

## Estructura de Tablas

Dada la tabla de asignaturas

*Asignaturas ( nombre(PK), nCreditos)*

Donde **nombre** es la clave y representa el nombre de la asignatura.

la tabla de grupos

*Grupos ( idGrupo, asignatura (FK ----> Asignaturas), plazasLibres )*

*PK compuesta = idGrupo + asignatura*

Que representa los grupos de cada asignatura, y donde **plazasLibres** indica el número de plazas libres que quedan en ese grupo para esa asignatura.

y de Matriculas:

*Matriculas( idMatricula(PK), alumno, asignatura, grupo)*

*FK Compuesta grupo+asignatura--> Grupos*

Donde **IdMatricula** se obtiene a partir de la secuencia **seq\_matricula**

(hemos quitado una cuarta tabla de alumnos para simplificar).

El script que crea las tablas está en [sql/script.sql](#) y el *main* ya lo corre automáticamente al llamar a *pruebaEjercicio6()*, que a su vez llama a *creaTablas()*.

## Descripción General

En *lsi.ubu.Esqueleto\_Problema6.java* está el fichero que se te pide completar, y en *lsi.ubu.Problema6.java* está la solución. Del mismo, se pide implementar el método:

```
public static void matricular(String p_alumno, String p_asig, int p_grupo) throws
SQLExceptionProblema6
```

que dado el alumno, asignatura y grupos que toma como argumentos, intenta insertar la fila correspondiente en la tabla de matriculas, y además decrementa el número de alumnos del grupo de esa asignatura en uno.

## Algoritmo

Para ello, se sugiere seguir estos pasos (pero se puede hacer de otras formas):

1. Insertar una fila en la tabla de matriculas para el alumno *p\_alumno*, la asignatura *p\_asig* y el grupo *p\_grupo*.
2. Actualizar la tabla de grupos decrementando el campo **plazasLibres** en una unidad para el grupo *p\_grupo* de la asignatura *p\_asig*.

El método tomará una conexión del pool al principio, y cometerá la transacción. Al finalizar cerrará todos los recursos utilizados.

## Excepciones

El método lanza sólo excepciones del tipo *SQLExceptionProblema6* que son dos posibles (Ver *lsi.ubu.SQLExceptionProblema6.java*):

1. La primera registrará el problema de que **no exista el grupo o la asignatura pasadas por parámetro**. Su código será el "1" y el mensaje de error **"Probablemente la asignatura o el grupo no exista."**.

Para detectar este problema la clase *lsi.ubu.OracleTableError.java* tiene codificado el

error **2291** de Oracle como ***FK\_VIOLATED***, que precisamente es el error que ocurre (entre otras ocasiones) cuando se viola una clave ajena por inserción de una fila hija sin la correspondiente fila padre. Se valora que tu solución no dependa de que este error sea codificado como 2291, usando para ello el método `checkExceptionToCode` y la clase `OracleTableError.java`.

2. La segunda registrará el problema de intentar matricular a un alumno en un grupo sin plazas vacantes. Su código será el "2" y el mensaje de error **"No hay plazas en ese grupo."**.

El resto de excepciones serán tratadas dentro del propio método.

Toda excepción sea del tipo que sea retrocederá la transacción.

- Las excepciones **que no sean** del tipo `SQLExceptionProblema6` dejará el mensaje del error con `1.error(...)` en el *logger*.
- Y las **que sean** de tipo `SQLExceptionProblema6`, se lanzarán o propagarán al main que hace la llamada al método.

## Pruebas automáticas

Las pruebas automáticas se invocan desde el *main* y están en el método `pruebaEjercicio6`. Aunque no hayas empezado a hacer nada el fichero `Problema6.java` compila y ya te muestra los mensajes de las pruebas aún no superadas. Esas pruebas hacen 4 cosas

1. Matricular a un alumno en un grupo lleno (el grupo 1 de OFIM). La prueba comprobará que salta la excepción `SQLExceptionProblema6`, indicando que no hay plazas en ese grupo (por eso, de momento, al ejecutarlo, te sale NO se da cuenta de que el grupo esta lleno MAL).
2. Matricular a un alumno en un grupo que no existe ( el grupo 3 de OFIM). La prueba comprobará que salta la excepción `SQLExceptionProblema6`, indicando que no o no existe la asignatura o no existe el grupo (por eso, de momento, al ejecutarlo, te sale NO se da cuenta de que el grupo no existe MAL).
3. Matricular a un alumno en una asignatura que no existe (ALGEBRA). La prueba comprobará que salta la excepción `SQLExceptionProblema6`, indicando que no o no existe la asignatura o no existe el grupo (por eso, de momento, al ejecutarlo, te sale NO se da cuenta de que la asignatura no existe MAL).
4. Finalmnete hacemos una matricula sin problemas, (grupo1 de FPROG). En ese caso no tiene que saltar ninguna excepción y ese grupo quedará con 3 plazas libres, y además la fila de esa matrícula se habrá insertado. La prueba genera un *String* con el contenido de la base de datos y comprueba si tiene todos estos cambios. De momento te saldrá Matricula MAL a la espera que programes correctamente el método.

El mensaje que sale cuando todo funciona (salvo por la fecha y hora de la excepción que se registra en el *logger*) es:

```
-- Mensajes de SQL*Plus creando y borrando tablas que omitimos--
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 -
Production
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 -
Production
ERROR lsi.ubu.SQLExceptionProblema6 <init>    No hay plazas en ese grupo.
Se da cuenta de que el grupo esta lleno OK
ERROR lsi.ubu.SQLExceptionProblema6 <init>    Probablemente la asignatura o el grupo
no exista.
Se da cuenta de que el grupo no existe OK
ERROR lsi.ubu.SQLExceptionProblema6 <init>    Probablemente la asignatura o el grupo
no exista.
Se da cuenta de que la asignatura no existe OK
Matricula OK
FIN.....
```

(el mensaje rojo del *logger* podría cambiar de posición al ejecutarse en paralelo en otro hilo sin que nos demos cuenta).