



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`,`copiaCl`)
    values (nroReg, n_fotos, c_estudios, c_notarial,c_Cl);
```

#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;

```