

Nom :

Prénom :

Bases de données – LI341

Examen partiel du 13 novembre 2014

Durée : 2h – 20 points – CORRIGÉ

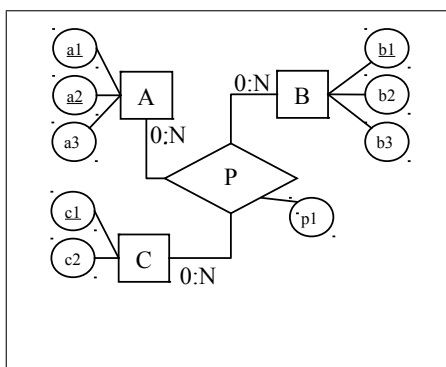
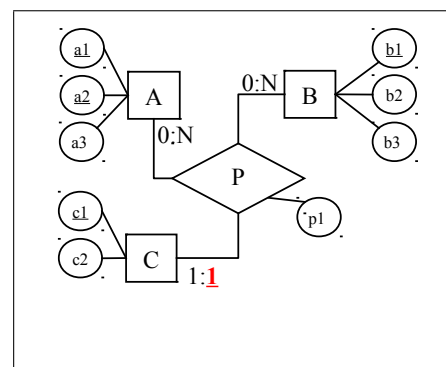
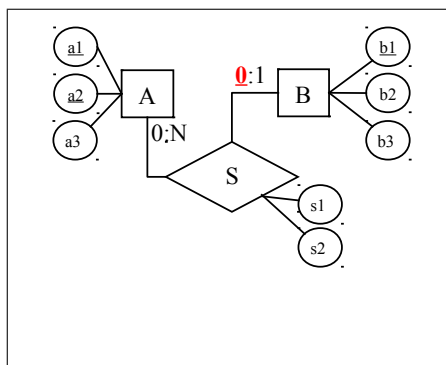
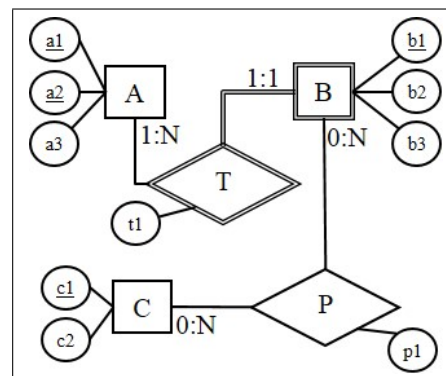
Documents autorisés

1 Modèle E/A (6 pts)

Dans les questions 1 et 2, on demande de compléter le schéma relationnel S qui contient déjà la relation **A** (**a1**, **a2**, **a3**). Pour chaque nouvelle relation (autre que **A**) préciser son nom, le nom de ses attributs, et souligner la clé primaire. Les attributs des clés étrangères portent le même nom que la clé primaire de la table référencée.

Question 1 (4 points)

Compléter le schéma relationnel S pour obtenir trois schémas relationnels S_1 , S_2 , S_3 et S_4 qui correspondent respectivement aux quatre schémas Entité-Association EA_1 , EA_2 , EA_3 et EA_4 ci-dessous :

(a) Schéma EA_1 (b) Schéma EA_2 (c) Schéma EA_3 (d) Schéma EA_4

Solution: S1 :

B (b1, b2, b3)

C (c1, c2)

P (a1, a2, b1, c1, p1)

S2 :

B (b1, b2, b3)

C (c1, c2, a1, a2, b1, p1)

S3 :

B (b1, b2, b3)

S (b1, a1, a2, s1, s2) ne pas souligner a1,a2

S4 :

B (b1, a1, a2, b2, b3, t1)

C (c1, c2)

P (c1,b1, a1, a2, p1) P doit contenir a1 et a2

Question 2 (1 point)

Dans le schéma E/A EA_2 , on modifie la cardinalité entre B et P en remplaçant $[0 : N]$ par $[0 : 1]$. Faut-il modifier la structure des tables ? Quelle contrainte faut-il ajouter au schéma ?

Solution:

même schéma, on ajoute **unique(b1)** dans C

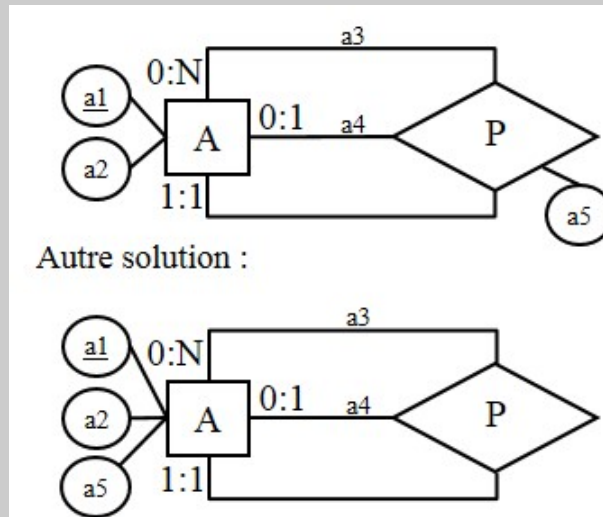
Question 3 (1 point)

Soit la table A :

```
CREATE TABLE A (  
    a1 number primary key ,  
    a2 number ,  
    a3 number references A ,  
    a4 number unique references A  
    a5 number );
```

Dessiner un schema E/A correspondant à la table A.

Solution:



2 Contraintes (5 pts)

Une base a trois tables A, B, C :

```
create table A (
    a1 number primary key ,
    a2 number );

create table B (
    a1 number references A(a1) on delete cascade ,
    b1 number ,
    b2 number ,
    primary key (a1 , b1 ));

create table C (
    a1 number ,
    b1 number ,
    c1 number ,
    c2 number ,
    primary key (a1 , b1 , c1 ) ,
    foreign key (a1 , b1) references B(a1 , b1) on delete cascade );
```

La base est remplie initialement avec 7 nuplets :

A	<u>a1</u>	a2	B	<u>a1</u>	<u>b1</u>	b2	C	<u>a1</u>	<u>b1</u>	<u>c1</u>	c2
	1	10		1	1	10		1	1	5	10
	2	20		1	2	20		1	2	5	20
				2	3	20					

Question 4 (1 point)

Expliquez ce qui se passe si on exécute l'instruction SQL :

```
insert into C values (2, 1, 1, 30);
```

Solution: Refus car il n'y a aucun nuplet dans B avec a1=2 et b1=1

Question 5 (1 point)

On considère la base initiale avec ses 7 nuplets. Que se passe-t-il si on envoie la commande :

```
delete A where a2=20;
```

Solution: supprime 2 nuplets : 1 dans A, 1 dans B

Question 6 (1 point)

On considère la base initiale avec ses 7 nuplets. Que se passe-t-il si on envoie la commande :

delete A where a2=10;

Solution: supprime 5 nuplets : 1 dans A, 2 dans B, 2 dans C

Dans les questions suivantes, on considère un **nouveau** schéma défini par les commandes suivantes :

```
create table A(  
  a1 number primary key ,  
  a2 number not null );  
create table B(  
  b1 number primary key ,  
  b2 number references A(a1) on delete restrict );  
alter table A add foreign key(a2) references B(b1) on delete restrict ;
```

Question 7 (1 point)

Les 2 tables **A** et **B** sont vides. Les commandes suivantes, prises séparément, sont-elles refusées ? Si oui, pourquoi ? On considère que les deux tables sont vides avant chaque commande. *Entourer la bonne réponse et donner une justification.*

Solution: insert into A values(3, null) ; –refus car a2 not null
insert into A values(1, 10) ; – refus car 10 n'existe pas dans B.a1
insert into B values(10,1) ; –refus car 1 n'existe pas dans A.a1

Question 8 (1 point)

On suppose que chaque table (A et B) contient 100 nuplets. Les commandes suivantes sont-elles refusées ? Si oui, pourquoi ? *Entourer la bonne réponse et donner une justification.*

Solution:

delete from A ; – refus car B.a1 est "on delete restrict"

delete from B ; – refus car A.a2 est "on delete restrict"

Question bonus (1 point)

Quelle suite d'instructions SQL aboutit à vider le contenu des deux tables A et B ?

Remarque : Il faut uniquement utiliser des instructions de *mise-à-jour de données*. En particulier, on conserve la définition des tables, i.e., on n'a pas le droit de faire un *drop table* ou *alter table*, etc...

Solution:

update B set b2 = null ;

delete A ;

delete B ;

3 Requêtes (7 pts)

Considérons le schéma relationnel suivant qui porte sur des compagnies aériennes, aéroports et vols d'avions.

Compagnie (nom, taille) **Vol** (numero, horaire, nomComp)
Etape (numeroVol, depart, destination) **Aéroport** (code, ville, pays)
Escale (aéroport, numeroVol, attente)

Chaque vol est renseigné par son numéro qui est unique, l'horaire auquel il doit démarrer et la compagnie qui l'opère. Un vol peut se dérouler en une seule étape dont on connaît l'aéroport de départ et l'aéroport d'arrivée. Il peut également se dérouler en deux étapes, c'est-à-dire avoir une seule escale. Dans ce cas, les deux étapes font référence au même numéro de vol. Un exemple d'une instance des tables **Vol** et **Etape** est fourni ci-dessous.

La table **Escale** indique pour les vols avec escale le temps d'attente dans l'aéroport de transit, avant de continuer l'étape suivante.

Enfin, pour chaque aéroport, identifié par son code, on connaît la ville et le pays où il se situe. Certaines villes ont plusieurs aéroports.

Vol			Etape		
numero	horaire	nomComp	numeroVol	depart	destination
AF1004	10h00	AirFrance	AF1004	CDG	LAX
AF2008	01h00	AirFrance	AF2008	CDG	DEL
...			AF2008	DEL	SYD
			...		

Pour chacune des questions suivantes, exprimez les requêtes dans les langages indiqués.

Question 9 (2 points)

Quels sont les numéros des vols qui relient un aéroport situé à Paris à l'aéroport SYD en faisant escale par l'aéroport DEL ?

Solution:

Calcul n-uplet

Simple requête de chemin dans la table Etape. La jointure avec la table Vol n'est pas nécessaire sauf si on veut rajouter l'information sur la compagnie.

$$R1(v) = \{v.numero \mid v \in Vol \wedge \exists p1 \in Etape(p1.numeroVol = v.numero \wedge \exists a1 \in Aéroport(a1.ville = 'Paris' \wedge a1.code = p1.depart \wedge p1.destination = 'DEL' \wedge \exists p2 \in Etape(p2.numeroVol = v.numeroVol \wedge p2.depart = 'DEL' \wedge p2.destination = 'SYD'))))\}$$

La réponse reste correcte si on oublie le prédicat $p2.depart = 'DEL'$ car l'aéroport de départ de la 2eme étape est forcément l'aéroport d'arrivée de la 1ere étape

SQL

On suppose pour les questions suivantes que les vols se déroulent en plusieurs étapes (n étapes et n-1 escales).

Question 10 (2 points)

Quel est l'aéroport de départ de chaque vol ? Dans l'exemple ci-dessus, le résultat retourné par cette requête est :

AF1004	CDG
AF2008	CDG

Solution:**Calcul n-uplet**
$$\{e1.numeroVol, e1.depart \mid e1 \in Etape \wedge \neg \exists e2 \in Etape (e1.numeroVol = e2.numeroVol \wedge e1.depart = e2.destination)\}$$
SQL

```
select e1.numeroVol, e1.depart
from Etape e1
where (e1.numeroVol, e1.depart) not in
      (select e2.numeroVol, e2.destination
       from Etape e2)
```

Autre solution possible avec le not exists (traduction requête en TRC)

Compagnie (nom, taille)

Etape (numeroVol, depart, destination)

Escale(aeroport, numeroVol, attente)

Vol (numero, horaire, nomComp)

Aeroport (code, ville, pays)

Question 11 (1 point)

Quels sont les numéros de vols pour lesquels toutes les escales durent moins de deux heures (i.e., aucune escale de plus de 2h) ?

Solution:

SQL

```
select e1.numeroVol
from Escale e1
where e1.numeroVol not in
( select e2.numeroVol
from Escale e2
where e2.attente > 2
);
```

Question 12 (2 points)

Quels sont les vols de la compagnie Air France dont l'attente totale à toutes les escales ne dépasse pas 3 heures ?

Solution:

```
select e.numeroVol
from Escale e, Vol v
where e.numeroVol = v.numero
and v.nomComp = 'Air_France'
group by e.numeroVol
having sum(e.attente) < 3 ;
```

4 Triggers (2 pts)

Dans cet exercice, on considère le schéma de la section **Requêtes**. On voudrait mettre en place un trigger qui met à jour les code des aéroports lorsqu'ils sont modifiés dans la table **Aeroport**.

Question 13 (2 points)

Complétez le trigger suivant qui effecute cette tâche.

Solution:

```
CREATE [ OR REPLACE ] TRIGGER modifie_escale
AFTER UPDATE OF code ON Aeroport
FOR EACH ROW
BEGIN
  set Etape.depart:=:new.code where Etape.depart=:old.code;
  set Etape.destination:=:new.code where Etape.destination =:old.code;
  set Escale.aeroport:=:new.code where Escale.aeroport =:old.code;
END
```