

```

/*
Tecnicos(idTec, nomTec, sueldoTec, horasTec, stsTec)
Vehiculos(idVehic, marcaVehic, modeloVehic)
Equipos(idEq, dscEq, costoHoraEq, horasAcumEq)
Alinean(idAlin, idTec, idVehic, fchAlin, horasAlin, idEq)
Auditoria(idAudit, fecha, tabla, campo, anterior, actual, clave)

```

Utilizando el lenguaje T-SQL realizar los siguientes ejercicios:

1. Programar un procedimiento almacenado que reciba como parámetro un identificador de técnico y cargue en el campo horasTec el total de horas que lleva trabajadas, si dicho total supera las 72 horas, debe además poner en el campo stsTec el valor 'A' si no supera las 72 horas debe poner 'P'.

```
*/
```

```

CREATE PROCEDURE Ejer_1
@idTec int
AS
BEGIN
DECLARE @horasTec int

SELECT @horasTec=SUM(T.horasAlin)
FROM Alinean A
WHERE A.idTec=@idTec

IF @horasTec > 72
    UPDATE Tecnicos
    SET horasTec=@HorasTec,stsTec='A'
    WHERE idTec=@idTec
ELSE
    UPDATE Tecnicos
    SET horasTec=@HorasTec,stsTec='P'
    WHERE idTec=@idTec

END

```

```
/*
```

2. Programar una función que reciba como parámetro un identificador de equipo y retorne el costo total de horas trabajadas por dicho equipo, debe además realizar una consulta que muestre el identificador del equipo, su marca, modelo y dicho total utilizando la función creada.

```
*/
```

```

CREATE FUNCTION Ejer_2(@idEq int)
RETURNS int
AS
BEGIN
DECLARE @retorno int

SELECT @retorno=SUM(horasAlin)
FROM Alinean
WHERE idEq=@idEq

```

```
RETURN @retorno
END
```

```
SELECT E.idEq,E.dscEq,dbo.Ejer_2(E.idEq) as TotalHoras
FROM Equipos E
```

```
/*
```

3. Implementar un disparador que luego de ingresada una alineación acumule las horas del técnico y las horas del equipo utilizado. ↗

```
*/
```

```
CREATE TRIGGER Ejer_3
```

```
ON Alinean
```

```
AFTER insert
```

```
AS
```

```
BEGIN
```

```
UPDATE Tecnicos
```

```
SET horasTec=horasTec+(SELECT SUM(horasAlin)
```

```
FROM inserted
```

```
WHERE inserted.idTec=Tecnicos.idTec)
```

```
WHERE Tecnicos.idTec IN (SELECT idTec
```

```
FROM inserted)
```

```
UPDATE Equipos
```

```
SET horasAcumEq=horasAcumEq+(SELECT SUM(horasAlin)
```

```
FROM inserted
```

```
WHERE inserted.idEq=Equipos.idEq)
```

```
WHERE Equipos.idEq IN (SELECT idEq
```

```
FROM inserted)
```

```
END
```

```
/*
```

4. Implementar un disparador que no permita el ingreso de alineaciones para técnicos que tienen menos de 5 vehículos diferentes alineados en el año ↗  
corriente, ↗

se debe dejar un registro en la tabla auditoria los casos rechazados.

```
*/
```

```
CREATE TRIGGER Ejer_4
```

```
ON Alinean
```

```
INSTEAD OF insert
```

```
AS
```

```
BEGIN
```

```
INSERT INTO Auditoria SELECT getdate(),'Alinean','',inserted.idTec
```

```
FROM inserted,Alinean
```

```
WHERE inserted.idTec=Alinean.idTec and
```

```
YEAR(fchTrab)=YEAR(getdate())
```

```
GROUP BY getdate(),'Alinean','',inserted.idTec
```

```
HAVING COUNT(DISTINCT(idVehic)) < 5
```

```
ELSE
```

```
INSERT INTO Alinean SELECT idTec, idVehic, fchAlin, horasAlin, idEq
```

```
FROM inserted,Alinean
```

```
WHERE inserted.idTec=Alinean.idTec and
```

```
YEAR(fchAlin)=YEAR(getdate())
```

```
GROUP BY idTec, idVehic, fchAlin, horasAlin, idEq
```

---

```
HAVING COUNT(DISTINCT(idVehic)) >= 5
```

```
END
```

```
/*
```

```
5. M5. Mediante un disparador, permitir que se borre un vehiculo.
```

```
*/
```

```
CREATE TRIGGER Ejer_5
```

```
ON Vehiculos
```

```
INSTEAD OF delete
```

```
AS
```

```
BEGIN
```

```
    DELETE
```

```
    FROM Alinean
```


```
    WHERE Alinean.idVehic IN (SELECT idVehic
                              FROM deleted)
```

```
    DELETE
```

```
    FROM Vehiculos
```

```
    WHERE idVehic IN (SELECT idVehic
                      FROM deleted)
```

```
/*
```

```
6. Si se modifica el sueldo de un técnico, se debe dejar constancia en la tabla auditoria dicho cambio. 
```

```
*/
```

```
CREATE TRIGGER Ejer_6
```

```
ON Tecnicos
```

```
AFTER update
```

```
AS
```

```
BEGIN
```

```
    IF UPDATE(sueldoTec)
```

```
        INSERT INTO Auditoria SELECT getdate()
```

```
        (, 'Tecnicos', 'sueldoTec', deleted.sueldoTec, inserted.sueldoTec, inserted.idTec
```

```
        FROM inserted, deleted
```

```
        WHERE inserted.idTec=deleted.idTec
```

```
END
```