

Devoir n°1

2 heures

Problème : base de données

Certains matériaux particuliers peuvent acquérir des états magnétiques qualifiés de paramagnétique et ferromagnétique. Le matériau est dit paramagnétique lorsqu'il ne possède pas d'aimantation spontanée, mais acquiert une aimantation sous l'effet d'un champ magnétique extérieur. Il est dit ferromagnétique lorsqu'il possède une aimantation même en l'absence de champ magnétique extérieur. Dans ces matériaux, la température T joue un rôle crucial : si T est supérieure à une température particulière T_C , nommée température de Curie, le matériau adopte un état paramagnétique. Dans le cas contraire ($T < T_C$), il adopte un état ferromagnétique. C'est par exemple le cas du fer, pour lequel la transition entre les deux états se produit à $T_C = 1043$ kelvin.

Il existe des bases de données contenant les propriétés de nombreux matériaux, dont des propriétés magnétiques. Dans cette partie, on donne un modèle simplifié d'une telle base, et on souhaite effectuer quelques requêtes sur celle-ci.

La base de données possède la structure suivante :

– La table `matériaux` contient un champ `id_materiau`, clé primaire de la table de valeur entière, un champ `nom` de type chaîne de caractères pour le nom du matériau et un champ `t_curie` de valeur entière pour la température de Curie du matériau en kelvin.

id_materiau	nom	t_curie
4534	cobalt	1 388
1254	dioxyde de chrome	386
8713	nickel	627
8284	YIG	560
...

– La table `fournisseurs`, contenant un champ `id_fournisseur`, clé primaire de type entier qui précise le code de chaque fournisseur, et un champ `nom_fournisseur` de type chaîne de caractères pour le nom du fournisseur.

id_fournisseur	nom_fournisseur
145	Worldwide Materials
13	Materials Compagny
...	...

– La table `prix` qui contient un champ `id_prix`, clé primaire de type entier, un champ `id_mat` dont les valeurs sont incluses dans l'ensemble des valeurs de la clé `id_materiau` de la table `matériaux`, un champ

id_four dont les valeurs sont incluses dans l'ensemble des valeurs de la clé id_fournisseur de la table fournisseurs, et un champ prix_kg de type flottant qui précise le prix au kg que ce fournisseur propose pour ce matériau, en euros. Un fournisseur qui ne propose pas un matériau donné n'a pas d'entrée correspondante dans cette table.

id_prix	id_mat	id_four	prix_kg
1	4567	145	50.40
2	8671	13	1357.30
3	1763	145	52.75
...

Les requêtes demandées dans cette partie sont à écrire en langage SQL.

1. Combien d'attributs possède la table fournisseurs ?
2. Expliquer le type de l'attribut t_curie.
3. L'attribut t_curie est-il une clé primaire et pourquoi ?
4. L'attribut id_four est-il une clé primaire et pourquoi ?
5. Comment désigne t-on l'attribut id_mat et pourquoi ?
6. Dans cette base de données, combien y-a-t-il d'entités et d'associations ?
7. De quel type est l'association entre les tables prix et fournisseurs ? Détaillez l'association et ses deux couples de cardinalité.
8. De quel type est l'association indirecte entre les tables matériaux et fournisseurs ? Quel est le rôle de la table prix ?
9. Écrire une requête qui détermine le nombre de matériaux de la base de données.
10. (Mines) Écrire une requête permettant d'obtenir le nom de tous les matériaux qui ont une température de Curie strictement inférieure à 500 kelvins.
11. Écrire une requête qui détermine la moyenne des prix au kg du fer de tous les fournisseurs.
12. Écrire une requête SQL qui le nombre de matériaux disponibles pour chaque fournisseur.
13. Écrire une requête SQL qui détermine la liste des 10 premiers fournisseurs par ordre alphabétique inverse.
14. Écrire une requête SQL qui détermine tous les couples fournisseurs, matériaux.
15. Un client potentiel souhaite acheter 4,5 kilogrammes de nickel et sélectionner le fournisseur le moins cher.
 - (a) Quelle est la requête permettant de déterminer l'identifiant du nickel ? On suppose bien sûr que l'on n'a pas accès facilement visuellement à la ligne correspondante.
 - (b) (Mines) On suppose que l'on connaît l'identifiant 8713 du nickel, et on pourra donc l'utiliser directement dans les requêtes.

Écrire une requête permettant d'obtenir les noms de tous les fournisseurs proposant du nickel et le prix proposé par chacun pour 4,5 kilogrammes de nickel.

- (c) (Mines) Modifier ou compléter la requête précédente afin d'obtenir le nom du fournisseur de nickel le moins cher ainsi que le prix à payer chez ce fournisseur pour ces 4,5 kilogrammes de nickel. En cas d'égalité du prix optimal entre plusieurs fournisseurs, on obtiendra les noms de tous les fournisseurs possibles.
- (d) Donner une autre solution plus simple mais ne renvoyant qu'un seul fournisseur.
16. (Mines) Écrire une requête permettant d'obtenir le nom de tous les matériaux et le prix moyen pour un kilogramme de chacun de ces matériaux (la moyenne étant calculée pour tous les fournisseurs proposant ce matériau), en se limitant aux prix moyens strictement inférieurs à 50 euros par kilogramme.



1. La table fournisseurs possède 2 attributs, qui sont id_fournisseur clé primaire, et nom_fournisseur.
2. L'attribut t_curie de la table materiaux est de type entier, entier naturel non nul d'après le choix de l'énoncé (pas de zéro absolu). On ne rentrera pas ici dans la discussion de la précision de la température de Curie.
3. L'attribut t_curie de la table materiaux n'est pas une clé primaire, car il est peut être possible que deux matériaux (voire des matériaux composites, ce qui permet d'en considérer beaucoup) aient la même température de Curie.
4. L'attribut id_four de la table prix n'est pas une clé primaire, c'est une clé étrangère, reliant la table prix à la table fournisseurs par sa clé primaire id_fournisseur.
5. L'attribut id_mat de la table prix est une clé étrangère, reliant la table prix à la table materiaux par sa clé primaire id_materiau.
6. Dans cette base de données, nous avons 3 entités (les 3 tables materiaux, fournisseurs et prix), et deux associations (qui permettent de modéliser une association non réalisée entre materiaux et fournisseurs) entre materiaux et prix ainsi que entre fournisseur et prix par le biais des clé secondaires id_four et id_mat respectivement.
7. Un fournisseur va proposer de 1(0) à plusieurs prix, soit un type 1 – * et un prix va concerner un et un seul fournisseur, de type 1-1.
8. Un fournisseur va proposer de 1(0) à plusieurs matériaux, et un matériau peut-être proposé par 0 à plusieurs fournisseurs. Nous avons une cardinalité de type * – *, qui ne peut pas se réaliser à l'aide de clés primaire-étrangère sans utiliser de table annexe.
La table prix permet de réaliser le lien entre les 2 tables materiaux et fournisseur.
9. Écrire une requête qui détermine le nombre de matériaux de la base de données.

```
SELECT COUNT(*) FROM materiaux
```

10. (Mines) Écrire une requête permettant d'obtenir le nom de tous les matériaux qui ont une température de Curie strictement inférieure à 500 kelvins.

```
SELECT nom FROM materiaux WHERE t_curie > 500
```

11. Écrire une requête qui détermine la moyenne des prix au kg du fer de tous les fournisseurs.

```
SELECT AVG(prix_kg) FROM prix  
JOIN materiaux ON id_mat = id_materiau  
WHERE nom = 'fer'
```

12. Écrire une requête SQL qui détermine le nombre de matériaux disponibles pour chaque fournisseur.

```
SELECT nom_fournisseur, count(id_mat) FROM fournisseurs
JOIN prix ON id_fournisseur = id_four
GROUP BY id_four
```

13. Écrire une requête SQL qui détermine la liste des dix premiers fournisseurs par ordre alphabétique inverse.

```
SELECT nom_fournisseur FROM fournisseurs
ORDER BY nom_fournisseur DESC
LIMIT 10
```

14. Écrire une requête SQL qui détermine tous les couples compatibles fournisseurs, matériaux.

```
SELECT nom_fournisseur, nom
FROM matériaux, fournisseurs, prix
WHERE id_mat = id_materiau AND id_four = id_fournisseur
```

ou

```
SELECT nom_fournisseur, nom FROM prix
JOIN matériaux ON id_materiau = id_mat
JOIN fournisseurs ON id_fournisseur = id_four
```

```
SELECT nom_fournisseur, nom FROM fournisseurs
JOIN matériaux
JOIN prix
ON id_mat = id_materiau AND id_fournisseur = id_four
```

La suivante ne fonctionne pas (id_fournisseur est alors inconnu)

```
SELECT nom_fournisseur, nom FROM fournisseurs, matériaux
JOIN prix ON id_mat = id_materiau AND id_four = id_fournisseur
```

mais celle-ci, avec les parenthèses autour du produit cartésien, oui :

```
SELECT nom_fournisseur, nom FROM (fournisseurs, matériaux)
JOIN prix ON id_mat = id_materiau AND id_four = id_fournisseur
```

15. (a) Quelle est la requête permettant de déterminer l'identifiant du nickel ? On suppose bien sûr que l'on n'a pas accès facilement visuellement à la ligne correspondante.

```
SELECT id_materiau FROM matériaux
WHERE nom = 'nickel'
```

- (b) (Mines) On suppose que l'on connaît l'identifiant 8713 du nickel, et on pourra donc l'utiliser directement dans les requêtes.

Écrire une requête permettant d'obtenir les noms de tous les fournisseurs proposant du nickel et le prix proposé par chacun pour 4,5 kilogrammes de nickel.

```
SELECT nom_fournisseur, prix_kg * 4.5 FROM fournisseurs
JOIN prix ON id_four = id_fournisseur
WHERE id_mat = 8713
```

- (c) (Mines) Modifier ou compléter la requête précédente afin d'obtenir le nom du fournisseur de nickel le moins cher ainsi que le prix à payer chez ce fournisseur pour ces 4,5 kilogrammes de nickel. En cas d'égalité du prix optimal entre plusieurs fournisseurs, on obtiendra les noms de tous les fournisseurs possibles.

```
SELECT nom_fournisseur, prix_kg * 4.5 FROM fournisseurs
JOIN prix ON id_four = id_fournisseur
WHERE id_mat = 8713
AND prix_kg =
(SELECT MIN(prix_kg) FROM prix WHERE id_mat = 8713)
```

- (d) Donner une autre solution plus simple mais ne renvoyant qu'un seul fournisseur.

```
SELECT nom_fournisseur, prix_kg * 4.5 FROM fournisseurs
JOIN prix ON id_four = id_fournisseur
WHERE id_mat = 8713
ORDER BY prix_kg ASC
LIMIT 1
```

16. (Mines) Écrire une requête permettant d'obtenir le nom de tous les matériaux et le prix moyen pour un kilogramme de chacun de ces matériaux (la moyenne étant calculée pour tous les fournisseurs proposant ce matériau), en se limitant aux prix moyens strictement inférieurs à 50 euros par kilogramme.

```
SELECT nom, AVG(prix_kg) AS moyenne FROM materiaux
JOIN prix ON id_mat = id_materiau
GROUP BY id_mat
HAVING moyenne < 50
```