**Universidad ORT Uruguay Facultad de Ingeniería Escuela de Tecnología**

OBLIGATORIO – BASE DE DATOS 2 DOCUMENTACION.



# Luis Ignacio López Perdomo – 278179



Ana Karelina Fabra – 222487

## Grupo: N3B Docente: Fernando Thul

**Analista en Tecnologías de la Información**

## Fecha de entrega del documento: 09-06-2022

Indicie

[REQUERIMIENTOS FUNCIONALES 3](#_bookmark0)

1. [DDL 3](#_bookmark1)
2. [DML 3](#_bookmark2)
3. [PRODEDIMIENTOS Y FUNCIONES 3](#_bookmark3)
4. [TRIGGERS 3](#_bookmark4)
5. DLL

Con estos datos se solicita, utilizando el lenguaje DDL, crear las tablas que permitan representar esta realidad teniendo en cuenta que es requisito EXCLUYENTE que las tablas estén por lo menos en tercera forma normal (3NF), la inclusión del MER en este punto será valorada pero no es un requisito obligatorio, crear todas las restricciones de integridad que necesita el modelo, teniendo en cuenta también las restricciones no estructurales (RNE) que permitan cumplir con las reglas del negocio, debe también crear los índices necesarios de acuerdo al criterio visto en clase.

Se crea la Base y las tablas con las restricciones, solicitadas.

CREATE DATABASE Pflanze

USE Pflanze

CREATE TABLE PLANTA (Planta\_ID INT IDENTITY(1,1) NOT NULL,

Nombre VARCHAR(50) NOT NULL,

Fecha\_Nac DATE NOT NULL,

Altura NUMERIC,

Fecha\_Ult\_Medicion DATETIME ,

Precio MONEY,

CONSTRAINT PK\_Planta PRIMARY KEY(Planta\_ID),

CONSTRAINT CK\_Planta\_Altura CHECK(Altura BETWEEN 0 AND 12000)

)

CREATE TABLE TAGS (Tag\_Id INT NOT NULL IDENTITY(1,1) ,

Tag\_Nombre VARCHAR(50),

CONSTRAINT PK\_Tags PRIMARY KEY(Tag\_Id)

)

CREATE TABLE TAGS\_DE\_PLANTAS (Tag\_Id INT NOT NULL,

Planta\_ID INT NOT NULL,

CONSTRAINT FK\_Tags\_De\_Plantas\_Planta\_ID FOREIGN KEY(Planta\_ID) REFERENCES Planta(Planta\_Id),

CONSTRAINT FK\_Tags\_De\_Plantas\_Tag\_Id FOREIGN KEY(Tag\_Id) REFERENCES Tags(Tag\_Id),

)

CREATE TABLE MANTENIMIENTO (Mantenimiento\_ID INT IDENTITY(1,1) NOT NULL,

Planta\_ID INT NOT NULL,

Fecha\_Hora\_Mant DATETIME NOT NULL,

Descripcion VARCHAR(60) NOT NULL,

Tipo\_Mantenimiento CHARACTER(1) NOT NULL ,

Horas\_Empleadas NUMERIC(10,2) ,

Costo\_Mantenimiento MONEY ,

CONSTRAINT PK\_Mantenimiento PRIMARY KEY(Mantenimiento\_ID),

CONSTRAINT FK\_Mantenimiento\_Planta\_ID FOREIGN KEY(Planta\_ID) REFERENCES Planta(Planta\_Id),

CONSTRAINT CK\_Tipo\_Mantenimiento CHECK(Tipo\_Mantenimiento IN('O','N'))

)

CREATE TABLE PRODUCTO ( Codigo\_Producto CHARACTER(5) NOT NULL,

Descripcion\_Producto VARCHAR(60) NOT NULL,

Precio\_Producto MONEY NOT NULL,

CONSTRAINT PK\_Productos PRIMARY KEY(Codigo\_Producto),

CONSTRAINT UK\_Productos UNIQUE(Descripcion\_Producto)

)

CREATE TABLE PRODUCTOS\_USADOS (Mantenimiento\_ID INT NOT NULL,

Codigo\_Producto CHARACTER(5) NOT NULL,

Cantidad\_Producto NUMERIC(10,2) ,

Costo\_Apl\_Producto MONEY,

CONSTRAINT FK\_Productos\_Usados\_Mantenimiento\_ID FOREIGN KEY(Mantenimiento\_ID) REFERENCES Mantenimiento(Mantenimiento\_ID),

CONSTRAINT FK\_Productos\_Usados\_Codigo\_Producto FOREIGN KEY(Codigo\_Producto) REFERENCES Producto(Codigo\_Producto),

)

1. DML

Utilizando el lenguaje SQL realizar las siguientes consultas:

1. Mostrar Nombre de Planta y Descripción del Mantenimiento para el último(s) mantenimiento hecho en el año actual

SELECT P.NOMBRE, M.DESCRIPCION

FROM PLANTA P join MANTENIMIENTO M

ON P.PLANTA\_ID=M.PLANTA\_ID

WHERE DATEPART(YEAR,M.FECHA\_HORA\_MANT)=DATEPART(YEAR,GETDATE())

AND M.FECHA\_HORA\_MANT=(SELECT MAX(FECHA\_HORA\_MANT)

FROM MANTENIMIENTO)

1. Mostrar la(s) plantas que recibieron más cantidad de mantenimientos

SELECT P.\*

FROM PLANTA P , (SELECT PLANTA\_ID FROM MANTENIMIENTO

GROUP BY Planta\_ID

HAVING COUNT(\*)= (SELECT MAX(MR.MANTENIMIENTOS\_REALIZADOS)

FROM (SELECT PLANTA\_ID, COUNT(MANTENIMIENTO\_ID) AS MANTENIMIENTOS\_REALIZADOS

FROM MANTENIMIENTO

GROUP BY PLANTA\_ID) MR)) PMR

WHERE P.PLANTA\_ID=PMR.PLANTA\_ID

1. Mostrar las plantas que este año ya llevan más de un 20% de costo de mantenimiento que el costo de mantenimiento de todo el año anterior para la misma planta ( solo considerar plantas nacidas en el año 2019 o antes)

SELECT P.\*

FROM PLANTA P, ( SELECT ANIO\_ACTUAL.PLANTA\_ID

FROM

(SELECT PLANTA\_ID, SUM(COSTO\_MANTENIMIENTO) COSTO\_TOTAL, DATEPART(YEAR,FECHA\_HORA\_MANT) ANIO\_MANT

FROM MANTENIMIENTO

WHERE DATEPART(YEAR,FECHA\_HORA\_MANT) = DATEPART(YEAR,GETDATE())

GROUP BY PLANTA\_ID, DATEPART(YEAR,FECHA\_HORA\_MANT)) ANIO\_ACTUAL,

(SELECT PLANTA\_ID, SUM(COSTO\_MANTENIMIENTO) COSTO\_TOTAL, DATEPART(YEAR,FECHA\_HORA\_MANT) ANIO\_MANT

FROM MANTENIMIENTO

WHERE DATEPART(YEAR,FECHA\_HORA\_MANT) = DATEPART(YEAR,GETDATE()) -1

GROUP BY PLANTA\_ID, DATEPART(YEAR,FECHA\_HORA\_MANT)) ANIO\_ANTERIOR

WHERE ANIO\_ACTUAL.PLANTA\_ID= ANIO\_ANTERIOR.PLANTA\_ID

AND ANIO\_ACTUAL.COSTO\_TOTAL > (ANIO\_ANTERIOR.COSTO\_TOTAL \* 20) /100 + ANIO\_ANTERIOR.COSTO\_TOTAL) PL\_AUMENTO\_COSTO

WHERE P.PLANTA\_ID = PL\_AUMENTO\_COSTO.PLANTA\_ID

AND DATEPART(YEAR,P.FECHA\_NAC) <= 2019

1. Mostrar las plantas que tienen el tag “FRUTAL”, a la vez tienen el tag “PERFUMA” y no tienen el tag “TRONCOROTO”. Y que adicionalmente miden medio metro de altura o más y tienen un precio de venta establecido

SELECT \*

FROM PLANTA P, (SELECT TDP.PLANTA\_ID

FROM TAGS\_DE\_PLANTAS TDP, TAGS T

WHERE T.TAG\_ID= TDP.TAG\_ID

AND T.TAG\_NOMBRE='FRUTAL'

INTERSECT

SELECT TDP.PLANTA\_ID

FROM TAGS\_DE\_PLANTAS TDP, TAGS T

WHERE T.TAG\_ID= TDP.TAG\_ID

AND T.TAG\_NOMBRE='PERFUMA'

EXCEPT

SELECT TDP.PLANTA\_ID

FROM TAGS\_DE\_PLANTAS TDP, TAGS T

WHERE T.TAG\_ID= TDP.TAG\_ID

AND T.TAG\_NOMBRE='TRONCOROTO') TAGS\_PLANTAS

WHERE P.PLANTA\_ID= TAGS\_PLANTAS.PLANTA\_ID

1. Mostrar las Plantas que recibieron mantenimientos que en su conjunto incluyen todos los productos existentes

--OPCION A , UTILIZANDO TRIGGER DE RESTRICCION DE MANTENIMIENTO\_ID Y CODIGO\_PRODUCTO

SELECT P.\*

FROM PLANTA P, (SELECT M.PLANTA\_ID

FROM MANTENIMIENTO M, (SELECT MANTENIMIENTO\_ID

FROM PRODUCTOS\_USADOS

GROUP BY MANTENIMIENTO\_ID

HAVING COUNT(CODIGO\_PRODUCTO)=(SELECT COUNT(\*) FROM PRODUCTO)) MANT\_ID

WHERE M.MANTENIMIENTO\_ID= MANT\_ID.MANTENIMIENTO\_ID) PLANT\_ID

WHERE P.PLANTA\_ID= PLANT\_ID.PLANTA\_ID

-- OPCION B , SINO NO HAY CONTROL DE DATOS DUPLICADOS EN LA TABLA PRODUCTOS USADOS

SELECT P.\*

FROM PLANTA P , (SELECT M.PLANTA\_ID

FROM MANTENIMIENTO M, (SELECT PU.MANTENIMIENTO\_ID

FROM (SELECT DISTINCT CODIGO\_PRODUCTO,MANTENIMIENTO\_ID

FROM Productos\_Usados ) PU

GROUP BY Mantenimiento\_ID

HAVING COUNT(PU.CODIGO\_PRODUCTO)= (SELECT COUNT(\*)

FROM PRODUCTO)) MP

WHERE M.MANTENIMIENTO\_ID=MP.MANTENIMIENTO\_ID) MANT\_ID

WHERE P.PLANTA\_ID= MANT\_ID.PLANTA\_ID

1. Para cada Planta con 2 años de vida o más y con un precio menor a 200 dólares: sumarizar su costo de Mantenimiento total ( contabilizando tanto mantenimientos de tipo “OPERATIVO” como de tipo “NUTRIENTES”) y mostrar solamente las plantas que su costo sumarizado es mayor que 100 dólares.

SELECT P.\*, CM.TOTAL\_MANTENIMIENTO

FROM Planta P,(SELECT SUM(COSTO\_MANTENIMIENTO) TOTAL\_MANTENIMIENTO,PLANTA\_ID

FROM MANTENIMIENTO

GROUP BY PLANTA\_ID) CM

WHERE P.PLANTA\_ID=CM.PLANTA\_ID

AND CM.TOTAL\_MANTENIMIENTO > 151

AND P.PRECIO >200

AND DATEPART(YEAR,P.FECHA\_NAC) <= DATEPART(YEAR,GETDATE())-2

1. PRODEDIMIENTOS Y FUNCIONES

USE Pflanze

-- Parte a

CREATE PROCEDURE UDP\_AumentarCostosPlanta

@Id\_Planta INT,

@Porcentaje NUMERIC(10,2),

@Fecha\_Inicial DATE,

@Fecha\_Final DATE

AS

BEGIN

Select Planta\_ID,(Costo\_Mantenimiento \* @Porcentaje) /100 AS AUMENTO

from Mantenimiento

where @Id\_Planta= Planta\_ID and CAST(Fecha\_Hora\_Mant AS DATE) between @Fecha\_Inicial and @Fecha\_Final

Update Mantenimiento

set Costo\_Mantenimiento = (Costo\_Mantenimiento \* @Porcentaje) /100 + Costo\_Mantenimiento

where @Id\_Planta= Planta\_ID and CAST(Fecha\_Hora\_Mant AS DATE) between @Fecha\_Inicial and @Fecha\_Final

END

EXECUTE UDP\_AumentarCostosPlanta 1,10 ,'2018-07-05' ,'2018-07-07'

--Parte B

CREATE FUNCTION UDF\_CalculaPromedioMantOperativos(@year int)

RETURNS INT

AS

BEGIN

DECLARE @promedio money

Select @promedio= AVG(Costo\_Mantenimiento)

from Mantenimiento

where DATEPART( YEAR, Fecha\_Hora\_Mant )=@year

and Tipo\_Mantenimiento='O'

RETURN @promedio

END

Select dbo.UDF\_CalculaPromedioMantOperativos(2015) as Promedio\_Anual

-- EXTRAS

-- CALCULA EL COSTO DEL MANTENIMIENTO PARA LOS TIPO DE MANTENIMIENTO NUTRIENTE

CREATE PROCEDURE UDP\_CostoMantenimientoNutrientes

AS

BEGIN

Update Mantenimiento

set Costo\_Mantenimiento= Suma\_Costo

from

(Select sum(PU.Costo\_Apl\_Producto) as Suma\_Costo,M.Mantenimiento\_ID as ID\_SC

from Mantenimiento M join Productos\_Usados PU

on M.Mantenimiento\_ID=PU.Mantenimiento\_ID

where M.Tipo\_Mantenimiento='N'

group by M.Mantenimiento\_ID) AS SC

where Mantenimiento\_id=SC.ID\_SC

END

1. TRIGGERS

-- EJERCICIO 4

-- Base de dato para Auditorias

--PARTE A

CREATE DATABASE Auditorias

USE Auditorias

CREATE TABLE Auditorias\_Produtos ( Codigo\_producto char(5),

Descripcion\_Producto varchar(60),

Precio\_Produto Numeric(10,2),

Fecha\_Modificacion Datetime,

Nombre\_Usuario varchar(60),

Evento varchar(10),

Codigo\_producto\_ant char(5),

Descripcion\_Producto\_ant varchar(60),

Precio\_Produto\_ant char(10)

)

Use Pflanze

CREATE TRIGGER Auditoria\_Producto

ON Producto

AFTER DELETE, UPDATE

AS

BEGIN

If EXISTS( Select \* FROM deleted) AND NOT EXISTS (Select \* from inserted)

INSERT INTO Auditorias.dbo.Auditorias\_Produtos

Select \*,getDate(),host\_name(),'DELETE','','',''

from deleted

If EXISTS( Select \* FROM deleted) AND EXISTS (Select \* from inserted)

INSERT INTO Auditorias.dbo.Auditorias\_Produtos

Select i.\*,getDate(),host\_name(),'UPDATE',d.Codigo\_Producto,d.Descripcion\_Producto,d.Precio\_Producto

from deleted d, inserted i

END

--Parte B

CREATE TRIGGER ControlarNuevoMantenimiento

ON MANTENIMIENTO

INSTEAD OF insert

AS

BEGIN

INSERT INTO Mantenimiento

SELECT I.Planta\_ID, I.Fecha\_Hora\_Mant,I.Descripcion,I.Tipo\_Mantenimiento,I.Horas\_Empleadas, I.Costo\_Mantenimiento

FROM inserted I,Planta P

WHERE

I.Planta\_ID=P.Planta\_ID AND

I.Fecha\_Hora\_Mant > P.Fecha\_Nac

END

-- PARTE ADICIONAL

-- CONTROLA QUE INSERT DE MANTINIMIENTO NO SEA NULO CUANDO EL TIPO DE MANTENIMIENTO ES 'N' NUTRIENTES

--SOLO SE PUEDE CREAR UN TRIGGER POR TABLA PERO LO PONEMOS APARTE PORQUE NO ERA REQUERIMIENTO DEL OBLIGATORIO ESTE TRIGGER

CREATE TRIGGER ControlarNuevoMantenimientoNutrientes

ON MANTENIMIENTO

INSTEAD OF insert

AS

BEGIN

INSERT INTO Mantenimiento

SELECT I.Planta\_ID, I.Fecha\_Hora\_Mant,I.Descripcion,I.Tipo\_Mantenimiento,I.Horas\_Empleadas, I.Costo\_Mantenimiento

FROM inserted I,Planta P

WHERE

I.Planta\_ID=P.Planta\_ID AND

I.Fecha\_Hora\_Mant > P.Fecha\_Nac

EXCEPT

SELECT I.Planta\_ID, I.Fecha\_Hora\_Mant,I.Descripcion,I.Tipo\_Mantenimiento,I.Horas\_Empleadas, I.Costo\_Mantenimiento

FROM inserted I,Planta P

WHERE

I.Planta\_ID=P.Planta\_ID AND

I.Fecha\_Hora\_Mant > P.Fecha\_Nac AND

I.Tipo\_Mantenimiento='O' AND I.Costo\_Mantenimiento= ''

END