Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Licenciatura en Ingeniería de Sistemas de Información

Departamento de Sistemas de Información, Control y Evaluación de Recursos Informáticos

Sistemas de Bases de Datos II

Equipo #1

Proyecto Final: “Implementación de Programación Almacenada para Base de Datos diseñada previamente”

Facilitador:

Ing. Henry Lezcano

Integrantes:

Aguilar, Milagros 3-740-771

Atencio, Anel 8-950-868

Márquez, Paola 8-949-1108

Rojas, Reynaldo 8-950-792

Grupo:

1IF131

II Semestre, 2020

**TABLA DE CONTENIDO**

[**INTRODUCCIÓN** 2](#_Toc57619718)

[**Misión de la Base de Datos** 3](#_Toc57619719)

[**Objetivos de la misión de la Base de Datos:** 3](#_Toc57619720)

[**Definición del Proyecto** 4](#_Toc57619721)

**[a.](#_Toc57619722)****[Ámbito](#_Toc57619722)** [4](#_Toc57619722)

**[b.](#_Toc57619723)****[Alcance o límites](#_Toc57619723)** [5](#_Toc57619723)

[**Análisis de Requerimientos** 5](#_Toc57619724)

[**DIAGRAMA ENTIDAD-RELACIÓN** 8](#_Toc57619725)

[**MODELO LÓGICO** 9](#_Toc57619726)

[**MODELO LÓGICO DEVELOPER** 10](#_Toc57619727)

[**MODELO FÍSICO** 11](#_Toc57619728)

[**ANEXO** 36](#_Toc57619729)

[**CONCLUSIÓN** 60](#_Toc57619730)

# **INTRODUCCIÓN**

A lo largo de la historia, la humanidad ha afrontado epidemias de diversas magnitudes, algunas devastadoras, como la llamada plaga de Atenas, en el año 430 a.C., en plena Segunda Guerra del Peloponeso. No se conoce la naturaleza de la plaga, pero según el historiador Tucídides, que la contrajo y sobrevivió, la ciudad de Pericles tardó 50 años en recuperarse y, durante esa época, la desesperanza fue tal que sus habitantes perdieron la fe en los dioses y en las leyes ante una inminente muerte. Según Tucídides, la plaga vino de Etiopía y pasó luego a Egipto, Libia y Grecia. Incluso los espartanos que asediaban Atenas se retiraron por el temor a la epidemia.

Los coronavirus son una extensa familia de virus que pueden causar enfermedades tanto en animales como en humanos. En los humanos, se sabe que varios coronavirus causan infecciones respiratorias que pueden ir desde el resfriado común hasta enfermedades más graves como el síndrome respiratorio de Oriente Medio (MERS) y el síndrome respiratorio agudo severo (SRAS). El coronavirus que se ha descubierto más recientemente causa la enfermedad por coronavirus COVID-19. La COVID-19 es la enfermedad infecciosa causada por el coronavirus que se ha descubierto más recientemente. Tanto el nuevo virus como la enfermedad eran desconocidos antes de que estallara el brote en Wuhan (China) en diciembre de 2019.

La pandemia de coronavirus COVID-19 es la crisis de salud global que define nuestro tiempo y el mayor desafío que hemos enfrentado desde la Segunda Guerra Mundial. Desde que su aparición en Asia a finales del año pasado, el virus ha llegado a cada continente, excepto a la Antártida. Los casos aumentan a diario en África, las Américas, y Europa.

Las herramientas que están más cercanas a tratar de resolver los problemas causados por este virus son las ecuaciones diferenciales y los modelos estocásticos, pero también la teoría de juegos, big data, machine learning y, en general, el análisis de datos juega ya un papel relevante, especialmente cuando queremos incluir los aspectos de las conductas sociales.

# **Misión de la Base de Datos**

Organizar correctamente la información de los pacientes infectados, con el fin de automatizar los procesos que se realizan diariamente en los hospitales del país.

# **Objetivos de la misión de la Base de Datos:**

**Objetivo General**

Proporcionar a los usuarios finales una visión abstracta de los datos acerca de los infectados por COVID-19 que existen en el país para así poder manejarlos y organizarlos de una manera más sencilla y con gran velocidad de respuesta.

**Objetivos Específicos**

* Servir de ayuda para el registro de casos de COVID-19 del país y todo lo necesario para un mayor control y realizar informes cada cierto tiempo.
* Poder tener un mejor control sobre los cuidados y seguimientos que se le realizarán a cada paciente, dependiendo de si se encuentra en el hospital o en aislamiento domiciliario.
* Dar a conocer en qué hospital están atendiendo a cada paciente y en qué sala se encuentra este para una mayor organización y para tener en cuenta en qué hospital pueden seguir admitiendo pacientes y qué cantidad de salas se encuentran disponibles para este fin.

# **Definición del Proyecto**

1. **Ámbito**

Los centros de salud, hospitales y demás instituciones se encuentran conscientes de que en estos momentos en los que se está viviendo una situación que nunca antes había ocurrido, es necesario realizar la mayor cantidad de pruebas posibles a todos los pacientes que se presentan con síntomas y proporcionar los resultados de manera rápida para así poder colocar a las personas que lo requieran en aislamiento dentro de sus hogares, o en casos más graves, atender a la mayor cantidad de pacientes posibles dentro del hospital, para que en el mejor de los casos, puedan tener una pronta recuperación y haya espacio para ofrecer una atención de calidad a cada uno de los nuevos pacientes.

Para tener un mejor control dentro de esta problemática, es necesario contar con una base de datos que identifique a los pacientes que se registran tanto para hacerse las pruebas como para internarse en el hospital, debido a que así se cuenta con información verídica acerca de cómo van aumentando los casos de infectados día tras día, qué tantas personas se recuperan y qué tantas no logran sobrevivir al Covid-19.

Esto ayudará a automatizar el trabajo que se encuentran realizando muchas personas para proporcionar los datos y la información correcta a toda la población, y ayudará a contar con un mejor orden y organización, además de que será mucho más fácil y rápido compartir estos datos a nivel interno en los centros de salud y hospitales.

1. **Alcance o límites**

* La base de datos solo es para el registro de infectados de nuestro país (Panamá).
* Sólo se registrarán los datos de pacientes infectados por COVID-19 confirmados.

# **Análisis de Requerimientos**

Para el análisis de los requerimientos necesarios para crear la base de datos utilizamos la técnica de investigación. Esta técnica nos pareció la más adecuada ya que nos parecía que investigar acerca del problema actual en las distintas páginas web o mediante las noticias nacionales, nos hacía ver cómo se iban relacionando los datos y cuáles eran los datos de más importancia que eran necesarios tener a mano en todo momento. Este método nos ayudó también ya que pudimos ser capaces de investigar cómo otros países estaban tratando los datos relacionados con las personas, siendo capaces de poder tener un control de la situación, por lo que mediante la investigación fuimos capaces de ver cuáles eran las medidas de administración de datos de las personas infectadas que ellos utilizaban y que nos serían muy útiles a la hora de desarrollar nuestra base de datos.

Mediante la investigación y el estudio del funcionamiento de la situación actual, se llegó a la conclusión de que los principales datos que son necesarios para la creación de esta base de datos son los siguientes:

* **Provincia**: Es necesario conocer la provincia en la cual se encuentran las personas que están infectadas con el virus para así poder llevar un control a escala nacional.
* **Distrito**: El conocer la cantidad de personas infectadas en cada distrito permite que las autoridades responsables puedan conocer con exactitud la situación de este para así poder tomar medidas para controlar la expansión del virus.
  + - **Corregimiento**: sabiendo el corregimiento en el que se encuentran las personas infectadas permite que se tenga un control más preciso por parte de las autoridades para así poder tomar decisiones adecuadas y mantener un control en estas zonas, aplicando medidas de seguridad y sanidad adecuadas para cada una de estas.
    - **Domicilio**: Es necesario saber dónde vive cada persona que está infectada para así poder llevar un control total y saber a su vez con cuantas personas se puede relacionar el infectado. Es preciso conocer la dirección del domicilio y cuántas personas viven en este.
    - **Persona**: Lo más importante en esta base de datos es saber qué persona está infectada ya que esta es la base para llevar un control preciso de la expansión del virus en el país. Se debe conocer la cédula, nombre, apellido y teléfono de la persona
    - **Enfermedades Previas**: Un dato importante para conocer la situación de las personas infectadas, es saber si estas cuentan con enfermedades previas porque con esto se puede realizar tratamientos distintos para cada persona.
    - **Sala**: Es necesario saber en qué sala está internado un paciente ya que así se tiene un control preciso del estado del infectado. Es importante conocer el número de camas que la sala posee y el piso del hospital en la que se encuentra.
    - **Hospital:** El hospital en el que se encuentra una persona infectada es relevante ya que así se mantiene un orden y a su vez son útiles para conocer la situación del hospital. Conocer el nombre, teléfono y dirección de este es muy importante.
    - **Doctor**: Los doctores encargados de realizar las distintas pruebas y llevar el control de los pacientes son importantes para esta base de daros, ya que así se puede conocer qué doctor está realizando las distintas pruebas y análisis de los pacientes. Se debe conocer la cédula, nombre y apellido de cada doctor que se encuentre laborando.
    - **Prueba**: Este dato es necesario para conocer con seguridad si una persona está infectada o no del virus. Es importante conocer el resultado de la prueba, y los síntomas que presentaba la persona.

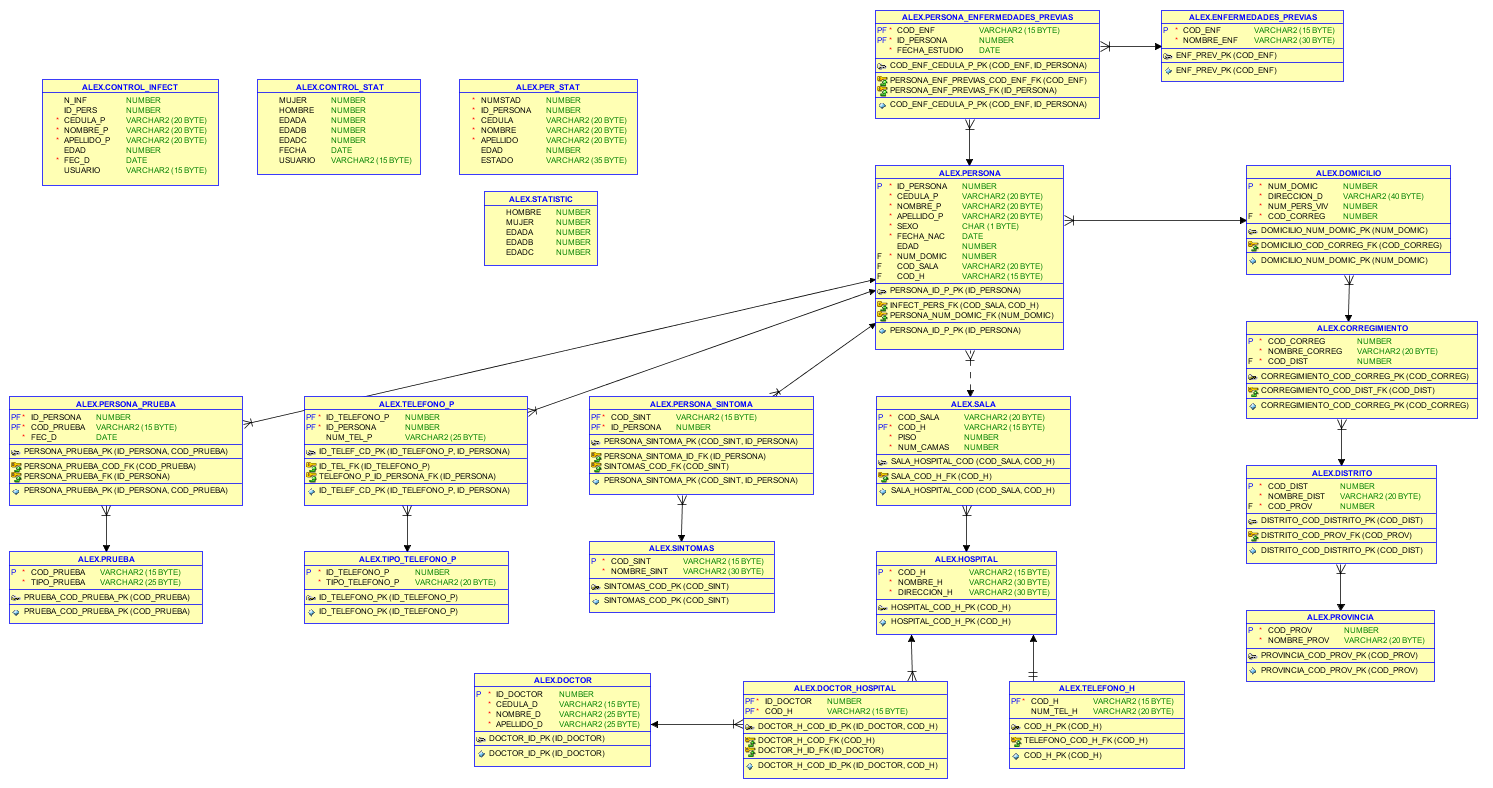
**Cómo utilizar o generar los datos**

* + - **Provincia**: Este dato se genera al conocer el distrito en el que vive la persona infectada.
    - **Distrito**: Este dato se genera al conocer el corregimiento en el que se encuentra el domicilio de la persona infectada.
    - **Corregimiento**: Este dato se genera a partir de la localización del domicilio de la persona, sabiendo en que calle o barrio esta vive.
    - **Domicilio**: Este dato se genera al hablar con la persona infectada y tomar su información personal.
    - **Persona**: Este dato se genera a medida que más personas se vayan realizando pruebas para conocer si poseen el virus o no.
    - **Enfermedades Previas**: Este dato se genera al hablar con la persona infectada y tomar su información personal.
    - **Sala**: Este dato se genera al saber en qué situación de salud se encuentra la persona infectada y el hospital al que fue llevado.
    - **Hospital:** Este dato se genera al conocer en qué sala se ha trasladado la persona infectada.
    - **Doctor**: Este dato se obtiene al hablar con el administrador del hospital y que nos informe de la lista de doctores disponibles en el hospital.
    - **Prueba**: Este dato se obtiene una vez que la persona decide ir al hospital y que ahí los doctores le realicen la prueba.

# **Diagrama Descripción generada automáticamenteDIAGRAMA ENTIDAD-RELACIÓN**

# **Diagrama, Dibujo de ingeniería Descripción generada automáticamenteMODELO LÓGICO**

# **MODELO LÓGICO DEVELOPER**



# **MODELO FÍSICO**

**--TABLAS**

create table Enfermedades\_previas (

Cod\_enf varchar2 (15),

constraint Enf\_prev\_pk primary key (Cod\_enf),

nombre\_enf varchar2 (30) not null

);

create table Prueba(

Cod\_prueba varchar2 (15),

constraint Prueba\_Cod\_prueba\_pk primary key (Cod\_prueba),

tipo\_prueba varchar2 (25) not null

);

create table Hospital(

Cod\_h varchar2 (15),

constraint Hospital\_Cod\_h\_pk primary key (Cod\_h),

nombre\_h varchar2 (30) not null,

direccion\_h varchar2 (30) not null

);

create table Doctor(

id\_doctor number,

Cedula\_d varchar2 (15) not null,

constraint Doctor\_id\_pk primary key (id\_doctor),

nombre\_d varchar2 (25) not null,

apellido\_d varchar2 (25) not null

);

create table Sintomas (

Cod\_sint varchar2 (15),

constraint Sintomas\_cod\_pk primary key (Cod\_sint),

nombre\_sint varchar2 (30) not null

);

create table Provincia (

Cod\_prov number,

constraint Provincia\_Cod\_prov\_pk primary key (Cod\_prov),

nombre\_prov varchar2 (20) not null

);

create table Distrito(

Cod\_dist number,

constraint Distrito\_Cod\_distrito\_pk primary key (Cod\_dist),

nombre\_dist varchar2 (20) not null,

Cod\_prov number not null,

constraint Distrito\_Cod\_prov\_fk foreign key (Cod\_prov) references Provincia (Cod\_prov)

);

create table Corregimiento(

Cod\_correg number,

constraint Corregimiento\_Cod\_correg\_pk primary key (Cod\_correg),

nombre\_correg varchar2 (20) not null,

Cod\_dist number not null,

constraint Corregimiento\_Cod\_dist\_fk foreign key (Cod\_dist) references Distrito (Cod\_dist)

);

create table Domicilio (

Num\_domic number,

constraint Domicilio\_Num\_domic\_pk primary key (Num\_domic),

direccion\_d varchar2 (40) not null,

num\_pers\_viv number not null,

Cod\_correg number not null,

constraint Domicilio\_Cod\_correg\_fk foreign key (Cod\_correg) references Corregimiento (Cod\_correg)

);

create table Sala(

Cod\_sala varchar2 (20),

Cod\_h varchar2 (15),

constraint Sala\_Cod\_h\_fk foreign key (Cod\_h) references Hospital (Cod\_h),

constraint Sala\_Hospital\_cod primary key (Cod\_sala, Cod\_h),

piso number not null,

num\_camas number not null

);

create table Persona(

id\_persona number,

Cedula\_p varchar2 (20) not null,

constraint Persona\_id\_p\_pk primary key (id\_persona),

nombre\_p varchar2 (20) not null,

apellido\_p varchar2 (20) not null,

sexo char not null,

fecha\_nac date not null,

edad number,

Num\_domic number not null,

constraint Persona\_Num\_domic\_fk foreign key (Num\_domic) references Domicilio (Num\_domic),

Cod\_sala varchar2 (20),

Cod\_h varchar2 (15),

constraint infect\_pers\_fk foreign key (Cod\_sala, Cod\_h) references Sala (Cod\_sala, Cod\_h)

);

create table Doctor\_Hospital (

id\_doctor number,

constraint Doctor\_h\_id\_fk foreign key (id\_doctor) references Doctor (id\_doctor),

Cod\_h varchar2 (15),

constraint Doctor\_h\_cod\_fk foreign key (Cod\_h) references Hospital (Cod\_h),

constraint Doctor\_h\_cod\_id\_pk primary key (id\_doctor, Cod\_h)

);

create table Persona\_Sintoma(

Cod\_sint varchar2 (15),

constraint Sintomas\_cod\_fk foreign key (Cod\_sint) references Sintomas (Cod\_sint),

id\_persona number,

constraint Persona\_Sintoma\_id\_fk foreign key (id\_persona) references Persona (id\_persona),

constraint Persona\_sintoma\_pk primary key (Cod\_sint, id\_persona)

);

create table Persona\_Enfermedades\_Previas(

Cod\_enf varchar2 (15) not null,

constraint Persona\_Enf\_Previas\_Cod\_enf\_fk foreign key (Cod\_enf) references Enfermedades\_previas (Cod\_enf),

id\_persona number not null,

constraint Persona\_Enf\_Previas\_fk foreign key (id\_persona) references Persona (id\_persona),

constraint Cod\_enf\_Cedula\_p\_pk primary key (Cod\_enf, id\_persona),

fecha\_estudio date not null

);

create table Persona\_Prueba(

id\_persona number,

constraint Persona\_prueba\_fk foreign key (id\_persona) references Persona (id\_persona),

Cod\_prueba varchar2 (15),

constraint Persona\_Prueba\_cod\_fk foreign key (Cod\_prueba) references Prueba (Cod\_prueba),

constraint Persona\_prueba\_pk primary key (id\_persona, Cod\_prueba),

fec\_d date not null

);

create table Telefono\_h(

Cod\_h varchar2(15),

constraint Telefono\_Cod\_h\_fk foreign key (Cod\_h)

references Hospital (Cod\_h),

constraint cod\_h\_pk primary key (Cod\_h),

num\_tel\_h varchar2 (20)

);

create table Tipo\_Telefono\_p (

ID\_telefono\_p number not null,

tipo\_telefono\_p varchar2 (20) not null,

constraint ID\_telefono\_pk primary key (ID\_telefono\_p)

);

create table Telefono\_p(

ID\_telefono\_p number not null,

id\_persona number,

constraint ID\_tel\_fk foreign key (ID\_telefono\_p)

references Tipo\_Telefono\_p (ID\_telefono\_p),

constraint Telefono\_p\_id\_persona\_fk foreign key (id\_persona)

references Persona (id\_persona),

constraint ID\_telef\_cd\_pk primary key (ID\_telefono\_p, id\_persona),

num\_tel\_p varchar2(25)

);

create table per\_stat(

numstad number not null,

id\_persona number not null,

cedula varchar2(20) not null,

nombre varchar2(20) not null,

apellido varchar2(20) not null,

edad number,

estado varchar2(35)

);

Create table Control\_Infect(

N\_inf number,

Id\_pers number,

cedula\_p varchar2 (20) not null,

nombre\_p varchar2 (20) not null,

apellido\_p varchar2 (20) not null,

edad number,

fec\_d date not null,

usuario varchar2(15)

);

create table statistic(

hombre number,

mujer number,

edada number,

edadb number,

edadc number

);

Create table Control\_stat(

mujer number,

hombre number,

edada number,

edadb number,

edadc number,

fecha date,

usuario varchar2(15)

);

**--SECUENCIAS**

CREATE SEQUENCE seq\_doc

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE seq\_pers

START WITH 1

INCREMENT BY 1;

create sequence trig\_per

start with 1

increment by 1;

create sequence trig\_stad

start with 1

increment by 1;

create sequence stat

start with 1

increment by 1;

**--FUNCIONES**

**/\* Funcion que calcula la edad de los pacientes\*/**

CREATE OR REPLACE FUNCTION calc\_edad (fecha\_nac PERSONA.fecha\_nac%TYPE)

RETURN NUMBER AS

BEGIN

RETURN FLOOR (months\_between (sysdate, fecha\_nac)/12);

END calc\_edad;

/

**/\* Funcion que calcula el porcentaje de pacientes que son hombres\*/**

CREATE OR REPLACE FUNCTION stat\_male

RETURN NUMBER AS

v\_hom NUMBER;

v\_total NUMBER;

v\_resp NUMBER;

BEGIN

SELECT count(sexo) INTO v\_hom FROM Persona

WHERE sexo='M';

SELECT count(sexo) INTO v\_total FROM Persona;

v\_resp:=(v\_hom/v\_total)\*100;

RETURN v\_resp;

END stat\_male;

/

**/\* Funcion que calcula el porcentaje de pacientes que son mujeres\*/**

CREATE OR REPLACE FUNCTION stat\_female

RETURN NUMBER AS

v\_muj NUMBER;

v\_total NUMBER;

v\_resp NUMBER;

BEGIN

SELECT count(sexo) INTO v\_muj FROM Persona

WHERE sexo='F';

SELECT count(sexo) INTO v\_total FROM Persona;

v\_resp:=(v\_muj/v\_total)\*100;

RETURN v\_resp;

END stat\_female;

/

**/\* Funcion que calcula el tipo de riesgo en los pacientes segun su edad\*/**

CREATE OR REPLACE FUNCTION population(

p\_edad persona.edad%TYPE)

RETURN VARCHAR2 AS

BEGIN

IF p\_edad>=60 THEN

RETURN 'Riesgo Alto';

ELSIF p\_edad >=30 THEN

RETURN 'Riesgo Regular';

ELSE

RETURN 'Riesgo Bajo';

END IF;

END population;

/

**/\* Funcion que calcula el porcentaje de pacientes entre 30 y 60 años\*/**

CREATE OR REPLACE FUNCTION stat\_edadb

RETURN NUMBER AS

v\_edadc NUMBER;

v\_total NUMBER;

v\_resp NUMBER;

BEGIN

SELECT count(edad) INTO v\_edadc FROM Persona

WHERE edad>=30 AND edad<60;

SELECT count(\*) INTO v\_total FROM Persona;

v\_resp:=(v\_edadc/v\_total)\*100;

RETURN v\_resp;

END stat\_edadb;

/

**/\* Funcion** **que calcula el porcentaje de pacientes menores a 30 años\*/**

CREATE OR REPLACE FUNCTION stat\_edadc

RETURN NUMBER AS

v\_edadc NUMBER;

v\_total NUMBER;

v\_resp NUMBER;

BEGIN

SELECT count(edad) INTO v\_edadc FROM Persona

WHERE edad<30;

SELECT count(id\_persona) INTO v\_total FROM Persona;

v\_resp:=(v\_edadc/v\_total)\*100;

RETURN v\_resp;

END stat\_edadc;

/

**/\* Funcion** **que calcula el porcentaje de pacientes mayores de 60 años\*/**

CREATE OR REPLACE FUNCTION stat\_edada

RETURN NUMBER AS

v\_edadc NUMBER;

v\_total NUMBER;

v\_resp NUMBER;

BEGIN

SELECT count(edad) INTO v\_edadc FROM Persona

WHERE edad>=60;

SELECT count(id\_persona) INTO v\_total FROM Persona;

v\_resp:=(v\_edadc/v\_total)\*100;

RETURN v\_resp;

END stat\_edada;

/

**--TRIGGERS**

**/\* Trigger que registra un control de infectados al ser ingresados en la BD\*/**

CREATE OR REPLACE TRIGGER per\_pac

BEFORE UPDATE OR DELETE OR INSERT ON Persona

FOR EACH ROW

DECLARE

v\_Control CHAR(1);

BEGIN

IF UPDATING THEN

v\_Control := 'U';

INSERT INTO Control\_Infect (n\_inf, id\_pers, cedula\_p, nombre\_p, apellido\_p, edad, fec\_d, usuario)

VALUES (trig\_per.nextval, :NEW.id\_persona, :NEW.cedula\_p, :NEW.nombre\_p, :new.apellido\_p, :new.edad, sysdate, user);

ELSIF INSERTING THEN

v\_Control := 'I';

INSERT INTO Control\_Infect (n\_inf, id\_pers, cedula\_p, nombre\_p, apellido\_p, edad, fec\_d, usuario)

VALUES (trig\_per.nextval, :NEW.id\_persona, :NEW.cedula\_p, :NEW.nombre\_p, :new.apellido\_p, :new.edad, sysdate, user);

ELSIF DELETING THEN

v\_Control := 'D';

INSERT INTO Control\_Infect (n\_inf, id\_pers, cedula\_p, nombre\_p, apellido\_p, edad, fec\_d, usuario)

VALUES (trig\_per.nextval, :NEW.id\_persona, :NEW.cedula\_p, :NEW.nombre\_p, :new.apellido\_p, :new.edad, sysdate, user);

END IF;

END per\_pac;

/

**/\* Trigger que tiene un control entre los porcentajes segun sexo y edad de los pacientes\*/**

CREATE OR REPLACE TRIGGER upd\_static

BEFORE INSERT OR UPDATE ON Statistic

FOR EACH ROW

DECLARE

v\_Control CHAR(1);

BEGIN

IF UPDATING THEN

v\_Control := 'U';

INSERT INTO control\_stat

values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);

ELSIF INSERTING THEN

v\_Control := 'I';

INSERT INTO control\_stat

values (:NEW.mujer, :NEW.hombre, :NEW.edada, :NEW.edadb, :NEW.edadc, sysdate, user);

END IF;

END upd\_static;

/

**--PROCEDIMIENTOS**

**/\* Procedimiento de insercion en tabla enfermedades previas\*/**

CREATE OR REPLACE PROCEDURE add\_enfermedad (

p\_code enfermedades\_previas.cod\_enf%TYPE,

p\_nombrenf enfermedades\_previas.nombre\_enf%TYPE

) AS

BEGIN

INSERT INTO Enfermedades\_previas (cod\_enf, nombre\_enf)

values (p\_code, p\_nombrenf);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_enfermedad;

/

**/\* Procedimiento de insercion en tabla prueba\*/**

CREATE OR REPLACE PROCEDURE add\_prueba (

p\_codpb prueba.cod\_prueba%TYPE,

p\_tipop prueba.tipo\_prueba%TYPE

) AS

BEGIN

INSERT INTO Prueba (cod\_prueba, tipo\_prueba)

values (p\_codpb, p\_tipop);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_prueba;

/

**/\* Procedimiento de insercion en tabla hospital\*/**

CREATE OR REPLACE PROCEDURE add\_hospital (

p\_codh hospital.cod\_h%TYPE,

p\_nombreh hospital.nombre\_h%TYPE,

p\_direccionh hospital.direccion\_h%TYPE

) AS

BEGIN

INSERT INTO Hospital (cod\_h, nombre\_h, direccion\_h)

values (p\_codh, p\_nombreh, p\_direccionh);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_hospital;

/

**/\* Procedimiento de insercion en tabla doctor\*/**

CREATE OR REPLACE PROCEDURE add\_doctor (

p\_cedulad doctor.cedula\_d%TYPE,

p\_nombred doctor.nombre\_d%TYPE,

p\_apellidod doctor.apellido\_d%TYPE

) AS

BEGIN

INSERT INTO Doctor (id\_doctor, cedula\_d, nombre\_d, apellido\_d)

values (seq\_doc.nextval, p\_cedulad, p\_nombred, p\_apellidod);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_doctor;

/

**/\* Procedimiento de insercion en tabla sintomas\*/**

CREATE OR REPLACE PROCEDURE add\_sintoma (

p\_cods sintomas.cod\_sint%TYPE,

p\_nombres sintomas.nombre\_sint%TYPE

) AS

BEGIN

INSERT INTO Sintomas (cod\_sint, nombre\_sint)

values (p\_cods, p\_nombres);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_sintoma;

/

**/\* Procedimiento de insercion en tabla provincia\*/**

CREATE OR REPLACE PROCEDURE add\_provincia (

p\_codp provincia.cod\_prov%TYPE,

p\_nombrep provincia.nombre\_prov%TYPE

) AS

BEGIN

INSERT INTO Provincia (cod\_prov, nombre\_prov)

values (p\_codp, p\_nombrep);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_provincia;

/

**/\* Procedimiento de insercion en tabla distrito\*/**

CREATE OR REPLACE PROCEDURE add\_distrito (

p\_coddis distrito.cod\_dist%TYPE,

p\_nombredis distrito.nombre\_dist%TYPE,

p\_codprov distrito.cod\_prov%TYPE

) AS

BEGIN

INSERT INTO Distrito (cod\_dist, nombre\_dist, cod\_prov)

values (p\_coddis, p\_nombredis, p\_codprov);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_distrito;

/

**/\* Procedimiento de insercion en tabla corregimiento\*/**

CREATE OR REPLACE PROCEDURE add\_correg (

p\_codcorr corregimiento.cod\_correg%TYPE,

p\_nombreco corregimiento.nombre\_correg%TYPE,

p\_codis corregimiento.cod\_dist%TYPE

) AS

BEGIN

INSERT INTO Corregimiento (cod\_correg, nombre\_correg, cod\_dist)

values (p\_codcorr, p\_nombreco, p\_codis);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_correg;

/

**/\* Procedimiento de insercion en tabla domicilio\*/**

CREATE OR REPLACE PROCEDURE add\_domicilio (

p\_numdom domicilio.num\_domic%TYPE,

p\_direcciond domicilio.direccion\_d%TYPE,

p\_numpers domicilio.num\_pers\_viv%TYPE,

p\_codcor domicilio.cod\_correg%TYPE

) AS

BEGIN

INSERT INTO Domicilio (num\_domic, direccion\_d, num\_pers\_viv, cod\_correg)

values (p\_numdom, p\_direcciond, p\_numpers, p\_codcor);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_domicilio;

/

**/\* Procedimiento de insercion en tabla persona\*/**

CREATE OR REPLACE PROCEDURE add\_persona (

p\_cedulap persona.cedula\_p%TYPE,

p\_nombrep persona.nombre\_p%TYPE,

p\_apellidop persona.apellido\_p%TYPE,

p\_sexo persona.sexo%TYPE,

p\_fechan persona.fecha\_nac%TYPE,

p\_numdom persona.num\_domic%TYPE,

p\_codsala persona.cod\_sala%TYPE,

p\_codh persona.cod\_h%TYPE

) AS

p\_edad persona.edad%TYPE;

BEGIN

p\_edad:=calc\_edad(p\_fechan);

INSERT INTO Persona (id\_persona, cedula\_p, nombre\_p, apellido\_p, sexo, fecha\_nac, edad, num\_domic, cod\_sala, cod\_h)

values (seq\_pers.nextval, p\_cedulap, p\_nombrep, p\_apellidop, p\_sexo, p\_fechan, p\_edad, p\_numdom, p\_codsala, p\_codh);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_persona;

/

**/\* Procedimiento de insercion en tabla doctor\_hospital\*/**

CREATE OR REPLACE PROCEDURE Add\_doctor\_hospital (

p\_id\_doctor doctor\_hospital.id\_doctor%TYPE,

p\_cod\_h doctor\_hospital.cod\_h%TYPE

) AS

BEGIN

INSERT INTO doctor\_hospital (id\_doctor, cod\_h)

values (p\_id\_doctor, p\_cod\_h);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END Add\_doctor\_hospital;

/

**/\* Procedimiento de insercion en tabla teléfono\_h\*/**

CREATE OR REPLACE PROCEDURE add\_telefono\_h (

p\_cod\_h telefono\_h.cod\_h%TYPE,

p\_num\_tel\_h telefono\_h.num\_tel\_h%TYPE

) AS

BEGIN

INSERT INTO telefono\_h (cod\_h, num\_tel\_h)

values (p\_cod\_h, p\_num\_tel\_h);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_telefono\_h;

/

**/\* Procedimiento de insercion en tabla sala\*/**

CREATE OR REPLACE PROCEDURE add\_sala (

p\_cod\_sala sala.cod\_sala%TYPE,

p\_cod\_h sala.cod\_h%TYPE,

p\_piso sala.piso%TYPE,

p\_num\_camas sala.num\_camas%TYPE

) AS

BEGIN

INSERT INTO sala (cod\_sala, cod\_h, piso, num\_camas)

values (p\_cod\_sala, p\_cod\_h, p\_piso, p\_num\_camas);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_sala;

/

**/\* Procedimiento de insercion en tabla tipo\_telefono\_p\*/**

CREATE OR REPLACE PROCEDURE add\_tipo\_telefono\_p (

p\_id\_telefono\_p IN tipo\_telefono\_p.id\_telefono\_p%TYPE,

p\_tipo\_telefono\_p IN tipo\_telefono\_p.tipo\_telefono\_p%TYPE

) AS

BEGIN

INSERT INTO Tipo\_Telefono\_p (id\_telefono\_p, tipo\_telefono\_p)

VALUES (p\_id\_telefono\_p, p\_tipo\_telefono\_p);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END add\_tipo\_telefono\_p;

/

**/\* Procedimiento de insercion en tabla teléfono\_p\*/**

CREATE OR REPLACE PROCEDURE add\_telefono\_p (

p\_id\_telefono\_p IN telefono\_p.id\_telefono\_p%TYPE,

p\_id\_persona IN telefono\_p.id\_persona%TYPE,

p\_num\_tel\_p IN telefono\_p.num\_tel\_p%TYPE

) AS

BEGIN

INSERT INTO telefono\_p(id\_telefono\_p, id\_persona, num\_tel\_p)

VALUES (p\_id\_telefono\_p, p\_id\_persona, p\_num\_tel\_p);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró la tabla');

COMMIT;

END add\_telefono\_p;

/

**/\* Procedimiento de insercion en tabla persona\_sintoma\*/**

CREATE OR REPLACE PROCEDURE add\_persona\_sintoma (

p\_cod\_sint persona\_sintoma.cod\_sint%TYPE,

p\_id\_persona persona\_sintoma.id\_persona%TYPE

) AS

BEGIN

INSERT INTO Persona\_sintoma (cod\_sint, id\_persona)

Values (p\_cod\_sint, p\_id\_persona);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_persona\_sintoma;

/

**/\* Procedimiento de insercion en tabla persona\_prueba\*/**

CREATE OR REPLACE PROCEDURE add\_persona\_prueba (

p\_id\_persona persona\_prueba.id\_persona%TYPE,

p\_cod\_prueba persona\_prueba.cod\_prueba%TYPE,

p\_fec\_d persona\_prueba.fec\_d%TYPE

) AS

BEGIN

INSERT INTO Persona\_prueba (id\_persona, cod\_prueba, fec\_d)

values (p\_id\_persona, p\_cod\_prueba, p\_fec\_d);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_persona\_prueba;

/

**/\* Procedimiento de insercion en tabla persona\_enfermedades\_previas\*/**

CREATE OR REPLACE PROCEDURE add\_persona\_ep (

p\_cod\_enf persona\_enfermedades\_previas.cod\_enf%TYPE,

p\_id\_persona persona\_enfermedades\_previas.id\_persona%TYPE,

p\_fecha\_estudio persona\_enfermedades\_previas.fecha\_estudio%TYPE

) AS

BEGIN

INSERT INTO Persona\_enfermedades\_previas (cod\_enf, id\_persona, fecha\_estudio)

values (p\_cod\_enf, p\_id\_persona, p\_fecha\_estudio);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE ('Ya existe este registro');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE ('No se insertaron los datos');

COMMIT;

END add\_persona\_ep;

/

**/\* Procedimiento que actualiza el estado en los pacientes segun su edad\*/**

CREATE OR REPLACE PROCEDURE act\_age

AS

CURSOR c\_edad IS

SELECT id\_persona, cedula\_p, nombre\_p, apellido\_p, edad

FROM Persona;

v\_idp persona.id\_persona%TYPE;

v\_nombre persona.nombre\_p%TYPE;

v\_apellido persona.apellido\_p%TYPE;

v\_edad persona.edad%TYPE;

v\_cedula persona.cedula\_p%TYPE;

v\_funcion per\_stat.estado%TYPE;

BEGIN

OPEN c\_edad;

LOOP

FETCH c\_edad INTO v\_idp, v\_cedula, v\_nombre, v\_apellido, v\_edad;

v\_funcion:=population(v\_edad);

INSERT INTO per\_stat (numstad, id\_persona, cedula, nombre, apellido, edad, estado)

values (stat.nextval,v\_idp, v\_cedula, v\_nombre, v\_apellido, v\_edad, v\_funcion);

EXIT WHEN c\_edad%NOTFOUND;

END LOOP;

CLOSE c\_edad;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontró los datos');

COMMIT;

END act\_age;

/

**/\* Procedimiento que actualiza los porcentajes de estadística por sexo y edad de los pacientes\*/**

create or replace PROCEDURE upd\_stat

AS

p\_hom statistic.hombre%TYPE;

p\_muj statistic.mujer%TYPE;

p\_edada statistic.edada%TYPE;

p\_edadb statistic.edadb%TYPE;

p\_edadc statistic.edadc%TYPE;

BEGIN

p\_hom:=stat\_male;

p\_muj:=stat\_female;

p\_edada:=stat\_edada;

p\_edadb:=stat\_edadb;

p\_edadc:=stat\_edadc;

INSERT INTO Statistic

values (p\_hom, p\_muj,p\_edada,p\_edadb,p\_edadc);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro los datos');

COMMIT;

END upd\_stat;

/

**--INVOCACIONES**

BEGIN

add\_enfermedad ('HIP', 'Hipertensión');

add\_enfermedad ('CAN', 'Cáncer');

add\_enfermedad ('DIA', 'Diabetes');

add\_enfermedad ('EPU', 'Enfermedades Pulmonares');

add\_enfermedad ('ECA', 'Enfermedades Cardiacas');

END;

/

BEGIN

add\_prueba ('S', 'Prueba Serológica');

add\_prueba ('H', 'Prueba por Hisopado');

END;

/

BEGIN

add\_hospital ('8-H1', 'Hospital Santo Tomás', 'Calle 37 Este');

add\_hospital ('8-H2', 'Clínica Hospital San Fernando', 'Vía España');

add\_hospital ('8-H3', 'Hospital Santa Fe', 'Ave. Simón Bolívar');

add\_hospital ('8-H4', 'Hospital Modular', 'Albrook');

add\_hospital ('8-H5', 'Complejo Dr. Arnulfo A. Madrid', 'Ave. Simón Bolívar');

add\_hospital ('3-H1', 'Hospital Manuel A. Guerrero', 'Calle 11');

add\_hospital ('3-H2', 'Hospital Colón 4 Altos', '4 Altos');

add\_hospital ('4-H1', 'Hospital Chiriquí', 'Ave. 3ra y Calle Central');

add\_hospital ('4-H2', 'Hospital Dr. Rafael Hernández', 'San Pablo Viejo');

add\_hospital ('13-H1', 'Hospital Nicolás A. Solano', 'La Chorrera');

add\_hospital ('13-H2', 'Clínica Hospital Panamericano', 'Calle Santa Rita');

add\_hospital ('6-H1', 'Hospital Gustavo N. Collado', 'Ave. Ranaldo Camarano');

add\_hospital ('7-H1', 'Hospital Anita Moreno', 'Ave. Dr. Belisario Porras');

add\_hospital ('1-H1', 'Hospital Guillermo S. Borbón', 'Isla Colon');

add\_hospital ('1-H2', 'Isla Colon Regional Hospital', 'Ave. H Norte');

add\_hospital ('9-H1', 'Hospital Dr. Luis Fábrega', 'Santiago');

add\_hospital ('9-H2', 'Policlínica Dr. Horacio Gómez', 'Santiago');

add\_hospital ('5-H1', 'Hospital de Metetí', 'Metetí');

add\_hospital ('2-H1', 'Hospital Aquilino Tejeira', 'Penonomé');

add\_hospital ('2-H2', 'Hospital Dr. Rafael Estévez', 'Aguadulce');

add\_hospital ('12-H1', 'Hospital de San Pedrito', 'Ngöbe-Buglé');

add\_hospital ('10-H1', 'Hospital Marvel Iglesia', 'Guna Yala');

END;

/

BEGIN

add\_doctor ('8-365-873', 'Tatiana', 'Ojo');

add\_doctor ('7-546-654', 'Vivian', 'Aguilar');

add\_doctor ('8-234-432', 'Denisse', 'Thorne');

add\_doctor ('8-465-543', 'Reynaldo', 'Castillero');

add\_doctor ('3-65-673', 'Giselle', 'Balmaceda');

add\_doctor ('4-100-657', 'Moisés', 'De Sedas');

add\_doctor ('1-432-532', 'Jeannette', 'Johnson');

add\_doctor ('2-348-43', 'Tommy', 'Liao');

add\_doctor ('9-543-67', 'Andrés', 'Soto');

add\_doctor ('6-234-6754', 'José', 'Lorenzo');

add\_doctor ('5-123-890', 'Gustavo', 'Santamaría');

add\_doctor ('3-234-567', 'Sunshine', 'Del Cid');

add\_doctor ('9-321-789', 'Raúl', 'Mendoza');

add\_doctor ('8-332-654', 'Italys', 'Ledezma');

add\_doctor ('7-654-2134', 'Daniel', 'Araúz');

END;

/

BEGIN

add\_sintoma ('F', 'Fiebre');

add\_sintoma ('TS', 'Tos seca');

add\_sintoma ('C', 'Cansancio');

add\_sintoma ('DG', 'Dolor de garganta');

add\_sintoma ('DC', 'Dolor de cabeza');

add\_sintoma ('D', 'Diarrea');

add\_sintoma ('DM', 'Molestias y dolores');

add\_sintoma ('CO', 'Conjuntivitis');

add\_sintoma ('DR', 'Dificultad para respirar');

add\_sintoma ('E', 'Erupciones cutáneas');

add\_sintoma ('IH', 'Incápacidad para hablar');

add\_sintoma ('P', 'Pérdida de olfato y gusto');

END;

/

BEGIN

add\_provincia (1, 'Bocas del Toro');

add\_provincia (2, 'Coclé');

add\_provincia (3, 'Colón');

add\_provincia (4, 'Chiriquí');

add\_provincia (5, 'Darién');

add\_provincia (6, 'Herrera');

add\_provincia (7, 'Los Santos');

add\_provincia (8, 'Panamá');

add\_provincia (9, 'Veraguas');

add\_provincia (10, 'Guna Yala');

add\_provincia (11, 'Emberá Wounaan');

add\_provincia (12, 'Ngäbe-buglé');

add\_provincia (13, 'Panamá Oeste');

add\_provincia(14, 'Madugandí');

add\_provincia (15, 'Wargandí');

END;

/

BEGIN

add\_distrito (0101, 'Almirante', 1);

add\_distrito (0102, 'Bocas del Toro', 1);

add\_distrito (0103, 'Changuinola', 1);

add\_distrito (0201, 'Penonomé', 2);

add\_distrito (0202, 'Aguadulce', 2);

add\_distrito (0203, 'Antón', 2);

add\_distrito (0301, 'Colón', 3);

add\_distrito (0302, 'Chagres', 3);

add\_distrito (0303, 'Portobelo', 3);

add\_distrito (0401, 'David', 4);

add\_distrito (0402, 'San Félix', 4);

add\_distrito (0403, 'Barú', 4);

add\_distrito (0501, 'Chepigana', 5);

add\_distrito (0502, 'Pinogana', 5);

add\_distrito (0503, 'Santa Fe', 5);

add\_distrito (0601, 'Chitré', 6);

add\_distrito (0602, 'Ocú', 6);

add\_distrito (0603, 'Las Minas', 6);

add\_distrito (0701, 'Las Tablas', 7);

add\_distrito (0702, 'Tonosí', 7);

add\_distrito (0703, 'Guararé', 7);

add\_distrito (0801, 'San Miguelito', 8);

add\_distrito (0802, 'Panamá', 8);

add\_distrito (0803, 'Balboa', 8);

add\_distrito (0901, 'Santiago', 9);

add\_distrito (0902, 'Atalaya', 9);

add\_distrito (0903, 'Calobre', 9);

add\_distrito (1101, 'Cémaco', 11);

add\_distrito (1102, 'Sambú', 11);

add\_distrito (1201, 'Besikó', 12);

add\_distrito (1202, 'Jirondai', 12);

add\_distrito (1203, 'Kankintú', 12);

add\_distrito (1301, 'Arraiján', 13);

add\_distrito (1302, 'La Chorrera', 13);

add\_distrito (1303, 'San Carlos', 13);

END;

/

BEGIN

add\_correg (010101, 'Barrio Francés', 0101);

add\_correg (010201, 'Bastimentos', 0102);

add\_correg (010301, 'El Silencio', 0103);

add\_correg (020101, 'Río Grande', 0201);

add\_correg (020201, 'Pocrí', 0202);

add\_correg (020301, 'El Valle', 0203);

add\_correg (030101, 'Cristóbal', 0301);

add\_correg (030201, 'Piña', 0302);

add\_correg (030301, 'Cacique', 0303);

add\_correg (040101, 'San Carlos', 0401);

add\_correg (040201, 'Las Lajas', 0402);

add\_correg (040301, 'Puerto Armuelles', 0403);

add\_correg (050101, 'La Palma', 0501);

add\_correg (050201, 'Yaviza', 0502);

add\_correg (050301, 'Agua Fría', 0503);

add\_correg (060101, 'Monagrillo', 0601);

add\_correg (060201, 'Llano Grande', 0602);

add\_correg (060301, 'Chumical', 0603);

add\_correg (070101, 'Valle Rico', 0701);

add\_correg (070201, 'Cambutal', 0702);

add\_correg (070301, 'Llano Abajo', 0703);

add\_correg (080101, 'Omar Torrijos', 0801);

add\_correg (080201, 'Don Bosco', 0802);

add\_correg (080301, 'San Miguel', 0803);

add\_correg (090101, 'Urracá', 0901);

add\_correg (090201, 'San Antonio', 0902);

add\_correg (090301, 'La Yeguada', 0903);

add\_correg (110101, 'Lajas Blancas', 1101);

add\_correg (110201, 'Río Sabalo', 1102);

add\_correg (120101, 'Cerro Banco', 1201);

add\_correg (120201, 'Man Creek', 1202);

add\_correg (120301, 'Piedra Roja', 1203);

add\_correg (130101, 'Vista Alegre', 1301);

add\_correg (130201, 'El Coco', 1302);

add\_correg (130301, 'Las Uvas', 1303);

END;

/

BEGIN

add\_domicilio (1803, 'Ave. Olmedo Solé', 7, 10101);

add\_domicilio (1567, 'Calle Sea Wall', 4, 10201);

add\_domicilio (1726, 'Sendero Caimán', 9, 10301);

add\_domicilio (6489, 'Calle Verde', 3, 20101);

add\_domicilio (8456, 'Calle Abajo', 3, 20201);

add\_domicilio (8623, 'Ave. Castor', 6, 20301);

add\_domicilio (560, 'Calle Fitzpatrick', 5, 30101);

add\_domicilio (23, 'Calle Sandía', 3, 30201);

add\_domicilio (789, 'Calle Indio', 5, 30301);

add\_domicilio (85, 'Calle América', 4, 40101);

add\_domicilio (236, 'Rincón Largo', 7, 40201);

add\_domicilio (889, 'Ave. Barco', 2, 40301);

add\_domicilio (1582, 'Calle Coco', 4, 50101);

add\_domicilio (432, 'Ave. Candelario', 8, 50201);

add\_domicilio (976, 'Sector Hielo', 3, 50301);

add\_domicilio (25, 'Calle Mercado', 6, 60101);

add\_domicilio (4, 'Calle Arriba', 1, 60201);

add\_domicilio (57, 'Ave. Montañosa', 4, 60301);

add\_domicilio (478, 'Calle Mala', 9, 70101);

add\_domicilio (245, 'Calle Dulce', 3, 70201);

add\_domicilio (267, 'Calle San Francisco', 2, 70301);

add\_domicilio (332, 'San Isidro Sector B', 5, 80101);

add\_domicilio (442, 'Calle F', 3, 80201);

add\_domicilio (90, 'Ave. Roja', 5, 80301);

add\_domicilio (1478, 'Calle Centavo', 3, 90101);

add\_domicilio (898, 'Calle Santa', 2, 90201);

add\_domicilio (76, 'Calle Y Norte', 6, 90301);

add\_domicilio (2, 'Calle Negra', 5, 110101);

add\_domicilio (37, 'Ave. S', 10, 110201);

add\_domicilio (1009, 'Ave. Dinero', 12, 120101);

add\_domicilio (4235, 'Calle Bilingüe', 9, 120201);

add\_domicilio (5344, 'Calle Rubí', 3, 120301);

add\_domicilio (468, 'Ave. Felicidad', 5, 130101);

add\_domicilio (89, 'Calle Tropical', 1, 130201);

add\_domicilio (743, 'Ave. Mora', 6, 130301);

END;

/

BEGIN

add\_sala ('SBT-1H1', '1-H1', 3, 8);

add\_sala ('SBT-1H2', '1-H2', 2, 6);

add\_sala ('SCL-2H1', '2-H1', 4, 4);

add\_sala ('SCL-2H2', '2-H2', 3, 4);

add\_sala ('SCO-3H1', '3-H1', 4, 5);

add\_sala ('SCO-3H2', '3-H2', 3, 6);

add\_sala ('SCH-4H1', '4-H1', 4, 6);

add\_sala ('SCH-4H2', '4-H2', 2, 4);

add\_sala ('SDA-5H1', '5-H1', 2, 8);

add\_sala ('SHE-6H1', '6-H1', 7, 6);

add\_sala ('SLS-7H1', '7-H1', 4, 4);

add\_sala ('SPA-8H1', '8-H1', 12, 6);

add\_sala ('SPA-8H2', '8-H2', 10, 4);

add\_sala ('SPA-8H3', '8-H3', 11, 4);

add\_sala ('SPA-8H4', '8-H4', 1, 2);

add\_sala ('SPA-8H5', '8-H5', 6, 6);

add\_sala ('SVE-9H1', '9-H1', 4, 6);

add\_sala ('SVE-9H2', '9-H2', 3, 6);

add\_sala ('SPO-13H1', '13-H1', 4, 6);

add\_sala ('SPO-13H2', '13-H2', 2, 4);

add\_sala ('SNG-12H1', '12-H1', 2, 12);

add\_sala ('SGY-10H1', '10-H1', 2, 10);

add\_sala ('UCISPA-8H5', '8-H5', 2, 5);

add\_sala ('UCISPA-8H1', '8-H1', 2, 2);

add\_sala ('UCISPA-8H3', '8-H3', 3, 4);

END;

/

BEGIN

add\_persona ('8-956-231', 'Lisa', 'Cortés', 'F', TO\_DATE('29/12/1982','DD/MM/YYYY'), 90, null, null);

add\_persona ('1-356-751', 'Melisa', 'Castillo', 'F', TO\_DATE('13/10/1967', 'DD/MM/YYYY'), 1567, 'SBT-1H1', '1-H1');

add\_persona ('2-437-769', 'Hector', 'González', 'M', TO\_DATE('10/01/1993','DD/MM/YYYY'), 6489, null, null);

add\_persona ('3-379-451', 'José', 'Magdaleno', 'M', TO\_DATE('27/05/1997','DD/MM/YYYY'), 560, 'SCO-3H1', '3-H1');

add\_persona ('4-587-998', 'Victoria', 'Pérez', 'F', TO\_DATE('24/03/1953','DD/MM/YYYY'), 889, null, null);

add\_persona ('5-187-7348', 'Ana', 'Ortiz', 'F', TO\_DATE('29/12/1972','DD/MM/YYYY'), 976, 'SDA-5H1', '5-H1');

add\_persona ('6-130-643', 'Rodrigo', 'Rodríguez', 'M', TO\_DATE('20/10/1964','DD/MM/YYYY'), 57, null, null);

add\_persona ('7-457-32', 'Martín', 'Márquez', 'M', TO\_DATE('12/10/1970','DD/MM/YYYY'), 245, 'SLS-7H1', '7-H1');

add\_persona ('8-587-949', 'Marquela', 'Pinzón', 'F', TO\_DATE('2/01/1991','DD/MM/YYYY'), 332, null, null);

add\_persona ('9-516-420', 'Paola', 'Jones', 'F', TO\_DATE('23/10/1999','DD/MM/YYYY'), 76, 'SVE-9H1', '9-H1');

add\_persona ('8-950-949', 'Victor', 'Jimenez', 'M', TO\_DATE('4/07/1995','DD/MM/YYYY'), 332, null, null);

add\_persona ('8-886-720', 'Román', 'Polanco', 'M', TO\_DATE('7/03/1975','DD/MM/YYYY'), 90, 'UCISPA-8H3', '8-H3');

add\_persona ('8-936-600', 'Sofía', 'Moreno', 'F', TO\_DATE('15/12/1959','DD/MM/YYYY'), 468, 'SPO-13H1', '13-H1');

add\_persona ('3-714-830', 'Alice', 'Rojas', 'F', TO\_DATE('6/12/1997','DD/MM/YYYY'), 789, 'SCO-3H2', '3-H2');

add\_persona ('12-450-92', 'Victoriano', 'Montezuma', 'M', TO\_DATE('13/11/1984','DD/MM/YYYY'), 1009, null, null);

add\_persona ('1-438-360', 'Casimiro', 'Bermudez', 'M', TO\_DATE('10/11/1990','DD/MM/YYYY'), 1726, 'SBT-1H2', '1-H2');

add\_persona ('2-571-457', 'Carmen', 'Alvarado', 'F', TO\_DATE('24/06/1978','DD/MM/YYYY'), 8623, 'SCL-2H2', '2-H2');

add\_persona ('4-614-730', 'Kathy', 'Arosemena', 'F', TO\_DATE('27/02/1987','DD/MM/YYYY'), 889, 'SCH-4H2', '4-H2');

add\_persona ('5-411-250', 'Daniela', 'López', 'F', TO\_DATE('19/05/1992','DD/MM/YYYY'), 432, 'SDA-5H1', '5-H1');

add\_persona ('6-248-581', 'Jonathan', 'De La Flor', 'M', TO\_DATE('31/08/1996','DD/MM/YYYY'), 25, 'SHE-6H1', '6-H1');

END;

/

BEGIN

Add\_doctor\_hospital (1, '3-H1');

Add\_doctor\_hospital (2, '3-H2');

Add\_doctor\_hospital (3, '7-H1');

Add\_doctor\_hospital (4, '9-H1');

Add\_doctor\_hospital (5, '9-H2');

Add\_doctor\_hospital (6, '12-H1');

Add\_doctor\_hospital (7, '1-H1');

Add\_doctor\_hospital (8, '1-H2');

Add\_doctor\_hospital (9, '6-H1');

Add\_doctor\_hospital (10, '4-H1');

Add\_doctor\_hospital (11, '4-H2');

Add\_doctor\_hospital (12, '8-H1');

Add\_doctor\_hospital (13, '8-H2');

Add\_doctor\_hospital (14, '8-H3');

Add\_doctor\_hospital (15, '8-H4');

Add\_doctor\_hospital (16, '8-H5');

Add\_doctor\_hospital (17, '5-H1');

Add\_doctor\_hospital (18, '2-H1');

Add\_doctor\_hospital (19, '2-H2');

Add\_doctor\_hospital (20, '10-H1');

Add\_doctor\_hospital (21, '13-H1');

Add\_doctor\_hospital (22, '13-H2');

END;

/

BEGIN

add\_persona\_sintoma ('F', 1);

add\_persona\_sintoma ('DG', 1);

add\_persona\_sintoma ('DR', 2);

add\_persona\_sintoma ('F', 2);

add\_persona\_sintoma ('TS', 3);

add\_persona\_sintoma ('DC', 4);

add\_persona\_sintoma ('F', 5);

add\_persona\_sintoma ('P', 6);

add\_persona\_sintoma ('E', 7);

add\_persona\_sintoma ('DG', 7);

add\_persona\_sintoma ('DR', 8);

add\_persona\_sintoma ('CO', 9);

add\_persona\_sintoma ('DM', 10);

add\_persona\_sintoma ('D', 11);

add\_persona\_sintoma ('C', 12);

add\_persona\_sintoma ('E', 12);

add\_persona\_sintoma ('IH', 13);

add\_persona\_sintoma ('TS', 14);

add\_persona\_sintoma ('F', 15);

add\_persona\_sintoma ('CO', 15);

END;

/

BEGIN

add\_persona\_ep ('CAN', 1, to\_date('10/3/2019','dd/mm/yyyy'));

add\_persona\_ep ('DIA', 2, to\_date('12/5/2015','dd/mm/yyyy'));

add\_persona\_ep ('ECA', 3, to\_date('13/4/2017','dd/mm/yyyy'));

add\_persona\_ep ('EPU', 4, to\_date('30/1/2018','dd/mm/yyyy'));

add\_persona\_ep ('HIP', 5, to\_date('10/8/2019','dd/mm/yyyy'));

add\_persona\_ep ('CAN', 6, to\_date('31/10/2018','dd/mm/yyyy'));

add\_persona\_ep ('DIA', 7, to\_date('12/3/2016','dd/mm/yyyy'));

add\_persona\_ep ('ECA', 8, to\_date('10/5/2017','dd/mm/yyyy'));

add\_persona\_ep ('EPU', 9, to\_date('24/6/2019','dd/mm/yyyy'));

add\_persona\_ep ('HIP', 10, to\_date('16/9/2017','dd/mm/yyyy'));

END;

/

BEGIN

add\_persona\_prueba (1, 'H', to\_date('23/4/2020','dd/mm/yyyy'));

add\_persona\_prueba (2, 'H', to\_date('30/6/2020','dd/mm/yyyy'));

add\_persona\_prueba (3, 'S', to\_date('12/5/2020','dd/mm/yyyy'));

add\_persona\_prueba (4, 'H', to\_date('7/6/2020','dd/mm/yyyy'));

add\_persona\_prueba (5, 'S', to\_date('13/6/2020','dd/mm/yyyy'));

add\_persona\_prueba (6, 'H', to\_date('10/5/2020','dd/mm/yyyy'));

add\_persona\_prueba (7, 'S', to\_date('8/5/2020','dd/mm/yyyy'));

add\_persona\_prueba (8, 'H', to\_date('30/5/2020','dd/mm/yyyy'));

add\_persona\_prueba (9, 'S', to\_date('12/4/2020','dd/mm/yyyy'));

add\_persona\_prueba (10, 'H', to\_date('25/5/2020','dd/mm/yyyy'));

add\_persona\_prueba (11, 'S', to\_date('9/6/2020','dd/mm/yyyy'));

add\_persona\_prueba (12, 'H', to\_date('4/7/2020','dd/mm/yyyy'));

add\_persona\_prueba (13, 'H', to\_date('23/3/2020','dd/mm/yyyy'));

add\_persona\_prueba (14, 'S', to\_date('18/4/2020','dd/mm/yyyy'));

add\_persona\_prueba (15, 'H', to\_date('29/6/2020','dd/mm/yyyy'));

END;

/

BEGIN

Add\_telefono\_h ('10-H1', '345-8910');

Add\_telefono\_h ('12-H1', '268-0943');

Add\_telefono\_h ('13-H1', '254-8926');

Add\_telefono\_h ('13-H2', '253-3447');

Add\_telefono\_h ('1-H1', '774-5689');

Add\_telefono\_h ('1-H2', '765-8910');

Add\_telefono\_h ('2-H1', '945-8543');

Add\_telefono\_h ('2-H2', '998-3219');

Add\_telefono\_h ('3-H1', '445-5432');

Add\_telefono\_h ('3-H2', '447-5617');

Add\_telefono\_h ('4-H1', '785-8532');

Add\_telefono\_h ('4-H2', '735-3210');

Add\_telefono\_h ('5-H1', '245-5670');

Add\_telefono\_h ('6-H1', '995-6789');

Add\_telefono\_h ('7-H1', '996-7480');

Add\_telefono\_h ('8-H1', '507-5830');

Add\_telefono\_h ('8-H2', '305-6300');

Add\_telefono\_h ('8-H3', '360-7400');

Add\_telefono\_h ('8-H4', '399-8082');

Add\_telefono\_h ('8-H5', '503-6600');

Add\_telefono\_h ('9-H1', '985-4287');

Add\_telefono\_h ('9-H2', '978-0045');

END;

/

BEGIN

add\_tipo\_telefono\_p (1, 'Residencial');

add\_tipo\_telefono\_p (2, 'Celular');

END;

/

**--Vistas**

* **Prov\_Dom**

**/\*Esta vista permite observer el nombre del domicilio junto con el distrito y a la provincia que pertenecen.\*/**

create view Prov\_Dom as select d.num\_domic, c.nombre\_correg, p.nombre\_prov,p.Cod\_prov

from Domicilio d join Corregimiento c

on d.Cod\_correg=c.Cod\_correg join Distrito dis

on dis.Cod\_dist=c.Cod\_dist join Provincia p

on p.Cod\_prov=dis.Cod\_prov;

* **Per\_Dom**

**/\*Esta consulta muestra la cédula, el nombre y el número de domicilio del paciente. Esta consulta está dirigida para los analistas de datos del MINSA ya que ellos requieron estos datos para ir contabilizando la cantidad de personas que se van infectando cada día para realizar un reporte diario.\*/**

create view Per\_Dom as select distinct Cedula\_p, apellido\_p, nombre\_p, Num\_domic

from Persona

order by apellido\_p;

* **Sin**

**/\*Un epidemiólogo puede utilizar esta consulta para ver todos los pacientes que han ingresado con fiebre y hacer una estadística de cuál es la cantidad de la población afectada por este síntoma y determinar si hay algún cambio en futuras pruebas, ya que es posible estén utilizando tratamientos y observar cómo es el comportamiento de las personas en cuanto a la fiebre y poder indicar algún factor en cuanto al tratamiento en cuestión.\*/**

create view Sin as select per.apellido\_p, nombre\_p as, sim.nombre\_sint, per.Cedula\_p

from Sintomas sim join Persona\_Sintoma persim

on sim.Cod\_sint = persim.Cod\_sint join Persona per

on per.id\_persona = persim.id\_persona

where sim.nombre\_sint in ('Fiebre');

* **Prov\_per\_dom**

**/\*Esta consulta pudiese ser útil para un Epidemiólogo, para poder conocer el comportamiento del virus si se seleccionan dos personas de una provincia determinada, en este caso, Panamá, que no tengan dos de los síntomas más comunes, los cuales son tos seca y fiebre, y además, que tengan o no alguna enfermedad previa, en este caso, diabetes, y así puede realizarles los estudios pertinentes.\*/**

create view Prov\_per\_dom as select per.apellido\_p, nombre\_p

from Provincia pro join Distrito dis

on pro.Cod\_prov = dis.Cod\_prov join Corregimiento cor

on dis.Cod\_dist = cor.Cod\_dist join Domicilio dom

on cor.Cod\_correg = dom.Cod\_correg join Persona per

on per.Num\_domic = dom.Num\_domic join Persona\_Sintoma pers

on per.id\_persona = pers.id\_persona join Persona\_Enfermedades\_Previas peref

on pers.id\_persona = peref.id\_persona

where pro.Cod\_prov = 8 and pers.Cod\_sint not in ('F', 'TS') or peref.Cod\_enf = 'DIA';

* **Prov\_per**

**/\*Esta consulta muestra el nombre del paciente y el nombre de la provincia en que se encuentra, con la condición de que estas provincias estén entre Bocas del Toro, Coclé, Colón, Chiriquí y Herrera. Esta consulta está dirigida principalmente al ministro de salud, ya que si este quiere tener un informe más claro de cómo está la situación en estas provincias en específico.\*/**

create view Prov\_per as select per.apellido\_p, nombre\_p, pro.nombre\_prov

from Provincia pro join Distrito dis

on pro.Cod\_prov = dis.Cod\_prov join Corregimiento cor

on dis.Cod\_dist = cor.Cod\_dist join Domicilio dom

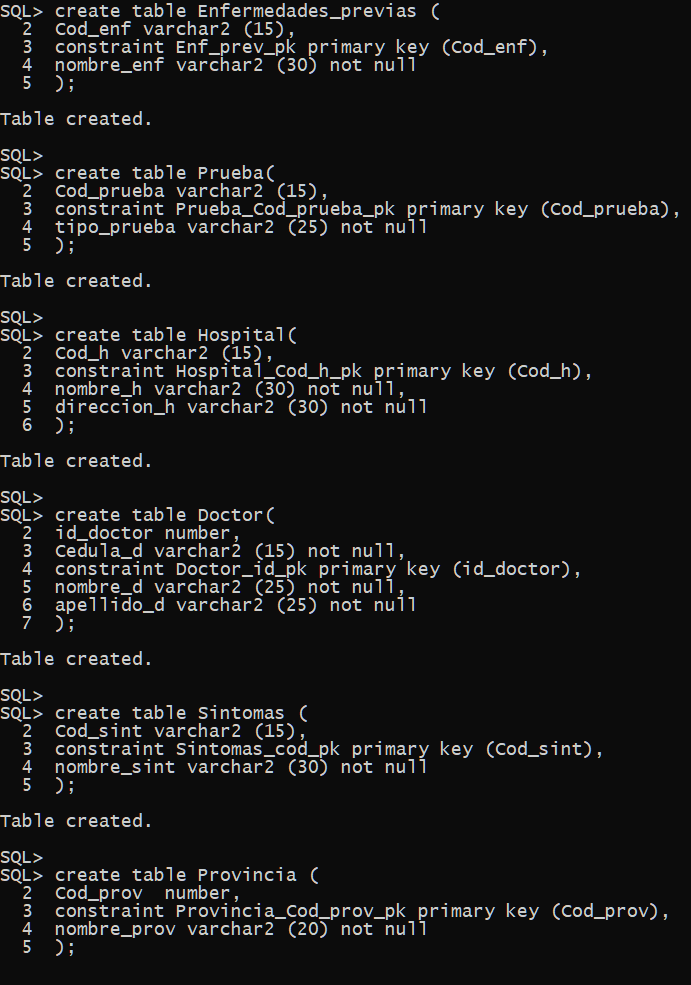
on cor.Cod\_correg = dom.Cod\_correg join Persona per

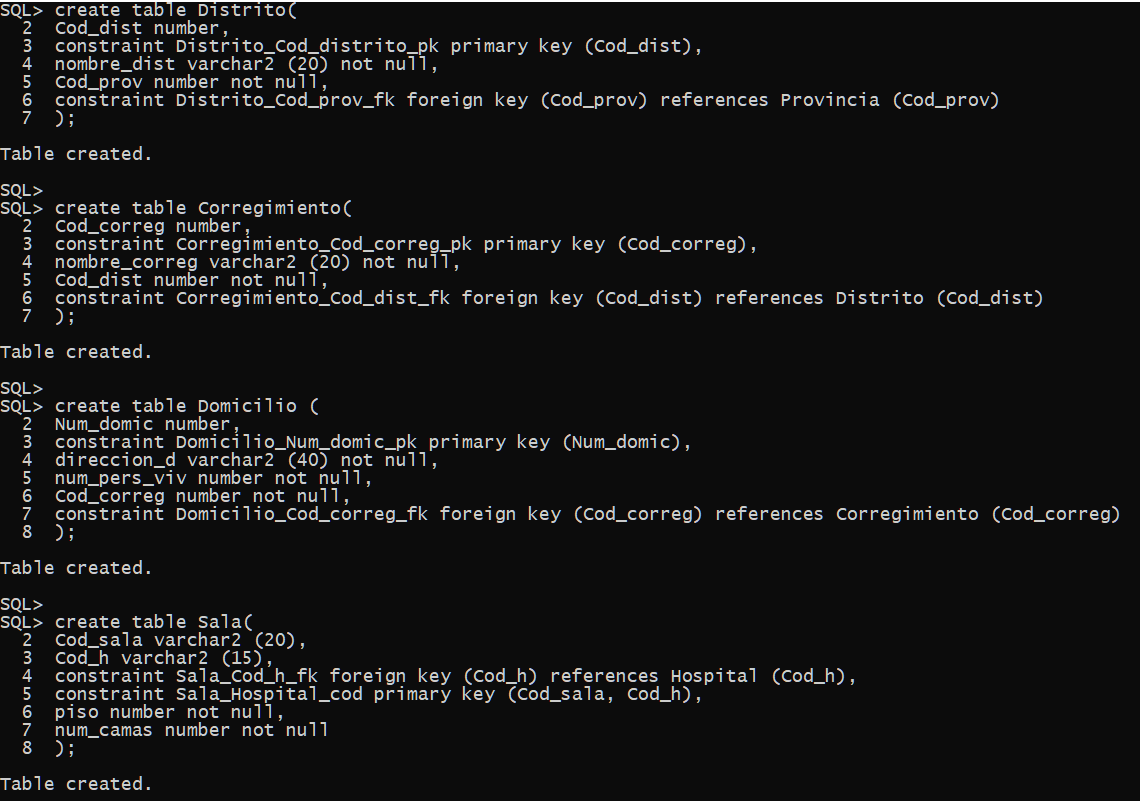
on per.Num\_domic = dom.Num\_domic

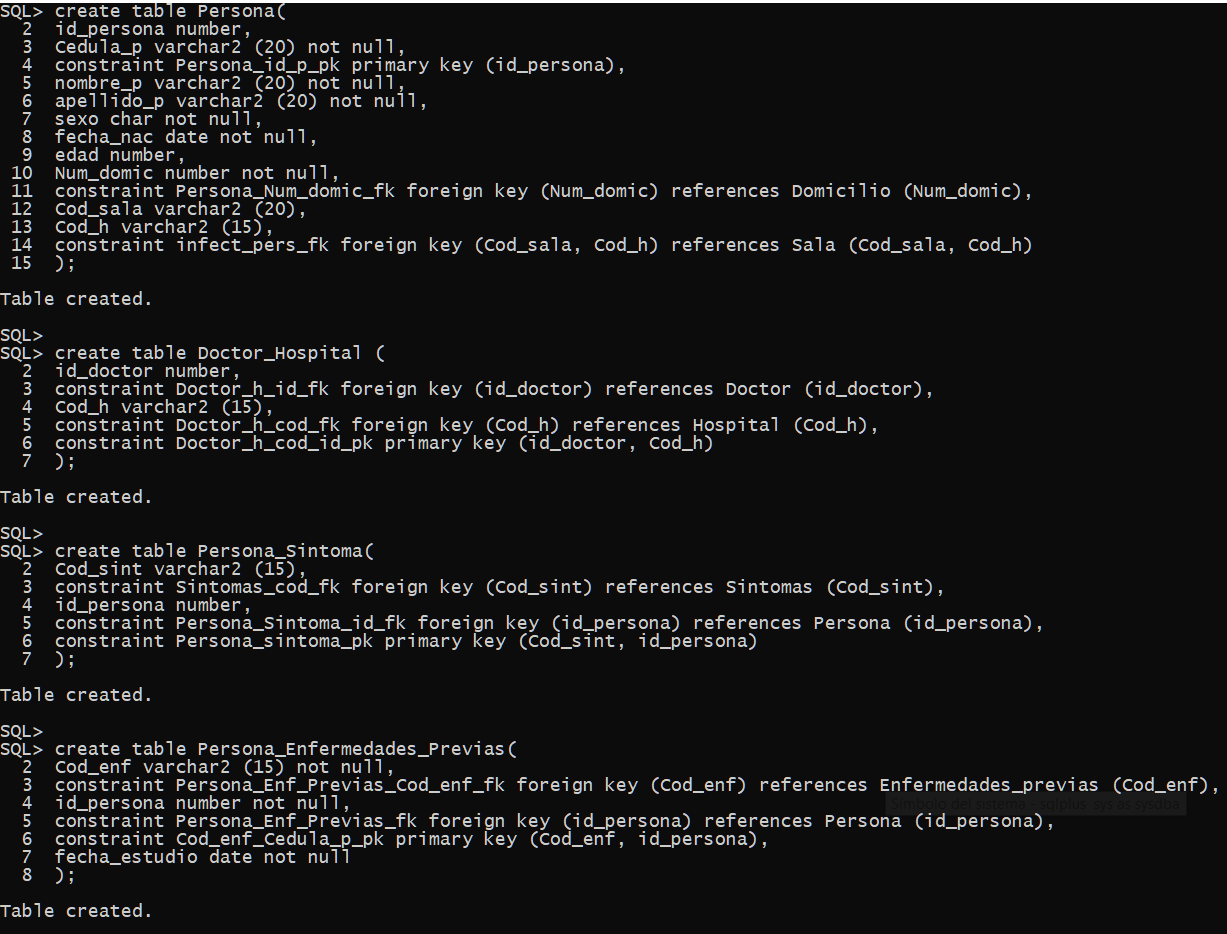
where pro.Cod\_prov between 1 and 5;

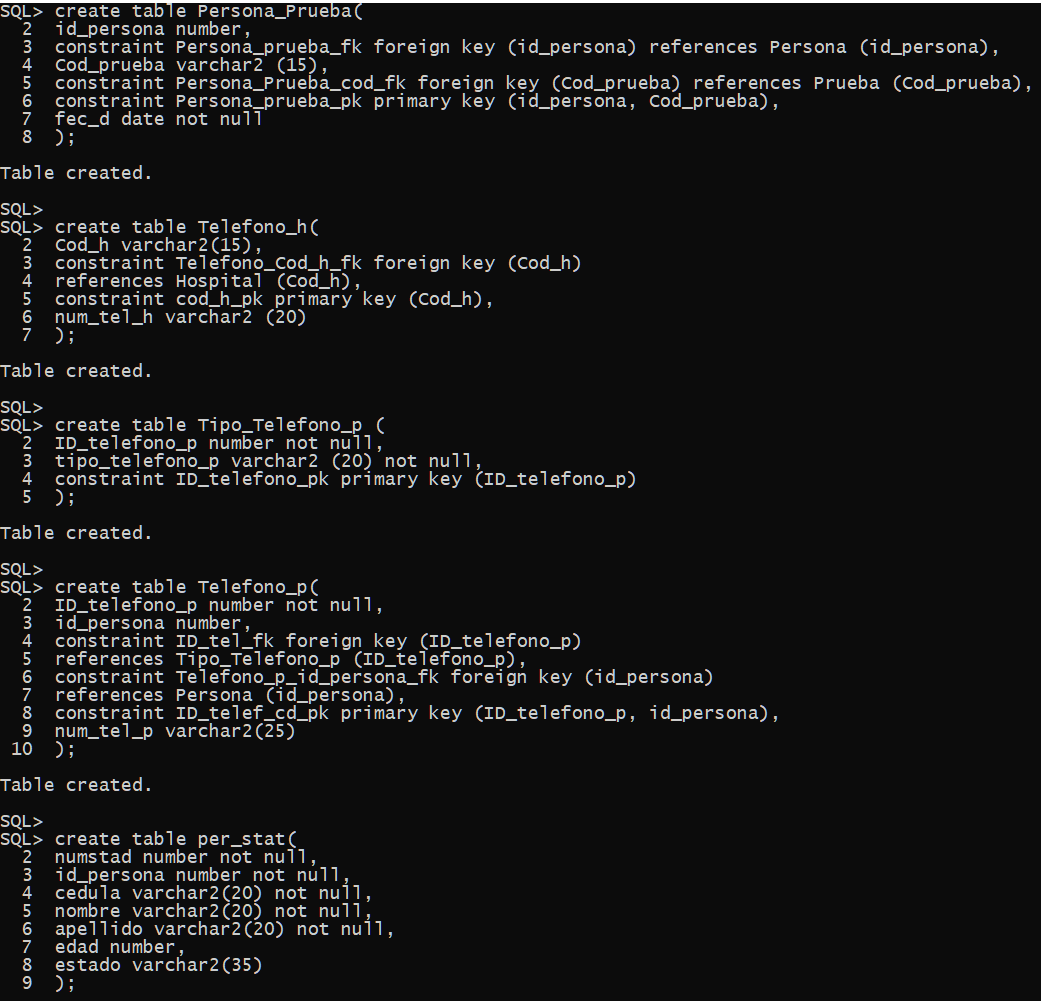
**ANEXO**

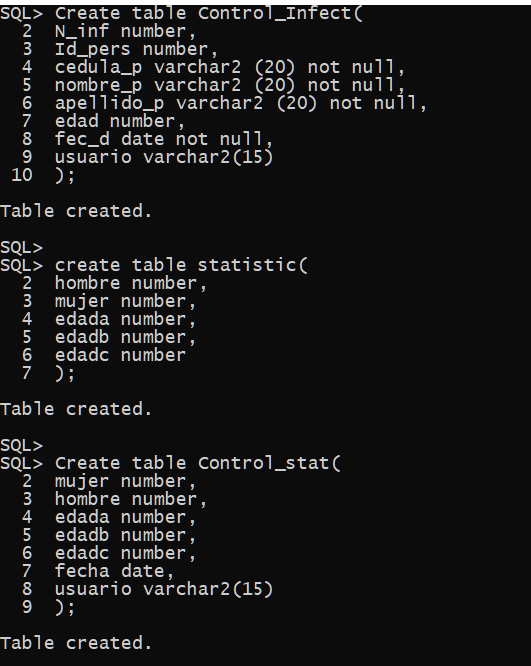
**Creación de Tablas**



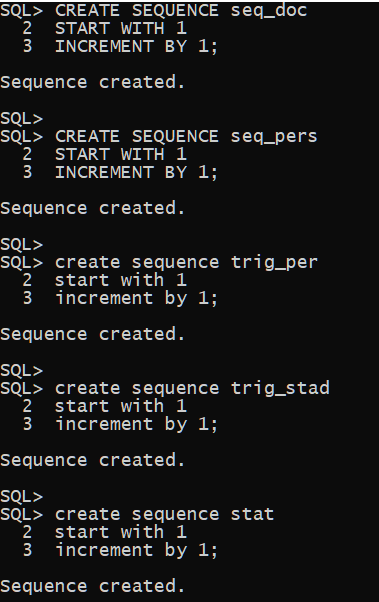




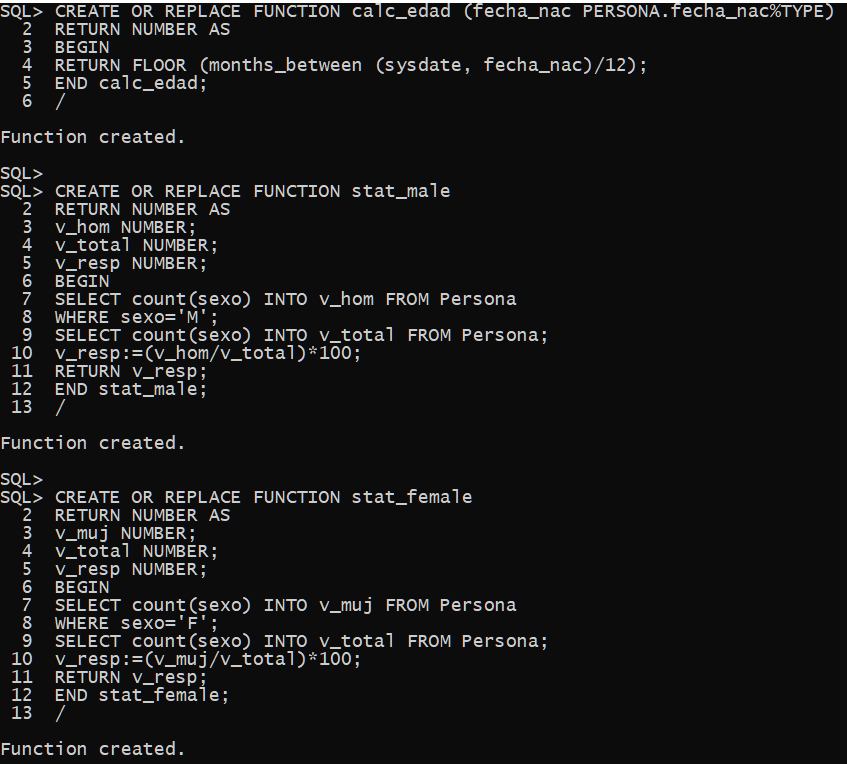


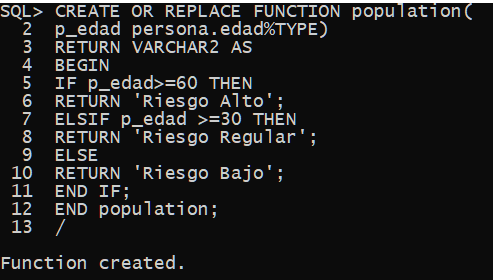


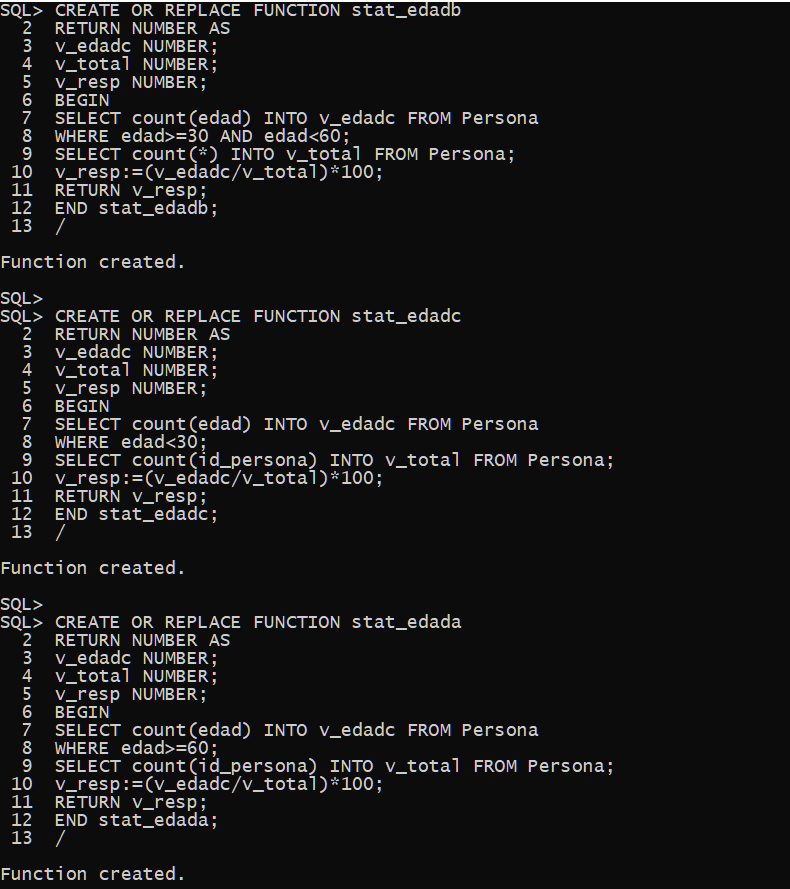
**Creación de Secuencias**



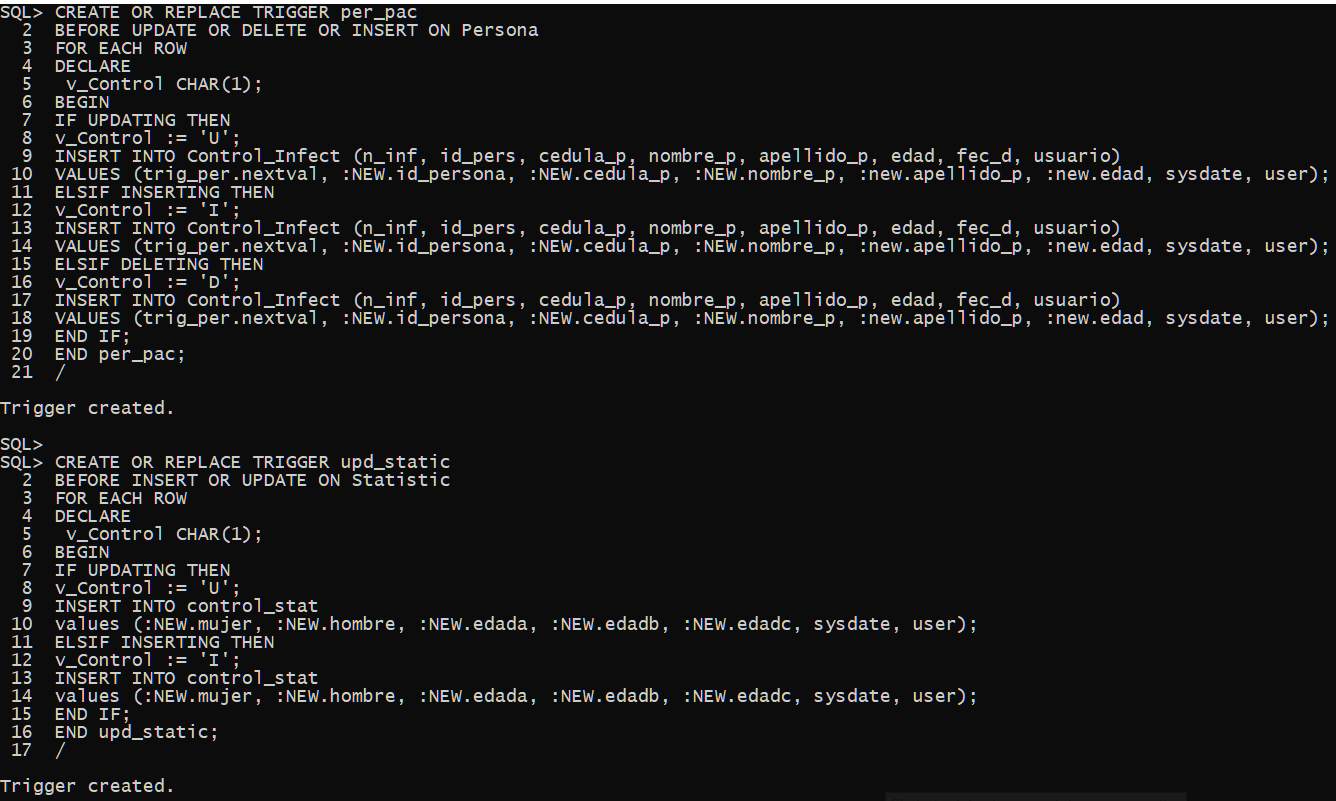
**Creación de Funciones**



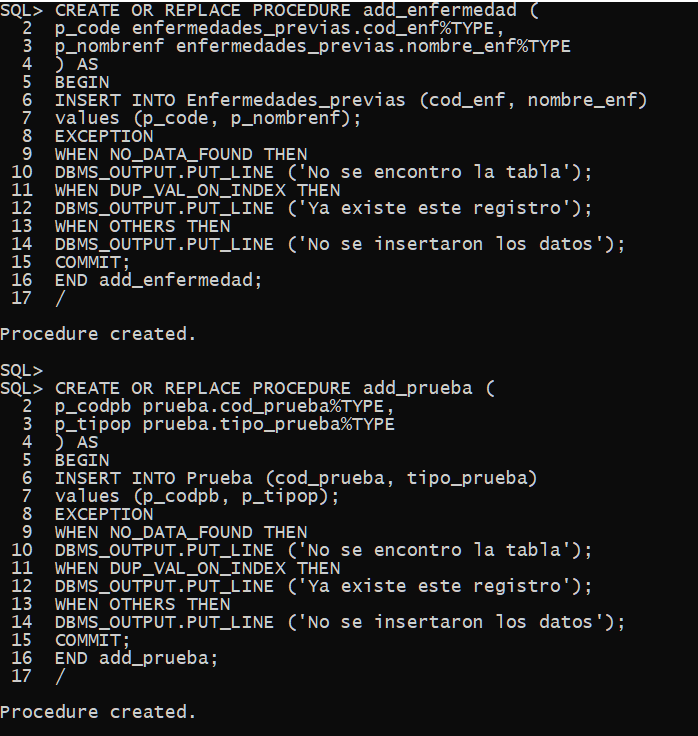


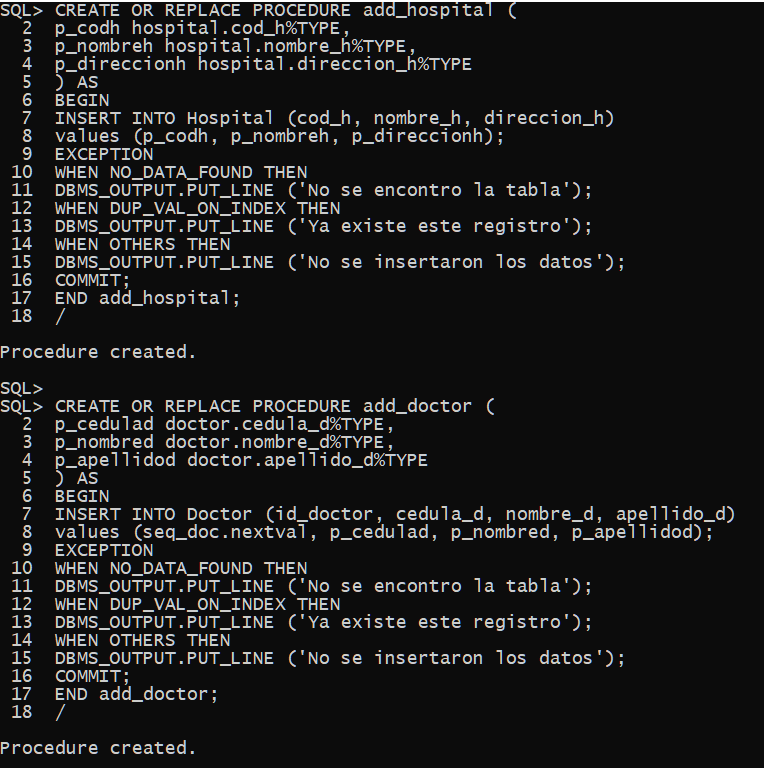


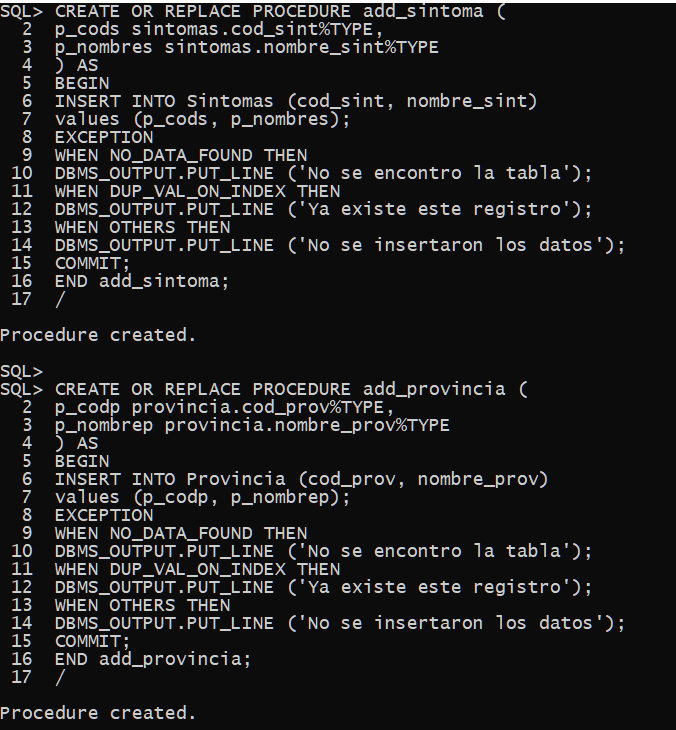
**Creación de Triggers**

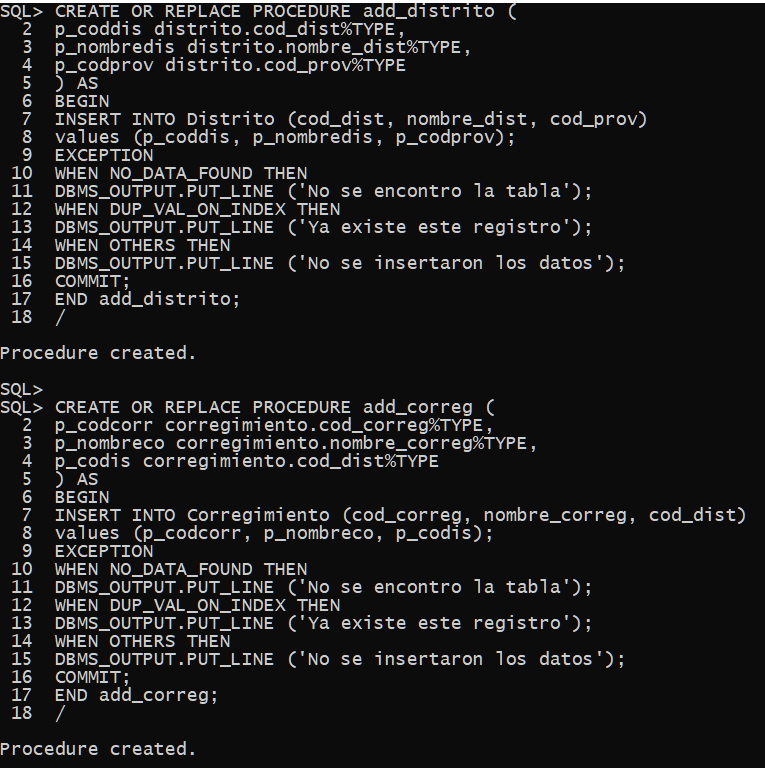


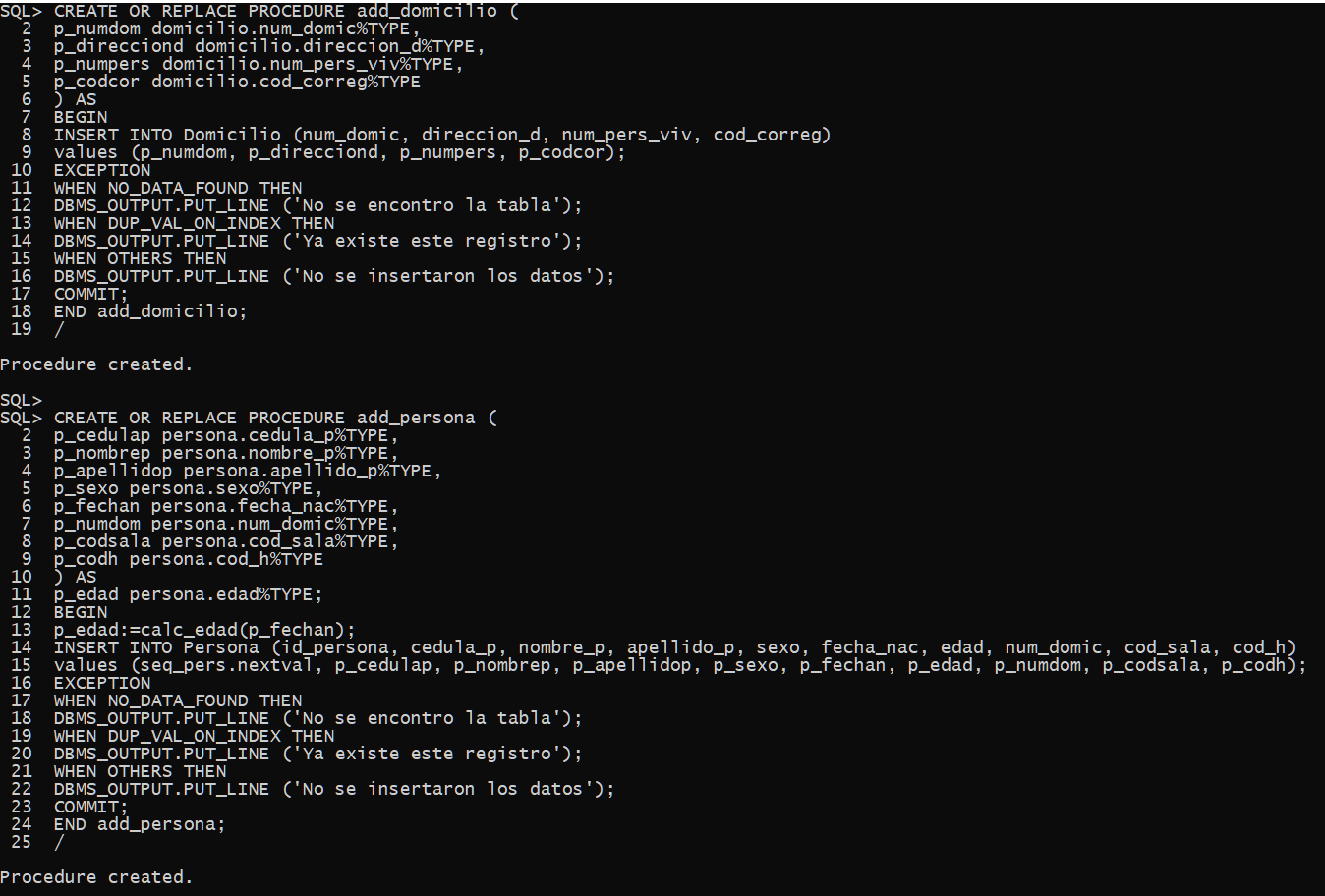
**Creación de Procedimientos**

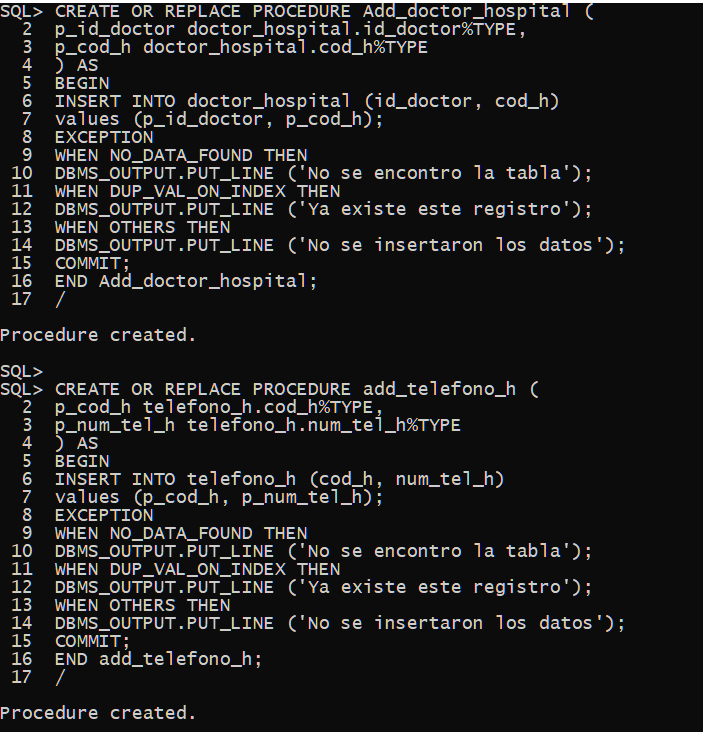


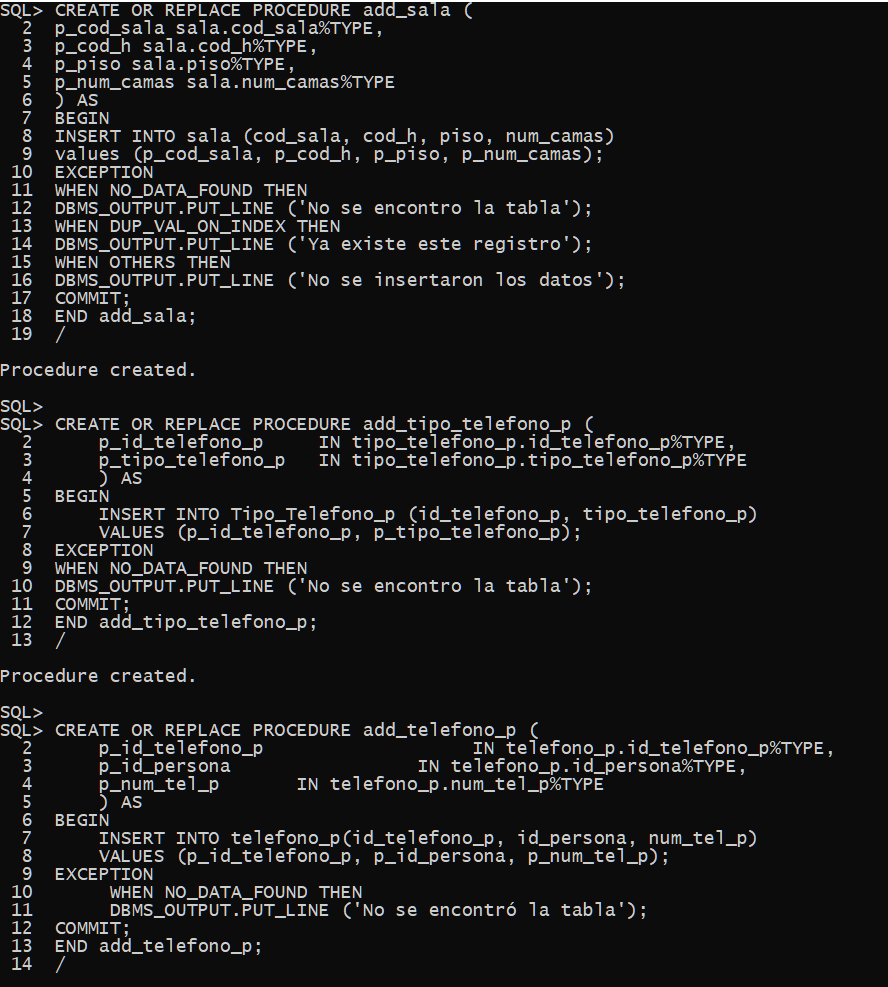


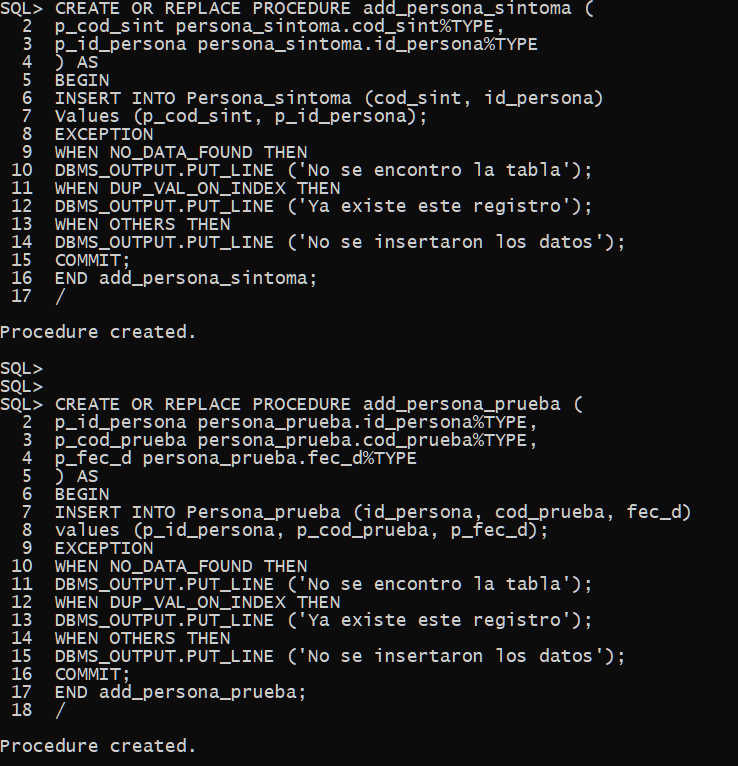


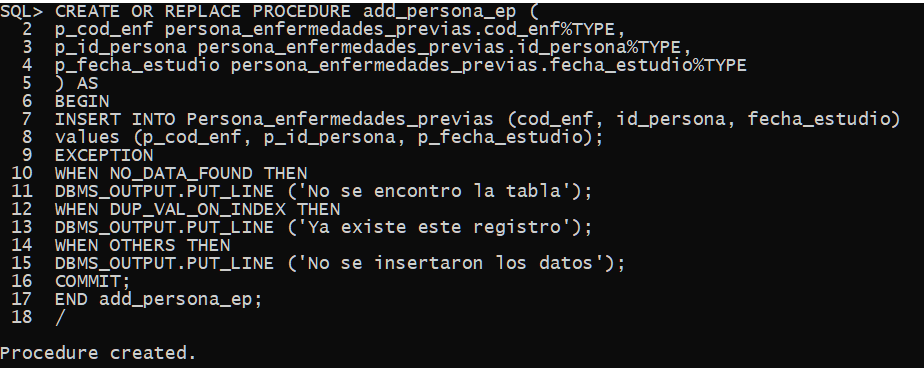


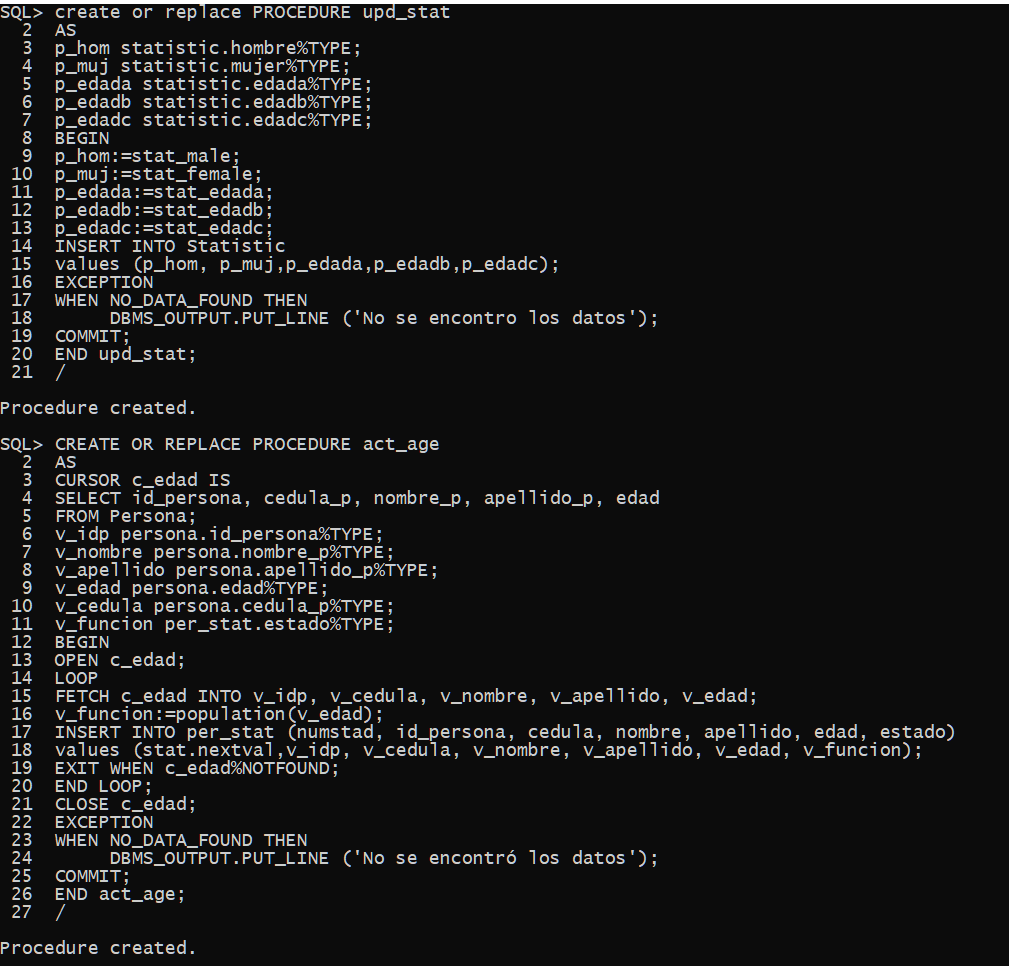




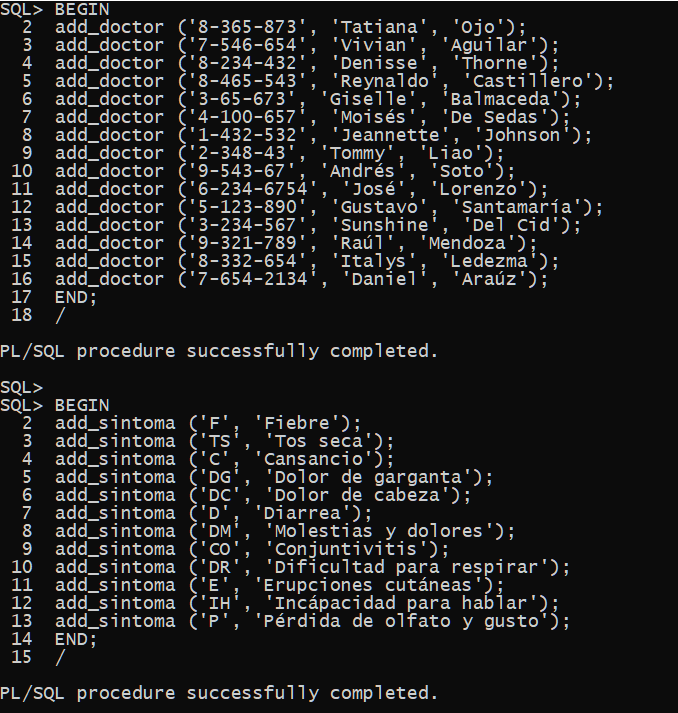
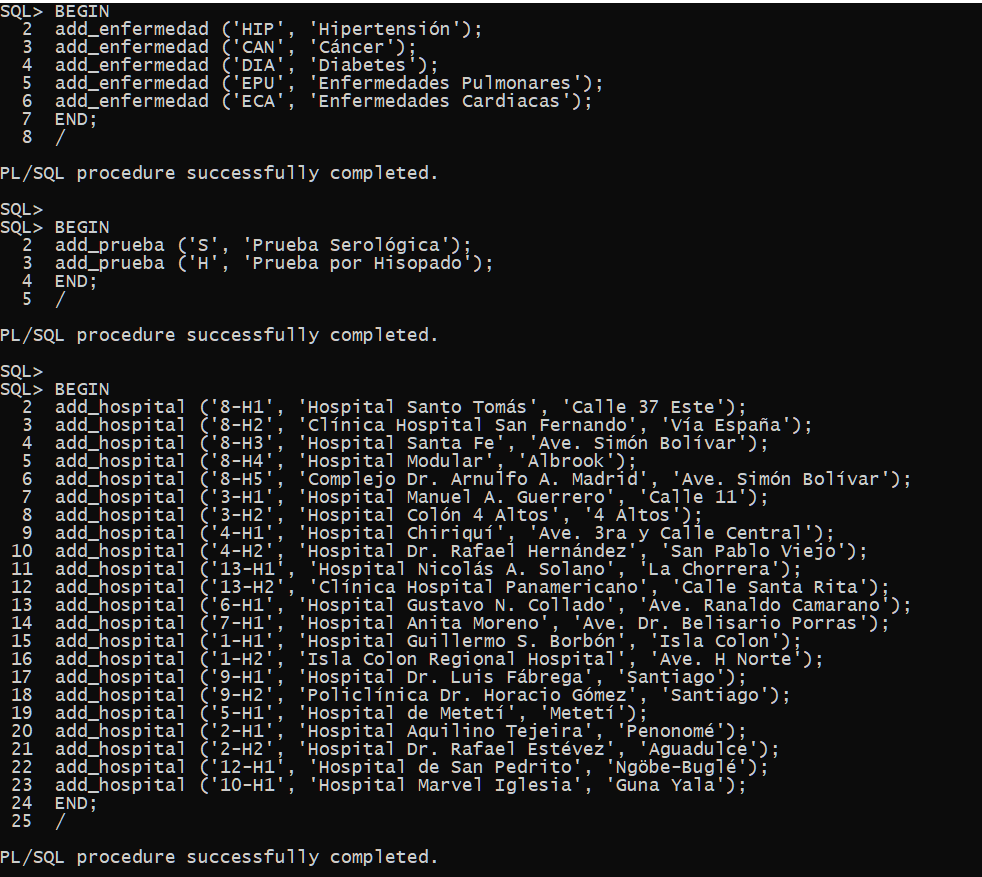


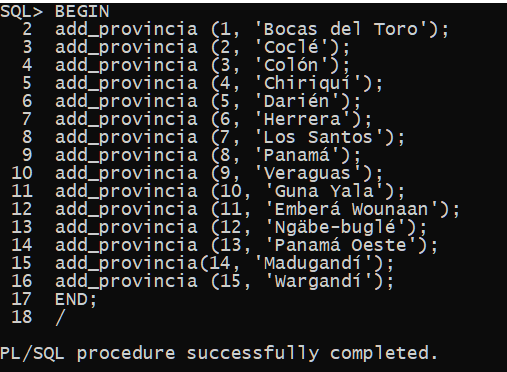


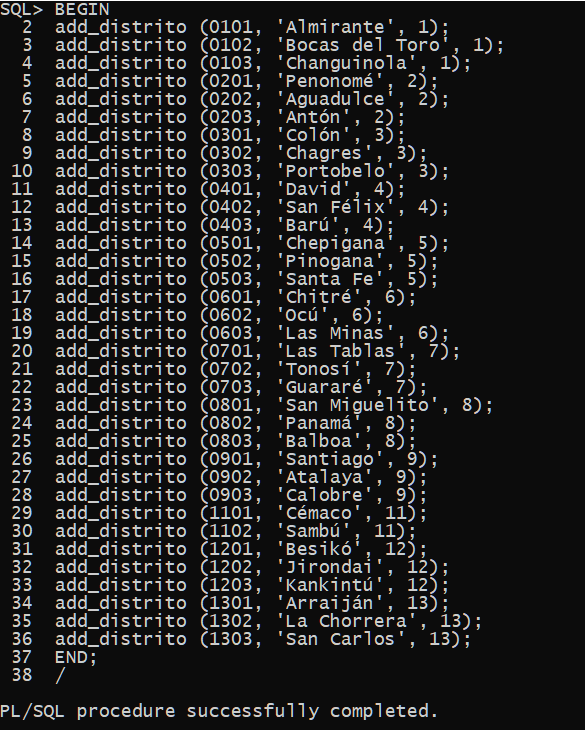


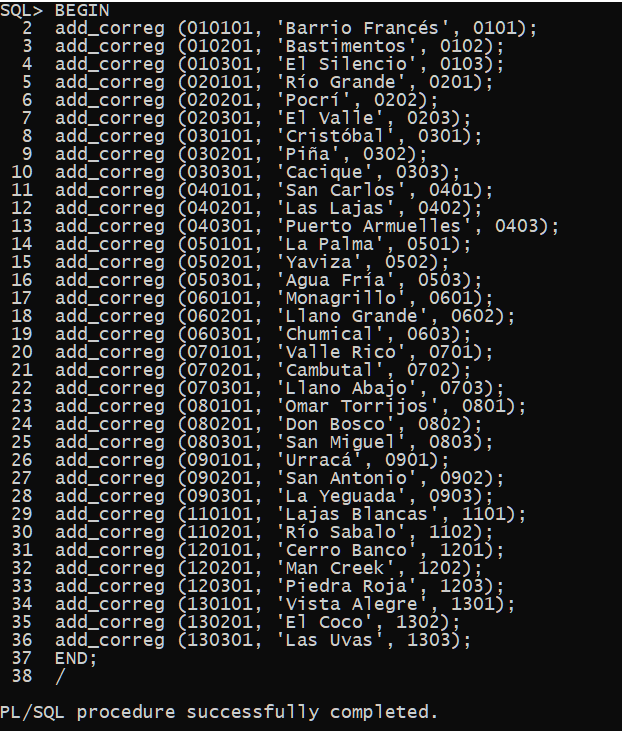


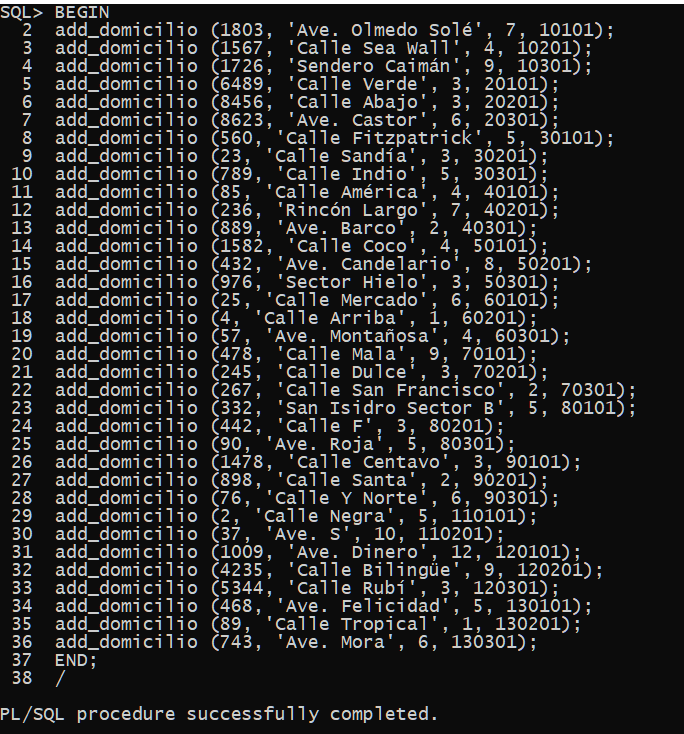
**Invocaciones**

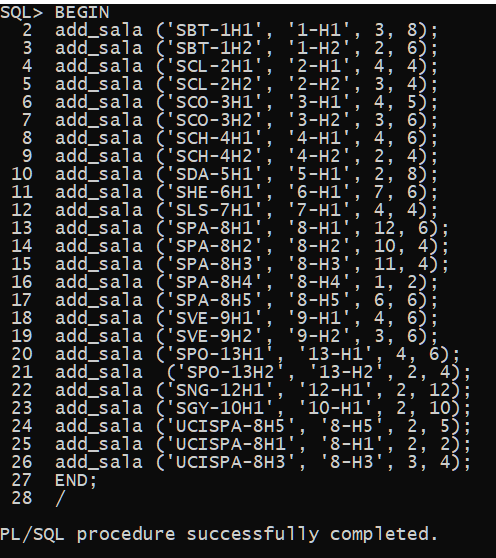




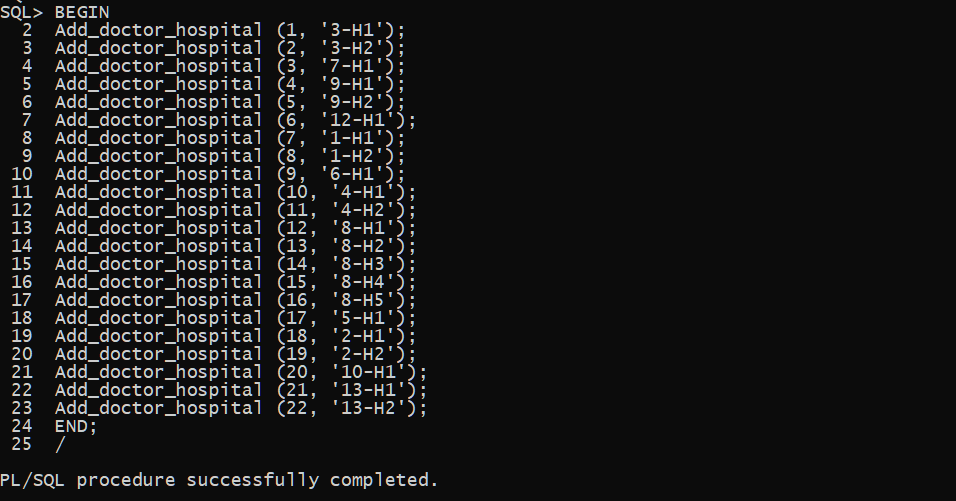


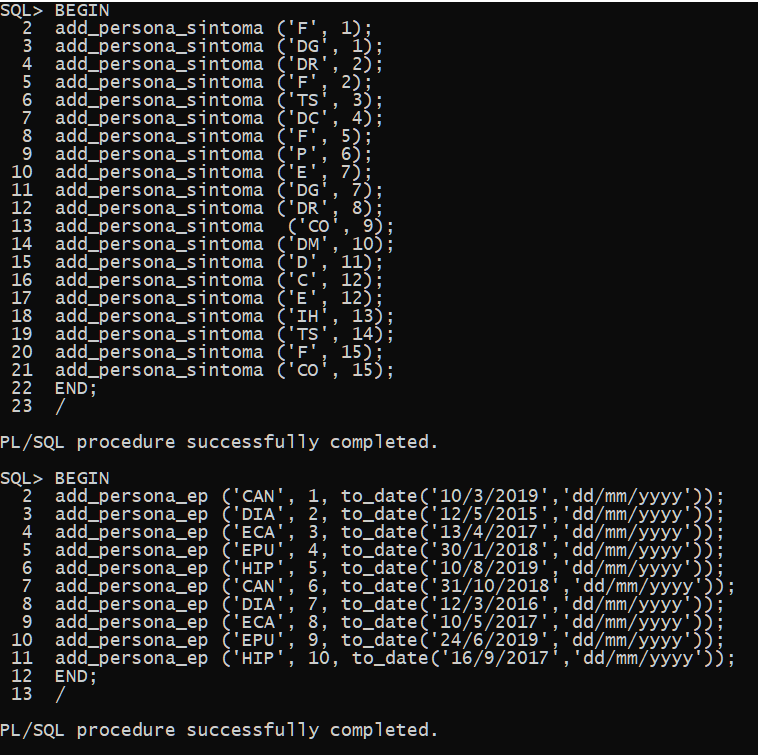


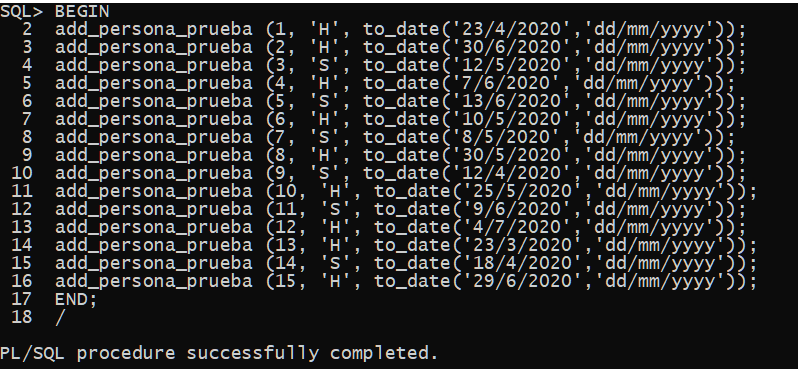


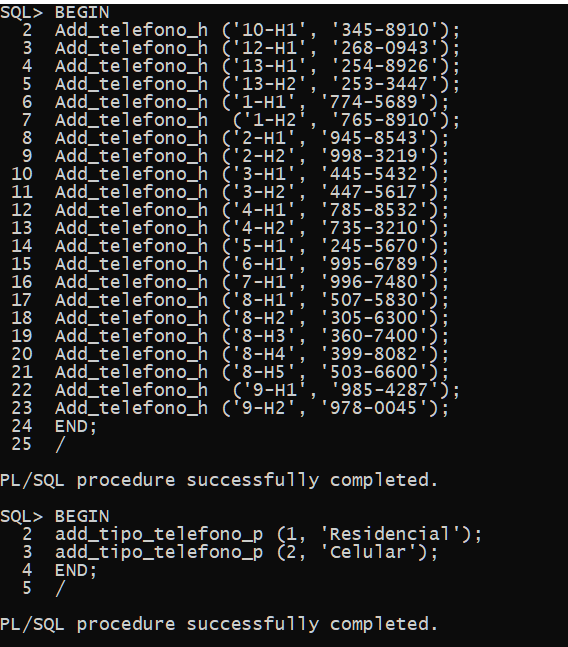


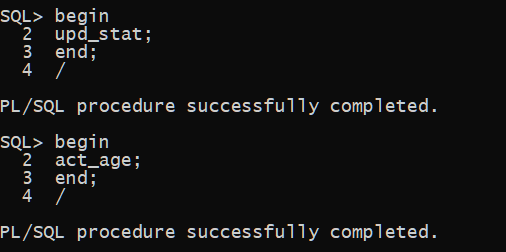




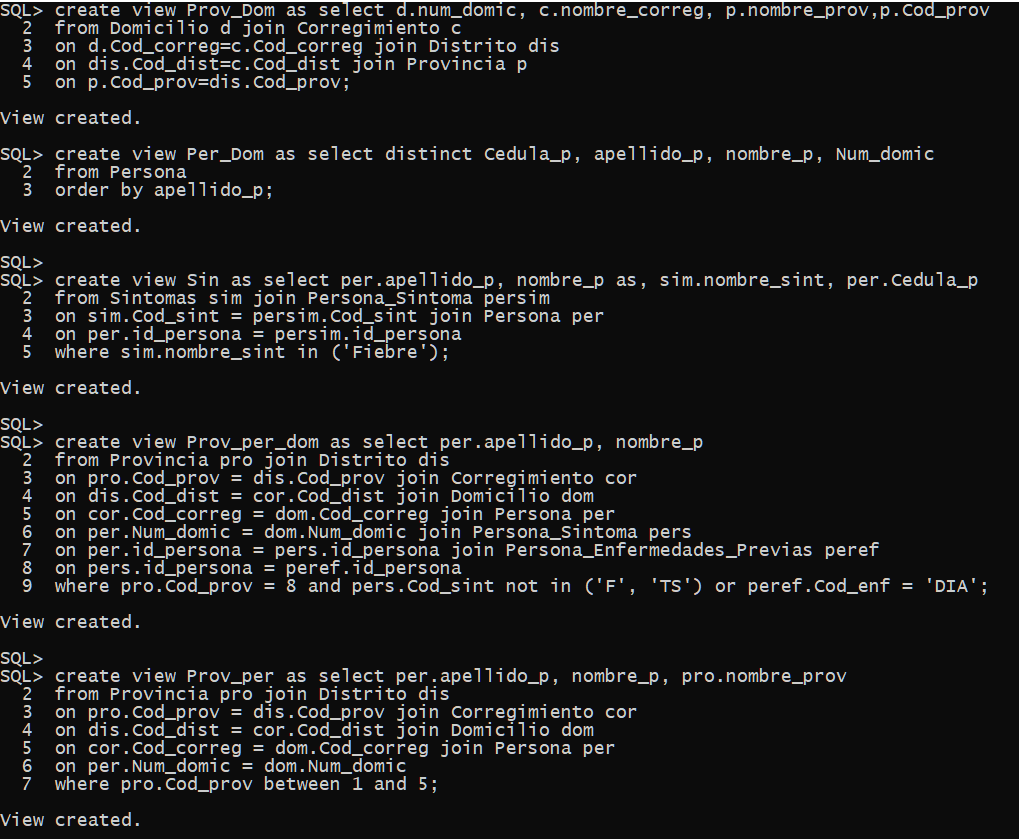




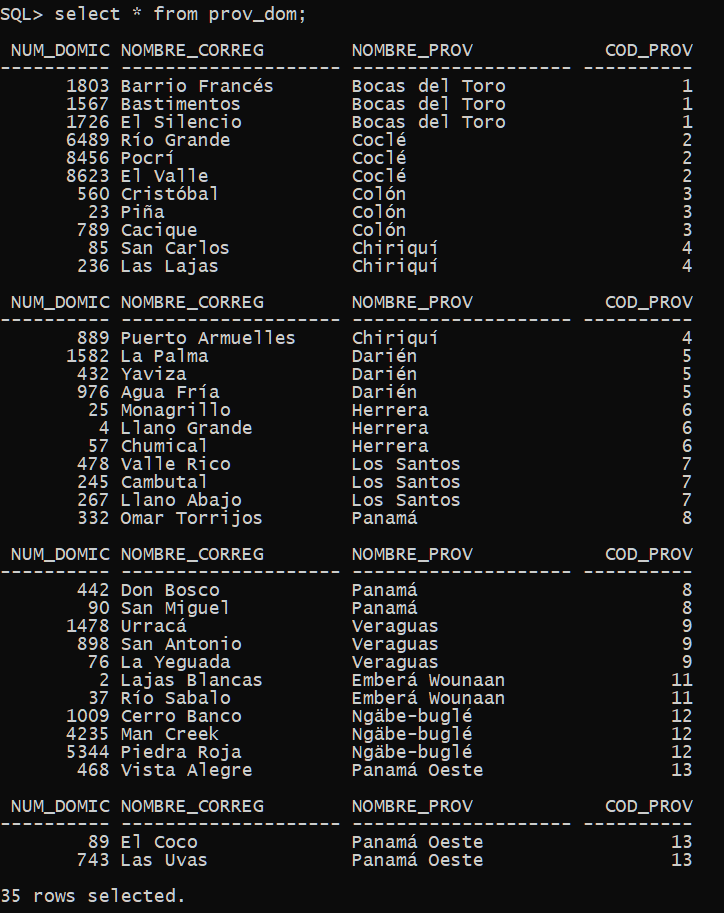


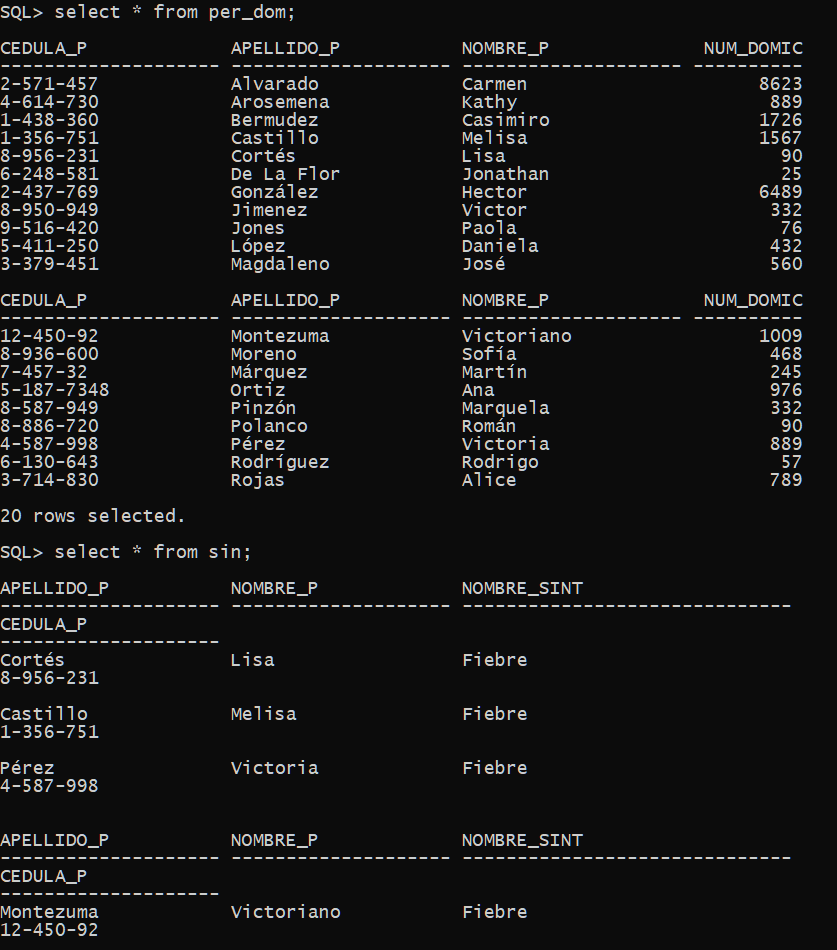


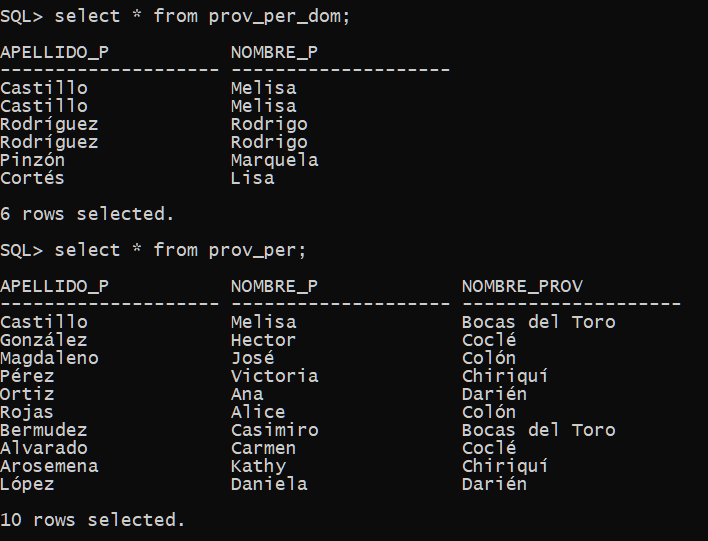
**Creación de Vistas**



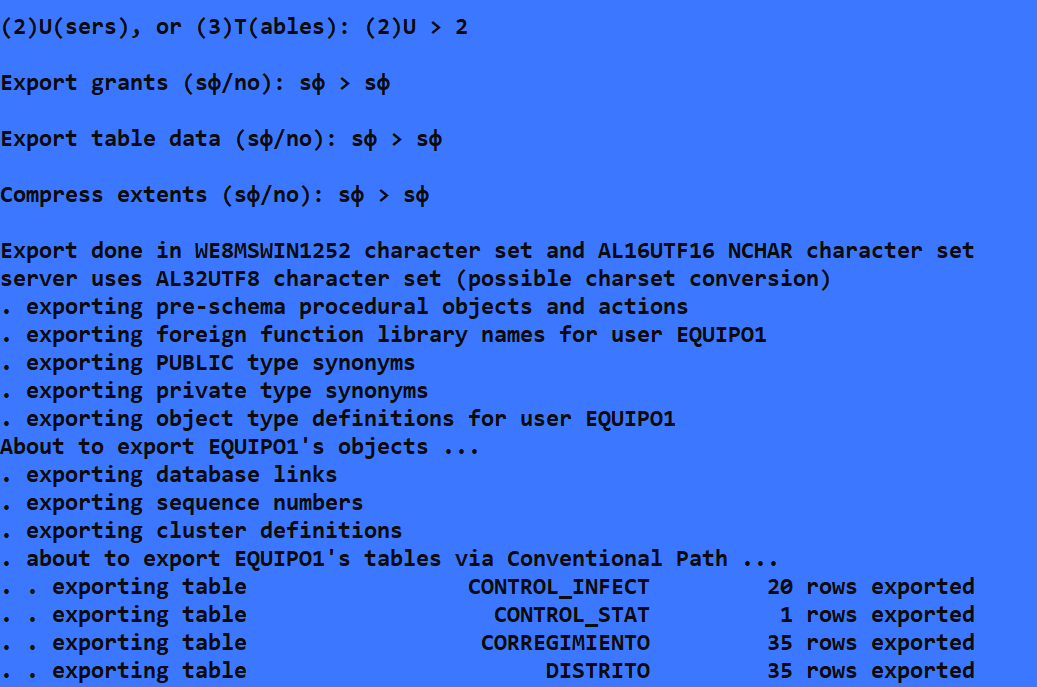
**Vistas**

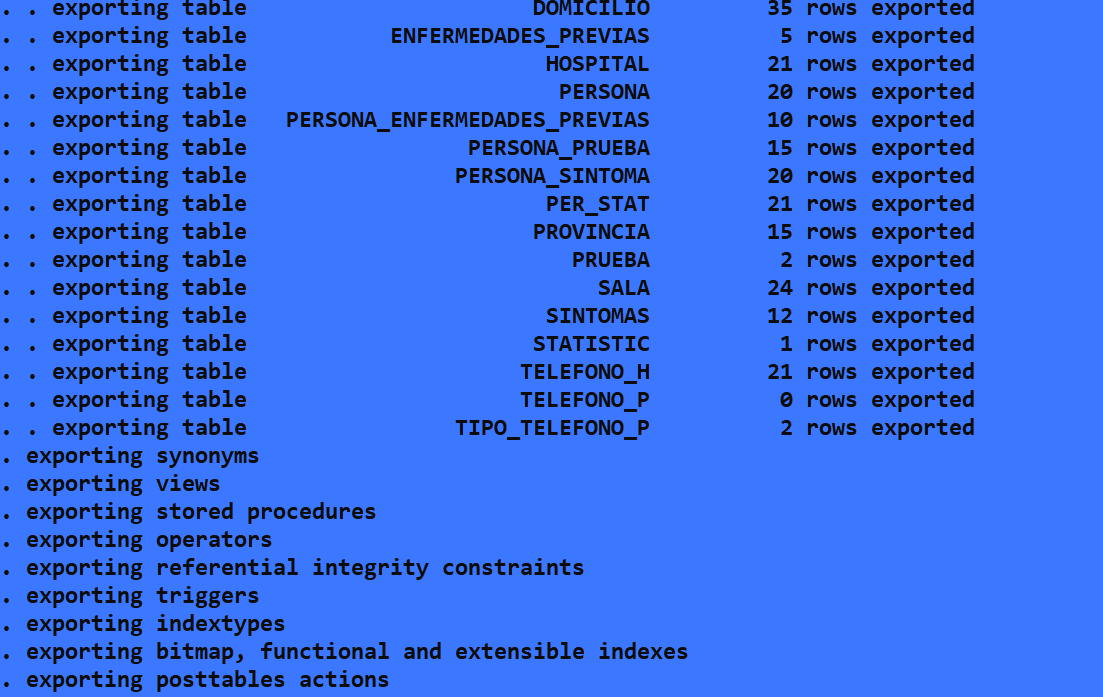






**Exportación**





**CONCLUSIÓN**

La crisis del coronavirus es el mayor desafío que nos ha tocado enfrentar por primera vez después de mucho tiempo, actualmente los países se encuentran en una lucha constante para contrarrestar la propagación de la enfermedad mediante la realización de pruebas, tratamientos, recomendaciones de salud, poniendo en cuarentena a la población entre muchas otras cosas más.

Las enfermedades tienen un gran impacto en la demografía de la población humana, pues las enfermedades son las causantes de muchas muertes en todo el planeta, por ello la modelación se ha vuelto una herramienta muy importante en el estudio de la propagación de las infecciones.

Además de problemas en la salud mundial esta pandemia también ha ocasionado muchos problemas en la industria, economía y muchos otros sectores ya que con la medida de la cuarentena las personas no pueden salir a trabajar y ni mucho menos pagar sus impuestos. Cada país necesita actuar de inmediato para prepararse, responder y recuperarse. Por ello se necesitan de bases de datos para almacenar todo tipo de datos que se necesiten en el presente o que quizás en un futuro se puedan necesitar.

Con la realización de este trabajo nos hemos percatado que tener un buen control de datos, no específicamente esta situación de pandemia actual sino también para cualquier otra problemática, es de mucha ayuda porque nos permite organizarnos mejor en situaciones de estrés y como principal fin de las bases de datos y los sistemas de información, nos ayuda a tomar las mejores decisiones ya sea dentro de una organización o a nivel poblacional de un país o región.