S.I.V.E.

Base de Datos II 3°IF

A.F.M Tech System

Rol	Apellidos	Nombre	C.I.	Email	Cel.
Integrante 3	Aguilera	Estela	4621249-5	estela231092@gmail.com	092856282
Sub-coordinadora	Aguirre	Ashelem	5254868-6	ashe_713@hotmail.com	095890552
Integrante 1	Fernández	Matías	4550079-0	mfalassio@gmail.com	091080985
Coordinador	Martínez	Federico	4591407-6	martinez.fl@gmail.com	094540813
Integrante 2	Tomassini	Dino	4235739-8	dinotomassini@gmail.com	099746158

Docente: DIEGO ANDRES **PEREIRA SAPERE**

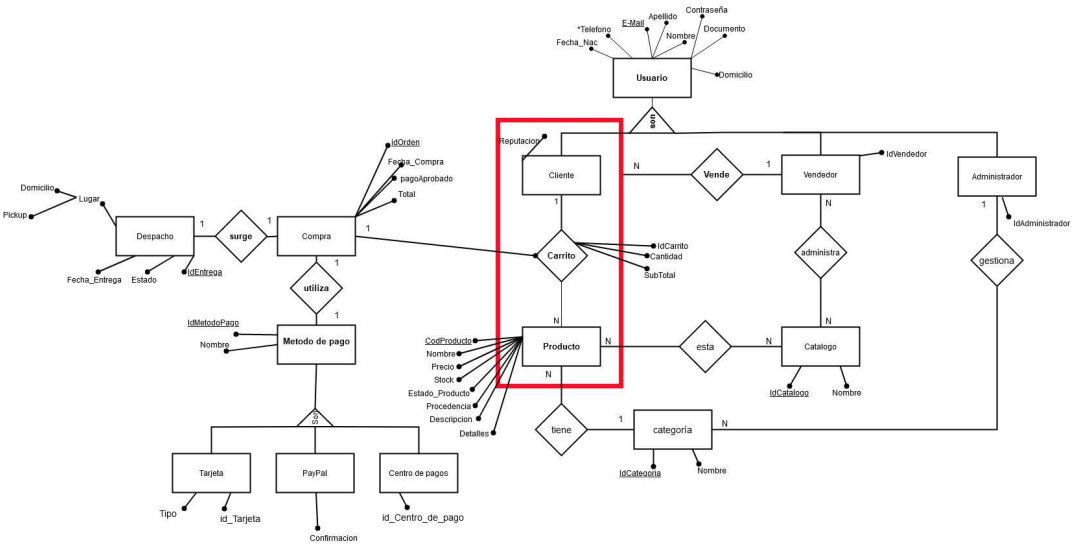
SEGUNDA ENTREGA

Fecha de culminación:
06/09/2021

INDICE

1. Modelo conceptual (DER) con las correcciones.

Versión completa.



2. Esquema Relacional normalizado (3ra. Forma Normal).

Usuario
PRS-Usuario(ATR, DMN, dom)
ATR={E-Mail,Password, Nombre, Apellido, Fecha_Nac, Documento,Calle, Numero,Esquina }
DMN={varchart,date,int}
dom={E-Mail->varchar,Password->varchar, Nombre->varchar, Apellido->varchar, Fecha_Nac->date, Documento->varchar,Calle->varchar, Numero->int,Esquina->varchar}
RS-Usuario=(PRS-Usuario, M, SC)
M={“todo los usuarios ingresados en el sistema”}
SC={PK->E-Mail,NN->Password, NN->Nombre, NN->Apellido, NN->Fecha_Nac, NN->Documento,NN->Calle, NN->Numero,NN->Esquina, Unique->Documento}

Telefono
PRS-Telefono(ATR, DMN, dom)
ATR={E-Mail, Telefono}
DMN={Varchar}
dom={E-Mail->varchar[40], Telefono->varchar[10]}
RS-Telefono=(PRS-Telefono, M, SC)
M={“todo los teléfonos ingresados en el sistema”}
SC={PK->(E-mail, telefono), NN->E-mail, NN->Telefono, FK=E-mail->Usuario(E-mail)}

Cliente
PRS-Cliente(ATR, DMN, dom)
ATR={correo, Reputacion}
DMN={varchar, int}
dom={correo->varchar[40], Reputacion->int}
RS-Cliente=(PRS-Cliente, M, SC)
M={“todo los clientes ingresados en el sistema”}
SC={PK->correo, NN->correo, NN->Reputacion, FK=correo>Usuario(correo)}

Vendedor
PRS-Vendedor(ATR, DMN, dom)
ATR={correo, IdVendedor}
DMN={varchar, int}
dom={correo->varchar(40), IdVendedor->int}
RS-Vendedor=(PRS-Vendedor, M, SC)
M={“todo los vendedores ingresados en el sistema”}
SC={PK->correo, NN->correo, NN->IdVendedor, Unique->IdVendedor,
FK=correo->Usuario(correo)}

Administrador
PRS-Administrador
ATR={correo, IdAdministrador}
DMN={varchar}
dom={correo->varchar(40), IdAdministrador->Int}
RS-Administrador=(PRS-Administrador, M, SC)
M={“todo los administradores ingresados en el sistema”}
SC={PK->correo, NN->correo, NN->IdAdministrador, Unique->IdAdministrador,
FK=correo->Usuario(correo)}

Catálogo
PRS-Catalogo(ATR, DMN, dom)
ATR={IdCatalogo, Nombre}
DMN={int, varchar}
dom={IdCatalogo->int, Nombre->varchar(20)}
RS-Catalogo=(PRS-Catalogo, M, SC)
M={“todo los catálogos ingresados en el sistema”}
SC={PK->IdCatalogo, NN->IdCatalogo, NN->Nombre}

Categoria
PRS-Categoria(ATR, DMN, dom)
ATR={IdCategoria, Nombre}
DMN={int, varchar}
dom={IdCategoria->int, Nombre->varchar(20)}
RS-Categoria=(PRS-Categoria, M, SC)
M={“todo las categorias ingresadas en el sistema”}
SC={PK->IdCategoria, NN->IdCategoria, NN->Nombre}

Producto
PRS-Producto(ATR, DMN, dom)
ATR={CodProducto, Nombre, Precio, Stock, Estado_Producto, Procedencia,Descripcion, Detalle}
DMN={int, varchar}
dom={CodProducto->int, Nombre->varchar(20), Precio->int, Stock->int, Estado_Producto->varchar(10), Procedencia->varchar(15),Descripcion->varchar(70), Detalle->varchar(60)}
RS-Producto=(PRS-Producto, M, SC)
M={“todo los productos ingresados en el sistema”}
SC={PK->CodProducto, NN->CodProducto, NN->Nombre, NN->Precio, NN->Stock, NN->Estado_Producto, NN->Procedencia,NN->Descripcion, NN->Detalle}

Compra
PRS-Compra(ATR, DMN, dom)
ATR={IdOrden, Fecha_Compra, pagoAprobado, Total, E-Mail, CodProducto}
DMN={int, date,boolean}
dom={IdOrden->int, Fecha_Compra->date, pagoAprobado->boolean, Total->int, correo->varchar(40), CodProducto->int}
RS-Compra=(PRS-Compra, M, SC)
M={“toda las compras ingresadas en el sistema”}
SC={PK->IdOrden, NN->IdOrden, NN->Fecha_Compra, NN->pagoAprobado, NN->Total, NN->correo, NN->CodProducto, Unique->correo, CodProducto, FK=E-Mail->Cliente(correo), CodProducto->Producto(CodProducto)}

Metodo de Pago
PRS-Metodo de Pago(ATR, DMN, dom)
ATR={IdMetodoPago, Nombre}
DMN={int, varchar}
dom={IdMetodoPago->int, Nombre->varchar(20)}
RS-Metodo de Pago=(PRS-Metodo de Pago, M, SC)
M={“todo los metodos de pagos ingresados en el sistema”}
SC={PK->IdMetodoPago, NN->IdMetodoPago, NN->Nombre}

Tarjeta
PRS-Tarjeta(ATR, DMN, dom)
ATR={IdMetodoPago, IdTarjeta}
DMN={int}
dom={IdMetodoPago->int, IdTarjeta->int}
RS-Tarjeta=(PRS-Tarjeta, M, SC)
M={“todas las las terjetas ingresadas en el sistema”}
SC={PK->IdMetodoPago, NN->IdMetodoPago, NN->IdTarjeta, Unique->IdTarjeta, FK=IdMetodoPago->Metodo de Pago(IdMetodoPago)}

PayPal
PRS-PayPal(ATR, DMN, dom)
ATR={IdMetodoPago, confirmacion}
DMN={int, boolean}
dom={IdMetodoPago->int, confirmacion->boolean}
RS-PayPal=(PRS-PayPal, M, SC)
M={“todos los métodos de pago paypal ingresados en el sistema”}
SC={PK->IdMetodoPago, NN->IdMetodoPago, NN->confirmacion,vFK=IdMetodoPago->Metodo de Pago(IdMetodoPago)}

Centro de Pagos
PRS-Centro de Pagos(ATR, DMN, dom)
ATR={IdMetodoPago, IdTarjeta}
DMN={int}
dom={IdMetodoPago->int, IdCentroDePago->int}
RS-Centro de Pagos=(PRS-Centro de pagos, M, SC)
M={“todo los centros de pagos ingresados en el sistema”}
SC={PK->IdMetodoPago, NN->IdMetodoPago, NN->IdCentroDePago, Unique->IdCentroDePago, FK=IdMetodoPago->Metodo de Pago(IdMetodoPago)}

Despacho
PRS-Despacho(ATR, DMN, dom)
ATR={IdEntrega, Estado_Entrega, Fecha_Entrega, Lugar}
DMN={int, varchar, date}
dom={IdEntrega->int, Estado_Entrega->varchar(20), Fecha_Entrega->date, Lugar->varchar(10)}
RS-Despacho=(PRS-Despacho, M, SC)
M={“todas las entregas registradas en el sistema”}
SC={PK->IdEntrega, NN->IdEntrega, NN->Estado_Entrega, NN->Fecha_Entrega, NN->Lugar}

Esta
PRS-Esta(ATR, DMN, dom)
ATR={IdCatalogo, CodProducto}
DMN={int}
dom={IdCatalogo->int, CodProducto->int}
RS-Esta=(PRS-Esta, M, SC)
M={“todas las interacciones entre Catalogo y Productos”}
SC={PK->(IdCatalogo, CodProducto), NN->IdCatalogo, NN->CodProducto, FK=IdCatalogo->Catalogo(IdCatalogo), CodProducto->Producto(CodProducto)}

Administra
PRS-Administra(ATR, DMN, dom)
ATR={E-Mail, IdCatalogo}
DMN={int, varchar}
dom={correo->varchar(40), IdCatalogo->int}
RS-Administra=(PRS-Administra, M, SC)
M={“todas las interacciones entre los catalogos y los vendedores”}
SC={PK->(E-Mail, IdCatalogo), NN->correo, NN->IdCatalogo, FK=correo->Vendedor(correo), IdCatalogo->Catalogo(IdCatalogo)}

Gestiona
PRS-Gestiona(ATR, DMN, dom)
ATR={IdCategoria, E-Mail}
DMN={int, varchar}
dom={IdCategoria->int, correo->varchar(40)}
RS-Gestiona=(PRS-Gestiona, M, SC)
M={“toda las interacciones entre los administradores y las categorías ”}
SC={PK->IdCategoria,NN->IdCategoria, NN->correo, FK=IdCategoria->Categoria(IdCategoria), correo->Administrador(correo)}

Tiene
PRS-Tiene(ATR, DMN, dom)
ATR={CodProducto, IdCategoria}
DMN={int}
dom={CodProducto->int, IdCategoria->int}
RS-Tiene=(PRS-Tiene, M, SC)
M={“todas las interacciones entre productos y categorias ingresadas al sistema”}
SC={PK->CodProducto, NN->CodProducto, NN->IdCategoria, FK=CodProducto->Productos(CodProducto),IdCategoria->Categorias(IdCategoria)}

Surge
PRS-Utiliza(ATR, DMN, dom)
ATR={IdOrden, IdEntrega}
DMN={int}
dom={IdOrden->int, IdEntrega->int}
RS-Surge=(PRS-Surge, M, SC)
M={“todas las interacciones entre las compras y las entregas ingresadas en el sistema”}
SC={PK->IdOrden, NN->IdOrden, NN->IdEntrega, FK=IdOrden->Compra(IdOrden), IdEntrega->Despacho(IdEntrega)}

Utiliza
PRS-Utiliza(ATR, DMN, dom)
ATR={IdOrden, IdMetododPago}
DMN={int}
dom={IdOrden->int, IdMetodoPago->int}
RS-Utiliza=(PRS-Utiliza, M, SC)
M={“todas las interacciones entre las compras y metodos de pagos ingresados en el sistema”}
SC={PK->IdOrden, NN->IdOrden, NN->IdMetodoPago, FK=IdOrden->Compra(IdOrden), IdMetodoPago->Metodo de Pago(IdMetodoPago)}

Carrito
PRS-Carrito(ATR, DMN, dom)
ATR={CodProducto, correo, IdOrden, IdCarrito, Cantidad, SubTotal}
DMN={int, varchar}
dom={CodProducto->int, correo->varchar(40), IdOrden->int, IdCarrito->int, Cantidad->int, SubTotal->int}
RS-Carrito=(PRS-Carrito, M, SC)
M={“todas las interacciones entre compras, cliente y productos ingresados en el sistema”}
SC={PK->(CodProducto, IdCarrito), NN->CodProducto, NN->correo, NN->IdOrden, NN->IdCarrito, NN->Cantidad, NN->SubTotal, FK=CodProducto->Producto(CodProducto), correo->Cliente(correo), IdOrden->Compra(IdOrden)}

Vende
PRS-Vende(ATR, DMN, dom)
ATR={IdCarrito, CodProducto, correo}
DMN={int, varchar}
dom={IdCarrito->int, CodProducto->int, correo->varchar(40)}
RS-Vende=(PRS-Vende, M, SC)
M={“todas la interacciones entre el vendedor ya la compra de un producto”}
SC={PK->(dCarrito, CodProducto), NN->IdCarrito, NN->CodProducto, NN->correo, FK=IdCarrito->Carrito(IdCarrito), CodProducto->Producto(CodProducto), correo->vendedor(correo)}

3. RNE.

Estado_Producto={Nuevo, Usado}

Procedencia={Nacional, Internacional}

Lugar={Domicilio, Pickup)

4. Sentencias DDL para la creación de tablas y estructuras necesarias.

```
CREATE DATABASE bd_sive

CREATE TABLE Usuario (
        correo          varchar(40)   NOT NULL,
        Password        varchar(20)   NOT NULL,
        Nombre          varchar(20)   NOT NULL,
        Apellido        varchar(20)   NOT NULL,
        Fecha_Nac       date          NOT NULL,
        Documento       varchar(20)   NOT NULL,
        Calle           varchar(15)   NOT NULL,
        Numero          int           NOT NULL,
        Esquina         varchar(15)   NOT NULL,
        PRIMARY KEY(correo),
        UNIQUE(Documento)
        );
CREATE TABLE Telefono (
        correo varchar(40) NOT NULL,
        Telefono varchar(10) NOT NULL,
        PRIMARY KEY(correo, Telefono),
        FOREIGN KEY(correo) REFERENCES Usuario(correo)
        );

CREATE TABLE Cliente(
        correo          varchar(40)   NOT NULL,
        Reputacion      int       NOT NULL,
        PRIMARY KEY(correo),
        FOREIGN KEY(correo) REFERENCES Usuario(correo)
        );
```

```
CREATE TABLE vendedor(
        correo          varchar(40)   NOT NULL,
        IdVendedor      int           NOT NULL,
        PRIMARY KEY(correo),
        UNIQUE(IdVendedor),
        FOREIGN KEY(correo) REFERENCES Usuario(correo)
        );

CREATE TABLE Administrador(
        correo          varchar(40)   NOT NULL,
        IdAdministrador int           NOT NULL,
        PRIMARY KEY(correo),
        UNIQUE(IdAdministrador),
        FOREIGN KEY(correo) REFERENCES Usuario(correo)
        );

CREATE TABLE Catalogo(
        IdCatalogo      int           NOT NULL,
        Nombre          varchar(20)   NOT NULL,
        PRIMARY KEY(IdCatalogo)
        );

CREATE TABLE Categoria(
        IdCategoria     int           NOT NULL,
        Nombre          varchar(20)   NOT NULL,
        PRIMARY KEY(IdCategoria)
        );

CREATE TABLE Producto(
        CodProducto     int           NOT NULL,
        Nombre          varchar(20)   NOT NULL,
        Precio          int           NOT NULL,
        Stock           Int           NOT NULL,
        Estado_Producto varchar(10)   NOT NULL,
        Procedencia     varchar(15)   NOT NULL,
        Descripcion     varchar(70)   NOT NULL,
        Detalle         varchar(60)   NOT NULL,
        PRIMARY KEY(CodProducto)
        );
```

```
CREATE TABLE Compras(
        IdOrden             int             NOT NULL,
        Fecha_Compra        date            NOT NULL,
        pagoAprobado        boolean         NOT NULL,
        Total               int             NOT NULL,
        correo              varchar(40)     NOT NULL,
        CodProducto         int             NOT NULL,
        PRIMARY KEY(IdOrden),
        FOREIGN KEY(correo) REFERENCES Cliente(correo),
        FOREIGN KEY(CodProducto) REFERENCES Producto(CodProducto)
        );

CREATE TABLE MetodoDePago(
        IdMetodoPago        int             NOT NULL,
        Nombre              varchar(20)     NOT NULL,
        PRIMARY KEY(IdMetodoPago)
        );

CREATE TABLE Tarjeta(
        IdMetodoPago        int     NOT NULL,
        IdTarjeta           int     NOT NULL,
        PRIMARY KEY(IdMetodoPago),
        UNIQUE(IdTarjeta),
        FOREIGN KEY(IdMetodoPago) REFERENCES
        MetodoDePago(IdMetodoPago)
        );

CREATE TABLE PayPal(
        IdMetodoPago        int             NOT NULL,
        confirmacion        boolean         NOT NULL,
        PRIMARY KEY(IdMetodoPago),
        FOREIGN KEY(IdMetodoPago) REFERENCES
        MetodoDePago(IdMetodoPago)
        );
```

```
CREATE TABLE CentroDePagos(
        IdMetodoPago        int      NOT NULL,
        IdCentroDePago      int      NOT NULL,
        PRIMARY KEY(IdMetodoPago),
        UNIQUE(IdCentroDePago),
        FOREIGN KEY(IdMetodoPago) REFERENCES
        MetodoDePago(IdMetodoPago)
        );

CREATE TABLE Despacho(
        IdEntrega           int             NOT NULL,
        Estado_Entrega      varchar(20)     NOT NULL,
        Fecha_Entrega       date            NOT NULL,
        Lugar               varchar(10)     NOT NULL,
        PRIMARY KEY(IdEntrega)
        );

CREATE TABLE Esta(
        IdCatalogo          int      NOT NULL,
        CodProducto         int      NOT NULL,
        PRIMARY KEY(IdCatalogo, CodProducto),
        FOREIGN KEY(IdCatalogo) REFERENCES Catalogo(IdCatalogo),
        FOREIGN KEY(CodProducto) REFERENCES Producto(CodProducto)
        );

CREATE TABLE Administra(
        correo          varchar(40)     NOT NULL,
        IdCatalogo      int             NOT NULL,
        PRIMARY KEY(correo, IdCatalogo),
        FOREIGN KEY(correo) REFERENCES vendedor(correo),
        FOREIGN KEY(IdCatalogo) REFERENCES Catalogo(IdCatalogo)
        );

CREATE TABLE Gestiona(
        IdCategoria     int             NOT NULL,
        correo          varchar(40)     NOT NULL,
        PRIMARY KEY(IdCategoria),
        FOREIGN KEY(IdCategoria) REFERENCES Categoria(IdCategoria),
        FOREIGN KEY(correo) REFERENCES Administrador(correo)
        );
```

```
CREATE TABLE Tiene(
        CodProducto         int     NOT NULL,
        IdCategoria         int     NOT NULL,
        PRIMARY KEY(CodProducto),
        FOREIGN KEY(CodProducto) REFERENCES  Producto(CodProducto),
        FOREIGN KEY(IdCategoria) REFERENCES Categoria(IdCategoria)
        );

CREATE TABLE Surge(
        IdOrden     int     NOT NULL,
        IdEntrega   int     NOT NULL,
        PRIMARY KEY(IdOrden),
        FOREIGN KEY(IdOrden) REFERENCES Compras(IdOrden),
        FOREIGN KEY(IdEntrega) REFERENCES Despacho(IdEntrega)
        );

CREATE TABLE Utiliza(
        IdOrden             int     NOT NULL,
        IdMetodoPago        int     NOT NULL,
        PRIMARY KEY(IdOrden),
        FOREIGN KEY(IdOrden) REFERENCES Compras(IdOrden),
        FOREIGN KEY(IdMetodoPago) REFERENCES
        MetodoDePago(IdMetodoPago)
        );

CREATE TABLE Carrito(
        CodProducto         int             NOT NULL,
        correo              varchar(40)     NOT NULL,
        IdOrden             int             NOT NULL,
        IdCarrito           int             NOT NULL,
        Cantidad            int             NOT NULL,
        SubTotal            int             NOT NULL,
        PRIMARY KEY(CodProducto, IdCarrito),
        FOREIGN KEY(CodProducto) REFERENCES Producto(CodProducto),
        FOREIGN KEY(correo) REFERENCES Cliente(correo),
        FOREIGN KEY(IdOrden) REFERENCES Compras(IdOrden)
        );
```

```
CREATE TABLE Vende(
        IdCarrito       int         NOT NULL,
        CodProducto     Int         NOT NULL,
        correo          varchar(40) NOT NULL,
        PRIMARY KEY(IdCarrito, CodProducto),
        FOREIGN KEY(CodProducto) REFERENCES Producto(CodProducto),
        FOREIGN KEY(correo) REFERENCES vendedor(correo)
        );

        FOREIGN KEY(IdCarrito) REFERENCES Carrito(IdCarrito),
```

5. Estudio de los permisos considerando los diferentes roles.

La base de datos contará con los roles DBA, app y BackupAdmin

DBA

Permisos sobre BD	Permisos sobre tablas
TODO LOS PERMISOS	TODO LOS PERMISO

app

Permisos sobre BD	Permisos sobre tablas			
SHOW DATABASE	SELECT	UPDATE	INSERT	DELETE

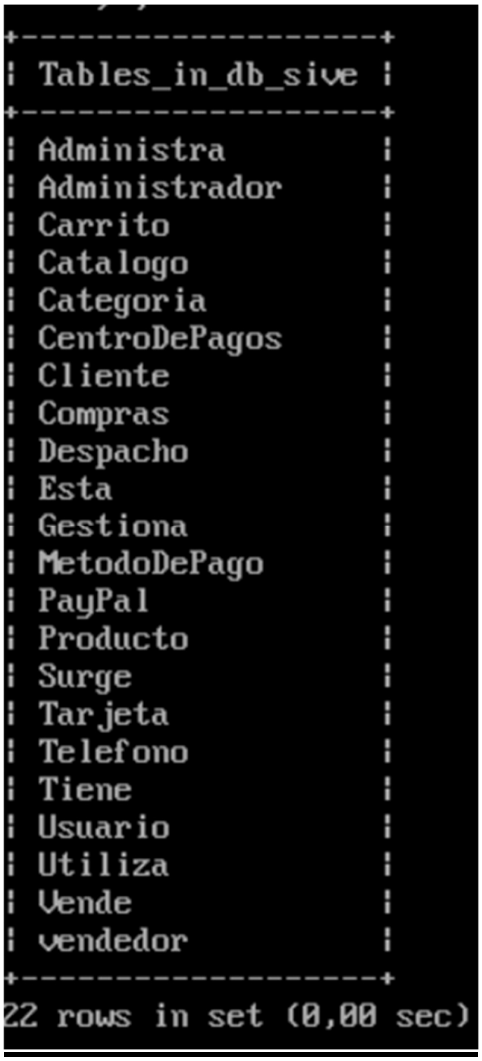
BackupAdmin

Permisos sobre BD	Permisos sobre tablas		
SHOW DATABASE	EVENT	LOCK TABLES	SELECT

6. Sentencias DCL de permisos a la Base de Datos.

```
CREATE ROLE IF NOT EXISTS DBA, app, BackupAdmin;

GRANT DBA TO  'dba'@'localhost'
GRANT app TO  'app'@'localhost'
GRANT BackupAdmin TO 'respaldo'@'localhost'



GRANT SHOW DATABASE ON db_sive TO app;
GRANT SELECT, UPDATE, INSERT, DELETE ON db_sive TO app;

GRANT SHOW DATABASE ON db_sive TO BackupAdmin;
GRANT EVENT, LOCK TABLES, SELECT ON db_sive TO
```

7. Primera implementación física de la Base de Datos en el servidor de la empresa.



```
+--------------------+
| Tables_in_db_sive  |
+--------------------+
| Administra         |
| Administrador      |
| Carrito            |
| Catalogo           |
| Categoria          |
| CentroDePagos      |
| Cliente            |
| Compras            |
| Despacho           |
| Esta               |
| Gestiona           |
| MetodoDePago       |
| PayPal             |
| Producto           |
| Surge              |
| Tarjeta            |
| Telefono           |
| Tiene              |
| Usuario            |
| Utiliza            |
| Vende              |
| vendedor           |
+--------------------+
22 rows in set (0,00 sec)
```

8. Gestión de usuarios y permisos

$ create user 'dba'@'localhost' identified by 'dba-sive.21';

$ create user 'app'@'localhost' identified by 'app-Sive.21';

$ create user 'respaldo'@'localhost' identified by 'respaldo-Sive.21

HOJA TESTIGO