

Informatique du Tronc commun TD n°1 Les bases de données – Corrigé

Exercice n°1 Publications scientifiques

Une base de données concernant des publications scientifiques doit être constituée. Pour modéliser la situation, trois types d'entité sont considérés : Scientifique, Article et Revue.

Un scientifique peut écrire un ou plusieurs articles. Un article peut être écrit par un ou plusieurs scientifiques et paraître dans une ou plusieurs revues.

Les données enregistrées sont le nom, le prénom et la date de naissance du scientifique, le titre et la date de fin d'écriture de l'article (ou date d'envoi aux différentes revues pour demander une publication), la date de parution dans une revue, le nom de la revue.

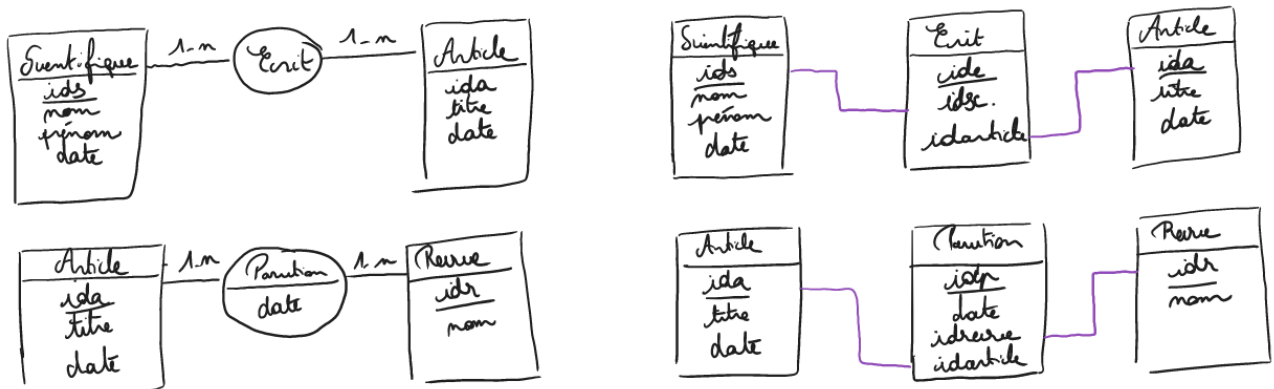
R1. Décrire les types d'associations entre les types d'entités en donnant les cardinalités.

Solution: Entre Scientifique et Article il y a un type d'association Écrit (ou A écrit). Un scientifique peut écrire un ou plusieurs articles, donc la cardinalité entre Scientifique et Écrit est 1-n. Un article peut être écrit par un ou plusieurs scientifiques, donc la cardinalité entre Article et Écrit par est 1-n. Il s'agit d'une association **

Entre Article et Revue il y a un type d'association Paraître (ou Publié par). Un article peut paraître dans une ou plusieurs revues, donc la cardinalité entre Article et Publié par est 1-n. Une revue peut plusieurs un ou plusieurs articles, donc la cardinalité entre Paraître et Revue est 1-n. L'association entre Article et Revue est également de type **

R2. En déduire les tables utilisées dans le modèle relationnel. Préciser les schémas de relation avec les attributs, les clés primaires et étrangères.

Solution:



L'association entre Scientifique et Article doit être décomposée en deux associations 1-* ce qui produit trois tables : Scientifique, Article et Écrit

L'association entre Article et Revue doit être décomposée en deux associations 1-* ce qui produit trois tables : Article, Revue et Parution

Schémas de relations :

Scientifique (ids, nom, prénom, naissance)

Article (ida, titre, date_écrit)

Écrit (ide, idarticle, idscientifique)

Revue (idr, nom)

Parution (idp, idrevue, idarticle, date)

`idarticle` et `idscientifique` de la table `Écrit` sont les clés étrangères en référence avec les clés primaires `ida` et `ids` des tables `Article` et `Scientifique`.
`idrevue` et `idarticle` sont les clés étrangères de la table `Parution` sont les clés étrangères en référence avec les clés primaires `idr` et `ida` des tables `Revue` et `Article`.

Exercice n°2 Bibliothèque

Une bibliothèque d'un établissement scolaire doit être gérée à l'aide d'une base de données. Pour concevoir un modèle, on considère deux types d'entité : `Livre` et `Élève`.

Le type d'entité `Livre` a pour attributs `Id`, `Titre`, `Auteur`.

Le type d'entité `Élève` a pour attributs `Id`, `Nom`, `Prénom`, `Classe`.

On considère un type d'association nommé `Emprunte` entre `Livre` et `Élève`. Un élève peut emprunter aucun, un ou plusieurs livres. Un livre est emprunté par un unique élève.

R1. Préciser le type d'association 1-1, 1-* ou *-*.

Solution: Le type d'association est 1-*, qui donne donc deux tables dans le schéma relationnel, en ajoutant une clé étrangère dans la table `Livre` en relation avec la clé primaire `Id` de la table `Élève`.

R2. Définir le schéma relationnel en précisant pour chaque table la clé primaire et d'éventuelles clés étrangères.

Solution:

`Élève` (`Id`, `Nom`, `Prénom`, `Classe`)

`Livre` (`Id`, `Titre`, `Auteur`, `idélève`), avec `idélève` une clé étrangère en référence avec la clé primaire `Id` de la table `Élève`, qui fait la relation entre les deux tables.

R3. Peut-on enregistrer deux élèves qui ont le même nom et le même prénom dans la table `Élève` ?

Solution: Il est possible d'enregistrer deux élèves qui ont le même nom et le même prénom, ils le seront avec deux valeurs de `Id` différentes, qui est la clé primaire qui permet d'identifier deux élèves.

On suppose maintenant qu'un livre est disponible en plusieurs exemplaires, donc peut être emprunté par un ou plusieurs élèves en même temps.

R4. Décrire les modifications concernant le modèle entité-association.

Solution: Le type d'association est maintenant *-*.

R5. Définir le schéma relationnel en précisant pour chaque table la clé primaire et les éventuelles clés étrangères.

Solution: Il faut maintenant trois tables. On peut ajouter une table `Emprunt` avec des clés étrangères qui vont faire le lien entre la table `Livre` et la table `Élève`

`Élève` (`Id`, `Nom`, `Prénom`, `Classe`)

`Livre` (`Id`, `Titre`, `Auteur`)

`Emprunt`(`Id`, `Idélève`, `Idlivre`, `date_emprunt`, `date_retour`)

Un auteur peut avoir écrit plusieurs livres présents dans la base et on souhaite donc éviter la redondance d'informations en n'enregistrant pas plusieurs fois le même nom dans la table `Livre`.

R6. Comment peut-on compléter/modifier le schéma relationnel ?

Solution: On peut ajouter une relation Auteur :Auteur (Id,Nom, Prénom)

Élève (Id, Nom, Prénom, Classe

Et on ajoute dans la table Livre, une clé étrangère idauteur, en relation avec la clé primaire Id de la table Auteur. Livre (Id, Titre, idauteur)

Emprunt(Id, Idélève, Idlivre, date_emprunt,date_retour)

Un livre peut avoir été écrit en commun par plusieurs auteurs.

R7. Comment peut-on compléter/modifier le schéma relationnel ?

Solution: L'association entre Livre et Auteur est maintenant de type ***, il faut donc ajouter une table pour gérer cela : Écrit (Id,Idlivre,Idauteur), avec Idlivre et Idauteur des clés étrangères en référence aux clés primaires Id des tables Livre et Auteur.

Exercice n°3 Élevage d'animaux

On étudie une base de données constituées des cinq tables :

- Animal (id : INT , sexe : CHAR, date_naissance : DATE, nom : CHAR, espece_id : INT, race_id : INT, mere_id : INT, pere_id : INT, disponible : INT)
- Client (id : INT, nom, prenom, adresse, code_postal : INT, ville, pays, email, date_naissance : DATE)
- Adoption (client_id, animal_id, date_reservation : DATE, date_adoption : DATE, prix : FLOAT, paye : INT) qui fait le lien entre les entités animaux et les entités clients.
- Espèce (id, nom_courant, nom_latin, description, prix : FLOAT)
- Race (id, nom, espece_id, description, prix)

Les types « évidents » ne sont pas indiqués.

CHAR indique qu'il s'agit d'une chaîne de caractères.

DATE : chaîne de caractères au format 'aaaa-mm-jj'

Les deux attributs disponible et paye sont des entiers, qui valent 0 ou 1. Pour disponible : 0 s'il ne l'est pas, 1 s'il l'est. Pour paye : 0 s'il n'a pas été payé, 1 s'il a été payé.

nom_courant : chat, chien, ...

Écrire les requêtes permettant d'obtenir :

R1. les noms des animaux classés par date de naissance,

Solution:

```
1 SELECT nom
2 FROM Animal
3 ORDER BY date_naissance
```

R2. les trois animaux les plus jeunes,

Solution:

```
1 SELECT nom
2 FROM Animal
3 ORDER BY date_naissance DESC
4 LIMIT 3
```

R3. les trois suivants,

Solution:

```
1 SELECT nom
2 FROM Animal
3 ORDER BY date_naissance DESC
4 LIMIT 3
5 OFFSET 3
```

R4. les espèces qui coûtent moins de 150 €,

Solution:

```
1 SELECT nom_courant
2 FROM Espèce
3 WHERE prix < 150
```

R5. les noms des animaux nés en 2012,

Solution:

```

1 SELECT nom
2 FROM Animal
3 WHERE DATE >= '2012-01-01' AND DATE <= '2012-12-31'
4 # Hors programme, mais si LIKE est décrit dans le sujet de concours
5 SELECT nom
6 FROM Animal
7 WHERE DATE LIKE '2012%'

```

R6. le nom de l'espèce de chaque animal,

Solution:

```

1 SELECT nom, nom_courant
2 FROM Espèce
3 JOIN Animal
4 ON espece_id=Espèce.id

```

R7. le nom, le nom de l'espèce, le nom de la race pour les chiens et les chats,

Solution:

```

1 SELECT Animal.nom, nom_courant, RACE.nom
2 FROM Animal
3 JOIN Espèce
4 JOIN Race
5 ON espece_id=Espèce.id AND race_id=Race.id
6 WHERE nom_courant IN ('chien', 'chat')

```

R8. les couples (nom du client, nom de l'animal) avec un renommage pour les distinguer, des animaux adoptés en 2023,

Solution:

```

1 SELECT Animal.nom AS Nom_animal, Client.nom AS Nom_client
2 FROM Animal
3 JOIN Client
4 JOIN Adoption
5 ON animal_id=Animal.id AND client_id=Client.id
6 WHERE date_adoption >= '2023-01-01' AND date_adoption <= '2023-12-31'

```

R9. le nombre d'animaux,

Solution:

```

1 SELECT COUNT(id)
2 FROM Animal

```

R10. le nombre de mâles,

Solution:

```
1 SELECT COUNT(id)
2 FROM Animal
3 WHERE sexe='M'
```

R11. le nombre de chien et de chat,

Solution:

```
1 SELECT nom_courant ,COUNT(Animal.id)
2 FROM Animal
3 JOIN Espèce
4 ON espece_id=Espèce.id
5 WHERE nom_courant IN ('chien','chat')
6 GROUP BY nom_courant
```

R12. le total des prix en boutique (le prix étant donné par l'espèce de l'animal),

Solution:

```
1 SELECT SUM(prix)
2 FROM Espèce
3 JOIN Animal
4 ON Espèce.id=espece_id
5 WHERE disponible=1
```

R13. le nombre d'animaux par sexe,

Solution:

```
1 SELECT sexe ,COUNT(id)
2 FROM Animal
3 GROUP BY sexe
```

R14. le nombre d'animaux par espèce avec le nom de l'espèce,

Solution:

```
1 SELECT nom_courant ,COUNT(Animal.id)
2 FROM Espèce
3 JOIN Animal
4 ON Espèce.id=espece_id
5 GROUP BY nom_courant
```

R15. les parents d'un animal,

Solution:

```
1 SELECT A3.id, A3.nom AS animal , A1.nom AS père ,A2.nom AS mère
2 FROM Animal AS A1
3 JOIN Animal AS A2
```

```
4 JOIN Animal AS A3
5 ON A1.id=A3.pere_id AND A2.id=A3.mere_id
```

R16. le nombre de mâles et de femelles par espèce,

Solution:

```
1 SELECT nom_courant , sexe , COUNT(id)
2 FROM Espèce
3 JOIN Animal
4 ON Espèce.id=espece_id
5 WHERE sexe='M'
6 GROUP BY nom_courant
7 UNION
8 SELECT nom_courant , sexe , COUNT(id)
9 FROM Espèce
10 JOIN Animal
11 ON Espèce.id=espece_id
12 WHERE sexe='F'
13 GROUP BY nom_courant
```

R17. le nom et prénom des clients ayant fait au moins deux achats et leur nombre d'adoption,

Solution:

```
1 SELECT nom , prenom , COUNT(*)
2 FROM Client
3 JOIN Adoption
4 ON client_id=Client.id
5 GROUP BY client_id
6 HAVING COUNT(client_id)>=2
```

R18. le nombres d'animaux par espèce mais sans prendre en compte les chiens et les chats,

Solution:

```
1 SELECT nom_courant , COUNT(Animal.id)
2 FROM Animal
3 JOIN Espèce
4 ON espece_id=Espèce.id
5 WHERE nom_courant NOT IN ('chien', 'chat')
6 GROUP BY nom_courant
```

R19. le nom et prénom des clients ayant adopté au moins deux chats,

Solution:

```
1 SELECT Client.nom , prenom
2 FROM Animal
3 JOIN Espèce
4 JOIN Client
5 JOIN Adoption
```

```
6 ON espece_id=Espèce.id AND client_id=Client.id AND animal_id=Animal.id
7 WHERE nom_courant='chat'
8 GROUP BY client_id
9 HAVING COUNT(animal_id)>=2
```

R20. les espèces qui coûtent moins que le prix moyen toute espèce réunie.

Solution:

```
1 SELECT nom_courant
2 FROM Espèce
3 WHERE prix < (SELECT AVG(prix) FROM Espèce)
```


Exercice n°4 Un peu de musique

On étudie la base de données suivante, constituée de 2 tables :

<u>id</u>	nom	prenom	ddn	ddm	nationalité
1	Lizt	Franz			française
2	Mozart	Wolfgang Amadeus			autrichienne
3	van Beethoven	Ludwig	17791215	18670326	allemande
...
6	Vivaldi	Antonio			italienne
...

Table des compositeurs (Compositeurs)

<u>id</u>	titre	annee	periode	forme	duree	idcompo
1	La flûte enchantée	1791	classique	opéra	165	2
2	Neuvième Symphonie	1824	romantique	symphonie	70	3
3	Les quatre saisons	1725	baroque	concerto	40	6
...

Table des Œuvres (Oeuvres).

- **Compositeurs** (id : entier, nom : chaîne de caractères, prénom : chaîne de caractère, pays : chaîne de caractère, ddn : entier (date de naissance de la forme AAAAMMJJ, soit 20211117 pour le 17 novembre 2021), ddm : entier (date de décès de la même forme que la date de naissance))
- **Oeuvres** (id : entier, titre : chaîne de caractères, annee : entier (année de la composition), periode : chaîne de caractère (baroque, classique, romantique, ...), forme : chaîne de caractère (concerto, symphonie, opéra, ...), duree : flottant (durée en minutes), idcompo : entier (identifiant du compositeur))

R1. Est-ce que nom peut servir de clé pour la table Compositeurs ? Pourquoi ? Proposer une clé de cette table. Donner les clés primaires et les clés étrangères de chaque table.

Solution:

Écrire les requêtes en langage SQL qui renvoient :

R2. Les titres des œuvres composées durant la période classique.

Solution:

```
1 SELECT titre FROM Oeuvres
2 WHERE periode='classique'
```

R3. La liste des titres des œuvres qui ne sont ni un concerto ni une symphonie.

Solution:

```
1 SELECT titre FROM Oeuvres
2 WHERE forme<>concerto AND forme<>symphonie
```

OU

```
1 SELECT titre FROM Oeuvres
2 WHERE forme NOT IN ('concerto', 'symphonie')
```

R4. La liste des œuvres qui ont été composées après 1650.

Solution:

```
1 SELECT titre FROM Oeuvres
2 WHERE annee>1650
```

R5. La durée moyenne d'un concerto.

Solution:

```
1 SELECT AVG(duree) FROM Oeuvres
2 WHERE forme='concerto'
```

R6. La liste des œuvres composées par van Beethoven.

Solution:

```
1 SELECT titre
2 FROM Oeuvres
3 JOIN Compositeurs
4 ON Compositeurs.id=idcompo
5 WHERE nom='van Beethoven'
```

R7. Le compositeur ayant composé la Flûte enchantée.

Solution:

```
1 SELECT prenom,nom
2 FROM Compositeurs
3 JOIN Oeuvres
4 ON Compositeurs.id=idcompo
5 WHERE titre='La flûte enchantée'
```

R8. La liste des œuvres (titre de l'œuvre, nom du compositeur, année de composition) en les classant par ordre alphabétique des compositeurs puis par année.

Solution:

```
1 SELECT titre,nom,annee
2 FROM Compositeurs AS C
3 JOIN Oeuvres
4 ON C.id=idcompo
5 ORDER BY nom,annee
```

R9. Le nombre de compositeurs regroupés par pays.

Solution:

```
1 SELECT pays,COUNT(Composteurs.id)
2 FROM Compositeurs
3 JOIN Pays
4 ON idpays=Pays.id
5 GROUP BY pays
```

R10. L'œuvre la plus ancienne et son année de composition.

Solution: Pour avoir l'année de composition de la plus ancienne œuvre, c'est facile : `SELECT MIN(annee) FROM Oeuvres`. Attention, il n'est pas possible de récupérer le titre de la plus ancienne œuvre avec `SELECT titre, MIN(annee) FROM Oeuvres` : vous aurez bien l'année de l'œuvre la plus ancienne et le premier titre de la table `Oeuvres`, sans aucun lien a priori avec l'année.

Il est nécessaire d'utiliser une requête imbriquées : on commence par récupérer l'année de composition de la plus ancienne œuvre, puis de chercher les titres des œuvres dont leur année de composition est égale à cette année de composition de la plus ancienne œuvre.

```
1 SELECT titre,annee
2 FROM Oeuvres
3 WHERE annee
4     IN (SELECT MIN(annee) FROM Oeuvres)
5 # ou
6 SELECT titre,annee
7 FROM Oeuvres
8 ORDER BY annee
9 LIMIT 1
```

R11. La liste des compositeurs qui sont nés après Mozart.

Solution: On récupère la date de naissance de Mozart, puis on cherche les compositeurs dont la date de naissance est supérieure à celle-ci.

```
1 SELECT prenom,nom FROM Compositeurs
2 WHERE ddn >
3     (
4     SELECT ddn FROM Compositeurs
5     WHERE nom='Mozart'
6     )
```

R12. La liste des compositeurs ayant composé au moins une symphonie.

Solution:

```
1 SELECT DISTINCT prenom,nom
2 FROM Compositeurs
3 JOIN Oeuvres
4 ON Compositeurs.id=idcompo
5 WHERE forme='symphonie'
```

R13. La liste des compositeurs n'ayant jamais composé de symphonie.

Solution: Il faut commencer par sélectionner la liste des compositeurs (`idcompo`) ayant composé au moins une symphonie, puis on sélectionne la liste des compositeurs qui ne sont pas dans cette liste.

```
1 SELECT prenom,nom FROM Compositeurs
2 JOIN Oeuvres ON Compositeurs.id=idcompo
3 WHERE idcompo
4     NOT IN (
5     SELECT idcompo
6     FROM Oeuvres
7     WHERE forme='symphonie')
```

```
7 WHERE forme='symphonie'
8 )
```

R14. La liste des compositeurs n'ayant composé que des symphonies.

Solution: Il faut commencer par sélectionner la liste des compositeurs (idcompo) ayant composé au moins une œuvre qui n'est pas une symphonie, puis on sélectionne la liste des compositeurs qui ne sont pas dans cette liste. Ne seront pas sélectionnés les compositeurs ayant composé au moins une œuvre qui n'était pas une symphonie.

```
1 SELECT prenom,nom FROM Compositeurs
2 JOIN Oeuvres ON Compositeurs.id=idcompo
3 WHERE idcompo
4 NOT IN (
5 SELECT idcompo FROM Oeuvres
6 WHERE forme<>'symphonie'
7 )
```

R15. La liste du nombre d'œuvres composées par chaque compositeur classé par nombre d'œuvres composées.

Solution:

```
1 SELECT nom, prenom, COUNT(titre)
2 FROM Compositeurs
3 JOIN Oeuvres ON Compositeurs.id=idcompo
4 GROUP BY nom
5 ORDER BY COUNT(titre)
```

R16. La liste du nombre de symphonies composées par chaque compositeur, classé par nombre croissant de symphonies composées.

Solution:

```
1 SELECT nom, prenom, COUNT(titre)
2 FROM Compositeurs
3 JOIN Oeuvres ON Compositeurs.id=idcompo
4 WHERE forme='symphonie'
5 GROUP BY nom
6 ORDER BY COUNT(titre)
```

R17. La liste des compositeurs ayant composé au moins 9 symphonies.

Solution:

```
1 SELECT nom,prenom FROM Compositeurs
2 JOIN Oeuvres ON Compositeurs.id=idcompo
3 WHERE forme='symphonie'
4 GROUP BY idcompo
5 HAVING COUNT(titre)>=9
```

R18. La liste des couples d'œuvres composées la même année.

Solution: Il est nécessaire ici de faire une jointure sur la même table, avec comme condition de jointure, l'égalité de l'année de composition. Pour éviter d'avoir les couples constitués de deux fois la même œuvre, on ajoute comme contrainte : 01.titre<>02.titre.

```

1 SELECT 01.titre,02.titre,annee FROM Oeuvres AS 01, Oeuvres AS 02
2 WHERE 01.annee=02.annee
3 AND 01.titre<>02.titre
4
5 SELECT 01.titre,02.titre,annee
6 FROM Oeuvres AS 01 JOIN Oeuvres AS 02 ON 01.annee=02.annee
7 WHERE 01.titre<>02.titre

```

R19. Qui a composé la plus vieille œuvre ?

Solution: Même raisonnement que pour chercher l'œuvre la plus longue.

```

1 SELECT nom,prenom
2 FROM Compositeurs
3 JOIN Oeuvres ON Compositeurs.id=idcompo
4 ORDER BY annee
5 LIMIT 1
6
7 SELECT nom,prenom
8 FROM Compositeurs
9 JOIN Oeuvres ON Compositeurs.id=idcompo
10 WHERE annee
11     IN (SELECT MIN(annee) FROM Oeuvres)

```

R20. La plus vieille œuvre composée par Bach.

Solution: Idée : commencer par sélectionner l'année de la plus vieille œuvre de Bach (c'est-à-dire d'année de composition minimale), puis sélectionner le titre de l'œuvre de Bach dont l'année est celle-ci.

```

1 SELECT titre
2 FROM Oeuvres
3 JOIN Compositeurs as C ON C.id=idcompo
4 WHERE nom='Bach'
5 ORDER BY annee
6 LIMIT 1
7
8 SELECT titre
9 FROM Oeuvres
10 JOIN Compositeurs as C ON C.id=idcompo
11 WHERE nom='Bach'
12 AND annee=
13     (SELECT MIN(annee) FROM Oeuvres
14     JOIN Compositeurs AS C ON C.id=idcompo
15     WHERE nom='Bach')

```

R21. Donner le titre de la deuxième œuvre la plus longue.

Solution:

```
1 SELECT titre  
2 FROM Oeuvres  
3 ORDER BY duree  
4 OFFSET 1  
5 LIMIT 1
```