

Temps: **3 hores**

Notes: **2 Juliol tarda      Revisió: 3 Juliol tarda**

Cada pregunta en un full separat, excepte la primera que cal entregar en dos fulls.

**1 (2 pt)** Considereu l'esquema de la base de dades següent:

```
Create table professors
(dni char(50),
nomProf char(50) unique,
telefon char(15),
primary key (dni));

create table despatxos
(modul char(5),
numero char(5),
superficie integer not null check(superficie>12),
primary key (modul,numero));

create table assignacions
(dni char(50),
modul char(5),
numero char(5),
instantInici integer,
instantFi integer,
primary key (dni, modul, numero, instantInici),
foreign key (dni) references professors,
foreign key (modul,numero) references despatxos);
```

Adicionalment, hi ha definida la restricció que un professor només pot tenir una assignació vigent, és a dir, una assignació a instantFi igual Null

**a)** Doneu una sentència SQL per obtenir els professors que no tenen cap assignació als despatxos més grans (superfície màxima) del modul 'c6'. Es vol el nom dels professors.

**b)** Donades les sentències de consulta següents, digueu si poden tornar files repetides i el màxim i mínim nombre de files que poden tornar tot raonant la vostra resposta. Useu A, P i D per al nombre de files de la taula d'assignacions, professors i despatxos. Suposeu que  $A > P > D > 0$ .

**b1)** SELECT d.modul, d.numero  
FROM assignacions a, despatxos d  
WHERE a.modul = d.modul AND a.numero = d.numero AND a.dni = '52112345'

**b2)** SELECT DISTINCT a.modul, a.numero  
FROM assignacions a

**b3)** SELECT COUNT(\*)

FROM despatxos d, assignacions a

WHERE d.modul = a.modul AND d.numero = a.numero AND  
d.superficie <= 15 and a.instantFi IS NULL

GROUP BY a.dni

**b4)** SELECT p1.nomProf

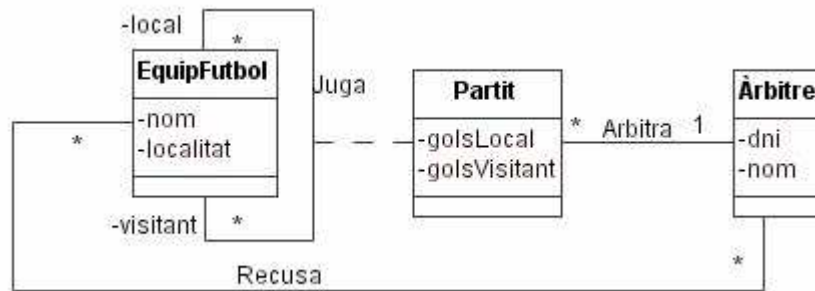
FROM professors p1, professors p2

WHERE p1.nomProf = p2.nomProf

**b5)** Repetiu b2) suposant que modul i numero a la taula assignacions no són clau forana que referencia despatxos.

**b6)** Repetiu b4) suposant que no hi ha restriccions UNIQUE.

**2 (2.5 pt)** Considereu el disseny conceptual en UML següent:



La clau externa de la classe *EquipFutbol* és el *nom* i la d'*Àrbitre* és el *dni*.

Considereu també les restriccions següents:

1- Un àrbitre no pot arbitrar un partit on hi juga un equip que l'ha recusat

2- No hi pot haver dos (o més) partits que tinguin el mateix equip local i el mateix equip visitant (però s'ha de permetre que hi pugui haver dos partits amb els mateixos equips i els rols intercanviats).

3- Si no hi ha cap partit, no es poden esborrar recusacions.

**a)** Transformeu el disseny conceptual anterior al model relacional.

**b)** Expliqueu com s'implementarien en l'SQL d'Informix les restriccions 1, 2 i 3 sobre les taules obtingudes. Si alguna de les restriccions s'ha d'implementar amb triggers, indiqueu per cada trigger: taula, esdeveniment activador (INSERT, DELETE, UPDATE [OF <column>]) i quan s'hauria d'executar (BEFORE, AFTER, FOR EACH ROW). Obsevació: Supposeu que tots els possibles esdeveniments estan autoritzats sobre totes les taules.

**c)** Si a l'apartat b) heu proposat usar algun trigger amb un esdeveniment activador INSERT, doneu la seva implementació. Observació: Cal implementar un sol trigger. Si teniu més d'un trigger amb INSERT, escolliu-ne un qualsevol.

**3 (2 pt)** Suposem que tenim la transacció **T** següent que fa consultes sobre la relació d'empleats. L'esquema de la relació d'empleats és: Empleats(idEmpleat, nomEmpleat, sou, numDept).

**T**

Q1: SELECT avg(sou) FROM Empleats;  
<operacions read-only>  
Q2: SELECT avg(sou) FROM Empleats;  
Commit;

Es demana:

- Suposem que totes les altres transaccions del sistema treballen a nivell SERIALITZABLE i que són read-only. **Explicar breument** quin és el **mínim** nivell d'aïllament al que hauria de treballar la transacció T, per evitar qualsevol interferència.
- Suposem que totes les altres transaccions del sistema treballen a nivell SERIALITZABLE i que només fan queries, updates i deletes. **Explicar breument** quin és el **mínim** nivell d'aïllament al que hauria de treballar la transacció T, per evitar qualsevol interferència.
- Suposem que totes les altres transaccions del sistema treballen a nivell SERIALITZABLE i que poden fer qualsevol operació. **Explicar breument** quin és el **mínim** nivell d'aïllament al que hauria de treballar la transacció T, per evitar qualsevol interferència.

#### **4 (2 pt)**

Tenim una relació R amb 200,000 tuples. Sobre un atribut x hem construït un índex (no agrupat) que és un arbre B+ d'ordre 50.

**Apartat 1.** Digueu l'alçada mínima i màxima que pot tenir aquest arbre B+. Justifiqueu la resposta.

**Apartat 2.** Fem ara les següents suposicions:

- A cada pàgina de dades hi caben 10 tuples com a màxim.
- Les pàgines, tant de l'índex com de tuples, són plenes al 70% de la seva capacitat màxima.

Volem obtenir totes les tuples que compleixen  $a < R.x < b$  i dubtem entre els dos mètodes següents:

- Recorregut seqüencial de totes les dades, descartant les tuples que no satisfan la condició.
- Accés seqüencial per valor usant l'índex.

Estimeu el nombre màxim de pàgines d'índex i de dades llegides per cada mètode. Per al segon mètode, aquest nombre dependrà del nombre de tuples que satisfan la condició, diguem-ne N.

Finalment, digueu quin mètode triaríeu en funció de N.

#### **5 (1.5 pt)**

a) (0.5 pts.) Som creadors d'una BD amb dues relacions: Departaments(dept,nom\_dept) i Treballadors(id,nom,dept,salari) on {dept} referencia Departaments i a més és not null; i una vista, SalariMitja(dept,salari\_m), que conté, per cadascun dels 15 departaments de l'empresa, el salari mitjà dels treballadors d'aquest departament. Supposeu que donem els següents privilegis a l'usuari Pere

```
GRANT SELECT ON SalariMitja TO Pere
GRANT SELECT(dept,nom_dept) ON Departaments TO Pere
GRANT SELECT(id,nom) ON Treballadors TO Pere
GRANT UPDATE(id,nom,dept) ON Treballadors TO Pere
```

Amb aquests privilegis en Pere pot esbrinar el salari del treballador 43543434. Doneu **una seqüència d'instruccions en SQL** que mostri com ho pot fer.

b) (1 pt.) Considereu el següent esquema:

```
DIARIS(nom_diari,ambit)
PERIODISTES(id,nom,telf,agencia)
ARTICLES(diari,id,data,titol,tema)
    {diari} referencia DIARIS
    {id} referencia PERIODISTES
```

Doneu una seqüència d'operacions en àlgebra relacional per obtenir les agències tals que cap dels seus periodistes associats hagi publicat més d'un article en un mateix diari i un mateix dia.