CREATE TABLE tipotweet

(

idtipotweet INT NOT NULL,

tipotweet VARCHAR2 (50) NOT NULL,

CONSTRAINT pktipotweet PRIMARY KEY(idtipotweet)

);

CREATE SEQUENCE tipotweetsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER tipotweetincrement BEFORE

INSERT

ON tipotweet

FOR EACH ROW

BEGIN :new.idtipotweet:=tipotweetsequence.NEXTVAL;

END;

-----------------------------------------------------------------------------------------------------------------

CREATE TABLE tweet

(

idtweet INT NOT NULL,

cuerpo VARCHAR2 (255) NOT NULL,

idtipotweet INT NOT NULL,

FOREIGN KEY (idtipotweet) REFERENCES tipotweet(idtipotweet),

CONSTRAINT pktweet PRIMARY KEY(idtweet)

);

CREATE SEQUENCE tweetsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER tweetincrement BEFORE

INSERT

ON tweet

FOR EACH ROW

BEGIN :new.idtweet:=tweetsequence.NEXTVAL;

END;

------------------------------------------------------------------------------------------------------------------------

CREATE TABLE cuentatwitter

(

idcuentatwitter INT NOT NULL,

consumerkey VARCHAR2 (100) NOT NULL,

consumersecret VARCHAR2 (100) NOT NULL,

accesstoken VARCHAR2 (100) NOT NULL,

accesstokensecret VARCHAR2 (100) NOT NULL,

iduniversidad INT NOT NULL,

CONSTRAINT uniqueuniversidadT UNIQUE(iduniversidad),

FOREIGN KEY (iduniversidad) REFERENCES universidad(iduniversidad),

CONSTRAINT pkcuentatwitter PRIMARY KEY(idcuentatwitter)

);

CREATE SEQUENCE cuentatwittersequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER cuentatwitterincrement BEFORE

INSERT

ON cuentatwitter

FOR EACH ROW

BEGIN :new.idcuentatwitter:=cuentatwittersequence.NEXTVAL;

END;

--------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE universidad

(

iduniversidad INT NOT NULL,

nombre VARCHAR2(50) NOT NULL,

meta NUMBER(38,2),

CONSTRAINT pkuniversidad PRIMARY KEY(iduniversidad)

);

CREATE SEQUENCE universidadsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER universidadincrement BEFORE

INSERT

ON universidad

FOR EACH ROW

BEGIN :new.iduniversidad:=universidadsequence.NEXTVAL;

END;

------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE sede

(

idsede INT NOT NULL,

nombre VARCHAR2(50) NOT NULL,

ubicacion VARCHAR2(100) NOT NULL,

iduniversidad INT NOT NULL,

FOREIGN KEY (iduniversidad) REFERENCES universidad(iduniversidad),

CONSTRAINT pksede PRIMARY KEY(idsede)

);

CREATE SEQUENCE sedesequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER sedeincrement BEFORE

INSERT

ON sede

FOR EACH ROW

BEGIN :new.idsede:=sedesequence.NEXTVAL;

END;

--------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE centroacopio

(

idcentroacopio INT NOT NULL,

idsede INT NOT NULL,

nombre VARCHAR2(50) NOT NULL,

FOREIGN KEY (idsede) REFERENCES sede(idsede),

CONSTRAINT pkcentroacopio PRIMARY KEY(idcentroacopio)

);

CREATE SEQUENCE centroacopiosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER centroacopioincrement BEFORE

INSERT

ON centroacopio

FOR EACH ROW

BEGIN :new.idcentroacopio:=centroacopiosequence.NEXTVAL;

END;

-----------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE estado

(

idestado INT NOT NULL,

estado VARCHAR2(50) NOT NULL,

CONSTRAINT pkestado PRIMARY KEY(idestado)

);

CREATE SEQUENCE estadosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER estadoincrement BEFORE

INSERT

ON estado

FOR EACH ROW

BEGIN :new.idestado:=estadosequence.NEXTVAL;

END;

-------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE material

(

idmaterial INT NOT NULL,

nombre VARCHAR2(50) NOT NULL,

valor NUMBER(38,2) NOT NULL,

CONSTRAINT pkmaterial PRIMARY KEY(idmaterial)

);

CREATE SEQUENCE materialsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER materialincrement BEFORE

INSERT

ON material

FOR EACH ROW

BEGIN :new.idmaterial:=materialsequence.NEXTVAL;

END;

--------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE tipocambio

(

idtipocambio INT NOT NULL,

tipocambio NUMBER(38,2) NOT NULL,

CONSTRAINT pktipocambio PRIMARY KEY(idtipocambio)

);

CREATE SEQUENCE tipocambiosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER tipocambioincrement BEFORE

INSERT

ON tipocambio

FOR EACH ROW

BEGIN :new.idtipocambio:=tipocambiosequence.NEXTVAL;

END;

----------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE servicio

(

idservicio INT NOT NULL,

nombre VARCHAR2(50) NOT NULL,

CONSTRAINT pkservicio PRIMARY KEY(idservicio)

);

CREATE SEQUENCE serviciosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER servicioincrement BEFORE

INSERT

ON servicio

FOR EACH

ROW BEGIN :new.idservicio:=serviciosequence.NEXTVAL;

END;

------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE departamento(

iddepartamento INT NOT NULL,

departamento VARCHAR2(50) NOT NULL,

iduniversidad INT NOT NULL,

FOREIGN KEY (iduniversidad) REFERENCES universidad(iduniversidad),

CONSTRAINT pkdepartamento PRIMARY KEY (iddepartamento)

);

------------------------------------------

ALTER TABLE departamento

ADD iduniversidad INT NOT NULL;

ALTER TABLE departamento

ADD FOREIGN KEY (iduniversidad) REFERENCES universidad(iduniversidad);

-------------------

CREATE SEQUENCE departamentosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER departamentoincrement BEFORE

INSERT

ON departamento

FOR EACH ROW

BEGIN :new.iddepartamento:=departamentosequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE usuario

(

idusuario INT NOT NULL,

nombre VARCHAR2 (50) NOT NULL,

apellido1 VARCHAR2 (50) NOT NULL,

apellido2 VARCHAR2 (50) NOT NULL,

contrasena VARCHAR2 (16) NOT NULL,

CONSTRAINT pkusuario PRIMARY KEY (idusuario)

);

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE estudiante

(

idestudiante INT NOT NULL,

descripcion VARCHAR2 (500),

fotografia VARCHAR2 (999),

fechaingreso DATE NOT NULL,

cantidadtecolones NUMBER(38,2) NOT NULL,

correo VARCHAR2(50) NOT NULL,

telefono VARCHAR2(8) NOT NULL,

idestado INT NOT NULL,

idusuario INT NOT NULL,

idsede INT NOT NULL,

CONSTRAINT uniquecorreo UNIQUE(correo),

CONSTRAINT uniqueusuario UNIQUE(idusuario),

CONSTRAINT uniquetelefono UNIQUE(telefono),

FOREIGN KEY (idestado) REFERENCES estado(idestado),

FOREIGN KEY (idusuario) REFERENCES usuario(idusuario),

FOREIGN KEY (idsede) REFERENCES sede(idsede),

CONSTRAINT pkestudiante PRIMARY KEY (idestudiante)

);

CREATE SEQUENCE estudiantesequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER estudianteincrement BEFORE

INSERT

ON estudiante

FOR EACH ROW

BEGIN :new.idestudiante:=estudiantesequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE correo

(

idcorreo INT NOT NULL,

idusuario INT NOT NULL,

correoelectronico VARCHAR2(500),

CONSTRAINT uniquecorreo1 UNIQUE(correoelectronico),

CONSTRAINT pkcorreo1 PRIMARY KEY (idcorreo),

FOREIGN KEY (idusuario) REFERENCES usuario(idusuario)

);

CREATE SEQUENCE correosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER correoincrement BEFORE

INSERT

ON correo

FOR EACH ROW

BEGIN :new.idcorreo:=correosequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE telefono

(

idtelefono INT NOT NULL,

telefono VARCHAR2(500),

idusuario INT NOT NULL,

CONSTRAINT pktelefono PRIMARY KEY (idtelefono),

CONSTRAINT uniquetelefono1 UNIQUE(telefono),

FOREIGN KEY (idusuario) REFERENCES usuario (idusuario)

);

CREATE SEQUENCE telefonosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER telefonoincrement BEFORE

INSERT

ON telefono

FOR EACH ROW

BEGIN :new.idtelefono:=telefonosequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE administrador

(

idadministrador INT NOT NULL,

idusuario INT NOT NULL,

iddepartamento INT NOT NULL,

CONSTRAINT uniqueusuario1 UNIQUE(idusuario),

CONSTRAINT pkadministrador PRIMARY KEY (idadministrador),

FOREIGN KEY (idusuario) REFERENCES usuario (idusuario),

FOREIGN KEY (iddepartamento) REFERENCES departamento(iddepartamento)

);

CREATE SEQUENCE administradorsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER administradorincrement BEFORE

INSERT

ON administrador

FOR EACH ROW

BEGIN :new.idadministrador:=administradorsequence.NEXTVAL;

END;

--------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE administradoracopio

(

idadministradoracopio INT NOT NULL,

idusuario INT NOT NULL,

idcentroacopio INT NOT NULL,

CONSTRAINT uniqueusuario2 UNIQUE(idusuario),

CONSTRAINT uniqueacopio UNIQUE(idcentroacopio),

CONSTRAINT pkadministradoracopio PRIMARY KEY (idadministradoracopio),

FOREIGN KEY (idusuario) REFERENCES usuario (idusuario),

FOREIGN KEY (idcentroacopio) REFERENCES centroacopio (idcentroacopio)

);

CREATE SEQUENCE administradoracopiosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER administradoracopioincrement BEFORE

INSERT

ON administradoracopio

FOR EACH ROW

BEGIN :new.idadministradoracopio:=administradoracopiosequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE facturaservicio

(

idfacturaservicio INT NOT NULL,

idusuario INT NOT NULL,

idservicio INT NOT NULL,

cantidadcolones NUMBER(38,2) NOT NULL,

fecha DATE NOT NULL,

CONSTRAINT pkfacturaservicio PRIMARY KEY (idfacturaservicio),

FOREIGN KEY (idusuario) REFERENCES usuario (idusuario),

FOREIGN KEY (idservicio) REFERENCES servicio (idservicio)

);

CREATE SEQUENCE facturaserviciosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER facturaservicioincrement BEFORE

INSERT

ON facturaservicio

FOR EACH ROW

BEGIN :new.idfacturaservicio:=facturaserviciosequence.NEXTVAL;

END;

----------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE TABLE estudiantematerial

(

idestudiantematerial INT NOT NULL,

idusuario INT NOT NULL,

idsede INT NOT NULL,

idmaterial INT NOT NULL,

cantidad NUMBER(38,2) NOT NULL,

fecharegistro DATE NOT NULL,

CONSTRAINT pkestudiantematerial PRIMARY KEY (idestudiantematerial),

FOREIGN KEY (idusuario) REFERENCES usuario (idusuario),

FOREIGN KEY (idsede) REFERENCES sede(idsede),

FOREIGN KEY (idmaterial) REFERENCES material (idmaterial)

);

CREATE SEQUENCE estudiantematerialsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER estudiantematerialincrement BEFORE

INSERT

ON estudiantematerial

FOR EACH ROW

BEGIN :new.idestudiantematerial:=estudiantematerialsequence.NEXTVAL;

END;

-----------------------------------------------------------**NUEVO**-----------------------------------------

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE promocion

(

idpromocion INT NOT NULL,

iduniversidad INT NOT NULL,

idmaterial1 INT NOT NULL,

idmaterial2 INT,

cantidad1 NUMBER(38,2) NOT NULL,

cantidad2 NUMBER(38,2),

bono NUMBER(38,2) NOT NULL,

fechainicio DATE NOT NULL,

fechafinal DATE NOT NULL,

tipo VARCHAR2(10) NOT NULL,

FOREIGN KEY (iduniversidad) REFERENCES universidad(iduniversidad),

FOREIGN KEY (idmaterial1) REFERENCES material(idmaterial),

FOREIGN KEY (idmaterial2) REFERENCES material(idmaterial),

CONSTRAINT pkpromocion PRIMARY KEY(idpromocion)

);

CREATE SEQUENCE promocionsequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER promocionincrement BEFORE

INSERT

ON promocion

FOR EACH ROW

BEGIN :new.idpromocion:=promocionsequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------

CREATE TABLE bono

(

idbono INT NOT NULL,

idusuario INT NOT NULL,

idpromocion INT NOT NULL,

FOREIGN KEY (idusuario) REFERENCES usuario(idusuario),

FOREIGN KEY (idpromocion) REFERENCES promocion(idpromocion),

CONSTRAINT pkbono PRIMARY KEY(idbono)

);

CREATE SEQUENCE bonosequence START WITH 1 INCREMENT BY 1;

CREATE

OR

replace TRIGGER bonoincrement BEFORE

INSERT

ON bono

FOR EACH ROW

BEGIN :new.idbono:=bonosequence.NEXTVAL;

END;

---------------------------------------------------------------------------------------------------------------------------