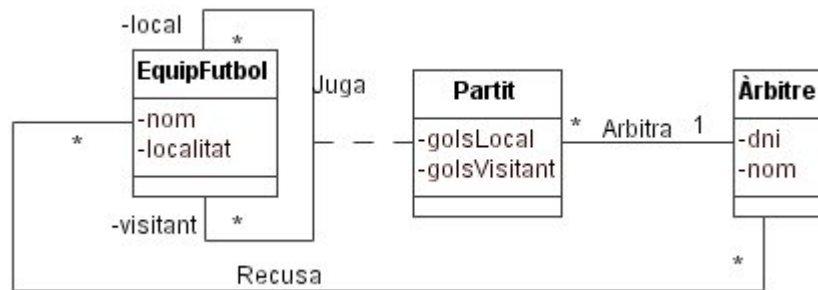


1. Considereu el disseny conceptual en UML següent:



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

1.1 Transformeu el disseny conceptual anterior al model relacional.

1.2 Expliqueu com s'implementarien en l'SQL la restricció següent:

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

1.3 Considereu que, inicialment, l'usuari A és el propietari de la relació *EquipFutbol*, i cap altre usuari en té privilegis. Digueu quin és exactament el conjunt d'usuaris que tenen el privilegi d'INSERT sobre la taula *EquipFutbol* després de l'execució de les instruccions que podeu trobar a continuació:

```

A: GRANT INSERT ON EquipFutbol TO B WITH GRANT OPTION;
A: GRANT INSERT ON EquipFutbol TO D WITH GRANT OPTION;
B: GRANT INSERT ON EquipFutbol TO C WITH GRANT OPTION;
C: GRANT INSERT ON EquipFutbol TO D WITH GRANT OPTION;
B: REVOKE INSERT ON EquipFutbol FROM C RESTRICT;
  
```

1.4 Donades les taules amb les extensions següents:

EquipFutbol(nom, localitat)

Barça Barcelona

R. Madrid Madrid

Jugador (dni, nom)

111 Gerard

222 Sergio

Contracte(EquipFutbol, Jugador, data_inici, data_fi, quantitat)

Barça 111 01-01-2010 31-12-2011 1000000

Barça 111 01-01-2012 NULL 10000000

R.Madrid 222 01-01-2010 31-12-2015 5000000

Suposant que hi ha una clau forana *Contracte.Jugador* que s'ha definit com "ON DELETE SET NULL" i "ON UPDATE CASCADE", i a més a més que hi ha una clau forana *Contracte.EquipFutbol* que s'ha definit com "ON DELETE CASCADE" i "ON UPDATE RESTRICT".

Digues quina serà l'extensió de la taula *Contracte* si s'executen les sentències següents:

- DELETE FROM Jugadors WHERE dni=111;
- DELETE FROM EquipFutbol WHERE nom='Barça';
- UPDATE Jugadors SET dni=333 WHERE dni=111;
- UPDATE EquipFutbol SET nom='FCB' WHERE nom='Barça';

2. Definir en SQL estàndard les següents assercions:

Tot empleat ha de tenir un sou més gran o igual que el sou mínim de la seva categoria

Empleat (nemp, dni, nom, sou, categoria)
Categoria(categoria, soumínim)

El nombre d'empleats d'administració ha de ser menor o igual al nombre d'empleats de producció

Empleats_adm (nemp, dni, nom, sou, categoria)
Empleats_prod (nemp, dni, nom, sou, categoria)

3. Considereu la relació Empleats (num-empl, nom-empl, sou) amb les dades següents:

Empleats(num-empl, nom-empl, sou)
123 Pere 1000
321 Maria 2000
213 Joan 3000

d.1) Hi ha tres usuaris (Pere, Maria i Joan) treballant amb la BD on es troba la relació Empleats i cap d'ells hi té accés a aquesta relació. Doneu les sentències SQL que calguin per a que la Maria pugui consultar i modificar les dades dels empleats amb un sou més petit de 2500 euros.

d.2) Un cop la Maria tingui l'accés que li heu donat a l'apartat anterior, justifiqueu si podrà o no executar la sentència SQL següent:

```
UPDATE Empleats  
SET sou = sou + 1000  
WHERE num_empl = 321;
```

4. 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 comptesBancaris  
(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 comptesBancaris(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 comptesBancaris(numCompte));  
-- Hi ha una fila per cada reintegrament fet al compte bancari.
```

4.1) 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.

4.2) 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.

4.3) 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 saldos1000p (num,saldo) AS  
SELECT cb.numCompte, cb.saldoDisponible -(sum(i.quantitat) - sum(r.quantitat)) FROM  
comptesBancaris 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   s incorrecta. Raoneu la resposta.

5. Considereu l'esquema de la base de dades seg ent:

```
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);  
-- assignaci  d'un professor a un despatx.  
-- instantFi te valor null quan una assignacio es encara vigent.
```

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

5.2 Supposeu que volem una sentència SQL per trobar els professors (dni, nomProf) que no tenen cap assignació vigent. Digueu, per cada una 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;
```

5.3 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'.

6. 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 segon
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(<u>num_dpt</u> ,pressupost)	(1,3000)		
Empleats(<u>num_empl</u> , 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 asserciò 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'asserciò 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'asserciò mitjançant disparadors, respecte a la seva implementació mitjançant procediments emmagatzemats.

7. Considereu l'esquema de la base de dades format per les taules següents:

Actor(nomactor, adreça, telèfon, anynaix, sexe, religió, caché)

on caché és el que cobra un actor per fer una pel·lícula

Tema(nomtema)

Pel·lícula(nompel, nomtema)

on nomtema referencia Tema

Habilitat(nomactor, nomtema)

on nomtema referencia Tema

on nomactor referencia Actor

Actuar(nomactor, nompel)

on nomactor referencia Actor

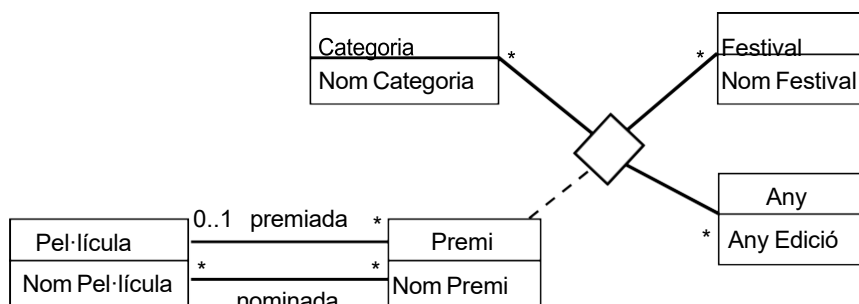
on nompel referencia Pel·lícula

a. 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.

b. 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'autoritacions, l'efecte d'una operació REVOKE sempre es pot anul·lar amb una única operació GRANT.

- c. 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

8.

8.1 Considereu les taules de la base de dades de l'exercici 5, 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.

8.2 Considereu la taula professors(dni,nomProf,telefon), propietat d'en Toni. Suposeu 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.

9. 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();

- 9.1 Expliqueu quin és el principal problema d'aquesta solució, atenent als criteris de qualitat de triggers vistos a classe.
- 9.2 Quins canvis faríeu a d) i e) per tal de superar aquest inconvenient? Per què?
- 9.3 Codifiqueu la nova funció c) que elimini els inconvenients de la solució donada quan es combini amb els canvis introduïts a 9.2.

10.

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?

11. Considereu el següent horari, on les operacions OpL(G) són R(G) o RU(G).

T1	T2	T3	T4	T5
		OpL(A)		
				R(D)
		OpL(C)		
				R(F)
		OpL(F)		
R(A)				
	R(C)			
R(F)				
	W(C)			
W(A)				
			R(C)	
			R(F)	
			W(C)	
				W(D)
				W(F)
		Op		
				Commit
			Commit	
Commit				
	Commit			
		Commit		

a) Proposeu una operació per cadascuna de les operacions OpL i Op per tal de que es provoquin dues i només dues interferències. Indiqueu per cada interferència: el nom de la interferència, entre quines dues transaccions es produeix i quin/s grànul/s implica.

b) Doneu el graf de precedències associat una vegada instanciades les operacions.

c) Quin és el mínim nivell d'aïllament necessari per tal de que s'evitin les interferències descrites a l'apartat 11.1?

d) Escollint el nivell d'aïllament determinat a l'apartat 11.3, com quedaria l'horari? 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. Quan més d'una transacció espera per un mateix grànul, considereu que la política de la cua és FIFO (First In, First Out).

12.

12.1 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 d e

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.

12.2 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	

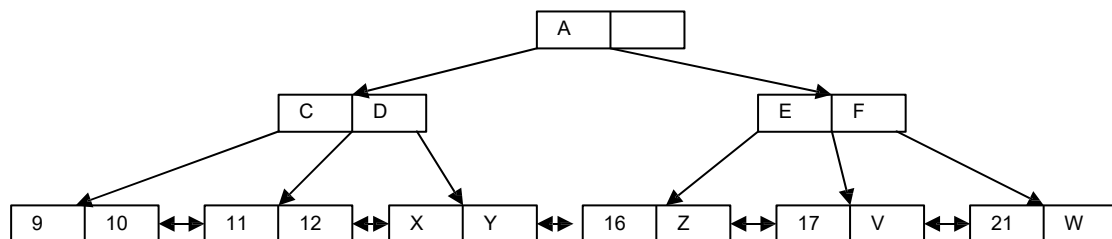
Suposant que les dues transaccions treballen amb nivell *de repeatable read*, que l'SGBD fa servir reserves S, X, que el gràdul és la fila, i que la taula R conté les files {'a1',1), ('a2',2)}, contesteu a les preguntes següents:

- 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?

12.3 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?

13. 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.

- 14.** Tenim una taula amb 200000 empleats, que està emmagatzemada en pàgines en les que hi ha un promig de 10 tuples per pàgina. Els números d'empleat van del 1 al 200000. Hi ha un índex definit sobre l'atribut numEmpl d'aquesta taula que està implementat com un arbre B+. Les pàgines de l'índex estan plenes a un 80%.

- a. Determineu la d òptima
- b. Supposeu per aquest apartat que l'ordre de l'arbre B+ és d=100. Determineu la quantitat de pàgines necessàries per a emmagatzemar l'índex.
- c. Supposeu per aquest apartat que l'ordre de l'arbre B+ és d=100. Indiqueu quina quantitat de pàgines (d'índex i de dades) cal accedir per resoldre la consulta següent, suposant primer que l'índex és agrupat, i després que és no agrupat.

`SELECT * FROM empleats WHERE numEmpl BETWEEN 10000 AND 20000`

- d. Supposeu per aquest apartat que l'ordre de l'arbre B+ és d=100. Indiqueu quina quantitat de pàgines (d'índex i de dades) cal accedir per resoldre la consulta següent, suposant primer que l'índex és agrupat, i després que és no agrupat.

`SELECT count(*) FROM empleats WHERE numEmpl BETWEEN 10000 AND 20000.`

- 15.** Tenim una base de dades bancària amb una taula de comptes que té una cardinalitat de 2.000.000 files. L'esquema d'aquesta taula és:

Comptes(numCompte, dniClient, saldo)

El número de compte és clau primària de la taula i els números de compte són enters repartits uniformement entre 1 i 2.000.000. Es demana el següent, justificant breument les respostes:

- a) Supposeu que existeix un índex no agrupat amb ordre d=150 per l'atribut numCompte. Quin és el nombre mínim i màxim de pàgines que ocupa l'índex?.

- b) Donada la consulta següent:

`SELECT * FROM comptes WHERE saldo>300;`

Suposant que ara es disposa d'un índex no agrupat per saldo amb d=150, que els nodes de l'índex estan plens al 85% de la seva capacitat, l'alçada de l'índex és 3 i que el factor de bloqueig del fitxer de dades és de 10 files per pàgina en mitjana. Calculeu el nombre màxim de files que poden complir la condició saldo>300 per a que sigui més útil resoldre-la emprant l'índex no agrupat per saldo en lloc de fer un recorregut seqüencial del fitxer de dades.

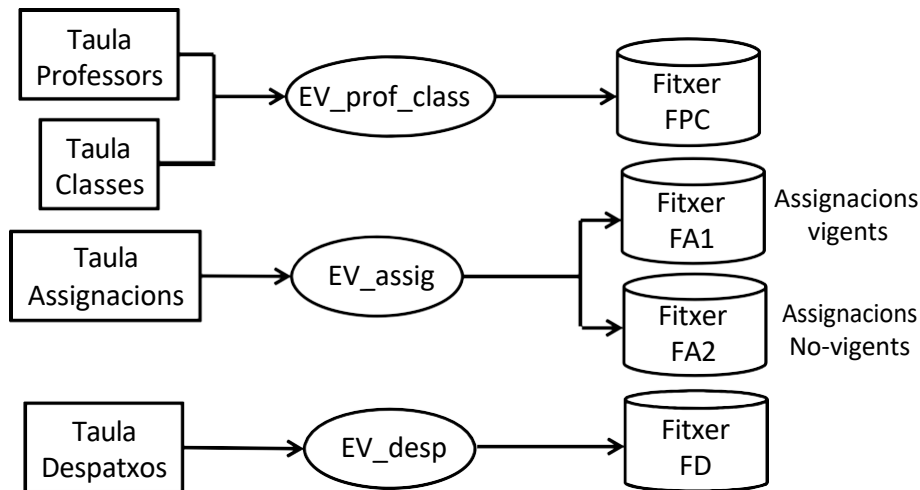
16. Considereu la base de dades de l'exercici 5.

16.1 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.

16.2 Supposeu ara, que a la base de dades de l'exercici 5 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';
```

16.3 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. Sabem 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'
```

16. Tenim una base de dades bancària amb una taula de comptes que té una cardinalitat de 2.000.000 files. L'esquema d'aquesta taula és:

Comptes(numCompte, dniClient, saldo)

El número de compte és clau primària de la taula i els números de compte són enters repartits uniformement entre 1 i 2.000.000. Es demana el següent, justificant breument les respostes:

c) Supposeu que existeix un índex no agrupat amb ordre $d=150$ per l'atribut numCompte. Quin és el nombre mínim i màxim de pàgines que ocupa l'índex?

d) Donada la consulta següent:

`SELECT * FROM comptes WHERE saldo>300;`

Suposant que ara es disposa d'un índex no agrupat per saldo amb $d=150$, que els nodes de l'índex estan plens al 85% de la seva capacitat, l'alçada de l'índex és 3 i que el factor de bloqueig del fitxer de dades és de 10 files per pàgina en mitjana. Calculeu el nombre màxim de files que poden complir la condició $\text{saldo}>300$ per a que sigui més útil resoldre-la emprant l'índex no agrupat per saldo en lloc de fer un recorregut seqüencial del fitxer de dades.

17. Calcula el cost òptim de la consulta següent:

Select *

From R, S

Where $R.c > 95$

and $R.a = S.a$

Tenint en compte:

$R(\underline{a}, b, c)$

Índex cluster per R.a; Índex no cluster per R.c; $d=75$ pels dos arbres; ocupació dels dos arbres: $2/3$

$\text{Min}(c) = 0$; $\text{Max}(c) = 100$; $\text{NDIST}(c) = 3000$

$\text{Card}(R) = 10000$; 20 tuples per pàgina; $B_R = 500$

$S(\underline{c}, a, d)$

Índex cluster per S.a; Índex no cluster per S.c; $d=75$ pels dos arbres; ocupació dels dos arbres: $2/3$

$\text{NDIST}(a) = 10000$

$\text{Card}(S) = 100000$; 20 tuples per pàgina; $B_S = 5000$

En el cas de block nested loop, considereu que $M=1$.

No cal comptar el cost d'escriure els resultats de les operacions ni del resultat.

