

Temps: 2 hores i 45 minuts**Notes 24 gener tarda Revisió: 25 gener tarda****Cada pregunta en un full separat****1) (2 punts)** Considereu l'esquema de la base de dades següent:

```
CREATE TABLE clients
(dni char(9),
nomClient char(50) UNIQUE NOT NULL,
ciutatResidencia char(15),
PRIMARY KEY (dni));
-- Hi ha una fila per cada client d'una entitat bancària.

CREATE TABLE comptesBancariss
(numCompte char(16),
dniClient char(9) NOT NULL,
saldoDisponible real,
PRIMARY KEY (numCompte),
FOREIGN KEY (dniClient) REFERENCES clients(dni));
-- Hi ha una fila per cada compte bancari d'una entitat. El saldoDisponible és el saldo
del compte bancari un cop aplicats els ingressos i reintegraments.

CREATE TABLE ingressos
(numCompte char(16),
instantIngres integer,
quantitat integer NOT NULL CHECK (quantitat>0),
PRIMARY KEY (numCompte, instantIngres),
FOREIGN KEY (numCompte) REFERENCES comptesBancariss(numCompte));
-- Hi ha una fila per cada ingr  s fet al compte bancari.

CREATE TABLE reintegraments
(numCompte char(16),
instantReintegrament integer,
quantitat integer NOT NULL CHECK (quantitat>0),
PRIMARY KEY (numCompte, instantReintegrament),
FOREIGN KEY (numCompte) REFERENCES comptesBancariss(numCompte));
-- Hi ha una fila per cada reintegrament fet al compte bancari.
```

1.1) Escriviu una sent  ncia SQL per obtenir el nom dels clients que tenen un o m  s comptes amb saldo disponible negatiu i cap ingr  s.

1.2) Escriviu una sent  ncia SQL per obtenir el dni i el nom dels clients que tenen algun compte bancari en el que s'han ingressat m  s de 50000 euros entre l'instant 1000 i el 2000, i en el que no s'ha fet cap reintegrament en el mateix per  ode.

1.3) Es vol obtenir els saldos dels comptes bancaris a l'instant 1000. Concretament es vol per cada n  mero de compte, el saldo disponible a l'instant 1000. Per fer-ho alg   ha implementat aquesta vista:

```
CREATE VIEW saldoss1000p (num,saldo) AS
SELECT cb.numCompte, cb.saldoDisponible -(sum(i.quantitat) - sum(r.quantitat))
FROM comptesBancariss cb, ingressos i, reintegraments r
WHERE cb.numCompte=i.numCompte and i.instantIngres>1000 and
      cb.numCompte = r.numCompte and r.instantReintegrament>1000
GROUP BY cb.numCompte, cb.saldoDisponible
```

Doneu una extensi   de les taules de la base de dades i de la vista que demostrin que la implementaci      incorrecta. Raoneu la resposta.

2) (2 punts) Considereu l'esquema de la base de dades següent:

```
CREATE TABLE Departaments (  
    num_dpt int primary key,  
    pressupost int not null check (pressupost>0));  
  
CREATE TABLE Empleats (  
    num_empl int primary key,  
    sou int not null check (sou>=0),  
    num_empl_cap int references Empleats,  
    num_dpt int not null references Departaments);  
  
CREATE or REPLACE FUNCTION pr_primer() RETURNS trigger AS $$  
BEGIN  
    UPDATE Empleats SET sou=3000 WHERE sou=new.pressupost;  
    RETURN null;  
END;  
$$LANGUAGE plpgsql;  
  
CREATE TRIGGER primer  
AFTER INSERT on Departaments  
FOR EACH ROW EXECUTE PROCEDURE pr_primer();  
  
CREATE or REPLACE FUNCTION pr_segona() RETURNS trigger AS $$  
BEGIN  
    UPDATE departaments SET pressupost=pressupost+500;  
    RETURN null;  
END;  
$$LANGUAGE plpgsql;  
  
CREATE TRIGGER segona  
AFTER UPDATE on Empleats  
FOR EACH ROW EXECUTE PROCEDURE pr_segona();
```

Es demana:

- a.** Supposeu que el contingut inicial de la base de dades és el següent:

Departaments(num_dpt,pressupost)	(1,3000)		
Empleats(num_empl, sou, num_empl_cap, num_dpt)	(1,1000,null,1)	(2,2000,1,1)	(3,2000,1,1)

Digueu quin és el contingut final de la base de dades, després de l'execució de la sentència SQL: *INSERT INTO Departaments VALUES (2,2000)*. Justifiqueu breument la resposta, explicant les accions que segueix el SGBD a conseqüència d'aquesta inserció.

b. Repetiu l'apartat a) suposant que se substitueix la sentència SQL dins el procediment *pr_segona* per: *UPDATE departaments SET pressupost=pressupost-1000*. Agafeu com a contingut inicial, de la base de dades, el contingut inicial de l'apartat a).

c. Definiu una asserció en SQL Standard per garantir el compliment de la restricció següent: "Tot empleat que té un cap, ha de tenir un cap que pertany al departament de l'empleat".

d. Expliqueu com implementaríeu l'asserció anterior en PostgreSQL mitjançant disparadors i mitjançant procediments emmagatzemats:

d.1. Per cada disparador cal que indiqueu: l'esdeveniment activador, la taula i el tipus de disparador. En cas de l'esdeveniment UPDATE cal també les columnes rellevants. Per cada disparador, cal també que justifiqueu breument el tipus de disparador escollit.

d.2. Pels procediments emmagatzemats, expliqueu breument la solució proposada.

d.3. Expliqueu un possible avantatge d'implementació de l'asserció mitjançant disparadors, respecte a la seva implementació mitjançant procediments emmagatzemats.

3) (2,5 punts)

a. Sigui un SGBD sense cap mecanisme de control de concurrència, i suposem que es produeix l'horari següent (R= Read, RU= Read for Update, W= Write; les accions s'han numerat per facilitar fer-hi referència):

Acc#	T1	T2	T3	T4
10			R(E)	
20	RU(A)			
30	W(A)			
40			RU(F)	
50			W(F)	
60		RU(E)		
70				R(A)
80				RU(F)
90				W(F)
100		W(E)		
110		R(B)		
120	R(B)			
130		R(C)		
140				COMMIT
150			R(E)	
160	RU(A)			
170	W(A)			
180			COMMIT	
190		COMMIT		
200	COMMIT			

Contesteu, **argumentant les respostes**, a les preguntes següents:

a.1. Quin és el graf de precedències associat a l'horari donat?

a.2. Quines interferències es produeixen? per cada interferència cal que doneu: nom de la interferència, transaccions i grànuls implicats.

a.3. Quins horaris serials donen resultats equivalents a l'horari proposat?

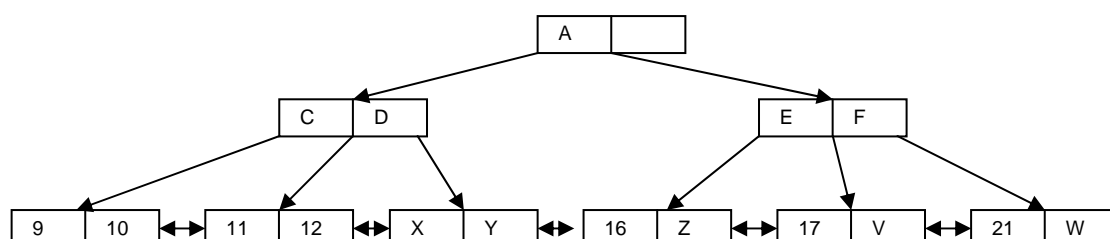
a.4. L'horari proposat, és recuperable? En cas afirmatiu doneu la definició de quan un horari és recuperable. En cas negatiu indiqueu alguna de les transaccions i operacions de l'horari que mostrin que no ho és.

b. Supposeu ara que tenim un mecanisme de control de concurrència basat en reserves S, X i que les transaccions T1, T3 i T4 treballen a un nivell d'aïllament de READ COMMITTED i que T2 treballa amb un nivell d'aïllament de REPEATABLE READ. Contesteu a les preguntes següents:

b.1. Com quedaria l'horari? L'horari ha d'incloure, a més de les peticions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves i l'ordre d'execució de totes aquestes peticions.

b.2. Quins horaris serials hi són equivalents?

4) (1,5 punts) Donat l'arbre B+ d'ordre 1 (**d=1**) que correspon a un índex definit sobre una taula T per a un atribut que és clau primària de la taula:



Es demana:

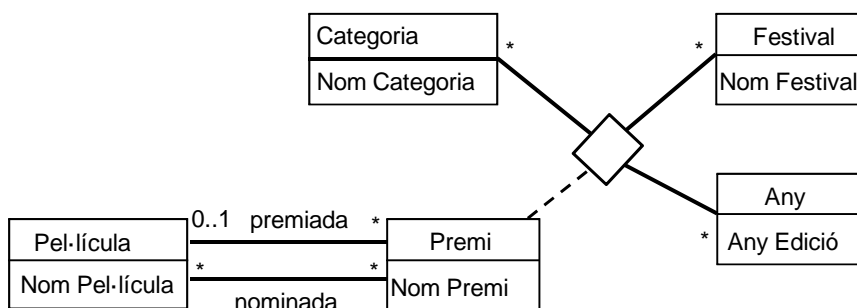
- Indiqueu quins són els valors possibles de X, Y, Z, V i W.
- Indiqueu quins són els valors possibles de C i D.
- Indiqueu quins són els valors possibles de E i F.
- Indiqueu quins són els valors possibles de A.

Justifiqueu convenientment la resposta de cada apartat.

5) (2 punts) Considereu l'esquema de la base de dades format per les taules següents:

Actor(<u>nomactor</u> , adreça, telèfon, anynaix, sexe, religió, caché)	on caché és el que cobra un actor per fer una pel·lícula
Tema(<u>nomtema</u>)	
Pel·lícula(<u>nompel</u> , nomtema)	on nomtema referencia Tema
Habilitat(<u>nomactor</u> , <u>nomtema</u>)	on nomtema referencia Tema on nomactor referencia Actor
Actuar(<u>nomactor</u> , <u>nompel</u>)	on nomactor referencia Actor on nompel referencia Pel·lícula

- Escriviu una seqüència d'operacions d'àlgebra relacional per obtenir per cadascun dels actors, els noms de totes les pel·lícules on ha actuat però que no eren d'un tema en que l'actor tenia habilitat. Concretament es demana aquesta informació en parelles: nom actor, nom pel·lícula.
- A quins atributs afecta la Llei orgànica de protecció de dades personals (LOPD), i en quin/s dels seus tres nivells classificaríeu aquests atributs?
- Digues si són certes o falses les sentències següents. Justifica la resposta.
 - Un Servidor en un Entorn SQL agrupa un conjunt de Catàlegs, que en PostgreSQL s'anomenen Esquemes.
 - L'esquema d'informació és l'esquema per defecte al que es connecta un usuari al fer la connexió amb la base de dades.
 - L'esquema de la base de dades que us hem donat és un esquema conceptual segons l'arquitectura de tres nivells ANSI/SPARC.
 - Donat un cert diagrama d'autoritzacions, l'efecte d'una operació REVOKE sempre es pot anul·lar amb una única operació GRANT.
- Suposant el model conceptual en UML següent, dona el disseny lògic que s'obté fent la traducció a model relacional.



Clau externa Categoria: Nom Categoria
Clau externa Festival: Nom Festival
Clau externa Any: Any Edició
Clau externa Pel·lícula: Nom Pel·lícula

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

```
create table professors
(dni char(9),
nomProf char(50) unique,
telefon char(15),
sou integer not null check(sou>0),
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(9),
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);
-- assignació d'un professor a un despatx.
-- instantFi te valor null quan una assignacio es encara vigent.
```

1.1 Escriviu una sentència SQL per obtenir la suma del sou dels professors que tenen alguna assignació no vigent a despatxos del mòdul 'Omaga' i que tenen 2 o més assignacions a un mateix mòdul però a despatxos diferents.

1.2 Supposeu que volem una sentència SQL per trobar els professors (dni, nomProf) que no tenen cap assignació vigent. Digueu, per cadascuna de les sentències següents **si permet obtenir els professors indicats o no**. En cas de que no, **doneu una extensió de la base de dades i mostreu** que la consulta dona un resultat incorrecte.

- a)

```
SELECT p.dni, p.nomProf
FROM professors p
WHERE p.dni NOT IN (SELECT a.dni FROM assignacions a
                    WHERE a.instantFi IS NULL);
```
- b)

```
SELECT p.dni, p.nomProf
FROM professors p
WHERE not exists (SELECT *
                  FROM assignacions a, professors p1
                  WHERE a.dni = p1.dni
                  AND a.instantFi IS NULL);
```
- c)

```
SELECT p.dni, p.nomProf
FROM professors p, assignacions a
WHERE p.dni <> a.dni AND a.instantFi IS NULL;
```
- d)

```
SELECT p.dni, p.nomProf
FROM professors p, assignacions a
WHERE p.dni = a.dni AND a.instantFi IS NULL
GROUP BY p.dni, p.nomProf
HAVING count(*) = 0;
```

1.3 Escriviu una seqüència d'operacions d'àlgebra relacional per obtenir els dni dels professors que han estat assignats a un despatx on alguna vegada ha estat assignat el professor amb dni '123'. Tingueu en compte que en el resultat de la consulta no volem que surti el professor '123'.

2. (2 punts)

2.1 Considereu les taules de la base de dades de l'exercici 1, i les vistes següents definides sobre elles:

```
CREATE VIEW personalactualOmega AS
select dni, modul, numero
from assignacions
where instantFi IS NULL and modul='Omega';
```

```
CREATE VIEW dadespersonalactualOmega (nomProf, modul, numero,
telefon) AS
select p.nomProf, pa.modul, pa.numero, p.telefon
from professors p, personalactualOmega pa
where p.dni=pa.dni;
```

a) Són actualitzables aquestes vistes segons l'estàndard SQL? Justifiqueu la resposta.

b) Considereu la consulta següent:

```
SELECT d1.nomprof, d2.nomprof
FROM dadespersonalactualOmega d1, dadespersonalactualOmega d2
WHERE d1.modul=d2.modul and
      d1.numero = d2.numero and
      d1.nomprof <> d2.nomprof;
```

b.1) Expliqueu breument què retorna la consulta.

b.2) Doneu una sentència SQL sobre taules, que compleixi els criteris de qualitat establerts a l'assignatura, i que retorni el mateix resultat que la consulta anterior sobre vistes.

c) Si es pot fer una solució que compleix els criteris de qualitat i que permet accedir a les mateixes dades consultant les taules, digueu quines avantatges pot aportar el fer-ho amb vistes.

2.2 Considereu la taula professors(dni,nomProf,telefon), propietat d'en Toni. Supposeu també la seqüència de sentències següent relativa a autoritzacions sobre la taula professors. Cada sentència està numerada i s'indica el nom de l'usuari que vol executar-la.

- 1 - Toni: GRANT SELECT ON professors TO Albert WITH GRANT OPTION
- 2 - Albert: GRANT SELECT ON professors TO Carme WITH GRANT OPTION
- 3 - Carme: GRANT SELECT(dni,telefon) ON professors TO Dolors WITH GRANT OPTION
- 4 - Carme: GRANT SELECT(dni,nomProf) ON professors TO Se WITH GRANT OPTION
- 5 - Toni: GRANT SELECT ON professors TO Se
- 6 - Toni: GRANT SELECT(telefon) ON professors TO Xavi
- 7 - Dolors: GRANT SELECT(dni,telefon) ON professors TO Xavi WITH GRANT OPTION
- 8 - Se: GRANT SELECT(dni,telefon) ON professors TO Xavi
- 9 - Dolors: GRANT SELECT(dni) ON professors TO Elena
- 10 - Se: GRANT SELECT(dni) ON professors TO Elena
- 11 - Toni: REVOKE SELECT ON professors FROM Se RESTRICT
- 12 - Carme: REVOKE SELECT(dni,telefon) ON professors FROM Dolors RESTRICT
- 13 - Dolors: REVOKE SELECT(dni) ON professors FROM Se CASCADE
- 14 - Albert: REVOKE SELECT ON professors FROM Carme CASCADE
- 15 - Toni: REVOKE SELECT ON professors FROM Albert RESTRICT

a) Quines d'aquestes sentències, si n'hi ha cap, no s'executaran amb èxit o no tindran cap efecte sobre la base de dades? Raoneu la resposta. Assumirem que les sentències que no s'executin amb èxit, no tindran cap efecte i es continuarà amb la sentència següent.

b) En acabar d'executar les sentències anteriors, quins privilegis tindran els usuaris Xavi i Albert sobre la taula professors?

c) Definiu els rols "rol-Xavi" i "rol-Albert" de tal manera que aquests rols tinguin els privilegis del Xavi i l'Albert després d'executar les sentències.

3. (2 punts) Donada la taula següent:

```
CREATE TABLE items (item integer primary key,  
                      name      char(25),  
                      qtt       integer,  
                      preu_total decimal(9,2));
```

i la regla de negoci: “ ***Una única sentència de modificació (update) no pot augmentar la quantitat total en estoc dels productes en més d’un 50%***”, ens demanen implementar amb triggers aquesta regla de negoci. Una possible solució en PostgreSQL seria (cada part de la solució està assenyalada amb una lletra per facilitar fer-hi referència):

- a)

```
CREATE TABLE temp(old_qtt integer);
```
- b)

```
CREATE FUNCTION update_items_before() RETURNS trigger AS $$  
BEGIN  
DELETE FROM temp;  
INSERT INTO temp SELECT sum(qtt) FROM items;  
RETURN NULL;  
END $$ LANGUAGE plpgsql;
```
- c)

```
CREATE FUNCTION update_items_after() RETURNS trigger AS $$  
DECLARE  
oldqtt integer default 0;  
newqtt integer default 0;  
BEGIN  
SELECT old_qtt into oldqtt FROM temp;  
SELECT sum(qtt) into newqtt FROM items;  
IF (newqtt>oldqtt*1.5) THEN  
RAISE EXCEPTION 'Violació regla de negoci';  
END IF;  
RETURN NULL;  
END $$ LANGUAGE plpgsql;
```
- d)

```
CREATE TRIGGER regla_negociBS BEFORE UPDATE ON items  
FOR EACH STATEMENT EXECUTE PROCEDURE update_items_before();
```
- e)

```
CREATE TRIGGER regla_negociAS AFTER UPDATE ON items  
FOR EACH STATEMENT EXECUTE PROCEDURE update_items_after();
```

- 3.1** Expliqueu quin és el principal problema d’aquesta solució, atenent als criteris de qualitat de triggers vistos a classe.
- 3.2** Quins canvis faríeu a d) i e) per tal superar aquest inconvenient? Per què?
- 3.3** Codifiqueu la nova funció c) que elimini els inconvenients de la solució donada quan es combini amb els canvis introduïts a 3.2.

4. (2 punts)

4.1 Donada la taula R(a,b) (clau primària subratllada) que conté les files {(‘a1’,1), (‘a2’,2)} i les dues transaccions següents:

T1: insert into R values (‘a3’,3); update R set b=b*2; commit;

T2: select * from R; select * from R; commit;

Suposant que el SGBD no implementa cap mecanisme de control de concurrència, digueu quins serien els possibles resultats de les dues sentències *select* de T2, en funció de ls possibles ordres d’execució de les transaccions. En cas que alguns d’aquests resultats siguin conseqüència de l’existència d’interferències, digueu quines serien aquestes interferències. Argumenteu breument les vostres respostes.

Considereu que les accions (R, RU, W) corresponents a una mateixa sentència SQL s’executen sempre totes seguides.

4.2 Supposeu ara que la transacció T2 s’executa concurrentment amb una transacció T3, en l’ordre que tot seguit s’indica:

T2	T3
select * from R	
	select * from R where a=‘a1’
	update R set b=b*2 where a=‘a1’
	commit
select * from R	
commit	

Suposant que les dues transaccions treballen amb nivell *de repeatable read*, que l’SGBD fa servir reserves S, X, que el grànul és la fila, i que la taula R conté les files {(‘a1’,1), (‘a2’,2)}, contesteu a les preguntes següents:

- Com quedaria l’horari en termes d’operacions de baix nivell? L’horari ha d’incloure, a més de les peticions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves i l’ordre d’execució de totes aquestes peticions.
- Quins horaris serials hi són equivalents?

4.3 Ens informen que la BD que conté la taula R, a causa d’un desastre, passa a estar innaccessible. En aquest escenari contesteu, argumentant les vostres respostes, a les preguntes següents:

- Quines propietats de les transaccions es veuen compromeses?
- Quines fonts d’informació necessita l’SGBD per reconstruir la BD?

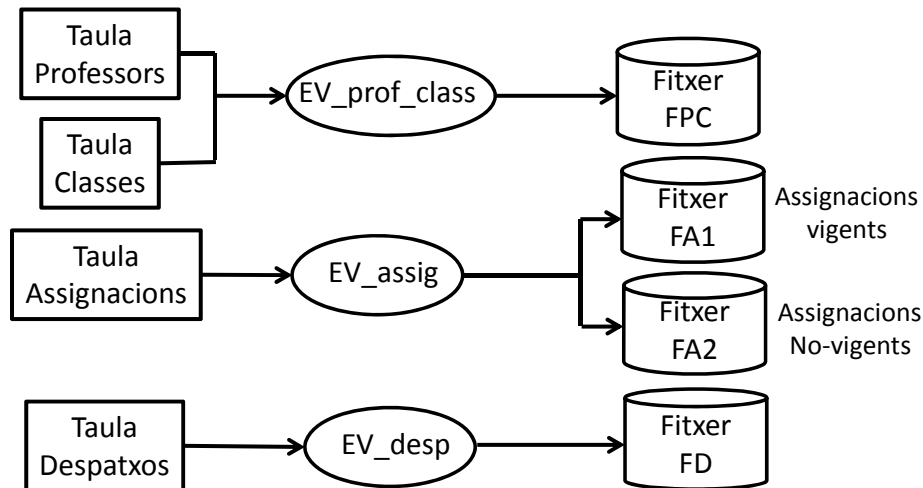
5. (2 punts) Considereu la base de dades de l'exercici 1.

5.1 Doneu un possible model conceptual en UML que generi, quan es fa la traducció a model relacional, les taules anteriors. En el UML hi ha d'haver: classes, atributs, associacions amb les seves multiplicitats, i les claus externes.

5.2 Supposeu ara, que a la base de dades de l'exercici 1 s'hi afegeix la taula *classes*, amb l'esquema següent:

```
classes (aula, horari, grup, dni)
-- on dni és clau forana que referencia professors.
```

a) Digueu a quin tipus correspon cadascun dels espais virtuals indicats a continuació:



b) Com sabeu, els diferents tipus d'espais virtuals afavoreixen o penalitzen certs tipus de consultes. Tenint en compte només els espais virtuals abans identificats, digueu quines de les consultes següents es veuen afavorides o penalitzades. Justifiqueu breument la resposta.

```
SELECT p.dni,p.nom,c.aula,c.grup,c.horari
FROM professors p, classes c
WHERE p.dni=c.dni;
```

```
SELECT d.modul, d.numero, d.superficie
FROM despatxos d, assignacions a
WHERE d.modul=a.modul AND d.numero=a.numero
      AND a.instantFI is NULL and d.modul='Omega';
```

5.3 Supposeu ara que la taula d'assignacions s'emmagatzema emprant un espai virtual de taula i que té aproximadament 10.000 tuples, que estan emmagatzemades en pàgines amb una mitjana de 10 tuples per pàgina. Sabeu també que hi ha aproximadament 1000 assignacions amb *dni* > '444' i que hi ha 300 assignacions al mòdul 'Omega'. De les assignacions amb *dni* > '444' n'hi ha 50 que són al mòdul 'Omega'. A més, se sap que s'han definit dos índexs B+ un per *dni* i un per mòdul. L'arbre B+ per *dni* és d'ordre *d*=157, és no agrupat, i té una ocupació de les pàgines de l'índex del 70 % en mitjana. L'arbre B+ per mòdul és d'ordre *d*=227, és no agrupat, i té una ocupació de les pàgines de l'índex del 60% en mitjana.

Donada la consulta següent, estimeu (i justifiqueu breument) el nombre de pàgines (d'índex i de dades) que es llegiran si la consulta es resol emprant els dos índexs amb estratègia d'intersecció de RIDs.

```
SELECT * FROM assignacions a WHERE a.dni > '444' AND a.modul='Omega'
```

Temps: 3 hores**Notes 26 gener matí Revisió: 27 gener matí****1. (2 punts)** Supposeu una base de dades amb les taules següents:

```
CREATE TABLE professors (  
    dni integer PRIMARY KEY,  
    nom_prof char(50),  
    categoria char(10),  
    sou integer);  
  
CREATE TABLE materies(  
    id_mat integer PRIMARY KEY,  
    nom_mat char(50) UNIQUE);  
  
CREATE TABLE assignatures (  
    id_mat integer REFERENCES materies,  
    codi_assig integer,  
    nom_assig char(50),  
    PRIMARY KEY (id_mat, codi_assig));  
// Feu atenció que una assignatura s'identifica per la matèria de què  
tracta i un codi d'assignatura.  
  
CREATE TABLE coordinadors (  
    id_mat integer,  
    codi_assig integer,  
    curs char(20),  
    dni integer REFERENCES professors,  
    PRIMARY KEY (id_mat, codi_assig, curs),  
    FOREIGN KEY (id_mat, codi_assig) REFERENCES assignatures);  
// Taula que guarda els coordinadors d'assignatura de cada curs.  
  
CREATE TABLE especialistes (  
    dni integer REFERENCES professors,  
    id_mat integer REFERENCES materies,  
    PRIMARY KEY (dni, id_mat));  
// Taula que guarda els especialistes de cada matèria.
```

- 1.1** Donar una sentència SQL per obtenir el nom dels professors que no han coordinat cap assignatura de la matèria amb nom Bases de Dades.
- 1.2** Donar una sentència SQL per obtenir els noms dels professors que han coordinat totes les assignatures de la matèria amb nom Matemàtiques.
- 1.3** Donar una seqüència d'operacions de l'àlgebra relacional per obtenir els noms de les matèries tals que hi ha algun professor que és especialista d'aquesta matèria i no ha sigut coordinador de cap assignatura d'aquesta matèria.

2. (2 punts) Partiu de la base de dades de l'exercici 1.**2.1** Supposeu l'assertió:

```
CREATE assertion LimitC check (  
    NOT EXISTS (SELECT * FROM professors  
                WHERE categoria = 'C' AND  
                sou > (SELECT avg(sou) FROM professors)));
```

Digueu quines sentències d'actualització poden violar l'assertió anterior. Per cada sentència cal indicar:

- Si és una sentència insert, delete o update.
- La taula sobre la que s'aplica (i en cas de sentències update a més cal indicar el/s atributs rellevants).
- Una extensió de la taula i un exemple de sentència que produeixi la violació suposant que la taula tingui aquesta extensió.

2.2 Es vol garantir que tot professor assignat com a coordinador d'una assignatura sigui especialista en la matèria de l'assignatura. Una possible implementació d'aquesta restricció amb un disparador és la següent:

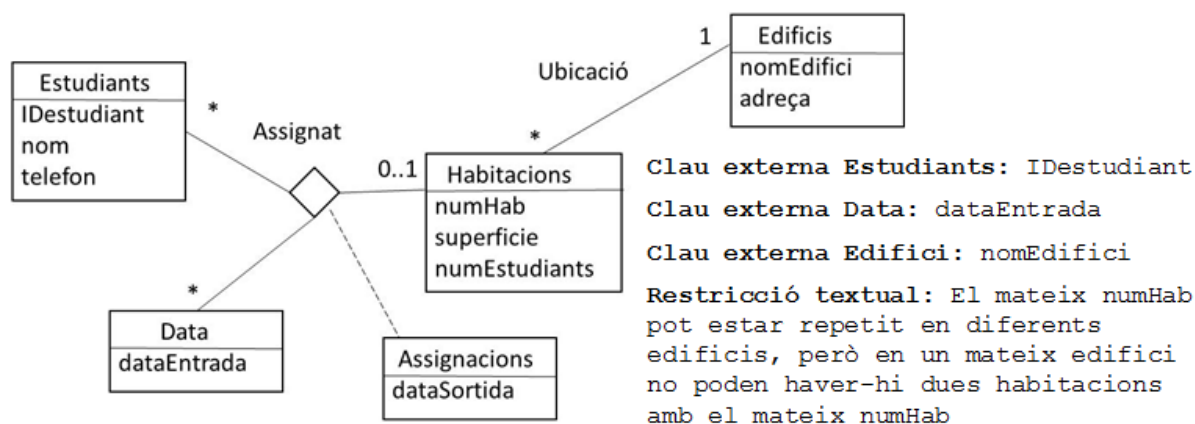
```
CREATE OR REPLACE FUNCTION restriccio() RETURNS Trigger AS $$
BEGIN
    IF (NOT EXISTS (SELECT * FROM especialistes
                    WHERE dni = NEW.dni AND id_mat = NEW.id_mat)) THEN
        RAISE EXCEPTION 'Professor no especialista en la matèria';
    END IF;
    RETURN NEW;
END;
$$language plpgsql;

CREATE TRIGGER restriccio BEFORE INSERT ON coordinadors
FOR EACH ROW EXECUTE PROCEDURE restriccio();
```

Es demana:

- Digueu si aquest disparador cobreix totes les sentències d'actualització que poden violar la restricció. En cas afirmatiu expliqueu perquè no cal cap altre disparador més, en cas negatiu digueu per quines sentències d'actualització i taules s'haurien de definir els disparadors que falten (i en cas de sentències update a més cal indicar el/s atributs rellevants).
- Digueu si es pot o no 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 pot fer.

3. (2 punts) Considerant el disseny conceptual en UML següent, d'assignació d'habitacions en un campus universitari:



3.1 Doneu disseny lògic que s'obté de traduir el UML anterior a model relacional (cal incloure: taules, atributs, restriccions de clau primària, i restriccions de clau forana).

3.2 Suposant les taules resultants de l'apartat 3.1:

- Doneu un exemple d'una vista que sigui actualitzable i definida com a WITH CHECK OPTION.
- Doneu una sentència d'actualització de la vista anterior que falli degut al WITH CHECK OPTION. Expliqueu breument perquè falla.

3.3 Supposeu que el propietari de les taules resultants de l'apartat 3.1 és en Pere, Supposeu també la seqüència de sentències següent relativa a autoritzacions sobre aquestes taules. Cada sentència indica el nom de l'usuari que l'executa.

Pere: GRANT select(numHab, superfície, numEstudiants) ON habitacions TO Joan WITH GRANT OPTION;

Pere: GRANT update(superfície, numEstudiants) ON habitacions TO Ana WITH GRANT OPTION;

Pere: GRANT select(numHab, superfície, numEstudiants) ON habitacions TO Pol WITH GRANT OPTION;

Pol: GRANT select(numHab, superfície, numEstudiants) ON habitacions TO Laia;

Ana: GRANT update(superfície, numEstudiants) ON habitacions TO Laia;

Joan: GRANT select(numHab, superfície, numEstudiants) ON habitacions TO Laia;

Un cop fet això en Pere decideix fer dos REVOKES:

Pere: REVOKE select(numHab, superfície, numEstudiants) ON habitacions FROM Pol RESTRICT;

Pere: REVOKE select(numHab, superfície, numEstudiants) ON habitacions FROM Joan RESTRICT;

Es demana:

- Dibuixar el diagrama d'autoritzacions després d'haver executat totes les sentències GRANT.
- Dibuixar el diagrama d'autoritzacions després d'executar els dos REVOKE, i explica breument l'efecte de cadascun d'ells.
- Justificar si la Laia podrà o no executar la sentència següent després de l'execució de totes les sentències GRANT i REVOKE.

UPDATE habitacions SET numEstudiants=numEstudiants+1 WHERE numHab=12;

4.(2 punts) Considereu una BD amb files A,...,L, emmagatzemades en 4 pàgines tal i com es mostra a continuació:

Pàgina 1: A,B,C

Pàgina 2: D,E,F

Pàgina 3: G,H,I

Pàgina 4: J,K,L

Suposarem que només es poden adquirir reserves amb modalitat exclusiva (X), és a dir que tant per R com per RU/W cal una reserva amb modalitat X. Definirem un "comboi " com un punt en el temps en el qual una transacció T té una reserva sobre un objecte O, i com a mínim existeixen dues transaccions esperant reservar aquest mateix objecte O.

4.1 Suposant que les reserves s'adquireixen a nivell de files i que tenim les tres transaccions següents:

T1: Lock(E,X), RU(E), Lock(H,X), R(H), W(E),Unlock(E), Unlock(H)

T2: Lock(A,X), RU(A), Lock(E,X), R(E), W(A), Unlock(A), Unlock(E)

T3: Lock(G,X), R(G), Lock(D,X), RU(D), W(D), Unlock(D), Unlock(G)

existeix algun horari a on tinguem un comboi? Si la resposta es afirmativa, cal donar el graf d'espera associat. En cas negatiu, argumenteu la resposta.

4.2 Pel mateix cas que l'apartat 4.1, existeix algun horari a on tinguem una abraçada mortal? Si la resposta es afirmativa, cal donar el graf d'espera associat. En cas negatiu, argumenteu la resposta.

4.3 Supposeu ara que les reserves s'adquireixen a nivell de pàgines, i que per tant les transaccions queden definides de la manera següent,

T1: Lock(P2,X), RU(E), Lock(P3,X), R(H), W(E),Unlock(P2), Unlock(P3)

T2: Lock(P1,X), RU(A), Lock(P2,X), R(E), W(A), Unlock(P1), Unlock(P2)

T3: Lock(P3,X), R(G), Lock(P2,X), RU(D), W(D), Unlock(P2), Unlock(P3)

existeix algun horari a on tinguem un comboi? Si la resposta es afirmativa, cal donar el graf d'espera associat. En cas negatiu, argumenteu la resposta.

4.4 Pel mateix cas que l'apartat 4.3, existeix algun horari a on tinguem una abraçada mortal? Si la resposta es afirmativa, cal donar el graf d'espera associat. En cas negatiu, argumenteu la resposta.

4.5 Usant les transaccions T1 i T2 de l'apartat 4.1, mostreu si és possible un horari on es vegi que el protocol de reserva en dues fases no és suficient per assegurar l'absència d'interferències. En cas de que no sigui possible trobar un horari com el que es demana, justifiqueu la resposta.

5. (2 punts) Suposa una base de dades on hi ha la taula següent que té 1.000.000 files, distribuïdes uniformement entre 20.000 pàgines segons un espai virtual de taula:

`clients(idClient, nom, descripcio, venedor, saldo, classificacio, sucursal)`

Els clients tenen identificadors que van del 1 al 1.000.000, i pertanyen a 500 sucursals. Cada sucursal té una mitjana de 2.000 clients, i una mitjana de 50 venedors. Considereu que sobre la taula hi ha definits els índexs:

- índex no agrupat per l'atribut *idClient*. Ordre d=300, ple al 90%, 2 nivells.
- índex agrupat pels atributs *sucursal*, *venedor*. Ordre d=80, ple al 75%, 3 nivells.

5.1 Indiqueu el cost en número de pàgines d'índex i de dades de resoldre les consultes següents:

- a) `select * from clients where idClient > 800.000 and idClient <=805.000;`
- b) `select * from clients where classificacio = 'A';`
- c) `select distinct(venedor) from clients where sucursal = 500;`

5.2 Supposem ara que la taula *clients* està en un espai virtual d'agrupació junt amb les files d'una altra taula *sucursals(sucursal, adreça, ciutat, telefon, director)*.

- a) Dibuixeu quina seria l'estructura de una pàgina de dades en aquest cas.
- b) Doneu una sentència SQL que es vegi afavorida per l'organització de les dues taules segons l'espai virtual d'agrupació.
- c) Expliqueu com quedarien afectats els costos de les tres consultes de l'apartat 5.1 a per aquest canvi d'organització de les pàgines de dades.

1. (2 punts) Suposeu una base de dades amb les taules següents:

```
CREATE TABLE professors (  
    dni integer PRIMARY KEY,  
    nom_prof char(50),  
    categoria char(10),  
    sou integer);  
  
CREATE TABLE materies(  
    id_mat integer PRIMARY KEY,  
    nom_mat char(50) UNIQUE);  
  
CREATE TABLE assignatures (  
    id_mat integer REFERENCES materies,  
    codi_assig integer,  
    nom_assig char(50),  
    PRIMARY KEY (id_mat, codi_assig));  
// Feu atenció que una assignatura s'identifica per la matèria de què  
tracta i un codi d'assignatura.  
  
CREATE TABLE coordinadors (  
    id_mat integer,  
    codi_assig integer,  
    curs char(20),  
    dni integer REFERENCES professors,  
    PRIMARY KEY (id_mat, codi_assig, curs),  
    FOREIGN KEY (id_mat, codi_assig) REFERENCES assignatures);  
// Taula que guarda els coordinadors d'assignatura de cada curs.  
  
CREATE TABLE especialistes (  
    dni integer REFERENCES professors,  
    id_mat integer REFERENCES materies,  
    PRIMARY KEY (dni, id_mat));  
// Taula que guarda els especialistes de cada matèria.
```

- 1.1** Doneu una sentència SQL per obtenir els identificadors de les assignatures per les que no hi ha hagut cap coordinador de l'assignatura que no sigui especialista de la matèria a la qual pertany l'assignatura.
- 1.2** Indiqueu quina és la cardinalitat de la relació resultant de l'execució de les consultes següents. Considereu que les taules de la base de dades tenen P professors, M matèries, A assignatures. Tingueu en compte també que tota matèria té una o més assignatures, i que totes les assignatures tenen coordinador. Raona breument la resposta. Si no pots donar una única xifra, proporciona una fita inferior i una superior de la cardinalitat.

- a)

```
select count(*)  
from coordinadors natural inner join assignatures  
group by id_mat
```
- b)

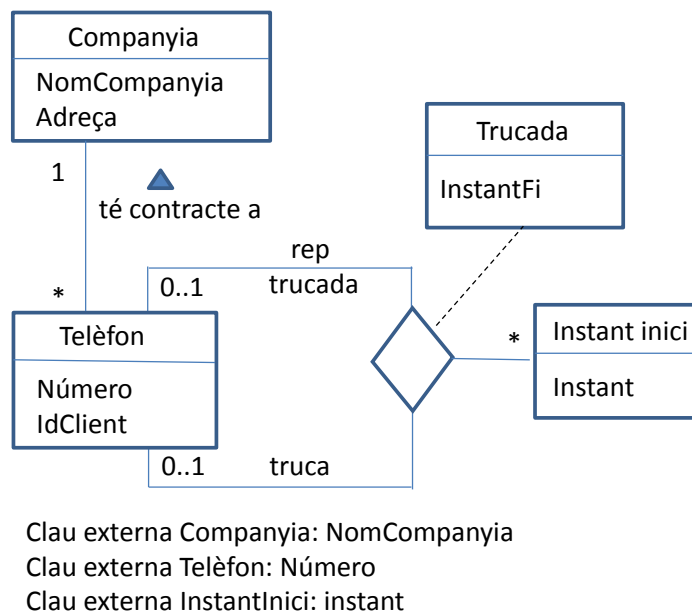
```
select *  
from matèries m  
where not exists (select *  
                  from assignatures a  
                  where a.id_mat = m.id_mat)
```

- c) `select *`
`from professors, signatures`
- d) `A = signatures * coordinadors`
`B = A[dni]`

1.3 Doneu una seqüència d'operacions de l'àlgebra relacional per obtenir el dni dels coordinadors que no són especialistes i que coordinen assignatures de matèries que no tenen especialistes.

2. (2 punts)

2.1 Considerant el disseny conceptual en UML següent, doneu el disseny lògic que s'obté de traduir el UML anterior a model relacional (cal incloure: taules, atributs, restriccions de clau primària, restriccions de clau forana, i altres restriccions que es puguin derivar del model).



2.2 Considereu la base de dades següent:

`departaments(idDept, nomDept, idEmplCap)`

{idEmplCap} referencia Empleats

`empleats(idEmp, nomEmpl, idDept)`

{idDept} referencia departaments

Suposant que hi ha tres departaments amb identificador X, Y i Z, i que els identificadors dels seus caps són E1, E2 i E3 respectivament. Doneu les sentències SQL necessàries per tal que cada cap de departament pugui accedir per consultar únicament els empleats del seu departament.

2.3 L'arquitectura ANSI/SPARC d'un sistema de gestió de base de dades té tres nivells: nivell intern, nivell conceptual i nivell extern. Contesteu les preguntes següents:

- Poseu exemples de tres components que podria tenir la base de dades de l'apartat 2.2 i que podrien formar part respectivament de l'esquema intern, l'esquema conceptual i l'esquema extern d'aquesta base de dades.
- Explica què s'entén per independència física de dades en el marc de l'arquitectura ANSI/SPARC. Explica-ho usant els components que et facin falta dels que has posat com a exemple en l'apartat 2.3.a.
- Explica què s'entén per independència lògica de dades en el marc de l'arquitectura ANSI/SPARC. Explica-ho usant els components que et facin falta dels que has posat com a exemple en l'apartat 2.3.a.

3.(2 punts) Considereu l'esquema de la base de dades següent:

```
CREATE TABLE ProductoresTV (  
    id_prod int primary key,  
    descripcio char(20),  
    pressupost_prod int not null check (pressupost_prod > 0),  
    vendes real);  
  
CREATE TABLE Equips (  
    id_equip int primary key,  
    descripcio char(20),  
    pressupost_eq int not null check (pressupost_eq >= 0),  
    id_prod int not null references ProductoresTV);  
  
CREATE or REPLACE FUNCTION spPrimer() RETURNS trigger AS $$  
BEGIN  
    IF (new.vendes > old.vendes) THEN  
        UPDATE Equips  
        SET pressupost_eq = pressupost_eq + 0.001*new.vendes  
        WHERE id_prod = old.id_prod;  
    END IF;  
    RETURN null;  
END;  
$$LANGUAGE plpgsql;  
  
CREATE TRIGGER primer  
AFTER UPDATE OF vendes ON ProductoresTV  
FOR EACH ROW EXECUTE PROCEDURE spPrimer();  
  
CREATE or REPLACE FUNCTION spSegon() RETURNS trigger AS $$  
BEGIN  
    UPDATE ProductoresTV  
    SET pressupost_prod = pressupost_prod + new.pressupost_eq  
    WHERE id_prod = old.id_prod;  
    RETURN null;  
END;  
$$LANGUAGE plpgsql;  
  
CREATE TRIGGER segon  
AFTER UPDATE OF pressupost_eq ON Equips  
FOR EACH ROW EXECUTE PROCEDURE spSegon();
```

Suposeu que el contingut inicial de la base de dades és el següent:

```
INSERT INTO ProductoresTV VALUES(1, 'ImageTV', 20000, 500000);  
INSERT INTO ProductoresTV VALUES(2, 'KRTV', 15000, 250000);  
INSERT INTO Equips VALUES (11, 'So', 10000, 1);  
INSERT INTO Equips VALUES (12, 'Camares', 10000, 1);  
INSERT INTO Equips VALUES (13, 'Marqueting', 20000, 1);  
INSERT INTO Equips VALUES (14, 'Documentacio', 5000, 1);  
INSERT INTO Equips VALUES (21, 'Localitzacions', 10000, 2);  
INSERT INTO Equips VALUES (22, 'Reporters', 10000, 2);
```

3.1 Digueu quin és el contingut final de la base de dades, després de l'execució de la sentència SQL següent i justifiqueu breument la resposta, explicant les accions que segueix el SGBD a conseqüència d'aquest update:

```
UPDATE ProductoresTV SET vendes = 560000 WHERE id_prod = 1;
```

3.2 Quina seria la diferència respecte l'apartat 3.1 suposant que els tipus dels triggers canvien de FOR EACH ROW a FOR EACH STATEMENT?

3.3 Definiu una asserció en SQL Standard per garantir el compliment de la restricció següent: “Tota productora que tingui un pressupost més gran o igual que 20000 ha de tenir al menys 3 equips que treballen per ella”

4. (2 punts) Donat un SGBD sense cap mecanisme de control de concurrència, suposem que es produeix l'horari següent:

Temps	T1	T2	T3
10		RU(B)	
20		W(B)	
30			
40	RU(A)		
50	W(A)		
60	R(D)		
70	R(B)		
80			R(E)
100			RU(A)
120			W(A)
130			RU(F)
140		RU(B)	
150		W(B)	
160			W(F)
170	RU(E)		
180	W(E)		
190		COMMIT	
200			COMMIT
210	COMMIT		

4.1 Dibuixeu el graf de precedències associat a l'horari.

4.2 Es produeixen interferències? En cas de resposta negativa, argumenteu breument la vostra resposta. En cas de resposta positiva digueu quina/es interferències es produeixen (cal donar el nom de la interferència, grànul i transaccions implicades). Quins horaris serials hi són equivalents? Justifica la resposta.

4.3 Suposem que s'incorporen tècniques de reserva. Les transaccions estan definides com SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED.

- Determinar i executar la seqüència d'operacions read i write que es generen, el seu ordre i les peticions de lock i unlock necessàries mostrant les esperes que s'hi produeixen. En cas que cap acció de l'horari pugui executar-se en un moment donat, no continueu i justifiqueu el motiu.
- Si ha alguna interferència en l'horari resultat de l'apartat 4.4.a digues quina/es és/són i entre quines transaccions es produeix/en. Sinó n'hi ha cap justifica la resposta.

4.4 Responen les mateixes preguntes de l'apartat 4.4 en cas de que les transaccions estan definides com SET TRANSACTION ISOLATION LEVEL READ COMMITTED.

5. (2 punts) Suposa una base de dades on hi ha una taula T que té 300.000 files. Hi ha un índex no agrupat sobre aquesta taula definit sobre l'atribut ATR que és clau primària i és de tipus char i ocupa 6 bytes. Els apuntadors a les pàgines de l'índex ocupen 4 bytes i els RID ocupen també 4 bytes. Les pàgines són de 1004 bytes i el factor de bloqueig de les pàgines de dades és 25.

5.1 Determineu la d òptima d'aquest índex

5.2 Considereu ara que la d òptima és 50 i que l'arbre està ple al 69%. Es vol que la consulta SELECT * FROM T where ATR = constant tingui un cost de 2 accessos a pàgines de disc, independentment del valor concret de la constant. Tenint en compte que es pot tenir més d'un nivell de l'arbre a memòria, és això possible? Si la resposta és afirmativa, determineu el nombre pàgines de l'índex que caldria tenir a memòria per tal de que això fos possible. En cas de no ho sigui, justifiqueu la resposta.

5.3 Supposeu ara que l'índex està definit sobre un atribut que no és clau primària. L'atribut té en mitjana 5 valors repetits. Responen a la mateixa pregunta que a l'apartat 5.2.

Temps: 3 hores

Notes 31 gener matí Revisió: 1 febrer tarda

Cada pregunta s'ha de lliurar en un full separat, excepte la pregunta 1, que s'ha de lliurar els apartats 1.1 i 1.2 en un full, i els 1.3 i 1.4 en un altre

1. (2.5 punts) Supposeu una base de dades amb les taules següents:

```
CREATE TABLE persones
(nif CHAR(9),
 nom CHAR(30) NOT NULL,
 ciutatRes CHAR(20),
 PRIMARY KEY (nif));
-- ciutatRes és on viu la persona

CREATE TABLE tipus
(tipusId CHAR(20),
 numMaxInscrits INTEGER,
 PRIMARY KEY (tipusId));

CREATE TABLE activitats
(tipusId CHAR(20),
 instIni INTEGER,
 descripcio CHAR(30),
 nifP CHAR(9) NOT NULL,
 preu INTEGER,
 ciutat CHAR(20),
 PRIMARY KEY (tipusId, instIni),
 FOREIGN KEY (tipusId) REFERENCES tipus(tipusId),
 FOREIGN KEY (nifP) REFERENCES persones(nif));
-- el nifP és el de la persona que és professor de l'activitat
-- ciutat és on es fa l'activitat

CREATE TABLE inscripcions
(tipusId CHAR(20),
 instIni INTEGER,
 nifI CHAR(9), -- persona inscrita
 instPagament INTEGER,
 PRIMARY KEY (tipusId, instIni, nifI),
 FOREIGN KEY (tipusId, instIni)
 REFERENCES activitats(tipusId,instIni),
 FOREIGN KEY (nifI) REFERENCES persones);
-- el nifI és el de la persona que s'inscriu a l'activitat
```

Amb les sentències de càrrega de les taules:

```
INSERT INTO persones VALUES
('111', 'Anna', 'Barcelona'),
('222', 'Alicia', 'Sabadell'),
('333', 'Rosa', 'Gava');

INSERT INTO tipus VALUES
('body_pump', 15),
('zumba', 20),
('aerobic', 25);

INSERT INTO activitats VALUES
('body_pump', 212, null, '111', 25, 'Barcelona'),
('zumba', 318, null, '222', 25, 'Sabadell'),
('aerobic', 118, null, '222', 20, 'Sabadell');

INSERT INTO inscripcions VALUES
('zumba', 318, '333', 10),
('aerobic', 118, '333', 10);
```

1.1 Per cadascuna de les situacions que es plantegen a continuació indica si és o no possible, tenint en compte les restriccions definides a la base de dades. En cas que no sigui possible indica la restricció que ho evita, en cas que sigui possible dóna els inserts per tal que passi.

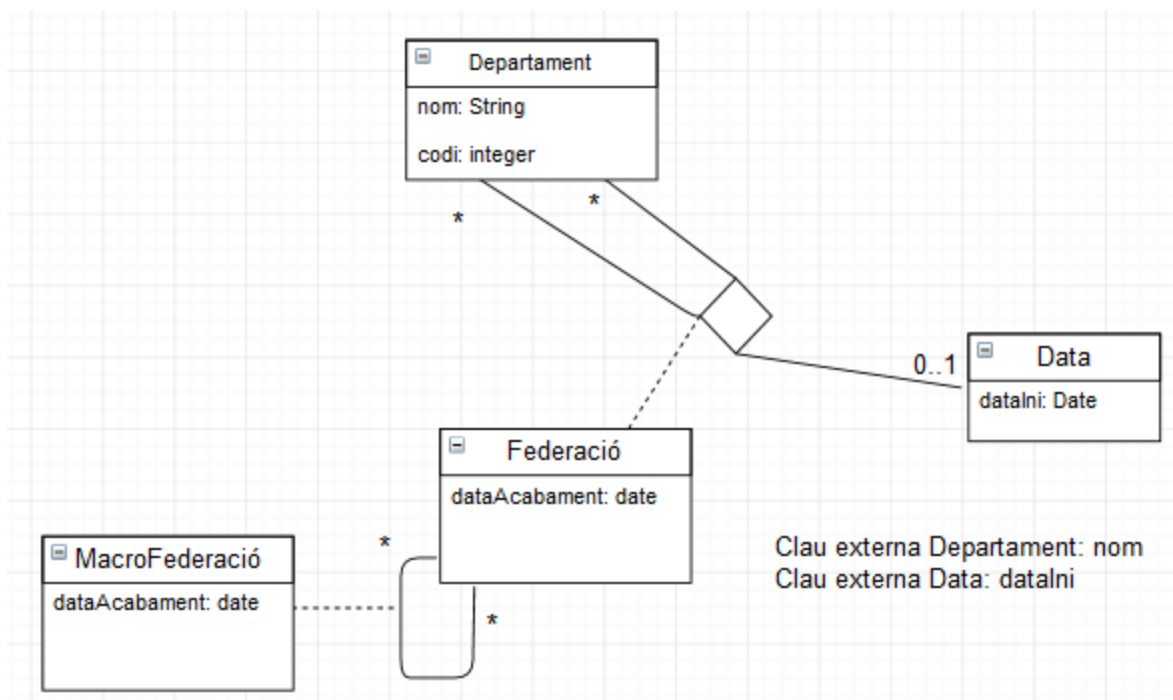
- És possible que hi hagi dues activitats del mateix tipus que comencen al mateix instant impartides per professors diferents?
- És possible que una persona s'inscrigui en una activitat de la qual ell mateix és professor?
- És possible que una persona s'inscrigui en dues activitats del mateix tipus?

1.2 Donada la vista:

```
CREATE VIEW nose(identI, nomI, identP, nomP, quantitat) AS
SELECT pI.nif, pI.nom, pP.nif, pP.nom, COUNT(*)
FROM persones PI, persones pP,
     inscripcions i NATURAL INNER JOIN activitats a
WHERE pI.nif = i.nifI AND a.nifP = pP.nif
GROUP BY pI.nif, pP.nif
```

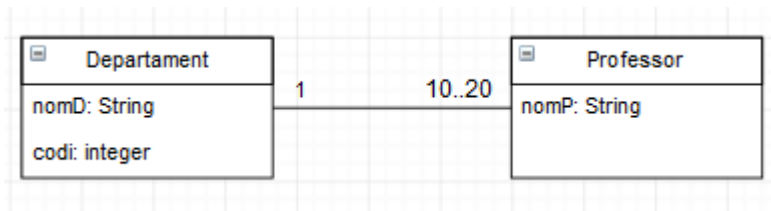
- Explica breument quines dades s'obtenen quan es fa un select de la vista.
 - Quin és el contingut de la vista després d'executar els inserts que us donem en l'enunciat?
 - És actualitzable? Per què?
- 1.3** Doneu una sentència SQL per obtenir el nom de les persones que són professors d'alguna activitat en la que no hi ha cap inscrit que visqui en la ciutat en la que es fa l'activitat.
- 1.4** Doneu un conjunt de sentències d'àlgebra relacional que obtenir el nif de les persones que són professors de dues o més activitats que es fan en una mateixa ciutat.

2. (2.5 punts) Considerant el disseny conceptual en UML següent:



2.1 Doneu disseny lògic que s'obté de traduir el UML anterior a model relacional (cal incloure: taules, atributs, restriccions de clau primària, i restriccions de clau forana).

2.2 Considereu ara el següent disseny conceptual:



que ha donat lloc a les taules:

```
Departament(nomD, codi)
```

```
Professor(nomP, nomD)
```

nomD és clau forana que referencia a Departament

nomD és NOT NULL

- Seria possible expressar en PostgreSQL la cardinalitat 10..20 fent servir alguna restricció de columna o de taula sobre les taules anteriorment definides? En cas de que sigui possible, doneu la sentència. En cas contrari, justifiqueu la resposta.
- Seria possible expressar en SQL estàndard la cardinalitat 10..20 fent servir una asserció. En cas de sigui possible, doneu la sentència. En cas contrari, justifiqueu la resposta.
- Doneu els triggers necessaris sobre la taula Professor per tal de assegurar el compliment de les cardinalitats. Podeu fer servir la plantilla de triggers de PostgreSQL:

```
CREATE TRIGGER nom [BEFORE/AFTER] [UPDATE (of column)/DELETE/INSERT] ON taula
[FOR EACH ROW/FOR EACH STATEMENT] execute procedure nomp();
```

```
CREATE FUNCTION nomp() RETURNS trigger AS $$
```

```
BEGIN
```

```
END; $$ LANGUAGE plpgsql;
```

3. (2.5 punts) Considera la base de dades i la transacció següent:

<i>Esquema</i>	<i>Extensió</i>
oficines(numOficina, adreça)	Oficines
comptes(numOficina, numCompte, saldo)	12 Barcelona
{numOficina} referencia oficinas	24 Vic
	comptes
	12 450 60
	24 670 200

<i>T1</i>
UPDATE comptes SET saldo = saldo + 50 WHERE numOficina = 12 AND numCompte = 450;
UPDATE comptes SET saldo = saldo - 50 WHERE numOficina = 24 AND numCompte = 670;
commit;

3.1 Interferències de lectura no repetible

- Escriu un horari en que participi la transacció T1 i on hi hagi únicament una interferència de lectura no repetible. En el nou horari s'ha d'afegir una transacció T2 que només pot utilitzar sentències select.

b) Indica quin és el nivell d'aïllament mínim que evita aquest tipus d'interferències.

3.2 Interferències d'anàlisi inconsistent

a) Escriu un horari en que participi la transacció T1 i on hi hagi únicament una interferència d'anàlisi inconsistent. En el nou horari s'ha d'afegir una transacció T2 que només pot utilitzar sentències `select`.

b) Indica quin és el nivell d'aïllament mínim que evita aquest tipus d'interferències.

3.3 Interferències de fantasmes

a) Escriu un horari en que participi la transacció T1 i on hi hagi únicament una interferència de fantasmes. En el nou horari s'ha d'afegir una transacció T2 que només pot utilitzar sentències `select`.

b) Tradueix l'horari a operacions R, RU i W tenint en compte que el grànul és la fila.

c) Indica quin és el nivell d'aïllament mínim que evita aquest tipus d'interferències.

d) Dona l'horari de l'apartat b amb les reserves i les esperes incorporades segons el nivell d'aïllament indicat.

4. (2.5 punts) Considerant la base de dades de l'exercici anterior. Suposa que a la taula oficines hi ha 3.000 oficines, i que a la taula comptes hi ha 15.000.000 comptes.

4.1 Indica si com a resultat de cadascuna de les consultes anteriors es poden obtenir o no resultats repetits. En cas que sí, dona un contingut de les taules que faci que s'obtinguin resultats repetits. En cas que no, explica perquè no pot passar.

a) `SELECT COUNT(*) FROM comptes WHERE saldo>200 GROUP BY numOficina;`

b) `SELECT numCompte FROM comptes WHERE numOficina=555;`

c) `SELECT numOficina FROM oficines NATURAL INNER JOIN comptes;`

4.2 Suposa que sobre la taula comptes hi ha únicament definit un índex agrupat (arbre B+) multi-atribut definit pels atributs numOficina, numCompte. La d de l'arbre és 70, i està ple al 80%. Suposa que cada oficina té una mitjana de 5.000 comptes, i que el factor de bloqueig de les pàgines de dades és 100. Explica quina és la manera òptima de resoldre cadascuna de les consultes següents. Calcula els cost de cada consulta, indicant el que significa cada factor que intervé en el càlcul.

a) `SELECT * FROM comptes ORDER BY numOficina, numCompte;`

b) `SELECT * FROM comptes WHERE numOficina=350 AND numCompte = 18;`

c) `SELECT numOficina, numCompte FROM comptes WHERE numOficina=480;`

4.3 Suposa que tenim un índex no agrupat (arbre B+) definit sobre l'atribut saldo que té 50.000 nodes fulla, i que l'índex està ple als 70%.

a) Indica quina és la d de l'arbre. Explica com l'has obtingut.

b) Explica on surten els RIDs en un índex arbre B+. Indica quants RIDs hi haurà en total en els nodes de l'índex indicat. Explica com has fet els càlculs.

Temps: 3 hores

Notes 4 juliol Revisió: 5 juliol matí

Cada pregunta s'ha de lliurar en un full separat, excepte la pregunta 1, que s'ha de lliurar els apartats 1.1 i 1.2 en un full, i els 1.3 i 1.4 en un altre

1. (2.5 punts) Supposeu una base de dades amb les taules següents:

```
create table departaments
(codiDept integer,
nomDept char(50) unique,
telefonDept char(15),
primary key (codiDept));

create table professors
(dni char(50),
nomProf char(50) unique,
telefonProf char(15),
sou integer not null check(sou>0),
dniCoord char(50),
codiDept integer not null,
primary key (dni),
foreign key (dnicoord) references professors,
foreign key (codiDept) references departaments);
-- dniCoord correspon al dni del coordinador del professor
-- codiDept correspon al departament al que pertany el professor

create table despatxos
(modul char(5),
numero char(5),
superficie integer not null check(superficie>12),
codiDept integer,
primary key (modul,numero),
foreign key (codiDept) references departaments);
-- codiDept correspon al departament al que pertany el despatx

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);
-- instantFi té valor null quan una assignacio és encara vigent.
```

1.1 Doneu una sentència SQL per obtenir els departaments tals que totes les assignacions als seus despatxos han sigut assignacions de professors del departament. En la consulta han d'aparèixer també els departaments pels que no hi ha hagut cap assignació a despatxos del departament.

1.2 Escriviu les sentències SQL per tal que l'usuari X (el propietari de les 4 taules) permeti que es puguin fer les operacions següents sobre les taules anteriors:

- a) L'usuari A pot modificar tots els atributs de la taula assignacions i de la taula professors, i també modificar el departament dels despatxos. Pot passar aquests privilegis a altres usuaris.
- b) L'usuari A permet que l'usuari C pugui modificar el departament dels despatxos.
- c) L'usuari X permet que l'usuari C modifiqui la taula despatxos. L'usuari C pot passar aquests privilegis a altres usuaris.
- d) L'usuari X desautoritza l'usuari A per modificar l'atribut codiDept de la taula despatxos en mode RESTRICT.
- e) Dibuixa el diagrama d'autoritzacions després d'executar les sentències de a), b), c) i abans de d), i el diagrama d'autoritzacions resultant després de la sentència d).

1.3 Doneu una seqüència d'operacions d'àlgebra relacional per obtenir el codi i nom dels departaments que han tingut o tenen professors assignats a despatxos d'un altre departament.

1.4 Suposant que la quantitat de departaments és DP, la quantitat de professors és P, la quantitat de despatxos és DX, i la quantitat d'assignacions és A, digueu i justifiqueu quin serà l'esquema i la cardinalitat mínima i màxima de la relació R que s'obté de cadascuna de les seqüències d'operacions d'àlgebra relacional següents:

- a) $R = \text{departaments} * \text{professors}$
- b) $A = \text{professors} \{ \text{codiDept} \rightarrow \text{cd} \}$
 $R = \text{departaments} [\text{codiDept} * \text{cd}] A$
- c) $A = \text{professors} \{ \text{codiDept} \rightarrow \text{cd} \}$
 $R = \text{departaments} [\text{codiDept} < \text{cd}] A$

2. (2.5 punts) Considereu la base de dades de l'exercici 1.

2.1 Doneu el disseny conceptual en UML tal que al traduir a model relacional donaria com a resultat l'esquema de la base de dades de l'exercici 1.

2.2 Definiu una asserció en SQL Standard per garantir el compliment de la restricció següent: "No existeix cap departament amb més de 5 professors que tinguin un coordinador que no és del departament".

2.3 Indiqueu els disparadors que caldria per implementar l'asserció definida a l'apartat 2.2 en PostgreSQL.

Per a cada disparador cal que indiqueu: l'esdeveniment activador, la taula, el tipus de disparador, i la justificació del tipus de disparador escollit. En cas que l'esdeveniment activador sigui un UPDATE cal també explicitar els atributs rellevants.

Tingueu en compte que les restriccions d'integritat definides a la BD (primary key, foreign key,...) es violen amb menys freqüència que la restricció comprovada per aquests disparadors.

3.(2.5 punts) Considereu les taules R(A,B) i S(C). Suposeu que les taules estan buides inicialment i que el grànul de concurrència és la tupla. Per a cadascun de les parelles de transaccions següents determineu tots els possibles resultats [X,Y], on X és el resultat del primer count de T1 i Y el resultat del segon.

Per a cada resultat és imprescindible justificar la resposta en base als nivells d'aïllament de les transaccions, i les potencials esperes degudes a locks. Justifiqueu també perquè no hi ha cap altre resultat possible a part dels que proposeu.

3.1 T1:

```
Set Transaction Isolation Level Read Committed;
Select count(*) From R;
Select count(*) From S;
Commit;
```

T2:

```
Set Transaction Isolation Level Serializable;
Insert Into R Values (1,2);
Insert Into S Values (3);
Commit;
```

3.2 T1:

```
Set Transaction Isolation Level Read Committed;
Select count(*) From R;
Select count(*) From S;
Commit;
```

T2:

```
Set Transaction Isolation Level Serializable;
Insert Into R Values (1,2);
Insert Into R Values (3,4);
Commit;
```

3.3 T1:

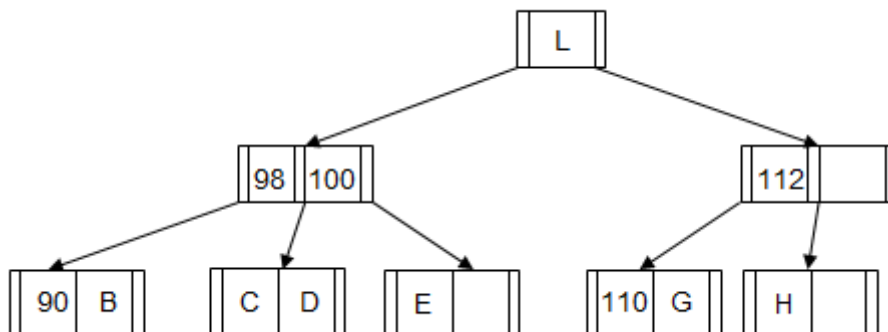
```
Set Transaction Isolation Level Repeatable Read;
Select count(*) From R;
Select count(*) From R;
Commit;
```

T2:

```
Set Transaction Isolation Level Serializable;
Insert Into R Values (1,2);
Commit;
```


4. (2.5 punts)

4.1 Donat l'arbre B+ següent, d'ordre $d=1$, i definit per a l'atribut a de la taula $T(\underline{a}, b)$, indiqueu quins són els valors que poden prendre B, C, D, E, G, H, L. Justifica breument les respostes.



4.2 Considereu la taula assignacions de la base de dades de la pregunta 1. Suposeu que la taula té 100.000 de files, i un factor de bloqueig de 10 files per pàgina. Suposeu també que: hi ha 18 assignacions del professor 123 a la taula assignacions, de les quals només 2 són assignacions vigents (`instantFi` null); hi ha 3000 assignacions al mòdul A en despatxos amb número superior a 100, de les quals només 450 són assignacions vigents.

Expliqueu com es resol i mostreu com es calcula el cost de la consulta següent en els casos indicats a sota la consulta.

```
select * from assignacions
where modul = 'A'
      and numero >= '100'
      and dni = '123'
      and instantFi is null
```

- En cas de que no existeix cap índex.
- En cas que s'utilitzi un índex arbre B+ agrupat multiatribut definit per als atributs `modul`, `numero` amb 3 nivells, una d de 146 i una ocupació del 80%.
- En cas que s'utilitzi un índex arbre B+ no agrupat definit per l'atribut `dni` amb 2 nivells, una d de 292 i una ocupació del 70%.

4.3 Sabent que la taula professors de la base de dades de la pregunta 1 té 6 atributs. Indica quants índexs (arbres B+) agrupats i quants índexs no agrupats, sobre un únic atribut, poden existir de manera simultània per a aquesta taula. Justifica la resposta.

1. (2 punts) Supposeu una base de dades amb les taules següents:

```
CREATE TABLE EquipsFutbol(  
    nomEquip char(30),  
    localitat char(50),  
    nomEstadi char(100) UNIQUE NOT NULL,  
    PRIMARY KEY(nomEquip));  
  
CREATE TABLE Partits(  
    nomEquipLocal char(30),  
    dataPartit date,  
    nomEquipVisitant char(30) NOT NULL,  
    golsEquipL integer NOT NULL CHECK(golsEquipL >=0),  
    golsEquipV integer NOT NULL CHECK(golsEquipV >=0),  
    PRIMARY KEY (nomEquipLocal,dataPartit),  
    FOREIGN KEY (nomEquipLocal) REFERENCES EquipsFutbol,  
    FOREIGN KEY (nomEquipVisitant) REFERENCES EquipsFutbol,  
    UNIQUE (nomEquipVisitant, dataPartit),  
    CHECK(nomEquipLocal <> nomEquipVisitant));  
  
CREATE TABLE Jugadors(  
    dniJugador char(9),  
    nom char(50) NOT NULL,  
    nomEquip char(30) NOT NULL,  
    PRIMARY KEY (dniJugador),  
    FOREIGN KEY (nomEquip) REFERENCES EquipsFutbol);  
  
CREATE TABLE Alineacions(  
    nomEquipLocal char(30),  
    dataPartit date,  
    dniJugador char(9),  
    gols integer NOT NULL CHECK(gols>=0),  
    numTargetes integer NOT NULL CHECK(numTargetes>=0),  
    PRIMARY KEY (nomEquipLocal,dataPartit,dniJugador),  
    FOREIGN KEY (nomEquipLocal,dataPartit) REFERENCES Partits,  
    FOREIGN KEY (dniJugador) REFERENCES Jugadors);
```

1.1 Escriviu una sentència SQL per obtenir els equips que no han perdut cap partit jugat a fora i que han guanyat tots els partits jugats a casa. Es diu que un equip juga un partit a casa quan és l'equip local del partit, i que un equip juga a fora quan és l'equip visitant del partit. Per cadascun dels equips que surtin en el resultat de la consulta es vol el nom de l'equip i la quantitat total de gols que ha fet en els partits jugats a casa.

1.2 Supposeu que volem una sentència SQL per trobar els jugadors (dniJugador, nom) que no han estat en cap alineació. Digueu, per cadascuna de les sentències següents **si dóna el resultat correcte o incorrecte**.

- En cas de que el resultat que doni sigui **correcte**, digueu si hi ha alguna part de la consulta que no passa els criteris de qualitat de l'assignatura, i quina és aquesta part.
- En cas que el resultat que doni sigui **incorrecte**, donar un contingut de les taules, el resultat que dóna la consulta tal com està per aquest contingut i el resultat que hauria de donar si fos correcte.

```
a) SELECT j.dniJugador, j.nom
   FROM jugadors j
   WHERE j.dniJugador NOT IN (SELECT a.dniJugador
                              FROM alineacions a, jugadors ju
                              WHERE a.dniJugador=ju.dniJugador);
```

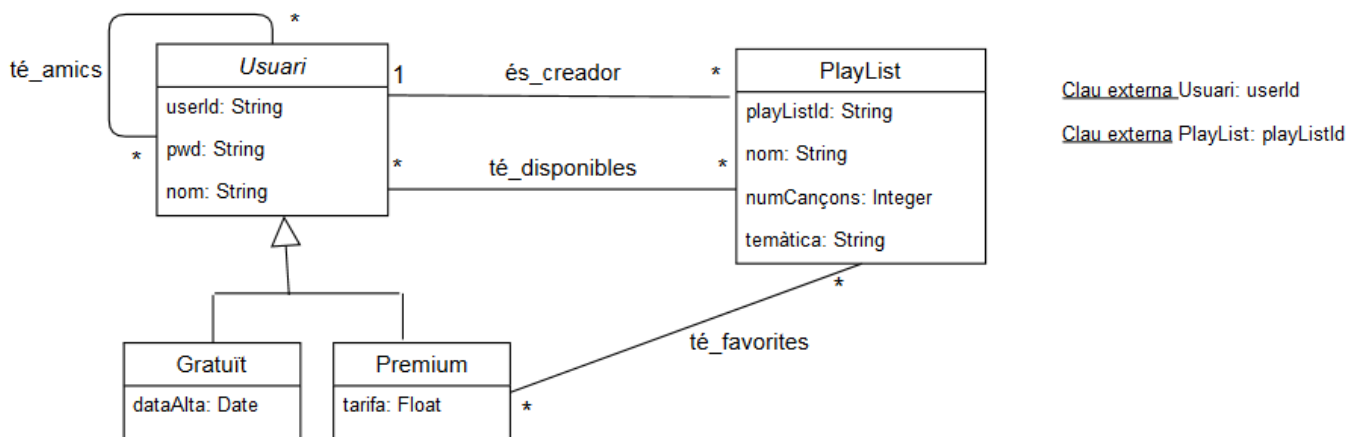
```
b) SELECT DISTINCT j.dniJugador, j.nom
   FROM jugadors j
   WHERE NOT EXISTS (SELECT * FROM alineacions a
                     WHERE a.dniJugador=j.dniJugador);
```

```
c) SELECT DISTINCT j.dniJugador, j.nom
   FROM jugadors j, alineacions a
   WHERE j.dniJugador = a.dniJugador and
         j.dniJugador NOT IN (SELECT al.dniJugador
                              FROM alineacions al);
```

1.3 Escriviu una seqüència d'operacions d'àlgebra relacional per obtenir els dni dels jugadors que han tingut alguna alineació en un partit on també ha estat alineat el jugador amb dni '999'. Tingueu en compte que en el resultat de la consulta no volem que surti el jugador '999'.

2. (2 punts)

2.1 Doneu la traducció a model relacional del diagrama de classes UML següent. Cal incloure: taules, atributs, claus primàries, claus foranes, i restriccions NOT NULL i UNIQUE derivades del diagrama.



2.2 Tenint en compte l'esquema de la base de dades que heu obtingut a l'apartat 2.1. Indiqueu si es pot implementar, en PostgreSQL, cadascuna de les restriccions textuais següents **com a restricció de columna o de taula**.

- En cas que es pugui, indica quina serà la restricció i en quina taula es definirà.
- En cas que no es pugui, explica perquè.

RI1 – Un usuari no es pot tenir com a amic a sí mateix.

RI2 – Un usuari només pot crear un màxim de 20 playlists.

RI3 – Les playlists favorites d'un usuari premium han de ser playlists que l'usuari tingui disponibles.

RI4 – No poden existir dues playlists amb el mateix nom i temàtica.

RI5 – El número de cançons d'una playlist ha de tenir valor i aquest valor ha de ser superior a 0.

2.3 Suposeu per aquest exercici una base de dades que té una taula amb els moviments realitzats amb les targetes de crèdit d'una entitat bancària. La taula `moviments` és propietat de l'usuari A. A continuació podeu veure l'esquema de la taula i un exemple del seu contingut.

`moviments(idTargeta, instant, botiga, import)`

'1111222244445555'	7777	'Zerka'	35
'2222333344445555'	8888	'Sous'	900
'2222333344445555'	9999	'Tnac'	330

- Escriu les sentències SQL necessàries per tal que l'usuari B pugui consultar i modificar únicament els moviments que tenen un import inferior a 1000.
- Un cop l'usuari B tingui l'accés que li heu donat a l'apartat anterior, justifiqueu si podrà o no executar la sentència SQL següent:

```

UPDATE moviments
SET import = import + 200
WHERE idTargeta = '2222333344445555' and instant = 8888;
  
```

3. (2 punts) Considereu una base de dades amb les taules següents:

```
departaments(idD, nomD, pressupost)
empleats(numE, nomE, sou, cap, idD)
    {cap} references empleats(numE)
    {idD} references departaments(idD)
edificis(nomEd, adreça)
assignacions(nomEd, idD, planta)
    {nomEd} references edificis(nomEd)
    {idD} references departaments(idD)
```

3.1 Supposeu que cadascuna de les regles que s'indiquen a continuació es vol implementar mitjançant triggers. Indiqueu i justifiqueu per cada regla quins triggers caldria definir, tenint en compte els criteris de qualitat establerts a l'assignatura. Per cada trigger cal indicar:

- esdeveniment que l'activa (cal indicar esdeveniment, taula i atributs rellevants)
- before/after
- row/statement

R1 – No hi pot haver cap empleat que tingui un sou superior al sou del seu cap. En cas que aquesta regla no es compleixi cal que salti una excepció.

R2 – El pressupost d'un departament s'ha de mantenir igual a la suma dels sous dels empleats del departament.

R3 – Només es pot afegir empleats si la suma dels pressupostos de tots els departaments és superior a 100.000. En cas que aquesta regla no es compleixi cal que salti una excepció.

3.2 A les taules `empleats` i `assignacions` es vol que quan s'esborri un departament s'apliquin les polítiques següents:

- taula `empleats`
FOREIGN KEY (idD) REFERENCES departaments(idD)
ON DELETE SET NULL
- taula `assignacions`
FOREIGN KEY (idD) REFERENCES departaments(idD)
ON DELETE CASCADE

Suposem que en comptes d'indicar aquestes polítiques al crear les taules, es vol implementar aquestes polítiques en el procediment emmagatzemat `tancaDepartament` que serveix per esborrar un departament que es passa com a paràmetre. A més, el procediment ha de llançar l'excepció "Departament no existeix" si el departament passat com a paràmetre no està a la base de dades. Completa la implementació del procediment seguint els criteris de qualitat establerts a l'assignatura.

```
CREATE FUNCTION tancaDepartament(prof departaments.idD%type)
RETURNS void AS $$
...
END;
$$LANGUAGE plpgsql;
```

4. (2 punts)

4.1 Disposeu de 3 operacions: R(A), RU(A) i W(A).

- Doneu un horari amb 2 transaccions que facin servir les operacions anteriors que necessiteu (una operació pot aparèixer més d'una vegada si és necessari), i on hi hagi una única interferència d'ACTUALITZACIÓ PERDUDA.
- Doneu la resta d'horaris possibles que presentin la mateixa interferència i que facin servir el mínim nombre d'operacions donades (igualment, una operació pot aparèixer més d'una vegada si és necessari).
- Justifica que l'horari que has proposat en l'apartat a) no té un horari serial equivalent.

4.2 Supposeu ara l'horari següent:

T1	T2	T3	T4
		OPX	
	R(A)		
			RU(B)
			RU(C)
RU(A)			
		R (F)	
	R(B)		
	R(C)		
W(A)			
R(A)			
		OPY	
			W(C)
			W(B)
			commit
commit			
	commit		
		commit	

- Sense tenir en compte OPX i OPY, doneu el graf de precedències corresponent a l'horari donat. En cas que l'horari tingui interferències, digues quines són (nom de la interferència, i transaccions i grànuls implicats).
- Tingueu en compte que OPX i OPY poden ser operacions simples (R, RU, W) o complexes (RU+W). Doneu tots els parells de valors de OPX i OPY que generin una interferència entre T1 i T3. Indiqueu quina interferència es produeix per a cada parell donat? Justifiqueu breument perquè es produeix cada interferència.
- Doneu l'horari amb les reserves i les esperes incorporades segons el nivell d'aïllament REPETEABLE READ. Per a aquest horari supposeu que OPX = R(A) i OPY = R(A).

5. (2 punts) Supposeu la taula logs següent, amb registres d'accés a un servei web:

IP	usuari	data	acció	durada
10.0.0.2	Joan	22/12/2018 11:30	accés	40
10.0.16.1	Carla	22/12/2018 11:40	compra	20
10.0.14.1	Jordi	22/12/2018 11:41	venda	22
10.0.1.2	Pere	23/12/2018 16:50	accés	10
10.0.0.4	Maria	23/12/2018 17:00	consulta	95
10.0.2.1	Carla	23/12/2018 17:14	venda	21
10.0.0.2	Pere	28/12/2018 09:00	compra	42
10.0.0.2	Pere	28/12/2018 09:01	compra	15
10.0.0.2	Pere	28/12/2018 09:02	compra	41
10.0.2.2	Joan	28/12/2018 20:00	consulta	32
10.0.3.2	Maria	30/12/2018 14:16	accés	33

5.1 Creeu un arbre B+ d'alçada 2 ($h=2$) i d'ordre 2 ($d=2$) sobre l'atribut durada, amb un 75% d'ocupació.

5.2 La taula logs ha crescut al llarg de l'any, i el servei web disposa ara de 300 usuaris únics, on cadascun d'ells ha fet en mitjana 50 accessos, 50 compres, 50 vendes, i 50 consultes. Les files de la taula es distribueixen uniformement entre 800 pàgines. Considereu que s'ha definit un índex agrupat multiatribut pels atributs acció,usuari d'ordre 100, ple al 75%. Per cadascuna de les consultes següents:

- Expliqueu com es resoldrà la consulta.
- Indiqueu el cost de la consulta en número de pàgines d'índex i número de pàgines de dades, mostrant els càlculs que feu.

a) `SELECT * FROM logs WHERE usuari = 'maria'`

b) `SELECT DISTINCT(usuari) FROM logs WHERE accio = 'venda'`

Temps: 3 h

Notes: 1 Juliol tarda - Revisió: 2 Juliol tarda

Cada pregunta en un full separat

1. (3 punts) Considereu l'esquema de la base de dades següent:

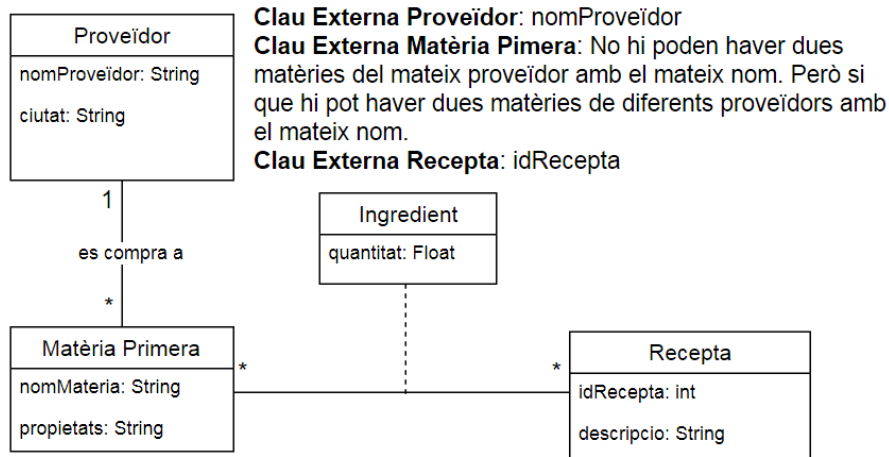
```
create table producte
(idProducte char(9),
nom char(20),
mida char(20),
preu integer check(preu>0),
primary key (idProducte),
unique (nom,mida));

create table domicili
(numTelf char(9),
nomCarrer char(20),
numCarrer integer check(numCarrer>0),
pis char(2),
porta char(2),
ciutat char(20),
primary key (numTelf));

create table comanda
(numComanda integer check(numComanda>0),
instantFeta integer not null check(instantFeta>0),
instantServida integer check(instantServida>0),
numTelf char(9),
import integer,
primary key (numComanda),
foreign key (numTelf) references domicili,
check (instantServida>instantFeta));

create table producteComprat
(numComanda integer,
idProducte char(9),
quantitat integer check(quantitat>0),
primary key(idProducte, numComanda),
foreign key (idProducte) references producte ,
foreign key (numComanda) references comanda);
// Hi ha una fila a la taula per cada producte comprat
// en una comanda.
```

- 1.1.** Doneu una sentència SQL per obtenir els productes comprats que no s'han demanat en comandes servides després de l'instant 333.
- 1.2.** Escriu una asserció en SQL que assegurí que l'import de les comandes sigui igual a la suma de l'import de cadascun dels productes comprats en la comanda. Cal tenir en compte que l'import d'un producte comprat en una comanda és el resultat de multiplicar el preu del producte per la quantitat del producte que s'ha comprat en la comanda.
- 1.3.** Escriu una seqüència d'operacions d'àlgebra relacional per obtenir les comandes en les que s'ha comprat un únic producte.
- 1.4.** Suposem el diagrama UML següent. Feu la traducció de UML a model relacional, indicant taules, atributs, claus primàries, claus foranes i restriccions de UNIQUE i NOT NULL que es deriven del diagrama.



2. (2.5 punts)

2.1. Donat un SGBD sense cap mecanisme de control de concurrència, suposem que es produeix l'horari següent (les accions s'han numerat per facilitar la seva referència).

- Dibuixeu el graf de precedències associat a l'horari.
- Es produeixen interferències? En cas de resposta negativa, argumenteu breument la vostra resposta. En cas de resposta positiva digueu quina/es interferències es produeixen (cal donar el nom de la interferència, grànul i transaccions implicades).
- Quins horaris serials hi són equivalents? Justificar la resposta.
- És recuperable aquest horari? Justificar la resposta.

#Acc	T1	T2	T3	T4
10		RU(B)		
20		W(B)		
30			R(D)	
40	RUA)			
50	R(D)			
60	RU(B)			
70	W(B)			
80			RU(E)	
90			W(E)	
100			RU(A)	
110			W(A)	
120			RU(F)	
130				RU(D)
140				W(D)
150	W(A)			
160			W(F)	
170	RU(C)			
180	W(C)			
190		RU(C)		
200		W(C)		
210				COMMIT
220		COMMIT		
230			COMMIT	
240	COMMIT			

2.2. Suposem que s'incorporen tècniques de reserva. Les transaccions estan definides com SET TRANSACTION ISOLATION LEVEL REPEATABLE READ.

- Donar l'horari resultant d'aplicar les reserves per aquest nivell d'aïllament. El nou horari ha d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions.
- En cas que en algun instant de l'horari no es pugui executar cap operació, no continueu i justifiqueu el motiu.
- Si hi ha alguna interferència en l'horari resultat digues quina/es és/són i entre quines transaccions es produeix/en. Altrament, digues quin és l'horari serial equivalent.

3. (2.5 punts) Supposeu la base de dades de l'exercici 1. Supposeu que hi ha definit el procediment emmagatzemat nouProducteComprat que s'invoca per cadascun dels productes comprats en una comanda, passant com a paràmetre el número de comanda, el producte comprat i la quantitat del producte comprat en la comanda. En la mateixa base de dades hi ha definit el disparador disparadorBDComandes.

```
11 create or replace function nouProducteComprat(numCom integer, numProd char(9), qtt int)
12 returns void language plpgsql as $$
13 begin
14 if((not exists(select * from comanda
15               where numComanda = numCom)) or
16    (not exists(select * from producte
17               where idproducte = numProd))) then
18     raise exception 'La comanda o el producte no existeixen';
19 elseif(exists(select * from producteComprat
20               where idproducte = numProd and numComanda = numCom)) then
21     raise exception 'El producte comprat ja es a la comanda';
22 else insert into productecomprat values(numCom,numProd,qtt);
23 end if;
24 return;
25 exception
26     when raise_exception then raise exception '%',SQLERRM;
27     when others then raise exception 'Error intern';
28 end; $$

29 create or replace function procDisparadorBDComandes()
30 returns trigger language plpgsql as $$
31 declare varNumComanda integer;
32         varImport integer;
33 begin
34 for varNumComanda in select numComanda from comanda
35 loop
36     varImport := (select sum(p.preu*pc.quantitat)
37                  from producte p, productecomprat pc
38                  where p.idproducte = pc.idProducte and
39                        pc.numcomanda = varNumComanda);
40     update comanda
41     set import = varImport
42     where numComanda = varNumComanda;
43 end loop;
44 return null;
45 end; $$

46 create trigger disparadorBDComandes
47 after insert on producteComprat
48 for each row execute procedure procDisparadorBDComandes();
```

- 3.1. Partint del contingut de la base de dades que s'indica a continuació. Digueu quin és el contingut de les taules de la base de dades després de l'execució de cadascuna de les seqüències de sentències següents:

- a) select * from nouProducteComprat(333,'300',1);
select * from nouProducteComprat(333,'200',3);
- b) select * from nouProducteComprat(222,'100',1);
select * from nouProducteComprat(222,'300',1);

taula comanda				
numComanda	instantFeta	instantServida	numTelf	import
111	1	10	123	8
222	11	20	123	12
333	21	30	123	0

taula producte			
idProducte	nom	mida	preu
100	dodotus	gran	5
200	kloonex	petit	3
300	tollolets	normal	6

taula domicili					
numTelf	nomCarrer	numCarrer	pis	porta	ciutat
123	Pont de Baix	24	1	2	Manlleu

taula producteComprat		
numComanda	idProducte	quantitat
111	100	1
111	200	1
222	300	2

3.2. Trobeu les sentències del procediment emmagatzemat que no compleixen els criteris de qualitat treballats a l'assignatura. Per cada sentència/es que no compleixin els criteris de qualitat:

- Indiqueu el número/s de la/es sentències que no compleixen el criteri
- Indiqueu quin és el criteri de qualitat que no compleixen. Justifiqueu perquè no el compleixen
- Doneu una nova versió del procediment emmagatzemat `nouProducteComprat` que compleixi el criteri

3.3. El tipus del disparador que s'ha usat en la implementació que us donem és el millor segons els criteris de qualitat treballats a l'assignatura, però la implementació del procediment `procDisparadorBDComandes` que activa el disparador no és la millor perquè és una implementació NO incremental. Doneu una implementació alternativa del procediment que faci el mateix però que sigui incremental.

4. (2 punts) Suposem la taula `comanda` de l'exercici 1.

`comanda(numComanda, instantFeta, instantServida, numTelf, import)`

Hi ha 5.000.000 files a la taula `comanda` distribuïdes uniformement en 50.000 pàgines de dades. Les pàgines tenen una mida de 4096 bytes. S'ha decidit crear dos índexs, arbres B+, sobre la taula.

- Un índex agrupat definit sobre l'atribut `numComanda`. L'atribut `numComanda` ocupa 6 bytes. Els apuntadors a nodes de l'índex ocupen 3 bytes. Els apuntadors a files de la taula ocupen 4 bytes.
- Un índex no agrupat definit sobre els atributs `numTelf`, `instantFeta`. L'ordre d de l'índex és 146. L'arbre està ple al 75%. L'arbre té 3 nivells. Les comandes de la base de dades s'han fet des de 20.000 telèfons diferents, i cada telèfon ha fet una mitjana de 250 comandes.

Mostra en detall com has fet els càlculs que es demanen a continuació:

4.1. La consulta següent no es veu afavorida per usar cap dels dos índexs definits. Calcula el cost (número de pàgines accedides) de resoldre-la.

- `select * from comanda where import = 50`

4.2. Índex per `numComanda`.

a) Calcula l'ordre d òptim de l'índex

b) Suposant l'índex té una ocupació del 80%, i suposant l'ordre que has trobat en l'apartat anterior:

- Calcula quants valors hi haurà a cada node de l'arbre.
- Calcula quantes pàgines ocuparan els nodes fulla de l'arbre.
- Calcula quants nivells tindrà l'arbre B+.

c) Suposant les dades calculades en els apartats anteriors, calcula el cost (número de pàgines accedides) de resoldre la consulta següent usant l'índex

- `select * from comanda where numComanda = 55555`

4.3. Índex per `numTelf`, `instantFeta`. Calcula el cost (número de pàgines accedides) de resoldre la consulta següent usant aquest índex.

- `select * from comanda where numTelf = '123'`

Temps: 3 h

Notes 21 gener tarda Revisió: 21 gener tarda

Cada pregunta en un full separat

1) (2.5 punts) Considereu l'esquema de la base de dades següent:

```
CREATE TABLE EquipsFutbol(  
    nomEquip char(30),  
    localitat char(50),  
    nomEstadi char(100) UNIQUE NOT NULL,  
    totalSalaris real not null check(totalSalaris>=0),  
    PRIMARY KEY(nomEquip));  
  
CREATE TABLE Partits(  
    nomEquipLocal char(30),  
    dataPartit date,  
    nomEquipVisitant char(30),  
    golsEquipL integer NOT NULL CHECK(golsEquipL >=0),  
    golsEquipV integer NOT NULL CHECK(golsEquipV >=0),  
    PRIMARY KEY (nomEquipLocal,dataPartit,nomEquipVisitant),  
    FOREIGN KEY (nomEquipLocal) REFERENCES EquipsFutbol,  
    FOREIGN KEY (nomEquipVisitant) REFERENCES EquipsFutbol,  
    CHECK(nomEquipLocal <> nomEquipVisitant));  
  
CREATE TABLE Jugadors(  
    dniJugador char(9),  
    nom char(50) NOT NULL,  
    nomEquip char(30) NOT NULL,  
    salari real not null check(salari>=0),  
    PRIMARY KEY (dniJugador),  
    FOREIGN KEY (nomEquip) REFERENCES EquipsFutbol);  
■ nomEquip és l'equip on juga el jugador  
  
CREATE TABLE Alineacions(  
    nomEquipLocal char(30),  
    dataPartit date,  
    nomEquipVisitant char(30),  
    dniJugador char(9),  
    gols integer NOT NULL CHECK(gols>=0),  
    numTargetes integer NOT NULL CHECK(numTargetes>=0),  
    PRIMARY KEY (nomEquipLocal,dataPartit,  
                 nomEquipVisitant, dniJugador),  
    FOREIGN KEY (nomEquipLocal,dataPartit,nomEquipVisitant)  
                 REFERENCES Partits,  
    FOREIGN KEY (dniJugador) REFERENCES Jugadors);  
■ el jugador dniJugador ha estat alineat al partit identificat  
  per nomEquipLocal, dataPartit, nomEquipVisitant  
■ els gols i targetes són les que han posat al jugador durant  
  el partit.
```

1.1) Escriviu una sentència SQL per obtenir els equips de futbol que han jugat, com a equip local, en partits contra més de 3 equips diferents i que, a la vegada, en aquests equips locals, no hi juga cap jugador que tingui targetes. En el resultat de la consulta hi ha d'haver el nom dels equips locals i la quantitat de partits que han jugat a casa amb equips diferents.

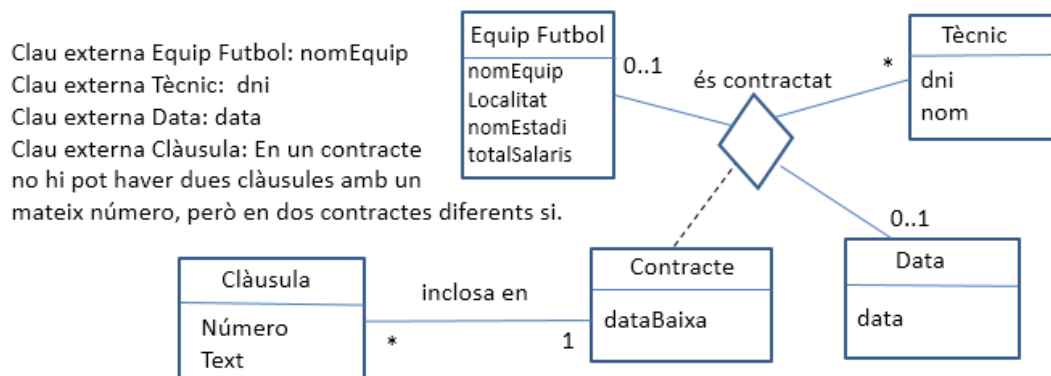
1.2) Vistes.

- Definiu una vista amb els partits empatats (és a dir, els que tant l'equip local com el visitant han fet el mateix número de gols). La vista ha de incloure tots els atributs de la taula *Partits* i la clàusula **WITH CHECK OPTION**.
- Doneu una sentència **update** de la vista, que inclogui la modificació de l'atribut *golsEquipL*, i que **NO** violi la restricció **WITH CHECK OPTION**.
- Doneu una sentència **update** de la vista, que inclogui la modificació de l'atribut *golsEquipL*, i que **SI** que violi la restricció **WITH CHECK OPTION**.

1.3) La cardinalitat de les taules de la base de dades és respectivament EF, P, J, A (amb EF,P,J,A > 0). Digueu quina serà la cardinalitat de la relació R per cadascuna de les seqüències d'operacions d'àlgebra relacional següents. En cas de no poder dir exactament quina serà dóna una cardinalitat mínima i una màxima. Justifiqueu la resposta.

- $R = \text{Partits} * \text{Alineacions}$
- $R = \text{Partits} \times \text{Alineacions}$
- $A = \text{Partits}[\text{golsEquipPL}]$
 $B = \text{Partits}[\text{golsEquipPV}]$
 $R = A \cup B$
- $A = \text{Alineacions}[\text{nomEquipLocal}, \text{dataPartit}, \text{nomEquipVisitant}]$
 $B = \text{Partits}[\text{nomEquipLocal}, \text{dataPartit}, \text{nomEquipVisitant}]$
 $R = A \cap B$

1.4) Supposeu el model conceptual en UML següent. Doneu el disseny lògic que s'obté fent la traducció a model relacional. Cal incloure, taules, atributs, claus primàries, claus forànies i restriccions **NOT NULL** i **UNIQUE** que siguin necessàries.



2) (2.5 punts) Supposeu la base de dades de l'exercici 1. Supposeu que a la base de dades s'afegeixen les taules següents. S'inclou a continuació **algunes** de les files que contenen les taules.

```
t_files(nomTaula, quantesFiles)
-- conté per cada taula quantes files hi ha a la taula,
-- en cas que no n'hi hagi cap contindrà un 0
EquipsFutbol 22
Jugadors     420

t_sentencies(nomSentencia, nomTaula, quantesSentencies)
-- conté per cada taula i per cada tipus de sentència, quantes vegades s'ha
-- executat una sentència del tipus sobre la taula
-- en cas que no n'hi hagi cap, contindrà un 0
insert      EquipsFutbol      2
insert      Jugadors          10
insert      Partits           6
delete      Partits           1
insert      Alineacions       80
update      Alineacions       10
```

2.1) Considereu les restriccions següents:

- Es mantingui correcte el valor de la columna *quantasFiles* de fila *Jugadors* de la taula *t_files*.
- Es mantingui correcte el valor de la columna *quantasSentencies* de les files on apareix *Jugadors* a la taula *t_sentencies*.
- Es mantingui correcte la columna *totalSalaris* de la taula *EquipsFutbol*, que ha de correspondre a la suma dels salaris dels jugadors de l'equip.

Indiqueu quins són els triggers que cal definir sobre la taula *Jugadors*. Per cada trigger cal indicar:

- Esdeveniment (Insert, Delete, Update – en cas d'update indicar atribut/s rellevants),
- Before/After
- Row/Statement

2.2) Explicar breument el què caldrà que facin els procediments a definir per al/s trigger/s *delete*. En cas de que no hagin estat necessaris triggers *delete*, explicar perquè no fan falta.

2.3) Implementar els procediments necessaris per al/s trigger/s *update*. En cas de que no hagin estat necessaris triggers *update*, explicar perquè no fan falta.

3) (2.5 punts) S'ha creat una taula *empleats*(numEmpl, *nom*, *sou*, *ciutat*, *categoria*). La taula *empleats* té 100.000 tuples i els valors de *numEmpl* estan repartits uniformement entre 1 i 100.000. El factor de bloqueig del fitxer de dades és de 20 files per pàgina. Es defineixen dos índexs sobre la taula *empleats*:

- un índex B+ agrupat per l'atribut *numEmpl*, amb ordre d=60 i amb nodes plens al 80% de la seva capacitat.
- un índex B+ no agrupat per l'atribut *sou*, amb ordre d=80 i amb nodes plens al 75% de la seva capacitat.

3.1) Expliqueu quina diferència hi ha entre un índex agrupat i un índex no agrupat.

3.2) Digueu quants índexs agrupats hi pot haver definits com a màxim sobre una taula. Justifiqueu la resposta.

3.3) Es vol obtenir totes les files que compleixen les condicions *numEmpl*>95.000 AND *sou*>2.000. Sabent que hi ha 400 files que compleixen *sou*>2.000, i hi ha 100 files que compleixen les dues condicions. Mostreu com calculeu el cost de la consulta en cadascun dels casos següents.

- Emprant recorregut seqüencial de la taula *empleats*.
- Emprant l'índex B+ agrupat per *numEmpl*.
- Emprant l'índex B+ no agrupat per *sou*.

3.4) Digueu si hi ha algun índex que es pogués afegir a la taula *empleats* i que millorés l'eficiència de la consulta. Justifiqueu la resposta en funció dels costos.

4) (2.5 punts) Donades les dues taules següents:

```
Clients(numClient, nom, instantNaixement, població, saldoTotal)
Comptes(numCompte, titular, vigent, saldo)
{titular} referencia Clients
```

El contingut en un cert instant de les taules és:

numClient	nom	instantNaixement	poblacio	saldoTotal
1	Anna Puig	422	Barcelona	28.900
2	Rosa Carrasco	145	Barcelona	33.250

numCompte	titular	vigent	saldo
11	1	N	28.000
21	1	S	-1.500
31	1	S	2.400
42	2	S	13.250
52	2	N	20.000

- 4.1) Supposeu que el grànul és la fila, que no existeix cap mecanisme per al control de la concurrència i que es produeix l'horari següent (les operacions s'han numerat per facilitar la seva referència).

#op	T1	T2	T3
1	select saldoTotal from clients where numClient =1		
2		update clients set saldoTotal= saldoTotal * 1.1 where numClient = 1	
3			select saldoTotal from clients where numClient = 1
4	select max(saldoTotal) from clients		
5			select * from comptes where titular = 2
6		update comptes set titular = 2 where titular = 1 and vigent = 'N'	
7			select saldo from comptes where titular =2
8	insert into clients values(3, 'Marta López', 323, 'Barcelona', 50.000)		
9			commit
10		Commit	
11	commit		

- Digueu si existeixen interferències en l'horari. Si la resposta és afirmativa, doneu el nom de la/es interferència/es, grànul i transaccions implicades.
 - Digueu quins horaris serials donen resultats equivalents a l'horari proposat.
 - Indiqueu, en cas que hi hagi interferències, quin és el nivell mínim d'aïllament necessari per evitar cadascuna d'elles.
 - Digueu si l'horari proposat és recuperable. Justifiqueu breument la resposta.
- 4.2) Supposeu que el grànul és la fila, que tenim un mecanisme per al control de la concurrència basat en reserves S, X i que la transacció T1 treballa a nivell d'aïllament de READ COMMITTED i T2 i T3 a nivell SERIALITZABLE.
- Doneu l'horari que ha d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions. Quan més d'una transacció espera per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out).
 - Indiqueu els horaris serials equivalents, en cas que n'hi hagi. Si no n'hi ha, justifiqueu la resposta.

Temps: 90 minuts**Notes 23 novembre****Cada pregunta s'ha de lliurar en un full separat**

Suposeu una base de dades amb les taules següents:

```
CREATE TABLE EquipsFutbol(  
  nomEquip char(30),  
  localitat char(50),  
  nomEstadi char(100) UNIQUE NOT NULL,  
  pressupost real NOT NULL check(pressupost>=0),  
  PRIMARY KEY(nomEquip));
```

```
CREATE TABLE Partits(  
  idPartit integer,  
  nomEquipLocal char(30) NOT NULL,  
  dataPartit date NOT NULL,  
  nomEquipVisitant char(30) NOT NULL,  
  golsEquipL integer NOT NULL CHECK(golsEquipL >=0),  
  golsEquipV integer NOT NULL CHECK(golsEquipV >=0),  
  PRIMARY KEY (idPartit),  
  FOREIGN KEY (nomEquipLocal) REFERENCES EquipsFutbol,  
  FOREIGN KEY (nomEquipVisitant) REFERENCES EquipsFutbol,  
  UNIQUE(nomEquipLocal,nomEquipVisitant));
```

```
CREATE TABLE Jugadors(  
  dniJugador char(9),  
  nom char(50) NOT NULL,  
  nomEquip char(30) NOT NULL,  
  salari real NOT NULL check(salari>=0),  
  PRIMARY KEY (dniJugador),  
  FOREIGN KEY (nomEquip) REFERENCES EquipsFutbol);  
■ nomEquip és l'equip on està contractat el jugador
```

```
CREATE TABLE Alineacions(  
  idPartit integer,  
  dniJugador char(9),  
  gols integer NOT NULL CHECK(gols>=0),  
  numTargetes integer NOT NULL CHECK(numTargetes>=0),  
  PRIMARY KEY (idPartit, dniJugador),  
  FOREIGN KEY (idPartit) REFERENCES Partits,  
  FOREIGN KEY (dniJugador) REFERENCES Jugadors);  
■ el jugador dniJugador ha estat alineat al partit  
  identificat per idPartit  
■ els gols i targetes són les que han posat al jugador  
  durant el partit.
```


1. (1 punt) Per cadascuna de les restriccions d'integritat següents.
- En cas que es pugui implementar com a restricció d'integritat de columna o de taula, en una sentència CREATE: Digueu la taula on s'haurien de definir, i la restricció d'integritat de columna o de taula en SQL.
 - Si no es pot: Expliqueu breument perquè no es pot.

- 1.1 Un equip de futbol no pot jugar un partit contra ell mateix.
- 1.2 El número de gols d'un equip local en un partit ha de ser igual a la suma dels gols dels seus jugadors que han sigut alineats al partit.
- 1.3 Un equip de futbol no pot jugar dos partits en una mateixa data.

2. (3 punts) Doneu una sentència SQL per obtenir els equips de futbol (nomEquip) que no han jugat mai com a equip visitant en un partit empatat. A més es vol que un equip només surti si el salari total dels jugadors contractats a l'equip és més gran que la mitjana del pressupost de tots els equips.

3. (1.5 punt) Doneu una seqüència d'operacions en àlgebra relacional per obtenir els jugadors alineats en l'equip local que ha jugat el partit amb idPartit 333. Per cada jugador cal que surti el DNIIugador, el nom del jugador i el seu nom d'equip.

4. (1 punt) Doneu una sentència assertion per assegurar que en un partit no s'ha alineat més de 30 jugadors.

5. (1 punt) Supposeu la vista següent:

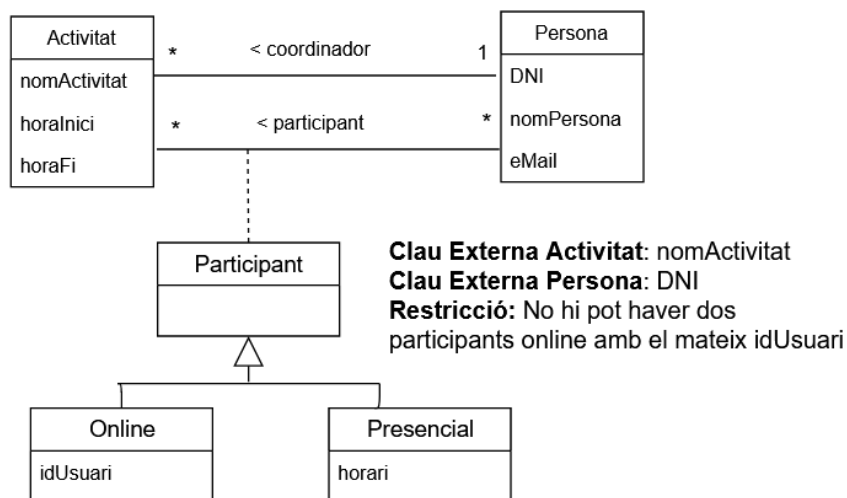
```
CREATE VIEW massaCar (nomEquip, pressupost, costJugadors) AS
SELECT e.nomEquip, e.pressupost, sum(j.salari)
FROM EquipsFutbol e NATURAL INNER JOIN Jugadors j
WHERE e.localitat in ('Barcelona', 'Madrid')
GROUP BY e.nomEquip;
```

- 5.1 Doneu un contingut de les taules de la base de dades que tingui com a mínim 3 files a la taula jugadors i que faci que quan es consulti la vista el resultat sigui el següent:

massaCar	nomEquip	pressupost	costJugadors
	Barça	50M	70M

- 5.2 Justifiqueu si la vista massaCar és o no actualitzable.

6. (1 punt) Supposeu que el propietari de les taules de la base de dades és M.
- 6.1 Doneu el diagrama d'autoritzacions, després de que s'executin les sentències següents:
M: GRANT update ON partits TO A WITH GRANT OPTION;
A: GRANT update ON partits TO B WITH GRANT OPTION;
B: GRANT update ON partits TO C;
- 6.2 Supposeu la situació donada pel diagrama d'autoritzacions anterior. Doneu una sentència UPDATE de la taula partits per la qual l'usuari A no tingui suficients permisos.
- 6.3 Supposeu la situació donada pel diagrama d'autoritzacions anterior. Doneu el nou diagrama d'autoritzacions un cop executada la sentència següent. Justifiqueu breument la resposta.
M: REVOKE GRANT OPTION FOR update FROM partits TO A RESTRICT;
7. (1.5 punts) Supposeu el model conceptual en UML següent. Doneu el disseny lògic que s'obté fent la traducció a model relacional. Cal incloure, taules, atributs, claus primàries, claus foranies i restriccions NOT NULL i UNIQUE que siguin necessàries.



Temps: 2 hores

Notes 27 gener Revisió: 28 gener

Cada pregunta s'ha de lliurar en un full separat**1. (1 punt) Donades les tres taules següents:**

```
professors(dni, nomProf, telefon, sou)
despatxos(modul, numero, superficie)
assignacions(dni, modul, numero, instantInici, instantFi)
    {dni} referencia professors
    {modul,numero} referencia despatxos
```

Considereu el següent fragment de codi Java que implementa, utilitzant JDBC, l'assignació de tots els professors que tenen un sou superior al valor de la variable sou al despatx identificat per les variables modul i numero, amb instantInici = 100 i instantFi = null. En cas que algun dels professors ja estigui assignat al despatx s'acaba el programa mostrant un missatge d'error.

```
try {
    /* Tenim ja una connexió c establerta amb la base de dades */
    int sou = 1200;
    String modul = "omega";
    String numero = "118";

    Statement st1 = c.createStatement();
    ResultSet rs1 = st1.executeQuery("select dni from professors where sou < " +
sou);

    while (rs1.next()) {
        String dniProf = rs1.getString(1);

        // Consultem si el professor ja ha estat assignat
        Statement st2 = c.createStatement();
        ResultSet rs2 = st2.executeQuery("select * from assignacions"
            + " where dni='" + dniProf + "'"
            + " and modul='" + modul + "'"
            + " and numero='" + numero + "'");

        // Si rs2 retorna resultat, aleshores el professor ja està assignat
        if (rs2.next()) {
            System.out.println("Error: Un dels professors ja està assignat");
            c.rollback();
            break; /* surt del bucle */
        }
        // En cas contrari, procedim a crear l'assignació
        else {
            Statement st3 = c.createStatement();
            st3.executeUpdate("insert into assignacions values ("
                + "'" + dniProf + "',"
                + "'" + modul + "',"
                + "'" + numero + "',"
                + "100, null)");

            st3.close();
        }
        rs2.close(); st2.close();
    }
    st1.close(); rs1.close(); c.commit(); c.close();
}
catch (ClassNotFoundException ce) {
    System.out.println ("Error al carregar el driver");
}
catch (SQLException se) {
    System.out.println ("Excepcio: " + se.getMessage());
}
```

Donat aquest fragment de codi, identifiqueu quins criteris de qualitat no es compleixen. Per cadascun d'ells, expliqueu perquè no es compleix, i expliqueu amb detall quins canvis serien necessaris en la codificació per tal de satisfer aquests criteris.

2. (2 punts) Considereu una base de dades amb les taules següents:

```
cantants (nom, pais, anyRetir)
albums (titolAlbum, nomCantant, anyPublicacio, preu)
        {nomCantant} referencia cantants
critiques (titolAlbum, nomCantant, font, puntuacio)
        {titolAlbum, nomCantant} referencia albums
estadístiques (nomCantant, millorPuntuació)
        {nomCantant} referencia cantants
```

Indiqueu quins són els triggers necessaris (tenint en compte els criteris de qualitat establerts a l'assignatura) a definir per tal que:

- **RI1.** Es generi una excepció en cas que un esdeveniment faci que no es compleixi la següent restricció: Un cantant no pot publicar nous àlbums en els anys posteriors a l'any de retir.
- **RI2.** Es mantingui el valor de l'atribut *millorPuntuació*. Aquest atribut ha de tenir com a valor, la puntuació més alta que ha rebut un cantant en els àlbums que ha publicat.
- **RI3.** Es generi una excepció en cas que un esdeveniment faci que no es compleixi la següent restricció: No es poden afegir crítiques si la base de dades conté menys de 10 cantants.

Per cada trigger cal indicar:

- Taula per a la que es defineix
- Esdeveniment que l'activa (INSERT, DELETE, UPDATE - en cas d'UPDATE indicar atribut/s rellevants)
- BEFORE/AFTER (justificar la resposta)
- ROW/STATEMENT (justificar la resposta)

3. (2 punts) Supposeu la base de dades següent, que s'ha creat en un SGBD sense cap mecanisme de control de concurrència. Considereu que el grànul és la fila.

```
autors (  codiAutor,    nom,          telefon,    poblacio)
         ca1          Autor1      111        Pobl1
         ca2          Autor2      222        Pobl2

obres (  idObra,      codiAutor,   titol,      anyEdicio)
         ob1          ca1         Bon dia     2020
         ob2          ca2         Hola        2020
         ob3          ca1         Bona nit    2020
```

Suposeu també les sentències SQL següents:

s1	UPDATE obres SET codiAutor = 'ca1' WHERE idObra= 'ob2';
s2	DELETE FROM obres WHERE idObra='ob1';
s3	INSERT INTO obres VALUES ('ob4', 'ca1', 'Adeu', 2020);
s4	SELECT * FROM obres WHERE codiAutor='ca1';
sX és un codi que us pot permetre fer referència a les sentències	

- 3.1** Doneu un horari on intervinguin dues transaccions, que poden executar una o més de les sentències SQL donades, i que presenti una interferència d'ACTUALITZACIÓ PERDUDA. Una sentència es pot usar únicament una vegada en l'horari, i s'ha d'incloure a l'horari el mínim número de sentències. En cas que no sigui possible formar un horari amb els requisits indicats, justifiqueu perquè no.
- 3.2** Doneu un horari on intervinguin dues transaccions, que poden executar una o més de les sentències SQL donades, i que presenti una interferència de FANTASMA. S'ha d'incloure a l'horari el mínim número de sentències, però una sentència es pot usar més d'una vegada. En cas que no sigui possible formar un horari amb els requisits indicats, justifiqueu perquè no.

4. (2 punts) Considereu l'horari següent:

	T1	T2
1	R(A)	
2		RU(A)
3		W(A)
4		RU(B)
5		W(B)
6	RU(B)	
7	W(B)	
8	COMMIT	
9		COMMIT

Suposeu que s'incorporen tècniques de control de concurrència amb reserves.

- 4.1** En cas que les transaccions treballin en nivell d'aïllament READ UNCOMMITTED.
- Doneu l'horari aplicant reserves corresponents al nivell d'aïllament (*)
 - Doneu el graf de precedències de l'horari resultant d'aplicar reserves
 - En cas que hi hagi alguna interferència en l'horari resultant digueu quina/es són, i el/s grànul/s implicat/s.

4.2 En cas que les transaccions treballin en nivell d'aïllament REPETEABLE READ.

- a) Doneu l'horari aplicant reserves corresponents al nivell d'aïllament (*)
- b) Doneu el graf de precedències de l'horari resultant d'aplicar reserves
- c) En cas que hi hagi alguna interferència en l'horari resultant digueu quina/es són, i el/s grànul/s implicat/s.

(*) En els horaris heu d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions. Quan més d'una transacció espera per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out).

5. (2 punts) Supposeu la taula *albums*:

`albums (titolAlbum, nomCantant, anyPublicacio, preu)`

Aquesta taula té 2.000.000 files, i el factor de bloqueig, de les pàgines de dades on s'emmagatzemen aquestes files, és de 40 files per pàgina.

Sabem que el cantant 'Juli Catedrals' ha fet 500 àlbums. També sabem que la condició `titolAlbum > 'xxx'` la compleixen 8.000 àlbums que han fet uns 1.000 cantants diferents.

Considereu la consulta següent.

```
SELECT *  
FROM albums  
WHERE nomcantant = 'Juli Catedrals'  
AND titolAlbum > 'xxx';
```

Doneu el cost en número de pàgines d'índex i número de pàgines de dades a les que cal accedir per resoldre la consulta:

5.1 Usant un índex agrupat definit sobre l'atribut nomCantant, amb ordre d=24, ple al 80%.

5.2 Usant un índex no agrupat definit sobre l'atribut titolAlbum, amb ordre d=16, ple al 70%

6. (1 punt) Considereu un índex arbre B+ amb els següents paràmetres: el valor pel que es crea l'índex té V bytes, la mida dels nodes és B bytes, els RID tenen T bytes, i els apuntadors a nodes de l'índex tenen A bytes.

6.1 Sigui nf la quantitat màxima de valors que un node del nivell fulla pot contenir. Expressau nf en funció dels paràmetres V, B, T, A

6.2 Suposa ara que $2d = 100$.

- a) Quin és el número màxim de files que el arbre B+ pot apuntar si té 3 nivells?
- b) Quants nodes es necessiten per emmagatzemar l'índex en aquest cas?

Suposeu una base de dades amb les taules següents:

```
create table departaments(  
  codiDept integer,  
  nomDept char(50) unique not null,  
  pressupost integer not null check(pressupost>=0),  
  primary key (codiDept));  
  
create table professors(  
  idProf integer,  
  nomProf char(50) not null,  
  codiDept integer not null,  
  sou real,  
  primary key (idProf),  
  foreign key (codiDept) references departaments);  
-- codiDept és del departament en el que treballa el professor  
  
create table assignatures(  
  idAssig char(5),  
  nomAssig char(50) not null,  
  codiDept integer not null,  
  profResponsable integer not null,  
  primary key (idAssig),  
  foreign key (codiDept) references departaments,  
  foreign key (profResponsable) references professors);  
-- codiDept és del departament al que està assignada l'assignatura  
-- profResponsable és el professor responsable de l'assignatura  
  
create table assignacions(  
  idProf integer,  
  idAssig char(5),  
  quadrimestre char(6),  
  horesAssig integer not null check (horesAssig>0),  
  primary key (idProf, idAssig, quadrimestre),  
  foreign key (idProf) references professors,  
  foreign key (idAssig) references assignatures);  
-- hi ha una fila si el professor idProf ha donat o dona  
-- classes de l'assignatura idAssig en el quadrimestre
```

- 1. (1.75 punts)** Considereu la base de dades donada a l'inici de l'examen. Doneu una sentència SQL per obtenir les assignatures (nomAssig) per a les que no hi ha cap assignació d'un professor de més de 3 hores (horesAssig) en el quadrimestre '2020-2'.

2. (1.75 punt) Considereu la base de dades donada a l'inici de l'examen. Doneu una seqüència d'operacions d'àlgebra relacional per obtenir l'identificador i el nom dels professors que han estat o estan assignats amb menys de 5 hores de docència (horesAssig) a assignatures d'un departament que no és el seu.

3. (3 punts) Considereu la base de dades donada a l'inici de l'examen.

3.1 Per cadascuna de les restriccions següents:

- Si es pot implementar com a restricció en la definició de les taules, doneu el codi SQL que cal afegir als CREATE TABLE de la BD per a implementar-la, i en quina taula s'ha d'afegir.
- Si no es pot, explicar per què.
 - a) Un professor no pot ser responsable de més d'una assignatura.
 - b) No s'ha de poder executar sentències UPDATE de l'atribut pressupost de la taula departaments.
 - c) Un professor ha de tenir sou positiu, a no ser que estigui de baixa, cosa que s'indicarà quan l'atribut sou tingui valor NULL.

3.2 Donar una sentència de creació d'una asserció per tal que a la base de dades es compleixi que: "Tots els professors assignats a una assignatura en el mateix quadrimestre, han de ser del mateix departament."

4. (2 punts)

4.1 Considereu la vista següent definida sobre la base de dades donada a l'inici de l'examen.

- En cas que la vista sigui actualitzable, doneu una sentència INSERT/DELETE/UPDATE que doni una excepció per al check de la vista.
- En cas que la vista no sigui actualitzable, expliqueu el motiu pel qual no ho és.

```
create view professorsDept5 as
  select p.idProf, p.nomProf
  from professors p
  where p.codiDept = 5
  with check option;
```

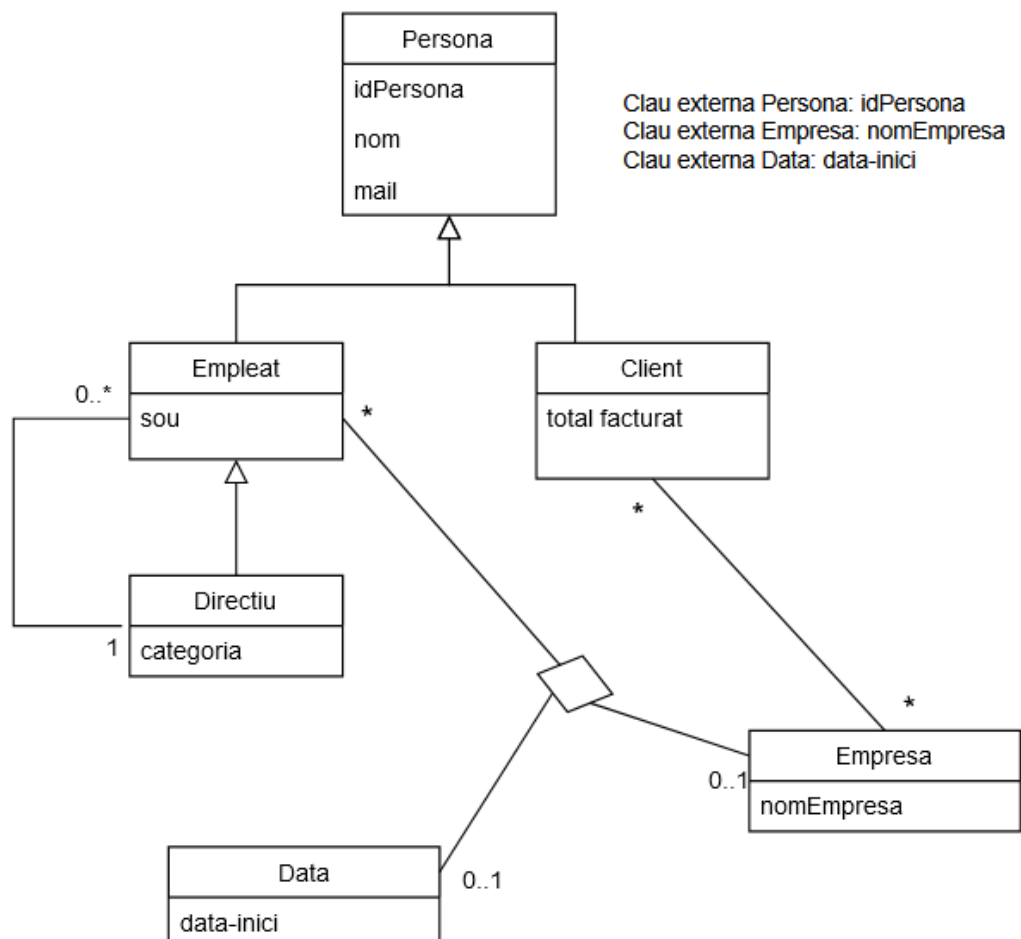

4.2 Suposem que les taules de la base de dades són propietat de la Meri. Considereu les sentències GRANT i REVOKE següents. Doneu els diagrames d'autoritzacions corresponents a l'instant abans d'executar les sentències REVOKE, i a l'instant després d'executar les sentències REVOKE. Justifica el diagrama d'autoritzacions de després dels REVOKE.

```

Meri: GRANT select ON professors TO Anna WITH GRANT OPTION;
Meri: GRANT update ON professors TO Cata WITH GRANT OPTION;
Meri: GRANT select ON professors TO Lola WITH GRANT OPTION;
Anna: GRANT select ON professors TO Laia;
Cata: GRANT update ON professors TO Laia;
Lola: GRANT select ON professors TO Laia;
Meri: REVOKE select ON professors FROM Anna RESTRICT;
Meri: REVOKE update ON professors FROM Cata RESTRICT;

```

5. (1.5 punts) Suposeu el model conceptual en UML següent. Doneu el disseny lògic que s'obté fent la traducció a model relacional. Cal incloure, taules, atributs, claus primàries, claus foranes i restriccions NOT NULL i UNIQUE que siguin necessàries.



Temps: 2,5 hores

Notes 22 juny Revisió: 28 juny

Cada pregunta s'ha de lliurar en un full separat**1. (1 punt)** Donades les 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 <=25),
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,
check (instantInici < instantFi));
```

Considereu el següent fragment de codi Java/JDBC. Implementeu el cos del **while** per tal que s'incrementi en 5 unitats la superfície dels despatxos de cadascun dels mòduls obtinguts en la variable varModul. Definiu també els Statements o PreparedStatements necessaris on creieu convenient. Cal que el programa tregui un missatge d'excepció si la superfície d'algun despatx passa a ser superior a 25. En cas que no hi hagi cap excepció el programa indicarà el número de despatxos modificats de cada mòdul.

```
try {
    /* Tenim ja una connexió c establerta amb la base de dades */
    /* Tenim un variable in que permet llegir el que escriu l'usuari */

    System.out.print("Escriu el nom d'un mòdul: ");
    String varModul = in.nextLine();

    while (!varModul.equals("acabar")) {

        //codi a implementar

        System.out.print("Escriu el nom d'un mòdul: ");
        varModul = in.nextLine();
    }
    c.commit();
}
catch (ClassNotFoundException ce) {
    System.out.println ("Error al carregar el driver");}
catch (SQLException se) {
    System.out.println ("Excepcio: "+ se.getMessage());}
```

Recordeu que podeu usar els mètodes/codis d'excepció de JDBC estudiats a classe:

Statements

- `Statement createStatement();`
- `ResultSet executeQuery(String sql);`
- `int executeUpdate(String sql);`

PreparedStatement

- `PreparedStatement prepareStatement(String sql);`
- `ResultSet executeQuery();`
- `int executeUpdate();`
- `void setXXX(int posicioParametre, XXX valor);`

ResultSet

- `boolean next();`
- `XXX getXXX(String nomColumna)`

SQLException

- `// 23502 -not_null_violation, 23503 -foreign_key_violation`
- `// 23505 -unique_violation, 23514 -check_violation`
- `String getSQLState();`

2. (1 punt) Supposeu la base de dades següent:

```
empleats1 (numEmp1, nomEmp1, ciutatEmp1);
empleats2 (numEmp2, nomEmp2, ciutatEmp2);
```

Suposeu que es vol implementar la restricció següent mitjançant disparadors:

Els valors de l'atribut ciutatEmp1 de la taula empleats1 han d'estar inclosos en els valors de l'atribut ciutatEmp2 de la taula empleats2

La idea és llançar una excepció en cas que s'intenti executar una sentència que violi la restricció.

2.1 Digueu quins són els esdeveniments rellevants (taula/es i esdeveniment/s).

2.2 Digueu i justifiqueu de quin tipus han de ser els disparadors (ROW / STATEMENT, BEFORE/AFTER).

2.3 Implementeu un dels disparadors que hagueu identificat per a la taula empleats1. En cas de que no n'hi hagi cap justifiqueu perquè. Podeu fer servir la plantilla següent.

```
CREATE FUNCTION comprovarCiutat() RETURNS trigger AS $$
BEGIN
    //implementació
END;
$$LANGUAGE plpgsql;

CREATE TRIGGER ex3_3
BEFORE/AFTER <esdeveniment>
FOR EACH ROW/STATEMENT EXECUTE PROCEDURE comprovarCiutat();
```

3. (2 punt) Considereu la base de dades de l'exercici 1 amb el contingut següent:

Professors	<u>dni</u>	nomProf	telefon	sou
	111	Joana	97743121	10000
	222	Martí	93818903	3000
	333	Lourdes	97261201	50000

Despatxos	<u>modul</u>	<u>numero</u>	superficie
	C6	101	13
	Omega	300	8
	Omega	301	8

Assignacions	<u>dni</u>	<u>modul</u>	<u>numero</u>	<u>instantIni</u>	<u>instantFi</u>
	333	C6	101	5	9
	111	C6	101	3	NULL
	222	Omega	300	10	NULL

Considereu també el procediment emmagatzemat següent:

```
CREATE TYPE despatx AS (d_modul char(5), d_numero char(5));

CREATE FUNCTION assignaProfessor(dni_p char(50), modul_d char(5),
                                numero_d char(5), inici_a integer)
    RETURNS SETOF despatx AS $$
DECLARE desp despatx;
BEGIN
    IF (NOT EXISTS(SELECT * FROM Professors WHERE dni=dni_p)
        OR NOT EXISTS(SELECT * FROM Despatxos
                        WHERE modul=modul_d AND numero=numero_d)) THEN
        RAISE EXCEPTION 'El professor o el despatx no existeix';
    ELSIF (EXISTS (SELECT * FROM Assignacions a
                   WHERE a.dni=dni_p AND a.instantFi IS NULL)) THEN
        UPDATE Assignacions SET instantFi = inici_a - 1
        WHERE dni=dni_p AND instantFi IS NULL;
    END IF;
    INSERT INTO Assignacions VALUES (dni_p,modul_d,numero_d,inici_a,NULL);
    FOR desp IN SELECT d.modul,d.numero
                FROM Professors p NATURAL JOIN Assignacions a
                NATURAL JOIN Despatxos d
                WHERE a.instantFi IS NULL
                GROUP BY d.modul, d.numero
                HAVING COUNT(*) >= 2
    LOOP
        RETURN NEXT desp;
    END LOOP;
EXCEPTION
    WHEN raise_exception THEN raise exception '%',sqlerrm;
END;
$$LANGUAGE plpgsql;
```

- 3.1** Indiqueu i justifiqueu quin és l'estat de la base de dades i el valor de retorn després de l'execució de la consulta: `SELECT * FROM assignaProfessor('111','Omega','300',15);`
- 3.2** Digueu quins són els criteris de qualitat que no es compleixen en la implementació del procediment donat i com caldria canviar el codi per tal que es complissin.

4. (3 punts) Transaccions:

4.1 Supposeu ara l'horari següent.

- Doneu el graf de precedències per aquest horari.
- En cas que hi hagi alguna interferència en l'horari resultant digueu quina/es són, les transaccions i el/s grànul/s implicat/s.
- És serialitzable? En cas afirmatiu doneu el horari serial equivalent. En cas negatiu perquè no.

	T1	T2	T3	T4
10			R(B)	
20		RU(A)		
30	RU(A)			
40		W(A)		
50		R(C)		
60		R(F)		
70	W(A)			
80				R(A)
90		RU(B)		
100	R(C)			
110				RU(C)
120		W(B)		
130			R(A)	
140				W(C)
150	commit			
160		commit		
170			commit	
180				commit

- 4.2 En cas que les transaccions treballin en nivell d'aïllament READ COMMITTED. Doneu l'horari un cop aplicades les reserves corresponents al nivell d'aïllament. En l'horari heu d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions. Quan més d'una transacció espera per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out).

- 4.3 Considereu ara un SGBD sense cap mecanisme de control de concurrència, on el grànul és la fila. I considereu l'horari següent que té una interferència d'anàlisi inconsistent. Expliqueu la interferència d'anàlisi inconsistent en base a l'horari i al contingut de la base de dades donat en l'exercici 3.

	T1	T2
1	select * from despatxos where modul='Omega' and numero=300;	
2		update despatxos set superficie = superficie + 8 where modul='Omega' and numero=300;
3		select * from despatxos where modul='Omega' and numero=301;
4	update despatxos set superficie = superficie + 5 where modul='Omega' and numero=301;	
5	Commit	
6		Commit

- 5. (3 punt)** Considereu la taula R(a,b,c,d). Aquesta taula té 100.000 files, i el factor de bloqueig, de les pàgines de dades on s'emmagatzemen aquestes files, és de 10 files per pàgina.

Considereu la consulta següent.

```
SELECT *  
FROM R  
WHERE b=5 and c >=50
```

Se sap que hi ha 70 files que compleixen la condició $b=5$, 100 files la condició $c \geq 50$ i només 1 fila les dues condicions alhora.

IMPORTANT: En tots els apartats cal detallar tots els càlculs que feu per calcular els costos.

- 5.1** Suposant que només podem definir un únic índex, arbre B+, per un únic atribut, i que aquest índex tindrà una ocupació del 80% i un ordre $d=50$, Quin tipus d'índex (agrupat o no agrupat) escolliríeu i per quin atribut hauria d'estar definit per tal de fer la consulta el més eficient possible? Justifiqueu les respostes en base els costos de totes les opcions possibles.
- 5.2** Suposant ara que en lloc d'un únic índex per un únic atribut, en podem definir 2 també per un únic atribut i del mateix tipus que abans (una ocupació del 80% i un ordre $d=50$) quin seria el cost fent servir l'estratègia de intersecció de RIDs?
- 5.3** Suposant ara que podem definir un índex no agrupat multiatribut pels atributs b, c. Quin seria el cost fent servir aquest índex en cas de tenir ordre $d=25$ i ocupació 80%?

Temps: 90 minuts

Notes 25 novembre

Cada pregunta s'ha de lliurar en un full separat

Suposeu una base de dades amb les taules següents:

```
CREATE TABLE EquipsFutbol(  
  nomEquip char(30),  
  localitat char(50),  
  nomEstadi char(100) UNIQUE NOT NULL,  
  pressupost real NOT NULL check(pressupost>=0),  
  PRIMARY KEY(nomEquip));
```

```
CREATE TABLE Partits(  
  idPartit integer,  
  nomEquipLocal char(30) NOT NULL,  
  dataPartit date NOT NULL,  
  nomEquipVisitant char(30) NOT NULL,  
  golsEquipL integer NOT NULL CHECK(golsEquipL >=0),  
  golsEquipV integer NOT NULL CHECK(golsEquipV >=0),  
  PRIMARY KEY (idPartit),  
  FOREIGN KEY (nomEquipLocal) REFERENCES EquipsFutbol,  
  FOREIGN KEY (nomEquipVisitant) REFERENCES EquipsFutbol,  
  UNIQUE(nomEquipLocal,nomEquipVisitant));
```

```
CREATE TABLE Jugadors(  
  dniJugador char(9),  
  nom char(50) NOT NULL,  
  nomEquip char(30) NOT NULL,  
  salari real NOT NULL check(salari>=0),  
  PRIMARY KEY (dniJugador),  
  FOREIGN KEY (nomEquip) REFERENCES EquipsFutbol);  
■ nomEquip és l'equip on està contractat el jugador
```

```
CREATE TABLE Alineacions(  
  idPartit integer,  
  dniJugador char(9),  
  gols integer NOT NULL CHECK(gols>=0),  
  numTargetes integer NOT NULL CHECK(numTargetes>=0),  
  PRIMARY KEY (idPartit, dniJugador),  
  FOREIGN KEY (idPartit) REFERENCES Partits,  
  FOREIGN KEY (dniJugador) REFERENCES Jugadors);  
■ el jugador dniJugador ha estat alineat al partit  
  identificat per idPartit  
■ els gols i targetes són les que han posat al jugador durant  
  el partit.
```

- 1. (2.5 punts)** Doneu una sentència SQL per obtenir els partits en els que ha guanyat un equip visitant, i en els que, a més, no s'ha alineat cap jugador a l'equip visitant que tingui un salari inferior a la mitjana dels salaris de tots els jugadors de la base de dades. Es vol que en el resultat hi hagi, la data del partit, el nom de l'equip local, el nom de l'equip visitant i l'estadi on es van jugar. Considereu que els partits es juguen a l'estadi de l'equip local.

2. (1.5 punt) Doneu una asserció que no permeti que els equips de futbol tinguin un pressupost inferior a la suma dels salaris dels seus jugadors.

3. (1 punt) Supposeu la vista `pichichi` que té per objectiu mostrar la classificació dels golejadors del campionat.

```
CREATE VIEW pichichi (dniJugador, totalGols) AS
  SELECT dniJugador, SUM(gols)
  FROM alineacions
  GROUP BY dniJugador;
```

Abans de l'inici del campionat es van insertar tots els equips i tots els jugadors de cada equip a les taules corresponents. Per registrar el primer partit es van fer els següents inserts:

```
INSERT INTO PARTITS VALUES (1, 'FCB', '2021/08/15', 'RSO', 4, 2);
INSERT INTO ALINEACIONS VALUES (1, '33333333C', 3, 0);
INSERT INTO ALINEACIONS VALUES (1, '11111111A', 0, 0);
INSERT INTO ALINEACIONS VALUES (1, '22222222B', 2, 0);
```

a) Quin és el resultat d'executar la sentència següent, un cop executats els inserts:

```
SELECT * FROM pichichi ORDER BY totalGols DESC
```

b) És actualitzable la vista? Per què?

c) Modifiqueu la definició de la vista per tal que mostri només els jugadors que han marcat en total més de 2 gols.

4. (1 punt) Supposeu que la Laia és la propietària de la taula `jugadors`.

a) Doneu el diagrama d'autoritzacions que hi haurà després de l'execució de les sentències següents:

```
Laia: GRANT update(salari) ON jugadors TO Emma WITH GRANT OPTION
```

```
Emma: GRANT update(salari) ON jugadors TO Aina
```

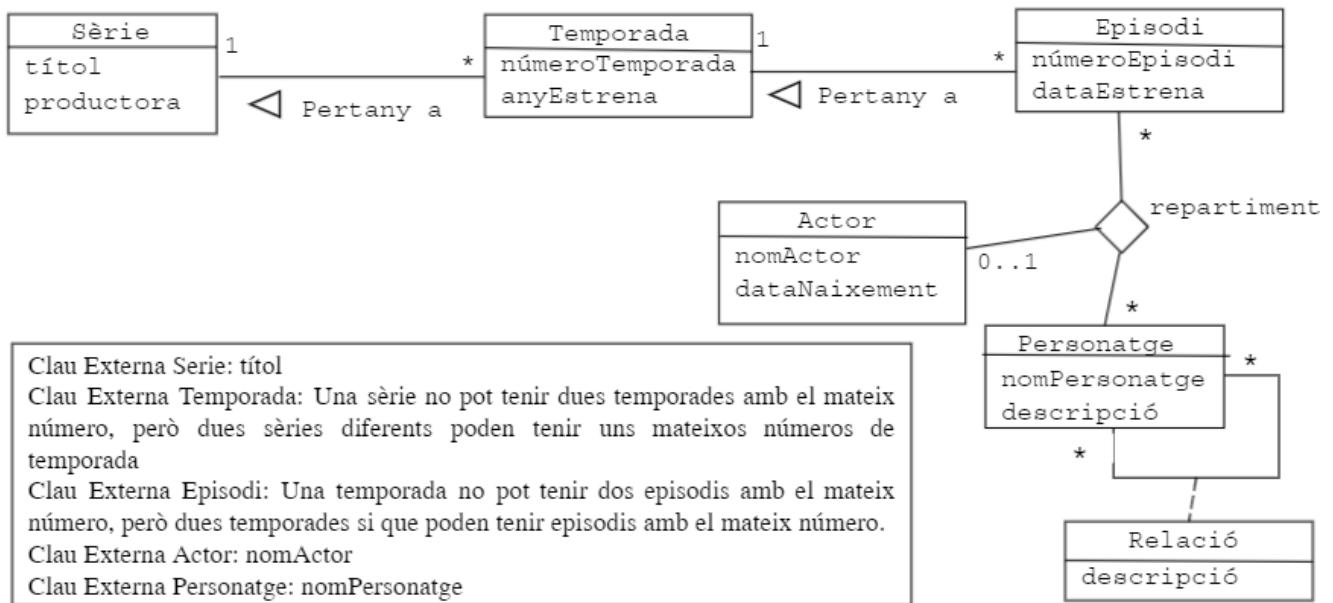
b) Partint del diagrama d'autoritzacions resultant de l'apartat anterior, doneu el diagrama d'autoritzacions després de l'execució de la sentència següent:

```
Laia: REVOKE GRANT OPTION FOR update(salari) ON jugadors FROM Emma CASCADE
```

c) Indiqueu quins privilegis mínims li farien falta al Josep per tal de poder executar la sentència següent:

```
UPDATE jugadors SET salari= salari +100
WHERE NOT EXISTS (SELECT a.idPartit FROM alineacions a
                  WHERE a.dniJugador=jugadors.dniJugador
                  AND a.gols=0);
```


- 5. (1.5 punt)** Doneu una seqüència d'operacions en àlgebra relacional per obtenir els noms dels equips que han marcat fora de casa (han fet algun gol com a visitants) en un únic partit.
- 6. (1.5 punts)** Supposeu el model conceptual en UML següent. Doneu el disseny lògic que s'obté fent la traducció a model relacional. Cal incloure, taules, atributs, claus primàries, claus forànies i restriccions NOT NULL i UNIQUE que siguin necessàries.



7. (1 punt)

- 7.1** L'actualització incorrecta de dades redundants, una avaria en un disc on estan els fitxers d'una BD o transaccions que no poden acabar per un tall de subministrament elèctric són exemples de situacions que comprometen la fiabilitat d'un SGBD. Expliqueu breument un dels mecanismes que tingui un SGBD per garantir l'objectiu de fiabilitat.
- 7.2** Un altre objectiu fonamental dels SGBDs és que permetin a molts usuaris accedir concurrentment a una BD. Expliqueu breument algun problema que es pot produir com a conseqüència de l'accés simultani de molts usuaris a una BD (en cas de no disposar dels mecanismes que ofereixen els SGBD per evitar aquest problema).

1. (1 punt) Donades les taules següents.

```

assignatures(idAssignatura, nomAssignatura, credits)

estudiants(idEstudiant, nomEstudiant, username)

assignaturaMatriculada(idEstudiant, idAssignatura, quadrimestre, nota)
    {idEstudiant} referencia estudiants
    {idAssignatura} referencia assignatures

```

Considereu el següent fragment de codi Java/JDBC. El programa té dos parts. En la primera part es mostra en quantes assignatures han estat matriculats cadascun dels estudiants indicats en la variable *ids*. En la segona part s'esborra de la base de dades una assignatura identificada en la variable *id* si no hi ha cap estudiant que s'hi hagi matriculat; si a l'assignatura s'hi ha matriculat algun estudiant, aleshores es dona un missatge d'error.

```

01  /* Tenim ja una connexió c establerta amb la base de dades */
02
03  /* consultar assignatures */
04  ResultSet rs = null;
05  int[] ids = {1, 2, 3};
06  PreparedStatement ps = c.prepareStatement("select distinct idAssignatura "+
07                                           "from assignaturaMatriculada "+
08                                           "where idEstudiant = ?");
09  for(int i=0; i<ids.length;i++){
10      ps.setInt(1,ids[i]);
11      rs = ps.executeQuery();
12      int quantesAssignatures=0;
13      while(rs.next()){quantesAssignatures = quantesAssignatures + 1;}
14      System.out.println(ids[i] + " ha estat matriculat en "
15                        + quantesAssignatures + " assignatures!");
16  }
17
18  /* esborrar assignatura */
19  int id = 4;
20  Statement stCons = c.createStatement();
21  Statement stDel = c.createStatement();
22  rs = stCons.executeQuery("select count(*) as quants "+
23                          "from assignaturaMatriculada "+
24                          "where idAssignatura = "+id);
25  if (rs.next() && rs.getInt("quants")>0)
26      {System.out.println("Error: no es pot esborrar assignatura "+
27                        "perquè hi ha estudiants matriculats a"+id);}
28  else {stDel.executeUpdate("delete from assignatures where idAssignatura ="+id);}
29
30  /* Commit i desconnexio de la base de dades */

```

Donat aquest fragment de codi, identifiqueu quins dels criteris de qualitat estudiats a l'assignatura no es compleixen. Per cadascun d'ells:

- Expliqueu perquè no es compleix
- Expliqueu en detall quins canvis seria necessari fer en el codi donat per tal de que es compleixi.

2. (2.5 punts) Considereu la base de dades següent.

```
tipusProductes(idTipus, nomTipus, preuMaxim)

productes(nomProducte, idTipus, preu)
    {idTipus} references tipusProductes

comandes(numComanda, import)

productesComanda(numComanda, nomProducte, quantitat)
    {numComanda} references comandes
    {nomProducte} references productes
```

2.1 Supposeu que les taules de la base de dades estan inicialment buides i que executeu una crida al següent procediment emmagatzemat *afegir_producte()*. Quin és l'estat de la taula *productes* després de l'execució? Justifiqueu la resposta.

```
CREATE or replace FUNCTION afegir_producte() RETURNS SETOF char(20)
AS $$
    DECLARE nom char(20);
    BEGIN
        FOR nom IN SELECT nomProducte FROM productes
            LOOP RETURN NEXT nom;
            END LOOP;
        IF NOT FOUND THEN
            RAISE EXCEPTION 'Error inesperat';
        END IF;
        INSERT INTO tipusProductes VALUES (1, 'aperitiu', 10);
        INSERT INTO productes VALUES ('patates', 1, 4);
    RETURN;
END; $$ LANGUAGE plpgsql;
```

2.2 Supposeu que cadascuna de les regles que s'indiquen a continuació es vol implementar mitjançant triggers. Indiqueu i justifiqueu, per a cada regla, quins triggers caldria definir, tenint en compte els criteris de qualitat de l'assignatura. Per cada trigger cal indicar:

- Esdeveniment que l'activa (cal indicar esdeveniment, taula i atributs rellevants)
 - *Before* o *After* (justifiqueu la resposta)
 - *Row* o *Statement* (justifiqueu la resposta)
- a) REGLA1: Hi ha d'haver una fila a la taula *logs* (tipusEsdeveniment, usuari, instant) per cada vegada que un usuari de la base de dades faci insercions o modificacions de comandes. Es vol mantenir amb triggers el contingut de la taula *logs*.
- b) REGLA2: L'import d'una comanda ha de ser igual a la suma dels resultats de multiplicar el preu de cada producte comprat en la comanda per la quantitat d'aquest producte que s'ha comprat. Es vol mantenir correcte el valor de l'atribut *import* quan quedi afectat per esdeveniments sobre la taula *productes*.
- c) REGLA3: No pot existir cap producte a la base dades amb un preu superior al *preuMaxim* del tipus del producte. Es vol generar una excepció en cas que no es compleixi aquesta regla.

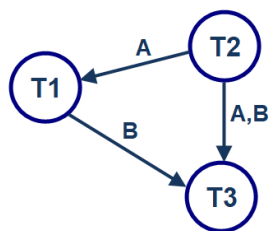
- 2.3** Implementeu un trigger que mantingui correcte el valor de l'atribut `import` de la taula `comandes` quan s'executi una sentència `INSERT` sobre la taula `productesComanda`. Tingueu en compte la REGLA2 de l'apartat anterior que defineix com es calcula l'import d'una comanda.

Podeu fer servir la plantilla de triggers de PostgreSQL següent:

```
CREATE TRIGGER nomTrigger
[BEFORE/AFTER] [UPDATE (of column)/DELETE/INSERT] ON taula
[FOR EACH ROW/FOR EACH STATEMENT] execute procedure nomp();

CREATE FUNCTION nomp() RETURNS trigger AS $$
DECLARE
    ...
BEGIN
    ...
END;
$$ LANGUAGE plpgsql;
```

- 3. (2 punt)** Donat el graf de precedències següent:



- 3.1** Digau per cadascuna de les interferències que es produeixen: entre quines transaccions es produeixen, i els grànuls implicats.
- 3.2** Doneu un horari que inclogui les mínimes operacions (R, RU, W, COMMIT) possibles i que es correspongui amb el graf de precedències.
- 3.3** Digau si l'horari que heu donat en l'apartat anterior és o no recuperable. Justifiqueu la resposta.
- 3.4** Modifiqueu l'horari que heu donat, afegint una única operació (R, RU, W), per tal que s'afegeixi una única interferència de lectura no repetible entre T2 i T3. Si no és possible, justifiqueu la resposta.

- 4. (2 punt)** Donat un SGBD sense cap mecanisme de control de concurrència, suposem que es produeix l'horari següent:

Temps	T1	T2	T3
10		RU(B)	
20		W(B)	
40	RU(A)		
50	W(A)		
60	R(D)		
70	R(B)		
80			R(E)
100			RU(A)
110			W(A)
120			RU(F)
130		R(F)	
140		RU(D)	
150		W(D)	
160			W(F)
170	RU(F)		
180	W(F)		
190		COMMIT	
200			COMMIT
210	COMMIT		

4.1 Supposeu que s'incorpora un mecanisme per al control de la concurrència basat en reserves S, X, on les transaccions T1 i T3 treballa amb SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED, i T2 amb SET TRANSACTION ISOLATION LEVEL READ COMMITTED. Doneu l'horari aplicant reserves que ha d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions. Quan més d'una transacció esperi per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out). En cas que cap acció de l'horari pugui executar-se en un moment donat, no continueu i justifiqueu el motiu.

4.2 Supposeu que s'incorpora un mecanisme per al control de la concurrència basat en reserves S, X, on les transaccions T1, T2 i T3 treballen amb SET TRANSACTION ISOLATION LEVEL SERIALIZABLE. Doneu l'horari aplicant reserves que ha d'incloure, a més de les operacions que executen les transaccions (R, RU, W, COMMIT), les operacions de petició i alliberament de reserves (LOCK, UNLOCK), i l'ordre d'execució de totes aquestes operacions. Quan més d'una transacció esperi per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out). En cas que cap acció de l'horari pugui executar-se en un moment donat, no continueu i justifiqueu el motiu.

5. (2.5 punts) Considereu la taula T(a,b). La taula T té 40000 tuples i està emmagatzemada en un fitxer on a cada pàgina hi caben en promig 10 files.

5.1 En la taula no hi ha cap índex definit. Doneu el cost de la consulta següent i mostreu el detall de com es calcula.

```
SELECT * FROM T where a = valor
```

5.2 Es defineix ara un índex B+ no agrupat per l'atribut a de T. L'índex té d=50 i els nodes estan plens al 70% d'ocupació. Doneu el cost de la consulta de l'apartat anterior i mostreu el detall de com es calcula.

5.3 Es defineix ara un índex B+ no agrupat per l'atribut b de T. L'índex té d=50 i els nodes estan plens al 70% d'ocupació. Doneu el cost de la consulta següent i mostreu el detall de com es calcula.

```
SELECT SUM(b) FROM T
```

5.4 Supposeu ara que només hi ha un índex B+ agrupat per l'atribut a de T. L'índex té d=50 i els nodes estan plens al 70% d'ocupació. No es defineix cap índex sobre l'atribut b. Supposeu que els valors de l'atribut a estan distribuïts uniformement entre 1 i 40000. Doneu el cost de la consulta següent i mostreu el detall de com es calcula.

```
SELECT * FROM T WHERE a>=33000 AND b<=1000
```