

ESCUELA COLOMBIANA DE INGENIERIA
MODELOS Y BASES DE DATOS
TRANSACCIONES Y SEGURIDAD
S13-S14: 2020-02

En este trabajo vamos a trabajar en una versión simplificada de un sistema de inscripciones (sin manejo de prerequisites). Estudie la definición de las tablas y el procedimiento INSCRIBIR.

<pre>CREATE TABLE MATERIAS(sigla CHAR(4) NOT NULL, nombre VARCHAR(30) NOT NULL); ALTER TABLE MATERIAS ADD CONSTRAINT PK_MATERIAS PRIMARY KEY(sigla); CREATE TABLE GRUPOS(materia CHAR(4) NOT NULL, numero NUMBER(2) NOT NULL, capacidad NUMBER(2) NOT NULL, inscritos NUMBER(2) NOT NULL); ALTER TABLE GRUPOS ADD CONSTRAINT PK_GRUPOS PRIMARY KEY (materia,numero); ALTER TABLE GRUPOS ADD CONSTRAINT FK_GRUPOS_MATERIAS FOREIGN KEY(materia) REFERENCES MATERIAS(sigla);</pre>	<pre>CREATE TABLE ESTUDIANTES(codigo NUMBER(7) NOT NULL, nombres VARCHAR(50)) NOT NULL; ALTER TABLE ESTUDIANTES ADD CONSTRAINT PK_ESTUDIANTES PRIMARY KEY(codigo); CREATE TABLE INSCRIPCIONES(materia CHAR(4) NOT NULL, numero NUMBER(2) NOT NULL, estudiante NUMBER(7) NOT NULL); ALTER TABLE INSCRIPCIONES ADD CONSTRAINT PK_INSCRIPCIONES PRIMARY KEY(materia,estudiante); ALTER TABLE INSCRIPCIONES ADD CONSTRAINT FK_INSCRIPCIONES_ESTUDIANTES FOREIGN KEY(estudiante) REFERENCES ESTUDIANTES(codigo); ALTER TABLE INSCRIPCIONES ADD CONSTRAINT FK_INSCRIPCIONES_GRUPOS FOREIGN KEY(materia,numero) REFERENCES GRUPOS(materia, numero);</pre>
---	---

```
CREATE OR REPLACE
PROCEDURE INSCRIBIR(xEstudiante IN NUMBER, xMateria IN CHAR, xNumero IN NUMBER) IS
  xInscritos NUMBER(2);
  xCapacidad NUMBER(2);

BEGIN
  SELECT inscritos, capacidad INTO xInscritos, xCapacidad
    FROM GRUPOS
    WHERE materia=xMateria AND numero=xNumero;
  IF (xInscritos < xCapacidad) THEN
    INSERT INTO INSCRIPCIONES(materia,numero,estudiante)
      VALUES (xMateria,xNumero,xEstudiante);
    UPDATE GRUPOS SET
      inscritos=inscritos+1
      WHERE materia=xMateria AND numero=xNumero;
  END IF;

END INSCRIBIR;
```

TRANSACCIONES

Considerando el código anterior:

1. ¿Qué posibles errores podrían presentar las instrucciones en los tres puntos marcados?

(SELECT 1<-, INSERT <--2, UPDATE <--3) Escriban las condiciones y las restricciones de integridad asociadas.

- Que no existan ni la materia ni el número, la materia tiene longitud 2 y xMateria no presenta una longitud por lo cual sería 10 y al permitir una longitud mayor eso podría causar un error, al igual que con el número que es longitud 2 y xNumero es longitud 10 por lo cual ocurriría el mismo error.
- En el insert ocurre lo mismo con xMateria y xNumero y también con xEstudiante que tiene longitud 10 y Estudiante tiene longitud 7 por lo cual se intentarían inserciones erróneas.
- En el Update ocurre lo mismo que ocurría en el insert y en el select ya que se intentaría actualizaciones con longitudes diferentes

2. ¿Qué posible error no se ha comunicado? Modifique el código para considerarlo.

Que xInscritos sea mayor a xCapacidad

```
CREATE OR REPLACE
PROCEDURE INSCRIBIR(xEstudiante IN NUMBER, xMateria IN CHAR, xNumero IN NUMBER) IS
  xInscritos NUMBER(2);
  xCapacidad NUMBER(2);
```

```

BEGIN
SELECT inscritos, capacidad INTO xInscritos, xCapacidad    <-1
        FROM GRUPOS
        WHERE materia=xMateria AND numero=xNumero;
IF (xInscritos < xCapacidad) THEN
    INSERT INTO INSCRIPCIONES(materia,numero,estudiante)    <-2
        VALUES (xMateria,xNumero,xEstudiante);
    UPDATE GRUPOS SET    <-3
        inscritos=inscritos+1
        WHERE materia=xMateria AND numero=xNumero;
ELSE
    THEN RAISE_APPLICATION_ERROR(-20001, 'datos no insertados')
END IF;

END INSCRIBIR;

```

3. Incluya las instrucciones básicas para manejar TODOS los posibles casos de excepción de este procedimiento. Generen una única excepción de aplicación con el mensaje apropiado.

```

BEGIN
    --BEGIN TRANSACTION
    Values(xMateria,cNumero,xEstudiante);
    IF (SQL%ROWCOUNT=0) THEN
        THEN RAISE_APPLICATION_ERROR(-20001, 'datos no insertados')
    END IF;

    SELECT inscritos, capacidad INTO xInscritos, xCapacidad
        FROM GRUPOS
        WHERE materia=xMateria AND numero=xNumero;
    IF (xInscritos < xCapacidad) THEN
        INSERT INTO INSCRIPCIONES(materia,numero,estudiante)
            VALUES (xMateria,xNumero,xEstudiante);

        UPDATE GRUPOS SET inscritos=inscritos+1
            WHERE materia=xMateria AND numero=xNumero;
        IF (SQL%ROWCOUNT=0) THEN
            RAISE_APPLICATION_ERROR(-20003, 'datos no actualizados')
        END IF;

    COMMIT;

    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE_APPLICATION_ERROR(-20999, SQLERRM);
END INSCRIBIR;

```

4. ¿Por qué inscribir debería ser una transacción? Incluya las instrucciones necesarias: COMMIT, ROLLBACK
 Por las propiedades de las transacciones(ACID).

CONCURRENCIA

Corrección

Suponiendo que: MBDA 01 tiene 20 cupos, ya están inscritos 19 estudiantes y los estudiantes 8754623 y 4859632 existen y no están inscritos a MBDA

1. ¿Cuáles serían los posibles resultados de ejecución correcta de los procesos de inscripción de estos dos estudiantes? Expliquen su respuesta.

Que alguno de los estudiantes obtuviera el cupo su capacidad se llenara por lo cual el otro estudiantes obtendría el error “dato no insertado”, esto sería un correcto comportamiento debido a que no se está bloqueando la base de datos y se está manteniendo una integridad en los datos (La cantidad de los inscritos no superan el cupo establecido).

Problemas/Soluciones

Considerando la ejecución concurrente propuesta (sólo los pasos posibles, actualice el resto):

1. Si el nivel de aislamiento de las transacciones es READ UNCOMMITTED : Lectura no confirmada (sin bloqueos) ¿cómo se comportan? ¿es correcto? ¿ilustra algún problema clásico?

Si inscriben de manera simultánea podría suceder lo siguiente:

1. ingresarían ambos o ninguno de los dos
2. Se cancelaría ambas transacciones
3. Se cancelaría una de las dos transacciones

Problema Clásico:

Dependencia no confirmada ya que es el nivel de aislamiento

2. Si el nivel de aislamiento de las transacciones es REPEATABLE READ : Lectura repetible : (Bloqueo exclusivo para actualizar y compartido para leer) ¿cómo se comportan? ¿es correcto? ¿ilustra algún problema clásico?

Bloqueo Mortal y Lectura Fantasma

3. Si en el caso 2. se adiciona el mecanismo de control de bloqueo mortal, ¿cómo se comportan? ¿es correcto?

Ingresa un solo estudiante

MBDA 01 tiene 20 cupos y ya están inscritos 19 estudiantes Los estudiantes existen y no se han inscrito a MBDA.		
	TA. INSCRIBIR 8754623 MBDA 01	TB. INSCRIBIR 4859632 MBDA 01
t1 t2	1	-
t3 t4	-	1
t5 t6	2	-
t7	-	2
	3	-
	-	3
	-	-

SEGURIDAD

Considerando que el mecanismo de seguridad seleccionado es de permisos mínimos sobre datos. Escriba los permisos que debería tener cada uno de los actores considerando el diagrama de casos de uso y las reglas de negocio.

- Los decanos y coordinadores pueden adicionar y modificar materias. El único dato que se permite modificar es el nombre. El decano es el único responsable eliminar materias.

GRANT INSERT, UPDATE (nombre), DELETE ON Materias TO Decano

GRANT INSERT, UPDATE (nombre) ON Materias TO Coordinador

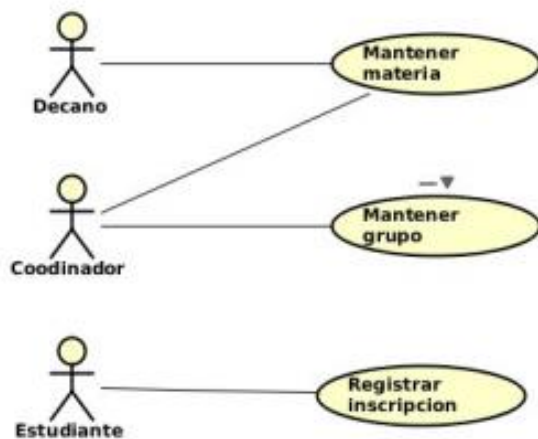
- Los coordinadores pueden crear los grupos asociados a las materias. El único dato que pueden modificar es la capacidad, respetando los estudiantes inscritos. Los grupos se pueden eliminar si no tienen inscripciones.

Vista:

```
CREATE VIEW GruposAsociados AS (  
SELECT *  
FROM Materias, Grupos  
WHERE Materias.sigla = Grupos.materia )
```

Permisos:

```
GRANT INSERT ON GruposAsociados TO Coordinador;  
GRANT UPDATE (capacidad) ON Grupos TO Coordinador;  
GRANT DELETE ON Grupos TO Coordinador;
```



- Los estudiantes pueden inscribir las materias que desean cursar. Las inscripciones se pueden eliminar pero no modificar.

GRANT INSERT, DELETE ON Inscripciones TO Estudiante;

- El decano puede consultar una síntesis de las inscripciones: sigla, grupo, inscritos y disponibles. Esta consulta únicamente está disponible el primer día de la semana de 8:00 a 5:00.

Vista:

```
CREATE VIEW Sintesis_Insp AS (  
SELECT sigla, grupo, inscritos, (capacidad - inscritos) AS 'disponibles'  
FROM Materias, Grupos WHERE Materias.sigla = Grupos.materia AND  
to_char(sysdate,'day') = 'Lunes' AND ( 8:00 <= to_char(sysdate,'HH24:mi') and  
to_char(sysdate,'HH24:mi') <= 17:00))
```

```
GRANT SELECT ON Sintesis_Insp TO Decano;
```

- Todos los actores pueden consultar los datos de las materias.

```
GRANT SELECT ALL ON Materias TO Decano, Coordinador, Estudiante;
```