

Apellidos, Nombre:.....

Contesta las preguntas del examen con claridad y brevedad.

1. a) 0,2 puntos b) 0,4 puntos

Sea el siguiente esquema relacional en el que todas las restricciones están definidas como NOT DEFERRABLE INITIALLY IMMEDIATE (valor por defecto):

```
CREATE TABLE EQUIPO (
  COD_EQ VARCHAR2(5 BYTE) NOT NULL CONSTRAINT EQUIPO_PK PRIMARY KEY,
  NOM_EQ VARCHAR2(25 BYTE) NOT NULL,
  PRESUPUESTO NUMBER NOT NULL CONSTRAINT EQUIPO_CHK1 CHECK (presupuesto > 0),
  DORSAL_CAPITÁN SMALLINT NOT NULL
  CONSTRAINT EQUIPO_FK1 REFERENCES JUGADOR(DORSAL));
```

```
CREATE TABLE JUGADOR (
  DORSAL SMALLINT NOT NULL CONSTRAINT JUGADOR_PK PRIMARY KEY,
  NOM_JU VARCHAR2(25) NOT NULL,
  EDAD SMALLINT,
  EQUIPO VARCHAR2(5) NOT NULL
  CONSTRAINT JUGADOR_FK1 REFERENCES EQUIPO(COD_EQ));
```

a. Sin cambiar el modo, indica qué restricciones deberían ser diferibles para que se pueda introducir datos teniendo en cuenta que se quieren tener las menos restricciones diferibles posibles.

SOLUCIÓN 1:

Cambiar la restricción EQUIPO_FK1 a DEFERRABLE INITIALLY IMMEDIATE

SOLUCIÓN 2:

Cambiar la restricción JUGADOR_FK1 a DEFERRABLE INITIALLY IMMEDIATE

b. Teniendo en cuenta el cambio propuesto en el apartado anterior y suponiendo que la base de datos está vacía, escribe la transacción que permitiría insertar la siguiente información en la base de datos. Las instrucciones deben ser instrucciones SQL.

Equipo:	cod_eq=VCF	Jugador:	dorsal=1	Jugador:	dorsal=2
	nom_eq=Valencia		nom_ju=Juan		nom_ju=Mateo
	Presupuesto=99000000		Edad=23		equipo=VCF
	dorsal_capitán=1		equipo=VCF		

SOLUCIÓN 1:

```
SET CONSTRAINT Equipo_FK1 DEFERRED;
INSERT INTO Equipo VALUES ('VCF', 'Valencia', 99000000, 1);
INSERT INTO Jugador VALUES (1, 'Juan', 23, 'VCF');
INSERT INTO Jugador VALUES (12, 'Mateo', NULL, 'VCF');
COMMIT;
```

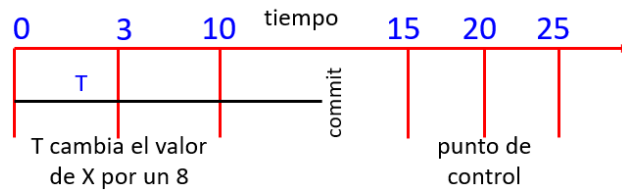
SOLUCIÓN 2:

```
SET CONSTRAINT Jugador_FK1 DEFERRED;
INSERT INTO Jugador VALUES (1, 'Juan', 23, 'VCF');
INSERT INTO Jugador VALUES (12, 'Mateo', NULL, 'VCF');
INSERT INTO Equipo VALUES ('VCF', 'Valencia', 99000000, 1);
COMMIT;
```

2. a) 0,3 puntos b) 0,4 puntos

a. Sean los siguientes supuestos:

- El diagrama de abajo muestra la ejecución confirmada de T .
- X es un elemento de datos que en el instante 0 tiene el valor 7.
- T es una transacción que en el instante 3 modifica el valor de X poniendo un 8.



Para cada una de las cuatro estrategias posibles de actualización indica qué valor o valores podría tener X en disco en los instantes 10, 15 y 25.

Estrategia	Instante 10	Instante 15	Instante 25
Inmediata-No Forzar	$X=7$ o $X=8$	$X=7$ o $X=8$	$X=8$
Inmediata-Forzar	$X=7$ o $X=8$	$X=8$	$X=8$
Diferida-No Forzar	$X=7$	$X=7$ o $X=8$	$X=8$
Diferida-Forzar	$X=7$	$X=8$	$X=8$

b. En un entorno monousuario, explica cómo se recupera el sistema en cada una de las cuatro estrategias de actualización cuando una transacción es anulada por el usuario mediante un *rollback*.

Dependiendo de la estrategia de actualización de la BD utilizada en el SGBD, anular una transacción implica tareas de recuperación distintas.

- Si la estrategia es diferida, se tiene la seguridad de que los bloques actualizados por la transacción anulada no han sido transferidos todavía a disco, por lo tanto sólo habrá que anular las actualizaciones, en los buffers de memoria principal.
- Si la estrategia es inmediata, los bloques actualizados por la transacción anulada pueden haber sido transferidos ya a disco, por lo tanto habrá que anular las actualizaciones, tanto en disco como en los buffers de memoria principal.
- Para anular las actualizaciones, tanto en los buffers de memoria principal como en el disco se usará el valor antes las entradas ([escribir, T , X , valor_antes, valor_despues]) que haya en el diario donde T es la transacción anulada.

El que la estrategia sea forzar/no forzar es irrelevante en este caso ya que la transacción no llega a la confirmación.

3. a) 0,1 puntos b) 0,3 puntos

a) Supón un entorno concurrente controlado por el protocolo Multiversión en el que hay tres versiones del elemento x (sean x_0 , x_1 y x_2), cuyas marcas de tiempo son las que se indican más abajo y que $t_i < t_j$ si $i < j$:

	MT_L	MT_E
x_0	t_1	t_1
x_1	t_7	t_4
x_2	t_{10}	t_{10}

- Sea la transacción $T1: \{r(x), w(x), c\}$, ¿qué condición o condiciones tendría que cumplir la marca de tiempo de $T1$ para que, en la operación de lectura, la versión que le proporcione el sistema sea la versión x_1 ?

$$t_4 \leq MT(T1) < t_{10}$$

- Sea la transacción $T2: \{r(y), w(x), c\}$, cuya marca de tiempo es t_6 ¿puede $T2$ realizar la operación de escritura? ¿qué condición o condiciones se evalúan para permitir su ejecución? ¿qué controlan estas condiciones?

NO se puede realizar la escritura.

$MT_E(x_1) \leq MT(T2)$, siendo x_1 la versión con la marca de tiempo mayor entre todas las versiones con marca de tiempo menor o igual a la marca de tiempo de $T2$,

Y $MT_L(x_1) \leq MT(T2)$

Controlan que la versión más joven anterior al inicio de $T2$ (x_1) no ha sido leída por una transacción posterior a $T2$ ($MT(T2)=t_6$), lo que en este caso no es cierto, ya que ha sido leída por una transacción con $MT = t_7$

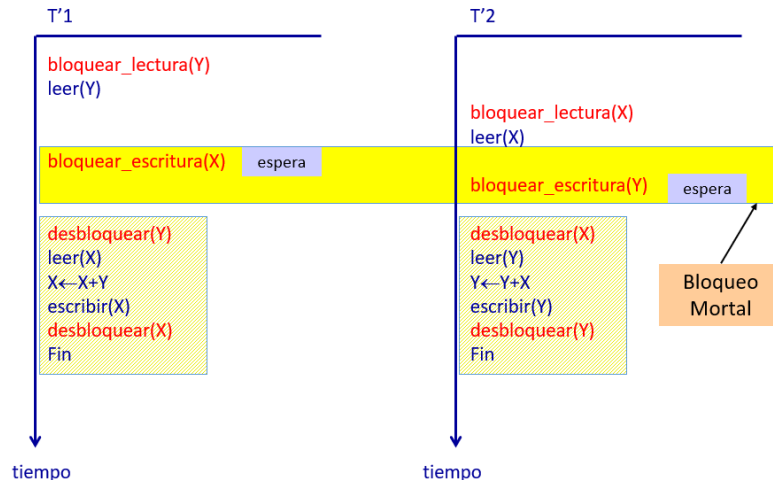
4. a) 0,2 puntos b) 0,3 puntos c) 0,3 puntos

a. ¿Qué significa la regla de las dos fases en los protocolos de bloqueos?

Regla de las dos fases (2F): En una transacción todas las operaciones de bloqueo (bloquear_lectura, bloquear_escritura o promoción_bloqueo) preceden a la primera operación de desbloqueo (desbloquear) de la transacción.

b. Explica con un ejemplo el problema del bloqueo mortal en los protocolos de bloqueos.

Bloqueo mortal: cada transacción T_i en un conjunto de dos o más transacciones está esperando un elemento de datos que está bloqueado por otra transacción T_j de dicho conjunto.



c. Explica cómo se puede detectar un bloqueo mortal en los protocolos de bloqueos. Aplícalo al ejemplo del apartado anterior.

El bloqueo mortal se puede detectar construyendo el grafo de espera de las transacciones, si el grafo tiene un ciclo, hay un bloqueo mortal:

Grafo de espera: grafo dirigido $G=(N,A)$, que consiste en un conjunto de nodos N y un conjunto de arcos A .

- ✓ El grafo contiene un nodo por cada transacción que se está ejecutando actualmente.
- ✓ El arco $a_{ij} = (T_i \rightarrow T_j)$ se crea si la transacción T_i está esperando bloquear un elemento X que está bloqueado por la transacción T_j (uno de los dos bloqueos es exclusivo). Cuando T_j libera el elemento X se borra el arco.

En el ejemplo anterior se puede ver el ciclo que detecta el bloqueo mortal:

