**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**

**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**

**DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN, CONTROL Y EVALUACIÓN DE RECURSOS INFORMÁTICOS**

**INGENIERÍA DE SISTEMAS DE INFORMACIÓN**

**“IMPLEMENTACIÓN DE UNA BASE DE DATOS PARA EL PROCESO DE DISTRIBUCIÓN DE DONACIONES A HOSPITALES*”***

**ASIGNATURA:**

**SISTEMAS DE BASE DE DATOS II**

**ELABORADO POR: CAMPOS, MILAGROS 8-948-227**

**CASTILLO GUADALUPE 8-929-2252**

**ROSAS, ELIONAYS 9-756-2182**

**URRIOLA, ALEJANDRO 9-755-1141**

**FACILITADOR: ING. HENRY LEZCANO**

**II SEMESTRE, 2020**

# 

Contenido

[Introducción 3](#_Toc57589454)

[Misión de la Base de Datos 4](#_Toc57589455)

[Objetivo de la misión de la Base de Datos 4](#_Toc57589456)

[Definición del proyecto 5](#_Toc57589457)

[Descripción de los datos que han sido identificados y utilización 6](#_Toc57589458)

[Modelado de la Base de Datos 7](#_Toc57589459)

[MODELO LOGICO NORMALIZADO 8](#_Toc57589460)

[**MODELO FÍSICO EN ORACLE** 10](#_Toc57589461)

[CREACION DE TABLAS 10](#_Toc57589462)

[PROCEDIMIENTOS UTILIZADOS 23](#_Toc57589463)

[IMPLEMENTACION DE TRIGGERS 45](#_Toc57589464)

[CONCLUSIÓN 49](#_Toc57589465)

# Introducción

En 2020, Panamá fue afectada terriblemente por el brote de covid-2019 este ha hecho que la organización y sistema de salud en general colapse, generando un desabastecimiento de los insumos médicos en los diferentes centros de salud y hospitales, por ello surge la necesidad de buscar un mecanismo mediante el cual optimizar los procesos de distribución de los insumos médicos, y así evitar el desabastecimiento de las instalaciones médicas. Para lograr esto hemos planteado la creación de una base de datos, de la cual mas adelante iremos viendo cada uno de sus aspectos mas importantes, desde su modelo lógico, modelo relacional y la implementación de algunos datos de prueba.

Nombre del proyecto de Base de Datos

**IMPLEMENTACIÓN DE UNA BASE DE DATOS PARA EL PROCESO DE DISTRIBUCIÓN DE DONACIONES A HOSPITALES**

# Misión de la Base de Datos

Lograr que las donaciones que realiza el pueblo panameño mediante el Plan Panamá solidario se distribuyan equitativamente a todos los centros médicos del país que se encuentran atendiendo a pacientes positivos de covid-19.

# Objetivo de la misión de la Base de Datos

* Mantener el control de las donaciones de insumos médicos llevadas a los distintos centros de distribución para evitar la mala disposición de estas.
* Proveer información de los diversos tipos de transporte utilizados para gestión de la distribución de los insumos médicos a las distintas instalaciones médicas.
* Sostener un adecuado manejo del personal que trabaja en los distintos centros de distribución.

# Definición del proyecto

* Ámbito

Actualmente el plan Panamá solidario en la sección de donaciones de salud se maneja con una base de datos que recopila información del donante y de la donación, lo que incluye todos los datos que describen el insumo, que llegan a los centros de distribución que se encuentran ubicados en distintas regiones del país. Destacamos que el sistema actual permite saber las necesidades urgentes que tiene el sector salud en cuanto a los insumos médicos, sin embargo, presenta puntos de mejora en el control de las donaciones, por lo que esta base de datos pretende solucionar esta problemática de manera efectiva proporcionando así agilidad en todos los procesos que sean necesarios para el desenvolvimiento óptimo de la distribución de insumos.

* Alcance o limites
* Nuestra base de datos debe ser capaz de almacenar la información del donante, insumos médicos (medicinas y equipo de protección), los distintos centros de distribución que se encuentran en diversas regiones del país, el personal que este posee (empacador y repartidor) y el transporte utilizado para la distribución.
* Debe permitir ingresar, eliminar y actualizar la información de todos los insumos médicos recibidos en los diversos centros de distribución.
* Análisis de requerimientos

Para la recopilación de datos de este proyecto decidimos utilizar la técnica de Revisión de documentos obtenidos del sitio web oficial del programa Panamá Solidario, de esta manera aseguramos la comprensión del surgimiento de la necesidad del nuevo sistema haciendo uso minucioso de cada dato almacenado que guardaba relación con el problema que se debía resolver adicional a eso realizamos algunas consultas a manera de nutrir al equipo de información importante sobre la situación actual en Panamá y el mundo por lo promovemos la obtención de información desde una vista mucho más amplia que la que se tenía actualmente asegurando la implementación de los mejores métodos o métodos más funcionales en nuestro sistema de base de datos

# Descripción de los datos que han sido identificados y utilización

Al tratarse de un proyecto basado un poco en el tema logístico los datos tienden a ser genéricos, pero a más detalle se puede tener una mejor toma de decisiones por lo que cada tabla cuenta con datos indispensables.

En la tabla Donador, manejas los datos personales de cada persona o empresa que hace una donación, se hace indispensable su dirección para de esta manera conocer los tiempos de llegada de las donaciones o de donde provienen.

En la tabla Centro de Distribución, manejamos todos los datos de ubicación entre ellos la región en que está ubicada de esta manera los usuarios de la base de datos pueden saber todas las áreas que pueden ser atendidas por este de igual manera para la tabla instalaciones médicas donde se detalla la región donde está ubicada.

Sabemos también del personal que labora en los centros de distribución su fecha de nacimiento y su dirección, para asegurar que no se encuentre en un rango de edad en alto riesgo y donde puede ser enviado como colaborador respectivamente relacionado al riesgo, pero en otro contexto tenemos también la tabla transporte donde se maneja la capacidad de carga de cada vehículo asegurando el uso óptimo de cada uno.

# Modelado de la Base de Datos

* Modelo conceptual E/R

Diagrama

Descripción generada automáticamente

Ilustración 1 Modelo conceptual E/R

# MODELO LOGICO NORMALIZADO

|  |  |  |
| --- | --- | --- |
| Centro\_distribucion | | |
| PK | codigo\_CEDIS | not null |
|  | nombre | not null |
|  | region | not null |
|  | provincia | not null |
|  | ubicación | not null |

|  |  |  |
| --- | --- | --- |
| Personal | | |
| PK | DNI | not null |
|  | nombre | not null |
|  | apellido | not null |
|  | fecha\_nac | not null |
|  | ciudad | not null |
|  | corregimiento | not null |
|  | calle |  |
|  | num\_casa | not null |
| FK | cod\_cargo | not null |

|  |  |  |
| --- | --- | --- |
| Donador | | |
| PK | DNI\_RUC | not null |
|  | nombre | not null |
|  | apellido | not null |
|  | provincia | not null |
|  | distrito |  |
|  | corregimiento |  |
|  | calle |  |
|  | num\_casa |  |

|  |  |  |
| --- | --- | --- |
| tipo\_telefono | | |
| PK | cod\_telf | not null |
|  | desc\_telf |  |

|  |  |  |
| --- | --- | --- |
| tipo\_email | | |
| PK | cod\_email | not null |
|  | desc\_email |  |

|  |  |  |
| --- | --- | --- |
| telefono\_personal | | |
| PK/FK | DNI | not null |
| PK/FK | cod\_telf | not null |
|  | telefono |  |

|  |  |  |
| --- | --- | --- |
| tipo\_insumo | | |
| PK | cod\_insumo | not null |
|  | desc\_insumo |  |

|  |  |  |
| --- | --- | --- |
| Insumo | | |
| PK | cod\_donacion | not null |
| FK | cod\_insumo | not null |

|  |  |  |
| --- | --- | --- |
| email\_donador | | |
| PK/FK | DNI\_RUC | not null |
| PK/FK | cod\_email | not null |
|  | email |  |

|  |  |  |
| --- | --- | --- |
| telefono\_donador | | |
| PK/FK | DNI\_RUC | not null |
| PK/FK | cod\_telf | not null |
|  | telefono |  |

|  |  |  |
| --- | --- | --- |
| Instalaciones\_medicas | | |
| PK | codigo\_IM | not null |
|  | nombre | not null |
|  | region | not null |
|  | provincia | not null |
|  | distrito | not null |
|  | corregimiento | not null |
|  | ubicación | not null |

|  |  |  |
| --- | --- | --- |
| tipo\_transporte | | |
| PK | cod\_trans | not null |
|  | desc\_trans |  |

|  |  |  |
| --- | --- | --- |
| telefono\_IM | | |
| PK/FK | codigo\_IM | not null |
| PK/FK | cod\_telf | not null |
|  | telefono |  |

|  |  |  |
| --- | --- | --- |
| Repartidor | | |
| PK | licencia | not null |
| FK | DNI | not null |
| FK | matricula | not null |

|  |  |  |
| --- | --- | --- |
| Transporte | | |
| PK | matricula | not null |
|  | cap\_carga | not null |
| FK | cod\_trans | not null |

|  |  |  |
| --- | --- | --- |
| Insumo\_personal\_IM | | |
| PK/FK | cod\_insumo | not null |
| PK/FK | DNI | not null |
| PK/FK | codigo\_IM | not null |
|  | fecha\_dist |  |
|  | cantidad\_llevada |  |

|  |  |  |
| --- | --- | --- |
| Donador\_insumo | | |
| PK/FK | DNI\_RUC | not null |
| PK/FK | cod\_donacion | not null |
|  | Cantidad | Not null |

|  |  |  |
| --- | --- | --- |
| Personal\_CEDIS | | |
| PK/FK | DNI | not null |
| PK/FK | codigo\_CEDIS | not null |

|  |  |  |
| --- | --- | --- |
| tipo\_personal | | |
| PK | cod\_cargo | not null |
|  | descripcion |  |

|  |  |  |
| --- | --- | --- |
| Donador\_CEDIS | | |
| PK/FK | DNI\_RUC | not null |
| PK/FK | codigo\_CEDIS | not null |

# **MODELO FÍSICO EN ORACLE**

# CREACION DE TABLAS

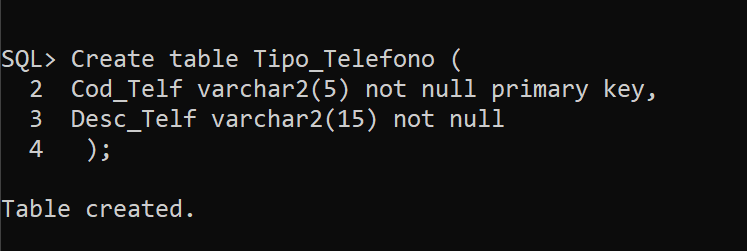
**--**Creacion de la tabla TIPO\_TELEFONO

Create table Tipo\_Telefono (

Cod\_Telf varchar2(5) not null primary key,

Desc\_Telf varchar2(15) not null

 );



**-- Creación de la tabla llamada DONADOR**

Create table DONADOR (

 DNI\_RUC  varchar(20) not null primary key,

 Nombre  varchar (10) not null,

 Apellido varchar (10) not null,

 provincia varchar(20) not null,

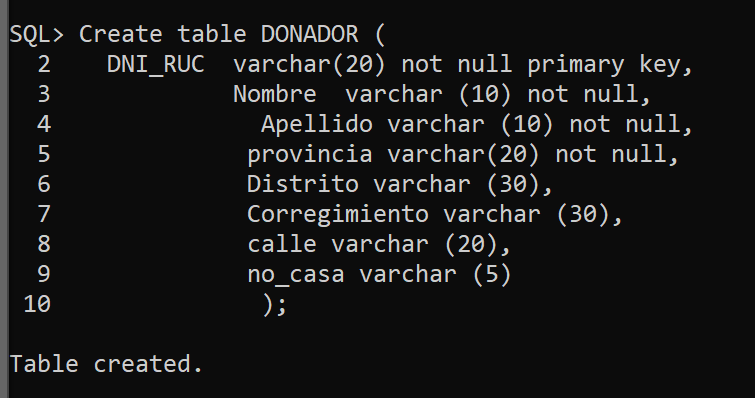
  Distrito varchar (30),

  Corregimiento varchar (30),

  calle varchar (20),

   no\_casa varchar (5)

     );



**--Creacion de la tabla TELEFONO\_DONADOR**

Create table Telefono\_donador (

DNI\_RUC varchar2(20) not null,

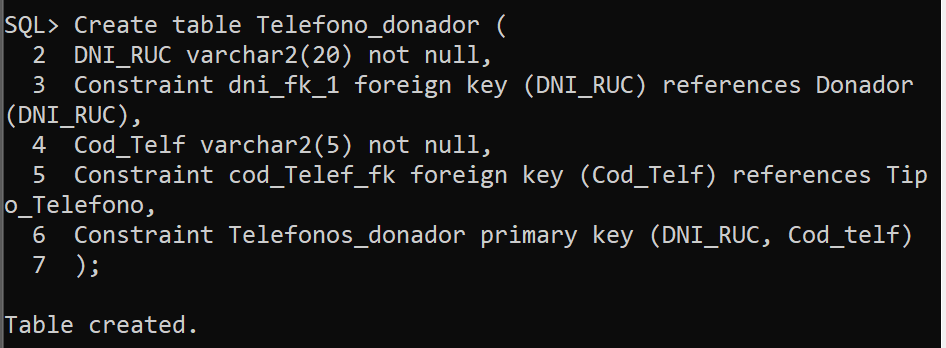
Constraint dni\_fk\_1 foreign key (DNI\_RUC) references Donador (DNI\_RUC),

Cod\_Telf varchar2(5) not null,

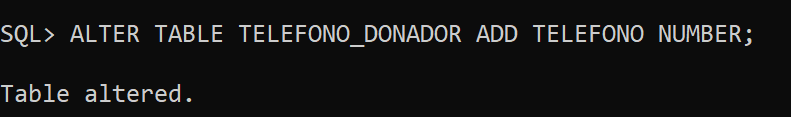
Constraint cod\_Telef\_fk foreign key (Cod\_Telf) references Tipo\_Telefono,

Constraint Telefonos\_donador primary key (DNI\_RUC, Cod\_telf)

);



ALTER TABLE TELEFONO\_DONADOR ADD TELEFONO NUMBER;



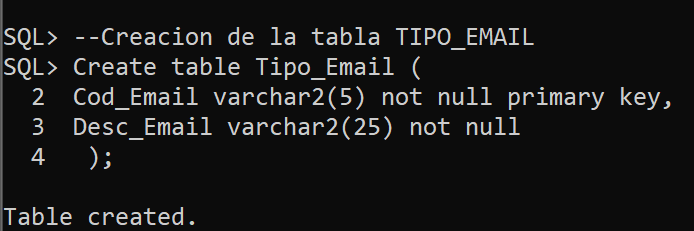
**--Creacion de la tabla TIPO\_EMAIL**

Create table Tipo\_Email (

Cod\_Email varchar2(5) not null primary key,

Desc\_Email varchar2(25) not null

 );



**--Creacion de la tabla EMAIL\_donador**

Create table Email\_donador (

DNI\_RUC varchar2(20),

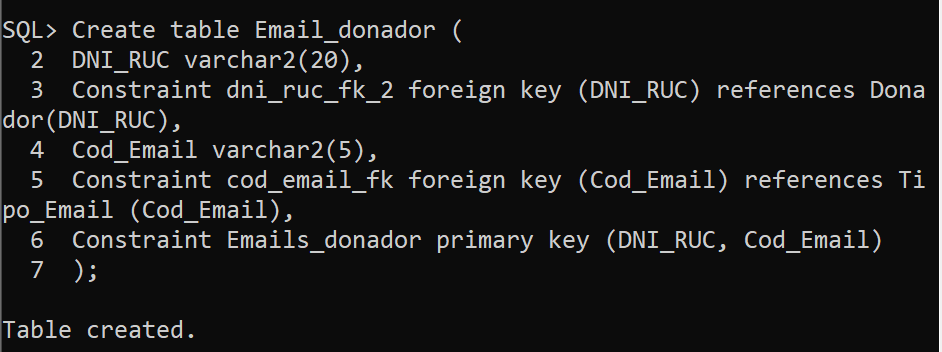
Constraint dni\_ruc\_fk\_2 foreign key (DNI\_RUC) references Donador(DNI\_RUC),

Cod\_Email varchar2(5),

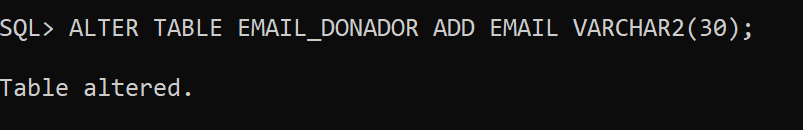
Constraint cod\_email\_fk foreign key (Cod\_Email) references Tipo\_Email (Cod\_Email),

Constraint Emails\_donador primary key (DNI\_RUC, Cod\_Email)

);



ALTER TABLE EMAIL\_DONADOR ADD EMAIL VARCHAR2(30);



**--Creacion de la tabla Centro distribución**

Create table Centro\_distribucion(

Codigo\_cedis varchar2(5) not null primary key,

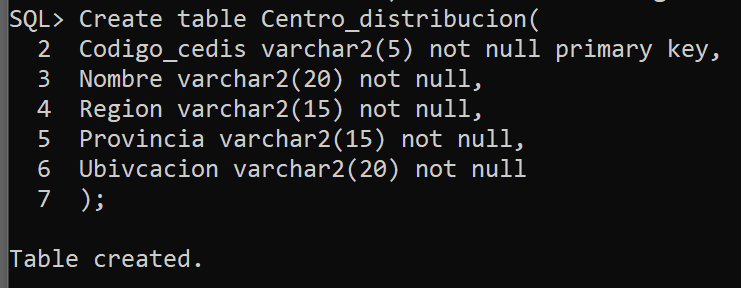
Nombre varchar2(20) not null,

Region varchar2(15) not null,

Provincia varchar2(15) not null,

Ubicacion varchar2(20) not null

);



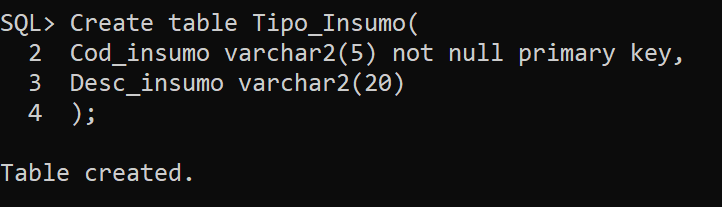
**--Creacion de la tabla Tipo insumo**

Create table Tipo\_Insumo(

Cod\_insumo varchar2(5) not null primary key,

Desc\_insumo varchar2(20)

);



**--Creacion de la tabla Insumo**

Create table Insumo(

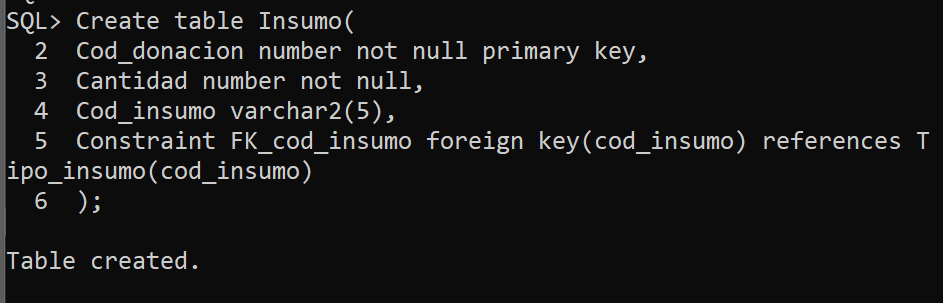
Cod\_donacion number not null primary key,

Cantidad number not null,

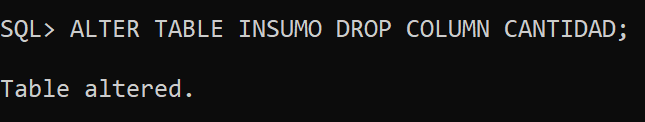
Cod\_insumo varchar2(5),

Constraint FK\_cod\_insumo foreign key(cod\_insumo) references Tipo\_insumo(cod\_insumo)

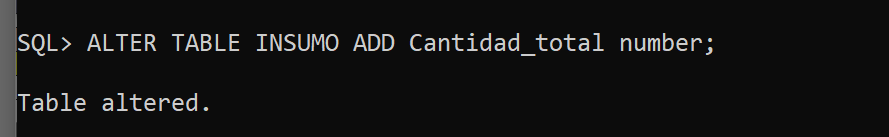
);



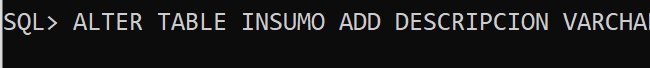
ALTER TABLE INSUMO DROP COLUMN CANTIDAD;



ALTER TABLE INSUMO ADD Cantidad\_total number;



ALTER TABLE INSUMO ADD DESCRIPCION VARCHAR2(20);



**--Creacion de la tabla Instalaciones Medicas**

Create table Instalaciones\_medicas(

Codigo\_IM varchar2(5) not null primary key,

Nombre varchar2(20) not null,

Region varchar2(15) not null,

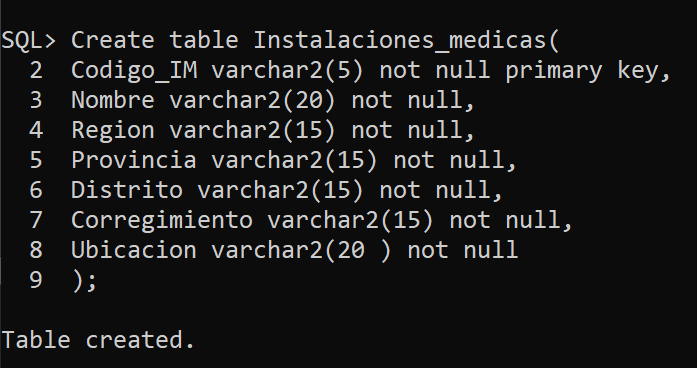
Provincia varchar2(15) not null,

Distrito varchar2(15) not null,

Corregimiento varchar2(15) not null,

Ubicacion varchar2(20 ) not null

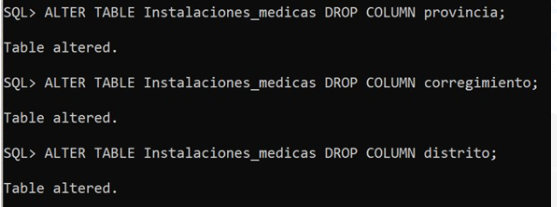
);



ALTER TABLE Instalaciones\_medicas DROP COLUMN provincia;

ALTER TABLE Instalaciones\_medicas DROP COLUMN corregimiento;

ALTER TABLE Instalaciones\_medicas DROP COLUMN distrito;



**--Creacion de la tabla teléfono de Instalación medica**

Create table telefono\_im (

Codigo\_IM varchar2(5) not null,

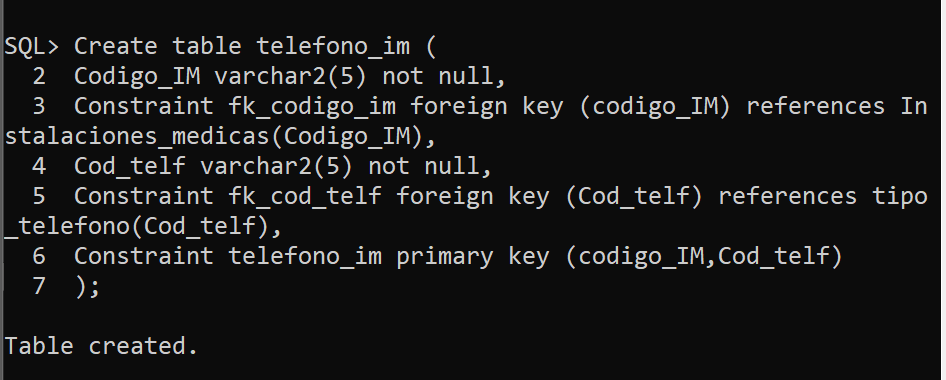
Constraint fk\_codigo\_im foreign key (codigo\_IM) references Instalaciones\_medicas(Codigo\_IM),

Cod\_telf varchar2(5) not null,

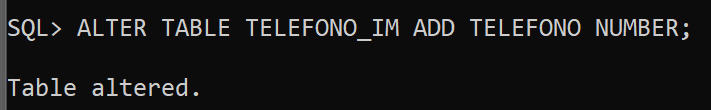
Constraint fk\_cod\_telf foreign key (Cod\_telf) references tipo\_telefono(Cod\_telf),

Constraint telefono\_im primary key (codigo\_IM,Cod\_telf)

);



ALTER TABLE TELEFONO\_IM ADD TELEFONO NUMBER;



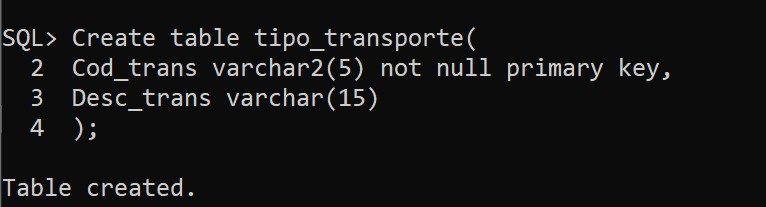
**--Creacion de la tabla Tipo transporte**

Create table tipo\_transporte(

Cod\_trans varchar2(5) not null primary key,

Desc\_trans varchar(15)

);



**--Creacion de la tabla transporte**

Create table transporte(

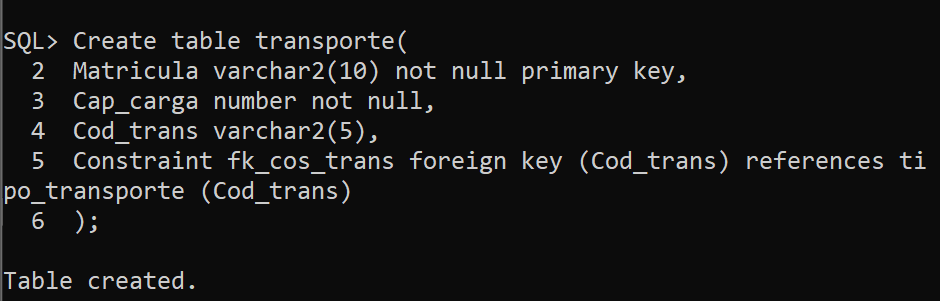
Matricula varchar2(10) not null primary key,

Cap\_carga number not null,

Cod\_trans varchar2(5),

Constraint fk\_cos\_trans foreign key (Cod\_trans) references tipo\_transporte (Cod\_trans)

);



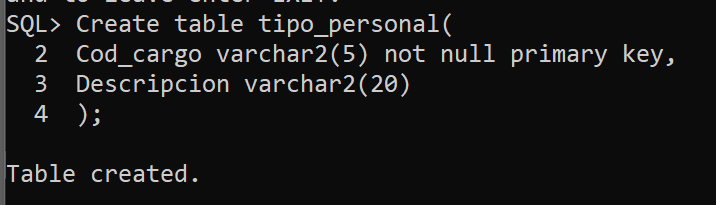
**--Creación de tabla Tipo personal**

Create table tipo\_personal(

Cod\_cargo varchar2(5) not null primary key,

Descripcion varchar2(20)

);



-- **Creación de tabla personal**

Create table Personal(

DNI varchar2(15) not null primary key,

Nombre varchar2(15) not null,

Apellido varchar2(15) not null,

Fecha\_nac date not null,

Ciudad varchar2(15 ) not null,

Corregimiento varchar2(15) not null,

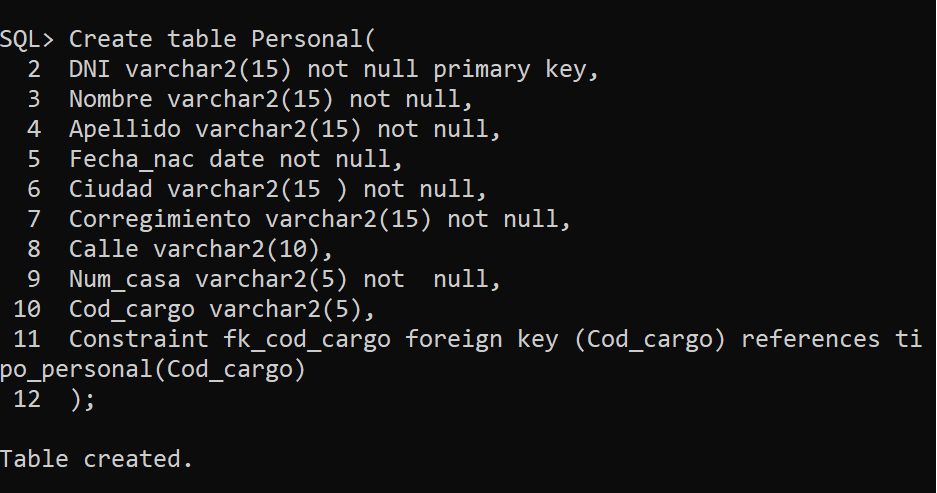
Calle varchar2(10),

Num\_casa varchar2(5) not  null,

Cod\_cargo varchar2(5),

Constraint fk\_cod\_cargo foreign key (Cod\_cargo) references tipo\_personal(Cod\_cargo)

);



-- **Creación de tabla teléfono personal**

Create table telefono\_personal(

Dni varchar2(15) ,

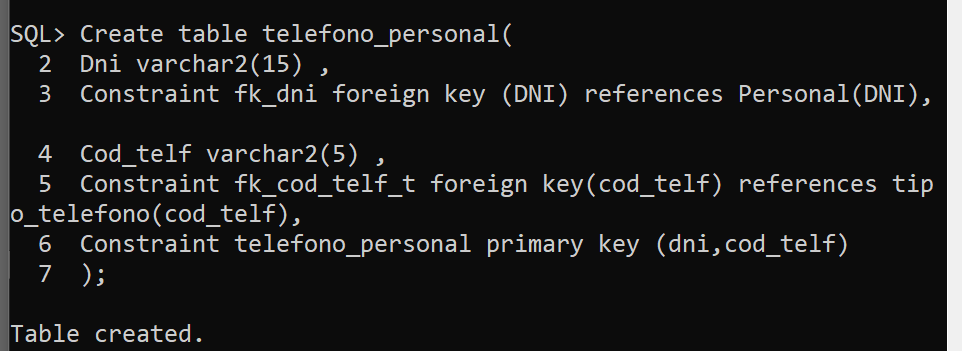
Constraint fk\_dni foreign key (DNI) references Personal(DNI),

Cod\_telf varchar2(5) ,

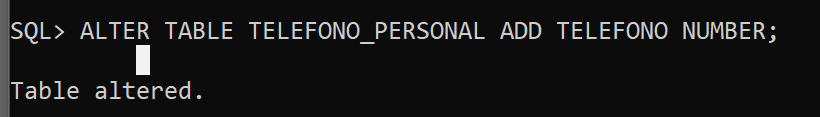
Constraint fk\_cod\_telf\_t foreign key(cod\_telf) references tipo\_telefono(cod\_telf),

Constraint telefono\_personal primary key (dni,cod\_telf)

);



ALTER TABLE TELEFONO\_PERSONAL ADD TELEFONO NUMBER;



-- **Creación de tabla Repartidor**

Create table Repartidor(

Licencia varchar2(5) not null primary key,

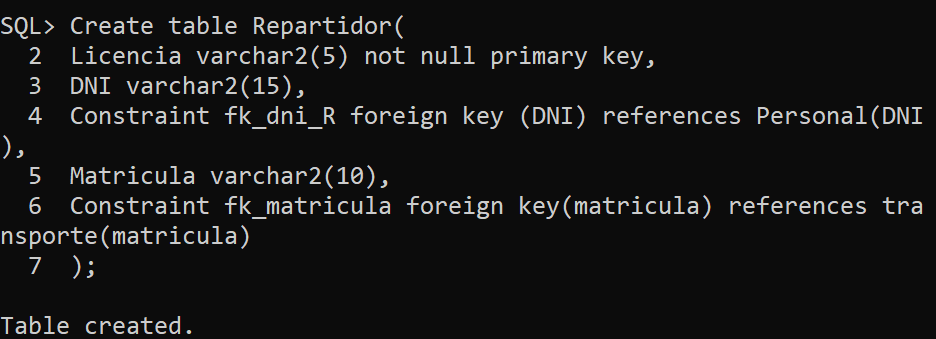
DNI varchar2(15),

Constraint fk\_dni\_R foreign key (DNI) references Personal(DNI),

Matricula varchar2(10),

Constraint fk\_matricula foreign key(matricula) references transporte(matricula)

);



-- **Creación de tabla Donador\_Insumo**

Create table Donador\_insumo (

Dni\_Ruc varchar2(15),

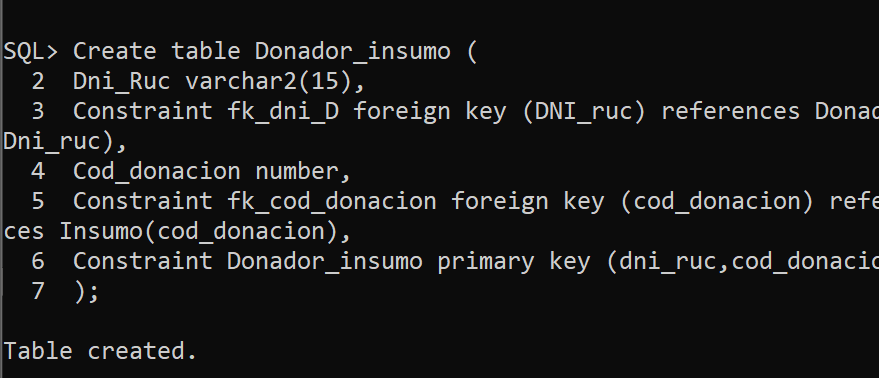
Constraint fk\_dni\_D foreign key (DNI\_ruc) references Donador(Dni\_ruc),

Cod\_donacion number,

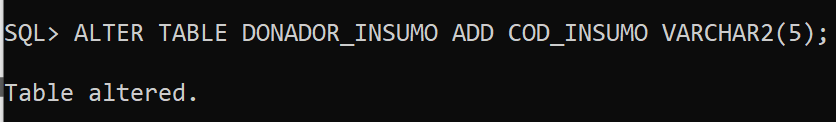
Constraint fk\_cod\_donacion foreign key (cod\_donacion) references Insumo(cod\_donacion),

Constraint Donador\_insumo primary key (dni\_ruc,cod\_donacion)

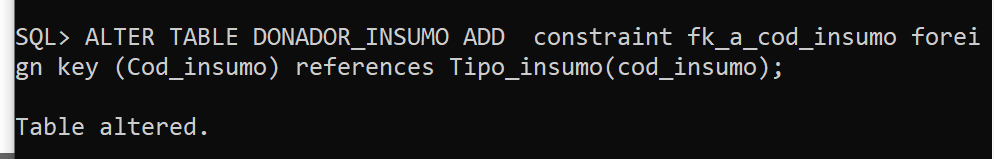
);



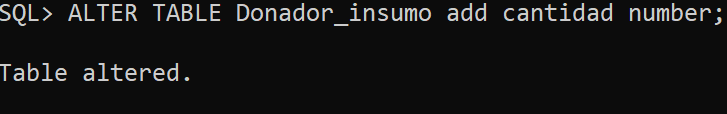
ALTER TABLE DONADOR\_INSUMO ADD COD\_INSUMO VARCHAR2(5);



ALTER TABLE DONADOR\_INSUMO ADD  constraint fk\_a\_cod\_insumo foreign key (Cod\_insumo) references Tipo\_insumo(cod\_insumo);



ALTER TABLE Donador\_insumo add cantidad number;



-- **Creación de tabla Insumo\_Personal\_Instalación\_medica**

Create table Insumo\_personal\_IM(

Cod\_insumo varchar2(5),

Constraint fk\_cod\_insumo\_I foreign key (cod\_insumo) references tipo\_Insumo(cod\_insumo),

DNI varchar2(15),

Constraint fk\_dni\_I foreign key (DNI) references Personal(DNI),

Codigo\_IM varchar2(5) not null,

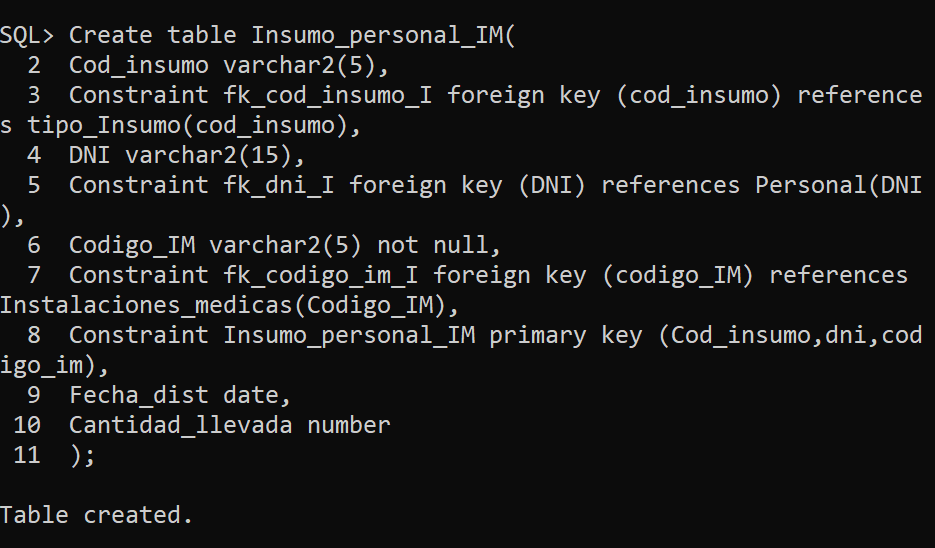
Constraint fk\_codigo\_im\_I foreign key (codigo\_IM) references Instalaciones\_medicas(Codigo\_IM),

Constraint Insumo\_personal\_IM primary key (Cod\_insumo,dni,codigo\_im),

Fecha\_dist date,

Cantidad\_llevada number

);



-- **Creación de tabla Donador\_Cedis**

Create table Donador\_CEDIS(

Dni\_Ruc varchar2(15),

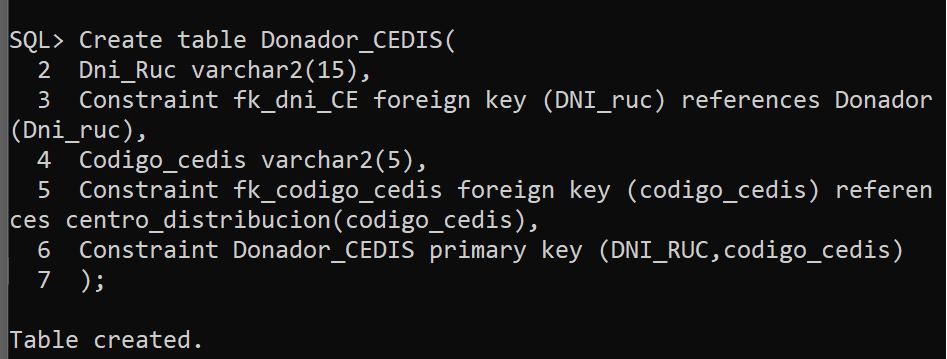
Constraint fk\_dni\_CE foreign key (DNI\_ruc) references Donador(Dni\_ruc),

Codigo\_cedis varchar2(5),

Constraint fk\_codigo\_cedis foreign key (codigo\_cedis) references centro\_distribucion(codigo\_cedis),

Constraint Donador\_CEDIS primary key (DNI\_RUC,codigo\_cedis)

);



-- **Creación de tabla Personal Cedis**

Create table personal\_CEDIS(

DNI varchar2(15),

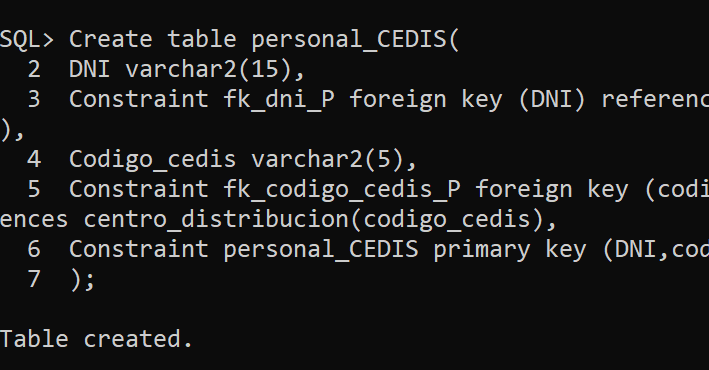
Constraint fk\_dni\_P foreign key (DNI) references Personal(DNI),

Codigo\_cedis varchar2(5),

Constraint fk\_codigo\_cedis\_P foreign key (codigo\_cedis) references centro\_distribucion(codigo\_cedis),

Constraint personal\_CEDIS primary key (DNI,codigo\_cedis)

);



**------CREACIÓN DE TABLA PARA REGISTRAR STATUS DEL PERSONAL**

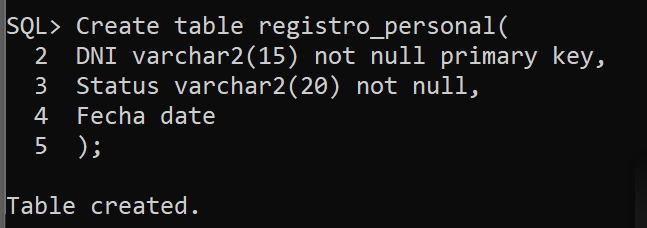
Create table registro\_personal(

DNI varchar2(15) not null primary key,

Status varchar2(20) not null,

Fecha date

);



**----CREACION DE TABLA PARA REGISTRAR TRANSPORTES NUEVOS O DAÑADOS**

Create table registro\_Transporte(

Matricula varchar2(15) not null primary key,

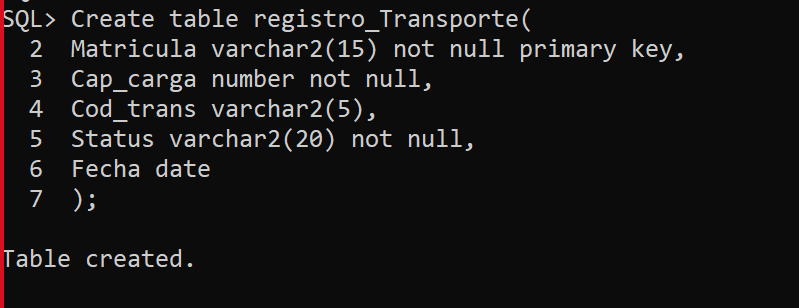
Cap\_carga number not null,

Cod\_trans varchar2(5),

Status varchar2(20) not null,

Fecha date

);



# PROCEDIMIENTOS UTILIZADOS

**--PROCEDIMIENTO PARA INSERTAR EN TIPO TELEFONO**

Create or Replace Procedure P\_Tipo\_tel(

  P\_cod\_telf Tipo\_Telefono.Cod\_Telf%TYPE,

  p\_desc\_telf Tipo\_Telefono.Desc\_Telf%TYPE

) AS

BEGIN

INSERT INTO Tipo\_Telefono (Cod\_Telf, Desc\_Telf) VALUES (  P\_cod\_telf,  p\_desc\_telf);

Exception

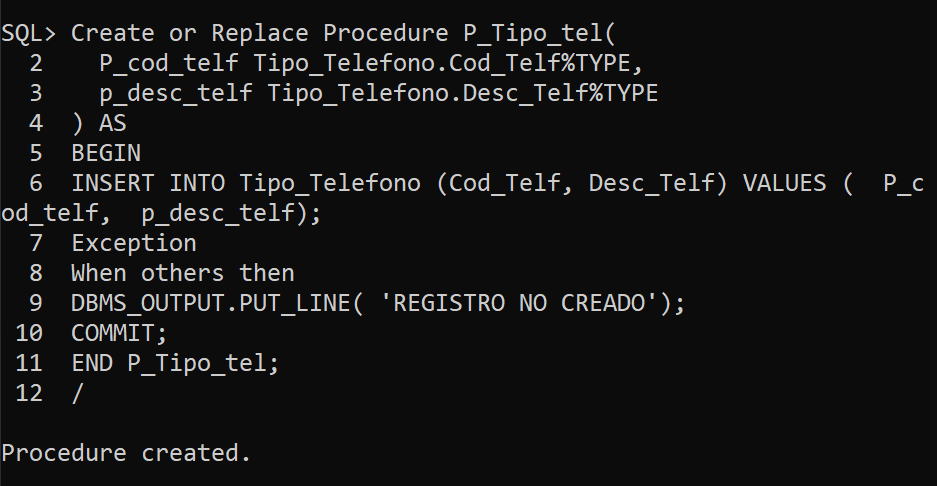
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Tipo\_tel;

/



**--PROCEDIMIENTO PARA INSERTAR EN TIPO CORREOS**

CREATE OR REPLACE PROCEDURE P\_Tipos\_Correos(

p\_Cod\_Correo             Tipo\_Email.Cod\_Email%TYPE,

p\_Desc\_Correo            Tipo\_Email.Desc\_Email%TYPE

) AS

BEGIN

INSERT INTO Tipo\_Email (Cod\_Email, Desc\_Email) VALUES (  p\_Cod\_Correo,  p\_Desc\_Correo);

Exception

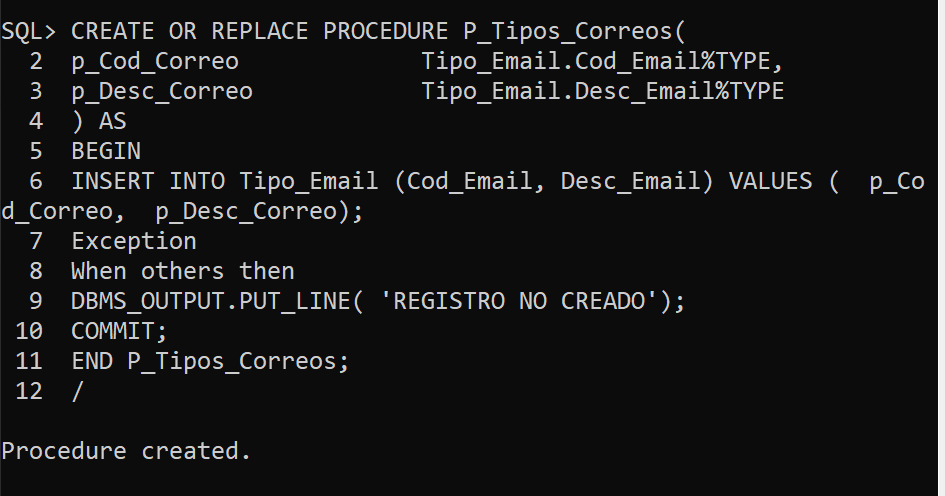
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Tipos\_Correos;

/



**--PROCEDIMIENTO PARA INSERTAR EN TIPO INSUMO**

Create or Replace Procedure P\_Tipo\_Insum(

 P\_cod\_insumo     Tipo\_Insumo.cod\_insumo%TYPE,

 p\_desc\_insum      Tipo\_Insumo.desc\_insumo%TYPE

) AS

BEGIN

INSERT INTO Tipo\_Insumo(cod\_insumo,desc\_insumo) VALUES (P\_cod\_insumo, p\_desc\_insum);

Exception

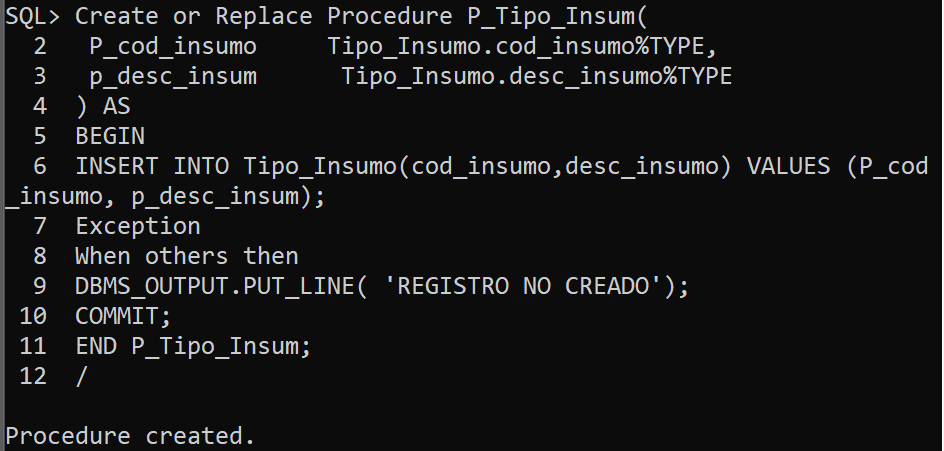
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Tipo\_Insum;

/



**--PROCEDIMIENTO PARA INSERTAR EN TIPO TRANSPORTE**

Create or Replace Procedure P\_Tipo\_Transp(

 P\_cod\_trans      Tipo\_transporte.cod\_trans%TYPE,

 p\_desc\_trans     Tipo\_transporte.desc\_trans%TYPE

) AS

BEGIN

INSERT INTO Tipo\_transporte(cod\_trans,desc\_trans) VALUES (P\_cod\_trans, p\_desc\_trans);

Exception

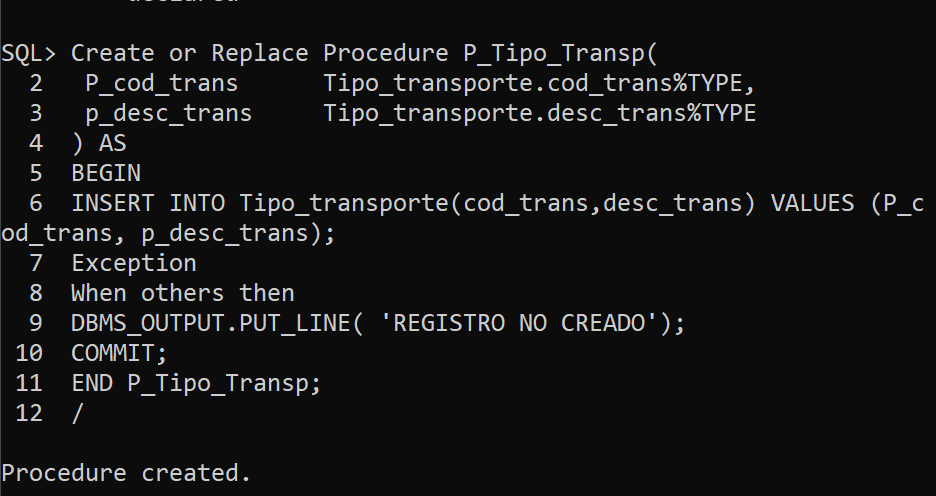
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Tipo\_Transp;

/



**--PROCEDIMIENTO PARA INSERTAR EN TIPO PERSONAL**

Create or Replace Procedure P\_Tipo\_Person(

P\_cod\_cargo Tipo\_personal.cod\_cargo%TYPE,

p\_descripcion Tipo\_personal.descripcion%TYPE

) AS

BEGIN

INSERT INTO Tipo\_personal(cod\_cargo,descripcion) VALUES (P\_cod\_cargo, p\_descripcion);

Exception

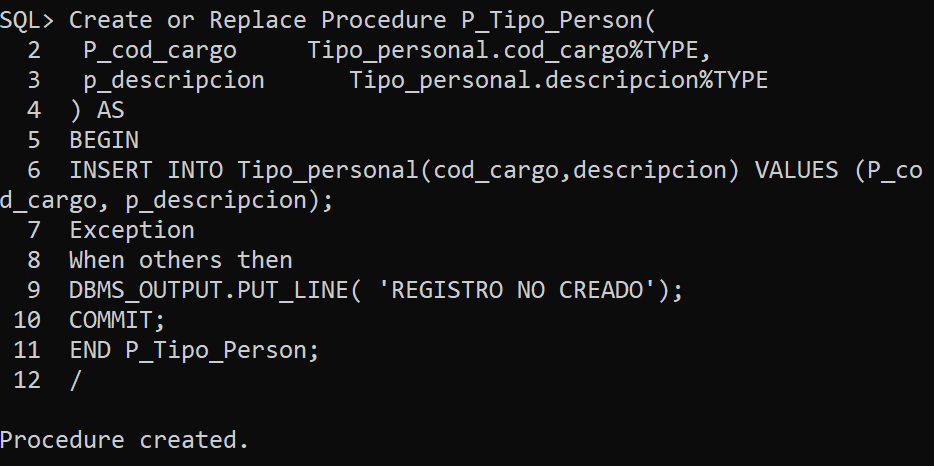
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Tipo\_Person;

/



**--PROCEDIMIENTO PARA INSERTAR EN EMAIL DEL DONADOR**

CREATE OR REPLACE PROCEDURE P\_Email\_donad(

P\_email             Email\_donador.email%TYPE,

p\_dni\_ruc            Email\_donador.dni\_ruc%TYPE,

p\_cod\_email             Email\_donador.Cod\_Email%TYPE

) AS

BEGIN

INSERT INTO Email\_donador (email,dni\_ruc,cod\_email) VALUES (P\_email,P\_dni\_ruc,P\_cod\_email);

Exception

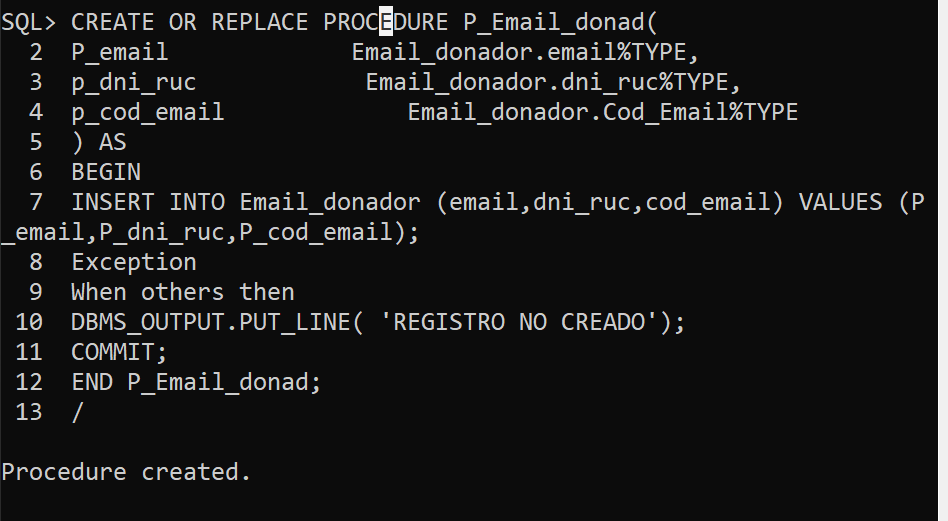
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Email\_donad;

/



**--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA DONADOR**

CREATE OR REPLACE PROCEDURE p\_DONADOR (

P\_dni\_ruc             DONADOR.dni\_ruc%Type,

p\_NOMBRE      DONADOR.nombre%TYPE,

p\_APELLIDO      DONADOR.apellido%TYPE,

p\_PROVINCIA    DONADOR.provincia%TYPE,

p\_DISTRITO    DONADOR.distrito%TYPE,

p\_CORREGIMIENTO    DONADOR.corregimiento%TYPE,

p\_CALLE DONADOR.calle%TYPE,

p\_NUM\_CASA DONADOR.no\_casa%TYPE

)

AS

BEGIN

INSERT INTO DONADOR ( dni\_ruc,NOMBRE, APELLIDO, PROVINCIA, DISTRITO, CORREGIMIEnTO, CALLE, NO\_CASA)

Values (p\_dni\_ruc, p\_NOMBRE, p\_APELLIDO, p\_PROVINCIA, p\_DISTRITO, p\_CORREGIMIENTO, p\_CALLE, p\_NUM\_CASA);

EXCEPTION

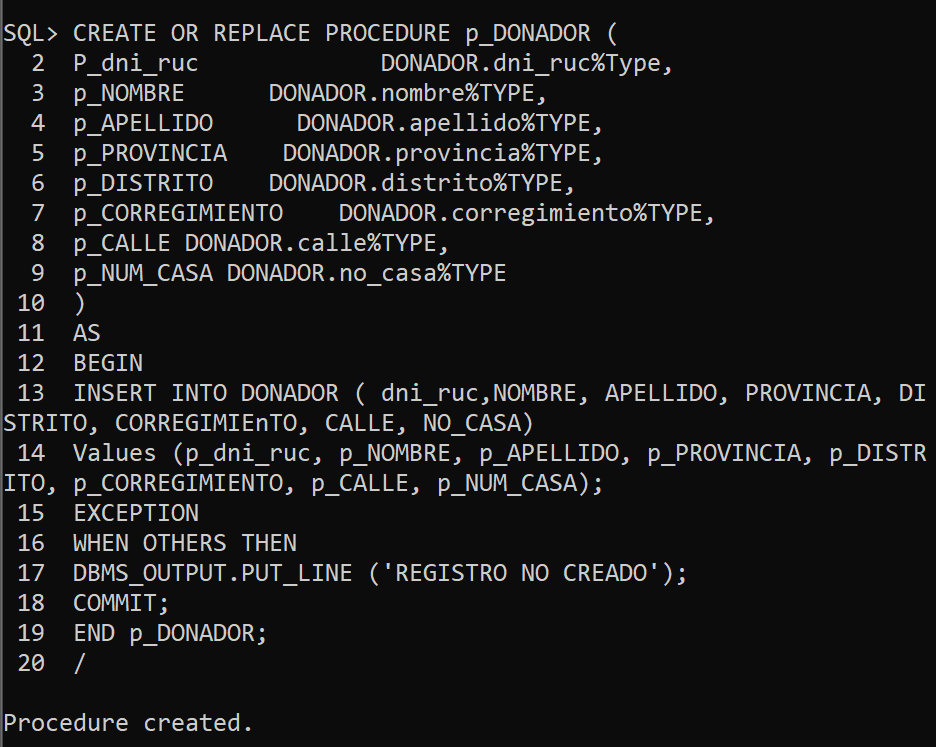
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('REGISTRO NO CREADO');

COMMIT;

END p\_DONADOR;

/



**--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA TELEFONO\_DONADOR**

CREATE OR REPLACE PROCEDURE P\_Telefono\_donador(

p\_DNI\_RUC Telefono\_donador.DNI\_RUC%TYPE,

p\_Cod\_Telf Telefono\_donador.Cod\_Telf%TYPE,

p\_telefono Telefono\_donador.telefono%TYPE

) AS

BEGIN

INSERT INTO Telefono\_Donador ( DNI\_RUC, Cod\_Telf,TELEFONO ) VALUES (p\_DNI\_RUC, p\_Cod\_Telf,P\_TELEFONO);

Exception

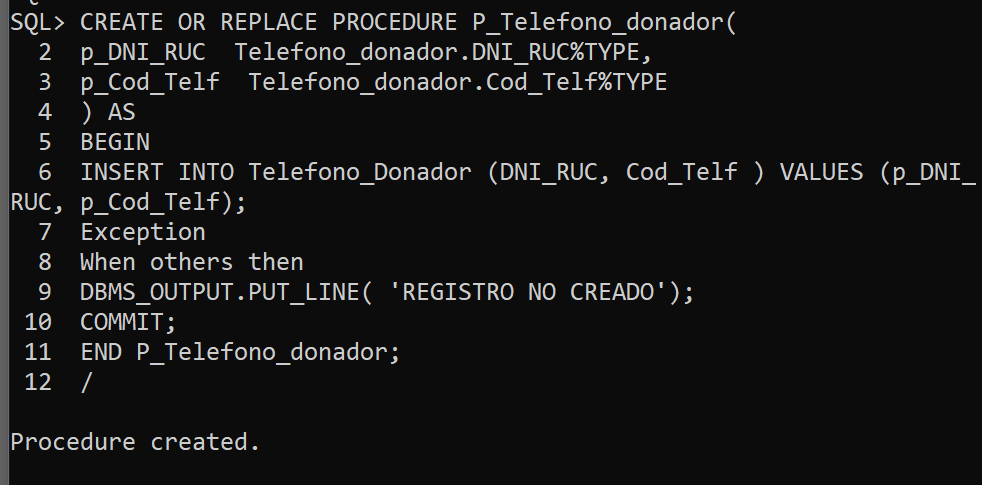
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Telefono\_donador;

/



--**-PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA TELEFONO\_PERSONAL**

Create or Replace Procedure P\_telf\_Personal(

P\_DNI telefono\_personal.Dni%TYPE,

  P\_cod\_telf Telefono\_personal.Cod\_Telf%TYPE,

  p\_Telefono Telefono\_personal.telefono%TYPE

) AS

BEGIN

INSERT INTO Telefono\_personal (Dni,Cod\_Telf, telefono)

VALUES (  p\_DNI,P\_cod\_telf,  p\_Telefono);

Exception

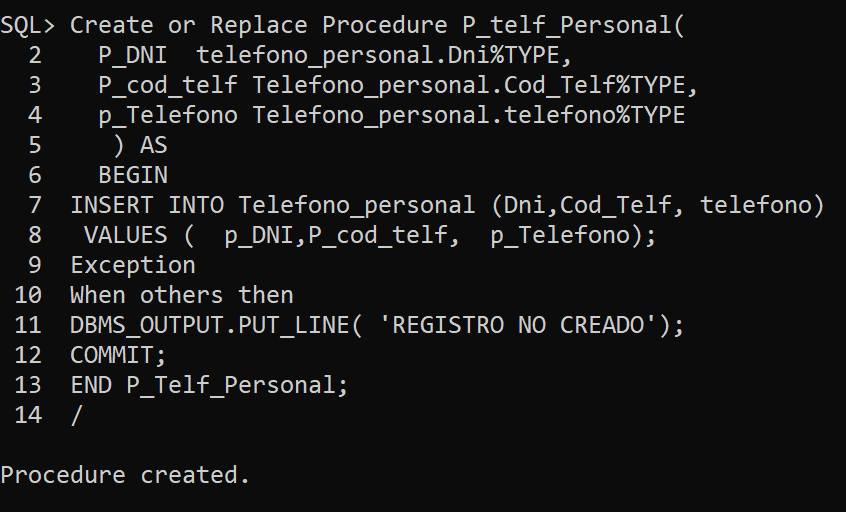
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Telf\_Personal;

/



--**-PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA TELEFONO\_INSTALACION\_MEDICA**

CREATE OR REPLACE PROCEDURE P\_Telefono\_IM(

p\_Codigo\_IM  telefono\_im.Codigo\_IM%TYPE,

p\_Cod\_telf  telefono\_im.Cod\_telf%TYPE,

P\_TELEFONO TELEFONO\_IM.TELEFONO%TYPE

) AS

BEGIN

INSERT INTO telefono\_im (Codigo\_IM, Cod\_telf,TELEFONO) VALUES (p\_Codigo\_IM, p\_Cod\_telf,P\_TELEFONO);

Exception

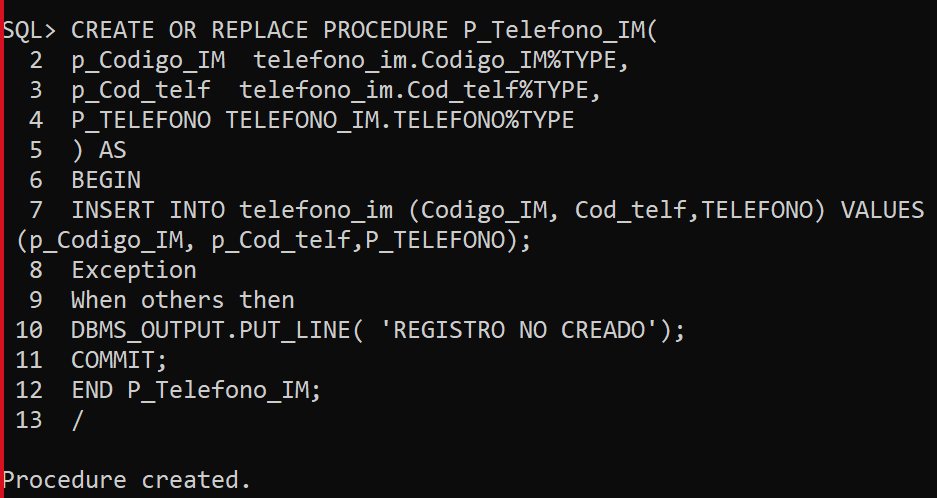
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Telefono\_IM;

/



------**PROCEDIMIENTO DE INSUMO\_PERSONAL\_IM**

Create or replace procedure P\_distribucion(

p\_cod\_insumo insumo\_personal\_im.cod\_insumo%TYPE,

p\_DNI insumo\_personal\_im.DNI%TYPE,

p\_codigo\_im insumo\_personal\_im.codigo\_im %TYPE,

p\_fecha\_dist insumo\_personal\_im.fecha\_dist%TYPE,

p\_cantidad\_llevada insumo\_personal\_im.cantidad\_llevada%TYPE

)as

BEGIN

Insert into insumo\_personal\_im values(p\_cod\_insumo, p\_DNI, p\_codigo\_im, p\_fecha\_dist, p\_cantidad\_llevada);

Exception

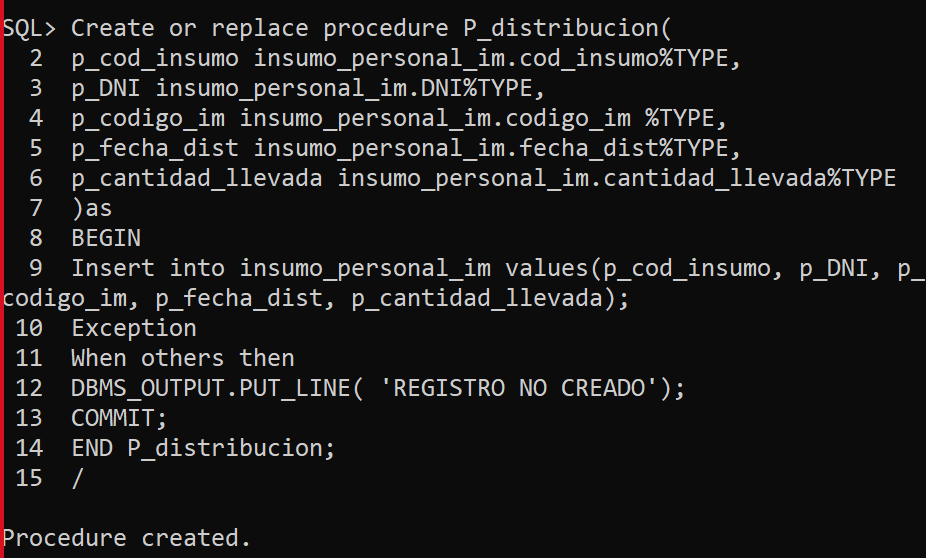
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_distribucion;

/



**--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA DONADOR\_INSUMO**

CREATE OR REPLACE PROCEDURE P\_Donador\_insumo (

p\_Dni\_Ruc  Donador\_insumo.DNI\_RUC%TYPE,

p\_Cod\_donacion  Donador\_insumo.Cod\_donacion%TYPE,

P\_COD\_INSUMO DONADOR\_INSUMO.COD\_INSUMO%TYPE,

p\_cantidad Donador\_insumo.cantidad%TYPE

) AS

BEGIN

INSERT INTO Donador\_insumo (Dni\_Ruc,Cod\_donacion, COD\_INSUMO,cantidad) VALUES (p\_Dni\_Ruc, p\_Cod\_donacion, P\_COD\_INSUMO,p\_cantidad);

Exception

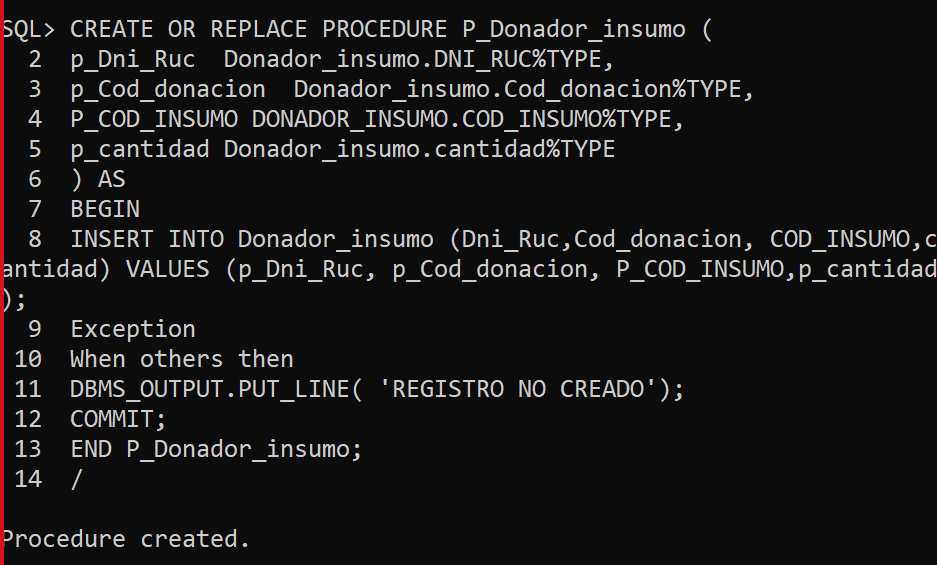
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Donador\_insumo;

/



**--PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA CENTRO\_DISTRIBUCION**

CREATE OR REPLACE PROCEDURE p\_Centro\_distr(

P\_cod\_cedis             Centro\_distribucion.codigo\_cedis%TYPE,

p\_nombre      Centro\_distribucion.nombre%Type,

p\_Region      Centro\_distribucion.Region%Type, p\_Provincia    Centro\_distribucion.Provincia%Type,

p\_Ubicacion    Centro\_distribucion.ubicacion %Type

)

AS

BEGIN

INSERT INTO Centro\_distribucion (Codigo\_cedis,Nombre,Region,Provincia, Ubicacion)

Values (p\_cod\_cedis, p\_nombre, p\_Region, p\_Provincia, p\_Ubicacion);

EXCEPTION

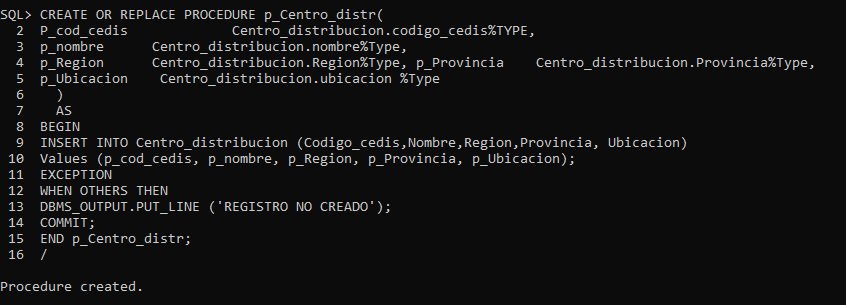
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('REGISTRO NO CREADO');

COMMIT;

END p\_Centro\_distr;

/



--**-PROCEDIMIENTO PARA ALMACENAMIENTO DE LA TABLA INSTALACIONES MEDICAS**

CREATE OR REPLACE PROCEDURE p\_instalaciones (

P\_cod\_IM           Instalaciones\_medicas.codigo\_IM%TYPE,

p\_nombre      Instalaciones\_medicas.nombre%TYPE,

p\_Region      Instalaciones\_medicas.Region%TYPE,

p\_Ubicacion    Instalaciones\_medicas.ubicacion%Type

)

AS

BEGIN

INSERT INTO Instalaciones\_medicas (Codigo\_IM, Nombre,Region, Ubicacion)

Values (p\_cod\_IM, p\_nombre, p\_Region, p\_Ubicacion);

EXCEPTION

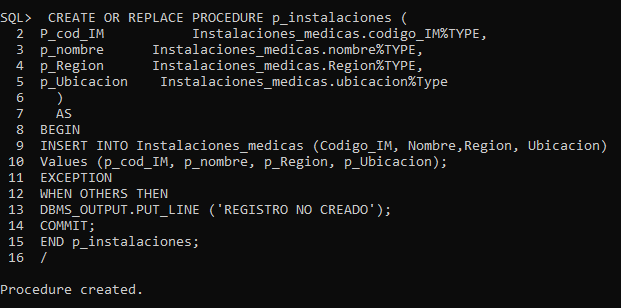
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('REGISTRO NO CREADO');

COMMIT;

END p\_instalaciones;

/



**--PROCEDIMIENTO PARA CARGAR DATOS EN LA TABLA DONADOR\_CEDIS**

Create or replace procedure P\_donador\_cedis(

P\_Dni\_Ruc Donador\_CEDIS.Dni\_Ruc%TYPE,

P\_Codigo\_cedis Donador\_CEDIS.Codigo\_cedis%TYPE

)AS

BEGIN

Insert into Donador\_CEDIS(Dni\_Ruc, Codigo\_cedis)

Values(p\_Dni\_Ruc,p\_Codigo\_cedis);

Exception

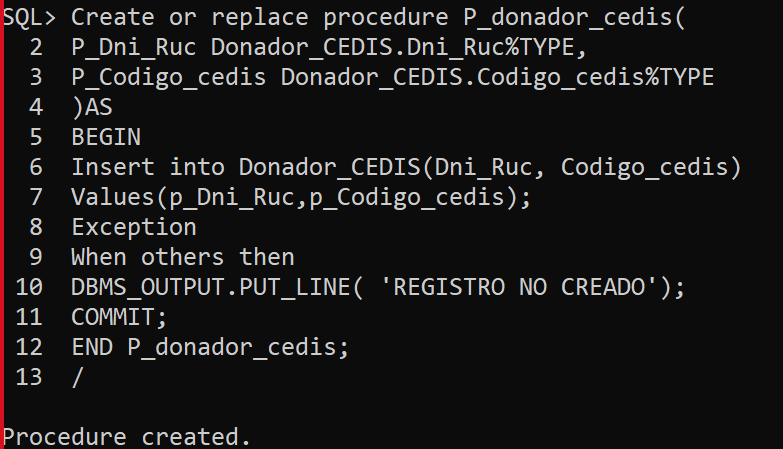
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_donador\_cedis;

/



**--PROCEDIMIENTO PARA CARGAR DATOS EN LA TABLA PERSONAL\_CEDIS**

Create or replace procedure P\_personal\_cedis(

P\_Dni personal\_CEDIS.Dni%TYPE,

P\_Codigo\_cedis personal\_CEDIS.Codigo\_cedis%TYPE

)AS

BEGIN

Insert into personal\_CEDIS(Dni, Codigo\_cedis)

Values(p\_Dni, p\_Codigo\_cedis);

Exception

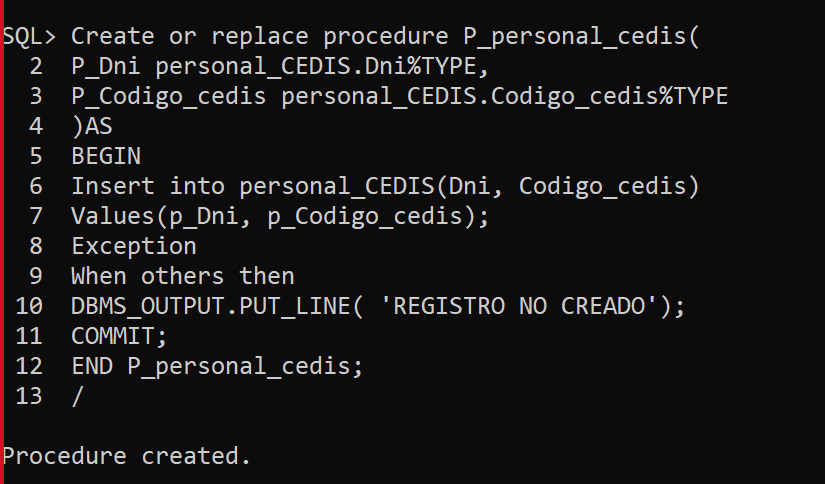
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_personal\_cedis;

/



**--PROCEDIMIENTO PARA CARGAR DATOS EN LA TABLA TRANSPORTE**

Create or replace procedure P\_transporte(

P\_Matricula transporte.matricula%TYPE,

P\_Cap\_carga transporte.cap\_carga%TYPE,

P\_Cod\_trans transporte.cod\_trans%TYPE

)AS

BEGIN

Insert into transporte(matricula, cap\_carga, cod\_trans)

Values(p\_matricula, p\_cap\_carga, P\_cod\_trans);

Exception

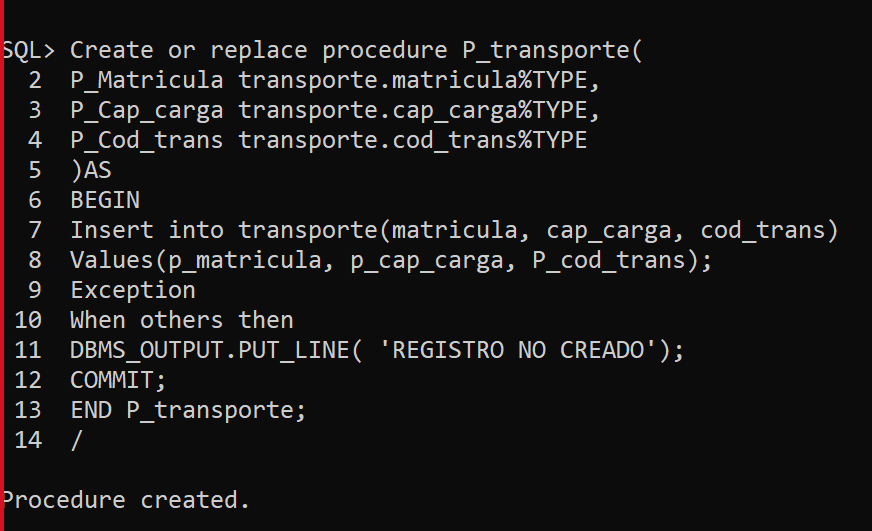
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_transporte;

/



**--PROCEDIMIENTO PARA CARGAR DATOS A LA TABLA PERSONAL**

CREATE OR REPLACE PROCEDURE P\_Personal (

p\_Dni    Personal.DNI%TYPE,

p\_Nombre  Personal.Nombre%TYPE,

p\_Apellido  Personal.Apellido%TYPE,

p\_Fecha\_nac  Personal.fecha\_nac%TYPE,

p\_Ciudad  Personal.ciudad%TYPE,

p\_Corregimiento  Personal.corregimiento%TYPE,

p\_calle  Personal.calle%TYPE,

p\_num\_casa  Personal.num\_casa%TYPE,

p\_cod\_cargo  Personal.cod\_cargo%TYPE

 ) AS

BEGIN

INSERT INTO Personal  (Dni,Nombre, Apellido, Fecha\_nac, Ciudad, Corregimiento, Calle, num\_casa, cod\_cargo) VALUES (p\_Dni,p\_Nombre, p\_Apellido, p\_Fecha\_nac, p\_Ciudad, p\_Corregimiento, p\_Calle, p\_num\_casa, p\_cod\_cargo);

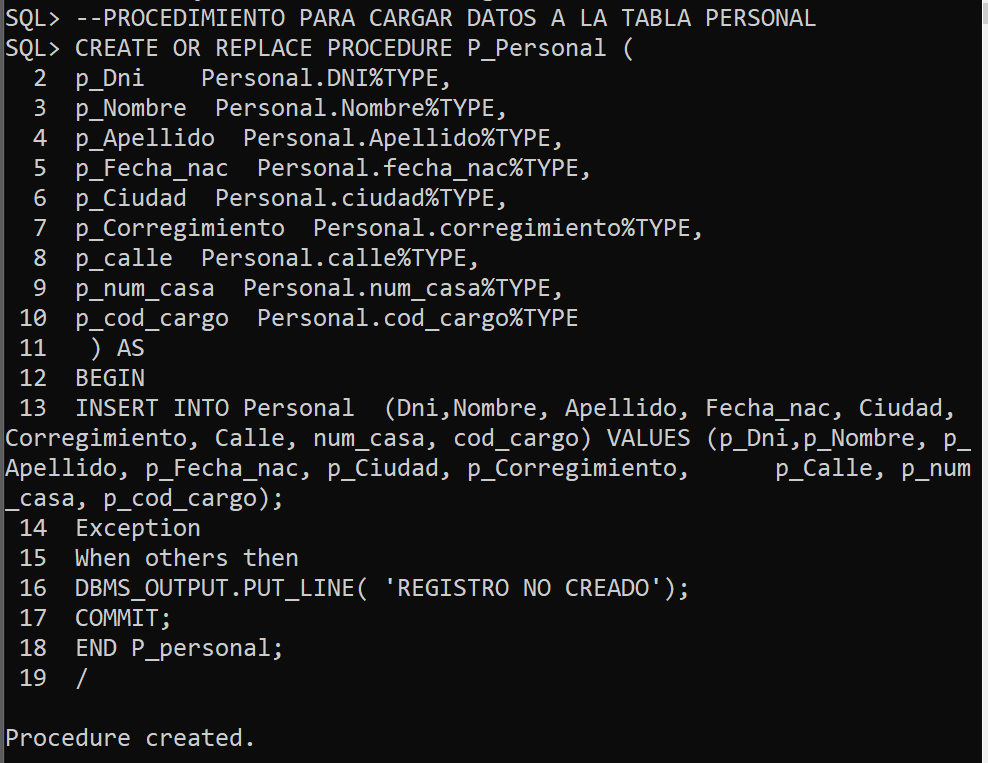
Exception

When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_personal;

/

**--PROCEDIMIENTO PARA CARGAR DATOS EN LA TABLA INSUMO**

CREATE OR REPLACE PROCEDURE P\_Insumo (

p\_Cod\_donacion  Insumo.cod\_donacion%TYPE,

p\_Cod\_insumo  Insumo.Cod\_insumo%TYPE,

p\_cantidad Insumo.cantidad\_TOTAL%TYPE,

P\_Descripcion Insumo.descripcion%type

) AS

BEGIN

INSERT INTO insumo (Cod\_donacion,  Cod\_insumo, cantidad\_TOTAL, descripcion) VALUES (P\_Cod\_donacion,  p\_Cod\_insumo, p\_cantidad,p\_descripcion);

Exception

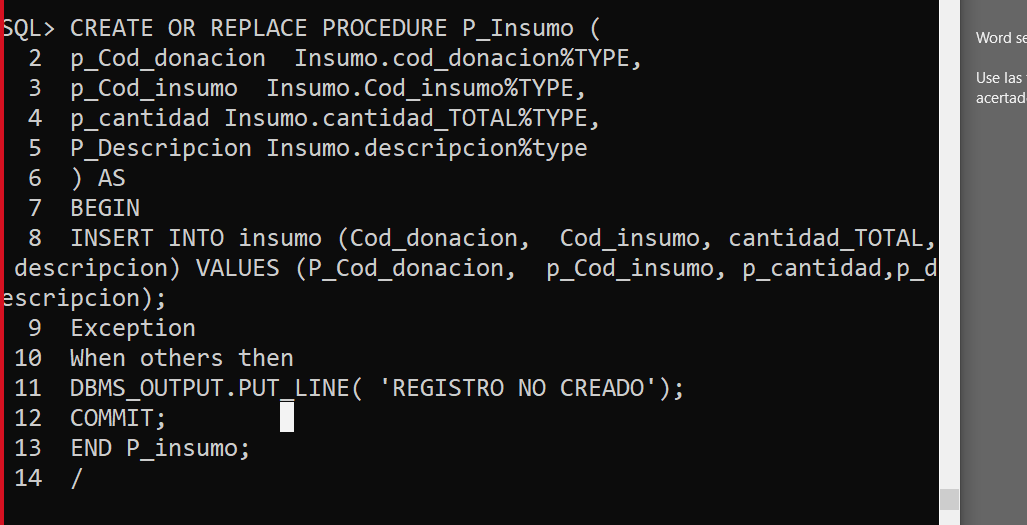
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_insumo;

/



**----PROCEDIMIENTO PARA CARGAR DATOS EN LA TABLA REPARTIDOR**

CREATE OR REPLACE PROCEDURE P\_Repartidor(

v\_Licencia Repartidor.Licencia%TYPE,

v\_DNI  Repartidor.DNI%TYPE,

v\_Matricula  Repartidor.Matricula%TYPE

)

AS

BEGIN

INSERT INTO  Repartidor (Licencia, DNI, Matricula ) VALUES (v\_Licencia, v\_DNI, v\_Matricula);

Exception

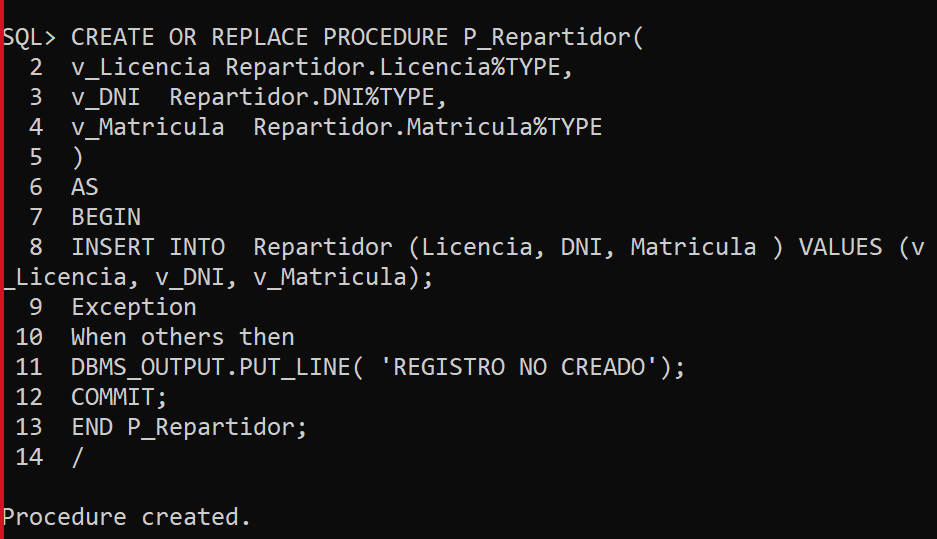
When others then

DBMS\_OUTPUT.PUT\_LINE( 'REGISTRO NO CREADO');

COMMIT;

END P\_Repartidor;

/



**----**FUNCION PARA PROCEDIMIENTO TABLA INSUMO\_INSTALACION MEDICA Y PERSONAL

Create or replace function f\_cal\_cantidad(

f\_cantidad  insumo.CANTIDAD\_TOTAL%TYPE,

f\_cantidad\_llevada  Insumo\_personal\_IM.cantidad\_llevada%TYPE

)return number

AS

Cantidad\_nueva number;

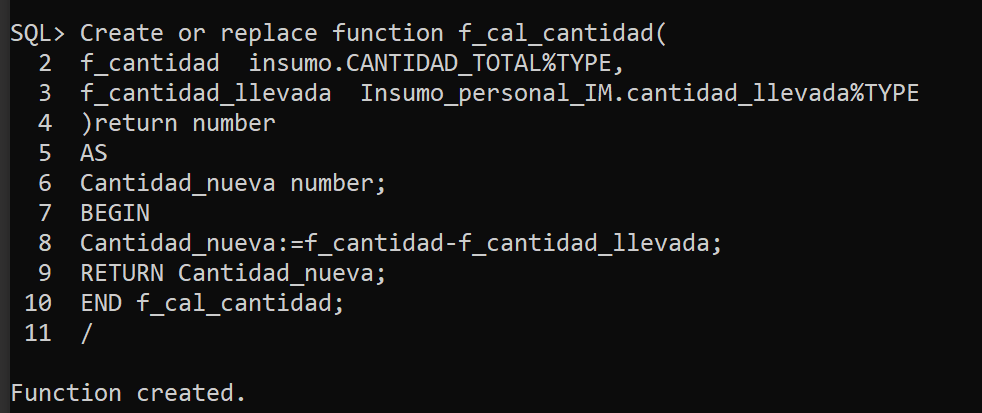
BEGIN

Cantidad\_nueva:=f\_cantidad-f\_cantidad\_llevada;

RETURN Cantidad\_nueva;

END f\_cal\_cantidad;

/



**--PROCEDIMIENTO PARA CARGAR Y ACTUALIZAR TABLA INSUMO\_INSTALACION MEDICA Y PERSONAL**

Create or replace procedure P\_cantidad\_distribuida

AS

p\_cod\_insumo insumo.cod\_insumo%TYPE;

p\_id\_insumo Insumo\_personal\_IM.cod\_insumo%TYPE;

p\_cantidad insumo.cantidad\_total%TYPE;

p\_cantidad\_llevada  Insumo\_personal\_IM.cantidad\_llevada%TYPE;

p\_descripcion insumo.descripcion%TYPE;

CURSOR recorre\_insumo is

Select cod\_insumo, cantidad\_total, descripcion from insumo;

BEGIN

OPEN recorre\_insumo;

LOOP

FETCH recorre\_insumo INTO p\_cod\_insumo, p\_cantidad, p\_descripcion;

--Select cantidad\_total into p\_cantidad from insumo where cod\_insumo=p\_cod\_insumo;

/\*Select cantidad\_llevada into p\_cantidad\_llevada from Insumo\_personal\_IM where cod\_insumo=p\_id\_insumo;\*/

IF (p\_cantidad\_llevada>p\_cantidad) THEN

DBMS\_OUTPUT.PUT\_LINE('Insumo agotado');

ELSIF (p\_cantidad\_llevada<=p\_cantidad) THEN

UPDATE INSUMO

Set

Cantidad\_total= f\_cal\_cantidad(p\_cantidad,p\_cantidad\_llevada)

Where p\_cod\_insumo=p\_id\_insumo;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insumo no encontrado');

END IF;

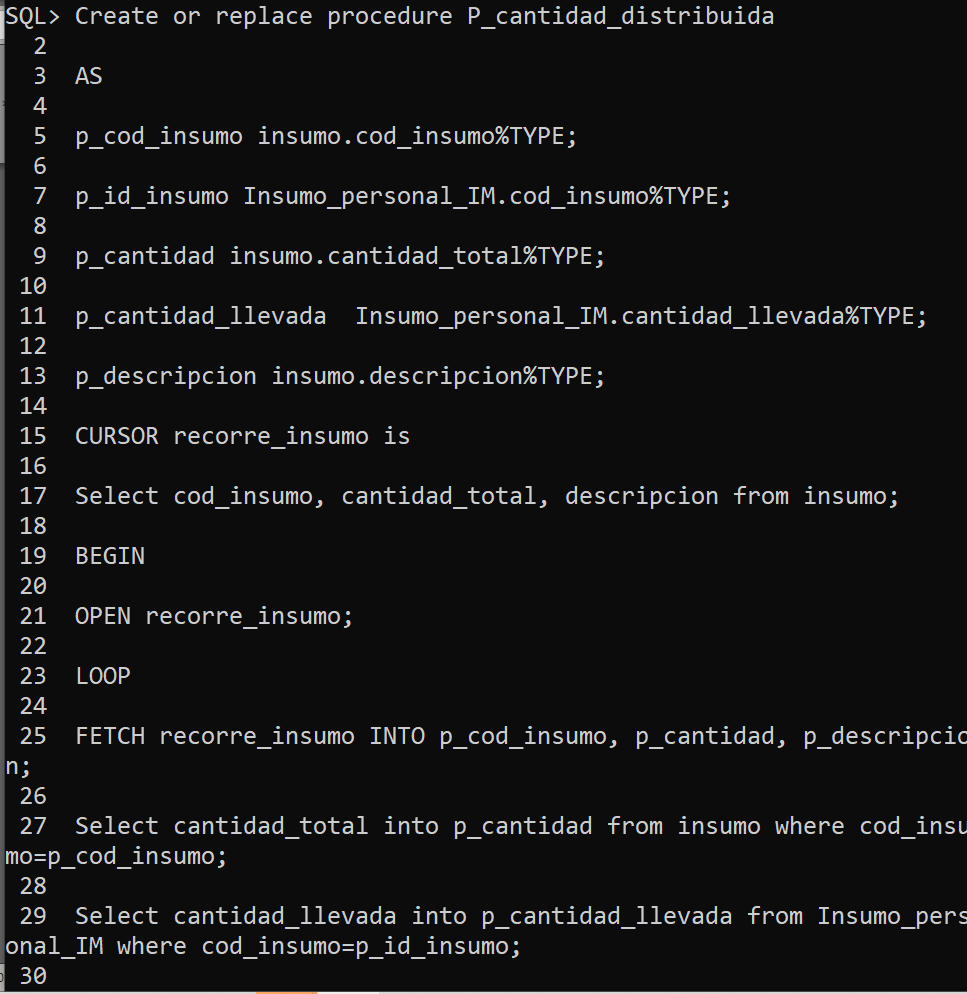
EXIT WHEN recorre\_insumo%NOTFOUND;

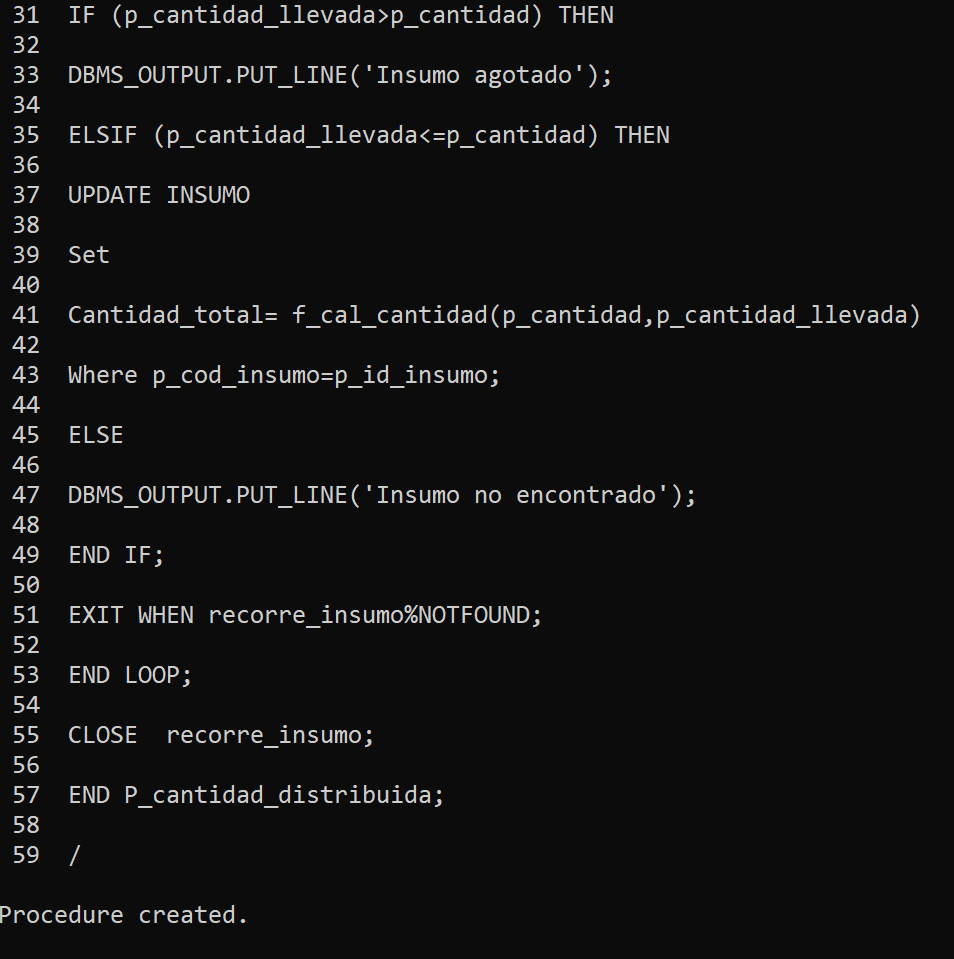
END LOOP;

CLOSE  recorre\_insumo;

END P\_cantidad\_distribuida;

/





# IMPLEMENTACION DE TRIGGERS

**--TRIGGER PARA INSERTAR Y ELIMINAR REGISTROS DE LA TABLA PERSONAL**

Create or replace trigger T\_personal

After insert or delete on Personal

For each row

BEGIN

IF INSERTING THEN

INSERT INTO registro\_personal ( DNI, Status, Fecha)

VALUES (:new.DNI, ‘INSERTADO’, SYSDATE);

ELSIF DELETING THEN

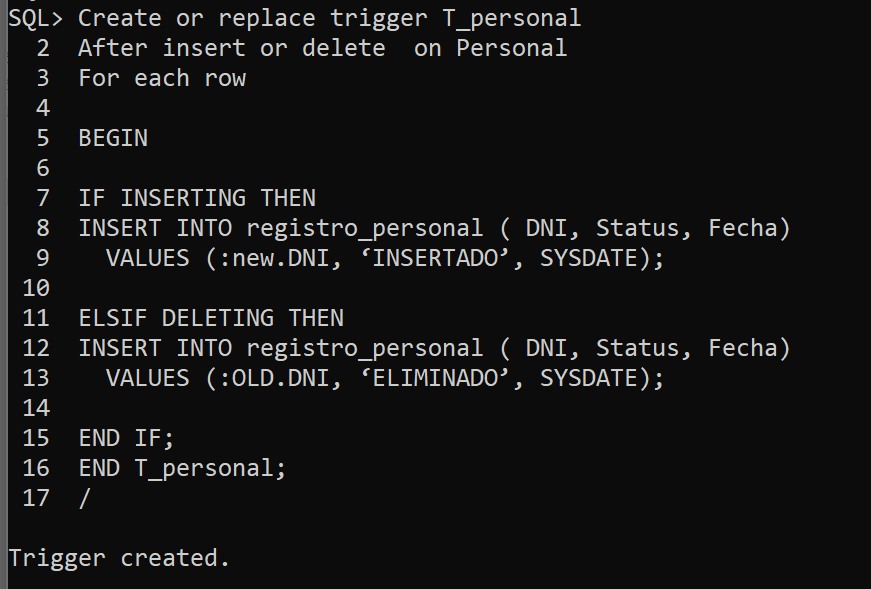
INSERT INTO registro\_personal ( DNI, Status, Fecha)

VALUES (:old.DNI, ‘ELIMINADO’, SYSDATE);

END IF;

END T\_personal;

/



**---TRIGGER PARA INSERTAR O ELIMINAR TRANSPORTE NUEVO O DAÑADO**

Create or replace trigger T\_Transporte

After insert or delete on Transporte

For each row

BEGIN

IF INSERTING THEN

INSERT INTO registro\_transporte ( matricula,cap\_carga, cod\_trans, Status, Fecha)

VALUES (:new.matricula,:new.cap\_carga, :new.cod\_trans,‘TRANSPORTE NUEVO’, SYSDATE);

ELSIF DELETING THEN

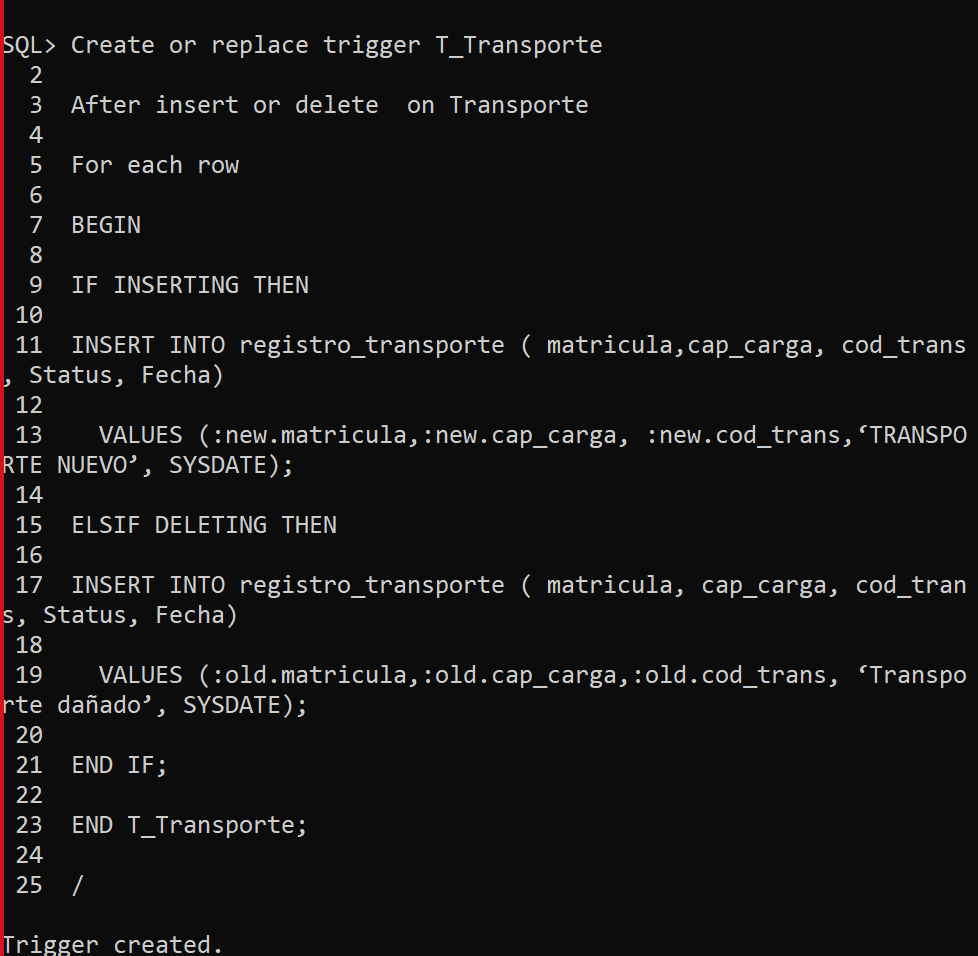
INSERT INTO registro\_transporte ( matricula, cap\_carga, cod\_trans, Status, Fecha)

VALUES (:old.matricula,:old.cap\_carga,:old.cod\_trans, ‘Transporte dañado’, SYSDATE);

END IF;

END T\_Transporte;

/



**---TRIGGER PARA ACTUALIZAR LA TABLA TRANSPORTE**

CREATE OR REPLACE TRIGGER Act\_Transporte

AFTER INSERT ON Registro\_transporte

for each row

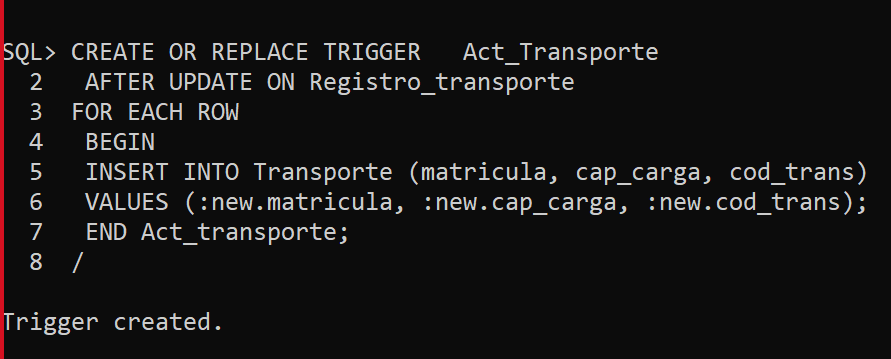
BEGIN

INSERT INTO Transporte (matricula, cap\_carga, cod\_trans)

VALUES (:new.matricula, :new.cap\_carga, :new.cod\_trans);

END Act\_transporte;

/



**--TRIGGER PARA ACTUALIZAR CANTIDAD TOTAL DE INSUMOS**

Create or replace trigger Act\_Insumo

After  Insert  on Donador\_insumo

For each row

BEGIN

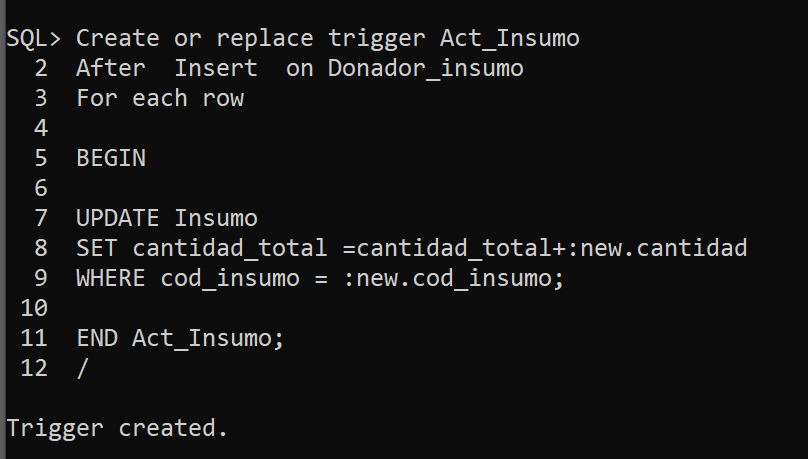
UPDATE Insumo

SET cantidad\_total =cantidad\_total+:new.cantidad

WHERE cod\_insumo = :new.cod\_insumo;

END Act\_Insumo;

/



**------VISTAS—**

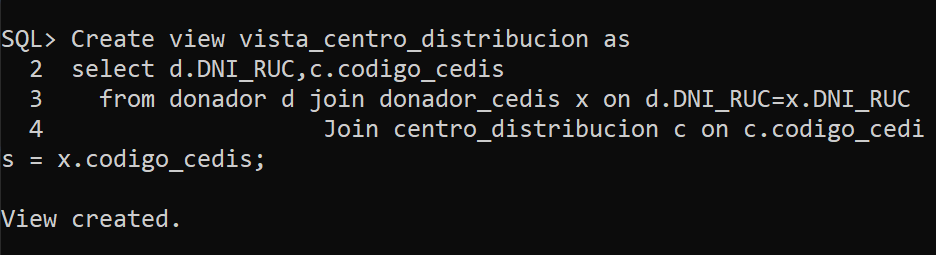
**-------VISTA DE CENTRO DE DISTRIBUCIÓN-----**

Create view vista\_centro\_distribucion as

select d.DNI\_RUC,c.codigo\_cedis

from donador d join donador\_cedis x on d.DNI\_RUC=x.DNI\_RUC

Join centro\_distribucion c on c.codigo\_cedis = x.codigo\_cedis;



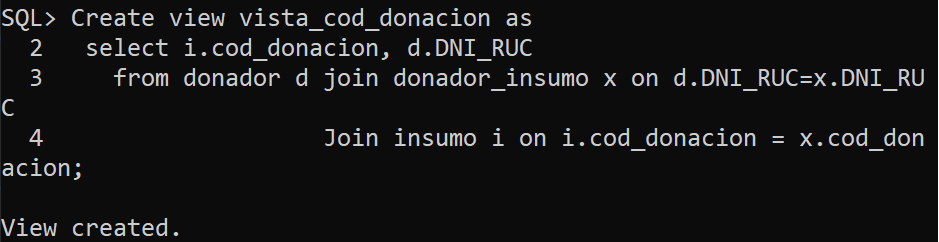
**--------VISTA DE COD\_DONACION---**

Create view vista\_cod\_donacion as

select i.cod\_donacion, d.DNI\_RUC

from donador d join donador\_insumo x on d.DNI\_RUC=x.DNI\_RUC

Join insumo i on i.cod\_donacion = x.cod\_donacion;

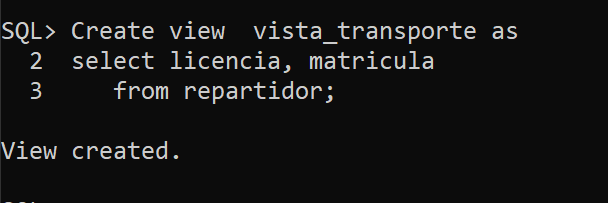


**-----VISTA DE TRANSPORTE**

Create view vista\_transporte as

select licencia, matricula

from repartidor;



**---------------VISTA DE PERSONAL**

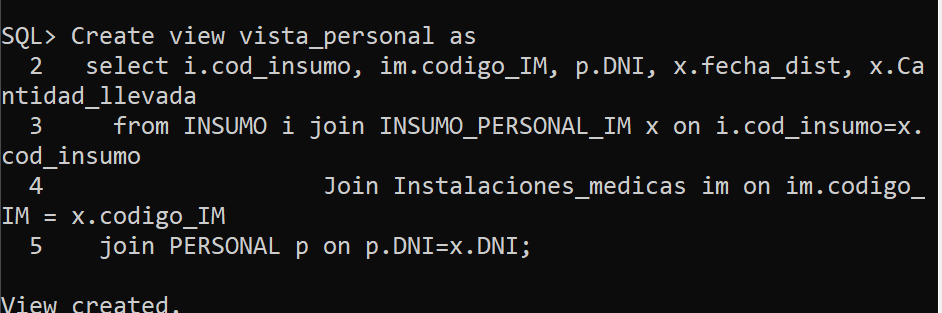
Create view vista\_personal as

select i.cod\_insumo, im.codigo\_IM, p.DNI, x.fecha\_dist, x.Cantidad\_llevada

from INSUMO i join INSUMO\_PERSONAL\_IM x on i.cod\_insumo=x.cod\_insumo

Join Instalaciones\_medicas im on im.codigo\_IM = x.codigo\_IM

join PERSONAL p on p.DNI=x.DNI;



# CONCLUSIÓN

El desarrollo de este proyecto nos ha permitido el desarrollo e implementación de los diferentes tipos de técnicas aprendidas durante el curso, de una manera más analítica ya que nos exigía de un poco más de compromiso y esfuerzo para el desarrollo del mismo, también se debe resaltar que esto nos ha abierto un panorama mucho mas amplio de cómo funcionan todos y cada uno de los aspectos aplicados en la implementación del esquema de base de datos ya que dependía casi al 100% de nosotros, y la importancia del estar preparados ante problemas adversos que puedan ocurrir, hemos logrado crear un nivel de conciencia de la importancia de que los centros médicos y el gobierno central, cuenten con una logística bien definida para el abastecimiento de los insumos que requiere, ya que hemos podido notar que sin un sistema de este tipo el sistema de salud nacional podría verse gravemente afectado.