

Temps: 2.45 h

Notes: 1 Juliol tarda Revisió: 3 Juliol tarda

Cada pregunta en un full separat

- 1 (3 pt)** Considereu l'esquema de la base de dades de la cadena de pizzeries que hem usat durant el curs:

```
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 check(import>0),
primary key (numComanda),
foreign key (numTelf) references domicili,
check (instantServida>instantFeta));

create table liniaComanda
(numComanda integer,
idProducte char(9),
quantitat integer check(quantitat>0),
primary key (numComanda,idProducte),
foreign key (numComanda) references comanda,
foreign key (idProducte) references producte);
```

- a) (1.25 pt)** Doneu una sentència SQL per obtenir els números de les comandes que, o bé són a repartir a domicili (numTelf té valor diferent de nul) i tenen un import superior a 30, o bé són a recollir a la pizzeria (numTelf té valor nul) i totes les seves línies de comanda són de productes de mida “gran”.

- b) (0,5 pt)** Suposa la consulta SQL següent:

```
select c.numTelf
from comanda c, liniaComanda lc
where c.numComanda = lc.numComanda
group by c.numTelf, c.numComanda
having count(*) = count(distinct lc.quantitat)
```

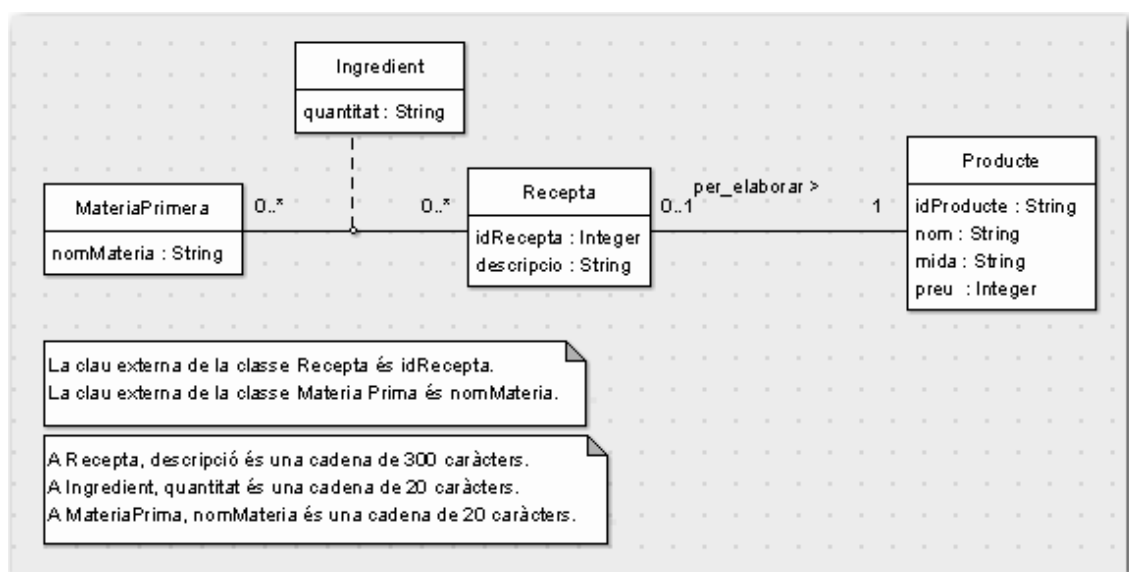
Completa l'extensió de les taules *comanda* i *liniaComanda* següents (amb valors diferents de nul), de tal manera que si s'executa la consulta s'obtinguin dues files com a resultat. Suposa que els productes que existeixen a la taula productes tenen identificadors "p111", "p222" i "p333".

numComanda	instantFeta	instantServida	numTelf	import
1	10	15	931111111	50
2	20	25	932222222	100
3	40	60	932222222	55
4			934444444	68

numComanda	idProducte	quantitat
1	p111	
1	p222	
1	p333	
2	p111	
2	p222	
3		
3		
4		
4		
4		

c) (0,5 pt) Doneu una seqüència d'operacions d'àlgebra relacional per obtenir els noms dels carrers des dels que s'ha fet alguna comanda però no se n'ha fet cap amb un instantServida superior a 2000. S'entén que les comandes fetes des d'un carrer, són les comandes fetes des d'algun dels domicilis del carrer.

d) (0,75 pt) Suposem que s'ha estès el model conceptual UML a partir del qual s'ha dissenyat la base de dades anterior, per afegir informació sobre receptes d'alguns dels productes que es comercialitzen a la cadena de pizzereries.



Doneu les sentències SQL de creació de les taules corresponents a la transformació a model relacional de les classes *Ingredient* i *Recepta*.

2 (2 pt) Sobre la BD de l'exercici 1, es vol fer la inserció de les línies de comandes mitjançant un procediment emmagatzemat. Addicionalment es vol garantir mitjançant disparadors que només es puguin donar d'alta comandes a domicili als domicilis situats a Barcelona o Hospitalet, i també es vol mantenir actualitzat de forma automàtica l'atribut import de la taula de comandes (l'import d'una comanda es calcula com la suma dels resultats obtinguts de multiplicar la quantitat de producte de cadascuna de les línies de la comanda pel preu d'aquest producte). Suposarem que l'import de la comanda només s'actualitza quan s'insereixen línies de comanda, és a dir, que no se'n fan modificacions posteriors.

L'ABD encarrega la creació del procediment emmagatzemat i dels disparadors a un programador inexpert que presenta la solució que es mostra tot seguit.

Es demana:

Digueu quins són els errors d'eficiència que hi ha a la solució proposada pel programador, tot donant una breu explicació de cada error (numereu cada error per facilitar-ne la referència). Expliqueu breument com solucionaríeu cadascun dels errors.

Nota: Les línies es numeren per facilitar-ne la referència.

```
1. Create Procedure insercio_Linies(numCom int, numprod char(9),qtt int)
2. define codi_error,codi_sql int;
3. define mis_error varchar(50);
4. on exception set codi_error,codi_sql,mis_error
5.   if codi_error=-746 then raise exception -746,0,mis_error;
6.   else raise exception -746,0,'Error intern';
7.   end if;
8. end exception;
9. if (not exists(select * from comanda where numComanda=numCom)) then
10.   raise exception -746,0,'La comanda no existeix';
11. elif (not exists(select * from producte where idproducte=numProd)) then
12.   raise exception -746,0,'El producte no existeix';
13. elif (exists (select * from liniaComanda where idproducte=numProd and
numComanda=numCom)) then
14.   raise exception -746,0,'La línia ja existeix';
15. elif (qtt>0) then
16.   insert into liniaComanda values (numCom,numProd,qtt);
17. else raise exception-746,0,'La quantitat ha de ser més gran que 0';
18. end if;
19. end procedure;

20. create procedure compr_com()
21.   if ((select count(*)
22.   from domicili d, comanda c
23.   where c.numtelf=d.numtelf and d.ciutat<>'Barcelona' and d.ciutat<>'Hospitalet')>0) then
24.   raise exception -746,0,'Només comandes a domicili a Barcelona o
Hospitalet';
25.   end if;
26. end procedure;
```

```
27. create trigger insertcom
28. insert on Comanda
29. after (execute procedure compr_com());

30. create procedure calcular_import_com (idcom like comanda.numcomanda)
31. define importCom int;
32. define qtt,preu_p int;

33. let importCom,qtt,preu_p=0,0,0;
34. foreach cursor1 for select l.quantitat, p.preu into qtt, preu_p
35.         from liniaComanda l, producte p, comanda c
36.         where l.idproducte=p.idproducte and
37.         c.numcomanda=l.numcomanda and l.numcomanda=idcom
38.         Let importCom=importCom+qtt*preu_p;
39. end foreach;

40. Update comanda
41. set import=importCom
42. where numComanda=idcom;
43. end procedure;

44. create trigger insertlinies
45. insert on liniacomanda
46. referencing new as nou
47. for each row (execute procedure calcular_import_com (nou.numComanda));
```

3 (2.5 pt) Per cadascun dels horaris següents indiqueu:

- a) si són serialitzables,
- b) si poden ser generats (mateix ordre de les operacions) fent servir un protocol de reserva de dues fases (PR2F),
- i c) si poden ser generats fent servir un PR2F estricte.

Justifiqueu convenientment les respostes. En particular, si un horari pot ser generat per un PR2F o per un PR2F estricte, mostreu els locks i unlocks de cada transacció d'acord amb aquest protocol que demostrin que pot ser generat.

HORARI 1

T1	T2	T3
		R(D)
R(A)		
W(A)		
		W(D)
	R(D)	
R(B)		
W(B)		
	R(B)	
		C
	C	
C		

HORARI 2

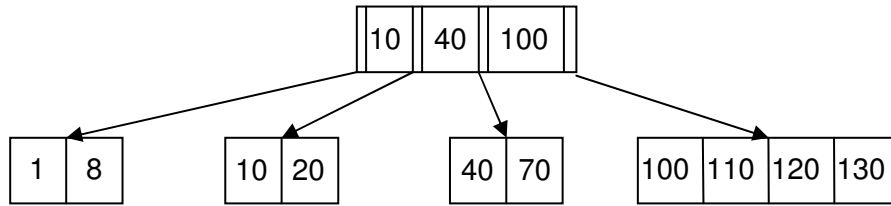
T1	T2	T3
R(A)		
W(A)		
	R(B)	
		R(D)
		W(D)
R(B)		
W(B)		
	R(D)	
		R(D)
		W(D)
		C
	C	
C		

HORARI 3

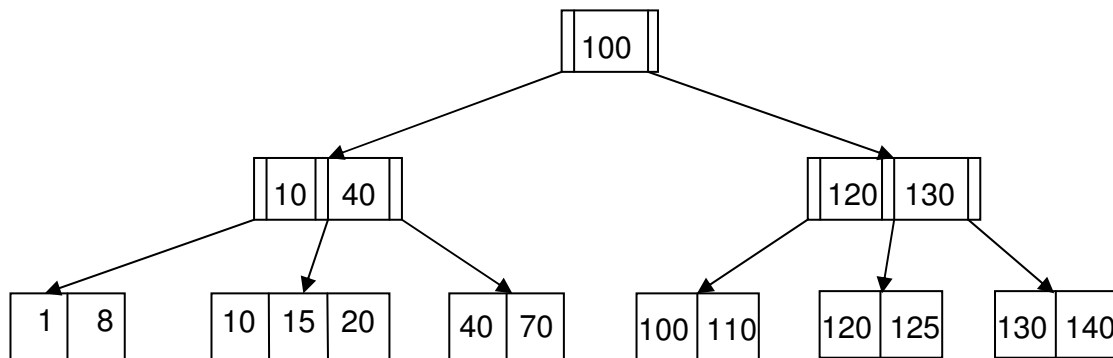
T1	T2	T3
R(A)		
W(A)		
		R(D)
		W(D)
R(B)		
W(B)		C
	R(D)	
	W(D)	
	C	
C		

4 (2.5 pt)

a) Donat l'arbre d'ordre $d=2$ següent:



doneu una possible seqüència d'insercions seguides de supressions (no es poden barrejar) que doni lloc a aquest arbre:



Per a cada inserció, doneu el bocí d'arbre afectat després de la inserció. Nota: poden haver-hi diverses solucions; només cal donar-ne una.

b) Tenim un arbre B+ d'ordre $d \geq 2$. Pot passar que dues insercions consecutives incrementin l'alçada en 2? Si la resposta és que sí, doneu un exemple on això passi. Si la resposta és que no, raoneu per què.