Documentación de Bases de Datos



## GNC: Gestión de Nacimientos y Cría

### Integrantes:

Samuel Rodríguez

Agustín Rodríguez

Emiliano Silva

Eduardo Mena

Contenido

[Documentación de Bases de Datos 1](#_Toc529983676)

[GNC: Gestión de Nacimientos y Cría 1](#_Toc529983677)

[Integrantes: 1](#_Toc529983678)

[**Tabla ALIMENTOS** 4](#_Toc529983679)

[**Estructura** 4](#_Toc529983680)

[**Constraints** 4](#_Toc529983681)

[**Triggers** 4](#_Toc529983682)

[**Tabla CONSUMO\_ALIMENTO\_TERNERAS** 5](#_Toc529983683)

[**Estructura** 5](#_Toc529983684)

[**Contraints** 5](#_Toc529983685)

[**Tabla CONSUMO\_MEDICAMENTO\_TERNERA** 5](#_Toc529983686)

[**Estructura** 5](#_Toc529983687)

[**Constraints** 6](#_Toc529983688)

[**Tabla Enfermedad\_Ternera** 6](#_Toc529983689)

[**Estructura** 6](#_Toc529983690)

[**Constraints** 6](#_Toc529983691)

[**Tabla ENFERMEDADES** 7](#_Toc529983692)

[**Estructura** 7](#_Toc529983693)

[**Constraints** 7](#_Toc529983694)

[**Triggers** 7](#_Toc529983695)

[**Tabla GUACHERAS** 7](#_Toc529983696)

[**Estructura** 7](#_Toc529983697)

[**Constraints** 8](#_Toc529983698)

[**Triggers** 8](#_Toc529983699)

[**Tabla MADRES** 8](#_Toc529983700)

[**Estructura** 8](#_Toc529983701)

[**Constraints** 8](#_Toc529983702)

[**Triggers** 8](#_Toc529983703)

[**Tabla MEDICAMENTOS** 9](#_Toc529983704)

[**Estructura** 9](#_Toc529983705)

[**Constraints** 9](#_Toc529983706)

[**Triggers** 9](#_Toc529983707)

[**Tabla PADRES** 9](#_Toc529983708)

[**Estructura** 9](#_Toc529983709)

[**Constraints** 10](#_Toc529983710)

[**Triggers** 10](#_Toc529983711)

[**Tabla PESOS** 10](#_Toc529983712)

[**Estructura** 10](#_Toc529983713)

[**Constraints** 10](#_Toc529983714)

[**Tabla TERNERAS** 11](#_Toc529983715)

[**Estructura** 11](#_Toc529983716)

[**Constraints** 11](#_Toc529983717)

[**Triggers** 12](#_Toc529983718)

[**Tabla UNIDADES** 12](#_Toc529983719)

[**Estructura** 12](#_Toc529983720)

[**Constaints** 12](#_Toc529983721)

[**Triggers** 12](#_Toc529983722)

[**Tabla USUARIOS** 12](#_Toc529983723)

[**Estructura** 12](#_Toc529983724)

[**Constraints** 13](#_Toc529983725)

[**Triggers** 13](#_Toc529983726)

[**Modelo de la base de datos** 13](#_Toc529983727)

[**Base de datos corporativa** 14](#_Toc529983728)

[**DATABASE LINK** 14](#_Toc529983729)

[**D\_ENFERMEDAD** 14](#_Toc529983730)

[**H\_TERN\_ENF** 15](#_Toc529983731)

[**D\_TERNERA** 16](#_Toc529983732)

# **Tabla ALIMENTOS**

## Estructura

DESCRIBE ALIMENTOS

Nombre Nulo Tipo

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

ID\_ALIMENTO NOT NULL NUMBER(2)

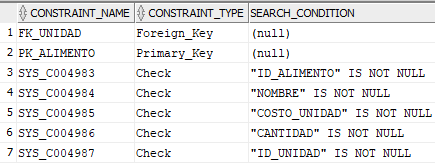
NOMBRE NOT NULL VARCHAR2(25)

COSTO\_UNIDAD NOT NULL NUMBER(5,2)

CANTIDAD NOT NULL NUMBER(5,2)

ID\_UNIDAD NOT NULL NUMBER(38)

## Constraints



## Triggers



# **Tabla CONSUMO\_ALIMENTO\_TERNERAS**

## Estructura

DESCRIBE CONSUMO\_ALIMENTO\_TERNERAS

Nombre Nulo Tipo

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

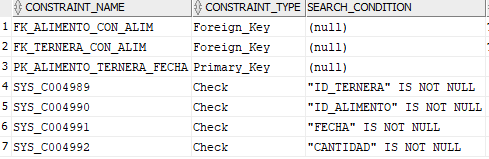
ID\_TERNERA NOT NULL NUMBER(38)

ID\_ALIMENTO NOT NULL NUMBER(38)

FECHA NOT NULL DATE

CANTIDAD NOT NULL NUMBER(5,2)

## Contraints



# **Tabla CONSUMO\_MEDICAMENTO\_TERNERA**

## Estructura

DESCRIBE CONSUMO\_MEDICAMENTO\_TERNERA

Nombre Nulo Tipo

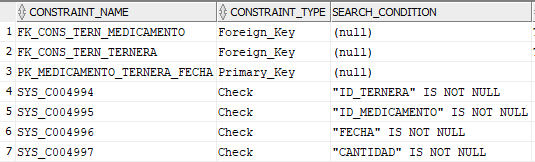
ID\_TERNERA NOT NULL NUMBER(38)

ID\_MEDICAMENTO NOT NULL NUMBER(38)

FECHA NOT NULL DATE

CANTIDAD NOT NULL NUMBER(2,2)

## Constraints



# **Tabla Enfermedad\_Ternera**

## Estructura

DESCRIBE ENFERMEDAD\_TERNERA

Nombre Nulo Tipo

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

ID\_TERNERA NOT NULL NUMBER(38)

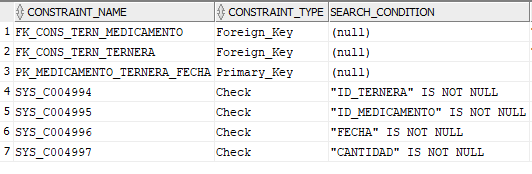
ID\_ENFERMEDAD NOT NULL NUMBER(38)

FECHA\_DESDE NOT NULL DATE

FECHA\_HASTA DATE

OBSERVACION VARCHAR2(250)

## Constraints



# **Tabla ENFERMEDADES**

## Estructura

DESCRIBE ENFERMEDADES

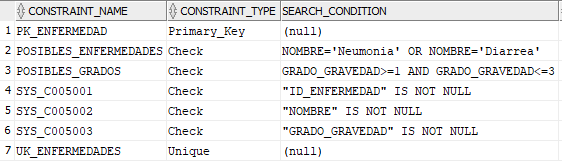
Nombre Nulo Tipo

ID\_ENFERMEDAD NOT NULL NUMBER(38)

NOMBRE NOT NULL VARCHAR2(100)

GRADO\_GRAVEDAD NOT NULL NUMBER(2)

## Constraints



## Triggers



# **Tabla GUACHERAS**

## Estructura

DESCRIBE GUACHERAS

Nombre Nulo Tipo

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

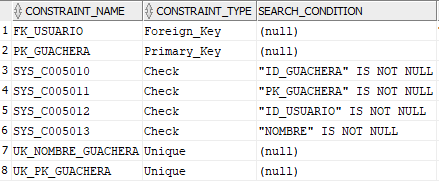
ID\_GUACHERA NOT NULL NUMBER(38)

PK\_GUACHERA NOT NULL VARCHAR2(12)

ID\_USUARIO NOT NULL NUMBER(38)

NOMBRE NOT NULL VARCHAR2(70)

## Constraints



## Triggers



# **Tabla MADRES**

## Estructura

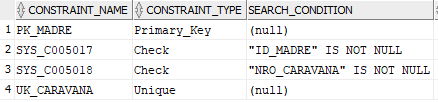
DESCRIBE MADRES

Nombre Nulo Tipo

ID\_MADRE NOT NULL NUMBER(4)

NRO\_CARAVANA NOT NULL NUMBER(10)

## Constraints



## Triggers



# **Tabla MEDICAMENTOS**

## Estructura

DESCRIBE MEDICAMENTOS

Nombre Nulo Tipo

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

ID\_MEDICAMENTO NOT NULL NUMBER(38)

NOMBRE NOT NULL VARCHAR2(100)

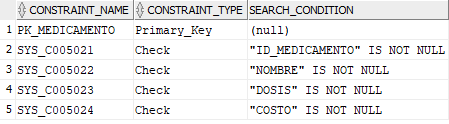
DOSIS NOT NULL VARCHAR2(100)

COSTO NOT NULL VARCHAR2(100)

FECHA\_DESDE DATE

FECHA\_HASTA DATE

## Constraints



## Triggers



# **Tabla PADRES**

## Estructura

DESCRIBE PADRES

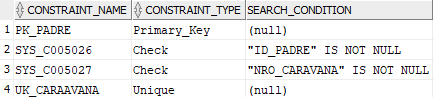
Nombre Nulo Tipo

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

ID\_PADRE NOT NULL NUMBER(4)

NRO\_CARAVANA NOT NULL NUMBER(10)

## Constraints



## Triggers



# **Tabla PESOS**

## Estructura

DESCRIBE PESOS

Nombre Nulo Tipo

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

ID\_PESO NOT NULL NUMBER(38)

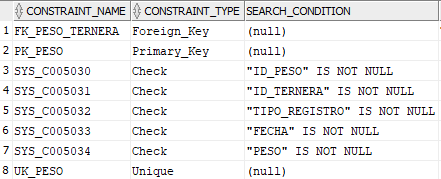
ID\_TERNERA NOT NULL NUMBER(38)

TIPO\_REGISTRO NOT NULL VARCHAR2(50)

FECHA NOT NULL DATE

PESO NOT NULL NUMBER(4,2)

## Constraints



# **Tabla TERNERAS**

## Estructura

DESCRIBE TERNERAS

Nombre Nulo Tipo

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

ID\_TERNERA NOT NULL NUMBER(4)

NRO\_CARAVANA NOT NULL NUMBER(10)

ID\_MADRE NOT NULL NUMBER(38)

ID\_PADRE NOT NULL NUMBER(38)

ID\_GUACHERA NOT NULL NUMBER(38)

FECHA\_NACIMIENTO NOT NULL DATE

PESO\_NACIMIENTO NOT NULL NUMBER(4,2)

RAZA NOT NULL VARCHAR2(12)

PARTO NOT NULL VARCHAR2(12)

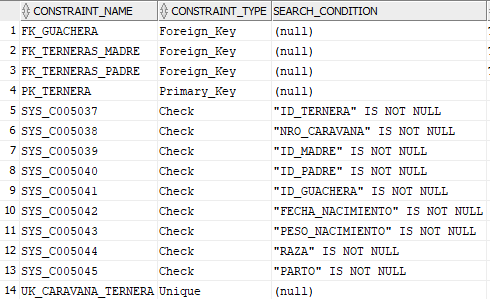
FECHA\_MUERTE DATE

CAUSA\_MUERTE VARCHAR2(500)

FECHA\_BAJA DATE

CAUSA\_BAJA VARCHAR2(500)

## Constraints



## Triggers



# **Tabla UNIDADES**

## Estructura

DESCRIBE UNIDADES

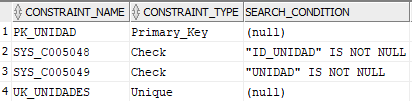
Nombre Nulo Tipo

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

ID\_UNIDAD NOT NULL NUMBER(38)

UNIDAD NOT NULL VARCHAR2(25)

## Constaints



## Triggers



### 

# **Tabla USUARIOS**

## Estructura

DESCRIBE USUARIOS

Nombre Nulo Tipo

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

ID\_USUARIO NOT NULL NUMBER(38)

USUARIO NOT NULL VARCHAR2(100)

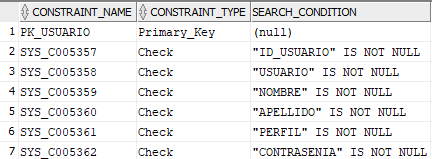
NOMBRE NOT NULL VARCHAR2(50)

APELLIDO NOT NULL VARCHAR2(50)

PERFIL NOT NULL VARCHAR2(50)

CONTRASENIA NOT NULL VARCHAR2(16)

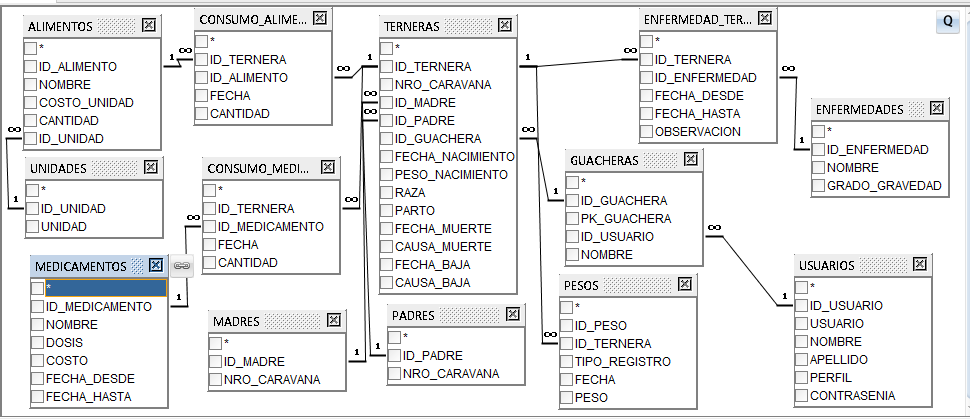
## Constraints



## Triggers



# **Modelo de la base de datos**



# **Base de datos corporativa**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | | | |  |
|  | **Tablas Fuentes en ODS** | | | | **Tablas Fuentes en ODS** | | | **Tablas DWH** |
| **Hecho/Dimensión** | **Fase 1 –Extracción** | | | | **Fase 2 -Transformación** | | | **Fase 3 - Carga** |
| **TERNERA\_ENF** | **H\_TERN\_ENF** | | | | **ODS\_H\_TERN\_ENF** | | | **HT\_TERN\_ENF** |
| **ENFERMEDAD** | **D\_ENFERMEDAD** | | | | **ODS\_D\_ENFERMEDAD** | | | **DT\_ENFERMEDAD** |
| **TERNERA** | **D\_TERNERA** | | | | **ODS\_D\_TERNERA** | | | **DT\_TERNERA** |
| **TIEMPO** |  | | | |  | | | **DT\_TIEMPO** |
|  |  | | | |  | | |  |
|  | | |  |

## Extracción de datos

### DATABASE LINK

Es una forma de realizar una conexión unidireccional entre dos bases de datos con el fin de poder accede a los objetos de la otra bd.

CREATE DATABASE LINK DBL\_COGNC

CONNECT TO TAMBO IDENTIFIED BY VALUES 'TAMBO'

USING 'localhost:1521/xe';

### D\_ENFERMEDAD

**--Crear Tabla**

CREATE TABLE D\_ENFERMEDAD

(ENFERMEDAD VARCHAR(40))

TABLESPACE "USERS";

**--Insertar datos desde la bd usando db Link**

INSERT INTO D\_ENFERMEDAD(ENFERMEDAD) SELECT DISTINCT H\_TERN\_ENF.ENFERMEDAD FROM H\_TERN\_ENF

WHERE NOT EXIST(SELECT 1

FROM D\_ENFERMEDAD

WHERE UPPER(H\_TERN\_ENF.ENFERMEDAD)=UPPER(D\_ENFERMEDAD.ENFERMEDAD););

**--Creación de las vistas**

CREATE OR REPLACE VIEW V\_CONTROL\_D\_ENFERMEDAD

AS

SELECT DISTINCT ENFERMEDAD

FROM H\_TERN\_ENF

WHERE

NOT EXISTS

(

SELECT 1

FROM D\_ENFERMEDAD

WHERE UPPER (H\_TERN\_ENF.ENFERMEDAD)=UPPER(D\_ENFERMEDAD.ENFERMEDAD) );

### H\_TERN\_ENF

**--Creando tabla**

CREATE TABLE H\_TERN\_ENF

(

TERNERA NUMBER(11,0),

ENFERMEDAD VARCHAR(30),

FECH\_DESDE DATE,

FECH\_HASTA DATE

)

TABLESPACE "USERS";

**--Insertar datos desde la bd usando db Link**

INSERT INTO H\_TERN\_ENF(TERNERA,ENFERMEDAD,FECH\_DESDE,FECH\_HASTA) (SELECT TAMBO.TERNERAS.NRO\_CARAVANA,

TAMBO.ENFERMEDADES.NOMBRE,

TAMBO.ENFERMEDAD\_TERNERA.FECHA\_DESDE,

TAMBO.ENFERMEDAD\_TERNERA.FECHA\_HASTA

FROM TAMBO.ENFERMEDAD\_TERNERA

INNER JOIN TAMBO.TERNERAS

ON TAMBO.TERNERAS.ID\_TERNERA = TAMBO.ENFERMEDAD\_TERNERA.ID\_TERNERA

INNER JOIN TAMBO.ENFERMEDADES

ON TAMBO.ENFERMEDADES.ID\_ENFERMEDAD = TAMBO.ENFERMEDAD\_TERNERA.ID\_ENFERMEDAD)

Esta inserción está hecha con join porque requiero traer el número de caravana de las terneras enfermas y el nombre de la enfermedad de la que esta enferma.

### D\_TERNERA

**--Creando tabla**

CREATE TABLE D\_TERNERA

(TERNERA NUMBER(11,0),

DESCRIPCION VARCHAR(30) DEFAULT ‘NRO CARABANA’)

TABLESPACE “USERS”

**--Insertar datos desde la bd usando db Link**

INSERT INTO D\_TERNERA(TERNERA) SELECT DISTINCT H\_TERN\_ENF.TERNERA FROM H\_TERN\_ENF

WHERE NOT EXISTS(

SELECT 1

FROM D\_TERNERA

WHERE UPPER(H\_TERN\_ENF.TERNERA)=UPPER(D\_TERNERA.TERNERA));

**--Creando vista**

CREATE OR REPLACE VIEW V\_CONTROL\_D\_TERNERA

AS

SELECT DISTINCT TERNERA

FROM H\_TERN\_ENF

WHERE

EXISTS

(

SELECT 1

FROM D\_TERNERA

WHERE UPPER (H\_TERN\_ENF.TERNERA)=UPPER(D\_TERNERA.TERNERA)

);

## Transformación de los datos

### Tabla ODS\_ENFERMEDAD

**--Creando la tabla**

CREATE TABLE ODS\_D\_ENFERMEDAD

(

ENFERMEDAD\_PK VARCHAR(40)

)

TABLESPACE "USERS";

**--Carga de datos desde D\_ENFERMEDAD a ODS\_D\_ENFERMEDAD**

CREATE OR REPLACE PROCEDURE P\_ODS\_D\_ENFEREDAD

IS

BEGIN

DELETE FROM ODS\_D\_ENFERMEDAD;

INSERT INTO ODS\_D\_ENFERMEDAD

( ENFERMEDAD\_PK)

SELECT UPPER(TRIM(ENFERMEDAD))

FROM D\_ENFERMEDAD;

COMMIT;

END P\_ODS\_D\_ENFEREDAD;

### Tabla ODS\_D\_TERNERA

--**Creando la tabla**

CREATE TABLE ODS\_D\_TERNERA

(TERNERA\_PK VARCHAR(40),

DESC\_TERNERA VARCHAR(30))

TABLESPACE "USERS";

**--Carga de datos desde D\_TERNERA a ODS\_D\_TERNERA**

CREATE OR REPLACE PROCEDURE P\_ODS\_D\_TERNERA

IS

BEGIN

DELETE FROM ODS\_D\_TERNERA;

INSERT INTO ODS\_D\_TERNERA

( TERNERA\_PK, DESC\_TERNERA)

SELECT UPPER(TRIM(TERNERA)),

DESCRIPCION

FROM D\_TERNERA;

COMMIT;

END P\_ODS\_D\_TERNERA;

### Tabla ODS\_H\_TERN\_ENF

**--Creando la tabla**

CREATE TABLE ODS\_H\_TERN\_ENF

(TERNERA NUMBER(11),

ENFEREMEDAD VARCHAR(40),

FECHA\_DESDE INTEGER,

FECHA\_HASTA INTEGER)

TABLESPACE "USERS";

**--Carga de datos de ODS\_H\_TERN\_ENF desde H\_TERN\_ENF**

CREATE OR REPLACE PROCEDURE P\_ODS\_H\_TERN\_ENF

IS

BEGIN

DELETE FROM ODS\_H\_TERN\_ENF;

INSERT INTO ODS\_H\_TERN\_ENF

( TERNERA, ENFERMEDAD, FECHA\_DESDE, FECHA\_HASTA)

SELECT UPPER(TRIM(TERNERA)),

UPPER(TRIM(ENFERMEDAD)),

to\_number(TO\_CHAR(FECH\_DESDE,'YYYYMMDD')),

to\_number(TO\_CHAR(FECH\_HASTA,'YYYYMMDD'))

FROM H\_TERN\_ENF;

COMMIT;

END P\_ODS\_H\_TERN\_ENF;

## Carga de los datos en el DWH

### Tabla DT\_TERNERA

**--Creando la tabla**

CREATE TABLE DT\_ENFERMEDAD

(SK\_TERNERA INTEGER PRIMARY KEY,

PK\_TERNERA VARCHAR(40))

TABLESPACE "USERS";

**--Secuencia para DT\_Ternera**

CREATE SEQUENCE SEQ\_TERNERA INCREMENT BY 1 START WITH 1;

**--Ttigger para la secuencia de ternera**

CREATE OR REPLACE TRIGGER T\_TERNERA\_SEQ

BEFORE INSERT

ON DT\_TERNERA

FOR EACH ROW

BEGIN

SELECT SEQ\_TERNERA.NEXTVAL

INTO :NEW.SK\_TERNERA

FROM DUAL;

END T\_TERNERA\_SEQ;

**--Procedimiento para la carga de datos desde ODS\_D\_TERNERA**

CREATE OR REPLACE PROCEDURE P\_DWH\_DT\_TERNERA

IS

BEGIN

MERGE INTO DT\_TERNERA a

USING (SELECT TERNERA\_PK,DESC\_TERNERA FROM ODS\_D\_TERNERA) o

ON (a.PK\_TERNERA = o.TERNERA\_PK)

WHEN NOT MATCHED THEN

INSERT (PK\_TERNERA, DES\_TER)

VALUES (o.TERNERA\_PK, o.DESC\_TERNERA);

COMMIT;

END P\_DWH\_DT\_TERNERA;

### Tabla DT\_ENFERMEDAD

**--Creando la tabla**

CREATE TABLE DT\_ENFERMEDAD

(SK\_ENFERMEDAD INTEGER PRIMARY KEY,

PK\_ENFERMEDAD VARCHAR(40))

TABLESPACE "USERS";

**--Secuencia para DT\_ENFERMEDAD**

CREATE SEQUENCE SEQ\_ENFERMEDAD INCREMENT BY 1 START WITH 1;

**--Trigger para secuencia de DT\_ENFERMEDAD**

CREATE OR REPLACE TRIGGER T\_ENFERMEDAD\_SEQ

BEFORE INSERT

ON DT\_ENFERMEDAD

FOR EACH ROW

BEGIN

SELECT SEQ\_ENFERMEDAD.NEXTVAL

INTO :NEW.SK\_ENFERMEDAD

FROM DUAL;

END T\_ENFERMEDAD\_SEQ;

**--Procedimiento para carga de datos desde ODS\_D\_ENFERMEDAD**

CREATE OR REPLACE PROCEDURE P\_DWH\_DT\_ENFERMEDAD

IS BEGIN MERGE INTO DT\_ENFERMEDAD a

USING (SELECT ENFERMEDAD\_PK FROM ODS\_D\_ENFERMEDAD) o

ON (a.PK\_ENFERMEDAD = o.ENFERMEDAD\_PK)

WHEN NOT MATCHED THEN

INSERT (PK\_ENFERMEDAD)

VALUES

(o.ENFERMEDAD\_PK);

COMMIT;

END P\_DWH\_DT\_ENFERMEDAD;

### DT\_TIEMPO

CREATE TABLE DT\_TIEMPO

(SK\_TIEMPO INTEGER PRIMARY KEY,

FECHA DATE,

ANIO INTEGER)

TABLESPACE "USERS";

**--Procedimiento de carga de datos**

CREATE OR REPLACE PROCEDURE P\_DWT\_DT\_TIEMPO

is

FECHA\_DESDE DATE;

FECHA\_HASTA DATE;

BEGIN

FECHA\_DESDE :=TO\_DATE('20160101','YYYYMMDD');

FECHA\_HASTA :=TO\_DATE('20191230','YYYYMMDD');

WHILE FECHA\_DESDE <= FECHA\_HASTA

LOOP

INSERT INTO DT\_TIEMPO

(SK\_TIEMPO,FECHA,ANIO)

VALUES

(TO\_CHAR(FECHA\_DESDE,'YYYYMMDD'),

FECHA\_DESDE,

TO\_CHAR(FECHA\_DESDE,'YYYY'));

FECHA\_DESDE:=FECHA\_DESDE+1;

END LOOP;

COMMIT ;

END P\_DWT\_DT\_TIEMPO;

### Tabla HT\_TERN\_ENF

**--Creando la tabla**

CREATE TABLE HT\_TERN\_ENF

(SK\_ENNFERMEDAD INTEGER NOT NULL,

SK\_TERNERA INTEGER NOT NULL,

SK\_TIEMPO INTEGER NOT NULL,

CANTIDAD INTEGER NOT NULL,

CONSTRAINT FK\_HTTERNENF\_DTENF FOREIGN KEY (SK\_ENFERMEDAD)

REFERENCES DT\_ENFERMEDAD(SK\_ENFERMEDAD),

CONSTRAINT FK\_HTTERNENF\_DTTIEMPO FOREIGN KEY (SK\_TIEMPO)

REFERENCES DT\_TIEMPO(SK\_TIEMPO)

CONSTRAINT FK\_HTTERNENF\_TERNERA FOREIGN KEY (SK\_TERNERA)

REFERENCES DT\_TERNERA (SK\_TERNERA))

TABLESPACE "USERS";

**--Añadiendo la PK**

ALTER TABLE HT\_TERN\_ENF

ADD CONSTRAINT PK\_HTTERNENF

(SK\_ENFERMEDAD, SK\_TERNERA,SK\_TIEMPO);

**--Columnas para auditoria**

ALTER TABLE HT\_TERN\_ENF ADD

(ALTA\_USUARIO VARCHAR(50),

ALTER\_FECHA DATE);

**--Trigger para auditoria**

CREATE OR REPLACE TRIGGER T\_AUD\_HT\_TERN\_ENF

BEFORE INSERT OR UPDATE

ON HT\_TERN\_ENF

FOR EACH ROW

BEGIN

SELECT USER,

SYSDATE

INTO :NEW.ALTA\_USUARIO,

:NEW.ALTER\_FECHA

FROM DUAL;

END T\_AUD\_HT\_TERN\_ENF;

**--Procedimiento de carga de datos**

CREATE OR REPLACE PROCEDURE P\_HT\_TERN\_ENF

IS

BEGIN

INSERT INTO HT\_TERN\_ENF

(SK\_ENFERMEDAD,SK\_TERNERA,

SK\_TIEMPO,CANTIDAD

)

SELECT DT\_ENFERMEDAD.SK\_ENFERMEDAD,

DT\_TERNERA.SK\_TERNERA,

DT\_TIEMPO.SK\_TIEMPO,

1 CANTIDAD

FROM ODS\_H\_TERN\_ENF

LEFT OUTER JOIN DT\_ENFERMEDAD

ON DT\_ENFERMEDAD.PK\_ENFERMEDAD=ODS\_H\_TERN\_ENF.ENFERMEDAD

LEFT OUTER JOIN DT\_TERNERA

ON DT\_TERNERA.PK\_TERNERA=ODS\_H\_TERN\_ENF.TERNERA

LEFT OUTER JOIN DT\_TIEMPO

ON DT\_TIEMPO.SK\_TIEMPO=ODS\_H\_TERN\_ENF.FECHA\_DESDE;

COMMIT;

END P\_HT\_TERN\_ENF;

## Generando reporte en Excel

Se creo una conexión entre Excel y la base de datos corporativa y con esa conexión se creo una tabla dinámica para generar el reporte

