**ESCUELA COLOMBIANA DE INGENIERIA**

**MODELOS Y BASES DE DATOS**

**TRANSACCIONES Y SEGURIDAD**

**S13-S14: 2020-1**

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

|  |  |
| --- | --- |
| 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); | CREATE TABLE ESTUDIANTES(  codigo NUMBER(7),  nombres VARCHAR(50));  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); |

**C**REATE 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\_APLICATION\_ERROR(-20003, ‘Cupos Alcanzados)

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.
   * No existan ni la materia ni el número, se encuentra que la materia tiene longitud 2 y xMateria no presenta longitud por lo cual seria 10 y esto permitiría una longitud mayor, al igual que con el numero que es longitud 2 y xNumero es longitud 10 por lo cual ocurriría también un error de estos
   * En el insert ocurre lo mismo anterior con xMateria y xNumero, pero esta vez también con xEstudiante que tiene longitud 10(por defecto) y Estudiante tiene longitud 7 por lo cual se intentarían inserciones erroneas
   * En el Update ocurre lo mismo que ocurría en el insert y en el select ya que se intentaría actualizaciones de tipos diferentes
2. ¿Qué posible error no se ha comunicado? Modifique el código para considerarlo.

* Que xInscritos sea mayor a xCapacidad

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

BEGIN

--BEGIN TRANSACTION

IF length(xMateria) > 4 OR length(xNumero) > 2 OR length(xEstudiante) > 7

THEN RAISE\_APLICATION\_ERROR(-20002, ‘Longitud de valores No Valida’)

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;

ELSE

THEN RAISE\_APLICATION\_ERROR(-20003, ‘Cupos Alcanzados’)

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE\_APLICATION\_ERROR(-20999, SQLERRM);

END INSCRIBIR;

1. ¿Por qué inscribir debería ser una transacción? Incluya las instrucciones necesarias: COMMIT, ROLLBACK

* Debería ser una transacción ya que se esta modificando la base de datos entonces se debe asegurar que esta tenga una adecuada integridad, ya que es un sistema de inscripción de materias múltiples estudiantes estarían inscribiendo de manera simultánea muchas materias, básicamente por las propiedades que presenta ACID de una base de datos integra(Concurrencia)

# 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 y este se aumentara a capacidad 20 – 20 por lo cual el otro estudiantes obtendría el error “Cupos Alcanzados”, esto seria un correcto comportamiento ya que no se esta bloqueando la base de datos y se esta manteniendo una integridad en los datos(Los inscritos no superan el cupo establecido).

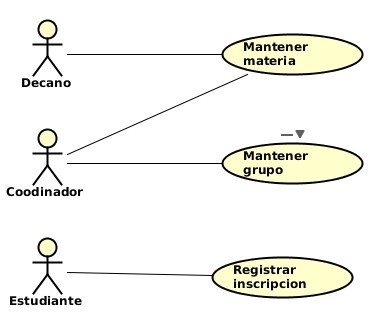
**Problemas/Soluciones**

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:
     + Si se cae el servidor este afectaría la información
     + Se cancelaria ambas transacciones
     + Se cancelaria una de las dos transacciones
   * Problema Clásico:
     + Dependencia no confirmada(Lectura Sucia)
2. Si el nivel de aislamiento de las transacciones es REPEABLE 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?

|  |  |  |
| --- | --- | --- |
| 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 t3 t4 t5 t6 t7 | 1  -  2  -  3  -  - | -  1  -  2  -  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;

**CIFRADO**

1. Si queremos mantener en secreto las inscripciones de cada uno de los cursos, ¿Qué dato, tabla o tablas se requerirían mantener cifradas para proteger esta información? Sea eficiente en la solución y justifique su respuesta.

2. Considerando que el método de cifrar de la ESCUELA es de sustitución (por el carácter siguiente en la tabla ASCII1) y el método de cifrar de la NACIONAL es de permutación (dos caracteres a la izquierda).

* ¿Qué algoritmo debe conocer cada uno de ellos? ECE, DCE, ECN, DCN
* Si el ESCUELA quiere enviar a NACIONAL el mensaje “BUSCAMOS UN PROFESOR PARA UN NUEVO GRUPO DE MINERIA” ¿Cómo se cifra el mensaje? ¿Cómo se descifra el mensaje?
* Si NACIONAL quiere contestar el mensaje con “UN EXPERTO DISPONIBLE”, ¿cómo se cifra y descifra el mensaje?

DATE “Supongamos que A y B son dos usuarios que desean comunicarse entre sí usando un esquema de cifrado de clave pública. Entonces Ay B publicarán un algoritmo de cifrado (incluyendo en cada caso la clave de cifrado correspondiente) pero por supuesto, mantendrán el algoritmo de descifrado y la clave en secreto (incluso entre sí). Hagamos que los algoritmos de cifrado sean ECA y ECB (para cifrar mensajes que serán enviados a A y B, respectivamente) y hagamos que los algoritmos de descifrado correspondientes sean DC A y DCB, respectivamente. ECA y DC A son inversos entre sí, al igual que ECB y DCB. Ahora supongamos que A desea enviar a B un fragmento de texto plano P. En lugar de calcular ECB (P) y transmitir el resultado, A aplica primero a P el algoritmo de descifrado DCA y luego cifra el resultado y lo transmite como el texto cifrado C. C = ECB ( DCA ( P ) ) Al recibir C, el usuario B aplica el algoritmo de descifrado DCB y luego el algoritmo de cifrado ECA produciendo el resultado final P: ECA ( DCB ( C ) ) = P Ahora B sabe que el mensaje en efecto proviene de A, ya que ECA producirá P sólo si el algoritmo DCA fue utilizado en el proceso de cifrado y ese algoritmo sólo lo conoce a A.

---------

ECA ( DCB ( ECB ( DCA ( P ) ) ) )

ECA ( DCA ( P) ) /\* ya que DCB y ECB se cancelan \*/

= P /\* ya que ECA y DCA se cancelan \*/”

1 El carácter que sigue a espacio es ! Y el de Z es [.