Luis Ivan Herrera Equihua

12386391

Descripción breve

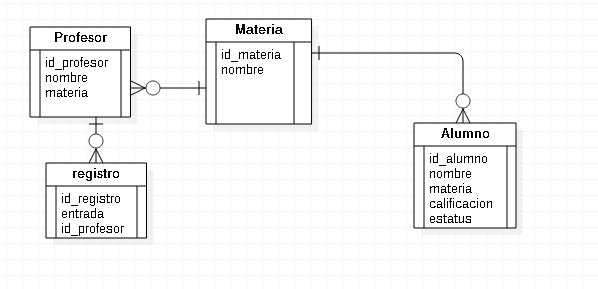
Una universidad desea elaborar un sistema data-céntrico para validar calificaciones de alumnos y la hora de registro de los profesores. El profesor se identifica por su clave de profesor, y se le asigna nombre de materia que impartirá y alumnos registrados a dicha materia, además de un campo de tipo Date para que diariamente actualize su hora de entrada a su materia (campo transitivo). La materia se identifica por el nombre de la materia y un número de secuencia consecutivo iniciando desde 1. Los alumnos se identifican con su número de cuenta y por el identificador de la materia, que debe de coincidir con el del profesor. También cuenta con campo nombre, calificación final y estatus. Este último campo puede ser solamente “aprobado” o “reprobado”, dependiendo de la calificación final

Bases de datos 2

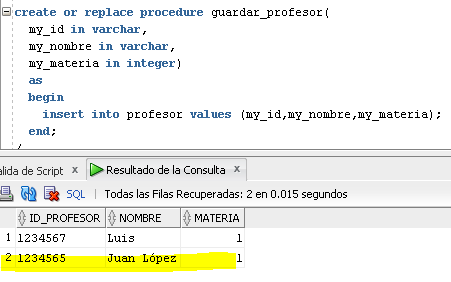
Triggers de automatización 2

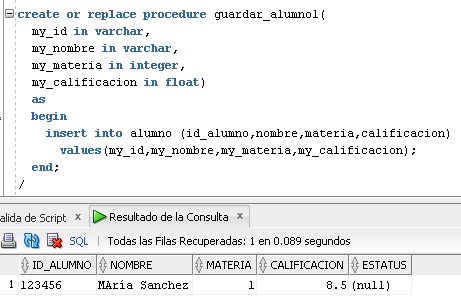


### El diagrama Entidad-relación de este problema

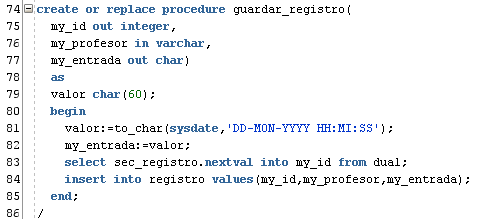


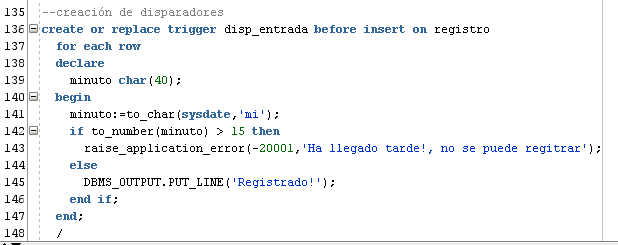
### Guardar un maestro cuyo nombre sea Juan López y que esté asignado a la materia Programación. Tendrá un solo alumno cuyo nombre es Maria Sanchez y cuya calificación final es 8.5.



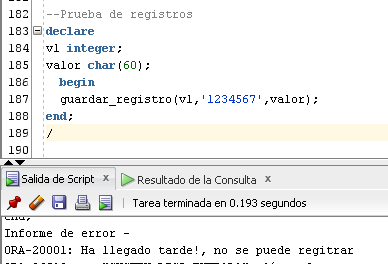


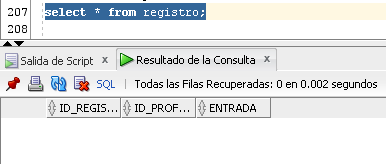
### Genere un disparador de tipo automatización para impedir que el profesor se pueda registrar después del minuto 15 de cada hora (para probarlo, cambia el minuto 15, por los 15 segundos de cada minuto).



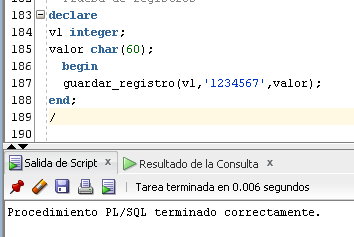


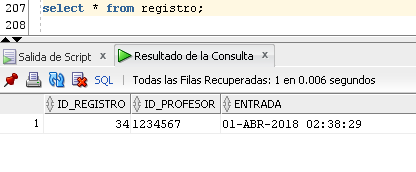
Prueba pasados los minutos limite





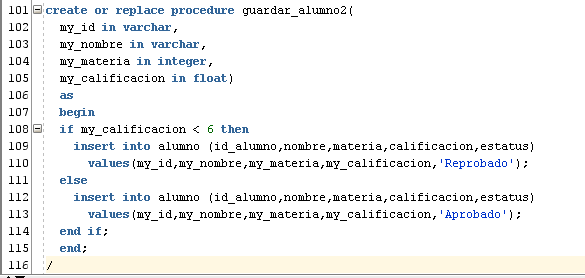
Antes de los minutos limite

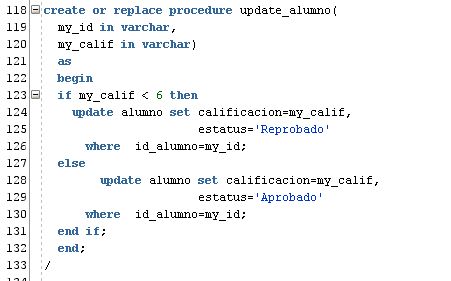




### Genere un disparador que dependiendo de la calificación inserte automáticamente aprobado o reprobado en el campo “estatus”, sin que este se tenga que hacer manualmente, todo ello dependiendo de la calificación ingresada.

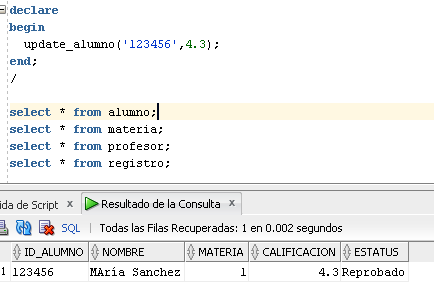
intenté de todas las maneras para hacer un trigger y tenía el error de que la tabla se encontraba mutando inclusive usando cursores, tablas :new-:old, after y before así que la validación la puse dentro del SP :^)





Prueba





## Código

--PRactica automatización

--drop table profesor;

create table profesor(

id\_profesor varchar(9) primary key,

nombre varchar(60),

materia integer);

create table registro(

id\_registro integer primary key,

id\_profesor varchar(9),

entrada varchar(60));

create table alumno(

id\_alumno varchar(9) primary key,

nombre varchar (60),

materia integer,

calificacion float,

estatus varchar(10));

create table materia(

id\_materia integer primary key,

nombre varchar (80)

);

--Creación de foreign keys

alter table profesor

add constraint fk1\_profesor\_materia

foreign key (materia)

references materia(id\_materia);

alter table registro

add constraint fk1\_registro\_profesor

foreign key (id\_profesor)

references profesor (id\_profesor);

alter table alumno

add constraint fk1\_alumno\_materia

foreign key (materia)

references materia (id\_materia);

--creación de secuencia para la materia y registros

create sequence sec\_materia

start with 1

increment by 1

nomaxvalue;

create sequence sec\_registro

start with 1

increment by 1

nomaxvalue;

--procedimiento almacenado para guardar

create or replace procedure guardar\_materia(

my\_id out integer,

my\_nombre in varchar)

as

begin

select sec\_materia.nextval into my\_id from dual;

insert into materia values (my\_id,my\_nombre);

end;

/

select \* from profesor;

create or replace procedure guardar\_profesor(

my\_id in varchar,

my\_nombre in varchar,

my\_materia in integer)

as

begin

insert into profesor values (my\_id,my\_nombre,my\_materia);

end;

/

create or replace procedure guardar\_registro(

my\_id out integer,

my\_profesor in varchar,

my\_entrada out char)

as

valor char(60);

begin

valor:=to\_char(sysdate,'DD-MON-YYYY HH:MI:SS');

my\_entrada:=valor;

select sec\_registro.nextval into my\_id from dual;

insert into registro values(my\_id,my\_profesor,my\_entrada);

end;

/

create or replace procedure guardar\_alumno1(

my\_id in varchar,

my\_nombre in varchar,

my\_materia in integer,

my\_calificacion in float)

as

begin

insert into alumno (id\_alumno,nombre,materia,calificacion)

values(my\_id,my\_nombre,my\_materia,my\_calificacion);

end;

/

create or replace procedure guardar\_alumno2(

my\_id in varchar,

my\_nombre in varchar,

my\_materia in integer,

my\_calificacion in float)

as

begin

if my\_calificacion < 6 then

insert into alumno (id\_alumno,nombre,materia,calificacion,estatus)

values(my\_id,my\_nombre,my\_materia,my\_calificacion,'Reprobado');

else

insert into alumno (id\_alumno,nombre,materia,calificacion,estatus)

values(my\_id,my\_nombre,my\_materia,my\_calificacion,'Aprobado');

end if;

end;

/

create or replace procedure update\_alumno(

my\_id in varchar,

my\_calif in varchar)

as

begin

if my\_calif < 6 then

update alumno set calificacion=my\_calif,

estatus='Reprobado'

where id\_alumno=my\_id;

else

update alumno set calificacion=my\_calif,

estatus='Aprobado'

where id\_alumno=my\_id;

end if;

end;

/

--creación de disparadores

create or replace trigger disp\_entrada before insert on registro

for each row

declare

minuto char(40);

begin

minuto:=to\_char(sysdate,'mi');

if to\_number(minuto) > 15 then

raise\_application\_error(-20001,'Ha llegado tarde!, no se puede regitrar');

else

DBMS\_OUTPUT.PUT\_LINE('Registrado!');

end if;

end;

/

create or replace trigger disp\_calificacion before update on alumno

for each row

declare

begin

if :new.calificacion < 6 then

update alumno set estatus='Reprobado' where :old.id\_alumno=:new.id\_alumno;

else

update alumno set estatus='Aprobado' where :old.id\_alumno=:new.id\_alumno;

end if;

end;

/

drop trigger disp\_calificacion;

declare

valor integer;

begin

guardar\_materia(valor,'Programación');

end;

/

select \* from materia;

begin

guardar\_profesor('1234565','Juan López',1);

end;

/

select \* from profesor;

--Prueba de registros

declare

v1 integer;

valor char(60);

begin

guardar\_registro(v1,'1234567',valor);

end;

/

--Prueba de reprobar

declare

begin

guardar\_alumno2('123456','MAría Sanchez',1,8.5);

end;

/

declare

begin

update\_alumno('123456',4.3);

end;

/

select \* from alumno;

select \* from materia;

select \* from profesor;

select \* from registro;

drop trigger disp\_calificacion;