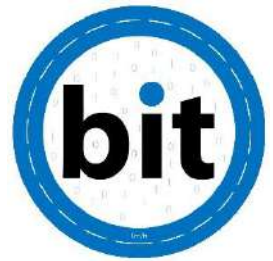


DDL

Tercera entrega 1/11/2019

Git lab /Actividades/BD01005/ddl



CREATE table

```
usuario(  
  IDUsuario serial primary key,  
  NombreDeUsuario varchar(20) not null unique,  
  Hash_Contra char(60) not null,  
  /* La contrasenia ser hasheada con bcrypt */  
  Email varchar(255) NOT null,  
  FechaNac date,  
  Telefono varchar(15) NOT null,  
  /* segn E.164 los nmeros telefonicos internacionales pueden ser de hasta  
  15 dgitos incluyendo el codigo pas */  
  PrimerNombre varchar(50) NOT null,  
  PrimerApellido varchar(50) NOT null,  
  PreguntaSecreta varchar(50),  
  RespuestaSecreta varchar(50),  
  Creador integer,  
  FechaCreacion date not null,  
  Sexo char(1) NOT null,  
  Rol char(1) not null,  
  Imagen byte default null,  
  invalido boolean default 'f' not null,  
  CHECK (Rol in ('A', 'O', 'T')),  
  CHECK (Sexo IN ('M',  
    'F',  
    'O')),  
  foreign key(Creador) references Usuario(IDUsuario)  
);
```

CREATE table

```
cliente(  
  RUT varchar(12) unique not null,  
  Nombre varchar(100) not null,  
  IDCliente serial primary key,  
  fechaRegistro datetime year to day not null,  
  invalido boolean not null,  
  UsuarioRegistro integer not null references Usuario(IDUsuario)  
);
```

```

CREATE table
    lugar(
        IDLugar serial primary key,
        Nombre varchar(100) not null,
        Capacidad INTEGER CHECK (Capacidad > 0), /*PUEDE SER NOT NULL POR LOS Establecimiento*/
    /
        GeoX FLOAT NOT null,
        GeoY FLOAT NOT null,
        UsuarioCreador integer NOT null references usuario(IDUsuario),
        fechaRegistro datetime year to second not null ,
        Tipo varchar(15) NOT null check (Tipo IN
            ("Patio","Puerto","Establecimiento", 'Zona', 'Subzona')),
        inhabilitado boolean default 'f' not null
    );

```

```

create table
    incluye(
        menor integer primary key,
        mayor integer,
        foreign key(menor) references lugar(idlugar),
        foreign key(mayor) references lugar(idlugar)
    );

```

```

create table
    perteneceA(
        IDLugar integer primary key references Lugar(IDLugar),
        ClienteID integer references Cliente(IDCliente)
    );

```

```

CREATE table
    trabajaen(
        ID serial primary key,
        IDLugar integer,
        IDUsuario integer not null,
        FechaInicio date NOT null,
        FechaFin date,
        foreign key(IDLugar) references lugar(IDLugar) ON DELETE CASCADE,
        foreign key(IDUsuario) references usuario(IDUsuario) ON DELETE CASCADE
    );

```

```

CREATE table
    conexion(
        IDTrabajaEn integer,
        HoraIngreso datetime year to second not null,
        HoraSalida datetime year to second,
        Usuario integer,
        primary key(Usuario,HoraIngreso),
        foreign key(Usuario) references usuario(IDUsuario) ON DELETE CASCADE,
        foreign key(IDTrabajaEn) references trabajaen(ID) ON DELETE CASCADE
    );

```

```

CREATE table
vehiculo(
  IDVehiculo serial primary key,
  VIN char(17) unique not null,
  Marca varchar(50),
  Modelo varchar(50),
  Color char(6), /* representación ineficiente; 6char = 6hex = 16^6 = 2^24 < 2^32 = int
(4char) < 6char */
  /* representacion del color por hexadecimal*/
  Tipo varchar(7) check(Tipo in ('Auto', 'MiniVan', 'SUV', 'Camion', 'Van')),
  Anio integer check(Anio >= 1900 and Anio <= 10000),
  Cliente Integer NOT null,
  foreign key(Cliente) references Cliente(IDCliente) ON DELETE CASCADE
);

```

```

create table
vehiculoIngresa(
  IDVehiculo integer,
  Fecha datetime year to second,
  TipoIngreso varchar(10) not null check (TipoIngreso in ('Precarga', 'Alta', 'Baja')),
  Usuario integer,
  Detalle bson,
  primary key(IDVehiculo, Usuario, Fecha),
  foreign key(IDVehiculo) references Vehiculo(IDVehiculo) ON DELETE CASCADE,
  foreign key(Usuario) references Usuario(IDUsuario) ON DELETE CASCADE
);

```

```

CREATE table
informedanios(
  ID serial,
  Descripcion varchar(255) NOT null,
  Fecha datetime year to day NOT null,
  Tipo varchar(7) NOT null check (Tipo in ('Total', 'Parcial')),
  IDVehiculo integer NOT null,
  idlugar integer NOT null,
  idusuario integer NOT null,
  primary key(ID, IDVehiculo),
  foreign key(idusuario) references usuario(IDUsuario) ON DELETE CASCADE,
  foreign key(IDVehiculo) references vehiculo(IDVehiculo) ON DELETE CASCADE,
  foreign key(idlugar) references lugar(idlugar) ON DELETE CASCADE
);

```

```

CREATE table
registrodanios(
  idvehiculo integer not null,
  informedanios integer not null,
  idregistro serial,
  descripcion varchar(255) not null,
  foreign key(informedanios, idvehiculo) references informedanios(ID, IDVehiculo) ON DEL
ETE CASCADE,
  primary key(idvehiculo, informedanios, idregistro)
);

```

```

CREATE table
    imagenregistro (
        vehiculo integer,
        informe integer,
        nrolista integer,
        nroimagen serial,
        imagen BYTE NOT null,
        primary key(vehiculo, informe, nrolista, nroimagen),
        foreign key(vehiculo, informe, nrolista) references registrodanios(idvehiculo, informe
danios, idregistro) ON DELETE CASCADE
    );

```

```

CREATE table
    actualiza (
        vehiculo1 integer,
        informe1 integer,
        registro1 integer,
        vehiculo2 integer,
        informe2 integer,
        registro2 integer,
        tipo varchar(15) NOT null check (tipo in ('Anulacion', 'Correccion')),
        CHECK (vehiculo1=vehiculo2),
        foreign key(vehiculo1, informe1, registro1) references registrodanios(idvehiculo, info
rmedanios, idregistro) ON DELETE CASCADE,
        foreign key(vehiculo2, informe2, registro2) references registrodanios(idvehiculo, info
rmedanios, idregistro) ON DELETE CASCADE,
        primary key(vehiculo1, informe1, registro1)
    );

```

```

CREATE table
    posicionado (
        IDLugar integer,
        IDVehiculo integer,
        desde datetime year to second,
        hasta datetime year to second,
        posicion integer NOT null check (posicion > 0),
        IDUsuario integer,
        foreign key(IDVehiculo) references vehiculo(IDVehiculo) ON DELETE CASCADE,
        foreign key(IDLugar) references lugar(IDLugar) ON DELETE CASCADE,
        foreign key(IDUsuario) references usuario(IDUsuario) ON DELETE CASCADE,
        primary key(IDLugar, IDVehiculo, desde)
    );

```

```

CREATE table
    TipoTransporte(
        IDTipo serial primary key,
        Nombre varchar(50) not null unique
    );

```

```

create table
    Habilitado(
        IDLugar integer,
        IDTipo integer,
        foreign key(IDLugar) references lugar(IDLugar),
        foreign key(IDTipo) references TipoTransporte(IDTipo),
        primary key(IDTipo, IDLugar)
    );

```

```

CREATE table
    MedioTransporte(
        IDTipo integer,
        IDLegal varchar(50), /* VIN/Matricula en caso de aplicar */
        Nombre varchar(50) NOT null,
        Tipo varchar(50) NOT null,
        Creador integer NOT null,
        FechaCreacion date not null,
        CantCamiones integer NOT null check (CantCamiones > -1),
        CantAutos integer NOT null check (CantAutos > -1),
        CantSUV integer NOT null check(CantSUV > -1),
        CantVan integer NOT null check(CantVan > -1),
        CantMinivan integer NOT null check (CantMinivan > -1),
        primary key(IDTipo, IDLegal),
        foreign key(Tipo) references TipoTransporte(Nombre) on delete cascade,
        foreign key(IDTipo) references TipoTransporte(IDTipo) ON DELETE CASCADE,
        foreign key(Creador) references usuario(IDUsuario) ON DELETE CASCADE
    );

```

```

CREATE table
    permite(
        IDTipo integer,
        IDLegal varchar(50),
        Usuario integer,
        invalido boolean,
        foreign key(IDTipo, IDLegal) references MedioTransporte(IDTipo, IDLegal) ON DELETE CAS
CADE,
        foreign key(Usuario) references usuario(IDUsuario) ON DELETE CASCADE,
        primary key(IDTipo, IDLegal, Usuario)
    );

```

CREATE table

```
lote(  
  IDLote serial,  
  nombre varchar(20),  
  Origen integer NOT null,  
  Destino integer NOT null,  
  CreadorID integer NOT null,  
  FechaCreacion datetime year to day not null,  
  Prioridad varchar(10) NOT null check (Prioridad in ('Normal', 'Alta')),  
  Estado varchar(10) not null check (Estado in ('Abierto', 'Cerrado')),  
  invalido boolean not null default 'f',  
  primary key(IDLote),  
  foreign key(Origen) references lugar(IDLugar) ON DELETE CASCADE,  
  foreign key(Destino) references lugar(IDLugar) ON DELETE CASCADE,  
  foreign key(CreadorID) references usuario(IDUsuario) ON DELETE CASCADE);
```

CREATE table

```
integra(  
  IDVehiculo integer,  
  Lote integer,  
  Fecha datetime year to minute,  
  invalidado boolean not null,  
  IDUsuario integer not null,  
  primary key(IDVehiculo, Lote, Fecha),  
  foreign key(IDVehiculo) references vehiculo(IDVehiculo) ON DELETE CASCADE,  
  foreign key(Lote) references lote(IDLote) ON DELETE CASCADE,  
  foreign key(IDUsuario) references usuario(IDUsuario) ON DELETE CASCADE );
```

CREATE table

```
transporte(  
  transporteID serial primary key,  
  Usuario integer NOT NULL,  
  IDTipo integer NOT NULL,  
  IDLegal varchar(50) NOT NULL,  
  FechaHoraCreacion datetime year to minute not null,  
  FechaHoraSalida datetime year to minute,  
  foreign key(IDTipo, IDLegal, Usuario) references Permite(IDTipo, IDLegal, Usuario) ON  
DELETE CASCADE  
);
```

```

CREATE table
transporta( transporteID integer,
IDLote integer,
Estado varchar(10) NOT null check (Estado in ("Proceso", "Fallo", "Exitoso","Cancelado
")),
FechaHorallegadaEstm datetime year to minute,
FechaHorallegadaReal datetime year to minute,
primary key(transporteID, IDLote),
foreign key(transporteID) references transporte(transporteID) ON DELETE CASCADE,
foreign key(IDLote) references lote(IDLote) ON DELETE CASCADE );

```

```

create table
link(
link varchar(255) not null,
transportista integer primary key,
foreign key(transportista) references usuario(idusuario) ON DELETE CASCADE
);

```

```

create table
evento(
id serial primary key,
datos bson not null,
fechaAgregado datetime year to second
);

```