

Fundamentos de Ingeniería de Software

Profesor: Raimundo Vega/Juan Pablo Salazar

Ingeniería Civil Informática.

Universidad Austral de Chile

**Todo Transportes Procedimiento Matricular**

**Integrantes:** José Luis Acuña Oyarce.

Luis Barrientos Fajardo.

Kurt Poehler Widemann.

**Fecha**: 05 de marzo de 2015.

USE `TTPT` ;

-- -----------------------------------------------------

-- procedure MATRICULAR

-- -----------------------------------------------------

USE `TTPT`;

DROP procedure IF EXISTS `TTPT`.`MATRICULAR`;

DELIMITER $$

USE `TTPT`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `MATRICULAR`(

IN nroReg VARCHAR(6), #Numero de Registro de Matricula

IN nombre VARCHAR(255),

IN rut VARCHAR(11),

IN fecha\_Nac DATE, #Fecha de Nacimiento

IN telefono NUMERIC,

IN email VARCHAR(255),

IN tipoCliente VARCHAR(10), #Franquicia Tributaria

IN licencia VARCHAR(40),

IN fecha DATE, #Fecha Actual

IN usuario VARCHAR(45), #Usuario que agregó al Nuevo Estudiante

IN curso VARCHAR(6),

IN hraTeoria TIME, #Horario de Clase Teórica

IN hraPractica TIME, #Horario de Clase Práctica

IN l\_v BOOL, #Lunes-Viernes

IN n\_fotos BOOL, #Fotos Tamaño Carnet

IN c\_estudios BOOL, #Certificado de Estudio

IN c\_notarial BOOL, #Certificado Notarial

IN c\_CI BOOL, #Copia de Cedula de Identidad

IN descuento INT,

IN docuPago VARCHAR(10), #Documento de Pago (Boleta o Factura)

IN medioPago VARCHAR(25), #Medio de Pago (Efectivo, Documento, Beca, SENCE, Tarjeta)

IN pagado BOOL, #Si el pago fue realizado o no

IN codigoPago VARCHAR(6) #Código asociado al pago

)

BEGIN

declare error\_msg varchar(20) default ""; #Declara una variable para poder modificar el mensaje de error

declare cuposLlenos bool;

declare cursoEnMarcha bool;

declare id\_cliente int;

declare id\_compra int;

declare id\_documento int;

declare listo bool default false;

declare id\_clase int;

declare tipoClase varchar(1);

declare clases cursor for

select cl.idCLASE, cl.Tipo

from clase cl

where cl.Curso = curso;

declare continue handler for sqlstate '02000' set listo = true; #Asigna a la variable Listo el Listener del evento que se dispara al terminar de recorrerse el cursor

DECLARE EXIT HANDLER FOR SQLEXCEPTION #Declara un listener de escape para el error SQLException

#Cuerpo del listener

BEGIN

GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED\_SQLSTATE,

@errno = MYSQL\_ERRNO, @text = MESSAGE\_TEXT;

SET @full\_error = CONCAT("ERROR ", @errno, " (", @sqlstate, "): ", @text);

SELECT @full\_error;

END;

start transaction; #Cuerpo del Procedimiento

#Revisa que la cantidad de estudiantes matriculados no exceda a la de los cupos en el curso

select if((c.cupos-count(m.curso)) = 0 , true, false)

into cuposLlenos

from curso c, matricula m

where c.codigo = curso And m.curso = curso;

#Revisa que el curso que está intentando agregar todavía esté vigente

select if( datediff((select c.FechaTermino from curso c where c.codigo = curso),fecha) > 0, true, false )

into cursoEnMarcha;

if cuposLlenos then

rollback;

set error\_msg = 'Cupos llenos';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = error\_msg;

end if;

#Este if indica si el curso ya no está en marcha

if !cursoEnMarcha then

rollback;

set error\_msg = 'Curso terminó';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = error\_msg;

end if;

#Comienza el proceso de inserción en las tablas de la base de datos

insert into cliente (`Nombre`,`RUT`,`Fecha\_Nac`,`Telefono`,`e\_Mail`,`TipoCliente`)

values (nombre, rut, fecha\_Nac, telefono, email, tipoCliente);

select last\_insert\_id() into id\_cliente;

insert into compra (`Cliente`,`Producto`,`Fecha`)

values (id\_cliente, licencia, fecha);

select last\_insert\_id() into id\_compra;

insert into pago (`Codigo`, `Medio\_Pago`, `Compra`, `Fecha`, `Descuento`)

values (codigoPago, medioPago, id\_compra, fecha, descuento);

if STRCMP(docuPago, 'Boleta') = 0 then

insert into boleta (`Codigo`) value (codigoPago);

else

insert into factura (`Codigo`) value (codigoPago);

end if;

insert into documento (`Tipo`,`Funcionario`,`Fecha`)

values ('Matrícula', (select u.Funcionario from usuario u where u.Nombre = usuario), fecha);

select last\_insert\_id() into id\_documento;

insert into matricula (`Documento`, `Curso`, `Codigo`, `HoraTeoria`, `HoraPractica`, `CodigoCompra`, `FirmaGerencia`, `FirmaCliente`, `L\_V`)

values (id\_documento, curso, nroReg, hraTeoria, hraPractica, id\_compra, true, true, l\_v);

insert into estudiante (`idESTUDIANTE`,`FotosCarnet`,`CertEstudios`,`CertNotarial`,`copiaCI`)

values (nroReg, n\_fotos, c\_estudios, c\_notarial,c\_CI);

#Termina el proceso de insertar en las tablas de la base de datos

open clases; #Abre el cursor clases

clases\_loop: loop #E itera sobre él

fetch clases into id\_clase, tipoClase; #Descarga el contenido del cursor en las variables “id\_clase” y “tipoClase”

if listo then leave clases\_loop; end if; #De haber recorrido todo el cursor sale del ciclo.

if STRCMP(tipoClase, 'T') = 0 then #Si el tipo de clase es teórica entonces asocia al estudiante con una clase teórica

insert into teoria\_estudiante (`Teoria`,`Estudiante`) values (id\_clase, nroReg);

elseif STRCMP(tipoClase, 'P') = 0 then #Si la clase es práctica entonces asocia al estudiante con una clase práctica

insert into practica\_estudiante (`Practica`,`Estudiante`) values (id\_clase, nroReg);

else #En el remoto caso de producirse un error, y se corrompiesen los datos, cualquier “tipo de clase” extra finaliza el proceso y comunica el error

rollback;

set error\_msg = concat('Valor desconocido ',tipoClase);

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = error\_msg;

leave clases\_loop;

end if;

end loop clases\_loop;

close clases;

commit;

END$$

DELIMITER ;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;