

Tableau de bord / Mes cours / EIIN512B - ECUE Bases de donnees relationnelles / CC2 / CC2 2021-2022

**Commencé le** mercredi 17 novembre 2021, 09:00

**État** Terminé

**Terminé le** mercredi 17 novembre 2021, 10:20

**Temps mis** 1 heure 20 min

**Points** 8,50/13,00

**Note** **13,08** sur 20,00 (65%)

#### Description

Vous allez devoir interroger les tables agence, client, compte et emprunt qui ont été créées par les requêtes suivantes:

CREATE TABLE **agence** (

idA integer PRIMARY KEY,  
nom character varying(30) NOT NULL,  
ville character varying(30) NOT NULL

);

CREATE TABLE **client** (

idCl integer PRIMARY KEY,  
nom character varying(30) NOT NULL,  
prenom character varying(30) NOT NULL,  
ville character varying(30) NOT NULL

);

CREATE TABLE **compte** (

numCo integer NOT NULL,  
idA integer NOT NULL REFERENCES agence,  
idCl integer NOT NULL REFERENCES client,  
solde integer NOT NULL,  
PRIMARY KEY(numCo,idA)

);

CREATE TABLE **emprunt** (

IdE integer PRIMARY KEY,  
idA integer NOT NULL references agence,  
idCl integer NOT NULL references client,  
montant integer NOT NULL,  
dateE date NOT NULL

);

- Une AGENCE est décrite par son Identifiant idA, son nom et sa ville.
- Un CLIENT est décrit par son Identifiant idCl, son nom, son prénom et sa ville.
- Un COMPTE est décrit par son Numero numCo, l'identifiant idA de l'agence auquel il est rattaché, l'identifiant idCl du client à qui il appartient et son solde.
- Un EMPRUNT est décrit par son Identifiant idE, l'identifiant idC du client qui emprunte, le montant emprunté et la date de l'emprunt dateE.

Remarque: la date donne le jour de l'emprunt, il peut bien sur y avoir plusieurs emprunts le même jour.

[Description](#)

Pour avoir tous les points d'une question il faut non seulement que votre réponse passe le test, mais il faut aussi qu'elle soit correcte quelque soit le contenu des tables de la base.

Normalement les données de la base de donnée devraient permettre de repérer la plupart des réponses erronées involontaires, mais vos réponses seront relues, et si une erreur est repérée vous perdrez des points.

En revanche le test ne permet pas de déceler les réponses qui utiliseraient le résultat attendu sur le jeu de données pour "tromper l'ennemi", une telle réponse incorrecte ne rapporterait pas de point, même si le logiciel l'affiche comme correcte

Qui plus est , une telle réponse est une tentative de fraude et comme telle vous conduirait tout droit au conseil de discipline.

## Question 1

Correct

Note de 1,00 sur 1,00

Identifiants des agences classés par ordre croissant

pour toutes les agences qui abritent au moins un compte d'un client qui ne réside pas dans la ville de l'agence.

**Réponse :**

```
1 select distinct agence.idA from client
2 natural join compte
3 inner join agence on agence.idA = compte.idA
4 where agence.ville != client.ville
5 order by 1;
```

	Résultat attendu	Résultat obtenu	
✓	idA ----- 12 24	idA ----- 12 24	✓

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 SELECT DISTINCT idA
2 FROM client
3 JOIN compte USING(idC1)
4 JOIN agence USING(idA)
5 WHERE agence.ville<>client.ville
6 ORDER BY idA;
```

Correct

Note pour cet envoi : 1,00/1,00.

Question 2

Partiellement correct

Note de 0,50 sur 1,00

Pour tous les couples d'agences homonymes, afficher sur trois colonnes intitulées nom, agence1, agence2 leur nom commun et leurs deux identifiants.  
On ne dupliquera pas d'information.

On ordonnera par nom, agence1,agence2.

Réponse :

```
1  /*select a1.nom as 'nom', a1.idA  as 'agence1', a2.idA  as 'agence2'
2  from agence as a1
3  union
4  select * from agence as a2
5  where a1.nom = a2.nom --and a1.idA  != a2.idA
6  order by 1,2,3*/
7
8
9  select *
10 from agence as a1
11 left join agence as a2
12 where a1.nom = a2.nom and a1.idA != a2.idA
13
```

	Résultat attendu			Résultat obtenu					
✖	nom	agence1		idA	nom	ville	idA	nom	✖
	agence2			ville					
	-----	-----	-----	-----	-----	-----	-----	-----	
	-----			-----					
	A1	11	12	11	A1	V1	12	A1	
	A1	11	13	V2					
	A1	11	14	11	A1	V1	13	A1	
	A1	11	15	V3					
	A1	12	13	11	A1	V1	14	A1	
	A1	12	14	V4					
	A1	12	15	11	A1	V1	15	A1	
	A1	13	14	V5					
	A1	13	15	12	A1	V2	11	A1	
	A1	14	15	V1					
	A2	24	25	12	A1	V2	13	A1	
	A3	31	32	V3					
	A3	31	34	12	A1	V2	14	A1	
	A3	31	35	V4					
	A3	32	34	12	A1	V2	15	A1	
	A3	32	35	V5					
	A3	34	35	13	A1	V3	11	A1	
	A4	41	42	V1					
	A4	41	44	13	A1	V3	12	A1	
	A4	42	44	V2					
				13	A1	V3	14	A1	
				V4					
				13	A1	V3	15	A1	
				V5					
				14	A1	V4	11	A1	

	Résultat attendu	Résultat obtenu					
		V1					
		14	A1	V4	12	A1	
		V2					
		14	A1	V4	13	A1	
		V3					
		14	A1	V4	15	A1	
		V5					
		15	A1	V5	11	A1	
		V1					
		15	A1	V5	12	A1	
		V2					
		15	A1	V5	13	A1	
		V3					
		15	A1	V5	14	A1	
		V4					
		24	A2	V4	25	A2	
		V5					
		25	A2	V5	24	A2	
		V4					
		31	A3	V1	32	A3	
		V2					
		31	A3	V1	34	A3	
		V4					
		31	A3	V1	35	A3	
		V5					
		32	A3	V2	31	A3	
		V1					
		32	A3	V2	34	A3	
		V4					
		32	A3	V2	35	A3	
		V5					
		34	A3	V4	31	A3	
		V1					
		34	A3	V4	32	A3	
		V2					
		34	A3	V4	35	A3	
		V5					
		35	A3	V5	31	A3	
		V1					
		35	A3	V5	32	A3	
		V2					
		35	A3	V5	34	A3	
		V4					
		41	A4	V1	42	A4	
		V2					
		41	A4	V1	44	A4	
		V4					
		42	A4	V2	41	A4	
		V1					
		42	A4	V2	44	A4	
		V4					
		44	A4	V4	41	A4	
		V1					
		44	A4	V4	42	A4	
		V2					

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

Montrer les différences

Solution de l'auteur de la question (Sql):

```
1 SELECT nom,
2     A1.idA AS agence1,
3     A2.idA AS agence2
4 FROM agence A1
5 JOIN agence A2
6     USING(nom)
```

```
7 | WHERE A1.idA<A2.idA  
8 | ORDER BY nom, agence1,agence2;
```

**Partiellement correct**

Note pour cet envoi : 0,00/1,00.

Commentaire :

duplication

## Question 3

Incorrect

Note de 0,00 sur 1,00

Identifiants des agences qui ont un client dans la ville V1 et un client dans la ville V3 , triés par ordre croissant

Réponse :

```

1  /*select distinct agence.idA from client
2  natural join compte
3  inner join agence on agence.idA = compte.idA
4  where client.ville = 'V1' or client.ville = 'V3'
5  order by 1;*/
6
7  select * from client
8  natural join compte
9  inner join (select idA,nom from agence) as agence on compte.idA = agence.idA
10 where ville = 'V1' or ville = 'V3'
11 --group by
12 --select * from agence
13
14 --peut etre on peut faire une requete qui prend tout ce qui a au moins V1 puis on selec

```

	Résultat attendu	Résultat obtenu							
✗	idA ----- 12	idCl idA	nom nom	prenom	ville	numCo	idA	solde	✗
		-----	-----	-----	-----	-----	-----	-----	
		---	-----	-----					
		111	N1	P1	V1	1	11	100	
		11	A1						
		111	N1	P1	V1	1	12	-100	
		12	A1						
		113	N1	P1	V3	2	12	100	
		12	A1						
		111	N1	P1	V1	3	12	-200	
		12	A1						
		453	N4	P5	V3	5	12	200	
		12	A1						
		453	N4	P5	V3	1	13	200	
		13	A1						
		453	N4	P5	V3	1	14	200	
		14	A1						
		453	N4	P5	V3	1	15	200	
		15	A1						

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

Montrer les différences

Solution de l'auteur de la question (Sql):

```

1  SELECT idA
2  FROM client
3  JOIN compte USING(idCl)
4  JOIN agence USING(idA)
5  WHERE client.ville='V1'
6  INTERSECT
7  SELECT idA
8  FROM client

```

```
9 | JOIN compte USING(idCl)  
10 | JOIN agence USING(idA)  
11 | WHERE client.ville='V3'  
12 | ORDER BY 1;
```

**Incorrect**

Note pour cet envoi : 0,00/1,00.



## Question 4

Correct

Note de 1,00 sur 1,00

nom et ville de tous les clients, classés par ville puis par nom

Réponse :

```
1 | select nom, ville from client
2 | order by 2,1
```

	Résultat attendu		Résultat obtenu		
✓	nom	ville	nom	ville	✓
	-----	-----	-----	-----	
	N1	V1	N1	V1	
	N2	V1	N2	V1	
	N2	V1	N2	V1	
	N1	V2	N1	V2	
	N3	V2	N3	V2	
	N4	V2	N4	V2	
	N1	V3	N1	V3	
	N4	V3	N4	V3	
	N1	V4	N1	V4	

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 | SELECT nom, ville
2 | FROM client
3 | ORDER BY ville, nom;
```

Correct

Note pour cet envoi : 1,00/1,00.

## Question 5

Incorrect

Note de 0,00 sur 1,00

Identifiants et noms des agences auxquelles ne sont rattachés que des comptes à solde positif ou nul

Trier par identifiant d'agence puis par nom d'agence

Un agence sans compte doit apparaître dans le résultat.

## Réponse :

```

1  /*select idA, nom from agence as a
2  natural join compte
3  where solde in (
4      select solde from compte
5      where a.idA = compte.idA and solde >= 0
6  )*/
7
8  select * from compte
9  natural join agence
10
11 --order by nom, idA
12

```

	Résultat attendu		Résultat obtenu						
✗	idA	nom	numCo	idA	idCl	solde	nom	ville	✗
	-----	-----	-----	-----	-----	-----	-----	-----	
	-		-						
	11	A1	1	11	111	100	A1	V1	
	13	A1	1	12	111	-100	A1	V2	
	14	A1	2	12	113	100	A1	V2	
	15	A1	1	24	112	100	A2	V4	
	24	A2	1	44	114	-100	A4	V4	
	25	A2	1	32	322	100	A3	V2	
	31	A3	3	12	111	-200	A1	V2	
	32	A3							
	34	A3							
	35	A3							
	41	A4							
	42	A4							

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

Montrer les différences

## Solution de l'auteur de la question (Sql):

```

1  SELECT idA, nom
2  FROM agence
3  EXCEPT
4  SELECT idA, agence.nom
5  FROM agence
6  JOIN compte USING(idA)
7  WHERE solde < 0
8  ORDER BY 1,2;

```

Incorrect

Note pour cet envoi : 0,00/1,00.

## Question 6

Correct

Note de 1,00 sur 1,00

Donner toutes les clients avec leurs emprunts en affichant

- identifiant du client
- nom du client
- identifiant de l'emprunt
- montant

Un client sans emprunt ne doit pas apparaitre dans le résultat

Trier par identifiant de client , puis par identifiant d'emprunt

La deuxième colonne sera titrée "Nom du client" .

**Réponse :**

```
1 select idCl, client.nom as 'Nom du client', IdE, montant from client
2 natural join emprunt
3 where montant not null
```

	Résultat attendu				Résultat obtenu				
✓	idCl	Nom du client	IdE	montant	idCl	Nom du client	IdE	montant	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	----				----				
	111	N1	110111	200	111	N1	110111	200	
	111	N1	120111	300	111	N1	120111	300	
	112	N1	240112	200	112	N1	240112	200	
	452	N4	340452	300	452	N4	340452	300	

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 SELECT idCl, nom as "Nom du client",idE, montant
2 FROM
3 client
4 JOIN emprunt
5 USING(idCl)
6 ORDER BY idCl,idE;
```

Correct

Note pour cet envoi : 1,00/1,00.

## Question 7

Correct

Note de 1,00 sur 1,00

nom des villes , et nombre d'agences dans la ville, pour toutes les villes ayant au moins une agence

Trier par nombre d'agences croissant

La deuxième colonne sera titrée "Nombre d'agences"

## Réponse :

```
1 | select distinct ville, count(*) as "Nombre d'agences" from agence
2 | natural join agence
3 | group by 1
4 | order by 2
```

	Résultat attendu		Résultat obtenu		
✓	ville	Nombre d'agences	ville	Nombre d'agences	✓
	-----		-----		
	V3	1	V3	1	
	V1	3	V1	3	
	V2	3	V2	3	
	V5	3	V5	3	
	V4	4	V4	4	

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 | SELECT ville,
2 |     count(idA) as "Nombre d'agences"
3 | FROM agence
4 | GROUP BY ville
5 | ORDER BY "Nombre d'agences" ;
```

Correct

Note pour cet envoi : 1,00/1,00.

## Question 8

Correct

Note de 1,00 sur 1,00

Afficher par ordre décroissant les identifiants des clients n'ayant aucun compte

Réponse :

```
1 select idCl from (  
2     select * from client  
3  
4     except  
5  
6     select idCl, nom, prenom, ville from client  
7     natural join compte  
8  
9     order by 1 desc  
10 )
```

	Résultat attendu	Résultat obtenu	
✓	idCl ----- 453 452 231 221	idCl ----- 453 452 231 221	✓

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 SELECT idCl  
2 FROM client  
3 WHERE idCl NOT IN (SELECT idCl FROM compte)  
4 ORDER BY idCl DESC;
```

Correct

Note pour cet envoi : 1,00/1,00.

## Question 9

Correct

Note de 1,00 sur 1,00

Identifiants et noms des clients, ainsi que leur nombre de comptes, et la somme des soldes de leur comptes (agrégat SUM)

Un client sans compte ne doit pas apparaître dans le résultat

Trier par identifiant de client

La troisième colonne sera titrée "Nombre de comptes" et la dernière "Total des comptes"

## Réponse :

```
1 select idCl, nom, count(*) as "Nombre de comptes",
2 sum(solde) as "Total des comptes" from client
3 natural join compte
4 group by idCl
```

	Résultat attendu			Résultat obtenu			
✓	idCl	nom	Nombre de comptes	idCl	nom	Nombre de comptes	✓
	Total des comptes			Total des comptes			
	-----	-----	-----	-----	-----	-----	
	111	N1	3	111	N1	3	
	-200			-200			
	112	N1	2	112	N1	2	
	113	N1	1	113	N1	1	
	100			100			
	114	N1	2	114	N1	2	
	800			800			
	322	N3	2	322	N3	2	
	150			150			
	453	N4	4	453	N4	4	
	800			800			

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```
1 SELECT idCl,
2     nom,
3     count(numCo) as "Nombre de comptes",
4     sum(solde) as "Total des comptes"
5 FROM client
6 JOIN compte
7 USING(idCl)
8 GROUP BY idCl,nom
9 ORDER BY idCl;
```

Correct

Note pour cet envoi : 1,00/1,00.



## Question 10

Incorrect

Note de 0,00 sur 2,00

Trouver les identifiants des clients qui ont un compte dans chaque ville où il y a au moins une agence  
On classera les identifiants par ordre croissant

Réponse :

```
1  /*select distinct ville from agence;
2
3
4
5  select * from client
6  natural join compte
7  inner join agence on compte.idA = agence.idA
8  group by agence.ville*/
9
10 select * from client
11 natural join compte
12 inner join (select idA,nom from agence) as agence on compte.idA = agence.idA
13 group by agence.idA
14
15
16 ;select * from agence
```

Résultat attendu	Résultat obtenu	
------------------	-----------------	--

	Résultat attendu	Résultat obtenu							
✖	idCl	idCl	nom	prenom	ville	numCo	idA	solde	✖
	-----	idA	nom						
	453	-----	-----	-----	-----	-----	-----	-----	
		---	-----	-----					
		111	N1	P1	V1	1	11	100	
		11	A1						
		111	N1	P1	V1	1	12	-100	
		12	A1						
		453	N4	P5	V3	1	13	200	
		13	A1						
		453	N4	P5	V3	1	14	200	
		14	A1						
		453	N4	P5	V3	1	15	200	
		15	A1						
		112	N1	P1	V2	1	24	100	
		24	A2						
		322	N3	P3	V2	1	32	100	
		32	A3						
		114	N1	P1	V4	1	44	-100	
		44	A4						
		idA	nom	ville					
		-----	-----	-----					
		11	A1	V1					
		12	A1	V2					
		13	A1	V3					
		14	A1	V4					
		15	A1	V5					
		24	A2	V4					
		25	A2	V5					
		31	A3	V1					
	32	A3	V2						
	34	A3	V4						
	35	A3	V5						
	41	A4	V1						
	42	A4	V2						
	44	A4	V4						

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

## Montrer les différences

Solution de l'auteur de la question (Sql):

```
1 SELECT idCl
2 FROM client
3 EXCEPT
4 SELECT idCl FROM
5 (SELECT idCl, agence.ville FROM client, agence
6 EXCEPT
7 SELECT idCl, agence.ville FROM compte JOIN agence USING(idA)) AS T
```

**Incorrect**

Note pour cet envoi : 0,00/2,00.

## Question 11

Correct

Note de 2,00 sur 2,00

nom et ville du ou des client(s) ayant emprunté le plus d'argent (dont la somme des montants des emprunts est maximum)

On rappelle l'existence de l'agrégat SUM

Réponse :

```

1 select nom, ville from (
2     select nom, ville, max("Total des comptes") from (
3         select idCl, nom, ville, count(*) as "Nombre de comptes",
4             sum(solde) as "Total des comptes" from client
5         natural join compte
6     group by idCl
7 )
8 )

```

	Résultat attendu		Résultat obtenu		
✓	nom	ville	nom	ville	✓
	-----	-----	-----	-----	
	N4	V3	N4	V3	

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Sql):

```

1 WITH CC AS
2 (SELECT idCl, sum(montant) as NB
3 FROM emprunt
4 GROUP BY idCl)
5
6 SELECT nom, ville
7 FROM client JOIN CC AS CC1
8 USING(idCl)
9 WHERE CC1.NB =(SELECT Max(CC2.NB) FROM CC AS CC2)
10

```

Correct

Note pour cet envoi : 2,00/2,00.

◀ Description de la base de données

Aller à...

CC3 2020 QCM ▶