

Evaluation Base de données

Échéance 22 Oct à 13:00 **Points** 20 **Questions** 24

Disponibles 22 Oct à 8:30 - 22 Oct à 13:00 5 heures environ

Limite de temps 75 minutes

Instructions

Ce questionnaire a pour but d'évaluer les connaissances de l'étudiant dans le domaine de la conception et l'implementation d'une base de données (Relationnelle).

Fait avec amour pour vous mes très chers fellows

Note pour ce questionnaire: **13,5** sur 20

Envoyé le 22 Oct à 11:12

Cette tentative a duré 53 minutes.

Question 1

1 / 1 points

Lors de la conception d'une base de données, quels sont les cinq concepts qui régissent le modèle Entité/Association ?



Cardinalité, Dépendance fonctionnelle, Association, Attribut, Forme Normale



Identifiant, Entité, Attribut, Association, Cardinalité



Entité, Dépendance fonctionnelle, Identifiant, Cardinalité, Forme normale



Attribut, Association, Identifiant, Clé étrangère, Forme normale

Question 2

1 / 1 points

Une relation est en première forme normale si tous ses attributs sont atomiques (inhérent au modèle relationnel). En considérant la relation suivante :

Personne(#Nom, Profession)

Parmi les exemples ci-dessous, lequel traduit le mieux ce concept ?

☐ Aucune des réponses n'est en 1ere Forme Normale

☒ (Dupont, Géomètre)

☐ (Durand, Ingénieur-Professeur)

Question 3

0 / 1 points

La deuxième forme normale permet d'éliminer les dépendances entre des parties de clé et des attributs n'appartenant pas à une clé. Parmi les réponses ci-dessous, laquelle est en 2ème Forme Normale ?

☐ Aucune des réponses n'est en 2eme Forme Normale

☒ Personne(#Nom, #Profession, Salaire)

On note alors que la première DF est issue de la clé et qu'elle n'est pas élémentaire (puisque Profession détermine Salaire) et donc que le schéma n'est pas en 2NF.

☐ Personne(#Nom, #Profession=>Profession, Prenom) Profession(#Profession, Salaire)

Question 4

1 / 1 points

La troisième forme normale permet d'éliminer les dépendances entre les attributs n'appartenant pas à une clé. Parmi les réponses ci-dessous, laquelle est en 3 forme normale ?

☒ Profession(#Profession, Salaire=>Salaire) Salaire(#Salaire, Prime)

Ce schéma est en 3NF, car Prime est maintenant déterminé par une clé.

On remarque que cette décomposition préserve les DF, car par transitivité, Profession détermine Salaire qui détermine Prime, et donc Profession détermine toujours Prime.

- ☐ Aucune des réponses n'est en 3eme Forme Normale
- ☐ Profession(#Profession, Salaire, Prime)

Question 5

0 / 1 points

Quelles clauses placées après le type de donnée d'un champ permettent de vérifier que le contenu de ce champ est unique ?

☒ DEFAULT

☒ UNIQUE

☐ CONSTRAINT

☐ PRIMARY KEY

☐ NOT NULL

Question 6

1 / 1 points

Les 5 commandes de base en SQL sont

☐ CREATE, SELECT, INSERT, MODIFY, DELETE

☐ Aucune des réponses ci-dessus

☒ CREATE, SELECT, INSERT, UPDATE, DELETE

☐ CREATE, ALTER, INSERT, UPDATE, DELETE

Question 7**1 / 1 points**

Quelle est l'utilité de l'option NOT NULL placée après le type de donnée d'un champ ?

- ☐ interdire la valeur 0 dans un champ
- ☐ permettre l'absence de contenu dans un champ
- ☒ obliger l'entrée de contenu dans un champ
- ☐ interdire les espaces dans un champ

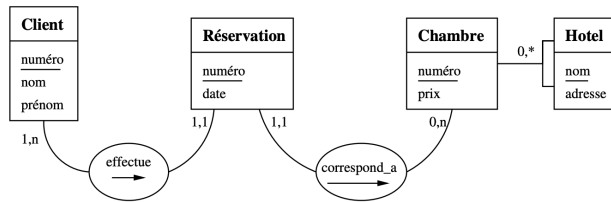
Question 8**0 / 1 points**

Les instructions **DML** sont des instructions SQL manipulant des données. **DML** signifie **D**ata **M**anipulation **L**anguage. Parmi les instructions SQL suivantes, lesquelles sont **DML** ?

- ☒ CREATE, DROP, ALTER, RENAME, TRUNCATE
- ☐ START, TRANSACTION, COMMIT, ROLLBACK
- ☐ GRANT, REVOKE
- ☐ SELECT, INSERT, UPDATE, DELETE

Question 9**0 / 0.5 points**

On souhaite gérer des réservations dans une compagnie d'hôtels. A cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



Est-il possible d'avoir des clients homonymes ?

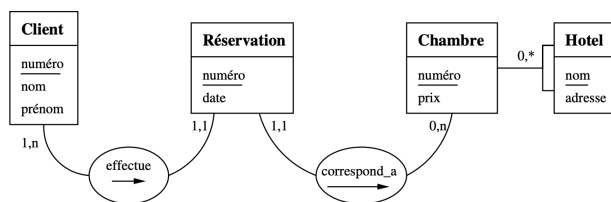
☐ Vrai

☒ Faux

Question 10

0,5 / 0.5 points

On souhaite gérer des réservations dans une compagnie d'hôtels. A cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



Un client peut-il réserver plusieurs chambres à une date donnée ?

☒ Vrai

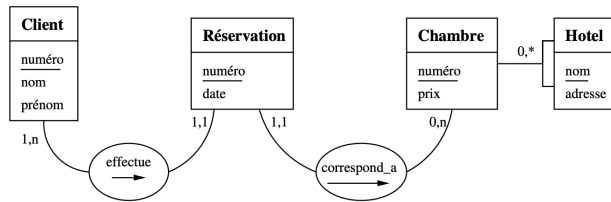
oui, mais il faut faire plusieurs réservations

☐ Faux

Question 11

0 / 0.5 points

On souhaite gérer des réservations dans une compagnie d'hôtels. A cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



Est-il possible de réserver une chambre sur plusieurs jour ?

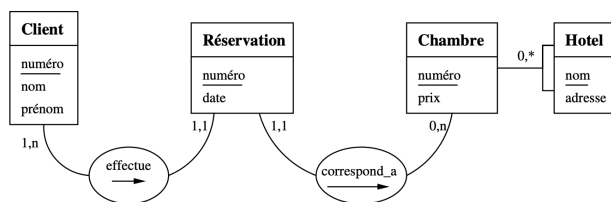
☐ Vrai

☒ Faux

Question 12

0 / 0.5 points

On souhaite gérer des réservations dans une compagnie d'hôtels. A cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



Est-il possible de savoir si une chambre est libre à une date donnée ?

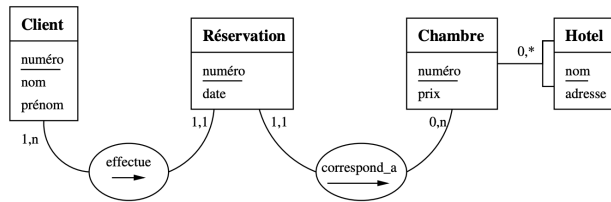
☐ Vrai

☒ Faux

Question 13

0,5 / 0.5 points

On souhaite gérer des réservations dans une compagnie d'hôtels. A cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



Est-il possible de réserver plusieurs fois une chambre à une date donnée ?

☒ Vrai

oui, rien ne garanti que la même chambre puisse être louée plusieurs fois

☐ Faux

Question 14

1 / 1 points

Les instructions **DDL** sont des instructions SQL qui traitent simplement de la description du schéma de base de données où il crée et modifie la structure des objets de base de données sur la base de données. Parmi les instructions suivantes lesquelles sont de types **DDL** ?

☒ CREATE, DROP, ALTER, RENAME, TRUNCATE

☐ GRANT, REVOKE

☐ START, TRANSACTION, COMMIT, ROLLBACK

☐ SELECT, INSERT, UPDATE, DELETE

Question 15

1 / 1 points

On considère la description suivante :

- Client(**numero**,nom,prenom)
- Hotel(**nom**,adresse)
- Chambre(**numero**,prix,**nomhotel**)
- Reservation(**numero**,date,numclient, nomhotel,numchambre)

Parmi les réponses suivantes quelle requête répond a la question **"trouver les clients ayant réservé plusieurs chambres dans le même hôtel le 15 avril 2006"**

Les champs en gras représentent la clé primaire.

☐ Impossible faire cette requête, il y a une erreur dans l'enoncé

☐ `SELECT numclient, COUNT(*) AS narres FROM Reservation GROUP BY numclient, nom hotel`

☒ `SELECT DISTINCT numclient FROM Reservation GROUP BY numclient, nomhotel HAVING COUNT(*) > 1`

Question 16

1 / 1 points

On considère la description suivante :

- Client(**numero**,nom,prenom)
- Hotel(**nom**,adresse)
- Chambre(**numero**,prix,**nomhotel**)
- Reservation(**numero**,date,numclient, nomhotel,numchambre)

Parmi les réponses suivantes quelle requête répond a la question **"donner les chambres disponibles le 15 avril 2006 à l'hôtel "Belaventure". "**

Les champs en gras représentent la clé primaire.

☒ `SELECT numero FROM Chambre WHERE nomhotel='Belaventure' AND numero NOT IN (SELECT numchambre FROM Reservation WHERE nomhotel='Belaventure' AND date = '15/04/2006')`

☐ `SELECT numero FROM Chambre WHERE nomhotel='Belaventure' AND numero IN (SELECT numchambre FROM Reservation WHERE nomhotel='Belaventure' AND date = '15/04/2006')`

☐ Impossible faire cette requête, il y a une erreur dans l'enoncé

Question 17**0,5 / 0.5 points**

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000
3	3	01-Feb-13	2000
4	1	01-Feb-25	8000

Donner le script SQL de creation de la table employes



```
CREATE TABLE employes (Employee_id INT NOT NULL, Nom
VARCHAR(255) NOT NULL,Prenom VARCHAR(255) NOT NULL, Salaire INT
NOT NULL, Date_entree DATE NOT NULL, Departement VARCHAR(255),
PRIMARY KEY(Employee_id) )
```



```
ADD NEW TABLE employes (Employee_id INT NOT NULL, Nom
VARCHAR(255) NOT NULL, Prenom VARCHAR(255) NOT NULL, Salaire INT
NOT NULL, Date_entree DATE NOT NULL, Departement VARCHAR(255),
PRIMARY KEY(Employee_id) )
```



```
ALTER TABLE employes (Employee_id INT NOT NULL, Nom VARCHAR(255)
NOT NULL, Prenom VARCHAR(255) NOT NULL, Salaire INT NOT NULL,
Date_entree DATE NOT NULL, Departement VARCHAR(255), PRIMARY
KEY(Employee_id) )
```

Question 18

0,5 / 0.5 points

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000

.	3	.	3	01-Feb-13	2000
	4	.	1	01-Feb-25	8000

Donner le script SQL de creation de la table revenus



```
CREATE TABLE revenus (Revevnus_id INT NOT NULL, Employe_ref_id INT NOT NULL,date DATE NOT NULL, amount INT NOT NULL, PRIMARY KEY(Revenus_id), FOREIGN KEY (Employe_ref_id) REFERENCES employes(Employe_id) )
```



```
db.createCollection('revenus')
```



```
ALTER TABLE revenus (Revevnus_id INT NOT NULL, Employe_ref_id INT NOT NULL,date DATE NOT NULL, amount INT NOT NULL, PRIMARY KEY(Revenus_id), FOREIGN KEY (Employe_ref_id) REFERENCES employes(Employe_id) )
```

Question 19

1 / 1 points

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT

```

|      7      |      Yohan      |      Soso      |      1230000      |      01-JAN-24      |      Banking
|
+-----+-----+-----+-----+-----+-----+
+

```

Table revenus

```

+-----+-----+-----+-----+
| Revenus_id | Employee_ref_id | date      | amount |
+-----+-----+-----+-----+
| .    1    |      1      | 01-JAN-11 | 1000 |
| .    2    |      2      | 01-Feb-20 | 5000 |
| .    3    |      3      | 01-Feb-13 | 2000 |
|      4    |      1      | 01-Feb-25 | 8000 |
+-----+-----+-----+-----+

```

Afficher le nom et le prenom des employes

- ☐ employes FIND Nom,Prenom
- ☒ SELECT Nom,Prenom FROM employes
- ☐ MATCH Nom,Prenom FROM employes
- ☐ SELECT * FROM employes

Question 20

0,5 / 0.5 points

Soit les tables suivantes :

Table employes

```

+-----+-----+-----+-----+-----+-----+
+
| Employee_id | Nom .      | Prenom    | Salaire | Date_entree | Departement
|
+-----+-----+-----+-----+-----+-----+
+
|      1      |      Bob   |      Kinto | 1000000 | 01-JAN-13   |      Finance
|
|      2      |      Jerry |      Kansxo | 6000000 | 01-Feb-16   |      IT
|
|      3      |      Philip |      Jose  | 8900000 | 01-May-23   |      Banking
|
|      4      |      John  |      Abraham | 2000000 | 01-JAN-14   |      Insurance

```

5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000
3	3	01-Feb-13	2000
4	1	01-Feb-25	8000

Afficher la liste des employés du département 'Banking'

- ☒ SELECT * FROM employees WHERE Department = 'Banking'
- ☐ SELECT * FROM Banking WHERE Departement = 'Banking'
- ☐ MATCH (n) RETURN n;

Question 21**0 / 1 points**

Soit les tables suivantes :

Table employees

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking

4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000
3	3	01-Feb-13	2000
4	1	01-Feb-25	8000

Afficher le min, le max et la moyenne des salaires par département

```
SELECT MIN(Salaire),Max(Salaire),AVG(Salaire) FROM employes GROUP BY Departement
```



```
SELECT MINIMUM(Salaire),MAXIMUM(Salaire),MOYENNE(Salaire) FROM employes
```



```
SELECT Departement, MIN(Salaire),Max(Salaire),AVG(Salaire) FROM employes
```



```
SELECT Departement, MIN(Salaire),Max(Salaire),AVG(Salaire) FROM employes GROUP BY Departement
```

Question 22**1 / 1 points**

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000
3	3	01-Feb-13	2000
4	1	01-Feb-25	8000

Afficher les departements ayant plus d'un employé classés par ordre décroissant

☐ MATCH Departement, SUM(*) FROM employes WHERE SUM(*) > 1

☒ SELECT Departement, COUNT(*) AS NbrEmp FROM employes GROUP BY Departement HAVING COUNT(*) > 1 ORDER BY NbrEmps DESC



```
SELECT Departement, SUM(*) AS NbrEmp FROM employes GROUP BY
Departement HAVING COUNT(*) > 1 ORDER BY NbrEmps DESC
```

Question 23

1 / 1 points

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
1	1	01-JAN-11	1000
2	2	01-Feb-20	5000
3	3	01-Feb-13	2000
4	1	01-Feb-25	8000

Afficher pour chaque employé le nom, le salaire et le montant de ses revenus et leurs date. Afficher NULL pour ceux qui n'ont pas de revenu.



```
SELECT emp.Nom, emp.Salaire, rev.date, SUM(rev.amount) FROM employes
emp LEFT JOIN Revenus rev ON emp. Employee_id = rev. Employee_ref_id
GROUP BY emp.Nom, emp.Salaire, rev.date
```



```
SELECT emp.Nom, emp.Salaire, rev.date, rev.amount FROM employes emp
LEFT JOIN Revenus rev ON emp. Employee_id = rev. Employee_ref_id
GROUP BY emp.Nom, emp.Salaire, rev.date
```



Une erreur dans l'enonce il n'est pas possible d'executer cette requête SQL

Question 24

0 / 1 points

Soit les tables suivantes :

Table employes

Employee_id	Nom	Prenom	Salaire	Date_entree	Departement
1	Bob	Kinto	1000000	01-JAN-13	Finance
2	Jerry	Kansxo	6000000	01-Feb-16	IT
3	Philip	Jose	8900000	01-May-23	Banking
4	John	Abraham	2000000	01-JAN-14	Insurance
5	Michael	Mathew	2200000	01-Feb-16	Finance
6	Alex	chreketo	4000000	01-JAN-05	IT
7	Yohan	Soso	1230000	01-JAN-24	Banking

Table revenus

Revenus_id	Employee_ref_id	date	amount
------------	-----------------	------	--------

+	-----+	+	-----+	+	-----+
	. 1	1		01-JAN-11	1000
	. 2 .	2		01-Feb-20	5000
	. 3 .	3		01-Feb-13	2000
	4 .	1		01-Feb-25	8000
+	-----+	+	-----+	+	-----+

Afficher pour chaque employé la somme de ses revenus



Une erreur dans l'enonce il n'est pas possible d'executer cette requête SQL



SELECT emp.Nom, SUM(rev.amount) FROM employes emp LEFT JOIN
Revenus rev ON emp. Employee_id = rev. Employee_ref_id GROUP BY
emp.Nom



SELECT emp.Nom, emp.Salaire, rev.date, rev.amount FROM employes emp
LEFT JOIN Revenus rev ON emp. Employee_id = rev. Employee_ref_id
GROUP BY emp.Nom, emp.Salaire, rev.date

Note du questionnaire: **13,5** sur 20