SI3

BDR 2 heures 13 Janvier 2021

Si vous manquez de place pour répondre à une question, poursuivez à la fin de la copie.

Vos réponses doivent être justifiées.

Nom : _____

Question	Points	Note
Dépendances et Formes Normales	2	
Normalisation	4	
Chocolats	9	
Recursivité	5	
Passage EA vers relationnel 1	3	
Total:	23	

Question 1: Dépendances et Formes Normales .....

(a) [1 point] La relation $R(A,B,C,D,E)$ satisfait l'ensemble de dépendances fonctionnelles DF = $\{A \rightarrow C, C \rightarrow BE, E \rightarrow A\}$. Les attributs sont atomiques. Déterminez toutes les clés candidates de R.

Solution:
D doit faire partie de toutes les clés candidates; B ne peut pas faire partie d'une clé candidate. D seul ne constitue pas une clé candidate.
Les clés candidates sont AD,CD et DE.

(b) [1 point] La relation R est elle en 1NF, 2NF, 3NF, BCNF?

Solution:
- 1NF seulement, car B attribut non primaire dépend fonctionnellement d'une sous clé C

Question 2: Normalisation .....

On considère la relation R(A, B, C, D) qui satisfait l'ensemble de dépendances fonctionnelles DF=$\{AB \rightarrow C, C \rightarrow A, C \rightarrow D, D \rightarrow B\}$.

(a) 1 point Calculez la fermeture transitive de chacun des 4 attributs. En déduire qu'il n'y a qu'une seule clé constituée d'un seul attribut. Laquelle?
Rappel : la fermeture d'un attribut $Attr$ est le plus grand ensemble FT tel que l'on peut déduire $Attr \rightarrow FT$ de l'ensemble des dépendances fonctionnelles vérifiées par la relation.

Solution:
- $A^+ = \{A\}$;
- $B^+ = \{B\}$;
- $C^+ = \{A,B,C,D\}$;
- $D^+ = \{B,D\}$;

C est la seule clé primaire constituée d'un seul attribut

(b) 1 point Deux autres clés candidates sont constituées de deux attributs. Quelles sont elles? Y a t il d'autres clés candidates?

Solution: Puisque $AD^+ = \{A,B,C,D\}$, AD et une clé candidate.
Puisque C est une clé candidate et que $AB \rightarrow C$, AB est aussi une clé candidate. La seule autre clé candidate potentielle ayant deux attributs serait BD, mais $\{B,D\}^+ = \{B,D\}$. Il ne peut pas y avoir de clé candidate à trois attributs, car soit elle contiendrait C , soit elle serait égale à ABD, et ne serait donc pas minimale.

(c) 1 point Décomposer R en 3NF.

Solution: Puisque tous les attributs sont primaires, R est en 3NF , il n'y a rien à faire.
Remarque:il était toutefois possible (et cela n'a pas été sanctionné) d'appliquer l'algorithme de décomposition en 3NF à R et de le surdécomposer mais ce n'était pas nécessaire.

(d) 1 point Décomposer R en BCNF

Solution:
Seule la dernière dépendance fonctionnelle viole la BCNF, il faut donc décomposer R en R1(D,B) et R2(A,C,D)

Question 3: Chocolats .....

On considère le schéma de bases de données suivant:

```
create table chocolatier(
idchocolatier integer  primary key,
nom varchar(25),
ville varchar(25));

create table Client(
idClient integer primary key,
nom varchar(25),
ville varchar(25));

create table Chocolat(
idChocolat integer primary key,
nomChocolat varchar(25),
pourcentageCacao integer,
origineFeve varchar(25),
biologique boolean, --vrai si chocolat bio
prix numeric,
idChocolatier integer references chocolatier);

create table Commande(
idCommande serial primary key,
idChocolat integer references chocolat,
idClient integer references client,
quantite integer, --en grammes
dateCommande date);
```

Pour toutes les questions, un chocolat est noir s'il contient plus de 40 pour cent de cacao.

(a) 1 point Donner la liste de tous les chocolatiers (identifiant et nom) dont le nom contient un b mais pas de t.

Solution:

```
select idchocolatier, nom
from chocolatier
where nom like '%b%' and  not nom like '%t%'
```

(b) 1 point Lister tous les clients (idCLient, nom et ville) qui ont commandé du chocolat bio.

Solution:

```
select distinct idclient, nom, ville
from client
join Commande
using(idClient)
join Chocolat using (idChocolat)
where biologique;
```

(c) 1 point Donner le nombre de chocolats par origine (origineFeve).

Solution:

```
select origineFeve, count(idchocolat)
from chocolat
group by origineFeve
```

(d) 1 point Ecrire une requête qui renvoie le nombre de kilos de chocolat noir qui ont été commandés en tout.

Solution:

```
select sum(quantite)/1000 as "Total chocolat noir vendu "
from chocolat
join commande
using(idchocolat)
where pourcentagecacao >40;
```

(e) 1 point Donner pour chaque chocolatier son identifiant, son nom et la moyenne des prix des chocolats bio qu'il propose.

Solution:

```
select idchocolatier, nom, avg(prix)
from chocolatier
join chocolat
using(idchocolatier)
where biologique
group by idchocolatier, nom;
```

(f) 2 points Ecrire une requête qui renvoie le nom et la ville du (ou des) client ayant commandé le plus de chocolat bio.
Remarque: la requête ci dessous recherche le ou les clients ayant commandé la plus grande quantité de chocolat bio sur l'ensemble de leurs commandes. Certains ont recherché les clients ayant commandé la quantité max d'un certain chocolat bio par commande, cette interprétation a aussi été acceptée.

Solution:

```
select nom , ville from client
join commande using(idclient)
join chocolat using (idchocolat)
where biologique
group by idclient, nom, ville
having sum(quantite) >= all
  (select sum(quantite)
   from commande
   join chocolat using(idchocolat)
   where biologique
```

```
   group by idclient);
```

(g) [2 points] Donner les identifiants et nom des chocolatiers qui ne proposent que du chocolat produit avec des féves d'origine "Colombie".

Solution:

```
select C1.idchocolatier, C1.nom
from chocolatier C1
where not exists
   (select idchocolat
from chocolat
where origineFeve <>"Colombie"
and chocolat.idchocolatier=C1.idchocolatier);
```

Question 4: Recursivité .....

Soit la table personne qui contient les informations suivantes :

```
CREATE TABLE personne
(
 identifiant integer primary key,
 nom character varying(30),
 prenom character varying(30),
 parent1 integer references personne,
 parent2 integer references personne,
 datenaissance date
)
```

parent1 et parent2 sont les identifiants des deux parents de la personne. **Attention il n'y a pas d'ordre sur les parents et deux enfants ayant les mêmes parents peuvent avoir parent1 et parent2 inversés.**

(a) 3 points Ecrire une requête non récursive qui crée la vue PremierNe(personne, enfant) où enfant est l'identifiant de l'enfant le plus agé de la personne d'identifiant personne.

Solution:

```
drop view if exists PremierNe;
with parenfant(personne,enfant,ddn) as (
select  p1.parent1 ,p1.identifiant , p1.datenaissance from personne
union
select  p1.parent2 ,p1.identifiant , p1.datenaissance from personne

create view PremierNe (personne , enfant) as (
select
pa1.personne ,pa1.enfant from parenfant where
pa1.ddn =select min (pa2.ddn ) from parenfant pa2 where pa2.personne
```

(b) 2 points Écrire une requête récursive qui retourne les noms, prénoms, et distance générationnelle de tous les "premiers nés" directs (enfant, distance 1) ou indirects (le premier né direct d'un de mes premiers nés direct ou indirect est un de mes premiers nés indirects) de la personne de la table dont la datedenaissance est la plus ancienne, classés par distance décroissante.

Solution:

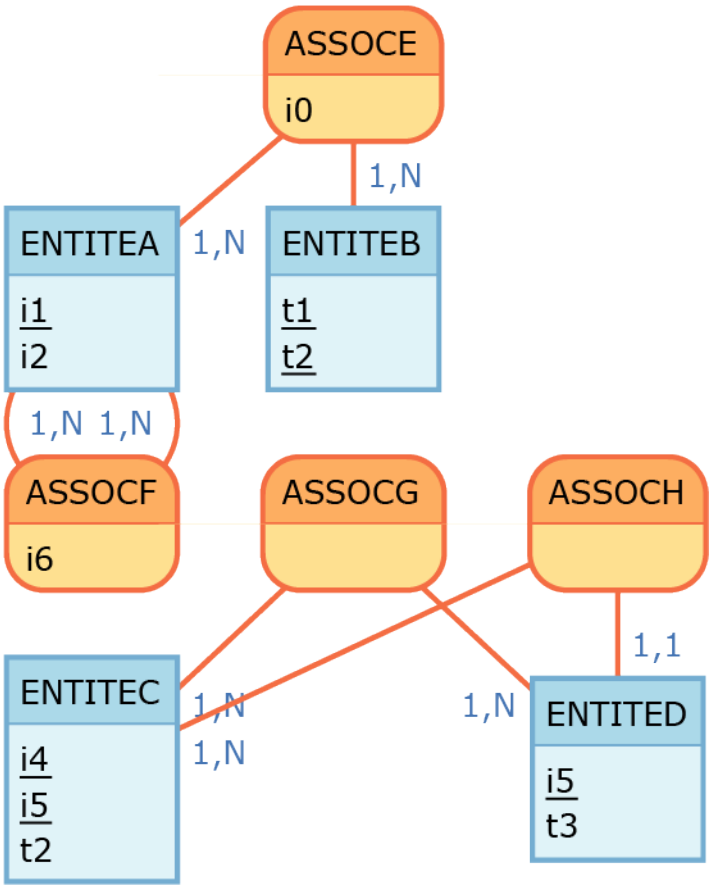
```
with recursive AineIndirect (personne, ainé, distance) as (
select personne,
   enfant,
   1
from PremierNe
union
select ai.personne,
       a.ainé,
       ai.distance+1
from PremierNe a
join AineIndirect ai
on ai.ainé=a.personne)

select *
from AineIndirect
where personne = (select identifiant
                  from personne
                  where datenaissance =
```

```
                      (select min(datenaissance)
                       from personne))
order by distance desc;
```

Question 5: Passage EA vers relationnel 1 . . . . . 3 points

Une future base de données a été modélisée par le diagramme Entités Associations suivant:



Pour chaque entité, les attributs de la clé primaire ont été soulignés. Les attributs dont le nom commence par i sont des entiers , les attributs dont le nom commence par t sont des varchar(40)

Donnez le sql qui correspond à la création des tables de la base de données relationnelle déduite du diagramme entités associations ci dessus. Précisez les attributs qui forment les clés primaires et les clés étrangères.

Solution:

```
CREATE TABLE ASSOCE (
  i1 integer,
  t1 integer,
  t2 varchar(40),
  i0 integer,
  PRIMARY KEY (i1, t1, t2)
);

CREATE TABLE ENTITEA (
  i1 integer,
  i2 integer,
  PRIMARY KEY (i1)
);

CREATE TABLE ENTITEB (
  t1 integer,
  t2 varchar(40),
  PRIMARY KEY (t1, t2)
);

CREATE TABLE ASSOCF (
  i1 integer,
  i1_1 integer,
  i6 integer,
  PRIMARY KEY (i1, i1_1)
);

CREATE TABLE ASSOCG (
  i4 integer,
  i5 integer,
  i5_1 integer,
  PRIMARY KEY (i4, i5, i5_1)
);

CREATE TABLE ENTITEC (
  i4 integer,
  i5 integer,
  t2 varchar(40),
  PRIMARY KEY (i4, i5)
```

```
);

CREATE TABLE ENTITED (
  i5 integer,
  t3 varchar(40),
  i4 integer,
  i5_1 integer,
  PRIMARY KEY (i5)
);

ALTER TABLE ASSOCE ADD FOREIGN KEY (t1, t2) REFERENCES ENTITEB (t1, t2)
ALTER TABLE ASSOCE ADD FOREIGN KEY (i1) REFERENCES ENTITEA (i1);
ALTER TABLE ASSOCF ADD FOREIGN KEY (i1_1) REFERENCES ENTITEA (i1);
ALTER TABLE ASSOCF ADD FOREIGN KEY (i1) REFERENCES ENTITEA (i1);
ALTER TABLE ASSOCG ADD FOREIGN KEY (i5_1) REFERENCES ENTITED (i5);
ALTER TABLE ASSOCG ADD FOREIGN KEY (i4, i5) REFERENCES ENTITEC (i4, i5)
ALTER TABLE ENTITED ADD FOREIGN KEY (i4, i5_1) REFERENCES ENTITEC (i4,
```