**Instituto Tecnológico Colonia**

**CTC**

**Obligatorio Base de Datos 2: Taller mecánico**

**Bruno Ríos**

**Matías Erramouspe**

**Tutor: Gustavo Erramouspe**

**Año: 2022**

Índice

[Lectura del problema 3](#_Toc107476331)

[Punto 1: 6](#_Toc107476332)

[Creación de tablas: 6](#_Toc107476333)

[Creación de triggers: 10](#_Toc107476334)

[Creación de índices: 14](#_Toc107476335)

[Ingreso de datos: 15](#_Toc107476336)

[Ejemplos de ingresos de datos erróneos 19](#_Toc107476337)

[Punto 2: 21](#_Toc107476338)

[a) 21](#_Toc107476339)

[b) 21](#_Toc107476340)

[C) 22](#_Toc107476341)

[D) 22](#_Toc107476342)

[E) 23](#_Toc107476343)

[Punto 3: 24](#_Toc107476344)

[PUNTO 4 24](#_Toc107476345)

[A) 24](#_Toc107476346)

[B) 25](#_Toc107476347)

[C) 25](#_Toc107476348)

[D) 26](#_Toc107476349)

[E) 26](#_Toc107476350)

[PUNTO 5 27](#_Toc107476351)

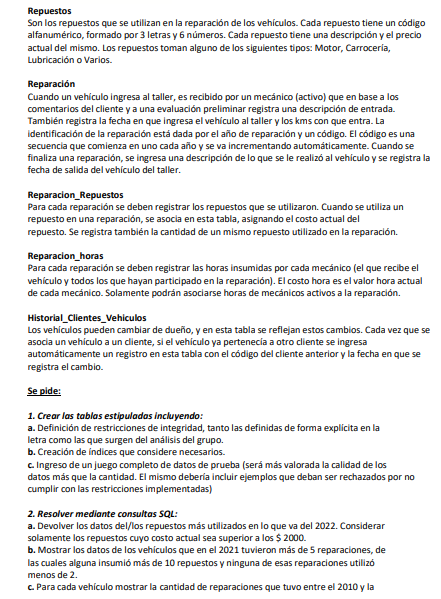
[A) 27](#_Toc107476352)

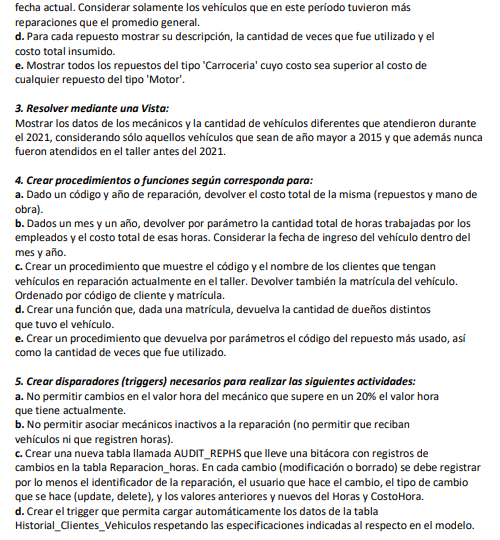
[B) 28](#_Toc107476353)

[C) 29](#_Toc107476354)

[D) 30](#_Toc107476355)

# Lectura del problema

****

****

# Punto 1:

## Creación de tablas:

a)

**Tabla Clientes:**

create table Clientes (

CliCod integer identity(1,1) primary key,

CliNom varchar(30) not null,

CliCi varchar(11) unique not null check(CliCi like '[1-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9][-][0-9]'),

CliTel varchar (9) not null,

CliDir varchar(30) not null,

CliMail varchar(40) unique,

FchRegistro date

);

En esta tabla se ingresan los clientes, en esta tabla el CliCod es la clave primaria y el código generado es identitiy empieza en 1 y por cada nuevo registro se suma 1 más, por lo que el CliCod se genera solo, después se tiene otros atributos que componen clientes, otro atributo a destacar es CliCi en el que se tiene que dar 11 caracteres, en este formato 2.321.246-0, y este no se puede repetir igual que CliMail. Todos los atributos que posean not null, significa que a la hora de insertar estos datos son obligatorios

**Tabla Vehiculos:**

create table Vehiculos(

VehiculoCod integer identity(1,1) primary key,

Matricula varchar(7) unique not null check(Matricula like '[A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9]'),

Marca varchar(20) not null,

Modelo varchar(20) not null,

Anio numeric(4) not null check(Anio >= 1990),

Color varchar(20) not null

);

La tabla Vehiculo se genera solo por el identity, la matricula son 7 datos en este formato 'RAW1435', además este no se puede repetir y no puede ser null. El atributo Anio no puede ser null y el valor tiene que ser igual a mayor a 1990

**Tabla Clientes\_Vehiculos:**

create table Clientes\_Vehiculos(

CliCod integer references Clientes(CliCod),

VehiculoCod integer references Vehiculos(VehiculoCod) ,

Fecha date not null,

primary key (CliCod, VehiculoCod)

);

Esta tabla representa la relación entre Clientes y Vehiculos, muestra a que Cliente pertenece cada vehiculo, en esta tabla un solo Vehiculo le pertenece a un solo cliente, si se detecta que el vehiculo ingresado ya existe en esta tabla se cambia el anterior cliente al con el nuevo cliente.

En esta tabla tanto CliCod y VehiculoCod hacen son claves foráneas de otras tablas, y estos dos atributos hacen la clave primaria de esta clave.

**Tabla Mecanicos:**

create table Mecanicos(

MecCod integer identity(1,1) primary key,

MecNom varchar(30) not null,

MecCi varchar(11) unique check(MecCi like '[1-9][.][0-9][0-9][0-9][.][0-9][0-9][0-9][-][0-9]'),

MecTel varchar(9),

MecFchIng date,

MecValorHora numeric(10) check(MecValorHora > 0),

MecActivo varchar(1) check(MecActivo in ('S','N'))

);

En esta tabla se ingresan los Mecanicos, en MecCi se ingresa la cedula con este formato

'5.549.242-1', además en MecValorHora se hace una comprobación que el valor ingresado sea mayor a 0, Aprate en MecActivo se hace una comprobación en la que este atributo solo puede ser S o N.

**Tabla Repuestos:**

create table Repuestos(

RepuestoCod varchar(9) check(RepuestoCod like '[a-Z][a-Z][a-Z][0-9][0-9][0-9][0-9][0-9][0-9]') primary key,

RepuestosDsc varchar(128),

RepuestoCosto numeric(10,2),

RepuestoTipo varchar(20) check(RepuestoTipo in ('Motor','Carroceria','Lubricacion','Varios'))

);

En esta tabla se crean los Repuestos, el RepuestoCod tiene este formato 'UER836746' además RepuestoTipo hay una comprobación en la que dice que solo se pueden ingresar estos 4 tipos de repuestos 'Motor', 'Carrocería', 'Lubricación', 'Varios'

**Tabla Reparacion:**

create table Reparacion(

RepCod integer,

RepAnio numeric(4),

VehiculoCod integer references Vehiculos(VehiculoCod),

FchEntrada date,

FchSalida date,

MecCod integer references Mecanicos(MecCod),

RepDscEntrada varchar(128),

RepDscSalida varchar(128),

KmsEntrada numeric(10),

primary key(RepCod, RepAnio),

check(FchSalida >= FchEntrada)

);

En esta tabla se registran las Reparaciones, tanto el RepCod y el RepAnio son generados por un Trigger y estos dos atributos forman la clave primaria, VehiculoCod hace referencia a la tabla Vehículo, siendo este el vehículo a reparar, y MecCod hace referencia al mecánico que recibió el Auto, en este caso FchSalida es mayor o igual a FchEntrada del auto.

**Tabla Reparacion\_Repuestos:**

create table Reparacion\_Repuestos(

RepCod integer,

RepAnio numeric(4),

RepuestoCod varchar(9) references Repuestos(RepuestoCod),

Cantidad numeric(6),

CostoUnitario numeric(10,2),

primary key (RepCod, RepAnio, RepuestoCod),

foreign key(RepCod, RepAnio) references Reparacion(RepCod, RepAnio)

);

En esta tabla se hace una relación entre la Reparación y los repuestos, en este caso la clave foránea tiene 2 datos dado que este hace referencia a reparación y su clave primaria está formado por 2 atributos. Aparte la clave primaria está compuesta de 3 atributos RepCod, RepAnio, RepuestoCod

**Tabla Reparacion\_horas:**

create table Reparacion\_horas(

RepCod integer,

RepAnio numeric(4),

MecCod integer references Mecanicos(MecCod),

Horas numeric(4),

CostoHora numeric(10,2),

primary key (RepCod, RepAnio, MecCod),

foreign key(RepCod, RepAnio) references Reparacion(RepCod, RepAnio)

);

En esta tabla se asocia una reparación con diferentes mecánicos, en esta tabla se guarda el costo por hora, y las horas trabajadas del mecanico, la clave primaria está construida por 3 atributos RepCod, RepAnio, MecCod y la clave foránea hace referencia a reparación y esta tiene 2 datos, RepCod, RepAnio

**Tabla Historial\_Clientes\_Vehiculos:**

create table Historial\_Clientes\_Vehiculos(

CliCod integer references Clientes(CliCod),

VehiculoCod integer references Vehiculos(VehiculoCod),

FchCambio date,

primary key(CliCod, VehiculoCod)

);

Cada vez que se asocia un vehículo a un cliente, si el vehículo ya pertenecía a otro cliente se ingresa automáticamente un registro en esta tabla con el código del cliente anterior y la fecha en que se registra el cambio. Esta tabla hace CliCod referencia a Clientes y VehiculoCod referencia a Vehiculos, y la clave primaria está construida por estos 2 atributos

## Creación de triggers:

Genera el Codigo de Reparacion

CREATE trigger [dbo].[Cod\_Rep]

on [dbo].[Reparacion]

instead of Insert

as

Begin

Declare @Anio numeric(4);

Declare @Rep integer;

set @Anio = (Select Year(FchEntrada) From inserted);

if not exists(Select RepAnio from Reparacion where RepAnio = @Anio)

Begin

set @Rep = 1;

end;

else

Begin

set @Rep = (select max(RepCod)+1 from Reparacion where RepAnio= @Anio)

end;

insert into Reparacion (RepCod, RepAnio, VehiculoCod, FchEntrada, FchSalida, MecCod,

RepDscEntrada, RepDscSalida, KmsEntrada)

Values (@Rep,

@Anio,

(Select VehiculoCod from inserted),

(Select FchEntrada from inserted),

(Select FchSalida from inserted),

(Select MecCod from inserted),

(Select RepDscEntrada from inserted),

(Select RepDscSalida from inserted),

(Select KmsEntrada from inserted))

end;

GO

insert into Reparacion values( 0, 0, 10, '01/08/2021', '08/08/2021',5, 'El dueño solicito un cambio de asientos','Se cambiaron los asientos delanteros y traseros', 230670)



En este Trigger de la tabla Reparacion se genera tanto el código de reparación (RepCod) y el Año de reparacion (RepAnio). El código RepAnio se genera del valor año de FchEntrada desde la tabla Insert, se revisa la tabla reparación donde el RepAnio ya se encuentre en esta tabla, si no existe una fila en la que coincidan los RepAnio el RepCod es igual a 1, en cambio si ya existe una reparación con el mismo RepAnio se selecciona el RepCod anterior con el mismo año y se le agrega más uno

Pone el costo de repuesto en reparación repuestos

CREATE TRIGGER [dbo].[Reparacion\_Rep\_Cost]

ON [dbo].[Reparacion\_Repuestos]

instead of INSERT

AS

BEGIN

Declare @Costo numeric(10,2);

Set @Costo = (Select RepuestoCosto from Repuestos

where RepuestoCod =(Select RepuestoCod from inserted))

insert into Reparacion\_Repuestos(

RepCod,

RepAnio,

RepuestoCod,

Cantidad,

CostoUnitario)

Values(

(Select RepCod from inserted),

(Select RepAnio from inserted),

(Select RepuestoCod from inserted),

(Select Cantidad from inserted),

@Costo)

END

Go

insert into Reparacion\_Repuestos values(1, 2020, 'BAS125652', 2, 0)



En este Trigger se hace un instead of insert en la tabla Reparacion\_Repuestos, en el que en vez de hacer un insert en Reparación\_Repuestos, se selecciona el costo del repuesto de la tabla repuesto donde el RepuestoCod sea igual al RepuestoCod de la tabla inserted, una vez obtenido el costo del repuesto se hace el insert en Reparacion\_Repuestos con el costo selecionado

Pone el costo de hora del mecánico en reparación horas

ALTER TABLE [dbo].[Reparacion\_Repuestos] ENABLE TRIGGER [Reparacion\_Rep\_Cost]

GO

CREATE TRIGGER Rep\_horas\_val

ON Reparacion\_horas

instead of insert

AS

BEGIN

Declare @ValorH numeric(10);

set @ValorH = (select MecValorHora from Mecanicos

where MecCod = (Select MecCod from inserted))

SET NOCOUNT ON;

insert into Reparacion\_horas(

RepCod,

RepAnio,

MecCod,

Horas,

CostoHora)

Values(

(Select RepCod from inserted),

(Select RepAnio from inserted),

(Select MecCod from inserted),

(Select Horas from inserted),

@ValorH)

END

GO

insert into Reparacion\_horas values(1, 2020, 1, 16, 0)



En este Trigger se hace un instead of insert en Reparacion\_Horas, en el que en vez de hacer un insert en Reparación\_horas, se selecciona el valor por hora de la tabla Mecánicos donde el MecCod sea igual al MecCod desde la tabla inserted, una vez obtenido valor por hora de la tabla Mecánicos se hace el insert en Reparación\_horas con el valor selecionado

Al modificar Mecánicos se asigna el nuevo costo de hora del mecánico en reparación horas

CREATE TRIGGER Act\_Val\_Ho

ON Mecanicos

AFTER UPDATE

AS

Declare @ValorH numeric(10);

Declare @Mec int;

Set @ValorH = (Select MecValorHora from inserted)

Set @Mec = (Select MecCod from inserted)

Update Reparacion\_horas

set CostoHora = @ValorH

where MecCod = @Mec

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for trigger here

END

GO

update Mecanicos set MecValorHora = 75

where MecCod =1

select \* from Reparacion\_Horas



En este trigger después de hacer un Update la tabla Mecanicos se selecciona el Valor por hora y el MecCod de la tabla inserted y se actualiza el Costo Hora en la tabla Reparacion\_Horas donde el MecCod sea igual al MecCod de la tabla inserted

Al modificar repuesto se asigna el nuevo costo en reparación Repuesto

CREATE TRIGGER [dbo].[Up\_Costo\_Rep]

ON [dbo].[Repuestos]

AFTER UPDATE

AS

BEGIN

Declare @Costo numeric(10,2);

Declare @Rep varchar(9)

Set @Costo = (Select RepuestoCosto from inserted)

Set @Rep = (Select RepuestoCod from inserted)

Update Reparacion\_Repuestos

set CostoUnitario = @Costo

where RepuestoCod = @Rep

SET NOCOUNT ON;

-- Insert statements for trigger here

END

GO

ALTER TABLE [dbo].[Repuestos] ENABLE TRIGGER [Up\_Costo\_Rep]

GO

select \* from Reparacion\_Repuestos

where RepuestoCod = 'AJS671909'

update Repuestos set RepuestoCosto = 1000

where RepuestoCod = 'AJS671909'



En este trigger después de hacer un Update la tabla Repuestos se selecciona el Costo y el RepuestoCod de la tabla inserted y se actualiza el Costo en la tabla Reparacion\_Repuestos donde el RepuestoCod sea igual al RepuestoCod de la tabla inserted

B)

## Creación de índices:

create index VehiculoCod\_Clientes\_Vehiculos on Clientes\_Vehiculos(VehiculoCod)

create index MecCod\_Reparacion on Reparacion(MecCod);

create index VehiculoCod\_Reparacion on Reparacion(VehiculoCod);

create index RepAnio\_Reparacion on Reparacion(RepAnio);

create index RepAnio\_Reparacion\_Repuestos on Reparacion\_Repuestos(RepAnio)

create index RepuestoCod\_Reparacion\_Repuestos on Reparacion\_Repuestos(RepuestoCod)

create index RepAnio\_Reparacion\_horas on Reparacion\_horas(RepAnio)

create index MecCod\_Reparacion\_horas on Reparacion\_horas(MecCod)

create index VehiculoCod\_Historial\_Clientes\_Vehiculos on Historial\_Clientes\_Vehiculos(VehiculoCod)

Se crearon índices en las tablas en las que una clave foránea no sea clave primaria por ejemplo create index MecCod\_Reparacion on Reparacion(MecCod);

En el caso en el que la clave primaria se componga de 2 atributos se crea un índice en el atributo secundario ejemplo: create index RepAnio\_Reparacion on Reparacion(RepAnio);

No se crea una tabla en las claves foráneas que forman el atributo primario en la clave primaria ejemplo primary key (RepCod, RepAnio, MecCod), no se crea índice en el atributo RepCod.

C)

## Ingreso de datos:

Clientes:

insert into Clientes values('Matias', '2.321.246-1', '092332123', 'Sarandi 544', 'Chormitosjiji@gmail.com', '24/04/2019')

insert into Clientes values('Jose', '4.221.646-6', '096781221', 'Guaymayen 609', 'JoseXd420@gmail.com', '10/02/2020')

insert into Clientes values('Maria', '6.942.069-0', '092345720', 'Luisito 507', 'AlvertitoFullHd@gmail.com', '14/11/2020')

insert into Clientes values('Joaco', '4.502.169-8', '097033221', 'Gorgito 510', 'HideOnTuki@gmail.com', '14/11/2021')

insert into Clientes values('Valeria', '6.514.132-9', '095022217', 'Faker 001', 'HideOnStream@gmail.com', '19/01/2020')

insert into Clientes values('Bruno', '3.615.769-8', '099138256', 'Ranitas 600', 'FrEBrunGamer@gmail.com', '04/03/2019')

insert into Clientes values('Patricia', '4.703.809-4', '091389089', 'Dafran 420', 'TracerOtp@gmail.com', '14/11/2008')

insert into Clientes values('Jose', '3.570.199-9', '092328621', 'Ramus 750', 'FF15JgDiff@gmail.com', '26/10/2016')

insert into Clientes values('Camilo', '4.571.199-9', '092329622', 'Luisito 602', 'NotIntiSion@gmail.com', '22/05/2015')

insert into Clientes values('Matias', '3.807.200-1', '091346893', 'Jaimi 144', 'QuienEsElMensajero@gmail.com', '29/03/2014')

insert into Clientes values('Pablo', '2.471.232-3', '092456684', 'Illo 401', 'CalvitoGamer@gmail.com', '27/06/2020')

insert into Clientes values('Maxi', '6.372.503-4', '092321326', 'Rodri 630', 'Matakrakens@gmail.com', '06/10/2017')

insert into Clientes values('Juan', '5.532.020-2', '099566210', 'Ramus 751', 'TrackLab@gmail.com', '26/10/2016')

insert into Clientes values('Facundo', '6.552.170-2', '096520622', 'Blue 701', 'HdBlueXx14@gmail.com', '24/03/2022')

insert into Clientes values('Alverto', '3.602.777-1', '092308602', 'Sarandi 350', 'Alvertito43@gmail.com', '26/10/2016')

Vehículos:

1. insert into Vehiculos values('AZH2395', 'Mercedes', 'X2000', 1999, 'Rojo')
2. insert into Vehiculos values('HJZ9245', 'Toyota', 'Auris', 2015, 'Negro')
3. insert into Vehiculos values('MJZ3234', 'Toyota', 'Corolla', 2014, 'Gris')
4. insert into Vehiculos values('ADC6942', 'Tesla', 'Model S', 2018, 'Blanco')
5. insert into Vehiculos values('RAW1435', 'Tesla', 'Model X', 2020, 'Gris')
6. insert into Vehiculos values('ADS7040', 'Audi', 'Judini', 2019, 'Negro')
7. insert into Vehiculos values('BLU4125', 'Ferrari', 'Marw', 2018, 'Azul')
8. insert into Vehiculos values('PAR3145', 'Tesla', 'Model S', 2018, 'Rojo')
9. insert into Vehiculos values('MSG7021', 'Toyota', 'Auris', 2015, 'Negro')
10. insert into Vehiculos values('AZJ2495', 'Mercedes', 'X2000', 1999, 'Verde')
11. insert into Vehiculos values('AZJ2375', 'Mercedes', 'X2000', 1999, 'Rojo')
12. insert into Vehiculos values('HJA9255', 'Toyota', 'Auris', 2015, 'Negro')
13. insert into Vehiculos values('MHZ2234', 'Toyota', 'Corolla', 2014, 'Gris')
14. insert into Vehiculos values('ZAD4942', 'Tesla', 'Model S', 2018, 'Blanco')
15. insert into Vehiculos values('RYW1135', 'Tesla', 'Model X', 2020, 'Gris')
16. insert into Vehiculos values('ADH7342', 'Audi', 'Judini', 2019, 'Azul')
17. insert into Vehiculos values('BLZ5525', 'Ferrari', 'Marw', 2018, 'Azul')
18. insert into Vehiculos values('PRR1145', 'Tesla', 'Model S', 2018, 'Rojo')
19. insert into Vehiculos values('MSG7121', 'Toyota', 'Auris', 2015, 'Blanco')
20. insert into Vehiculos values('AHJ2595', 'Mercedes', 'X2000', 1999, 'Negro')

Clientes\_Vehiculos:

insert into Clientes\_Vehiculos Values(1,3, '03/06/2016')

insert into Clientes\_Vehiculos Values(2,2, '04/05/2017')

insert into Clientes\_Vehiculos Values(1,1, '03/06/2004')

insert into Clientes\_Vehiculos Values(3,4, '13/05/2019')

insert into Clientes\_Vehiculos Values(5,5, '03/08/2020')

insert into Clientes\_Vehiculos Values(6,8, '13/10/2020')

insert into Clientes\_Vehiculos Values(4,6, '30/05/2021')

insert into Clientes\_Vehiculos Values(7,7, '08/01/2018')

insert into Clientes\_Vehiculos Values(7,9, '14/08/2016')

insert into Clientes\_Vehiculos Values(8,10, '01/01/2000')

insert into Clientes\_Vehiculos Values(9,11, '11/07/2003')

insert into Clientes\_Vehiculos Values(10,12, '23/07/2018')

insert into Clientes\_Vehiculos Values(12,14, '14/03/2019')

insert into Clientes\_Vehiculos Values(11,13, '09/11/2018')

insert into Clientes\_Vehiculos Values(13,15, '15/01/2022')

insert into Clientes\_Vehiculos Values(14,16, '26/03/2022')

insert into Clientes\_Vehiculos Values(15,17, '09/11/2018')

insert into Clientes\_Vehiculos Values(15,18, '20/09/2019')

insert into Clientes\_Vehiculos Values(14,19, '28/06/2017')

insert into Clientes\_Vehiculos Values(10,20, '09/08/2005')

Mecanicos

insert into Mecanicos values('Jose', '2.345.152-9', '092450987', '20/05/2019', 60, 'S')

insert into Mecanicos values('Maria', '3.379.891-0', '092679414', '15/03/2018', 65, 'S')

insert into Mecanicos values('Pedro', '4.245.112-4', '099250141', '14/01/2020', 50, 'S')

insert into Mecanicos values('Matias', '5.549.242-1', '095455947', '14/11/2016', 80, 'S')

insert into Mecanicos values('Juan', '3.325.959-7', '094510097', '23/06/2019', 60, 'S')

insert into Mecanicos values('Luis', '4.221.451-3', '094511347', '23/06/2009', 70, 'N')

insert into Mecanicos values('Martin', '6.538.143-4', '099435975', '07/06/2015', 46, 'S')

insert into Mecanicos values('Emiliano', '5.432.845-4', '095543741', '12/11/2012', 63, 'S')

insert into Mecanicos values('Julian', '9.724.712-8', '098731645', '29/01/2019', 57, 'N')

Repuestos

insert into Repuestos values('AZC134052','Piston para el motor', '400', 'Motor')

insert into Repuestos values('BAS125652','Parte delantera del auto', '450', 'Carroceria')

insert into Repuestos values('HKJ267512','Tipo de aceite', '470', 'Lubricacion')

insert into Repuestos values('ZAS835124','Camara de combustion', '450', 'Motor')

insert into Repuestos values('DJS675124','Pack de tornillos', '300', 'Varios')

insert into Repuestos values('JZA234153','Carburador para el motor', '399.99', 'Motor')

insert into Repuestos values('AZA424611','Parte tracera del auto', '415.15', 'Carroceria')

insert into Repuestos values('MKA261511','Bujia para el motor', '470', 'Motor')

insert into Repuestos values('JAS124159','Bomba de gasolina', '450', 'Motor')

insert into Repuestos values('AJS671909','Pack Asientos', '999.99', 'Varios')

insert into Repuestos values('SHS570145','Parachoques', '100', 'Carroceria')

insert into Repuestos values('JDF231264','Correa de distribucion', '4500', 'Motor')

insert into Repuestos values('MFB543712','Lona para camioneta', '9500', 'Carroceria')

insert into Repuestos values('KLK126543','Radiador', '1900', 'Motor')

insert into Repuestos values('UGO876423','Motor de maquina levanta vidrio', '1100', 'Carroceria')

insert into Repuestos values('AMI843263','Carburador', '5200', 'Motor')

insert into Repuestos values('SIO985634','Motor de bloqueo de puerta', '7800', 'Carroceria')

insert into Repuestos values('MAF645234','Tapa de cilindros', '8200', 'Motor')

insert into Repuestos values('UER836746','Tornillo tapa de clilindros', '140', 'Motor')

insert into Repuestos values('JFG736201','Junta de tapa de valvulas', '250', 'Motor')

insert into Repuestos values('LOK354809','Junta del carter', '180', 'Motor')

Reparacion

insert into Reparacion values( 0, 0, 1, '05/07/2020', '08/07/2020',1, 'Choque en la parte delantera','Se cambio toda la parte delantera del auto, aparte de pintarlo', 5000)

insert into Reparacion values( 0, 0, 1, '05/09/2019', '08/09/2019',2, 'Fallo del motor','Se cambio una parte del motor', 1500)

insert into Reparacion values( 0, 0, 2, '03/11/2018', '06/11/2018',2, 'Problemas en el arranque del motor','Se cambio parte del motor', 10000)

insert into Reparacion values( 0, 0, 3, '05/10/2020', '06/10/2020',2, 'Problemas en el arranque del motor','Se cambio parte del motor', 15000)

insert into Reparacion values( 0, 0, 4, '18/11/2018', '25/11/2018',2, 'Problemas en el motor','Se cambio parte del motor', 15000)

insert into Reparacion values( 0, 0, 7, '20/12/2019', '04/01/2020',5, 'El vidrio de la puerta del conductor no funcionaba','Se cambio el motor del levanta vidrios', 157600)

insert into Reparacion values( 0, 0, 1, '20/06/2022', null,5, 'El dueño pidio un cambio de juntas del carter y tapa de valvulas','Se cambarion las juntas', 205780)

insert into Reparacion values( 0, 0, 2, '15/03/2020', '26/03/2020',8, 'El dueño pidio un cambio de la correa de distribucion','Se cambio la correa', 46780)

insert into Reparacion values( 0, 0, 14, '20/02/2019', '04/03/2019',8, 'El conductor pidio un cambio de asientos','Se cambiaron los asientos', 10760)

insert into Reparacion values( 0, 0, 15, '17/10/2021', '02/11/2021',8, 'Al vehiculo no le funcionaba el carburador','Se cambio el carburador', 63340)

insert into Reparacion values( 0, 0, 12, '01/11/2018', '03/11/2018',7, 'El dueño pidio un cambio de aceite','Se le cambio el aceite al vehiculo', 105670)

insert into Reparacion values( 0, 0, 17, '28/02/2022', '04/03/2022',3, 'El radiador estaba dañado','Se cambio el radiador', 98750)

insert into Reparacion values( 0, 0, 15, '20/11/2017', '04/12/2017',2, 'Cambio del carburador','Se cambio el carburador', 210130)

insert into Reparacion values( 0, 0, 15, '12/04/2020', '17/04/2020',2, 'parachoques roto','Se cambio el parachoques', 14000)

insert into Reparacion values( 0, 0, 10, '12/03/2021', '17/03/2021',1, 'Muchas daños','Se aplicaron muchas reparaciones', 18000)

insert into Reparacion values( 0, 0, 2, '17/10/2022', '02/11/2022',8, 'Al vehiculo no le funcionaba el carburador','Se cambio el carburador', 73340)

insert into Reparacion values( 0, 0, 10, '25/11/2021', '04/12/2021',1, 'Habian tornillos dañados','Se cambiaron los tornillos dañados', 321090)

insert into Reparacion values( 0, 0, 10, '02/02/2021', '06/11/2021',2, 'Los pistones estaban dañados','Se cambiaron los pistones', 105780)

insert into Reparacion values( 0, 0, 10, '04/03/2021', '04/03/2021',3, 'El dueño pidio un cambio de aceite','Se cambio el aceite', 127900)

insert into Reparacion values( 0, 0, 10, '07/05/2021', '08/05/2021',8, 'El parachoques trasero y delantero estaban dañados','Se cambiaron ambos parachoques', 197930)

insert into Reparacion values( 0, 0, 10, '01/05/2021', '08/05/2021',5, 'El dueño solicito un cambio de asientos','Se cambiaron los asientos delanteros y traseros', 230670)

insert into Reparacion values( 0, 0, 18, '07/03/2021', '08/03/2021',1, 'El aceite estaba pasado','Se cambio el aceite', 124570)

insert into Reparacion\_Repuestos values(1, 2020, 'BAS125652', 2, 0)

insert into Reparacion\_Repuestos values(1, 2019, 'MKA261511', 1, 0)

insert into Reparacion\_Repuestos values(1, 2018, 'ZAS835124',1, 0)

insert into Reparacion\_Repuestos values(1, 2018, 'JZA234153',1, 0)

insert into Reparacion\_Repuestos values(1, 2018, 'MKA261511',1, 0)

insert into Reparacion\_Repuestos values(1, 2018, 'JAS124159',2, 0)

insert into Reparacion\_Repuestos values(2, 2018, 'ZAS835124',1, 0)

insert into Reparacion\_Repuestos values(2, 2019, 'UGO876423', 1, 0)

insert into Reparacion\_Repuestos values(1, 2022, 'JFG736201', 1, 0)

insert into Reparacion\_Repuestos values(1, 2022, 'LOK354809', 1, 0)

insert into Reparacion\_Repuestos values(3, 2020, 'JDF231264', 1, 0)

insert into Reparacion\_Repuestos values(3, 2019, 'AJS671909', 1, 0)

insert into Reparacion\_Repuestos values(1, 2021, 'AMI843263',1, 0)

insert into Reparacion\_Repuestos values(3, 2018, 'HKJ267512',2, 0)

insert into Reparacion\_Repuestos values(2, 2022, 'KLK126543',1, 0)

insert into Reparacion\_Repuestos values(4, 2020, 'SHS570145', 1, 0)

insert into Reparacion\_Repuestos values(2, 2021, 'HKJ267512', 2, 0)

insert into Reparacion\_Repuestos values(2, 2021, 'MKA261511', 2, 0)

insert into Reparacion\_Repuestos values(3, 2022, 'AMI843263', 1, 0)

insert into Reparacion\_Repuestos values(3, 2021, 'DJS675124', 11, 0)

insert into Reparacion\_Repuestos values(4, 2021, 'AZC134052', 3, 0)

insert into Reparacion\_Repuestos values(5, 2021, 'HKJ267512', 2, 0)

insert into Reparacion\_Repuestos values(6, 2021, 'SHS570145', 2, 0)

insert into Reparacion\_Repuestos values(7, 2021, 'AJS671909', 2, 0)

insert into Reparacion\_Repuestos values(8, 2021, 'HKJ267512', 3, 0)

insert into Reparacion\_horas values(1, 2020, 1, 16, 0)

insert into Reparacion\_horas values(1, 2019, 2, 5, 0)

insert into Reparacion\_horas values(1, 2018, 2, 8, 0 )

insert into Reparacion\_horas values(1, 2018, 3, 12, 0 )

insert into Reparacion\_horas values(2, 2018, 2, 5, 0 )

insert into Reparacion\_horas values(2, 2018, 3, 7, 0 )

insert into Reparacion\_horas values(3, 2018, 8, 2, 0)

insert into Reparacion\_horas values(2, 2019, 5, 2, 0)

insert into Reparacion\_horas values(1, 2022, 3, 2, 0)

insert into Reparacion\_horas values(1, 2022, 5, 1, 0)

insert into Reparacion\_horas values(1, 2022, 4, 1, 0)

insert into Reparacion\_horas values(3, 2020, 8, 4, 0)

insert into Reparacion\_horas values(3, 2019, 9, 4, 0)

insert into Reparacion\_horas values(1, 2021, 8, 1, 0 )

insert into Reparacion\_horas values(2, 2022, 3, 2, 0 )

insert into Reparacion\_horas values(2, 2022, 4, 1, 0 )

insert into Reparacion\_horas values(1, 2017, 9, 14, 0)

insert into Reparacion\_horas values(1, 2017, 4, 14, 0)

insert into Reparacion\_horas values(1, 2017, 2, 4, 0)

insert into Reparacion\_horas values(4, 2020, 1, 20, 0)

insert into Reparacion\_horas values(4, 2020, 2, 14, 0)

insert into Reparacion\_horas values(2, 2021, 1, 15, 0)

insert into Reparacion\_horas values(3, 2022, 8, 11, 0)

insert into Reparacion\_horas values(3, 2021, 1, 5, 0)

insert into Reparacion\_horas values(4, 2021, 2, 10, 0)

insert into Reparacion\_horas values(4, 2021, 5, 6, 0)

insert into Reparacion\_horas values(5, 2021, 3, 1, 0)

insert into Reparacion\_horas values(6, 2021, 8, 7, 0)

insert into Reparacion\_horas values(7, 2021, 5, 9, 0)

insert into Reparacion\_horas values(8, 2021, 1, 2, 0)

## Ejemplos de ingresos de datos erróneos

Ejemplo de restricción de tabla unique:

insert into Clientes values('Matias', '2.321.246-1', '092332123', 'Sarandi 544', 'Chormitosjii@gmail.com', '24/04/2019')

Msg 2627, Level 14, State 1, Line 9

Infracción de la restricción UNIQUE KEY 'UQ\_\_Clientes\_\_FA1DF9A9369E5D91'. No se puede insertar una clave duplicada en el objeto 'dbo.Clientes'. El valor de la clave duplicada es (2.321.246-1).

Se terminó la instrucción.

Ejemplo de restricción de tabla not null:

insert into Clientes values(null, '2.321.246-0', '092332123', 'Sarandi 544', 'Chormitosjiji@gmail.com', '24/04/2019')

Msg 515, Level 16, State 2, Line 30

No se puede insertar el valor NULL en la columna 'CliNom', tabla 'Obligatorio2BD.dbo.Clientes'. La columna no admite valores NULL. Error de INSERT.

Se terminó la instrucción.

Ejemplo en el que el dato ingresado es mas grande que el valor definido en la tabla:

insert into Clientes values('Matias', '2.321.246-0', '092332123', 'Sarandi 544', 'Chormitosjiji@gmail.com', '24/04/2019')

Msg 2628, Level 16, State 1, Line 30

Los datos binarios o de la cadena se truncan en la columna "CliTel" de la tabla "Obligatorio2BD.dbo.Clientes". Valor truncado: "092332123".

Se terminó la instrucción.

Ejemplo en el que el Check no recibe el valor esperado:

insert into Clientes values('Matias', 'A.321.246-0', '092332123', 'Sarandi 544', 'Chormitosjij1@gmail.com', '24/04/2019')

Msg 547, Level 16, State 0, Line 31

Instrucción INSERT en conflicto con la restricción CHECK 'CK\_\_Clientes\_\_CliCi\_\_4222D4EF'. El conflicto ha aparecido en la base de datos 'Obligatorio2BD', tabla 'dbo.Clientes', column 'CliCi'.

Se terminó la instrucción.

Ejemplo en el que la clave foránea no existe

insert into Reparacion\_horas values(1, 2020, 10, 16, 0)

Msg 547, Level 16, State 0, Procedure Rep\_horas\_val, Line 13 [Batch Start Line 4]

Instrucción INSERT en conflicto con la restricción FOREIGN KEY 'FK\_\_Reparacio\_\_MecCo\_\_10566F31'. El conflicto ha aparecido en la base de datos 'Obligatorio2BD', tabla 'dbo.Mecanicos', column 'MecCod'.

Se terminó la instrucción.

Ejemplo en el que la clave primaria se repite

insert into Reparacion\_horas values(1, 2020, 1, 16, 0)

Msg 2627, Level 14, State 1, Procedure Rep\_horas\_val, Line 13 [Batch Start Line 4]

Infracción de la restricción PRIMARY KEY 'PK\_\_Reparaci\_\_2C8B3D5D7869BC7B'. No se puede insertar una clave duplicada en el objeto 'dbo.Reparacion\_horas'. El valor de la clave duplicada es (1, 2020, 1).

Se terminó la instrucción.

# Punto 2:

## a)

select R.\*, sum(RR.Cantidad) Cantidad\_usos

from Repuestos R inner join Reparacion\_Repuestos RR

on RR.RepuestoCod like R.RepuestoCod

where R.RepuestoCosto > 2000 and RR.RepAnio = 2022

group by R.RepuestoCod, R.RepuestosDsc, R.RepuestoCosto, R.RepuestoTipo

having sum(RR.Cantidad) >= all (

select sum(RR1.Cantidad)

from Reparacion\_Repuestos RR1 inner join Repuestos R1

on R1.RepuestoCod = RR1.RepuestoCod

where R1.RepuestoCosto > 2000 and RR1.RepAnio = 2022

group by RR1.RepuestoCod)



Se muestran los datos del/los repuestos, y la cantidad de usos que durante el 2022 tuvo un costo mayor a 2000 y que a su vez fue el más usado .

## b)

select V.\*, count(R.VehiculoCod) Cant\_Reparaciones

from Vehiculos V right join Reparacion R

on R.VehiculoCod = V.VehiculoCod

where R.RepAnio = 2021

group by V.VehiculoCod, V.Matricula, V.Marca, V.Modelo, V.Anio, V.Color

having count(R.VehiculoCod) > 5 and V.VehiculoCod = any (select R1.VehiculoCod

from Reparacion R1 inner join Reparacion\_Repuestos RR

on RR.RepCod = R1.RepCod and RR.RepAnio = R1.RepAnio

where R1.RepAnio = 2021 and RR.Cantidad > 10)

and V.VehiculoCod not in (select R2.VehiculoCod

from Reparacion R2 inner join Reparacion\_Repuestos RR1

on RR1.RepCod = R2.RepCod and RR1.RepAnio = R2.RepAnio

where R2.RepAnio = 2021 and RR1.Cantidad < 2)



Se muestran los datos de los vehículos que en 2021 tuvieron mas de 5 reparaciones, pero solo si alguna de esas reparaciones utilizo mas de 10 repuestos y que ninguna halla utilizado menos de 2 repuestos.

## C)

Select V.VehiculoCod, count(R.VehiculoCod) 'Cantidad de repar'

from Vehiculos V inner join Reparacion R

on V.VehiculoCod = R.VehiculoCod

where year(R.FchSalida) between '2010' and '2022'

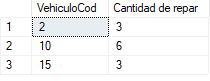
group by V.VehiculoCod

having count(R.VehiculoCod) >

( Select COUNT(R.VehiculoCod )/ Count(Distinct(R.VehiculoCod)) as Count

from Reparacion R

where year(R.FchSalida) between '2010' and '2022')



Se muestra para cada vehículo la cantidad de reparaciones que tuvo entre 2021 y la actualidad, pero solo los vehículos que superan el promedio general de reparaciones.

## D)

select Distinct(R.RepuestoCod), R.RepuestosDsc,

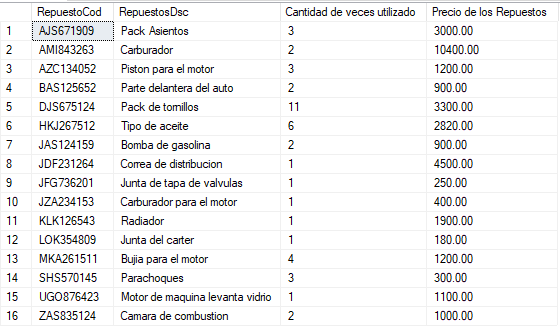
Sum(RR.Cantidad) as 'Cantidad de veces utilizado',

sum (RR.CostoUnitario \*RR.Cantidad) as 'Precio de los Repuestos'

from Repuestos R inner join Reparacion\_Repuestos RR

on R.RepuestoCod = RR.RepuestoCod

Group by R.RepuestoCod, R.RepuestosDsc



Se muestra para cada repuesto su código, descripción, la cantidad de veces que se uso en reparaciones y el costo total de esto.

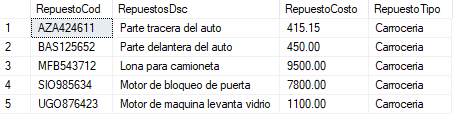
## E)

Select R.\*

from Repuestos R

where R.RepuestoTipo = 'Carroceria' and R.RepuestoCosto > any (Select R1.RepuestoCosto from Repuestos R1

Where R1.RepuestoTipo='Motor')



Se muestra todos los repuestos de carrocería que su costo supera a algún repuesto de motor.

# Punto 3:

Create View V\_Mec\_DifVec as(

Select M.\*,Count(Distinct(R.VehiculoCod)) as 'Cantidad de Vehiculos'

from Mecanicos M inner join Reparacion R on M.MecCod = R.MecCod inner join Vehiculos V on V.VehiculoCod = R.VehiculoCod

where R.RepAnio= 2021 and V.Anio > 2015 and V.VehiculoCod not in ( Select R1.VehiculoCod from Reparacion R1 where R1.RepAnio < 2021)

group by M.MecCod, M.MecNom , M.MecCi, M.MecTel, M.MecFchIng, M.MecValorHora, M.MecActivo);

Select \* From V\_Mec\_DifVec



Se muestran todos los datos de los mecánicos, y los distintos vehículos que atendieron en 2021, pero solo los vehículos que son mayores de 2015 y que nunca hallan sido atendidos antes de 2021.

# PUNTO 4

Funciones y Proecedimientos

## A)

Create function CostoRep(@RepCod int, @RepAnio numeric(4))

returns numeric(10,2)

as

begin

declare @CostoTotal numeric(10,2)

declare @C numeric(10,2)

declare @D numeric(10,2)

set @C = (select sum(Horas \* CostoHora) from Reparacion\_horas where RepCod = @RepCod and RepAnio = @RepAnio)

set @D = (select sum(Cantidad \* CostoUnitario) from Reparacion\_Repuestos where RepCod = @RepCod and RepAnio = @RepAnio)

set @CostoTotal = @C + @D

return @CostoTotal

end

select dbo.CostoRep(1, 2021)



Se muestra el costo total de una reparación ingresada por el usuario, que se hace multiplicando el costo unitario por la cantidad de cada repuesto utilizado y multiplicando el costo hora por la cantidad de horas de cada mecánico que participo.

## B)

Create Procedure [dbo].[HorasCostoDadoAyM] (@TotalHoras int out,@TotalCosto numeric (10,2) out,@Mes int, @Anio numeric(4))

as

begin

set @TotalHoras = (select sum(horas) from Reparacion\_horas RH inner join Reparacion R on R.RepAnio = RH.RepAnio and R.RepCod = RH.RepCod

where Month(R.FchEntrada) = @Mes and R.RepAnio = @Anio)

set @TotalCosto =(select sum(Horas\* CostoHora) from Reparacion\_horas RH inner join Reparacion R on R.RepAnio = RH.RepAnio and R.RepCod = RH.RepCod

where Month(R.FchEntrada) = @Mes and R.RepAnio = @Anio)

end;

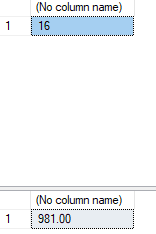
declare @A int

declare @B numeric (10,2)

exec dbo.HorasCostoDadoAyM @A Out, @B out, 5, 2021

select @A

select @B



Se muestra la cantidad de horas y el costo total de los mecánicos que trabajaron en una reparación durante un mes y un año ingresado por el usuario, la cantidad de horas se obtiene con la suma de todas horas trabajadas de todos los mecánicos donde el año y el mes coincidan con los que ingreso el usuario, y el costo total se obtiene multiplicando el total de las horas por el costo hora de cada mecánicos donde el año y el mes coincidan con los que ingreso el usuario.

## C)

Create PROCEDURE CliEnRepar

AS

BEGIN

SELECT C.CliCod, C.CliNom, V.Matricula from Clientes C inner join Clientes\_Vehiculos CV on CV.CliCod = C.CliCod inner join

Vehiculos V on CV.VehiculoCod = V.VehiculoCod inner join Reparacion R on V.VehiculoCod = R.VehiculoCod

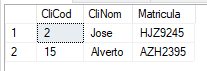
Where R.FchSalida > GETDATE() or R.FchSalida is null

order by C.CliCod, V.Matricula

END

GO

Exec dbo.CliEnRepar



Se muestra el código y el nombre del cliente y la matrícula del vehículo que este actualmente en reparación, esto se sabe si la fecha de salida no existe o si es mayor a la fecha actual.

## D)

create function CantDueniosVCMat(@Matricula varchar(7))

returns int

as

begin

declare @Contador1 int

declare @Contador2 int

declare @Resultado int

set @Contador1 = (select count(\*) from Vehiculos V inner join Historial\_Clientes\_Vehiculos HCV on V.VehiculoCod = HCV.VehiculoCod

where Matricula = @Matricula)

set @Contador2 = (select count(\*) from Vehiculos V inner join Clientes\_Vehiculos CV on V.VehiculoCod = CV.VehiculoCod

where Matricula = @Matricula)

set @Resultado = @Contador1 + @Contador2

return @Resultado

end

select dbo.CantDueniosVCMat('AZH2395')



Se muestra la cantidad de clientes que tuvieron un vehículo, que es ingresada su matricula por el usuario, esto se obtiene contando la cantidad de veces que esta el vehículo en la tabla del historial de clientes y vehículos, y de clientes vehículos.

## E)

Create procedure [dbo].[RepuestoMasUsado] @RepuestoCod varchar(11) out

as

begin

set @RepuestoCod=(select R.RepuestoCod from Repuestos R inner join Reparacion\_Repuestos RR

on R.RepuestoCod = RR.RepuestoCod

group by R.RepuestoCod

having sum(RR.Cantidad) >= all (select sum(RR1.Cantidad) from Repuestos R1 inner join Reparacion\_Repuestos RR1

on R1.RepuestoCod = RR1.RepuestoCod group by R1.RepuestoCod))

end

declare @A Varchar(11)

exec dbo.RepuestoMasUsado @A out

select @A



Se muestra el código y la cantidad de veces usadas del repuesto más usado, se obtiene comprobando que la cantidad de veces usadas sea mayor o igual a todas las otras.

# PUNTO 5

TRIGGERS

## A)

create trigger Val\_Hora\_Mec

on Mecanicos

after update

as

begin

declare @horas\_nuevo numeric(4)

declare @horas\_viejo numeric(4)

set @horas\_nuevo = (select top 1 MecValorHora from inserted)

set @horas\_viejo = (select top 1 MecValorHora from deleted)

if @horas\_nuevo > (@horas\_viejo \* 1.20)

begin

raiserror('No se puede aumentar las horas más de un 20 porciento', 1, 1);

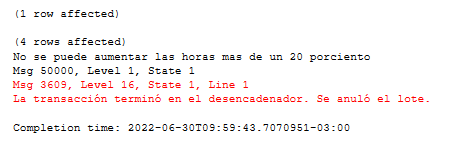
rollback;

end

end

update Mecanicos set MecValorHora = 75

where MecCod = 1



No permite aumentar las horas de un mecánico mas de un 20%, se comprueba obteniendo el valor hora antes y después de actualizarlo, si el valor nuevo es 20% o mas alto que el valor viejo la actualización dará error mostrando un mensaje de error sobre porque no funciono la actualización.

## B)

CREATE TRIGGER inac\_Mec

ON Reparacion

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

declare @tipo varchar(1)

declare @IdMec int

set @IdMec =(select MecCod from inserted)

set @tipo = (select MecActivo from Mecanicos where MecCod= @IdMec)

if (@tipo = 'N')

Begin

raiserror('No se puede asociar un Mecancico inactivo',1,1);

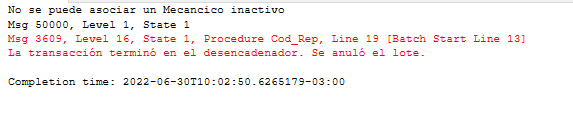
rollback;

End

END

GO

insert into Reparacion values( 0, 0, 7, '20/12/2019', '04/01/2020',6, 'El vidrio de la puerta del conductor no funcionaba','Se cambio el motor del levanta vidrios', 157600)



Create TRIGGER [dbo].[inac\_Mec\_RH]

ON [dbo].[Reparacion\_horas]

AFTER INSERT, UPDATE

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

declare @tipo varchar(1)

declare @IdMec int

set @IdMec =(select top 1 MecCod from inserted)

set @tipo = (select top 1 MecActivo from Mecanicos where MecCod= @IdMec)

if (@tipo = 'N')

Begin

raiserror('No se puede asociar un Mecancico inactivo',1,1);

rollback;

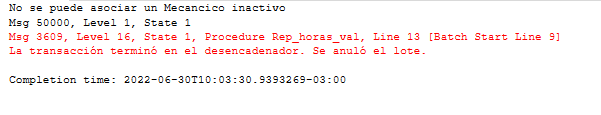
End

-- Insert statements for trigger here

END

GO

insert into Reparacion\_horas values(1, 2020, 6, 16, 0)



No permite ingresar mecánicos inactivos en reparación ni en reparación horas, esto se comprueba obteniendo el código del mecánico ingresado, si en la tabla mecánicos este está inactivo, la acción se detendrá dando un mensaje de error sobre lo que paso.

## C)

CREATE TABLE [dbo].[AUDIT\_REPHS](

[RepCod] [int] NOT NULL,

[RepAnio] numeric(4,0) NOT NULL,

[UsuarioQueLoCambio] [varchar](30) NULL,

[TipoCambio] [varchar](30) NOT NULL,

[HorasViejo] [numeric](4, 0) NULL,

[HorasNuevo] [numeric](4, 0) NULL,

[CostoHoraViejo] [numeric](10, 2) NULL,

[CostoHoraNuevo] [numeric](10, 2) NULL

)

Create trigger [dbo].[Cambios\_Reparacion\_Horas]

on [dbo].[Reparacion\_horas]

after delete, update

as

begin

if not exists(select top 1 \* from inserted)

begin

insert into AUDIT\_REPHS

values(

(select top 1 RepCod from deleted),

(select top 1 RepAnio from deleted),

(select SYSTEM\_USER),

'delete',

(select top 1 Horas from deleted),

null,

(select top 1 CostoHora from deleted),

null)

end

else

begin

insert into AUDIT\_REPHS

values(

(select top 1 RepCod from deleted),

(select top 1 RepAnio from deleted),

(select SYSTEM\_USER),

'update',

(select top 1 Horas from deleted),

(select top 1 Horas from inserted),

(select top 1 CostoHora from deleted),

(select top 1 CostoHora from inserted))

end

end

update Reparacion\_horas set CostoHora =75, Horas =10

where RepCod =1 and RepAnio =2017 and MecCod =2

Select \* from AUDIT\_REPHS



Al borrar o actualizar algo en la tabla reparación horas, se activará un trigger que insertara un identificador de la reparación, el usuario en el sistema, las horas antiguas y nuevas, y el costo hora antiguo y nuevo, y todo esto quedará en la tabla Audit rephs.

## D)

Create TRIGGER [dbo].[H\_Cli\_Veh\_ACT]

ON [dbo].[Clientes\_Vehiculos]

AFTER INSERT

AS

BEGIN

Declare @CliNew int

Declare @Veh int

Declare @CliOld int

Set @CliNew = (Select CliCod from inserted)

Set @Veh = (Select VehiculoCod from inserted)

if exists (select CliCod from Clientes\_Vehiculos where CliCod <> @CliNew and VehiculoCod = @Veh)

Begin

set @CliOld = (select CliCod from Clientes\_Vehiculos where CliCod <> @CliNew and VehiculoCod = @Veh)

insert into Historial\_Clientes\_Vehiculos values (@CliOld, @Veh, (select Fecha from inserted))

delete from Clientes\_Vehiculos

where CliCod = @CliOld and VehiculoCod = @Veh

end

END

insert into Clientes\_Vehiculos Values(2,3, Cast(GETDATE()as date))

select \* from Historial\_Clientes\_Vehiculos



select \* from Clientes\_Vehiculos



Al ingresar un nuevo dueño para un vehículo, el trigger comprueba si el vehículo anteriormente tenía un dueño, si esto es correcto este dato se elimina y ademas es ingresado en la tabla historial clientes vehículos, mientras que el nuevo Cliente con el mismo vehículo es ingresado en la tabla Cliente vehículo.