UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

EVALUACIÓN	Parcial	**GRUPO**	Matutinos	**FECHA**	Diciembre 23
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	- Puntos: 35 - Duración 2 horas - Sin material				

Se desea implementar un sistema on-line para la toma de evaluaciones en la universidad para todas las asignaturas. Se le brinda el siguiente MR:

Asignaturas (CodigoAsignatura, nombreAsignatura, profesorResponsable)

FK: profesorResponsable -> Profesores.nroDocumento

Profesores (NroDocEstudiante, nombre, apellido, fechaNacimiento, fechaIngreso, Titulo)

Estudiantes (NroDoc, nombre, apellido, fechaNacimiento)

Cuestionarios (TituloCuestionario, FechaPublicacion,fechaLimite, codigoAsignatura, nroDocProfesor)

FK: nroDocProfesor-> Profesores.nroDocumento

FK: CodigoAsignatura -> Asignaturas.CodigoAsignatura

Preguntas (CodigoPregunta, puntaje, textoPregunta, nroDocProfesor , CodigoAsignatura,cuestionario)

FK: nroDocProfesor-> Profesores.nroDocumento
FK: CodigoAsignatura -> Asignaturas.CodigoAsignatura
FK: cuestionario -> Cuestionario.tituloCuestionario

Respuestas (IDRespuesta, textoRespuesta, esCorrecta, codigoPregunta)

FK: CodigoPregunta ->Preguntas.Codigopregunta

CuestionariosContestados (CodigoPregunta,IdRespuesta, NroDocEstudiante, fechaRealizacion)

FK: CodigoPregunta -> Preguntas.Codigopregunta
FK: NroDocEstudiante -> Estudiantes.NroDocEstudiante
FK: idRespuesta -> Respuestas.IdRespuesta

Se pide:

1. Crear una vista que devuelva toda la información de los estudiantes que realizaron todos los cuestionarios.

```
CREATE VIEW EstudiantesRealizaronTodosCuestionarios AS
SELECT E.*
FROM Estudiantes E
JOIN CuestionariosContestados CC ON E.NroDoc = CC.NroDocEstudiante
JOIN Preguntas p on p.CodigoPregunta=cc.CodigoPregunta
GROUP BY E.NroDoc, E.nombre, E.apellido, e.FechaNacimiento
HAVING COUNT(distinct p.TituloCuestionario) = (SELECT COUNT( *)
                                               FROM Cuestionarios)
```

2. Realizar una función o procedimiento según corresponda, que reciba un número de documento de estudiante y un título del cuestionario. Y devuelva el puntaje total del estudiante. Tener en cuenta todas las **preguntas correctas** del estudiante. **(10 puntos)**

```
CREATE FUNCTION dbo.CalcularPuntajeEstudiante(
        @NroDocEstudiante INT, @TituloCuestionario NVARCHAR(100))
RETURNS INT AS
BEGIN
        DECLARE @PuntajeTotal INT;
        SELECT @PuntajeTotal = ISNULL(SUM(P.puntaje), 0)
        FROM CuestionariosContestados CC
        JOIN preguntas p on p.codigoPregunta = cc.codigoPregunta
        JOIN cuestionario c on p.cuestionario=c.TituloCuestionario
        JOIN Respuestas r on CC.idRespuesta = r.idRespuesta and
cc.codigoPregunta = p.codigoPregunta
```

```
        where cc.NroDocEstudiante = @NroDocEstudiante and c.tituloCuestionario =
@TituloCuestionario AND r.EsCorrecta=1

        RETURN @PuntajeTotal;

END;
```

3. Realizar una función o procedimiento según corresponda, que reciba por parámetro un cuestionario, un **puntaje total** y actualice el puntaje de cada pregunta del cuestionario de manera **distribuida**. (Es decir, que cada pregunta tenga el mismo puntaje y la suma de todas estas sea el puntaje total.)

```
CREATE PROCEDURE ActualizarPuntajePreguntas (

        @tituloCuestionario NVARCHAR(100),

        @puntajeTotal INT

)

AS

BEGIN

        DECLARE @cantidadPreguntas INT

        -- Obtener la cantidad de preguntas en el cuestionario

        SELECT @cantidadPreguntas = COUNT(*)

        FROM Preguntas

        WHERE cuestionario = @tituloCuestionario

        -- Actualizar el puntaje de cada pregunta de manera distribuida

        UPDATE Preguntas

        SET puntaje = @puntajeTotal / @cantidadPreguntas

        WHERE cuestionario = @tituloCuestionario

END
```

4. Realizar un trigger que permita mantener un historial de cambios en la tabla Estudiantes. Cada vez que se inserta o actualiza un registro se debe dejar registro

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

en la tabla de auditoría. Suponga dada la tabla auditoría según considere.**(10 puntos)**

```
CREATE TABLE Auditoria (

ID INT identity PRIMARY KEY IDENTITY(1,1),

 NroDoc INT,

 NombreNuevo NVARCHAR(50),

 ApellidoNuevo NVARCHAR(50),

FechaNacNueva DATETIME,

NombreAnt NVARCHAR(50),

 ApellidoAnt NVARCHAR(50),

FechaNacAnt DATETIME,

 FechaCambio DATETIME,

usuarioQueModifico  NVARCHAR(50));


CREATE TRIGGER tr_Estudiantes_InsertUpdate

ON Estudiantes AFTER INSERT, UPDATE

AS BEGIN

        if Exists (select * from inserted ) and exists (select * from deleted) -- update

         INSERT INTO Auditoria

            SELECT (i.NroDoc, i.NombreNuevo, i.ApellidoNuevo, i.FechaNacNueva,
        d.NombreAnt, d.ApellidoAnt, d.FechaNacAnt GETDATE(), host_name()  FROM
        inserted i join deleted d on i.nroDoc = d.nroDoc;

        else

    INSERT INTO Auditoria ( NroDoc, NombreNuevo, ApellidoNuevo, FechaNacNueva,
FechaCambio, usuarioQueModifico)

    SELECT i.NroDoc, i.NombreNuevo, i.ApellidoNuevo, i.FechaNacNueva, GETDATE()
, host_name(), FROM inserted i;
```

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 902 15 05 Fax 908 13 70
www.ort.edu.uy

```
 PRINT 'Se ha insertado o actualizado un estudiante.';

END;
```