



Examen de Control -- Ampliación de Bases de datos

----- Marzo, 2023

===== SOLUCION =====

APELLIDOS:

NOMBRE:

1. (1,5 punto) Dibujando una tabla (ver el margen) con dos transacciones concurrentes T1 y T2

a) Plantea una situación de deadlock escribiendo las operaciones de lectura y escritura que necesites. **SOLUCION:** busca qué es un deadlock ----->

b) ¿Usando las mismas operaciones, qué cambiarías para que no sucediera el deadlock ?

SOLUCION: cambiar orden, que la T1 escriba X e Y antes que ninguna escritura de T2.

T1	T2
write X	
	Write Y
Write Y	
	write X

2. (5,5 puntos) Dado el esquema relacional desnormalizado, tenemos el atributo derivado **TotalHoras**, que es la suma de las dedicaciones de los proyectos en los que trabaja el investigador.

Investigador: INV(NombreI, DNII, TotalHoras)
trabaja en: TR(DNII, NumP, HorasDedicación)
proyecto: PR(NombreP, NumP, Lugar, NumD)

- a) Escribe en SQL la consulta que queremos agilizar (por la que queremos conservar el atributo derivado)

SOLUCION(si no tuviéramos TotalHoras):

```
Select NombreI DNII, sum(HorasDedicación)
from INV, TR where INV.DNII = TR.DNII
group by DNII;
```

- b) Escribe en PLSQL un trigger que se active cuando insertemos una fila en TR. Lo único que hace es llamar al procedimiento **insertoConsistente** (DNII, NumP, HorasDedicación) y gestionar cualquier excepción.

SOLUCION:

```
create or replace TRIGGER MiTri
AFTER INSERT ON TR
FOR EACH ROW
DECLARE
BEGIN
    insertoConsistente(:new.DNII, :new.NumP, :new.HorasDedicación);

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('- - Hay un ERROR - - ');
END;
```

- c) Escribe el procedimiento **insertoConsistente** (DNII, NumP, HorasDedicación) que mantenga la consistencia de los datos. Lo primero que hace es comprobar que existe el investigador en **INV** (Si no existe da un mensaje, ejecuta el comando de terminar transacción y termina el procedimiento). Después actualiza lo necesario. Y lo último es confirmar las actualizaciones mediante el comando de terminar transacción.

SOLUCION:

```
Create or replace procedure insertoConsistente (DNIIx INT, NumPx INT, HorasDedicaciónx
INT) as
siesta INT := 0;
BEGIN
    --- Existe el investigador ?
select count(DNII) into siesta from INV where DNII = DNIIx;
IF siesta = 0 Then
    dbms_output.put_line('ERROR: No está el investigador: ' || DNIIx);
    commit; -- también: rollback
ELSE
    --- Actualizo TotalHoras en INV
    update inv
    set totalHoras := totalHoras + HorasDedicacionx
    where DNII = DNIIx;
    --- Confirmar
    commit;
END IF;
END;
```



- d) En un supuesto diferente, en el que en vez de una tabla `INV(NombreI, DNII, TotalHoras)`, tuviéramos una tabla por cada investigador `INV_XXXXX(NumP, TotalHoras)`, donde XXXX es el DNII del investigador. Se pide que escribas solo las instrucciones que cambiarían en el `insertoConsistente` para que siga funcionando correctamente.

SOLUCION:

```
NombreTabla VARCHAR (13) := 'INV_' || DNIIx;

--- ver si existe el INV.
select count(*) into siesta from tabs
      where table_name like NombreTabla;

----- si esta = 0
EXECUTE IMMEDIATE
'BEGIN UPDATE ' || NombreTabla || '
  set totalHoras := totalHoras + HorasDedicacionx
  where DNII = DNIIx END;';
```

3. (3 puntos) Aplicando las definiciones de FN, para cada relación: 1.- Indica cuales son las claves candidatas. 2.- Indica cual es la mejor FN en la que está la relación. 3.- Indica qué condición cumple para estar en dicha FN y cual no cumple para estar en la siguiente mejor FN.

- a. Dada la relación `AA(A1,A2,A3,A4,A5)` con $DFs = \{A1 A2 \rightarrow A4 A5, A2 \rightarrow A3, A1 A2 \rightarrow A3\}$

SOLUCION: la PK es `A1A2`: Está en 1ªFN no atrib multivalorados,
no está en 2ªFN porque hay una DF parcial (explicar lo que es) de la PK $A2 \rightarrow A3$

- b. Dada la relación `BB(B1,B2,B3,B4,B5)` con $DFs = \{B1 B2 \rightarrow B3, B5 \rightarrow B4, B1 B2 \rightarrow B5\}$

SOLUCION: la PK es `B1B2`: Está en 2ªFN no DFs parciales de clave,
No está en 3ªFN porque hay una DF transitiva (explicar lo que es) $B5 \rightarrow B4$

- c. Dada la relación `CC(C1,C2,C4,C5)` con las $DFs = \{C1 C2 \rightarrow C4, C1 C2 \rightarrow C5\}$

SOLUCION: la PK es `C1C2`: Está en FNBC, cumple que las DFs: sus lados izq son superclaves
y no existe una DF $X \rightarrow Y$ donde Y pertenezca a un CC