NULL, id_producto numeric(10, 0) NOT NULL, montoParcial money NOT NULL, cantidad numeric(10, 0) NOT NULL, CONSTRAINT PK_DETALLE CUENTA PRIMARY KEY NONCLUSTERED CONSTRAINT FK_VENTA_FISICA_DETALLE CUENTA FOREIGN KEY REFERENCES VENTA_FISICA(id_ventaF), CONSTRAINT FK_PRODUCTO TIENDA_DETALLE CUENTA FOREIGN KEY REFERENCES PRODUCTO TIENDA(id_producto))</pre>
PostSQL	

DETALLE CUENTA Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_ventaF		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
id_producto		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
montoParcial		money	NO	Columna que va a guardar el costo p
cantidad		numeric(10, 0)	NO	Calumna que va a guardar la cantida

DETALLE CUENTA Primary and Unique Keys				



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - id_empleado**
 - id_centro**
 - nombre
 - paterno
 - materno
 - CURP
 - domicilio
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

EMPLEADO	
Logical Entity Name	EMPLEADO
Default Table Name	EMPLEADO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos del empleado como su nombre completo,
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE EMPLEADO(id_empleado numeric(10, 0) IDENTITY(1,1), id_centro numeric(10, 0) NOT NULL, nombre varchar(20) NOT NULL, paterno varchar(15) NOT NULL, materno varchar(15) NOT NULL, CURP varchar(18) NOT NULL, domicilio varchar(40) NOT NULL, CONSTRAINT PK_EMPLEADO PRIMARY KEY NONCLUSTERED (id_empleado) CONSTRAINT FK_CENTRO_EMPLEADO FOREIGN KEY (id_centro) REFERENCES CENTRO(id_centro)) go</pre>
PostSQL	

EMPLEADO Columns					
ColumnName	Domain	Datatype	NULL	Definition	
id_empleado		numeric(10, 0)	NO	Es la llave de la tabla y sera un con	
id_centro		numeric(10, 0)	NO	Es la llave foranea de la tabla que s	
nombre		varchar(20)	NO	Columna que va a almacenar los no	
paterno		varchar(15)	NO	Columna que va a almacenar el ape	
materno		varchar(15)	NO	Columna que va a almacenar el ape	



Embarcadero ER/Studio® Web Report Generation

ESTADO							
Logical Entity Name	ESTADO						
Default Table Name	ESTADO						
Physical Only	NO						
Do Not Generate	False						
Owner							
Definition	Esta tabla es un catalogo el cual va a almacenar todos los estados de la república						
Note							
PreSQL							
DDL Code	<pre>CREATE TABLE ESTADO(id_estado numeric(10, 0) IDENTITY(1,1), nombre varchar(40) NOT NULL, CONSTRAINT PK_ESTADO PRIMARY KEY NONCLUSTERED (id_estado)) go</pre>						
PostSQL							
ESTADO Columns							
ColumnName	Domain	Datatype	NULL	Definition			
id_estado		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint			
nombre		varchar(40)	NO	Columna que va a almacenar el nombre del estado			
ESTADO Primary and Unique Keys							
Key Name	Key Type	Keys					
PK_ESTADO	PrimaryKey	id_estado					
ESTADO Foreign Keys							
Parent Relationships:							
Table Name	Type	Cardinality	Constraint Name	VerbPhrase	InversePhrase	Foreign Keys	Is S



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
- Logical
- Equipo7_PETCO
- Main Model
- Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - id_guarderia
 - fecha_ini
 - fecha_fin
 - costo
 - id_centro
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
- Columns
- Indexes
- Foreign Keys

GUARDERIA	
Logical Entity Name	GUARDERIA
Default Table Name	GUARDERIA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Catalogo de todas las guarderias que existiran asi como de la informacion de los servicios que se ofertan
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE GUARDERIA(id_guarderia numeric(10, 0) IDENTITY(1,1), fecha_ini datetime NOT NULL, fecha_fin datetime NOT NULL, costo money NOT NULL, id_centro numeric(10, 0) NOT NULL, CONSTRAINT PK_GUARDERIA PRIMARY KEY NONCLUSTERED (id_guarderia) CONSTRAINT FK_CENTRO_GUARDERIA FOREIGN KEY (id_centro) REFERENCES CENTRO(id_centro)) go</pre>
PostSQL	

GUARDERIA Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_guarderia		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint
fecha_ini		datetime	NO	Columna que va a almacenar la fecha de inicio
fecha_fin		datetime	NO	Columna que va a almacenar la fecha de fin
costo		money	NO	Columna que va a almacenar el costo
id_centro		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere al centro

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface. On the left, the logical model tree is visible, showing a project named 'Equipo7_PETCO' with a 'Main Model' containing various tables like ADMINISTRATIVO, CAR_PROD, CARRITO, etc., and a specific table 'MASCOTA' expanded to show its columns: id_mascota, id_empleado, id_usuario, id_guarderia, nombre, especie, sexo, raza, razgos, and edad.

The main area displays two reports for the 'MASCOTA' table:

- MASCOTA**: A configuration report with fields like Logical Entity Name (MASCOTA), Default Table Name (MASCOTA), Physical Only (NO), Do Not Generate (False), Owner (empty), Definition (description in Spanish), Note (empty), and PreSQL (empty). The DDL Code section contains the SQL create statement for the table, including primary key and foreign key constraints.
- MASCOTA Columns**: A detailed report showing the columns of the MASCOTA table with their definitions. The columns listed are id_mascota, id_empleado, and id_usuario.

```

CREATE TABLE MASCOTA(
    id_mascota numeric(10, 0) IDENTITY(1,1),
    id_empleado numeric(10, 0) NOT NULL,
    id_usuario numeric(10, 0) NOT NULL,
    id_guarderia numeric(10, 0) NOT NULL,
    nombre varchar(10) NOT NULL,
    especie varchar(10) NOT NULL,
    sexo varchar(10) NOT NULL,
    raza varchar(10) NOT NULL,
    razgos varchar(40) NOT NULL,
    edad numeric(2, 0) NOT NULL,
    CONSTRAINT PK_MASCOTA PRIMARY KEY NONCLUSTERED (id_mascota)
    CONSTRAINT FK_PERSONAL_AYUDA_MASCOTA FOREIGN KEY (id_empleado)
        REFERENCES PERSONAL_AYUDA(id_empleado),
    CONSTRAINT FK_GUARDERIA_MASCOTA FOREIGN KEY (id_guarderia)
        REFERENCES GUARDERIA(id_guarderia),
    CONSTRAINT FK_USUARIO_COMUN_MASCOTA FOREIGN KEY (id_usuario)
        REFERENCES USUARIO_COMUN(id_usuario)
)
go

```



Embarcadero ER/Studio® Web Report Generation

MEDICAMENTO

Logical Entity Name	MEDICAMENTO
Default Table Name	MEDICAMENTO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla es un catalogo el cual va a almacenar los datos de los medicamentos.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE MEDICAMENTO(id_medicamento numeric(10, 0) IDENTITY(1,1) nombre_medicamento varchar(10) NOT NULL, costo money NOT NULL, cantidad numeric(4, 0) NOT NULL, CONSTRAINT PK_MEDICAMENTO PRIMARY KEY NONCLUSTERED (id_medicamento)) go</pre>
PostSQL	

MEDICAMENTO Columns

ColumnName	Domain	Datatype	NULL	Definition
id_medicamento		numeric(10, 0)	NO	Es la llave de la tabla y sera
nombre_medicamento		varchar(10)	NO	En esta columna se guardara
costo		money	NO	En esta columna se guardara
cantidad		numeric(4, 0)	NO	En esta columna se guardara

MEDICAMENTO Primary and Unique Keys

Key Name	Key Type	Keys
PK_MEDICAMENTO	PrimaryKey	id_medicamento



Embarcadero ER/Studio® Web Report Generation

Title

- └ P1.DM1
- + Logical
- + Equipo7_PETCO
- + Main Model
- + Tables
 - + ADMINISTRATIVO
 - + CAR_PROD
 - + CARRITO
 - + CATEGORIA_PRODI
 - + CENTRO
 - + COMPRA_CARRITO
 - + CONSULTA
 - + DETALLE CUENTA
 - + EMPLEADO
 - + ESTADO
 - + GUARDERIA
 - + MASCOTA
 - + MEDICAMENTO
 - + OFERTA
 - └ id_oferta
 - └ tipo
 - └ porcentaje
 - └ fecha_ini
 - └ fecha_fin
 - + PERSONAL_AYUDA
 - + PRODUCTO_CENTR
 - + PRODUCTO_TIEND
 - + RECETA
 - + REGISTRO_BRAZAL
 - + TELEFONO_PERSO
 - + USUARIO_COMUN
 - + VENDEDOR
 - + VENTA_FISICA
 - + VETERINARIO
- + Columns
- + Indexes
- + Foreign Keys

OFERTA	
Logical Entity Name	OFERTA
Default Table Name	OFERTA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla es un catalogo el cual va a almacenar las ofertas que tenemos
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE OFERTA(id_oferta numeric(10, 0) IDENTITY(1,1), tipo char(1) NOT NULL, porcentaje decimal(5, 2) NOT NULL, fecha_ini datetime NOT NULL, fecha_fin datetime NOT NULL, CONSTRAINT CK_CS2 CHECK (tipo in('N','L')), CONSTRAINT CK_CS3 CHECK (fecha_fin <40), CONSTRAINT PK_OFERTA PRIMARY KEY NONCLUSTERED (id_oferta)) go</pre>
PostSQL	

OFERTA Columns					
ColumnName	Domain	Datatype	NULL	Definition	
id_oferta		numeric(10, 0)	NO	Es la llave de la tabla y sera un cons	
tipo		char(1)	NO	Columna que va a almacenar si la of	
porcentaje		decimal(5, 2)	NO	Columna que va a almacenar el porc	
fecha_ini		datetime	NO	Columna que va a almacenar cuando	
fecha_fin		datetime	NO	Columna que va a almacenar cuando	



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - id_empleado
 - usuario
 - contraseña
 - edad
 - mascota
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND/
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

PERSONAL_AYUDA	
Logical Entity Name	PERSONAL_AYUDA
Default Table Name	PERSONAL_AYUDA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los empleados que son personal de ayuda.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE PERSONAL_AYUDA(id_empleado numeric(10, 0) NOT NULL, usuario varchar(10) NOT NULL, contraseña varchar(20) NOT NULL, edad numeric(2, 0) NOT NULL, mascota char(1) NOT NULL, CONSTRAINT CK_CS1 CHECK (mascota in('P','G')), CONSTRAINT PK_PERSONAL_AYUDA PRIMARY KEY NONCLUSTERED CONSTRAINT RefEMPLEADO05 FOREIGN KEY (id_empleado) REFERENCES EMPLEADO(id_empleado)) go</pre>
PostSQL	

PERSONAL_AYUDA Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_empleado		numeric(10, 0)	NO	Es la llave de la tabla que sera igual a la del empleado.
usuario		varchar(10)	NO	Columna que va a almacenar el nombre de usuario.
contraseña		varchar(20)	NO	Columna que va a almacenar la contraseña.
edad		numeric(2, 0)	NO	Columna que va a almacenar la edad.
mascota		char(1)	NO	Columna que va a almacenar el tipo de mascota.



Embarcadero ER/Studio® Web Report Generation

PRODUCTO_CENTRAL

Logical Entity Name	PRODUCTO_CENTRAL
Default Table Name	PRODUCTO_CENTRAL
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla es un catalogo el cual va a almacenar los datos de los productos.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE PRODUCTO_CENTRAL(id_producto numeric(10, 0) NOT NULL, id_categoria numeric(2, 0) NOT NULL, id_oferta numeric(10, 0) NULL, detalles varchar(40) NOT NULL, precio money NULL, CONSTRAINT PK_PRODUCTO_CENTRAL PRIMARY KEY NONCLUSTERED (id_producto) CONSTRAINT FK_OFERTA_PRODUCTO_CENTRAL FOREIGN KEY (id_oferta) REFERENCES OFERTA(id_oferta), CONSTRAINT FK_CATEGORIA_PRODUCTO_CENTRAL FOREIGN KEY (id_categoria) REFERENCES CATEGORIA_PRODUCTO(id_categoria)) go</pre>
PostSQL	

PRODUCTO_CENTRAL Columns

ColumnName	Domain	Datatype	NULL	Definition
id_producto		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
id_categoria		numeric(2, 0)	NO	Es la llave foranea de la tabla que se
id_oferta		numeric(10, 0)	YES	Es la llave foranea de la tabla que se
detalles		varchar(40)	NO	Columna que va a almacenar la des
precio		money	YES	Columna que va a almacenar el pre

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface. On the left, there is a tree view of the database schema under 'P1.DM1'. The 'Tables' node is expanded, showing various tables like ADMINISTRATIVO, CAR_PROD, CARRITO, etc. The 'PRODUCTO_TIENDA' table is selected.

PRODUCTO_TIENDA Entity Definition:

Logical Entity Name	PRODUCTO_TIENDA
Default Table Name	PRODUCTO_TIENDA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla es un catalogo el cual va a almacenar los datos de los productos.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE PRODUCTO_TIENDA(id_producto numeric(10, 0) NOT NULL, id_categoria numeric(2, 0) NOT NULL, id_oferta numeric(10, 0) NULL, detalles varchar(40) NOT NULL, precio money NULL, CONSTRAINT PK_PRODUCTO_TIENDA PRIMARY KEY NONCLUSTERED (id_producto) CONSTRAINT FK_OFERTA_PRODUCTO_TIENDA FOREIGN KEY (id_oferta) REFERENCES OFERTA(id_oferta), CONSTRAINT FK_CATEGORIA_PRODUCTO_TIENDA FOREIGN KEY (id_categoria) REFERENCES CATEGORIA_PRODUCTO(id_categoria)) go</pre>
PostSQL	

PRODUCTO_TIENDA Columns:

ColumnName	Domain	Datatype	NULL	Definition
id_producto		numeric(10, 0)	NO	Es la llave de la tabla que sera igual
id_categoria		numeric(2, 0)	NO	Es la llave foranea de la tabla que se
id_oferta		numeric(10, 0)	YES	Es la llave foranea de la tabla que se
detalles		varchar(40)	NO	Columna que va a almacenar la des
precio		money	YES	Columna que va a almacenar el pre



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND/
 - RECETA
 - id_receta
 - cantidad
 - id_consulta
 - id_medicamento
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

RECETA	
Logical Entity Name	RECETA
Default Table Name	RECETA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los medicamento que se le mandaran a la farmacia
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE RECETA(id_receta numeric(10, 0) IDENTITY(1,1), cantidad numeric(2, 0) NOT NULL, id_consulta numeric(10, 0) NOT NULL, id_medicamento numeric(10, 0) NOT NULL, CONSTRAINT PK_RECETA PRIMARY KEY NONCLUSTERED (id_receta), CONSTRAINT FK_RECETA_CONSULTA FOREIGN KEY (id_consulta) REFERENCES CONSULTA(id_consulta), CONSTRAINT FK_RECETA_MEDICAMENTO FOREIGN KEY (id_medicamento) REFERENCES MEDICAMENTO(id_medicamento)) go</pre>
PostSQL	

RECETA Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_receta		numeric(10, 0)	NO	Es la llave de la tabla y sera unica
cantidad		numeric(2, 0)	NO	En esta columna se guardara la cantidad
id_consulta		numeric(10, 0)	NO	Es la llave foranea de la tabla que hace referencia a la consulta
id_medicamento		numeric(10, 0)	NO	Es la llave foranea de la tabla que hace referencia al medicamento

RECETA Primary and Unique Keys				
id_receta				

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface. On the left, the logical model tree is displayed under 'P1.DM1'. A selected node, 'Equipo7_PETCO', contains a 'Main Model' folder which includes a 'Tables' folder. This 'Tables' folder contains numerous tables such as 'ADMINISTRATIVO', 'CAR_PROD', 'CARRITO', 'CATEGORIA_PROD', 'CENTRO', 'COMPRA_CARRI', 'CONSULTA', 'DETALLE CUENT', 'EMPLEADO', 'ESTADO', 'GUARDERIA', 'MASCOTA', 'MEDICAMENTO', 'OFERTA', 'PERSONAL_AYU', 'PRODUCTO_CEN', 'PRODUCTO_TIEI', 'RECETA', and 'REGISTRO_BRAZALETE'. The 'REGISTRO_BRAZALETE' table is expanded, showing its columns: 'id_registro', 'id_mascota', 'ritmoC', 'fecha/hora', 'temp', 'nivelOx', 'tipo_Comida', and 'comida_Comio'. Below the table list is a 'Foreign Keys' section.

REGISTRO_BRAZALETE

Logical Entity Name	REGISTRO_BRAZALETE
Default Table Name	REGISTRO_BRAZALETE
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de como es el comportamiento de la mascota.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE REGISTRO_BRAZALETE(id_registro numeric(10, 0) IDENTITY(1,1), id_mascota numeric(10, 0) NOT NULL, ritmoC char(10) NOT NULL, [fecha/hora] datetime NOT NULL, temp varchar(10) NOT NULL, nivelOx varchar(10) NOT NULL, tipo_Comida varchar(10) NOT NULL, comida_Comio varchar(10) NOT NULL, CONSTRAINT PK_REGISTRO_BRAZALETE PRIMARY KEY NONCLUSTERED (id_registro) CONSTRAINT FK_MASCOTA_REGISTRO_BRAZALETE FOREIGN KEY (id_mascota) REFERENCES MASCOTA(id_mascota))</pre>
PostSQL	

REGISTRO_BRAZALETE Columns

ColumnName	Domain	Datatype	NULL	Definition
id_registro		numeric(10, 0)	NO	Es la llave de la tabla y sera un constraint.
id_mascota		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla MASCOTA.
ritmoC		char(10)	NO	Columna que va a almacenar como ritmoC.
fecha/hora		datetime	NO	Columna que va a almacenar la fecha y hora.
temp		varchar(10)	NO	Columna que va a almacenar la temperatura.



Embarcadero ER/Studio® Web Report Generation



TELEFONO_PERSONAL	
Logical Entity Name	TELEFONO_PERSONAL
Default Table Name	TELEFONO_PERSONAL
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los numeros telefonicos de todos los empleados.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE TELEFONO_PERSONAL(id_telefono numeric(1, 0) NOT NULL, id_empleado numeric(10, 0) NOT NULL, telefono numeric(10, 0) NOT NULL, CONSTRAINT PK_TELEFONO_PERSONAL PRIMARY KEY NONCLUSTERED (id_telefono), CONSTRAINT FK_PERSONAL_AYUDA_TELEFONO_PERSONAL FOREIGN KEY (id_empleado) REFERENCES PERSONAL_AYUDA(id_empleado)) go</pre>
PostSQL	

TELEFONO_PERSONAL Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_telefono		numeric(1, 0)	NO	Es la llave de la tabla que sera igual a la de la tabla de empleados.
id_empleado		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla de empleados.
telefono		numeric(10, 0)	NO	Columna que va a almacenar el numero de telefono.

TELEFONO_PERSONAL Primary and Unique Keys		
Key Name	Key Type	Keys
PK_TELEFONO_PERSONAL	PrimaryKey	id_telefono

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface with the following details:

- Tables List:** A tree view on the left showing various tables: ADMINISTRATIVO, CAR_PROD, CARRITO, CATEGORIA_PROD, CENTRO, COMPROA_CARRI, CONSULTA, DETALLE CUENT, EMPLEADO, ESTADO, GUARDERIA, MASCOTA, MEDICAMENTO, OFERTA, PERSONAL_AYU, PRODUCTO_CEN, PRODUCTO_TIEI, RECETA, REGISTRO_BRAZ, TELEFONO_PER, and USUARIO_COMUN.
- USUARIO_COMUN Table Definition:** A detailed view of the USUARIO_COMUN table.

USUARIO_COMUN	
Logical Entity Name	USUARIO_COMUN
Default Table Name	USUARIO_COMUN
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los usuarios, como su nombre completo.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE USUARIO_COMUN(id_usuario numeric(10, 0) IDENTITY(1,1), nom varchar(10) NOT NULL, pat varchar(10) NOT NULL, mat varchar(10) NOT NULL, domicilio varchar(40) NOT NULL, CURP varchar(18) NOT NULL, genero char(1) NOT NULL, telefono numeric(10, 0) NOT NULL, numTarjeta numeric(19, 0) NOT NULL, vigencia date NOT NULL, correo varchar(20) NOT NULL, contraseña varchar(20) NOT NULL, CHECK (genero in('H', 'M')), CONSTRAINT PK_USUARIO_COMUN PRIMARY KEY NONCLUSTERED (id_usuario))</pre>
PostSQL	
- USUARIO_COMUN Columns:** A table showing the columns of the USUARIO_COMUN table.

ColumnName	Domain	Datatype	NULL	Definition
id_usuario		numeric(10, 0)	NO	Es la llave de la tabla y sera un cons
nom		varchar(10)	NO	Columna que va a almacenar el nom
pat		varchar(10)	NO	Columna que va a almacenar el ape
mat		varchar(10)	NO	Columna que va a almacenar el ape



Embarcadero ER/Studio® Web Report Generation

Title

- P1.DM1
 - Logical
 - Equipo7_PETCO
 - Main Model
 - Tables
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND/
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - id_empleado
 - sueldo
 - sueldo_total
 - comision
 - VENDEDOR
 - VENTA_FISICA
 - VETERINARIO
 - Columns
 - Indexes
 - Foreign Keys

VENDEDOR	
Logical Entity Name	VENDEDOR
Default Table Name	VENDEDOR
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos del vendedor como su sueldo y sus comisiones.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE VENDEDOR(id_empleado numeric(10, 0) NOT NULL, sueldo money NOT NULL, sueldo_total money NOT NULL, comision money NOT NULL, CONSTRAINT PK_VENDEDOR PRIMARY KEY NONCLUSTERED (id_empleado), CONSTRAINT RefEMPLEADO6 FOREIGN KEY (id_empleado) REFERENCES EMPLEADO(id_empleado)) go</pre>
PostSQL	

VENDEDOR Columns				
ColumnName	Domain	Datatype	NULL	Definition
id_empleado		numeric(10, 0)	NO	Es la llave de la tabla que sera igual a la de empleado.
sueldo		money	NO	Columna que va a almacenar cual es el sueldo.
sueldo_total		money	NO	Columna que va a almacenar el sueldo total.
comision		money	NO	Columna que va a almacenar la comision.

VENDEDOR Primary and Unique Keys				
<		>	<	>

Embarcadero ER/Studio® Web Report Generation

The screenshot shows the Embarcadero ER/Studio interface with the following details:

- Left Panel (Title):**
 - Title: P1.DM1
 - Logical Entity: Equipo7_PETCO
 - Main Model: Equipo7_PETCO
 - Tables:
 - ADMINISTRATIVO
 - CAR_PROD
 - CARRITO
 - CATEGORIA_PRODI
 - CENTRO
 - COMPRA_CARRITO
 - CONSULTA
 - DETALLE CUENTA
 - EMPLEADO
 - ESTADO
 - GUARDERIA
 - MASCOTA
 - MEDICAMENTO
 - OFERTA
 - PERSONAL_AYUDA
 - PRODUCTO_CENTR
 - PRODUCTO_TIEND/
 - RECETA
 - REGISTRO_BRAZAL
 - TELEFONO_PERSO
 - USUARIO_COMUN
 - VENDEDOR
 - VENTA_FISICA
 - id_ventaF
 - id_empleado
 - total
 - Columns
 - Indexes
 - Foreign Keys
- Entity Definition:**

VENTA_FISICA	
Logical Entity Name	VENTA_FISICA
Default Table Name	VENTA_FISICA
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de la venta fisica realizada por el empleado.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE VENTA_FISICA(id_ventaF numeric(10, 0) NOT NULL, id_empleado numeric(10, 0) NOT NULL, total money NOT NULL, CONSTRAINT PK_VENTA_FISICA PRIMARY KEY NONCLUSTERED (id_ventaF) CONSTRAINT FK_VENDEDOR_VENTA_FISICA FOREIGN KEY (id_empleado) REFERENCES VENDEDOR(id_empleado)) go</pre>
PostSQL	
- Columns:**

ColumnName	Domain	Datatype	NULL	Definition
id_ventaF		numeric(10, 0)	NO	Es la llave de la tabla que sera igual a la de la tabla VENDEDOR
id_empleado		numeric(10, 0)	NO	Es la llave foranea de la tabla que se refiere a la tabla VENDEDOR
total		money	NO	Columna que va a almacenar el costo de la venta
- Primary and Unique Keys:**

Key Name	Key Type	Keys
PK_VENTA_FISICA	PrimaryKey	id_ventaF



Embarcadero ER/Studio® Web Report Generation

VETERINARIO

Logical Entity Name	VETERINARIO
Default Table Name	VETERINARIO
Physical Only	NO
Do Not Generate	False
Owner	
Definition	Esta tabla almacena los datos de los veterinarios que podran atender a los clientes.
Note	
PreSQL	
DDL Code	<pre>CREATE TABLE VETERINARIO(id_empleado numeric(10, 0) NOT NULL, CEDULA numeric(10, 0) NOT NULL, especialidad varchar(40) NOT NULL, CONSTRAINT PK_VETERINARIO PRIMARY KEY NONCLUSTERED (id_empleado), CONSTRAINT RefEMPLEADO4 FOREIGN KEY (id_empleado) REFERENCES EMPLEADO(id_empleado)) go</pre>
PostSQL	

VETERINARIO Columns

ColumnName	Domain	Datatype	NULL	Definition
id_empleado		numeric(10, 0)	NO	Es la llave de la tabla con la que identifica al empleado.
CEDULA		numeric(10, 0)	NO	Columna que va a almacenar los datos de la cedula.
especialidad		varchar(40)	NO	Columna que va a almacenar en que se especializa el veterinario.

VETERINARIO Primary and Unique Keys

Key Name	Key Type	Keys
PK_VETERINARIO	PrimaryKey	id_empleado

Inserciones

```

490 select * from catalogos.ESTADO--1
491 select * from general.USUARIO_COMUN--2
492 select * from catalogos.CENTRO--3
493 select * from empleados.EMPLEADO--4
494 select * from empleados.ADMINISTRATIVO--5
495 select * from catalogos.oferta--6
496 select * from catalogos.CATEGORIA_PRODUCTO--7

```

Results Messages

	id_estado	nombre
1	1	Aguascalientes
2	2	Baja California
3	3	Baja California Sur
4	4	Campeche
5	5	Coahuila
6	6	Colima
7	7	Chiapas
8	8	Chihuahua
9	9	Ciudad de Mexico
10	10	Durango
11	11	Estado de Mexico

Query executed successfully.

```

490 select * from catalogos.ESTADO--1
491 select * from general.USUARIO_COMUN--2
492 select * from catalogos.CENTRO--3
493 select * from empleados.EMPLEADO--4
494 select * from empleados.ADMINISTRATIVO--5
495 select * from catalogos.oferta--6
496 select * from catalogos.CATEGORIA_PRODUCTO--7

```

Results Messages

	id_usuario	nom	pat	mat	domicilio	CURP	genero	telefono	numTargeta	vigencia	correo	contraseña
1	1	SONIA	ESTRELLA	RICOS	Yobain 09	SOER920113MDFSTR01	M	5562719823	223516283947362739	2023-08-27	sofaer@yahoo.com	agts1763dh26
2	2	JUAN	PEREZ	GARCIA	Calle del Sol 123	PERJ900525HDFPN02	H	5551234567	987654321098765432	2023-09-10	juan.perez@gmail.com	abc123xyz
3	3	MARIA	GUTIERREZ	SANCHEZ	Avenida Principal 456	GUTM981207MDFSNH03	M	5549873210	135792468024681357	2023-07-15	maria.gutierrez@gmail.com	ingeniero123
4	4	CARLOS	MARTINEZ	RODRIGUEZ	Av. Reforma 789	MARC950521MDFRDC04	H	5587654321	246813578024681379	2023-11-05	carlos.martinez@gmail.com	clave123
5	5	ANA	SANCHEZ	MARTINEZ	Calle 10 de Mayo 456	SAMA930705MDFSNH08	M	5567092837	192837465098273645	2023-10-15	ana.sanchez@gmail.com	abcyz123
6	6	PEDRO	GOMEZ	HERNANDEZ	Av. Juarez 234	GOHP970423HDFRND02	H	5572983476	20349876027384659	2023-09-01	pedro.gomez@gmail.com	password123
7	7	LAURA	FLORES	CRUZ	Calle del Bosque 56	FOLC880210MDFLRA05	M	5547819203	293847560928374651	2023-07-30	laura.flores@gmail.com	secreto456
8	8	RICARDO	RAMIREZ	Lopez	Paseo de la Reforma 789	RALR950713HDFRC03	H	5550982736	384756092837465109	2023-11-20	ricardo.ramirez@gmail.com	qwerty789
9	9	PAULA	RODRIGUEZ	GOMEZ	Av. Insurgentes 456	ROGP900601HDFDLA09	M	5562738492	273645098273645091	2023-09-05	paula.rodriguez@gmail.com	asdf123
10	10	MIGUEL	SANTOS	MARTINEZ	Calle del Camen 234	SAMM920317HDFNTG07	H	5549937261	49503627405923746	2023-07-20	miguel.santos@gmail.com	zxcv6789
11	11	ISABEL	GARCIA	HERNANDEZ	Calle de la Luna 78	GAHI910215HDFBLS06	M	5572968473	49583627364509236	2023-10-10	isabel.garcia@gmail.com	qazwsx456

Query executed successfully.

```

491 select * from general.USUARIO_COMUN--2
492 select * from catalogos.CENTRO--3
493 select * from empleados.EMPLEADO--4
494 select * from empleados.ADMINISTRATIVO--5
495 select * from catalogos.oferta--6
496 select * from catalogos.CATEGORIA_PRODUCTO--7

```

Results Messages

	id_centro	id_estado	oficina Regional
1	1	1	1
2	2	2	0
3	3	3	0
4	4	4	0
5	5	5	0
6	6	6	0
7	7	7	1
8	8	8	1
9	9	9	0
10	10	10	1
11	11	11	0

Query executed successfully.

```

491 | select * from general.USUARIO_COMUN--2
492 | select * from catalogos.CENTRO--3
493 | select * from empleados.EMPLEADO--4
494 | select * from empleados.ADMINISTRATIVO--5
495 | select * from catalogos.oferta--6
496 | select * from catalogos.CATEGORIA_PRODUCTO--7

```

150 %

Results Messages

	id_empleado	id_centro	nombre	paterno	materno	CURP	domicilio
1	1	1	Juan	Pérez	López	PELJ900101HDFXNN02	Calle 123, Ciudad de México
2	2	1	Maria	Rodríguez	García	ROGM980202MTZKXX03	Avenida Principal 456, Monterrey
3	3	1	Alejandro	Gómez	Hernández	GOHA880520GTOXXX04	Calle Central 789, Guanajuato
4	4	1	Luis	González	Martínez	GOML921205HDFXYZ05	Avenida 456, Puebla
5	5	7	Ana	López	Sánchez	LOSA880410MTZXX06	Calle 789, Guadalajara
6	6	7	Pedro	Martínez	Romero	MARP900608GTOXYZ07	Calle 456, León
7	7	7	Mónica	Hernández	García	HEGM890201HDFXYZ08	Avenida Principal 123, Querétaro
8	8	7	Carlos	Torres	Pérez	TOPC910405HDFXYZ09	Calle 789, Toluca
9	9	8	Sofia	Vargas	Jiménez	VAJS950101HDFXYZ10	Avenida 123, Cancún
10	10	8	Fernando	Ortega	Rojas	ORRF930701HDFXYZ11	Calle Principal 456, Mérida
11	11	8	Laura	Díaz	Morales	DIAM850202HDFXYZ12	Avenida Central 789, Monterrey

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\lan (57) Equipo7_PETCO 00:00:00 20 rows

```

494 | select * from empleados.ADMINISTRATIVO--5
495 | select * from catalogos.oferta--6
496 | select * from catalogos.CATEGORIA_PRODUCTO--7

```

150 %

Results Messages

	id_empleado	id_estado
1	1	1
2	5	7
3	9	8
4	13	10
5	17	24

```

495 | select * from catalogos.oferta--6
496 | select * from catalogos.CATEGORIA_PRODUCTO--7

```

150 %

Results Messages

	id_oferta	tipo	porcentaje	fecha_ini	fecha_fin
1	1	N	0.75	2023-06-07 00:00:00.000	2023-06-18 00:00:00.000
2	2	L	0.85	2023-07-05 00:00:00.000	2023-07-12 00:00:00.000
3	3	N	0.60	2023-06-15 00:00:00.000	2023-06-25 00:00:00.000
4	4	L	0.95	2023-07-01 00:00:00.000	2023-07-07 00:00:00.000
5	5	N	0.50	2023-06-10 00:00:00.000	2023-06-15 00:00:00.000

```

496 select * from catalogos.CATEGORIA_PRODUCTO--7
497 select * from catalogos.producto_central--8
498 select * from catalogos.producto_tienda--9
499 select * from empleados.vendedor--10
500 select * from empleados.PERSONAL_AYUDA--11
501 select * from catalogos.Telefono_Personal--12
502 select * from empleados.VETEDTMADTO--13

```

Results Messages

	id_categoria	descripcion
1	1	Juguetes
2	2	Premios
3	3	Alimentos
4	4	Utensilios varios

```

496 select * from catalogos.CATEGORIA_PRODUCTO--7
497 select * from catalogos.producto_central--8
498 select * from catalogos.producto_tienda--9
499 select * from empleados.vendedor--10
500 select * from empleados.PERSONAL_AYUDA--11
501 select * from catalogos.Telefono_Personal--12
502 select * from empleados.VETEDTMADTO--13

```

Results Messages

	id_producto	id_categoria	id_oferta	detalles	precio
1	1234567890	1	3	Peluche chinitante "SqueakyBuddy"	25.99
2	2345678901	1	2	Juguete interactivo "PawPuzzle"	12.99
3	3456789012	1	NULL	Maquillaje base de larga duración	18.99
4	4567890123	1	1	Pelota masticable "DuraBite"	29.99
5	5678901234	1	4	Juguete de cuerda "Tug-o-Fun"	39.99
6	6789912345	2	NULL	Galletas de pollo "Crunchy Bites"	19.99
7	7891123456	2	5	Snacks de salmón "Fishty Delights"	14.99
8	8901734567	2	1	Premios de cerdo "Tasty Rewards"	9.99
9	9412345678	2	3	Chucherías de manzana "Fruity Treats"	22.99
10	1234765890	2	2	Bantitas de buey ahumado "Savory Sticks"	49.99
11	2345678900	3	4	Croquetas premium "NutriPrime"	59.99

Query executed successfully.

```

497 select * from catalogos.producto_central--8
498 select * from catalogos.producto_tienda--9
499 select * from empleados.vendedor--10
500 select * from empleados.PERSONAL_AYUDA--11
501 select * from catalogos.Telefono_Personal--12
502 select * from empleados.VETEDTMADTO--13

```

Results Messages

	id_producto	id_categoria	id_oferta	detalles	precio
1	1234567890	1	3	Peluche chinitante "SqueakyBuddy"	25.99
2	2345678901	1	2	Juguete interactivo "PawPuzzle"	12.99
3	3456789012	1	NULL	Maquillaje base de larga duración	18.99
4	4567890123	1	1	Pelota masticable "DuraBite"	29.99
5	5678901234	1	4	Juguete de cuerda "Tug-o-Fun"	39.99
6	6789912345	2	NULL	Galletas de pollo "Crunchy Bites"	19.99
7	7891123456	2	5	Snacks de salmón "Fishty Delights"	14.99
8	8901734567	2	1	Premios de cerdo "Tasty Rewards"	9.99
9	9412345678	2	3	Chucherías de manzana "Fruity Treats"	22.99
10	1234765890	2	2	Bantitas de buey ahumado "Savory Sticks"	49.99
11	2345678900	3	4	Croquetas premium "NutriPrime"	59.99

```

498 | select * from catalogos.producto_tienda--9
499 | select * from empleados.vendedor--10
500 | select * from empleados.PERSONAL_AYUDA--11
501 | select * from catalogos.Telefono_Personal--12
502 | select * from empleados.VETERINARIO--13

```

150 % 4

Results Messages

	id_empleado	sueldo	sueldo_total	comision
1	2	5000.00	5000.00	0.00
2	6	5000.00	5000.00	0.00
3	10	5000.00	5000.00	0.00
4	13	5000.00	5000.00	0.00
5	18	5000.00	5000.00	0.00

Query executed successfully.

```

499 | select * from empleados.vendedor--10
500 | select * from empleados.PERSONAL_AYUDA--11
501 | select * from catalogos.Telefono_Personal--12
502 | select * from empleados.VETERINARIO--13
503 | select * from general.GUARDERIA--14
504 | select * from general.MASCOTA--15
505 | select * from general.CONSULTA--16
506 | select * from general.PEGISTRO_RRHH--17

```

150 % 5

Results Messages

	id_empleado	usuario	contraseña	edad	mascota
1	3	blue	gastode1234	20	P
2	7	red	xdcodece68	25	G
3	11	black	hbcedh5	24	P
4	14	lock	jxbt4	27	G
5	19	rino	kcsxel2	30	P

Query executed successfully.

```

500 | select * from empleados.PERSONAL_AYUDA--11
501 | select * from catalogos.Telefono_Personal--12
502 | select * from empleados.VETERINARIO--13
503 | select * from general.GUARDERIA--14
504 | select * from general.MASCOTA--15
505 | select * from general.CONSULTA--16
506 | select * from general.PEGISTRO_RRHH--17

```

150 % 6

Results Messages

	id_telefono	id_empleado	telefono
1		3	5561889234
2	2	7	6789054321
3	3	11	1234567890
4	4	14	9876543210
5	5	19	5432167890

Query executed successfully.

```

501 | select * from catalogos.Telefono_Personal--12
502 | select * from empleados.VETERINARIO--13
503 | select * from general.GUARDERIA--14
504 | select * from general.MASCOTA--15
505 | select * from general.CONSULTA--16
506 | select * from general.PEGISTRO_RRHH--17

```

150 % 7

Results Messages

	id_empleado	CEDULA	especialidad
1	4	6534782349	Cardiología
2	8	8126453789	Neurología
3	12	1083645382	Radiología
4	16	5271865235	Dermatología
5	20	1452783940	Traumatología

Query executed successfully.

502 SELECT * FROM empleados.VERETERINARIO--10
 503 select * from general.GUARDERIA--14
 504 select * from general.MASCOTA--15
 505 select * from general.CONSULTA--16
 506 select * from general.REGISTRO_BRAZALETE--17

	id_guarderia	fecha_in	fecha_fn	costo	id_centro
1	1	2023-06-10 00:00:00.000	2023-06-15 00:00:00.000	1500	1
2	2	2023-06-11 00:00:00.000	2023-06-16 00:00:00.000	1500	1
3	3	2023-06-12 00:00:00.000	2023-06-17 00:00:00.000	1500	1
4	4	2023-06-13 00:00:00.000	2023-06-18 00:00:00.000	1500	1
5	5	2023-06-14 00:00:00.000	2023-06-19 00:00:00.000	1500	7
6	6	2023-06-15 00:00:00.000	2023-06-20 00:00:00.000	1500	7
7	7	2023-06-16 00:00:00.000	2023-06-21 00:00:00.000	1500	7
8	8	2023-06-17 00:00:00.000	2023-06-22 00:00:00.000	1500	7
9	9	2023-06-18 00:00:00.000	2023-06-23 00:00:00.000	1500	8
10	10	2023-06-19 00:00:00.000	2023-06-24 00:00:00.000	1500	8
11	11	2023-06-20 00:00:00.000	2023-06-25 00:00:00.000	1500	8

Query executed successfully.

503 SELECT * FROM general.GUARDERIA--14
 504 select * from general.MASCOTA--15
 505 select * from general.CONSULTA--16
 506 select * from general.REGISTRO_BRAZALETE--17

	id_mascota	id_empleado	id_usuario	id_guarderia	nombre	especie	sexo	raza	razgos	edad
1	1	3	3	1	LUKE	perro	macho	pastor	negro con café	1
2	2	11	8	2	MAX	perro	macho	labrador	dorado	2
3	3	19	14	3	BELLA	perro	hembra	bulldog	blanco y negro	3
4	4	7	9	4	MININA	gato	hembra	siamés	negro	2
5	5	14	16	5	LUCY	gato	hembra	aligrado	dómestico	2

Query executed successfully.

504 SELECT * FROM general.MASCOTA--15
 505 select * from general.CONSULTA--16
 506 select * from general.REGISTRO_BRAZALETE--17

	id_consulta	id_empleado	fecha	hora	diagnostico	detalles	costo_total	id_mascota
1	1	4	2023-06-09	18:10:00.000000	Conjuntivitis	reposo	190.00	1
2	2	8	2023-06-16	14:30:00.000000	Dermatitis alérgica	revision	245.00	2
3	3	12	2023-06-20	15:36:00.000000	Otis externa	reposo	270.00	3
4	4	16	2023-06-24	16:30:00.000000	Gastroenteritis	asilado	200.00	4
5	5	20	2023-06-28	17:39:00.000000	Alergias cutáneas	reposo	280.00	5
6	6	4	2023-07-02	09:15:00.000000	Conjuntivitis	reposo	180.00	1
7	7	8	2023-07-05	11:45:00.000000	Dermatitis alérgica	revision	210.00	2
8	8	12	2023-07-08	13:20:00.000000	Otis externa	reposo	290.00	3
9	9	16	2023-07-11	15:50:00.000000	Gastroenteritis	asilado	250.00	4
10	10	20	2023-07-14	18:30:00.000000	Alergias cutáneas	reposo	170.00	5
11	11	4	2023-07-17	10:00:00.000000	Conjuntivitis	reposo	220.00	1

Query executed successfully.

505 SELECT * FROM general.CONSULTA--16
 506 select * from general.REGISTRO_BRAZALETE--17

	id_registro	id_mascota	rtmoC	fecha/hora	temp	nivelOx	tipo_Comida	comida_Comio
1	1	60-100	2023-06-28 17:39:00.000	37	80	croquetas	No	
2	2	60-100	2023-06-29 12:39:00.000	37	70	croquetas	No	
3	3	60-100	2023-06-27 13:39:00.000	36	80	croquetas	No	
4	4	60-100	2023-06-26 12:39:00.000	37	90	croquetas	No	
5	5	60-100	2023-06-25 13:39:00.000	36	80	croquetas	No	

Query executed successfully.

507 select * from catalogos.MEDICAMENTO--18
 508 select * from general.venta_fisica--19
 509 SELECT * from general.detalle_cuenta--20
 510 SELECT * from general.carrito--21
 511 select * from general.car_prod--22

	id_medicamento	nombre_medicamento	costo	cantidad
1	1	Ibuprofeno	700.00	500
2	2	Aspirina	100.00	500
3	3	Atrabica	150.00	500
4	4	Locell	180.00	500
5	5	Ibuprofeno	700.00	500
6	6	Anestesia	500.00	500

Query executed successfully.

```

507 | select * from catalogos.MEDICAMIENTO--18
508 | select * from general.venta_fisica--19
509 | SELECT * from general.detalle_cuenta--20
510 | SELECT * from general.carrito--21
511 | select * from general.car_prod--22

```

Results Messages

	id_ventaF	id_empleado	total
1	1	2	25.99
2	2	6	12.99
3	3	10	18.99
4	4	13	29.99
5	5	18	39.99
6	6	2	19.99
7	7	10	14.99
8	8	6	9.99
9	9	13	22.99
10	10	18	49.99

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 | 10 rows

```

509 | SELECT * from general.detalle_cuenta--20
510 | SELECT * from general.carrito--21
511 | select * from general.car_prod--22

```

Results Messages

	id_ventaF	id_producto	montoParcial	cantidad
1	1	1234567890	25.99	3
2	2	2345678901	12.99	2
3	3	3456789012	18.99	5
4	4	4567890123	29.99	1
5	5	5678901234	39.99	4
6	6	6789912345	19.99	3
7	7	7891123456	14.99	2
8	8	8901734567	9.99	1
9	9	9412345678	22.99	5
10	10	1234765890	49.99	3

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 | 10 rows

```

510 | SELECT * from general.carrito--21
511 | select * from general.car_prod--22

```

Results Messages

	id_cantito	id_usuario	cantidad
1	123456	1	3
2	234567	10	2
3	345678	5	4
4	456789	15	1
5	567890	7	5
6	678901	20	3
7	789012	2	3
8	890123	12	4
9	901234	8	2
10	102345	18	3
11	112456	3	5

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 | 26 rows

```

510 | SELECT * from general.carrito--21
511 | select * from general.car_prod--22

```

Results Messages

	id_producto	id_cantito	montoParcial	cantidad
1	1234567890	123456	25.99	3
2	2345678901	234567	12.99	2
3	3456789012	345678	18.99	4
4	4567890123	567890	29.99	1
5	5678901234	789012	39.99	5
6	6789912345	890123	19.99	3
7	7891123456	901234	14.99	3
8	8901734567	102345	9.99	4
9	9412345678	113456	22.99	2
10	1234765890	124567	49.99	3
11	2345678900	135678	25.99	5

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 | 20 rows

```

512 | select * from general.compra_carrito--23
513 | select * from general.receta --24

```

Results Messages

	id_ventaC	id_producto	id_cantito	id_usuario	detalles	montoParcial
1	1	1234567890	123456	1	Compra del producto	25.99
2	2	2345678901	234567	10	Compra del producto	12.99
3	3	3456789012	345678	5	Compra del producto	18.99
4	4	4567890123	567890	7	Compra del producto	29.99
5	5	5678901234	789012	2	Compra del producto	39.99
6	6	6789912345	890123	12	Compra del producto	19.99
7	7	7891123456	901234	8	Compra del producto	14.99
8	8	8901734567	102345	18	Compra del producto	9.99
9	9	9412345678	113456	3	Compra del producto	22.99
10	10	1234765890	124567	19	Compra del producto	49.99
11	11	2345678900	135678	6	Compra del producto	25.99

Query executed successfully.

DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 | 20 rows

```
513 | select * from general.receta --24
```

Results Messages

	id_receta	id_consulta	id_medicamento	cantidad
1	1	1	1	2
2	2	2	3	1
3	3	3	2	3
4	4	4	4	1
5	5	5	5	2
6	6	6	1	3
7	7	7	2	2
8	8	8	5	1
9	9	9	3	3
10	10	10	1	1
11	11	11	4	2

Query executed successfully. DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\ian (57) Equipo7_PETCO 00:00:00 15 rows

Procedimientos almacenados

Realice un procedimiento almacenado para registrar una mascota y su brazalete

```
39 begin tran
40 SELECT * from general.MASCOTA
41 SELECT * from general.REGISTRO_BRAZALETE
42 execute general.registro_mascota_brazalete 14, 16, 5, 'LUCY', 'gato',
43 'hembra', 'atigrado', 'doméstico', 2, '60-100', '36', '80', 'croquetas', 'No'
44 SELECT 'DESPUES DEL PROCEDIMIENTO'
45 SELECT * from general.MASCOTA
46 SELECT * from general.REGISTRO_BRAZALETE
47 rollback tran
```

Results Messages

	id_mascota	id_empleado	id_usuario	id_guarderia	nombre	especie	sexo	raza	razgos	edad
1	3	3	1	LUKE	perro	macho	pastor	negro con café	1	
2	11	8	2	MAX	perro	macho	labrador	dorado	2	
3	19	14	3	BELLA	perro	hembra	bulldog	blanco y negro	3	
4	7	9	4	MININA	gato	hembra	siamés	negro	2	
5	14	16	5	LUCY	gato	hembra	atigrado	doméstico	2	

	id_registro	id_mascota	ritmoC	fecha/hora	temp	nivelOx	tipo_Comida	comida_Comio
1	1	1	60-100	2023-06-28 17:39:00.000	37	80	croquetas	No
2	2	2	60-100	2023-06-29 12:39:00.000	37	70	croquetas	No
3	3	3	60-100	2023-06-27 13:39:00.000	36	80	croquetas	No
4	4	4	60-100	2023-06-26 12:39:00.000	37	90	croquetas	No
5	5	5	60-100	2023-06-25 13:39:00.000	36	80	croquetas	No

Realice un procedimiento almacenado para el actualizar de las lecturas en los brazaletes

```
--Prueba de funcionamiento
begin tran
SELECT * from general.REGISTRO_BRAZALETE
execute general.nuevo_registro_brazalete 5, 5, '60-100', '2024-06-28 17:39:00', '36', '80', 'croquetas', 'No'
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from general.REGISTRO_BRAZALETE
rollback tran
```

Results Messages

	id_registro	id_mascota	ritmoC	fecha/hora	temp	nivelOx	tipo_Comida	comida_Comio
1	1	1	60-100	2023-06-28 17:39:00.000	37	80	croquetas	No
2	2	2	60-100	2023-06-29 12:39:00.000	37	70	croquetas	No
3	3	3	60-100	2023-06-27 13:39:00.000	36	80	croquetas	No
4	4	4	60-100	2023-06-26 12:39:00.000	37	90	croquetas	No
5	5	5	60-100	2023-06-25 13:39:00.000	36	80	croquetas	No

Mensaje
1 Actualización realizada correctamente
(No column name)
1 DESPUES DEL PROCEDIMIENTO

	id_registro	id_mascota	ritmoC	fecha/hora	temp	nivelOx	tipo_Comida	comida_Comio
1	1	1	60-100	2023-06-28 17:39:00.000	37	80	croquetas	No
2	2	2	60-100	2023-06-29 12:39:00.000	37	70	croquetas	No
3	3	3	60-100	2023-06-27 13:39:00.000	36	80	croquetas	No
4	4	4	60-100	2023-06-26 12:39:00.000	37	90	croquetas	No
5	5	5	60-100	2023-06-25 13:39:00.000	36	80	croquetas	No

Realice un procedimiento almacenado para registrar una consulta

```

120  --Prueba de funcionamiento
121  begin tran
122  SELECT * from general.CONSULTA
123  execute general.RegistrarConsulta 16, '2023-06-28', '17:39:00', 'mordedura', 'reposo', 0.00, 5
124  SELECT 'DESPUES DEL PROCEDIMIENTO'
125  SELECT * from general.CONSULTA
126  rollback tran
127
128  --Si se prueba es necesario quitar la insercion del indexado
129  DBCC CHECKIDENT ('[consulta]', RESEED, 5);
130

```

Results Messages

	id_consulta	id_empleado	fecha	hora	diagnostico	detalles	costo_total	id_mascota
1	1	4	2023-06-09	18:10:00.000000	Conjuntivitis	reposo	190.00	1
2	2	8	2023-06-16	14:30:00.000000	Dermatitis alérgica	revision	245.00	2
3	3	12	2023-06-20	15:36:00.000000	Ottis externa	reposo	270.00	3
4	4	16	2023-06-24	16:30:00.000000	Gastroenteritis	asilado	200.00	4
5	5	20	2023-06-28	17:39:00.000000	Alergias cutáneas	reposo	280.00	5
6	6	4	2023-07-02	09:15:00.000000	Conjuntivitis	reposo	180.00	1
7	7	8	2023-07-05	11:45:00.000000	Dermatitis alérgica	revision	210.00	2
8	8	12	2023-07-08	13:20:00.000000	Ottis externa	reposo	290.00	3

Mensaje
1 Consulta registrada correctamente.

Query executed successfully. DESKTOP-10BROF9\PAISQL2019 ... DESKTOP-10BROF9\lan (57) Equipo7_PETCO

Realice un procedimiento almacenado para el registro y venta de medicamentos

```

162  --cambio de cantidad medicamento
163  begin tran
164  SELECT * from catalogos.MEDICAMENTO
165  execute catalogos.RegistrarMedicamento 1,1,'Ibuprofeno',700,2
166  SELECT 'DESPUES DEL PROCEDIMIENTO'
167  SELECT * from catalogos.MEDICAMENTO
168  rollback tran
169  --insertar medicamento
170  begin tran
171  SELECT * from catalogos.MEDICAMENTO

```

Results Messages

	id_medicamento	nombre_medicamento	costo	cantidad
1	1	Ibuprofeno	700.00	500
2	2	Aspirina	100.00	500
3	3	Atrabica	150.00	500
4	4	Loxcell	180.00	500
5	5	Ibuprofeno	700.00	500
6	6	Anestesia	500.00	500

Mensaje

1 Se cambio correctamente la cantidad de medicamento

(No column name)

1 DESPUES DEL PROCEDIMIENTO

Realice un procedimiento para dar de alta una venta, y otro para cancelar una venta en línea así como cancelar una venta física.

```

216
217  ----Producto con categoria existente
218  begin tran
219  SELECT * from catalogos.PRODUCTO_TIENDA
220  execute catalogos.AltaProducto 1,null, 12,1,'chicharrones fritos',15.00
221  SELECT 'DESPUES DEL PROCEDIMIENTO'
222  SELECT * from catalogos.PRODUCTO_TIENDA
223  rollback tran
224  ----Producto sin categoria
225  begin tran

```

Results Messages

	id_producto	id_categoria	id_oferta	detalles	precio
1	1234567890	1	3	Peluche chirriante "SqueakyBuddy"	25.99
2	2345678901	1	2	Juguete interactivo "PawPuzzle"	12.99
3	3456789012	1	NULL	Maquillaje base de larga duración	18.99
4	4567890123	1	1	Pelota masticable "DuraBite"	29.99
5	5678901234	1	4	Juguete de cuerda "Tug-o-Fun"	39.99
6	6789912345	2	NULL	Galletas de pollo "Crunchy Bites"	19.99
7	7891123456	2	5	Snacks de salmón "Fishy Delights"	14.99
8	8901734567	2	1	Premios de cordero "Tasty Rewards"	9.99

Mensaje

1 Se inserto la venta en centro y tienda

Query executed successfully. | DESKTOP-10BROF9\PAIS

```
253 ---Pedido existente
254 begin tran
255 select * from general.COMPRA_CARRITO
256 execute general.CancelarVentaLinea 123456
257 select * from general.COMPRA_CARRITO
258 rollback
259 --Pedido inexistente
260 begin tran
261 select * from general.COMPRA_CARRITO
262 execute general.CancelarVentaLinea 122256
```

150 %

Results Messages

	id_ventaC	id_producto	id_carrito	id_usuario	detalles	montoParcial
1	1	1234567890	123456	1	Compra del producto	25.99
2	2	2345678901	234567	10	Compra del producto	12.99
3	3	3456789012	345678	5	Compra del producto	18.99
4	4	4567890123	567890	7	Compra del producto	29.99
5	5	5678901234	789012	2	Compra del producto	39.99
6	6	6789912345	890123	12	Compra del producto	19.99
7	7	7891123456	901234	8	Compra del producto	14.99
8	8	8901734567	102345	18	Compra del producto	9.99

	mensaje
1	Se cancelo correctamente el pedido 123456

```
283 | select * from general.VENTA_FISICA
284 | ---Pedido existente
285 | begin tran
286 | select * from general.VENTA_FISICA
287 | execute dbo.CancelarVentaFisica 10
288 | select * from general.VENTA_FISICA
289 | rollback
290 | ---Pedido inexistente
291 | begin tran
292 | select * from general.VENTA_FISICA
```

150 % ▶

Results Messages

	id_ventaF	id_empleado	total
1	1	2	25.99
2	2	6	12.99
3	3	10	18.99
4	4	13	29.99
5	5	18	39.99
6	6	2	19.99
7	7	10	14.99
8	8	6	9.99

	mensaje
1	Se cancelo correctamente el pedido 10

✔ Query executed successfully.

Realice un procedimiento almacenado para borrar un usuario, recibe como parámetros el nombre de usuario.

```

324 begin tran
325 SELECT * from general.USUARIO_COMUN
326 execute dbo.EliminaUsuario 'hugo'
327 SELECT 'DESPUES DEL PROCEDIMIENTO'
328 SELECT * from general.USUARIO_COMUN
329 rollback tran
330
331 -- Crear procedimientos almacenados para las tareas de
332 CREATE or alter PROCEDURE ContratarPersonal

```

150 %

Results Messages

	id_usuario	nom	pat	mat	domicilio	CURP	genero	telefo
1	1	SONIA	ESTRELLA	RICOS	Yobain 09	SOER920113MDFSTR01	M	5562
2	2	JUAN	PÉREZ	GARCÍA	Calle del Sol 123	PERJ900625HDFPNC02	H	5551
3	3	MARÍA	GUTIÉRREZ	SÁNCHEZ	Avenida Principal 456	GUTM981207MDFSNH03	M	5549
4	4	CARLOS	MARTÍNEZ	RODRÍGUEZ	Av. Reforma 789	MARC890521MDFRDC04	H	5581
5	5	ANA	SÁNCHEZ	MARTÍNEZ	Calle 10 de Mayo 456	SAMA930705MDFSNH08	M	5561
6	6	PEDRO	GÓMEZ	HERNÁNDEZ	Av. Juárez 234	GOHP870423HDFRND02	H	5572
7	7	LAURA	FLORES	CRUZ	Calle del Bosque 56	FOCL880210MDFLRA05	M	5541
8	8	RICARDO	RAMÍREZ	LÓPEZ	Paseo de la Reforma 789	RALR950713HDFRCD03	H	5550

Mensaje

1 usuario eliminado

Query executed successfully.

Triggers

Trigger que calcula el costo total de la consulta

```

8 -- trigger que calcula el costo total de la consulta
9 select * from general.CONSULTA
10 select * from general.RECETA where id_consulta = 2
11 select * from catalogos.MEDICAMENTO
12 insert into general.receta values
13 (2,3,2)
14
15 --trigger que limita que un cuidador tenga maximo 5 mascotas que atender
16 TRIGGER T_NTO_general_MASCOTA ON general.MASCOTA FOR INSERT, UPDATE, DELETE AS
    BEGIN
        IF (SELECT COUNT(*) FROM general.MASCOTA WHERE id_empleado = (SELECT id_empleado FROM inserted)) >= 5
            RAISERROR('El cuidador ya tiene el máximo de 5 mascotas.', 16, 1);
    END

```

150 %

Results Messages

	id_consulta	id_empleado	fecha	hora	diagnóstico	detalles	costo_total	id_mascota
1	1	4	2023-06-09	18:10:00.0000000	Conjuntivitis	reposo	190.00	1
2	2	8	2023-06-16	14:30:00.0000000	Dermatitis alérgica	revision	600.00	2
3	3	12	2023-06-20	15:35:00.0000000	Otis externa	reposo	270.00	3
4	4	16	2023-06-24	16:30:00.0000000	Gastroenteritis	asilado	200.00	4
5	5	20	2023-06-28	17:39:00.0000000	Alergias cutáneas	reposo	280.00	5
6	6	4	2023-07-02	09:15:00.0000000	Conjuntivitis	reposo	180.00	1
7	7	8	2023-07-05	11:45:00.0000000	Dermatitis alérgica	revision	210.00	2
8	8	12	2023-07-08	13:20:00.0000000	Otis externa	reposo	290.00	3

	id_receta	id_consulta	id_medicamento	cantidad
1	2	2	3	1
2	16	2	3	2

Query executed successfully.

Trigger que limita que un cuidador tenga máximo 5 mascotas que atender

```

16 | SELECT * FROM general.MASCOTA
17 | begin tran
18 | declare @cont int = 5
19 | declare @i int = 1
20 | while @i <= @cont
21 | begin
22 | INSERT INTO general.MASCOTA(id_empleado,id_guarderia,id_usuario,nombre,especie,sexo,raza,razgos,edad) values
23 | (11,1,3,'LUKE','perro','macho','pastor','negro con café', 1)
24 | set @i = @i+1
25 | end
% Messages
(1 row affected)
No se pueden agregar más de 5 mascotas.
Msg 3609, Level 16, State 1, Line 22
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2023-06-12T22:06:31.9695914-06:00

```

Trigger que hace la cantidad máxima de veterinarios por centro es de 3

```

28 | BEGIN TRANSACTION;
29 | update empleados.EMPLEADO set id_centro = 1
30 | where id_empleado in (8, 12, 16)
31 | SELECT * from empleados.VETERINARIO v inner join empleados.EMPLEADO e on v.id_empleado = e.id_empleado
32 | rollback TRANSACTION;
33 |
34 |
35 | ---trigger actualiza el costo del monto parcial de car prod
36 |
50 % Messages
No se pueden agregar más de 3 veterinarios para el centro
Msg 3609, Level 16, State 1, Line 30
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2023-06-12T22:08:00.5927454-06:00

```

Trigger actualiza el costo del monto parcial de car prod

```

30 |
31 | begin tran
32 | INSERT INTO general.CAR_PROD (id_carrito, id_producto, montoParcial, cantidad)
33 | VALUES (179012, 1234567890, 100, 3);
34 | SELECT * from general.CAR_PROD where id_carrito = 179012
35 | rollback tran
36 |
37 | --- trigger que actualiza el monto parcial de compra carrito
38 | begin tran
150 % Results Messages


|   | id_producto | id_carrito | montoParcial | cantidad |
|---|-------------|------------|--------------|----------|
| 1 | 5677901234  | 179012     | 29.99        | 2        |
| 2 | 1234567890  | 179012     | 25.99        | 3        |


```

Trigger que actualiza el monto parcial de compra carrito

```

43 --- trigger que actualiza el monto parcial de compra carrito
44 begin tran
45 INSERT INTO general.car_prod (id_carrito, id_producto, montoparcial, cantidad)
46 VALUES
47 (135678, 4567890123, 4, 3)
48 select * from catalogos.PRODUCTO_CENTRAL
49 select * from general.CAR_PROD
50 select * from general.COMPRA_CARRITO
51 rollback tran
52

```

Results Messages

	id_producto	id_categoria	id_oferta	detalles	precio
1	1234567890	1	3	Peluche chinito "SqueakyBuddy"	25.99
2	2345678901	1	2	Juguete interactivo "PawPuzzle"	12.99
3	3456789012	1	NULL	Maquillaje base de larga duración	18.99
4	4567890123	1	1	Pelota masticable "DuraBite"	29.99
5	5678901234	1	4	Juguete de cuerda "Tug-o-Fun"	39.99
6	6789112345	2	NULL	Galletas de pollo "Crunchy Bites"	19.99
7	7891123456	2	5	Snacks de salmón "Fishy Delights"	14.99
8	8901734567	2	1	Premios de cordero "Tasty Rewards"	9.99

	id_producto	id_cant	montoParcial	cantidad
1	1234567890	123456	25.99	3
2	2345678901	224567	12.99	2

Query executed successfully.

Trigger que aplica descuento y sumatoria de la venta física

```

53 --- trigger que aplica descuento y sumatoria de la venta fisica
54 begin tran
55 INSERT INTO general.venta_fisica (id_empleado, total)
56 VALUES
57 (2, 25.99)
58 INSERT INTO general.detalle_cuenta (id_ventaF, id_producto, montoparcial, cantidad)
59 VALUES
60 (11, 7891123456, 25.99, 3)
61 select * from catalogos.OFERTA
62 select * from catalogos.PRODUCTO TIENDA
63

```

Results Messages

	id_oferta	tip	porcentaje	fecha_ini	fecha_fin
1	1	N	0.75	2023-06-07 00:00:00.000	2023-06-18 00:00:00.000
2	2	L	0.85	2023-07-05 00:00:00.000	2023-07-12 00:00:00.000
3	3	N	0.60	2023-06-15 00:00:00.000	2023-06-25 00:00:00.000
4	4	L	0.95	2023-07-01 00:00:00.000	2023-07-07 00:00:00.000
5	5	N	0.50	2023-06-10 00:00:00.000	2023-06-15 00:00:00.000

	id_producto	id_categoria	id_oferta	detalles	precio
1	1234567890	1	3	Peluche chinito "SqueakyBuddy"	25.99
2	2345678901	1	2	Juguete interactivo "PawPuzzle"	12.99
3	3456789012	1	NULL	Maquillaje base de larga duración	18.99
4	4567890123	1	1	Pelota masticable "DuraBite"	29.99
5	5678901234	1	4	Juguete de cuerda "Tug-o-Fun"	39.99

Query executed successfully.

Trigger que pone la comision al vendedor

```

73 (2, 25.99)
74 INSERT INTO general.detalle_cuenta (id_ventaF, id_producto, montoparcial, cantidad)
75 VALUES
76 (10, 4567890123, 25.99, 3)
77 select * from general.VENTA_FISICA
78 select * from general.DETALLE_CUENTA
79 select * from empleados.VENDEDOR
80 rollback tran
81
82 --trigger que valida la existencia del estado antes de insertarse

```

Results Messages

	id_empleado	sueldo	sueldo_total	comision
4	4	4567890123	25.99	1
5	5	5678901234	39.99	4
6	6	789112345	19.99	3
7	7	7891123456	14.99	2
8	8	8901734567	9.99	1

	id_empleado	sueldo	sueldo_total	comision
1	2	5000.00	5003.8985	3.8985
2	6	5000.00	5000.00	0.00
3	10	5000.00	5000.00	0.00
4	13	5000.00	5000.00	0.00
5	18	5000.00	5000.00	0.00

Query executed successfully.

Trigger que valida la existencia del estado antes de insertarse un administrativo

```

--trigger que valida la existencia del estado antes de insertarse un administrativo
begin tran
insert into empleados ADMINISTRATIVO values (18, 33)
rollback tran

Completion time: 2023-06-12T22:14:48.4036884-06:00

150 % < > Messages
Msg 50000, Level 16, State 1, Procedure tr_verificar_estado_administrativo, Line 8 [Batch Start Line 82]
El estado especificado no existe en la tabla ESTADO.
Msg 3609, Level 16, State 1, Line 84
The transaction ended in the trigger. The batch has been aborted.

150 % < > Results Messages
DESKTOP-10BROF9\PAISQL2019 ... | DESKTOP-10BROF9\lan (56) | Equipo7_PETCO | 00:00:00 | 0 rows

```

Estadísticas

1. El tipo de mascota que más se deja en la guardería para su cuidado (tipo y cantidad).

```

8
9  SELECT * from general.masTipoMascota
10
11 --2 En qué época del año se deja un mayor nú
150 % < > Results Messages


|   | especie | cantidad |
|---|---------|----------|
| 1 | perrro  | 3        |


```

2. En qué época del año se deja un mayor número de mascotas en la guardería

```

39 begin tran
40 update general.GUARDERIA set fecha_ini = '2022-03-21' where id_guarderia = 20
41 SELECT * from general.GUARDERIA where id_guarderia = 20
42 select * from general.masEpoca
43 rollback tran
44
150 % < > Results Messages


|   | id_guarderia | fecha_ini               | fecha_fin               | costo  | id_centro |
|---|--------------|-------------------------|-------------------------|--------|-----------|
| 1 | 20           | 2022-03-21 00:00:00.000 | 2023-07-04 00:00:00.000 | 141000 | 24        |



|   | Epoca  | cantidad |
|---|--------|----------|
| 1 | Verano | 19       |


```

3. Listado de las mascotas y su dueño datos generales, número de días en la guardería, datos generales de ambos. EL reporte se obtiene en un periodo de fechas

```

58 | SELECT * from general.datosMascotaUsuario
59 |
60 |--4. Enfermedades más frecuentes (5), total de animales con dicha enfermedad
61 |--ordenados de mayor a menor
62 |
150 % < > Results Messages


|   | nombre_usuario           | telefono   | domicilio               | nombre_mascota | especie | sexo_mascota | raza     | razgos         | edad_mascota | num_dias |
|---|--------------------------|------------|-------------------------|----------------|---------|--------------|----------|----------------|--------------|----------|
| 1 | MARÍA GUTIÉRREZ SÁNCHEZ  | 5549873210 | Avenida Principal 456   | LUKE           | perro   | macho        | pastor   | negro con café | 1            | 5        |
| 2 | RICARDO RAMÍREZ LÓPEZ    | 5550982736 | Paseo de la Reforma 789 | MAX            | perro   | macho        | labrador | dorado         | 2            | 5        |
| 3 | JOSÉ GÓMEZ CRUZ          | 5582736459 | Av. Zaragoza 456        | BELLA          | perro   | hembra       | bulldog  | blanco y negro | 3            | 5        |
| 4 | PAULA RODRÍGUEZ GÓMEZ    | 5562738492 | Av. Insurgentes 456     | MININA         | gato    | hembra       | siamés   | negro          | 2            | 5        |
| 5 | ANTONIO SÁNCHEZ MARTÍNEZ | 5554321769 | Calle del Sol 345       | LUCY           | gato    | hembra       | atigrado | doméstico      | 2            | 5        |


```

4. Enfermedades más frecuentes (5), total de animales con dicha enfermedad ordenados de mayor a menor

```

69 | SELECT * from general.enfermedadesMayor order by cantidad desc
70 |
71 |--REPORTE VETERINARIA
72 |
150 % < > Results Messages


|   | diagnóstico         | cantidad |
|---|---------------------|----------|
| 1 | Alergias cutáneas   | 3        |
| 2 | Conjuntivitis       | 3        |
| 3 | Dermatitis alérgica | 3        |
| 4 | Gastroenteritis     | 3        |
| 5 | Otitis externa      | 3        |


```

1. Gastos de cada mascota, datos de la mascota, fechas y días de estancia, descripción de gatos y costo.

```

85 |
86 | SELECT * from general.INFO_GASTO_X_MASCOTA
87 |
88 |--2. Listado con datos generales de las consultas por periodo
89 |--incluyendo veterinario que lo atendió, diagnóstico y medic
90 |
150 % < > Results Messages


|   | nombre_mascota | especie | raza     | edad | razgos         | costo_veterinario | días_estancia | costo_estancia | total_gastado |
|---|----------------|---------|----------|------|----------------|-------------------|---------------|----------------|---------------|
| 1 | BELLA          | perro   | bulldog  | 3    | blanco y negro | 740.00            | 5             | 1500           | 2240.00       |
| 2 | LUCY           | gato    | atigrado | 2    | doméstico      | 650.00            | 5             | 1500           | 2150.00       |
| 3 | LUKE           | perro   | pastor   | 1    | negro con café | 590.00            | 5             | 1500           | 2090.00       |
| 4 | MAX            | perro   | labrador | 2    | dorado         | 715.00            | 5             | 1500           | 2215.00       |
| 5 | MININA         | gato    | siamés   | 2    | negro          | 740.00            | 5             | 1500           | 2240.00       |


```

2. Listado con datos generales de las consultas por periodos de tiempo, incluyendo veterinario que lo atendió, diagnóstico y medicamento.

```
101 SELECT * from general.datos_consulta order by fecha  
102  
103 --3. Inventario de medicamentos, con su costo por centro  
104  
105 --4. Enfermedades más frecuentes, por centro, ordenadas  
106  
107 CREATE OR ALTER VIEW general_enfermedad_x_centro
```

	nombre_veterinario	fecha	diagnostico	nombre_medicamento
1	Luis González Martínez	2023-06-09	Conjuntivitis	Ibuprofeno
2	Carlos Torres Pérez	2023-06-16	Dermatitis alérgica	Atrabica
3	Roberto Silva Gómez	2023-06-20	Otitis externa	Aspirina
4	Miguel Morales López	2023-06-24	Gastroenteritis	Loxcell
5	Héctor Jiménez Silva	2023-06-28	Alergias cutáneas	Ibuprofeno
6	Luis González Martínez	2023-07-02	Conjuntivitis	Ibuprofeno
7	Carlos Torres Pérez	2023-07-05	Dermatitis alérgica	Aspirina
8	Roberto Silva Gómez	2023-07-08	Otitis externa	Ibuprofeno
9	Miguel Morales López	2023-07-11	Gastroenteritis	Atrabica
10	Héctor Jiménez Silva	2023-07-14	Alergias cutáneas	Ibuprofeno
11	Luis González Martínez	2023-07-17	Conjuntivitis	Loxcell

4. Enfermedades más frecuentes, por centro, ordenadas de mayor a menor

```
115  SELECT * from general.enfermedad_x_centro order by id_centro  
116  
117 --5. Reporte de ventas de medicamentos en un periodo de tiempo, :  
118 --cantidad y monto total
```

Reporte de Enfermedades			
	id_centro	enfermedad	cantidad
1	1	Conjuntivitis	3
2	1	Dematitis alérgica	3
3	1	Gastroenteritis	3
4	1	Otitis extema	3
5	7	Alergias cutáneas	3

5. Reporte de ventas de medicamentos en un periodo de tiempo, incluyendo cantidad y monto total

```

128 | SELECT * from general.reporte_venta_medicamento order by fecha
129 |
130 | --REPORTE VENTAS
150 % < >
Results Messages

```

	fecha	nombre_medicamento	cantidad	monto_total
1	2023-06-09	Ibuprofeno	2	1400.0000
2	2023-06-16	Atrabica	1	150.0000
3	2023-06-20	Aspirina	3	300.0000
4	2023-06-24	Loxcell	1	180.0000
5	2023-06-28	Ibuprofeno	2	1400.0000
6	2023-07-02	Ibuprofeno	3	2100.0000
7	2023-07-05	Aspirina	2	200.0000
8	2023-07-08	Ibuprofeno	1	700.0000
9	2023-07-11	Atrabica	3	450.0000
10	2023-07-14	Ibuprofeno	1	700.0000
11	2023-07-17	Loxcell	2	360.0000

1. El centro con mayor número de ventas en un periodo de tiempo. (Separar las ventas en línea de las ventas físicas de cada estado).

```

145
146 | SELECT * from general.REPORTE_VENTAS_FISICAS
147
148 | --2. Los artículos más vendidos y los menos ver
149 | --Ventas físicas
150 % < >
Results Messages

```

	estado	id_centro	ventas_fisicas
1	San Luis Potosí	24	48

2. Los artículos más vendidos y los menos vendidos por categoría. Ventas físicas

```

184 | SELECT * from general.REPORTE_MAS_VENDIDOF()
185 |
186 | --Ventas virtuales
187 | CREATE OR ALTER VIEW general.datos_ventasV

```

Results

	categoria	producto	cantidad
1	Juguetes	Juguete de cuerda "Tug-o-Fun"	29
2	Juguetes	Juguete de cuerda "Tug-o-Fun"	29
3	Premios	Banitas de buey ahumado "Savory Sticks"	29
4	Premios	Banitas de buey ahumado "Savory Sticks"	29
5	Alimentos	Alimento seco "Tiny Bites"	29
6	Alimentos	Alimento seco "Tiny Bites"	29
7	Utensilios varios	Bebedero automático "HydroFlow"	29
8	Utensilios varios	Bebedero automático "HydroFlow"	29

Ventas virtuales

```

221 | SELECT * from general.REPORTE_MAS_VENDIDOV()
222 |
223 | --3. Época en la que más se vende. Época del año y monto tot
224 |
225 | --4. Los 5 empleados con mayor comisión, este reporte se obt
226 | --mensual
227 |

```

Results

	categoria	producto	cantidad
1	Juguetes	Juguete de cuerda "Tug-o-Fun"	5
2	Juguetes	Pelota masticable "DuraBite"	1
3	Premios	Premios de cordero "Tasty Rewards"	4
4	Premios	Chucherías de manzana "Fruity Treats"	2
5	Alimentos	Croquetas premium "NutriPrime"	5
6	Alimentos	Comida húmeda "Delicious Purnfection"	1
7	Utensilios varios	Bebedero automático "HydroFlow"	8
8	Utensilios varios	Dispensador de bolsas "CleanPaws"	1

Crearbase código

```

/*
 * EQUIPO 7 GUARDERIA DE MASCOTAS
 * CORIA MARTINEZ GUSTAVO
 * ORONZOR MONTES MANASES LEONEL
 * PANG ARAIZAGA IAN
 */

--drop database Equipo7_PETCO

create database Equipo7_PETCO
GO

USE Equipo7_PETCO
GO

/*
 * ESQUEMAS
 */
CREATE SCHEMA catalogos;
GO
CREATE SCHEMA empleados;
GO
CREATE SCHEMA general;
GO

/*
 * TABLE: ADMINISTRATIVO
 */
CREATE TABLE empleados.ADMINISTRATIVO(
    id_empleado numeric(10, 0) NOT NULL,
    id_estado numeric(10, 0) NOT NULL,
    CONSTRAINT PK_ADMINISTRATIVO PRIMARY KEY NONCLUSTERED (id_empleado)
)
go

IF OBJECT_ID('empleados.ADMINISTRATIVO') IS NOT NULL
    PRINT '<<< CREATED TABLE ADMINISTRATIVO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE ADMINISTRATIVO >>>'
go

/*

```

```

/* TABLE: Categoria_producto
*/
CREATE TABLE catalogos.CATEGORIA_PRODUCTO(
    id_categoria numeric(2,0) IDENTITY(1,1),
    descripcion varchar(40)
    Constraint PK_CATEGORIA_PRODUCTO Primary key NONCLUSTERED
(id_categoria)
)

IF OBJECT_ID('catalogos.CATEGORIA_PRODUCTO') IS NOT NULL
    PRINT '<<< CREATED TABLE CATEGORIA_PRODUCTO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CATEGORIA_PRODUCTO >>>'
go

/*
* TABLE: CAR_PROD
*/
CREATE TABLE general.CAR_PROD(
    id_producto numeric(10, 0) NOT NULL,
    id_carrito numeric(10, 0) NOT NULL,
    montoParcial money NOT NULL,
    cantidad numeric(10, 0) NOT NULL,
    CONSTRAINT PK_CAR_PROD PRIMARY KEY NONCLUSTERED (id_producto,
    id_carrito)
)
go

IF OBJECT_ID('general.CAR_PROD') IS NOT NULL
    PRINT '<<< CREATED TABLE CAR_PROD >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CAR_PROD >>>'
go

/*
* TABLE: CARRITO
*/
CREATE TABLE general.CARRITO(
    id_carrito numeric(10, 0) NOT NULL,
    id_usuario numeric(10, 0) NOT NULL,
    cantidad numeric(10, 0) NOT NULL,
    CONSTRAINT PK_CARRITO PRIMARY KEY NONCLUSTERED (id_carrito)
)

```

```
go
```

```
IF OBJECT_ID('general.CARRITO') IS NOT NULL
    PRINT '<<< CREATED TABLE CARRITO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CARRITO >>>'
go
```

```
/*
 * TABLE: CENTRO
 */
```

```
CREATE TABLE catalogos.CENTRO(
    id_centro      numeric(10, 0)  IDENTITY(1,1),
    id_estado      numeric(10, 0)  NOT NULL,
    oficinaRegional numeric(1, 0)  NOT NULL
        CONSTRAINT CK_CS6 CHECK (oficinaRegional in (0,1)),
    CONSTRAINT PK_CENTRO PRIMARY KEY NONCLUSTERED (id_centro)
)
go
```

```
IF OBJECT_ID('catalogos.CENTRO') IS NOT NULL
    PRINT '<<< CREATED TABLE CENTRO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CENTRO >>>'
go
```

```
/*
 * TABLE: COMPRA_CARRITO
 */
```

```
CREATE TABLE general.COMPRA_CARRITO(
    id_ventaC      numeric(10, 0)  IDENTITY(1,1) NOT NULL,
    id_producto    numeric(10, 0)  NOT NULL,
    id_carrito     numeric(10, 0)  NOT NULL,
    id_usuario     numeric(10, 0)  NOT NULL,
    detalles       varchar(40)    NOT NULL,
    montoParcial   money         NOT NULL,
    CONSTRAINT PK_COMPRA_CARRITO PRIMARY KEY NONCLUSTERED (id_ventaC)
)
go
```

```

IF OBJECT_ID('general.COMPRA_CARRITO') IS NOT NULL
    PRINT '<<< CREATED TABLE COMPRA_CARRITO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE COMPRA_CARRITO >>>'
go

/*
 * TABLE: CONSULTA
 */

CREATE TABLE general.CONSULTA(
    id_consulta numeric(10, 0) IDENTITY(1,1) NOT NULL,
    id_empleado numeric(10, 0) NOT NULL,
    fecha date NOT NULL,
    hora time(7) NOT NULL,
    diagnostico varchar(40) NOT NULL,
    detalles varchar(40) NOT NULL,
    costo_total money NOT NULL,
    id_mascota numeric(10, 0) NOT NULL,
    CONSTRAINT PK_CONSULTA PRIMARY KEY NONCLUSTERED (id_consulta)
)
go

IF OBJECT_ID('general.CONSULTA') IS NOT NULL
    PRINT '<<< CREATED TABLE CONSULTA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CONSULTA >>>'
go

/*
 * TABLE: RECETA
 */

CREATE TABLE general.RECETA(
    id_receta numeric(10, 0) IDENTITY(1,1) NOT NULL,
    id_consulta numeric(10, 0) NOT NULL,
    id_medicamento numeric(10, 0) NOT NULL,
    cantidad numeric(2, 0) NOT NULL
    CONSTRAINT PK_RECETA PRIMARY KEY NONCLUSTERED (id_receta)
)
go

IF OBJECT_ID('general.RECETA') IS NOT NULL
    PRINT '<<< CREATED TABLE RECETA >>>'
ELSE

```

```

PRINT '<<< FAILED CREATING TABLE RECETA >>>'
go

/*
 * TABLE: DETALLE CUENTA
 */

CREATE TABLE general.DETALLE CUENTA(
    id_ventaF      numeric(10, 0)  NOT NULL,
    id_producto    numeric(10, 0)  NOT NULL,
    montoParcial   money         NOT NULL,
    cantidad       numeric(10, 0)  NOT NULL,
    CONSTRAINT PK_DETALLE CUENTA PRIMARY KEY NONCLUSTERED (id_ventaF,
    id_producto)
)
go

IF OBJECT_ID('general.DETALLE CUENTA') IS NOT NULL
    PRINT '<<< CREATED TABLE DETALLE CUENTA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE DETALLE CUENTA >>>'
go

/*
 * TABLE: EMPLEADO
 */

CREATE TABLE empleados.EMPLEADO(
    id_empleado    numeric(10, 0)  IDENTITY(1,1),
    id_centro     numeric(10, 0)  NOT NULL,
    nombre        varchar(20)    NOT NULL,
    paterno       varchar(15)    NOT NULL,
    materno       varchar(15)    NOT NULL,
    CURP          varchar(18)    NOT NULL,
    domicilio     varchar(40)    NOT NULL,
    CONSTRAINT PK_EMPLEADO PRIMARY KEY NONCLUSTERED (id_empleado),
    CONSTRAINT UQ_curpE unique (curp)
)
go

IF OBJECT_ID('empleados.EMPLEADO') IS NOT NULL
    PRINT '<<< CREATED TABLE EMPLEADO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE EMPLEADO >>>'
```

```

go

/*
 * TABLE: ESTADO
 */

CREATE TABLE catalogos.ESTADO(
    id_estado numeric(10, 0) IDENTITY(1,1),
    nombre     varchar(40) NOT NULL,
    CONSTRAINT PK_ESTADO PRIMARY KEY NONCLUSTERED (id_estado)
)
go

IF OBJECT_ID('catalogos.ESTADO') IS NOT NULL
    PRINT '<<< CREATED TABLE ESTADO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE ESTADO >>>'
go

/*
 * TABLE: GUARDERIA
 */

CREATE TABLE general.GUARDERIA(
    id_guarderia numeric(10, 0) IDENTITY(1,1),
    fecha_ini    datetime NOT NULL,
    fecha_fin    datetime NOT NULL,
    costo         as datediff(day,fecha_ini,fecha_fin)*300,
    id_centro    numeric(10, 0) NOT NULL,
    CONSTRAINT PK_GUARDERIA PRIMARY KEY NONCLUSTERED (id_guarderia)
)
go

IF OBJECT_ID('general.GUARDERIA') IS NOT NULL
    PRINT '<<< CREATED TABLE GUARDERIA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE GUARDERIA >>>'
go

/*
 * TABLE: MASCOTA
 */

CREATE TABLE general.MASCOTA(
    id_mascota numeric(10, 0) IDENTITY(1,1),
    id_empleado numeric(10, 0) NOT NULL,

```

```

id_usuario numeric(10, 0) NOT NULL,
id_guarderia numeric(10, 0) NOT NULL,
nombre varchar(10) NOT NULL,
especie varchar(10) NOT NULL,
sexo varchar(10) NOT NULL,
raza varchar(10) NOT NULL,
razgos varchar(40) NOT NULL,
edad numeric(2, 0) NOT NULL,
CONSTRAINT PK_MASCOTA PRIMARY KEY NONCLUSTERED (id_mascota)
)
go

```

```

IF OBJECT_ID('general.MASCOTA') IS NOT NULL
    PRINT '<<< CREATED TABLE MASCOTA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE MASCOTA >>>'
go

/*
 * TABLE: MEDICAMENTO
 */

```

```

CREATE TABLE catalogos.MEDICAMENTO(
    id_medicamento numeric(10, 0) IDENTITY(1,1),
    nombre_medicamento varchar(10) NOT NULL,
    costo money NOT NULL,
    cantidad numeric(4, 0) NOT NULL
CONSTRAINT PK_MEDICAMENTO PRIMARY KEY NONCLUSTERED
(id_medicamento)
)
go

```

```

IF OBJECT_ID('catalogos.MEDICAMENTO') IS NOT NULL
    PRINT '<<< CREATED TABLE MEDICAMENTO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE MEDICAMENTO >>>'
go

/*
 * TABLE: OFERTA
 */

```

```

CREATE TABLE catalogos.OFERTA(
    id_oferta numeric(10, 0) IDENTITY(1,1),

```

```

    tipo      char(1)      NOT NULL
              CONSTRAINT CK_CS2 CHECK (tipo in('N','L') ),
    porcentaje  decimal(5,2)  NOT NULL,
    fecha_ini   datetime     NOT NULL,
    fecha_fin    datetime     NOT NULL,
    CONSTRAINT PK_OFERTA PRIMARY KEY NONCLUSTERED (id_oferta),
    CONSTRAINT CK_CS3 CHECK (datediff(day,fecha_ini,fecha_fin) <40)
)
go

```

```

IF OBJECT_ID('catalogos.OFERTA') IS NOT NULL
    PRINT '<<< CREATED TABLE OFERTA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE OFERTA >>>'
go

```

```

/*
 * TABLE: PERSONAL_AYUDA
 */

```

```

CREATE TABLE empleados.PERSONAL_AYUDA(
    id_empleado numeric(10, 0)  NOT NULL,
    usuario      varchar(10)    NOT NULL,
    contraseña   varchar(20)    NOT NULL,
    edad         numeric(2, 0)   NOT NULL,
    mascota      char(1)       NOT NULL
              CONSTRAINT CK_CS1 CHECK (mascota in('P','G') ),
    CONSTRAINT PK_PERSONAL_AYUDA PRIMARY KEY NONCLUSTERED
(id_empleado),
    CONSTRAINT UQ_USUARIO unique (usuario)
)
go

```

```

IF OBJECT_ID('empleados.PERSONAL_AYUDA') IS NOT NULL
    PRINT '<<< CREATED TABLE PERSONAL_AYUDA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE PERSONAL_AYUDA >>>'
go

```

```

/*
 * TABLE: PRODUCTO_CENTRAL
 */

```

```

CREATE TABLE catalogos.PRODUCTO_CENTRAL(
    id_producto  numeric(10, 0)  NOT NULL,
    id_categoria numeric(2,0)      NOT NULL,

```

```

id_oferta numeric(10, 0) NULL,
detalles varchar(40) NOT NULL,
precio money NOT NULL,
CONSTRAINT PK_PRODUCTO_CENTRAL PRIMARY KEY NONCLUSTERED
(id_producto)
)
go

```

```

IF OBJECT_ID('catalogos.PRODUCTO_CENTRAL') IS NOT NULL
    PRINT '<<< CREATED TABLE PRODUCTO_CENTRAL >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE PRODUCTO_CENTRAL >>>'
go

```

```

/*
 * TABLE: PRODUCTO_TIENDA
 */

CREATE TABLE catalogos.PRODUCTO_TIENDA(
    id_producto numeric(10, 0) NOT NULL,
    id_categoria numeric(2,0) NOT NULL,
    id_oferta numeric(10, 0) NULL,
    detalles varchar(40) NOT NULL,
    precio money NOT NULL,
    CONSTRAINT PK_PRODUCTO_TIENDA PRIMARY KEY NONCLUSTERED
(id_producto)
)
go

```

```

IF OBJECT_ID('catalogos.PRODUCTO_TIENDA') IS NOT NULL
    PRINT '<<< CREATED TABLE PRODUCTO_TIENDA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE PRODUCTO_TIENDA >>>'
go

```

```

/*
 * TABLE: REGISTRO_BRAZALETE
 */

CREATE TABLE general.REGISTRO_BRAZALETE(
    id_registro numeric(10, 0) IDENTITY(1,1),
    id_mascota numeric(10, 0) NOT NULL,

```

```

ritmoC      char(10)      NOT NULL,
[fecha/hora]  datetime     NOT NULL,
temp        varchar(10)   NOT NULL,
nivelOx     varchar(10)   NOT NULL,
tipo_Comida  varchar(10)   NOT NULL,
comida_Comio  varchar(10)   NOT NULL,
CONSTRAINT PK_REGISTRO_BRAZALETE PRIMARY KEY NONCLUSTERED
(id_registro)
)
go

```

```

IF OBJECT_ID('general.REGISTRO_BRAZALETE') IS NOT NULL
    PRINT '<<< CREATED TABLE REGISTRO_BRAZALETE >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE REGISTRO_BRAZALETE >>>'
go

```

```

/*
 * TABLE: TELEFONO_PERSONAL
 */

```

```

CREATE TABLE catalogos.TELEFONO_PERSONAL(
id_telefono  numeric(1, 0)  NOT NULL,
id_empleado  numeric(10, 0) NOT NULL,
telefono     numeric(10, 0) NOT NULL,
CONSTRAINT PK_TELEFONO_PERSONAL PRIMARY KEY NONCLUSTERED
(id_telefono)
)
go

```

```

IF OBJECT_ID('catalogos.TELEFONO_PERSONAL') IS NOT NULL
    PRINT '<<< CREATED TABLE TELEFONO_PERSONAL >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE TELEFONO_PERSONAL >>>'
go

```

```

/*
 * TABLE: USUARIO_COMUN
 */

```

```

CREATE TABLE general.USUARIO_COMUN(
id_usuario  numeric(10, 0)  IDENTITY(1,1),
nom        varchar(10)    NOT NULL,
pat        varchar(10)    NOT NULL,

```

```

mat      varchar(10)  NOT NULL,
domicilio  varchar(40)  NOT NULL,
CURP      varchar(18)  NOT NULL,
genero     char(1)    NOT NULL
          CONSTRAINT CK_CS7 CHECK (genero in('H','M') ),
telefono   numeric(10, 0) NOT NULL,
numTargeta  numeric(19, 0) NOT NULL,
vigencia    date      NOT NULL,
correo      varchar(50)  NOT NULL ,
contraseña  varchar(20)  NOT NULL,
CONSTRAINT PK_USUARIO_COMUN PRIMARY KEY NONCLUSTERED (id_usuario),
          CONSTRAINT UQ_crupU unique (curp),
          CONSTRAINT UQ_correoU unique (correo),
          CONSTRAINT UQ_tele unique (telefono),
          CONSTRAINT UQ_numT unique (numTargeta)
)
go

```

```

IF OBJECT_ID('general.USUARIO_COMUN') IS NOT NULL
PRINT '<<< CREATED TABLE USUARIO_COMUN >>>'
ELSE
PRINT '<<< FAILED CREATING TABLE USUARIO_COMUN >>>'
go

/*
 * TABLE: VENDEDOR
 */

```

```

CREATE TABLE empleados.VENDEDOR(
id_empleado  numeric(10, 0) NOT NULL,
sueldo       money        NOT NULL,
sueldo_total  money        NOT NULL,
comision     money        NOT NULL,
CONSTRAINT PK_VENDEDOR PRIMARY KEY NONCLUSTERED (id_empleado)
)
go

```

```

IF OBJECT_ID('empleados.VENDEDOR') IS NOT NULL
PRINT '<<< CREATED TABLE VENDEDOR >>>'
ELSE
PRINT '<<< FAILED CREATING TABLE VENDEDOR >>>'
go

/*
 * TABLE: VENTA_FISICA
*/

```

```
*/
```

```
CREATE TABLE general.VENTA_FISICA(
    id_ventaF numeric(10, 0) IDENTITY(1,1) NOT NULL,
    id_empleado numeric(10, 0) NOT NULL,
    total money NOT NULL,
    CONSTRAINT PK_VENTA_FISICA PRIMARY KEY NONCLUSTERED (id_ventaF)
)
go
```

```
IF OBJECT_ID('general.VENTA_FISICA') IS NOT NULL
    PRINT '<<< CREATED TABLE VENTA_FISICA >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE VENTA_FISICA >>>'
go
```

```
/*
 * TABLE: VETERINARIO
 */
```

```
CREATE TABLE empleados.VETERINARIO(
    id_empleado numeric(10, 0) NOT NULL,
    CEDULA numeric(10, 0) NOT NULL,
    especialidad varchar(40) NOT NULL,
    CONSTRAINT PK_VETERINARIO PRIMARY KEY NONCLUSTERED (id_empleado),
    CONSTRAINT UQ_cedula unique (CEDULA),
)
go
```

```
IF OBJECT_ID('empleados.VETERINARIO') IS NOT NULL
    PRINT '<<< CREATED TABLE VETERINARIO >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE VETERINARIO >>>'
go
```

```
/*
 * TABLE: ADMINISTRATIVO
 */
```

```
ALTER TABLE empleados.ADMINISTRATIVO ADD CONSTRAINT FK_ESTADO_ADMIN
    FOREIGN KEY (id_estado)
        REFERENCES catalogos.ESTADO(id_estado)
go
```

```
ALTER TABLE empleados.ADMINISTRATIVO ADD CONSTRAINT FK_EMPLEADO_ADMIN
    FOREIGN KEY (id_empleado)
    REFERENCES empleados.EMPLEADO(id_empleado)
go
```

```
/*
 * TABLE: CAR_PROD
 */
```

```
ALTER TABLE general.CAR_PROD ADD CONSTRAINT FK_CARRITO_CAR_PROD
    FOREIGN KEY (id_carrito)
    REFERENCES general.CARRITO(id_carrito)
go
```

```
ALTER TABLE general.CAR_PROD ADD CONSTRAINT
FK_PRODUCTO_CENTRAL_CAR_PROD
    FOREIGN KEY (id_producto)
    REFERENCES catalogos.PRODUCTO_CENTRAL(id_producto)
go
```

```
/*
 * TABLE: CARRITO
 */
```

```
ALTER TABLE general.CARRITO ADD CONSTRAINT FK_USUARIO_COMUN_CARRITO
    FOREIGN KEY (id_usuario)
    REFERENCES general.USUARIO_COMUN(id_usuario)
go
```

```
/*
 * TABLE: CENTRO
 */
```

```
ALTER TABLE catalogos.CENTRO ADD CONSTRAINT FK_ESTADO_CEN
    FOREIGN KEY (id_estado)
    REFERENCES catalogos.ESTADO(id_estado)
go
```

```
/*
 * TABLE: COMPRA_CARRITO
 */
```

```
ALTER TABLE general.COMPRA_CARRITO ADD CONSTRAINT
FK_USUARIO_COMUN_COMPRA_CARRITO
```

```

FOREIGN KEY (id_usuario)
REFERENCES general.USUARIO_COMUN(id_usuario)
go

ALTER TABLE general.COMPRA_CARRITO ADD CONSTRAINT
FK_CAR_PROD_COMPRA_CARRITO
FOREIGN KEY (id_producto, id_carrito)
REFERENCES general.CAR_PROD(id_producto, id_carrito)
go

/*
 * TABLE: CONSULTA
*/
ALTER TABLE general.CONSULTA ADD CONSTRAINT FK_VETERINARIO_CONSULTA
FOREIGN KEY (id_empleado)
REFERENCES empleados.VETERINARIO(id_empleado)
go

ALTER TABLE general.CONSULTA ADD CONSTRAINT FK_MASCOTA_CONSULTA
FOREIGN KEY (id_mascota)
REFERENCES general.MASCOTA(id_mascota)
go

/*
 * TABLE: RECETA
*/
ALTER TABLE general.RECETA ADD CONSTRAINT FK_RECETA_MEDICAMENTO
FOREIGN KEY (id_medicamento)
REFERENCES catalogos.Medicamento(id_medicamento)
go

ALTER TABLE general.RECETA ADD CONSTRAINT FK_RECETA_CONSULTA
FOREIGN KEY (id_consulta)
REFERENCES general.consulta(id_consulta)
go

/*
 * TABLE: DETALLE CUENTA
*/
ALTER TABLE general.DETALLE CUENTA ADD CONSTRAINT
FK_VENTA_FISICA_DETALLE CUENTA
FOREIGN KEY (id_ventaF)
REFERENCES general.VENTA_FISICA(id_ventaF)

```

```
go
```

```
ALTER TABLE general.DETALLE CUENTA ADD CONSTRAINT  
FK_PRODUCTO_TIENDA_DETALLE CUENTA  
FOREIGN KEY (id_producto)  
REFERENCES catalogos.PRODUCTO_TIENDA(id_producto)
```

```
go
```

```
/*  
 * TABLE: EMPLEADO  
 */
```

```
ALTER TABLE empleados.EMPLEADO ADD CONSTRAINT FK_CENTRO_EMPLEADO  
FOREIGN KEY (id_centro)  
REFERENCES catalogos.CENTRO(id_centro)
```

```
go
```

```
/*  
 * TABLE: GUARDERIA  
 */
```

```
ALTER TABLE general.GUARDERIA ADD CONSTRAINT FK_CENTRO_GUARDERIA  
FOREIGN KEY (id_centro)  
REFERENCES catalogos.CENTRO(id_centro)
```

```
go
```

```
/*  
 * TABLE: MASCOTA  
 */
```

```
ALTER TABLE general.MASCOTA ADD CONSTRAINT  
FK_PERSONAL_AYUDA_MASCOTA  
FOREIGN KEY (id_empleado)  
REFERENCES empleados.PERSONAL_AYUDA(id_empleado)
```

```
go
```

```
ALTER TABLE general.MASCOTA ADD CONSTRAINT FK_GUARDERIA_MASCOTA  
FOREIGN KEY (id_guarderia)  
REFERENCES general.GUARDERIA(id_guarderia)
```

```
go
```

```
ALTER TABLE general.MASCOTA ADD CONSTRAINT FK_USUARIO_COMUN_MASCOTA  
FOREIGN KEY (id_usuario)  
REFERENCES general.USUARIO_COMUN(id_usuario)
```

```
go
```

```
/*
 * TABLE: PERSONAL_AYUDA
 */
```

```
ALTER TABLE empleados.PERSONAL_AYUDA ADD CONSTRAINT
FK_EMPLEADO_PERSONAL_AYUDA
FOREIGN KEY (id_empleado)
REFERENCES empleados.EMPLEADO(id_empleado)
go
```

```
/*
 * TABLE: PRODUCTO_CENTRAL
 */
```

```
ALTER TABLE catalogos.PRODUCTO_CENTRAL ADD CONSTRAINT
FK_OFERTA_PRODUCTO_CENTRAL
FOREIGN KEY (id_oferta)
REFERENCES catalogos.OFERTA(id_oferta)
go
```

```
ALTER TABLE catalogos.PRODUCTO_CENTRAL ADD CONSTRAINT
FK_CATEGORIA_PRODUCTO_CENTRAL
FOREIGN KEY (id_categoria)
REFERENCES catalogos.categoria_producto(id_categoria)
go
```

```
/*
 * TABLE: PRODUCTO_TIENDA
 */
```

```
ALTER TABLE catalogos.PRODUCTO_TIENDA ADD CONSTRAINT
FK_OFERTA_PRODUCTO_TIENDA
FOREIGN KEY (id_oferta)
REFERENCES catalogos.OFERTA(id_oferta)
go
```

```
ALTER TABLE catalogos.PRODUCTO_TIENDA ADD CONSTRAINT
FK_CATEGORIA_PRODUCTO_TIENDA
FOREIGN KEY (id_categoria)
REFERENCES catalogos.categoria_producto(id_categoria)
go
```

```
/*
 * TABLE: REGISTRO_BRAZALETE
 */
```

```
*/
```

```
ALTER TABLE general.REGISTRO_BRAZALETE ADD CONSTRAINT  
FK_MASCOTA_REGISTRO_BRAZALETE  
FOREIGN KEY (id_mascota)  
REFERENCES general.MASCOTA(id_mascota)
```

```
go
```

```
/*  
* TABLE: TELEFONO_PERSONAL  
*/
```

```
ALTER TABLE catalogos.TELEFONO_PERSONAL ADD CONSTRAINT  
FK_PERSONAL_AYUDA_TELEFONO_PERSONAL  
FOREIGN KEY (id_empleado)  
REFERENCES empleados.PERSONAL_AYUDA(id_empleado)
```

```
go
```

```
/*  
* TABLE: VENDEDOR  
*/
```

```
ALTER TABLE empleados.VENDEDOR ADD CONSTRAINT FK_EMPLEADO_VENDEDOR  
FOREIGN KEY (id_empleado)  
REFERENCES empleados.EMPLEADO(id_empleado)
```

```
go
```

```
/*  
* TABLE: VENTA_FISICA  
*/
```

```
ALTER TABLE general.VENTA_FISICA ADD CONSTRAINT  
FK_VENDEDOR_VENTA_FISICA  
FOREIGN KEY (id_empleado)  
REFERENCES empleados.VENDEDOR(id_empleado)
```

```
go
```

```
/*  
* TABLE: VETERINARIO  
*/
```

```
ALTER TABLE empleados.VETERINARIO ADD CONSTRAINT  
FK_EMPLEADO_VETERINARIO  
FOREIGN KEY (id_empleado)  
REFERENCES empleados.EMPLEADO(id_empleado)
```

Cargainicial código

```
/*
 * EQUIPO 7 GUARDERIA DE MASCOTAS
 * CORIA MARTINEZ GUSTAVO
 * ORONZOR MONTES MANASES LEONEL
 * PANG ARAIZAGA IAN
 */
```

--Elaborar el script para la carga de información, el cual debe
--llamarse cargainicial.sql. este contendrá los inserts necesarios
--para cargar al menos 20 ventas para usuarios registrados y 10
--ventas en establecimiento, 10 usuarios registrados, 10 gestores, 5
--vendedores, 4 categorías con 5 productos cada uno, 5 ofertas y
--5 cestas pendientes de compra.

--Estados--1

```
use Equipo7_PETCO
```

```
INSERT INTO catalogos.ESTADO (nombre)
values ('Aguascalientes'),
```

```
        ('Baja California'),
        ('Baja California Sur'),
        ('Campeche'),
        ('Coahuila'),
        ('Colima'),
        ('Chiapas'),
        ('Chihuahua'),
        ('Ciudad de Mexico'),
        ('Durango'),
        ('Estado de Mexico'),
        ('Guanajuato'),
        ('Guerrero'),
        ('Hidalgo'),
        ('Jalisco'),
        ('Michoacan'),
        ('Morelos'),
        ('Nayarit'),
        ('Nuevo Leon'),
        ('Oaxaca'),
        ('Puebla'),
        ('Queretaro'),
        ('Quintana Roo'),
        ('San Luis Potosi'),
        ('Sinaloa'),
        ('Sonora'),
```

('Tabasco'),
 ('Tamaulipas'),
 ('Tlaxcala'),
 ('Veracruz'),
 ('Yucatan'),
 ('Zacatecas');

--Usuarios comun---2

INSERT INTO general.USUARIO_COMUN(nom, pat, mat, domicilio, curp, genero, telefono, numTargeta, vigencia , correo, contraseña)

VALUES

('SONIA', 'ESTRELLA', 'RICOS', 'Yobain 09', 'SOER920113MDFSTR01', 'M', 5562719823, '223516283947362739', '2023-08-27', 'soñaer@yahoo.com', 'agts1763dh26'),
 ('JUAN', 'PÉREZ', 'GARCÍA', 'Calle del Sol 123', 'PERJ900625HDFPNC02', 'H', 5551234567, '987654321098765432', '2023-09-10', 'juan.perez@gmail.com', 'abc123xyz'),
 ('MARÍA', 'GUTIÉRREZ', 'SÁNCHEZ', 'Avenida Principal 456', 'GUTM981207MDFSNH03', 'M', 5549873210, '135792468024681357', '2023-07-15', 'maria.gutierrez@gmail.com', 'ingeniero123'),
 ('CARLOS', 'MARTÍNEZ', 'RODRÍGUEZ', 'Av. Reforma 789', 'MARC890521MDFRDC04', 'H', 5587654321, '246813578024681379', '2023-11-05', 'carlos.martinez@gmail.com', 'clave123'),
 ('ANA', 'SÁNCHEZ', 'MARTÍNEZ', 'Calle 10 de Mayo 456', 'SAMA930705MDFSNH08', 'M', 5567092837, '192837465098273645', '2023-10-15', 'ana.sanchez@gmail.com', 'abcxyz123'),
 ('PEDRO', 'GÓMEZ', 'HERNÁNDEZ', 'Av. Juárez 234', 'GOHP870423HDFRND02', 'H', 5572983476, '203948576027384659', '2023-09-01', 'pedro.gomez@gmail.com', 'password123'),
 ('LAURA', 'FLORES', 'CRUZ', 'Calle del Bosque 56', 'FOCL880210MDFLRA05', 'M', 5547819203, '293847560928374651', '2023-07-30', 'laura.flores@gmail.com', 'secreto456'),
 ('RICARDO', 'RAMÍREZ', 'LÓPEZ', 'Paseo de la Reforma 789', 'RALR950713HDFRC03', 'H', 5550982736, '384756092837465109', '2023-11-20', 'ricardo.ramirez@gmail.com', 'qwerty789'),
 ('PAULA', 'RODRÍGUEZ', 'GÓMEZ', 'Av. Insurgentes 456', 'ROGP900601HDFDLA09', 'M', 5562738492, '273645098273645091', '2023-09-05', 'paula.rodriguez@gmail.com', 'asdfg123'),
 ('MIGUEL', 'SANTOS', 'MARTÍNEZ', 'Calle del Carmen 234', 'SAMM920317HDFNTG07', 'H', 5549837261, '495836274059283746', '2023-07-20', 'miguel.santos@gmail.com', 'zxcvb789'),
 ('ISABEL', 'GARCÍA', 'HERNÁNDEZ', 'Calle de la Luna 78', 'GAHI910215HDFBLS06', 'M', 5572968473, '495836273645092836', '2023-10-10', 'isabel.garcia@gmail.com', 'qazwsx456'),
 ('LUIS', 'MARTÍNEZ', 'RODRÍGUEZ', 'Av. Hidalgo 123', 'MARR850726HDFDLU05', 'H', 5558273645, '203948572736459283', '2023-09-15', 'luis.martinez@gmail.com', 'poiuyt123'),
 ('FERNANDA', 'HERNÁNDEZ', 'SÁNCHEZ', 'Calle de los Pinos 67', 'HESF880415MDFNRRN08', 'M', 5547982635, '293847560928364507', '2023-07-25', 'fernanda.hernandez@gmail.com', 'Ikjhgf456'),

('JOSÉ', 'GÓMEZ', 'CRUZ', 'Av. Zaragoza 456', 'GOJC910505HDFZSJ04', 'H', 5582736459, '384759283746502938', '2023-11-25', 'jose.gomez@gmail.com', 'mnbvcxz789'),
 ('ANDREA', 'RODRÍGUEZ', 'GÓMEZ', 'Calle del Roble 89', 'ROGA950213HDFDNR01', 'M', 5569873425, '273645092836459283', '2023-09-20', 'andrea.rodriguez@gmail.com', '0987654321'),
 ('ANTONIO', 'SÁNCHEZ', 'MARTÍNEZ', 'Calle del Sol 345', 'SAMA890713HDFNTN09', 'H', 5554321769, '495836274650293847', '2023-08-05', 'antonio.sanchez@gmail.com', '13579abc'),
 ('SOFÍA', 'MARTÍNEZ', 'HERNÁNDEZ', 'Av. Cuauhtémoc 567', 'MAHS910720HDFRTF00', 'M', 5547892637, '203947562739485629', '2023-07-05', 'sofia.martinez@gmail.com', '24680xyz'),
 ('RAÚL', 'GÓMEZ', 'RODRÍGUEZ', 'Calle del Carmen 456', 'GORR920523HDFLNL06', 'H', 5572983647, '384756092936475628', '2023-10-25', 'raul.gomez@gmail.com', '13579qwe'),
 ('ANAHÍ', 'RODRÍGUEZ', 'GÓMEZ', 'Av. Reforma 234', 'ROGA940205HDFNHN05', 'M', 5569873426, '273645092836475629', '2023-09-30', 'anahi.rodriguez@gmail.com', '09876poi'),
 ('GERARDO', 'SÁNCHEZ', 'MARTÍNEZ', 'Calle del Bosque 78', 'SAGG880726HDFRCD02', 'H', 5554321768, '495836274650293647', '2023-08-10', 'gerardo.sanchez@gmail.com', '13579mnb'),
 ('DANIELA', 'MARTÍNEZ', 'HERNÁNDEZ', 'Av. Hidalgo 567', 'MAHD890315HDFNLR09', 'M', 5547892636, '203947562784485629', '2023-07-10', 'daniela.martinez@gmail.com', '24680lkj'),
 ('HUGO', 'GÓMEZ', 'RODRÍGUEZ', 'Calle del Roble 12', 'GORH930505HDFRCD07', 'H', 5572983648, '384756092936475829', '2023-11-01', 'hugo.gomez@gmail.com', '13579zxc');

--centros--3

```
INSERT INTO catalogos.centro (id_estado, oficinaRegional)
VALUES
(1, 1),
(2, 0),
(3, 0),
(4, 0),
(5, 0),
(6, 0),
(7, 1),
(8, 1),
(9, 0),
(10, 1),
(11, 0),
(12, 0),
(13, 0),
(14, 0),
(15, 0),
(16, 0),
(17, 0),
```

(18, 0),
 (19, 0),
 (20, 0),
 (21, 0),
 (22, 0),
 (23, 0),
 (24, 1),
 (25, 0),
 (26, 0),
 (27, 0),
 (28, 0),
 (29, 0),
 (30, 0),
 (31, 0),
 (32, 0);

--Empleados--4

```
INSERT INTO empleados.Empleado (id_centro, nombre, paterno, materno, curp, domicilio)
VALUES
(1, 'Juan', 'Pérez', 'López', 'PELJ900101HDFXNN02', 'Calle 123, Ciudad de México'),
(1, 'María', 'Rodríguez', 'García', 'ROGM980202MTZXXX03', 'Avenida Principal 456, Monterrey'),
(1, 'Alejandro', 'Gómez', 'Hernández', 'GOHA880520GTOXXX04', 'Calle Central 789, Guanajuato'),
(1, 'Luis', 'González', 'Martínez', 'GOML921205HDFXYZ05', 'Avenida 456, Puebla'),
(7, 'Ana', 'López', 'Sánchez', 'LOSA880410MTZXXX06', 'Calle 789, Guadalajara'),
(7, 'Pedro', 'Martínez', 'Romero', 'MARP900608GTOXYZ07', 'Calle 456, León'),
(7, 'Mónica', 'Hernández', 'García', 'HEGM890201HDFXYZ08', 'Avenida Principal 123, Querétaro'),
(7, 'Carlos', 'Torres', 'Pérez', 'TOPC910405HDFXYZ09', 'Calle 789, Toluca'),
(8, 'Sofía', 'Vargas', 'Jiménez', 'VAJS950101HDFXYZ10', 'Avenida 123, Cancún'),
(8, 'Fernando', 'Ortega', 'Rojas', 'ORRF930701HDFXYZ11', 'Calle Principal 456, Mérida'),
(8, 'Laura', 'Díaz', 'Morales', 'DIAM850202HDFXYZ12', 'Avenida Central 789, Monterrey'),
(8, 'Roberto', 'Silva', 'Gómez', 'SIGR910405MTZXYZ13', 'Calle 123, Guadalajara'),
(10, 'Elena', 'Mendoza', 'Navarro', 'MENE960101MTZXYZ14', 'Avenida Principal 456, Puebla'),
(10, 'Javier', 'Rojas', 'Vargas', 'ROVJ900202GTOXYZ15', 'Calle Central 789, Ciudad de México'),
(10, 'Isabel', 'Gutiérrez', 'Flores', 'GUFI890301GTOXYZ16', 'Avenida 123, Guanajuato'),
(10, 'Miguel', 'Morales', 'López', 'MOLM920101MTZXYZ17', 'Calle 456, Monterrey'),
(24, 'Patricia', 'Romero', 'Sánchez', 'ROSP930202HDFXYZ18', 'Avenida Principal 789, León'),
(24, 'Ricardo', 'Flores', 'Gómez', 'FLOG880303MTZXYZ19', 'Calle Central 123, Tijuana'),
(24, 'Daniela', 'Navarro', 'Ortega', 'NADO910404HDFXYZ20', 'Avenida 456, Mérida'),
(24, 'Héctor', 'Jiménez', 'Silva', 'JISH890505HDFXYZ21', 'Calle 789, Monterrey');
```

--Gestores--5

```
INSERT INTO empleados.administrativo (id_empleado, id_estado)
VALUES
(1, 1),
(5, 7),
(9, 8),
(13,10),
(17, 24);
```

-- Ofertas-- 6

```
INSERT INTO catalogos.oferta (tipo, porcentaje, fecha_ini, fecha_fin)
VALUES
('N', 0.75, '2023-06-07', '2023-06-18'),
('L', 0.85, '2023-07-05', '2023-07-12'),
('N', 0.60, '2023-06-15', '2023-06-25'),
('L', 0.95, '2023-07-01', '2023-07-07'),
('N', 0.50, '2023-06-10', '2023-06-15');
```

--categoria_producto--7

```
INSERT INTO catalogos.CATEGORIA_PRODUCTO (descripcion)
VALUES
('Juguetes'),
('Premios'),
('Alimentos'),
('Utensilios varios');
```

-- producto_central y producto_tienda --8

```
INSERT INTO catalogos.producto_central (id_producto, id_categoria, id_oferta, detalles, precio)
VALUES
-- Juguetes
(1234567890, 1, 3, 'Peluche chirriante "SqueakyBuddy"', 25.99),
(2345678901, 1, 2, 'Juguete interactivo "PawPuzzle"', 12.99),
(3456789012, 1, NULL, 'Maquillaje base de larga duración', 18.99),
(4567890123, 1, 1, 'Pelota masticable "DuraBite"', 29.99),
(5678901234, 1, 4, 'Juguete de cuerda "Tug-o-Fun"', 39.99),
```

-- Premios

```
(6789912345, 2, NULL, 'Galletas de pollo "Crunchy Bites"', 19.99),
(7891123456, 2, 5, 'Snacks de salmón "Fishy Delights"', 14.99),
(8901734567, 2, 1, 'Premios de cordero "Tasty Rewards"', 9.99),
(9412345678, 2, 3, 'Chucherías de manzana "Fruity Treats"', 22.99),
```

(1234765890, 2, 2, 'Barritas de buey ahumado "Savory Sticks"', 49.99),

-- Alimentos

(2345678900, 3, 4, 'Croquetas premium "NutriPrime"', 59.99),
 (3556789012, 3, NULL, 'Comida húmeda "Delicious Purrfection"', 39.99),
 (4567890124, 3, 1, 'Snacks naturales "Tasty Tidbits"', 14.99),
 (5677901234, 3, 2, 'Alimento seco "Tiny Bites"', 299.99),
 (6789012345, 3, 5, 'Comida especializada "SlimKitty"', 199.99),

-- Utensilios varios

(7890123456, 4, 3, 'Cepillo de pelo "GentleGroom"', 79.99),
 (8901234567, 4, NULL, 'Dispensador de bolsas "CleanPaws"', 899.99),
 (9012345678, 4, 4, 'Bebedero automático "HydroFlow"', 149.99),
 (1234567880, 4, 1, 'Correa extensible "FlexiLeash"', 29.99),
 (3345678901, 4, 2, 'Transportador "EasyTravel"', 69.99);

```
INSERT INTO catalogos.producto_tienda (id_producto, id_categoria, id_oferta, detalles, precio)
```

```
VALUES
```

-- Juguetes

(1234567890, 1, 3, 'Peluche chirriante "SqueakyBuddy"', 25.99),
 (2345678901, 1, 2, 'Juguete interactivo "PawPuzzle"', 12.99),
 (3456789012, 1, NULL, 'Maquillaje base de larga duración', 18.99),
 (4567890123, 1, 1, 'Pelota masticable "DuraBite"', 29.99),
 (5678901234, 1, 4, 'Juguete de cuerda "Tug-o-Fun"', 39.99),

-- Premios

(6789912345, 2, NULL, 'Galletas de pollo "Crunchy Bites"', 19.99),
 (7891123456, 2, 5, 'Snacks de salmón "Fishy Delights"', 14.99),
 (8901734567, 2, 1, 'Premios de cordero "Tasty Rewards"', 9.99),
 (9412345678, 2, 3, 'Chucherías de manzana "Fruity Treats"', 22.99),
 (1234765890, 2, 2, 'Barritas de buey ahumado "Savory Sticks"', 49.99),

-- Alimentos

(2345678900, 3, 4, 'Croquetas premium "NutriPrime"', 59.99),
 (3556789012, 3, NULL, 'Comida húmeda "Delicious Purrfection"', 39.99),
 (4567890124, 3, 1, 'Snacks naturales "Tasty Tidbits"', 14.99),
 (5677901234, 3, 2, 'Alimento seco "Tiny Bites"', 299.99),
 (6789012345, 3, 5, 'Comida especializada "SlimKitty"', 199.99),

-- Utensilios varios

(7890123456, 4, 3, 'Cepillo de pelo "GentleGroom"', 79.99),
 (8901234567, 4, NULL, 'Dispensador de bolsas "CleanPaws"', 899.99),
 (9012345678, 4, 4, 'Bebedero automático "HydroFlow"', 149.99),
 (1234567880, 4, 1, 'Correa extensible "FlexiLeash"', 29.99),
 (3345678901, 4, 2, 'Transportador "EasyTravel"', 69.99);

-- Vendedor --9

```
INSERT INTO empleados.vendedor (id_empleado, sueldo, sueldo_total, comision)
VALUES
(2, 5000, 5000, 0),
(6, 5000, 5000, 0),
(10, 5000, 5000, 0),
(13, 5000, 5000, 0),
(18, 5000, 5000, 0);
```

--Personal Ayuda--10

```
INSERT INTO
empleados.PERSONAL_AYUDA(id_empleado,usuario,contraseña,edad,mascota)
values (3,'blue','gastde1234',20,'P'),
(7,'red','xdcdece68',25,'G'),
(11,'black','hbcedh5',24,'P'),
(14,'lock','jxsbt4',27,'G'),
(19,'rino','kcxseb2',30,'P');
```

--Telefonos personal ayuda-- 11

```
insert into catalogos.Telefono_Personal(id_telefono,id_empleado,telefono)
values (1, 3, 5561889234),
(2, 7, 6789054321),
(3, 11, 1234567890),
(4, 14, 9876543210),
(5, 19, 5432167890);
```

-- Veterinario-- 12

```
INSERT INTO empleados.VETERINARIO(id_empleado,CEDULA,especialidad)
values (4,6534782349,'Cardiologia'),
(8,8126453789,'Neurologia'),
(12,1083645382,'Radiologia'),
(16,5271865235,'Dermatologia'),
(20,1452783940,'Traumatologia');
```

--Guarderia--13

```
INSERT INTO general.GUARDERIA (id_centro,fecha_ini,fecha_fin)
values
(1,'2023-06-10', '2023-06-15'),
(1,'2023-06-11', '2023-06-16'),
```

```
(1,'2023-06-12', '2023-06-17'),
(1,'2023-06-13', '2023-06-18'),
(7,'2023-06-14', '2023-06-19'),
(7,'2023-06-15', '2023-06-20'),
(7,'2023-06-16', '2023-06-21'),
(7,'2023-06-17', '2023-06-22'),
(8,'2023-06-18', '2023-06-23'),
(8,'2023-06-19', '2023-06-24'),
(8,'2023-06-20', '2023-06-25'),
(8,'2023-06-21', '2023-06-26'),
(10,'2023-06-22', '2023-06-27'),
(10,'2023-06-23', '2023-06-28'),
(10,'2023-06-24', '2023-06-29'),
(10,'2023-06-25', '2023-06-30'),
(24,'2023-06-26', '2023-07-01'),
(24,'2023-06-27', '2023-07-02'),
(24,'2023-06-28', '2023-07-03'),
(24,'2023-06-29', '2023-07-04');
```

-- Mascota-- 14

```
INSERT INTO
general.MASCOTA(id_empleado,id_guarderia,id_usuario,nombre,especie,sexo,raza,razgos,
edad) values
(3,1,3,'LUKE', 'perro', 'macho', 'pastor', 'negro con café', 1),
(11,2,8,'MAX', 'perro', 'macho', 'labrador', 'dorado', 2),
(19,3,14,'BELLA', 'perro', 'hembra', 'bulldog ', 'blanco y negro', 3),
(7,4,9,'MININA', 'gato', 'hembra', 'siamés', 'negro', 2),
(14,5,16,'LUCY', 'gato', 'hembra', 'atigrado', 'doméstico', 2);
```

--CONSULTA --15

```
INSERT INTO general.CONSULTA (id_empleado, id_mascota, fecha, hora, diagnostico,
detalles, costo_total)
VALUES
(4, 1, '2023-06-09', '18:10:00', 'Conjuntivitis', 'reposo', 190),
(8, 2, '2023-06-16', '14:30:00', 'Dermatitis alérgica', 'revision', 245),
(12, 3, '2023-06-20', '15:36:00', 'Otitis externa', 'reposo', 270),
(16, 4, '2023-06-24', '16:30:00', 'Gastroenteritis', 'asilado', 200),
(20, 5, '2023-06-28', '17:39:00', 'Alergias cutáneas', 'reposo', 280),
(4, 1, '2023-07-02', '09:15:00', 'Conjuntivitis', 'reposo', 180),
(8, 2, '2023-07-05', '11:45:00', 'Dermatitis alérgica', 'revision', 210),
(12, 3, '2023-07-08', '13:20:00', 'Otitis externa', 'reposo', 290),
(16, 4, '2023-07-11', '15:50:00', 'Gastroenteritis', 'asilado', 250),
(20, 5, '2023-07-14', '18:30:00', 'Alergias cutáneas', 'reposo', 170),
(4, 1, '2023-07-17', '10:00:00', 'Conjuntivitis', 'reposo', 220),
(8, 2, '2023-07-20', '12:30:00', 'Dermatitis alérgica', 'revision', 260),
(12, 3, '2023-07-23', '15:00:00', 'Otitis externa', 'reposo', 180),
```

(16, 4, '2023-07-26', '17:45:00', 'Gastroenteritis', 'asilado', 290),
 (20, 5, '2023-07-29', '19:20:00', 'Alergias cutáneas', 'reposo', 200);

-- REGISTRO_BRAZALETE -- 16

```
INSERT INTO
general.REGISTRO_BRAZALETE(id_mascota,ritmoC,[fecha/hora],temp,nivelOx,tipo_Comida,comida_Comio) values
(1,'60-100','2023-06-28 17:39:00','37', '80','croquetas', 'No'),
(2,'60-100','2023-06-29 12:39:00','37', '70','croquetas', 'No'),
(3,'60-100','2023-06-27 13:39:00','36', '80','croquetas', 'No'),
(4,'60-100','2023-06-26 12:39:00','37', '90','croquetas', 'No'),
(5,'60-100','2023-06-25 13:39:00','36', '80','croquetas', 'No');
```

-- MEDICAMENTO -- 17

```
INSERT INTO catalogos.MEDICAMENTO(nombre_medicamento,costo, cantidad) values
('Ibuprofeno',700, 500),
('Aspirina', 100, 500),
('Atrabica', 150, 500),
('Loxcell', 180, 500),
('Ibuprofeno',700, 500),
('Anestesia',500, 500);
```

--Venta fisica--

```
INSERT INTO general.venta_fisica (id_empleado, total)
VALUES
(2, 25.99),
(6, 12.99),
(10, 18.99),
(13, 29.99),
(18, 39.99),
(2, 19.99),
(10, 14.99),
(6, 9.99),
(13, 22.99),
(18, 49.99);
```

-- Detalle cuenta --

```
INSERT INTO general.detalle_cuenta (id_ventaF, id_producto, montoparcial, cantidad)
VALUES
(1, 1234567890, 25.99, 3),
```

```
(2, 2345678901, 12.99, 2),
(3, 3456789012, 18.99, 5),
(4, 4567890123, 29.99, 1),
(5, 5678901234, 39.99, 4),
(6, 6789912345, 19.99, 3),
(7, 7891123456, 14.99, 2),
(8, 8901734567, 9.99, 1),
(9, 9412345678, 22.99, 5),
(10, 1234765890, 49.99, 3);
```

--Carrito--

```
INSERT INTO general.Carrito (id_carrito, id_usuario, cantidad)
VALUES
(123456, 1, 3),
(234567, 10, 2),
(345678, 5, 4),
(456789, 15, 1),
(567890, 7, 5),
(678901, 20, 3),
(789012, 2, 3),
(890123, 12, 4),
(901234, 8, 2),
(102345, 18, 3),
(113456, 3, 5),
(124567, 19, 1),
(135678, 6, 4),
(146789, 13, 2),
(157890, 9, 3),
(168901, 14, 5),
(179012, 4, 1),
(190123, 17, 4),
(201234, 11, 2),
(212345, 16, 3),
(223456, 7, 1),
(234467, 20, 4),
(245678, 5, 2),
(256789, 1, 3),
(267890, 10, 5),
(278901, 9, 1);
```

--Car-prod--

```
INSERT INTO general.car_prod (id_carrito, id_producto, montoparcial, cantidad)
VALUES
(123456, 1234567890, 25.99, 3),
```

(234567, 2345678901, 12.99, 2),
 (345678, 3456789012, 18.99, 4),
 (567890, 4567890123, 29.99, 1),
 (789012, 5678901234, 39.99, 5),
 (890123, 6789912345, 19.99, 3),
 (901234, 7891123456, 14.99, 3),
 (102345, 8901734567, 9.99, 4),
 (113456, 9412345678, 22.99, 2),
 (124567, 1234765890, 49.99, 3),
 (135678, 2345678900, 25.99, 5),
 (146789, 3556789012, 12.99, 1),
 (168901, 4567890124, 18.99, 4),
 (179012, 5677901234, 29.99, 2),
 (201234, 6789012345, 39.99, 3),
 (212345, 7890123456, 19.99, 5),
 (223456, 8901234567, 14.99, 1),
 (234567, 9012345678, 9.99, 4),
 (245678, 1234567880, 22.99, 2),
 (256789, 3345678901, 49.99, 3);

---compra_carrito

INSERT INTO general.compra_carrito (id_carrito, id_usuario, id_producto, detalles, montoparcial)

VALUES

(123456, 1, 1234567890, 'Compra del producto', 25.99),
 (234567, 10, 2345678901, 'Compra del producto', 12.99),
 (345678, 5, 3456789012, 'Compra del producto', 18.99),
 (567890, 7, 4567890123, 'Compra del producto', 29.99),
 (789012, 2, 5678901234, 'Compra del producto', 39.99),
 (890123, 12, 6789912345, 'Compra del producto', 19.99),
 (901234, 8, 7891123456, 'Compra del producto', 14.99),
 (102345, 18, 8901734567, 'Compra del producto', 9.99),
 (113456, 3, 9412345678, 'Compra del producto', 22.99),
 (124567, 19, 1234765890, 'Compra del producto', 49.99),
 (135678, 6, 2345678900, 'Compra del producto', 25.99),
 (146789, 13, 3556789012, 'Compra del producto', 12.99),
 (168901, 14, 4567890124, 'Compra del producto', 18.99),
 (179012, 4, 5677901234, 'Compra del producto', 29.99),
 (201234, 11, 6789012345, 'Compra del producto', 39.99),
 (212345, 16, 7890123456, 'Compra del producto', 19.99),
 (223456, 7, 8901234567, 'Compra del producto', 14.99),
 (234567, 20, 9012345678, 'Compra del producto', 9.99),--
 (245678, 5, 1234567880, 'Compra del producto', 22.99),
 (256789, 1, 3345678901, 'Compra del producto', 49.99);

--RECETA

```
INSERT INTO general.RECETA (id_consulta, id_medicamento, cantidad) VALUES
(1, 1, 2),
(2, 3, 1),
(3, 2, 3),
(4, 4, 1),
(5, 5, 2),
(6, 1, 3),
(7, 2, 2),
(8, 5, 1),
(9, 3, 3),
(10, 1, 1),
(11, 4, 2),
(12, 5, 3),
(13, 2, 1),
(14, 1, 2),
(15, 3, 3);
```

--Select de todas las tablas

```
select * from catalogos.ESTADO--1
select * from general.USUARIO_COMUN--2
select * from catalogos.CENTRO--3
select * from empleados.EMPLEADO--4
select * from empleados.ADMINISTRATIVO--5
select * from catalogos.oferta--6
select * from catalogos.CATEGORIA_PRODUCTO--7
select * from catalogos.producto_central--8
select * from catalogos.producto_tienda--9
select * from empleados.vendedor--10
select * from empleados.PERSONAL_AYUDA--11
select * from catalogos.Telefono_Personal--12
select * from empleados.VETERINARIO--13
select * from general.GUARDERIA--14
select * from general.MASCOTA--15
select * from general.CONSULTA--16
select * from general.REGISTRO_BRAZALETE--17
select * from catalogos.MEDICAMENTO--18
select * from general.venta_fisica--19
SELECT * from general.detalle_cuenta--20
SELECT * from general.carrito--21
select * from general.car_prod--22
select * from general.compra_carrito--23
select * from general.receta --24
```

DML código

```
/*
* EQUIPO 7 GUARDERIA DE MASCOTAS
```

```
* CORIA MARTINEZ GUSTAVO
* ORONZOR MONTES MANASES LEONEL
* PANG ARAIZAGA IAN
*/
```

-----PROCEDIMIENTOS
ALMACENADOS-----

```
--Realice un procedimiento almacenado para registrar una mascota y su brazalete
Create or alter procedure general.registro_mascota_brazalete
@p_id_empleado numeric(10, 0),
@p_id_usuario numeric(10, 0),
@p_id_guarderia numeric(10, 0),
@p_nombre varchar(10),
@p_especie varchar(10),
@p_sexo varchar(10),
@p_raza varchar(10),
@p_razgos varchar(40),
@p_edad numeric(2, 0),
@p Ritmo char(10),
@p_temp varchar(10),
@p_nivelOx varchar(10),
@p_tipo_comida varchar(10),
@p_comida_comio varchar(10)

as
begin
    INSERT INTO general.MASCOTA (id_empleado, id_usuario, id_guarderia,
        nombre, especie, sexo, raza, razgos, edad)
    VALUES (@p_id_empleado, @p_id_usuario, @p_id_guarderia, @p_nombre,
        @p_especie, @p_sexo, @p_raza, @p_razgos, @p_edad);
    INSERT INTO general.REGISTRO_BRAZALETE (id_mascota, ritmoC, temp,
        nivelOx, tipo_Comida, comida_Comio, [fecha/hora])
    VALUES ((select max(id_mascota) from general.MASCOTA), @p Ritmo, @p_temp,
        @p_nivelOx,
        @p_tipo_comida, @p_comida_comio, GETDATE());
    SELECT 'Mascota y brazalete registrados correctamente' AS Mensaje;
end

begin tran
SELECT * from general.MASCOTA
SELECT * from general.REGISTRO_BRAZALETE
execute general.registro_mascota_brazalete 14, 16, 5, 'LUCY', 'gato',
'hembra', 'atigrado', 'doméstico', 2, '60-100', '36', '80', 'croquetas', 'No'
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from general.MASCOTA
SELECT * from general.REGISTRO_BRAZALETE
rollback tran
```

```
--Si se prueba es necesario quitar la insercion del indexado
DBCC CHECKIDENT ('general.mascota', RESEED, 5);
DBCC CHECKIDENT ('general.registro_brazalete', RESEED, 5);
DBCC CHECKIDENT ('general.mascota', NORESEED);
DBCC CHECKIDENT ('general.registro_brazalete', NORESEED);
```

```
--Realice un procedimiento almacenado para el actualizar de las lecturas en los
--brazaletes
```

```
Create or alter procedure general.nuevo_registro_brazalete
@p_id_registro numeric(10, 0),
@p_id_mascota numeric(10, 0),
@p Ritmo char(10),
@p_fecha_hora datetime,
@p_temp varchar(10),
@p_nivelOx varchar(10),
@p_tipo_comida varchar(10),
@p_comida_comio varchar(10)

as
begin
    if exists (Select * from general.REGISTRO_BRAZALETE
               where id_registro = @p_id_registro and id_mascota =
@p_id_mascota)
        begin
            update general.REGISTRO_BRAZALETE set
                id_mascota = @p_id_mascota,
                ritmoC = @p_Ritmo,
                temp = @p_temp,
                nivelOx = @p_nivelOx,
                tipo_comida = @p_tipo_comida,
                comida_comio = @p_comida_comio,
                [fecha/hora] = @p_fecha_hora
            where id_registro = @p_id_registro and id_mascota = @p_id_mascota
            SELECT 'Actualizacion realizada correctamente' AS Mensaje;
        end
    else
        begin
            SELECT 'Algo fallo, revisa los valores' AS Mensaje;
        end
end
```

```
--Prueba de funcionamiento
begin tran
SELECT * from general.REGISTRO_BRAZALETE
```

```

execute general.nuevo_registro_brazalete 5, 5, '60-100', '2024-06-28 17:39:00', '36', '80',
'croquetas', 'No'
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from general.REGISTRO_BRAZALETE
rollback tran
-----
```

--Realice un procedimiento almacenado para registrar una consulta

```

CREATE or alter PROCEDURE general.RegistrarConsulta
    @id_empleado numeric(10, 0),
    @fecha date,
    @hora time(7),
    @diagnostico varchar(40),
    @detalles varchar(40),
    @costo_total money,
    @id_mascota numeric(10, 0)
AS
BEGIN
    INSERT INTO general.CONSULTA (id_empleado, fecha, hora, diagnostico, detalles,
costo_total, id_mascota)
    VALUES (@id_empleado, @fecha, @hora, @diagnostico, @detalles, @costo_total,
@id_mascota);

    SELECT 'Consulta registrada correctamente' AS Mensaje;
END
```

--Prueba de funcionamiento

```

begin tran
SELECT * from general.CONSULTA
execute general.RegistrarConsulta 16, '2023-06-28', '17:39:00', 'mordedura', 'reposo', 0.00, 5
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from general.CONSULTA
rollback tran
```

--Si se prueba es necesario quitar la insercion del indexado

```
DBCC CHECKIDENT ('general.consulta', RESEED, 15);
```

--Realice un procedimiento almacenado para el registro y venta de medicamentos

```

CREATE or alter PROCEDURE catalogos.RegistrarMedicamento
    @id_medicamento numeric(10, 0),
    @id_consulta numeric(10, 0),
    @nombre_medicamento varchar(10),
    @costo money,
    @cantidad numeric(4, 0)
```

```

AS
BEGIN
    if exists (Select * from catalogos.MEDICAMENTO
                where id_medicamento = @id_medicamento
                and nombre_medicamento = @nombre_medicamento and
                costo = @costo)
        begin
            update catalogos.MEDICAMENTO set cantidad = @cantidad
            where id_medicamento = @id_medicamento and
            nombre_medicamento = @nombre_medicamento and costo = @costo

            SELECT 'Se cambio correctamente la cantidad de medicamento' AS
Mensaje;
        end
        else
        begin
            INSERT INTO catalogos.MEDICAMENTO(nombre_medicamento,
costo,cantidad)
                VALUES (@nombre_medicamento, @costo, @cantidad);
        end
    END
-----
--incluyendo la actualización del stock

--cambio de cantidad medicamento
begin tran
SELECT * from catalogos.MEDICAMENTO
execute catalogos.RegistrarMedicamento 1,1,'Ibuprofeno',700,2
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from catalogos.MEDICAMENTO
rollback tran
--insertar medicamento
begin tran
SELECT * from catalogos.MEDICAMENTO
execute catalogos.RegistrarMedicamento null,1,'Vendas',150,500
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from catalogos.MEDICAMENTO
rollback tran

--Si se prueba es necesario quitar la insercion del indexado
DBCC CHECKIDENT ('catalogos.MEDICAMENTO', RESEED, 6);

Select * from catalogos.MEDICAMENTO
-----
```

--Realice un procedimiento para dar de alta una venta

--, y otro para cancelar una venta en línea
--así como cancelar una venta física.

```

CREATE or alter PROCEDURE catalogos.AltaProducto --
@id_categoria numeric(2, 0),
    @descripcion varchar(40),
    @id_producto numeric(10, 0),
    @id_oferta numeric(10, 0),
    @detalles varchar(40),
    @precio money
AS
BEGIN
    if exists (select id_categoria from catalogos.CATEGORIA_PRODUCTO where
    id_categoria = @id_categoria)
        begin
            INSERT INTO
catalogos.PRODUCTO_CENTRAL(id_producto,id_categoria,id_oferta,detalles,precio)
                VALUES (@id_producto,@id_categoria,@id_oferta,@detalles,@precio);
            INSERT INTO
catalogos.PRODUCTO_TIENDA(id_producto,id_categoria,id_oferta,detalles,precio)
                VALUES (@id_producto,@id_categoria,@id_oferta,@detalles,@precio);
            SELECT 'Se inserto la venta en centra y tienda' AS Mensaje;
        end
    else
        begin
            INSERT INTO catalogos.CATEGORIA_PRODUCTO(descripcion)
                VALUES (@descripcion);
            INSERT INTO
catalogos.PRODUCTO_CENTRAL(id_producto,id_categoria,id_oferta,detalles,precio)
                VALUES (@id_producto,(select max(id_categoria) from
catalogos.CATEGORIA_PRODUCTO),@id_oferta,@detalles,@precio);
            INSERT INTO
catalogos.PRODUCTO_TIENDA(id_producto,id_categoria,id_oferta,detalles,precio)
                VALUES (@id_producto,(select max(id_categoria) from
catalogos.CATEGORIA_PRODUCTO),@id_oferta,@detalles,@precio);
            SELECT 'Se creo una nueva categoria y se inserto la venta en centra y
tienda' AS Mensaje;
        end
    END

```

----Producto con categoria existente

```

begin tran
SELECT * from catalogos.PRODUCTO_TIENDA
execute catalogos.AltaProducto 1,null, 12,1,'chicharrones fritos',15.00
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from catalogos.PRODUCTO_TIENDA
rollback tran
----Producto sin categoria

```

```

begin tran
SELECT * from catalogos.PRODUCTO_TIENDA
execute catalogos.AltaProducto null,'comida', 16,null,'chicharrones fritos',15.00
SELECT 'DESPUES DEL PROCEDIMIENTO'
select * from catalogos.CATEGORIA_PRODUCTO
SELECT * from catalogos.PRODUCTO_TIENDA
rollback tran

--, y otro para cancelar una venta en línea

CREATE or alter PROCEDURE general.CancelarVentaLinea
    @id_carrito decimal(10,0)
as
begin
    if exists(select id_carrito from general.CARRITO where id_carrito = @id_carrito)
        begin
            update general.COMPRA_CARRITO
            set montoParcial = montoParcial*0.2
            where id_carrito = @id_carrito
            SELECT 'Se cancelo correctamente el pedido '+cast(@id_carrito as varchar)
        end
    else
        begin
            SELECT 'No se encontro la venta en linea' AS Mensaje;
        end
end

select * from general.CARRITO
---Pedido existente
begin tran
select * from general.COMPRA_CARRITO
execute general.CancelarVentaLinea 123456
select * from general.COMPRA_CARRITO
rollback
--Pedido inexistente
begin tran
select * from general.COMPRA_CARRITO
execute general.CancelarVentaLinea 122256
select * from general.COMPRA_CARRITO
rollback

--así como cancelar una venta física.
CREATE or alter PROCEDURE dbo.CancelarVentaFisica
    @id_ventaF decimal(10,0)
as
begin

```

```

if exists(select id_ventaF from general.VENTA_FISICA where id_ventaF =
@id_ventaF)
begin
    delete from general.DETALLE CUENTA where id_ventaF = @id_ventaF
    delete from general.VENTA_FISICA where id_ventaF = @id_ventaF
    SELECT 'Se elimino correctamente el pedido '+cast(@id_ventaF as varchar)
as mensaje
end
else
begin
    SELECT 'No se encontro la venta fisica' AS Mensaje;
end
end

select * from general.VENTA_FISICA
---Pedido existente
begin tran
select * from general.VENTA_FISICA
execute dbo.CancelarVentaFisica 10
select * from general.VENTA_FISICA
rollback
---Pedido inexistente
begin tran
select * from general.VENTA_FISICA
execute dbo.CancelarVentaFisica 15
select * from general.VENTA_FISICA
rollback
-----
```

```

--Realice un procedimiento almacenado para borrar un usuario, recibe como
--parámetros el nombre de usuario.
CREATE or alter PROCEDURE dbo.EliminaUsuario
    @nombre_usu varchar(20)
AS
BEGIN
    if exists (select um.id_usuario from general.USUARIO_COMUN um
               left join general.COMPRA_CARRITO cc on cc.id_usuario = um.id_usuario
               left join general.MASCOTA m on m.id_usuario = um.id_usuario
               left join general.REGISTRO_BRAZALETE rb on rb.id_mascota =
m.id_mascota
               left join general.CONSULTA c on c.id_mascota = m.id_mascota
               left join general.receta rec on rec.id_consulta = c.id_consulta
               where um.nom = @nombre_usu and cc.id_ventaC is null and id_carrito is
null and
               m.id_mascota is null and rb.id_registro is null and c.id_consulta is null and
               rec.id_medicamento is null)
    begin
        delete from general.USUARIO_COMUN where nom = @nombre_usu
```

```

        SELECT 'usuario eliminado' AS Mensaje;
    end
    else
    begin
        SELECT 'No se encontro el usuario' AS Mensaje;
    end

END
-----Elimina usuario comun
begin tran
SELECT * from general.USUARIO_COMUN
execute dbo.EliminaUsuario 'hugo'
SELECT 'DESPUES DEL PROCEDIMIENTO'
SELECT * from general.USUARIO_COMUN
rollback tran

-----
-- Crear procedimientos almacenados para las tareas del Gerente
CREATE or alter PROCEDURE ContratarPersonal
    @NombreUsuario VARCHAR(50),
    @TipoUsuario VARCHAR(20),
    @CentroCuidado VARCHAR(50),
    @Estado VARCHAR(50)
AS
BEGIN
    -- Crear el usuario
    EXEC sp_adduser @NombreUsuario, @NombreUsuario, @TipoUsuario;

    -- Asignar al centro de cuidado
    EXEC sp_addrolemember @TipoUsuario, @CentroCuidado;

    -- Asignar al estado
    EXEC sp_addrolemember @Estado, @NombreUsuario;
END;

-----
-- Crear procedimientos almacenados para las tareas del Gerente
CREATE or alter PROCEDURE ContratarPersonal
    @NombreUsuario VARCHAR(50),
    @TipoUsuario VARCHAR(20),
    @CentroCuidado VARCHAR(50),
    @Estado VARCHAR(50)
AS
BEGIN
    -- Crear el usuario
    EXEC sp_adduser @NombreUsuario, @NombreUsuario, @TipoUsuario;

    -- Asignar al centro de cuidado

```

```

EXEC sp_addrolemember @TipoUsuario, @CentroCuidado;

-- Asignar al estado
EXEC sp_addrolemember @Estado, @NombreUsuario;
END;
-----
```

```

----ReasignarPersonal
CREATE PROCEDURE ReasignarPersonal
    @NombreUsuario VARCHAR(50),
    @NuevoCentroCuidado VARCHAR(50)
AS
BEGIN
    -- Reasignar al nuevo centro de cuidado
    EXEC sp_droprolemember @NombreUsuario, @NombreUsuario;
    EXEC sp_addrolemember @NombreUsuario, @NuevoCentroCuidado;
END;
```

```

---DespedirPersonal
CREATE PROCEDURE DespedirPersonal
    @NombreUsuario VARCHAR(50)
AS
BEGIN
    -- Eliminar el usuario
    EXEC sp_dropuser @NombreUsuario;
END;
```

-----TRIGGER-----

-- trigger que calcula el costo total de la consulta

```

CREATE or alter TRIGGER general.CalcularCostoTotal
ON general.RECETA
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    UPDATE consulta
    SET costo_total = (
        SELECT ISNULL(SUM(m.costo * r.cantidad), 0) + 150
        FROM general.RECETA r
        INNER JOIN catalogos.MEDICAMENTO m ON r.id_medicamento = m.id_medicamento
        WHERE r.id_consulta = consulta.id_consulta
    )
    FROM general.CONULTA consulta
    WHERE consulta.id_consulta IN (
```

```

SELECT DISTINCT id_consulta FROM inserted
UNION
SELECT DISTINCT id_consulta FROM deleted
)
end

```

--trigger que limita que un cuidador tenga maximo 5 mascotas que atender

```

CREATE or alter TRIGGER general.LimitarCantidadMascotas
ON general.mascota
AFTER INSERT, UPDATE
AS
begin
    DECLARE @TotalMascotas INTeger
    set @TotalMascotas = (select COUNT(*) FROM general.MASCOTA where id_empleado =
(select id_empleado from inserted))
    IF @TotalMascotas > 5
        BEGIN
            print 'No se pueden agregar más de 5 mascotas.'
            ROLLBACK TRANSACTION
            RETURN
        END
    end

```

-- triger que hace la cantidad maxima de veterinarios por centro es de 3

```

CREATE or alter TRIGGER empleados.LimitarCantidadVeterinariosPorCentro
ON empleados.empleado
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @CentroID INTEGER, @TotalVeterinarios INTEGER

    SELECT @CentroID = e.id_centro
    FROM empleados.EMPLEADO e
    INNER JOIN inserted i ON e.id_empleado = i.id_empleado

    SELECT @TotalVeterinarios = COUNT(*)
    FROM empleados.VETERINARIO v
    INNER JOIN empleados.EMPLEADO e ON v.id_empleado = e.id_empleado
    WHERE e.id_centro = @CentroID

    IF @TotalVeterinarios > 3
        BEGIN
            print 'No se pueden agregar más de 3 veterinarios para el centro '
            ROLLBACK TRANSACTION
            RETURN
        END

```

```
END
END
```

---trigger actualiza el costo del monto parcial de car prod

```
CREATE TRIGGER general.ActualizarMontoParcialCarProd
ON general.CAR_PROD
AFTER INSERT
AS
BEGIN
    -- Actualizar el monto parcial en la tabla CAR_PROD
    UPDATE cp
    SET montoParcial = p.precio
    FROM general.CAR_PROD cp
    INNER JOIN inserted i ON cp.id_producto = i.id_producto
    INNER JOIN catalogos.PRODUCTO_CENTRAL p ON cp.id_producto = p.id_producto
    WHERE cp.id_carrito = i.id_carrito
END
```

--- trigger que actualiza el monto parcial de compra carrito

```
CREATE TRIGGER general.ActualizarMontoParcial
ON general.CAR_PROD
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    -- Actualizar las compras de carrito afectadas
    UPDATE compra
    SET montoParcial = ISNULL((SELECT SUM(p.precio * c.cantidad)
        FROM general.CAR_PROD c
        INNER JOIN catalogos.PRODUCTO_CENTRAL p ON c.id_producto = p.id_producto
        WHERE c.id_carrito = compra.id_carrito ), 0)
    FROM general.COMPRA_CARRITO compra
    WHERE compra.id_carrito IN (
        SELECT DISTINCT id_carrito FROM inserted
        UNION
        SELECT DISTINCT id_carrito FROM deleted
    )
END
```

--- trigger que aplica descuento y sumatoria de la venta fisica

```
CREATE or alter TRIGGER general.AplicarDescuentoSumarMontoDetalleCuenta
ON general.DETALLE CUENTA
AFTER INSERT
AS
```

```

BEGIN
    -- Actualizar montoParcial con descuento por oferta
    UPDATE dc
    SET montoParcial = CASE
        WHEN o.tipo = 'N' THEN p.precio * (1 - (o.porcentaje / 100))
        WHEN o.tipo = 'L' THEN p.precio * (1 - (o.porcentaje / 100))
        ELSE p.precio
    END
    FROM general.DETALLE CUENTA dc
    INNER JOIN inserted i ON dc.id_ventaF = i.id_ventaF AND dc.id_producto = i.id_producto
    INNER JOIN catalogos.PRODUCTO TIENDA p ON dc.id_producto = p.id_producto
    INNER JOIN catalogos.OFERTA o ON p.id_oferta = o.id_oferta

    -- Actualizar total en VENTA_FISICA
    UPDATE vf
    SET total = ISNULL((
        SELECT SUM(d.montoParcial * d.cantidad)
        FROM general.DETALLE CUENTA d
        WHERE d.id_ventaF = vf.id_ventaF
    ), 0)
    FROM general.VENTA_FISICA vf
    INNER JOIN inserted i ON vf.id_ventaF = i.id_ventaF
END

```

--trigger que pone la comision al vendedor

```

CREATE TRIGGER general.CalcularComisionSumarSueldoTotal
ON general.VENTA_FISICA
AFTER INSERT
AS
BEGIN
    -- Actualizar comision y sumar al sueldo total del vendedor
    UPDATE v
    SET comision = (i.total * 0.15),
        sueldo_total = sueldo_total + (i.total * 0.15)
    FROM empleados.VENDEDOR v
    INNER JOIN inserted i ON v.id_empleado = i.id_empleado
END

```

--trigger que valida la existencia del estado antes de insertarse

```

CREATE or alter TRIGGER empleados.tr_verificar_estado_administrativo
ON empleados.ADMINISTRATIVO
instead of INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE id_estado NOT IN (SELECT id_estado
    FROM catalogos.ESTADO))

```

```

BEGIN
    RAISERROR('El estado especificado no existe en la tabla ESTADO.', 16, 1)
    ROLLBACK TRANSACTION
END
END

```

Valida_trigger código

```

/*
 * EQUIPO 7 GUARDERIA DE MASCOTAS
 * CORIA MARTINEZ GUSTAVO
 * ORONZOR MONTES MANASES LEONEL
 * PANG ARAIZAGA IAN
 */

-- trigger que calcula el costo total de la consulta
select * from general.CONSULTA
select * from general.RECETA where id_consulta = 2
select * from catalogos.MEDICAMENTO
insert into general.receta values
(2,3,2)

--trigger que limita que un cuidador tenga maximo 5 mascotas que atender
SELECT * from general.MASCOTA
begin tran
declare @cont int = 5
declare @i int = 1
while @i <= @cont
begin
INSERT INTO
general.MASCOTA(id_empleado,id_guarderia,id_usuario,nombre,especie,sexo,raza,razgos,
edad) values
(11,1,3,'LUKE', 'perro', 'macho', 'pastor', 'negro con café', 1)
set @i = @i+1
end
rollback tran
-- triger que hace la cantidad maxima de veterinarios por centro es de 3

BEGIN TRANSACTION;
update empleados.EMPLEADO set id_centro = 1
where id_empleado in (8, 12, 16)
SELECT * from empleados.VETERINARIO v inner join empleados.EMPLEADO e on
v.id_empleado = e.id_empleado
rollback TRANSACTION;

---trigger actualiza el costo del monto parcial de car prod

begin tran

```

```

INSERT INTO general.CAR_PROD (id_carrito, id_producto, montoParcial, cantidad)
VALUES (179012, 1234567890, 100, 3);
SELECT * from general.CAR_PROD where id_carrito = 179012
rollback tran

```

```

--- trigger que actualiza el monto parcial de compra carrito
begin tran
INSERT INTO general.car_prod (id_carrito, id_producto, montoparcial, cantidad)
VALUES
(135678, 4567890123, 4, 3)
select * from catalogos.PRODUCTO_CENTRAL
select * from general.CAR_PROD
select * from general.COMPRA_CARRITO
rollback tran

```

```

--- trigger que aplica descuento y sumatoria de la venta fisica
begin tran
INSERT INTO general.venta_fisica (id_empleado, total)
VALUES
(2, 25.99)
INSERT INTO general.detalle_cuenta (id_ventaF, id_producto, montoparcial, cantidad)
VALUES
(11, 7891123456, 25.99, 3)
select * from catalogos.OFERTA
select * from catalogos.PRODUCTO_TIENDA
select * from general.DETALLE CUENTA
select * from general.VENTA_FISICA
rollback tran

```

```
DBCC CHECKIDENT ('general.venta_fisica', RESEED, 10);
```

```

--trigger que pone la comision al vendedor
begin tran
INSERT INTO general.venta_fisica (id_empleado, total)
VALUES
(2, 25.99)
INSERT INTO general.detalle_cuenta (id_ventaF, id_producto, montoparcial, cantidad)
VALUES
(10, 4567890123, 25.99, 3)
select * from general.VENTA_FISICA
select * from general.DETALLE CUENTA
select * from empleados.VENDEDOR
rollback tran

```

```

--trigger que valida la existencia del estado antes de insertarse un administrativo
begin tran
insert into empleados.ADMINISTRATIVO values (18, 33)
rollback tran

```

Seguridad código

```

/*
 * EQUIPO 7 GUARDERIA DE MASCOTAS
 * CORIA MARTINEZ GUSTAVO
 * ORONZOR MONTES MANASES LEONEL
 * PANG ARAIZAGA IAN
 */

use [Equipo7_PETCO]
go

-- Crear usuario administrador
CREATE LOGIN usuarioAdministrador WITH PASSWORD = '1234zaq*', CHECK_POLICY = OFF;
CREATE USER usuarioAdministrador FOR LOGIN usuarioAdministrador;
ALTER ROLE db_owner ADD MEMBER usuarioAdministrador;

-- Crear usuario gestor
CREATE LOGIN usuarioGestor WITH PASSWORD = '1234zaq*', CHECK_POLICY = OFF;
CREATE USER usuarioGestor FOR LOGIN usuarioGestor;
--GRANT INSERT, UPDATE ON Equipo7_PETCO TO usuarioGestor;

-- Crear usuario de solo consulta
CREATE LOGIN usuarioConsulta WITH PASSWORD = '1234zaq*', CHECK_POLICY = OFF;
CREATE USER usuarioConsulta FOR LOGIN usuarioConsulta;
--GRANT SELECT ON Equipo7_PETCO TO usuarioConsulta;

-----Usuario Comun (Registrado)
-- Crear usuario común
CREATE LOGIN usuarioComun WITH PASSWORD = 'Passw0rd$', CHECK_POLICY = OFF;
CREATE USER usuarioComun FOR LOGIN usuarioComun;

-- Asignar permisos
GRANT SELECT ON catalogos.producto_central TO usuarioComun; -- Permisos de consulta en el catálogo de productos
GRANT SELECT ON catalogos.centro TO usuarioComun; -- Permisos de consulta en la tabla de centros

-----
-----Usuario Personal de ayuda (Cuidadores)
-- Crear usuarios para el personal de cuidado en cada centro de guardería
CREATE LOGIN liderCuidadores WITH PASSWORD = 'Secur3ty$', CHECK_POLICY = OFF;
CREATE USER liderCuidadores FOR LOGIN liderCuidadores;

```

```
CREATE LOGIN cuidador1 WITH PASSWORD = '1234Abc*', CHECK_POLICY = OFF;
CREATE USER cuidador1 FOR LOGIN cuidador1;
```

```
CREATE LOGIN cuidador2 WITH PASSWORD = 'Pa$$w0rd', CHECK_POLICY = OFF;
CREATE USER cuidador2 FOR LOGIN cuidador2;
```

-- Crear más usuarios de cuidadores según sea necesario

-- Asignar roles y permisos

```
ALTER ROLE db_datareader ADD MEMBER liderCuidadores;
ALTER ROLE db_datawriter ADD MEMBER liderCuidadores;
```

```
ALTER ROLE db_datareader ADD MEMBER cuidador1;
ALTER ROLE db_datawriter ADD MEMBER cuidador1;
```

```
ALTER ROLE db_datareader ADD MEMBER cuidador2;
ALTER ROLE db_datawriter ADD MEMBER cuidador2;
```

-----Veterinario

```
CREATE LOGIN Veterinario WITH PASSWORD = 'P&ssword1', CHECK_POLICY = OFF;
CREATE USER Veterinario FOR LOGIN Veterinario;
```

-- Consultas médicas para las mascotas -- Otorgar permisos de consultas

Grant Insert ON general.consulta TO Veterinario; -- No tengo claro lo de que tabla hace esto.

-----Encargado de tienda

```
CREATE LOGIN EncargadoTienda WITH PASSWORD = 'Str0ngPass*', CHECK_POLICY = OFF;
CREATE USER EncargadoTienda FOR LOGIN EncargadoTienda;
```

-- Otorgar permisos de ventas personales

```
GRANT INSERT ON producto_tienda TO EncargadoTienda;
GRANT INSERT ON producto_central TO EncargadoTienda;
```

-- Otorgar permisos para actualizar el inventario

```
GRANT UPDATE ON producto_tienda TO EncargadoTienda;
GRANT UPDATE ON producto_central TO EncargadoTienda;
```

-----Personal Administrativo

```
CREATE LOGIN administrativo WITH PASSWORD = '1qaz2wsx*', CHECK_POLICY = OFF;
CREATE USER administrativo FOR LOGIN administrativo;
GRANT insert, update, delete, select ON general.Usuario_comun TO administrativo;
GRANT insert, update, delete, select ON empleados.veterinario TO administrativo;
GRANT insert, update, delete, select ON empleados.personal_ayuda TO administrativo;
```

```
GRANT insert, update, delete, select ON empleados.empleado TO administrativo;
GRANT insert, update, delete, select ON catalogos.Oferta TO administrativo;
GRANT insert, update, delete, select ON catalogos.Categoría_producto TO administrativo;
GRANT insert, update, delete, select ON catalogos.Producto_tienda TO administrativo;
GRANT insert, update, delete, select ON catalogos.Producto_central TO administrativo;
```

-----Gerente

```
-- Crear el usuario Gerente
CREATE LOGIN Gerente WITH PASSWORD = 'Sup3rPass*', CHECK_POLICY = OFF;
CREATE USER Gerente FOR LOGIN Gerente;

-- Otorgar permisos al usuario Gerente
EXEC sp_addrolemember 'db_securityadmin', 'Gerente';
GRANT ALTER ANY USER TO Gerente;
```

Reportes código

```
--REPORTE GUARDERIA
--1. El tipo de mascota que más se deja en la guardería para su cuidado (tipo y
--cantidad).
CREATE OR ALTER VIEW general.masTipoMascota
as
SELECT top 1 especie, COUNT(*) as cantidad from general.MASCOTA group by especie
order by COUNT(*) desc
GO

SELECT * from general.masTipoMascota

--2. En qué época del año se deja un mayor número de mascotas en la guardería
```

```
CREATE OR ALTER FUNCTION general.dimeEpoca (@fecha datetime)
```

```
returns varchar(20)
```

```
as
```

```
begin
```

```
    declare @epoca varchar(20)
```

```
    DECLARE @mes int
```

```
    SET @mes = MONTH(@fecha)
```

```
    IF @mes IN (12, 1, 2)
```

```
        SET @epoca = 'Invierno'
```

```
    ELSE IF @mes IN (3, 4, 5)
```

```
        SET @epoca = 'Primavera'
```

```
    ELSE IF @mes IN (6, 7, 8)
```

```
        SET @epoca = 'Verano'
```

```
    ELSE
```

```

SET @epoca = 'Otoño'
RETURN @epoca
end

CREATE OR ALTER VIEW general.masEpoca
as
select top 1 general.dimeEpoca(fecha_ini) as Epoca, count(general.dimeEpoca(fecha_ini))
as cantidad
from general.GUARDERIA group by general.dimeEpoca(fecha_ini) order by
count(general.dimeEpoca(fecha_ini)) desc
go

begin tran
update general.GUARDERIA set fecha_ini = '2022-03-21' where id_guarderia = 20
SELECT * from general.GUARDERIA where id_guarderia = 20
select * from general.masEpoca
rollback tran

--3. Listado de las mascotas y su dueño datos generales, número de días en la
--guardería, datos generales de ambos. El reporte se obtiene en un periodo de
--fechas

CREATE OR ALTER VIEW general.datosMascotaUsuario
as
SELECT (u.nom+' '+u.pat+' '+u.mat) as nombre_usuario, u.telefono, u.domicilio,
m.nombre as nombre_mascota, m.especie, m.sexo as sexo_mascota, m.raza, m.razgos,
m.edad as edad_mascota, datediff(day,g.fecha_ini,g.fecha_fin) as num_dias
from general.USUARIO_COMUN u
inner join general.MASCOTA m on u.id_usuario = m.id_usuario
inner join general.GUARDERIA g on g.id_guarderia = m.id_guarderia

SELECT * from general.datosMascotaUsuario

--4. Enfermedades más frecuentes (5), total de animales con dicha enfermedad
--ordenados de mayor a menor

CREATE OR ALTER VIEW general.enfermedadesMayor
as
SELECT diagnostico, count(diagnostico) as cantidad
from general.CONSULTA group by diagnostico

SELECT * from general.enfermedadesMayor order by cantidad desc

--REPORTE VETERINARIA

--1. Gastos de cada mascota, datos de la mascota, fechas y días de estancia,
--descripción de gatos y costo.

```

```

CREATE OR ALTER VIEW general.INFO_GASTO_X_MASCOTA
as
SELECT m.nombre as nombre_mascota, m.especie, m.raza, m.edad, m.razgos,
sum(co.costo_total) as costo_veterinario, datediff(day,g.fecha_ini,fecha_fin) as
dias_estancia,
g.costo as costo_estancia, (sum(co.costo_total)+g.costo) total_gastado
from general.MASCOTA m
left join general.CONSULTA co on m.id_mascota = co.id_mascota
left join general.GUARDERIA g on g.id_guarderia = m.id_guarderia
group by m.nombre, m.especie, m.raza, m.edad, m.razgos, g.costo, g.fecha_ini, fecha_fin

```

```
SELECT * from general.INFO_GASTO_X_MASCOTA
```

--2. Listado con datos generales de las consultas por periodos de tiempo,
--incluyendo veterinario que lo atendió, diagnóstico y medicamento.

```

CREATE OR ALTER VIEW general.datos_consulta
as
SELECT (e.nombre+' '+e.paterno+' '+e.materno) as nombre_veterinario,
c.fecha, c.diagnostico, m.nombre_medicamento from general.CONSULTA c
inner join empleados.VETERINARIO v on c.id_empleado = v.id_empleado
inner join empleados.EMPLEADO e on v.id_empleado = e.id_empleado
inner join general.receta r on r.id_consulta = c.id_consulta
inner join catalogos.MEDICAMENTO m on r.id_medicamento = m.id_medicamento
group by e.nombre, e.paterno, e.materno, c.fecha, c.diagnostico, m.nombre_medicamento

```

```
SELECT * from general.datos_consulta order by fecha
```

--3. Inventario de medicamentos, con su costo por centro.

--4. Enfermedades más frecuentes, por centro, ordenadas de mayor a menor

```

CREATE OR ALTER VIEW general.enfermedad_x_centro
as
SELECT c.id_centro, co.diagnostico as enfermedad, count(co.diagnostico) as cantidad from
catalogos.CENTRO c
inner join general.GUARDERIA g on c.id_centro = g.id_centro
inner join general.MASCOTA m on m.id_guarderia = g.id_guarderia
inner join general.CONSULTA co on co.id_mascota = m.id_mascota
group by c.id_centro, co.diagnostico

```

```
SELECT * from general.enfermedad_x_centro order by id_centro
```

--5. Reporte de ventas de medicamentos en un periodo de tiempo, incluyendo
--cantidad y monto total

```
CREATE OR alter VIEW general.reporte_venta_medicamento
```

```

as
SELECT co.fecha, m.nombre_medicamento, sum(r.cantidad) as cantidad,
(r.cantidad*m.costo) as monto_total
from general.CONSULTA co
inner join general.receta r on r.id_consulta = co.id_consulta
inner join catalogos.MEDICAMENTO m on r.id_medicamento = m.id_medicamento
group by m.nombre_medicamento, r.cantidad, m.costo, co.fecha, m.nombre_medicamento

```

```
SELECT * from general.reporte_venta_medicamento order by fecha
```

--REPORTE VENTAS

--1. El centro con mayor número de ventas en un periodo de tiempo. (Separar las ventas en línea de las ventas físicas de cada estado).

```
CREATE OR ALTER VIEW general.REPORTE_VENTAS_FISICAS
```

as

```

SELECT top 1 es.nombre as estado, c.id_centro,
sum(c.id_centro) as ventas_fisicas
from general.VENTA_FISICA vf
inner join empleados.VENDEDOR ve on vf.id_empleado = ve.id_empleado
inner join empleados.EMPLEADO e on e.id_empleado = vf.id_empleado
inner join catalogos.CENTRO c on c.id_centro = e.id_centro
inner join catalogos.ESTADO es on es.id_estado = c.id_estado
group by es.nombre, c.id_centro
order by sum(vf.id_empleado) desc

```

```
SELECT * from general.REPORTE_VENTAS_FISICAS
```

--2. Los artículos más vendidos y los menos vendidos por categoría.

--Ventas físicas

```
CREATE OR ALTER VIEW general.datos_ventasF
```

as

```

SELECT cp.descripcion as categoria, pt.detalles as producto,
sum(dc.cantidad) as cantidad_vendida from general.VENTA_FISICA vf
inner join general.DETALLE CUENTA dc on dc.id_ventaF = vf.id_ventaF
inner join catalogos.PRODUCTO TIENDA pt on pt.id_producto = dc.id_producto
inner join catalogos.CATEGORIA_PRODUCTO cp on cp.id_categoria = pt.id_categoria
group by cp.descripcion, pt.detalles, cp.id_categoria

```

```
CREATE OR ALTER FUNCTION general.REPORTE_MAS_VENDIDOF()
returns @result table (categoria varchar(40),
```

```
                                producto varchar(40),
                                cantidad numeric(4,0))
```

as

begin

```
    declare @max int = (SELECT count(id_categoria) from
catalogos.CATEGORIA_PRODUCTO)
```

```
    declare @count int = 1
```

```

WHILE @count <= @max
begin
    insert into @result
    SELECT top 1 * from general.datos_ventasF
        where categoria = (SELECT descripcion from
catalogos.CATEGORIA_PRODUCTO
                                         where id_categoria = @count)
    order by cantidad_vendida desc
    insert into @result
    SELECT top 1 * from general.datos_ventasF
        where categoria = (SELECT descripcion from
catalogos.CATEGORIA_PRODUCTO
                                         where id_categoria = @count)
    order by cantidad_vendida asc
    set @count = @count+1
end
return
end

```

SELECT * from general.REPORTE_MAS_VENDIDOF()

--Ventas virtuales

```

CREATE OR ALTER VIEW general.datos_ventasV
as
SELECT cap.descripcion as categoria, pc.detalles as producto,
sum(cp.cantidad) as cantidad_vendida from general.COMPRA_CARRITO cc
inner join general.CAR_PROD cp on cp.id_carrito = cc.id_carrito
inner join catalogos.PRODUCTO_CENTRAL pc on pc.id_producto = cp.id_producto
inner join catalogos.CATEGORIA_PRODUCTO cap on cap.id_categoria = pc.id_categoria
group by cap.descripcion, pc.detalles, cap.id_categoria

```

CREATE OR ALTER FUNCTION general.REPORTE_MAS_VENDIDOV()
returns @result table (categoria varchar(40),

producto varchar(40),
cantidad numeric(4,0))

as

begin

```

declare @max int = (SELECT count(id_categoria) from
catalogos.CATEGORIA_PRODUCTO)

```

declare @count int = 1

WHILE @count <= @max

begin

insert into @result

```

SELECT top 1 * from general.datos_ventasV
    where categoria = (SELECT descripcion from
catalogos.CATEGORIA_PRODUCTO
                                         where id_categoria = @count)
    order by cantidad_vendida desc

```

```

insert into @result
SELECT top 1 * from general.datos_ventasV
where categoria = (SELECT descripcion from
catalogos.CATEGORIA_PRODUCTO
where id_categoria = @count)
order by cantidad_vendida asc
set @count = @count+1
end
return
end

```

SELECT * from general.REPORTE_MAS_VENDIDO()

--3. Época en la que más se vende. Época del año y monto total

--4. Los 5 empleados con mayor comisión, este reporte se obtiene de manera
--mensual

--5. Inventario de las tiendas de cada tienda

Conclusión

Este proyecto ha sido exitoso y ha demostrado ser una herramienta valiosa para gestionar de manera eficiente y efectiva la información relacionada con las mascotas y sus dueños. A través de la implementación de esta base de datos, se logró centralizar la información en un único sistema, lo que facilitó el acceso y la actualización de los datos, así como la generación de informes y estadísticas.

La base de datos diseñada proporciona una solución integral para la gestión de la guardería de mascotas. Permite almacenar información detallada sobre cada mascota, incluyendo su nombre, raza, edad e historial médico. Además, se pueden registrar los datos de los dueños de las mascotas, incluyendo sus nombres, direcciones, números de contacto y cualquier información adicional relevante.

La base de datos también ofrece la capacidad de programar y gestionar las reservas de los dueños. Los registros de entrada y salida de las mascotas se pueden registrar fácilmente, lo que brinda un seguimiento preciso de su estancia en la guardería.

Además de las funcionalidades básicas de la base de datos, se implementaron medidas de seguridad para proteger la información confidencial de los dueños y las mascotas. Se establecieron permisos de acceso para controlar quién puede acceder y modificar los datos, garantizando la privacidad y la integridad de la información almacenada.

EQUIPO	CUMPLE	NO CUMPLE
DOCUMENTO DE WORD	x	
ENUNCIADO	x	
DISEÑO CONCEPTUAL		
* Se entregó archivo .dia	x	
DISEÑO LOGICO		
• Modelo relacional (en ERStudio)	x	
• Diccionario de datos (nombre del objeto, tipo (tabla, relación, procedimiento almacenado, índice, función, etc.), descripción, en caso de las tablas para cada atributo poner tipo de datos y longitud)	x	
NORMALIZACIÓN		
• Explicar cada forma normal	x	
DISEÑO FÍSICO		
DCL		
• Borrado en cascada	x	
• Uso de índices	x	
• Indices non-clustered y UNIQUE	x	
DDL		
• llaves artificiales	x	
• columnas calculadas		
• llaves naturales	x	
DML		
• 2 Funciones	x	
• Disparadores	x	
• 3 Vistas	x	
• Uso de transacciones	x	
• Manejo de errores	x	
• Llamadas a procedimiento almacenados	x	
Uso de restricciones vistas en clase:		
• Unique	x	
• check	x	
• FK	x	
• PK	x	
• Valores por default	x	
3 consultas (al menos)		
• Diferentes tipos de joins	x	
• funciones de agregación	x	
• Algebra relacional		x
• Subconsultas	x	
Buscador con like		x
Procedimiento almacenado para sentencias DML		
Insert	x	
Update	x	
Deleted	x	
2 ESQUEMAS	x	

SEGURIDAD	x	
CONSIDERACIONES (anotar las consideraciones semánticas una por renglón, las siguientes son ejemplos)		
CS1: P- perro , G-gato		x
CS2: N-normal, L-limitada	x	
CS3: No mayor a 40 dias	x	
CS4: precio-descuento	x	
CS5: 300*(fecha_fin-fecha_ini)	x	
CS6: 0-No, 1-Si	x	
ESTADISTICAS (anotar las realizadas, las siguientes son ejemplos)		
1. El tipo de mascota que más se deja en la guardería para su cuidado (tipo y cantidad).	x	
2. En qué época del año se deja un mayor número de mascotas en la guardería	x	
3. Listado de las mascotas y su dueño datos generales, número de días en la guardería, datos generales de ambos. EL reporte se obtiene en un periodo de fechas	x	
4. Enfermedades más frecuentes (5), total de animales con dicha enfermedad ordenados de mayor a menor	x	
1. Gastos de cada mascota, datos de la mascota, fechas y días de estancia, descripción de gatos y costo.	x	
2. Listado con datos generales de las consultas por períodos de tiempo, incluyendo veterinario que lo atendió, diagnóstico y medicamento.	x	
3. Inventario de medicamentos, con su costo por centro.		x
4. Enfermedades más frecuentes, por centro, ordenadas de mayor a menor	x	
5. Reporte de ventas de medicamentos en un periodo de tiempo, incluyendo cantidad y monto total	x	
1. El centro con mayor número de ventas en un periodo de tiempo. (Separar las ventas en línea de las ventas físicas de cada estado).	x	
2. Los artículos más vendidos y los menos vendidos por categoría. Ventas físicas	x	
3. Época en la que más se vende. Época del año y monto total		x
4. Los 5 empleados con mayor comisión, este reporte se obtiene de manera mensual		x
5. Inventario de las tiendas de cada tienda		x
<i>Las estadísticas deben mostrar 10 registros para cada caso</i>	x	

CARGA INICIAL (anotar las realizadas, las siguientes son ejemplos)		
20 ventas	x	
10 gestores	x	
5 vendedores	x	
4 categorias	x	
5 productos para cada categoria	x	
5 ofertas	x	
5 cestas pendientes de compra	x	
SCRPTS DOCUMENTADO (autor, fecha, descripción)	x	
ARCHIVOS A ENTREGAR (6 SCRIPTS)		
dia	x	
ERStudio	x	
seguridad.sql	x	
creabase.sql	x	
dml.sql	x	
informes.sql (estadisticas)	x	
carga inicial.sql	x	
valida triggers.SQL	x	

EQUIPO NO:		
Nombre: Coria Martinez Gustavo	Núm. Lista:	6
Nombre: Oronzor Montes Manases Leonel	Núm. Lista:	16
Nombre: Pang Araizaga Ian	Núm. Lista:	17