```
/*
Empleados(idEmp, nomEmp, sueldoEmp, horasEmp, stsEmp)
Obras(idObra, dscObra, dirObra)
Maquinas(idMaq, dscMaq, costoHoraMaq, horasAcumMaq)
Trabajan(idTrab, idEmp, idObra, fchTrab, horasTrab, idMaq)
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 empleado y cargue en el campo horasEmp el total de horas
 que lleva trabajadas, si dicho total supera las 48 horas, debe además poner en el
   campo stsEmp el valor 'Activo' si no supera las 48 horas debe poner 'Prueba'.
 */

 CREATE PROCEDURE Ejer_1
 @idEmp int
 AS
 BEGIN
 DECLARE @horasEmp int

 SELECT @horasEmp=SUM(T.horasTrab)
 FROM Trabajan T
 WHERE T.idEmp=@idEmp

 IF @horasEmp  > 48
    UPDATE Empleados
    SET horasEmp=@HorasEmp,stsEmp='Activo'
    WHERE idEmp=@idEmp
 ELSE
     UPDATE Empleados
     SET horasEmp=@HorasEmp,stsEmp='Prueba'
     WHERE idEmp=@idEmp

 END

 /*
 2. Programar una función que reciba como parámetro un identificador de máquina y
   retorne el costo total de horas trabajadas por dicha máquina,
 debe además realizar una consulta que muestre el identificador de la máquina, su
   descripción y dicho total utilizando la función creada.
 */

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

  SELECT @retorno=SUM(horasTrab)
  FROM Trabajan
  WHERE idMaq=@idMaq

  RETURN @retorno
 END
```

```
 SELECT M.idMaq,M.dscMaq,dbo.Ejer_2(M.idMaq) as TotalHoras
 FROM Maquinas

 /*
 3. Implementar un disparador que luego de ingresado un trabajo acumule las horas
   trabajadas del empleado y las horas trabajadas de la máquina
 */
 CREATE TRIGGER Ejer_3
 ON Trabajan
 AFTER insert
 AS
 BEGIN
  UPDATE Empleados
  SET horasEmp=horasEmp+(SELECT SUM(horasTrab)
                         FROM inserted
                         WHERE inserted.idEmp=Empleados.idEmp)
  WHERE empleados.idEmp IN (SELECT idEmp
                            FROM inserted)

  UPDATE Maquinas
  SET horasAcumMaq=horasAcumMaq+(SELECT SUM(horasTrab)
                                 FROM inserted
                                 WHERE inserted.idMaq=Maquinas.idMaq)
  WHERE maquinas.idMaq IN (SELECT idMaq
                           FROM inserted)
END

/*
4.  Implementar un disparador que no permita el ingreso de trabajos para empleados
  que tienen más de 10 obras diferentes en el año corriente,
se debe dejar un registro en la tabla auditoria los casos rechazados
*/

CREATE TRIGGER Ejer_4
ON Trabajan
INSTEAD OF insert
AS
BEGIN

            INSERT INTO Auditoria SELECT getdate(),'Trabajan','','',inserted.idEmp
                                  FROM inserted,trabajan
                                  WHERE inserted.idEmp=trabajan.idEmp and
                                        YEAR(fchTrab)=YEAR(getdate()
                                  GROUP BY getdate
                     (),'Trabajan','','',inserted.idEmp
                                  HAVING COUNT(DISTINCT(idObra)) > 10
 ELSE

            INSERT INTO Trabajan SELECT idEmp, idObra, fchTrab, horasTrab, idMaq
                                  FROM inserted,trabajan
                                  WHERE inserted.idEmp=trabajan.idEmp and
                                        YEAR(fchTrab)=YEAR(getdate()
                                  GROUP BY idEmp, idObra, fchTrab, horasTrab, idMaq
                                  HAVING COUNT(DISTINCT(idObra)) <= 10
```

```
END

/*
5.  Mediante un disparador, permitir que se borre un empleado.
*/

CREATE TRIGGER Ejer_5
ON Empleados
INSTEAD OF delete
AS
BEGIN

  DELETE
  FROM Trabajan
  WHERE Trabajan.idEmp IN (SELECT idEmp
                           FROM deleted)

  DELETE
  FROM Empleados
  WHERE idEmp IN (SELECT idEmp
                  FROM deleted)


/*
6.  Si se modifica el sueldo de un empleado, se debe dejar constancia en la tabla
  auditoria dicho cambio.
*/

CREATE TRIGGER Ejer_6
ON Empleados
AFTER update
AS
BEGIN
  IF UPDATE(sueldoEmp)

    INSERT INTO Auditoria  SELECT getdate
      (),'Empleados','sueldoEmp',deleted.sueldoEmp,inserted.sueldoEmp,inserted.idEm
      p
                           FROM inserted,deleted
                           WHERE inserted.idEmp=deleted.idEmp

END
```