

Temps: 3 h**Notes 26 gener tarda Revisió: 27 gener tarda****Cada pregunta en un full separat****1) (2 punts) Considereu l'esquema de la base de dades següent:**

```
create table especies
(especie char(20) primary key,
comestible char(1) not null check ((comestible='S') or (comestible='N')),
valor integer not null check (valor >= 0 and valor <= 10),
check ((comestible = 'S') or (valor = 0)) );
-- En aquesta taula hi ha una fila per cada especie de bolets.

create table boletaires
(nom char(20) primary key,
numLlicencia integer unique,
ciutat char(20),
edat integer check (edat >= 18) );
-- En aquesta taula hi ha una fila per cada caçador de bolets

create table troballes
(nom char(20),
especie char(20),
jornada integer,
quantitat integer check (quantitat > 0),
lloc char(20),
primary key (nom, especie, jornada),
foreign key (nom) references boletaires,
foreign key (especie) references especies);
-- En aquesta taula hi ha una fila per cada espècie de bolets que troba un
boletaire en una jornada.
```

- 1.1)** Escriviu una sentència SQL per obtenir el número de llicència dels boletaires que han trobat més de 15 espècies en una mateixa jornada.
- 1.2)** Escriviu una sentència SQL, en la que no s'utilitzin funcions d'agregació, per obtenir les ciutats on viuen boletaires que tenen troballes de totes les espècies emmagatzemades a la base de dades.
- 1.3)** Suposem que ara algú ha creat una nova taula a la base de dades amb l'esquema següent:

```
create table classificacio
(nomBoletaire char(20),
quantitatTotal integer not null check(quantitatTotal >=0),
primary key (nomBoletaire),
foreign key (nomBoletaire) references boletaires);
```

Doneu una (o en cas que no ho trobeu possible, més d'una) sentència d'inserció de files a la taula *Classificació* que l'ompli a partir del contingut de la resta de taules de la base de dades. Tingueu en compte el següent: Hi haurà una fila de la taula per cada boletaire de la base de dades. El valor de l'atribut quantitat total serà la suma de les quantitats de bolets trobades pel boletaire en totes les jornades. En cas que un boletaire de la base de dades no tingui troballes de cap espècie, el valor de la quantitat total ha de ser zero.

2) (2 punts)

Suposem que tenim la transacció **T** següent que fa dues 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;

Q2: SELECT avg(sou) FROM Empleats;

Commit;

Es demana:

- a) 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.
- b) Suposem que totes les altres transaccions del sistema treballen a nivell SERIALITZABLE i que només fan consultes, updates i deletes. **Explicar breument** quin és el **mínim** nivell d'aïllament al que hauria de treballar la transacció T, per evitar qualsevol interferència.
- c) 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.
- d) Ara considereu la següent variació, on les dues consultes estan a dues diferents transaccions:

T1: (Q1) SELECT AVG(sou) FROM Empleats;
Commit;

T2: (Q2) SELECT AVG(sou) FROM Empleats;
Commit;

Suposeu que les dues transaccions T1 i T2 treballen a nivell READ COMMITTED. Considereu els escenaris descrits a), b) i c) per les altres transaccions. En quin d'aquests escenaris, si n'hi ha cap, s'obtindrà sempre el mateix resultat per Q1 i Q2.

- e) Continuant amb la variació de les dues transaccions de l'apartat d), suposeu ara que T1 i T2 treballen a nivell SERIALITZABLE. En quin d'aquells escenaris, si n'hi ha cap, s'obtindrà sempre el mateix resultat per Q1 i Q2.

3) (2 punts)

Una empresa gran disposa de la taula Empleats(idEmpleat, nomEmpleat, ciutat, ...). S'ha creat un índex agrupat per idEmpleat i 2 índexs no agrupats, un per nomEmpleat i l'altre per ciutat.

Sabem que la taula d'empleats té aproximadament 100.000 empleats, que els idEmpleats són nombres entre 1 i 100.000 repartits més o menys uniformement, que l'empresa té uns 1.000 empleats a Barcelona. A més, se sap que els índexs són arbres B+ d'ordre 50, que les pàgines de l'índex són al 80% de la seva capacitat en mitjana, i a una pàgina de dades hi ha unes 5 files en mitjana.

Donada la consulta

```
SELECT * FROM Empleats e
WHERE e.idEmpleat > 20000 AND e.ciutat = "Barcelona"
```

estimeu (i justifiqueu breument) el nombre de pàgines (d'índex i de dades) que es llegiran si la consulta es resol:

- a) sense usar els índexs
- b) usant accés seqüencial per valor usant l'índex per idEmpleat
- c) usant accés seqüencial per valor usant l'índex per ciutat

4) (2 punts)

a) Considereu la taula Empleats(id, categoria, sou) i l'assertió següent:

```
CREATE ASSERTION Limitinterns CHECK (  
    NOT EXISTS ( SELECT *  
                  FROM Empleats  
                  WHERE categoria = 'intern'  
                    and sou > (SELECT AVG(sou) FROM Empleats)));
```

Es demana quines sentències d'actualització (inserció, esborrat i modificació) poden violar aquesta restricció. En el cas de la sentència de modificació digueu quin/s és/són els atributs rellevants. Per cadascuna d'aquestes sentències, doneu un extensió de la taula *Empleats* i un exemple de la sentència que produeixi la violació.

b) Considereu la taula Empleats(id, categoria, sou) anterior i les taules següents:

```
Tipusprojectes(tipus)  
Projectes(tipus, projecte, pressupost)  
    on {tipus} referencia Tipusprojectes  
Responsables(id, tipus, projecte, datainici, datafi)  
    on {tipus, projecte} referencia Projectes,  
    on {id} referencia Empleats  
Especialitzacions(id, tipus)  
    on {tipus} referencia Tipusprojectes,  
    on {id} referencia Empleats
```

Volem garantir que tot empleat assignat com a responsable de projecte, sigui especialista en aquest tipus de projecte. Un estudiant de BD ens ha fet el disparador següent:

```
CREATE OR REPLACE FUNCTION comprova_restriccio RETURNS Trigger AS $$  
BEGIN  
    if ( NOT EXISTS (SELECT * FROM Especialitzacions  
                     WHERE id=NEW.id AND tipus=NEW.tipus)) then  
        RAISE EXCEPTION 'L'empleat ha de ser especialista en el tipus de projecte';  
    end if;  
    RETURN new;  
END;  
$$language plpgsql;
```

```
CREATE TRIGGER Restriccio BEFORE INSERT on Responsables  
FOR EACH ROW execute procedure comprova_restriccio();
```

Es demana:

b.1) Aquest disparador cobreix totes les sentències d'actualització (inserció, esborrat i modificació) i sobre totes les taules que poden violar la restricció que volem garantir? En cas afirmatiu expliqueu perquè no cal cap altre disparador més, en cas negatiu digueu sobre quines taules i esdeveniments s'haurien de definir els disparadors que falten.

b.2) Es podria implementar aquesta restricció directament amb restriccions de columna i/o de taula? En cas afirmatiu digueu com ho faríeu, en cas negatiu digueu perquè no es podria fer.

5) (2 punts)

a) Considereu les relacions $R(A,B)$ i $S(A,B)$. Es vol obtenir la intersecció de R i S , però només disposem de 5 operadors relacionals: selecció, projecció, producte cartesià, natural join i reanomenament. És possible calcular la intersecció de R i S fent servir només aquests operadors? En cas afirmatiu, doneu l'expressió més simple equivalent a la intersecció. En cas negatiu, raoneu la resposta.

b) Per a cadascuna de les polítiques d'actuació en cas d'esborrat d'una clau primària amb claus foranes que hi fan referència, doneu els seus noms (utilitzant notació SQL estàndard) i una breu explicació del seu funcionament.

c) Considerant la taula i la inserció següents:

```
CREATE TABLE Empleats (
```

```
  Id INTEGER,
```

```
  Nom VARCHAR(50),
```

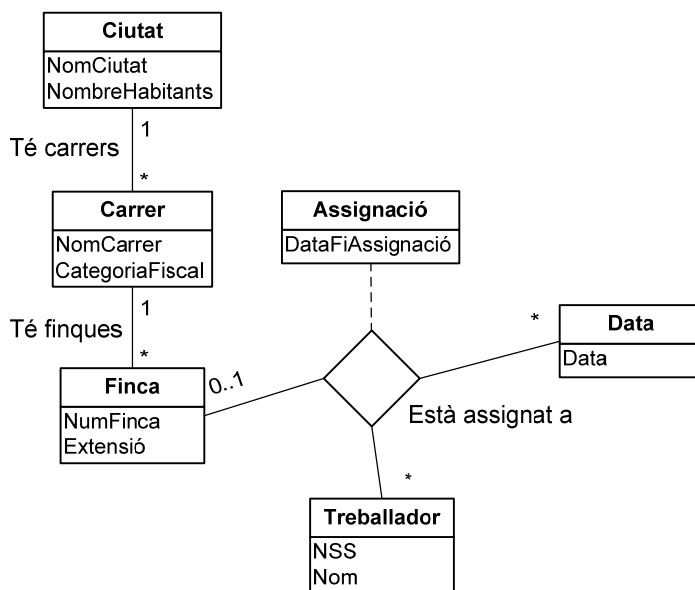
```
  Categoria VARCHAR(20),
```

```
  Sou FLOAT NOT NULL);
```

```
INSERT INTO Empleats (102, 'Pere', NULL, 1500);
```

Doneu la capçalera i el contingut de cadascun dels camps de la fila inserida. I doneu també la longitud total de la fila.

d) Considerant el disseny conceptual en UML següent:



Les claus externes i restriccions textuais del disseny anterior són les següents:

Clau externa Ciutat: NomCiutat

Clau externa Carrer: NomCarrer

Clau externa Finca: NumFinca

Clau externa Data: Data

Clau externa Treballador: NSS

Restricció1: Dins d'una mateixa ciutat, no poden haver carrers amb el mateix NomCarrer, però en ciutats diferents sí.

Restricció2: Dins d'un mateix carrer, no poden haver finques amb el mateix NumFinca, però en carrers diferents sí.

Transformeu el disseny conceptual anterior al model relacional.

