Un modèle relationnel, des requêtes SQL

INFORMATIQUE COMMUNE - TP n° 3.4 - Olivier Reynet

À la fin de ce chapitre, je sais :

- ☞ interpréter et utiliser un modèle relationnel de base de données
- ☞ utiliser les opérateurs de projection et de sélection sur un modèle simple (select from, where)
- ☞ utiliser les clefs primaires et étrangères dans une requête simple
- ☞ opérer une jointure interne entre plusieurs tables (join on)
- ☞ utiliser les fonctions d'agrégation pour un calcul simple (min, max, sum, avg, count)
- ☞ filtrer des agrégations d'après un critère (having)
- ☞ utiliser des opérateurs ensemblistes (intersect, union, except)

Ce TP s'inspire de l'épreuve d'informatique commune du concours CCMP 2022.

A Présentation de la base de données

On utilise une base de données contenant les propriétés de quelques matériaux (materials) dotés de propriétés magnétiques. Des fournisseurs (suppliers) proposent à la vente des matériaux au kg. On recense les offres de prix (price) de ces fournisseurs pour chaque matériau. On suppose qu'un fournisseur ne propose qu'une seule offre de prix par matériau.

Cette base de données possède le modèle relationnel décrit sur la figure 1. Son modèle physique est décrit sur la figure 2. Sur la figure 3, les tables (entêtes et enregistrements) sont décrites dans le détail.

B Sur le modèle

B1. Donner le nom de toutes les clefs primaires du modèle relationnel de la base de données.

> **Solution :** id_material dans la table material, id_supplier dans la table supplier, id_price dans la table price. Ces clefs sont généralement soulignées dans la représentation du modèle relationnel.

B2. Pourquoi n'a-t-on pas choisi le nom du fournisseur comme identifiant (clef primaire) ?

> **Solution :** Parce qu'on peut avoir des fournisseurs dont le nom est le même mais qui représentent des entreprises différentes. Par exemple, on peut imaginer une Materials Compagnie au Royaume-Uni, aux USA ou en Australie.

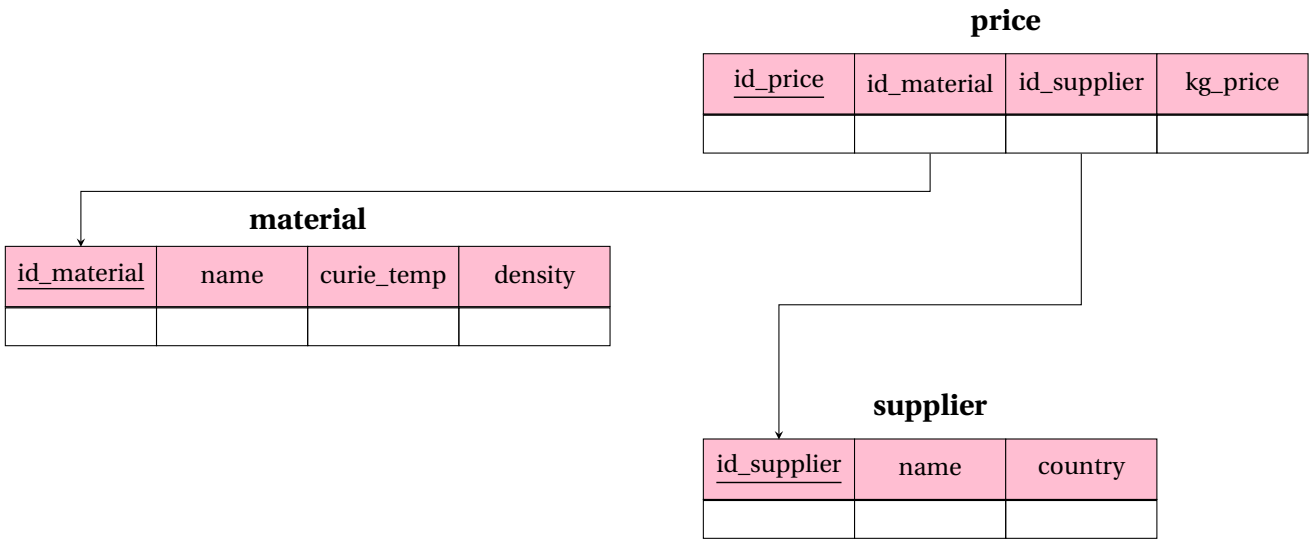


FIGURE 1 – Modèle relationnel de la base de données des matériaux magnétiques.

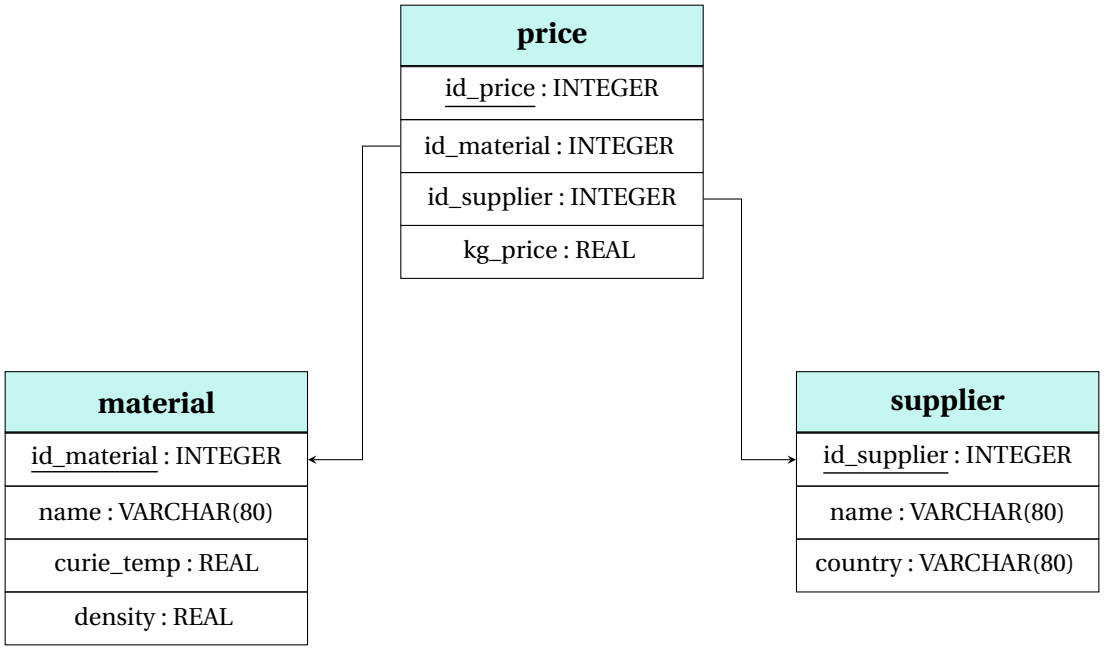
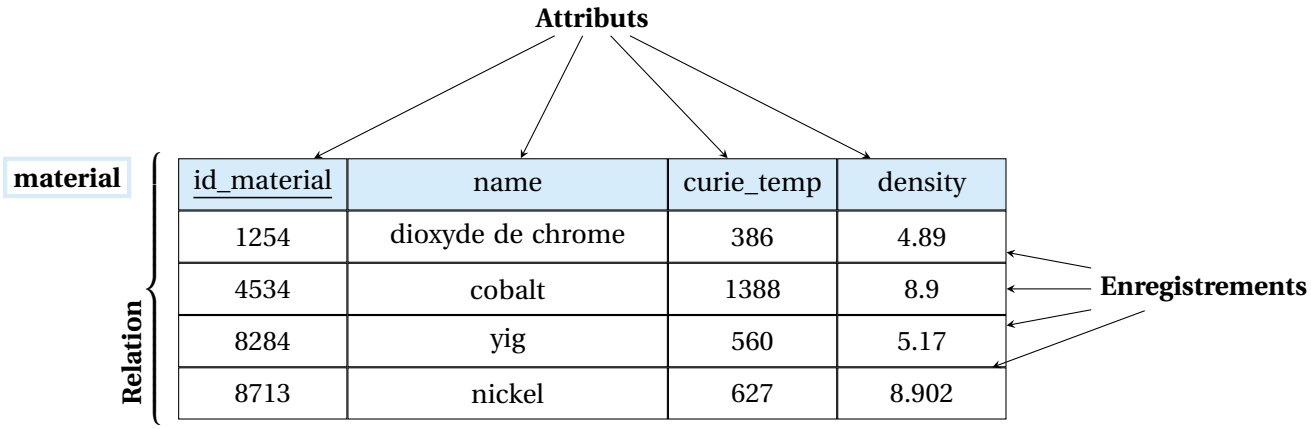


FIGURE 2 – Modèle physique associé au modèle relationnel de la base de données des matériaux magnétiques.



id_material	name	curie_temp	density
1254	dioxyde de chrome	386	4.89
4534	cobalt	1388	8.9
8284	yig	560	5.17
8713	nickel	627	8.902

supplier

id_supplier	name	country
15	Minerais de Paris	France
42	Minereizh	France
66	Mineral Gesellshaft	Allemagne
145	World Wide Materials	USA

price

id_price	id_material	id_supplier	kg_price
1	4534	145	50.4
5	1254	15	100.4
13	4534	66	49.97
22	4534	15	52.34
35	8713	15	25.4
85	8713	42	31.62
97	8284	145	1247.57
99	1254	66	86.45
111	4534	42	49.4
485	8284	15	1025.4
501	1254	145	83.97
685	8284	42	1031.62
751	8713	145	27.57
983	8713	66	25.4

FIGURE 3 – Description détaillée de la base de données relationnelle des matériaux magnétiques.

B3. Donner le nom des clefs étrangères du modèle.

Solution : id_material, id_supplier dans la table price.

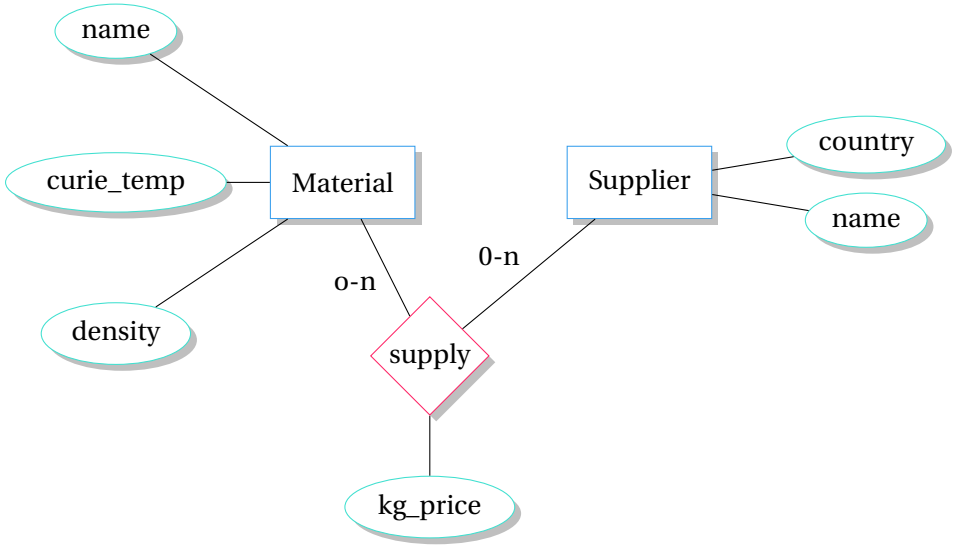
B4. La relation prix représente une association du modèle conceptuel. Comment pourrait-on la nommer ? Identifier les cardinalités de cette association. De quel type cette association est-elle ?

Solution : On pourrait la nommer supply (fournir) pour signifier *fournir un matériau à un prix donné*. Un même matériau peut-être fourni par plusieurs fournisseur. Un fournisseur propose plusieurs matériaux différents. On a donc les cardinalités suivantes : supplier $0-n$ supply $0-n$ material. C'est une association de plusieurs à plusieurs.

B5. Représenter le modèle conceptuel associé à ce modèle relationnel.

Solution : Un modèle entité-relation associé est proposé ci-dessous. On notera dans modèle proposé que la relation binaire supply possède un attribut kg_price. Ceci est cohérent avec les données présentées dans l'énoncé dans le sens où une même entreprise ne propose qu'un seul prix pour un matériau donné.

--→ HORS PROGRAMME Mais on pourrait imaginer que ce n'est pas le cas et qu'un entreprise pourrait proposer un même matériau à plusieurs prix différents selon la qualité du matériau. Dans ce cas, il faudrait imaginer une association supply ternaire avec une entité nouvelle qui spécifierait le prix ainsi que la qualité du matériau.



C Échauffement

Écrire une requête SQL qui permet d'obtenir :

C1. le nom de tous les matériaux.

C2. le nom de tous les matériaux dont la température de Curie est strictement inférieure à 500 kelvins.

C3. tous les attributs des matériaux dont la température de Curie est strictement inférieure à 600 kelvins ou la densité strictement inférieure à 8.

C4. le prix au kg minimum, moyen et maximum calculés sur la totalité des matériaux disponibles et chez tous les fournisseurs.

Solution :

```sql
SELECT *
FROM material;

SELECT name
FROM material
WHERE curie_temp < 500;

SELECT *
FROM material
WHERE curie_temp < 600
or density < 8;

SELECT MIN(kg_price), AVG(kg_price), MAX(kg_price)
FROM price;
```

D Jointons!

Écrire une requête SQL qui permet d'obtenir :

D1. le nom de tous les matériaux disponibles à la vente. On n'affichera une liste sans doublons dans l'ordre alphabétique inverse des noms de matériaux.

D2. le nom des fournisseurs de nickel ainsi que le prix auquel ils fournissent une tonne de nickel. On utilisera l'identifiant du nickel.

D3. le nom du fournisseur, l'**identifiant** du matériau qu'il fournit et le prix au kg de ce matériau. On affichera les résultats ordonnés d'après le nom du fournisseur et le prix au kg.

D4. le nom du fournisseur, le **nom** du matériau qu'il fournit et le prix au kg de ce matériau. On affichera les résultats ordonnés d'après le nom du fournisseur et le prix au kg. (double jointure)

D5. le nom de tous les matériaux dont le fournisseur est localisé en France. (double jointure)

Solution :

```sql
SELECT DISTINCT material.name
FROM material
JOIN price ON price.id_material = material.id_material
ORDER BY material.name DESC;

SELECT name, 1000 * kg_price
FROM supplier
JOIN price ON supplier.id_supplier = price.id_supplier
WHERE price.id_material = 8713;
```

```sql
SELECT name, id_material, kg_price
FROM supplier
JOIN price ON supplier.id_supplier = price.id_supplier
ORDER BY name, kg_price;

SELECT supplier.name, material.name, kg_price
FROM supplier
JOIN price ON supplier.id_supplier = price.id_supplier
JOIN material on price.id_material = material.id_material
ORDER BY supplier.name, kg_price DESC;

SELECT material.name
FROM material
JOIN price on price.id_material = material.id_material
JOIN supplier on price.id_supplier = supplier.id_supplier
WHERE supplier.country = "France";
```

E Sous-requête --→ HORS PROGRAMME mais dans de nombreuses épreuves...

Écrire une requête SQL qui permet d'obtenir :

E1. les prix au kg du nickel.

E2. les noms des fournisseurs de nickel.

E3. le nom des fournisseurs de nickel ainsi que le prix d'une tonne de nickel associé à chaque fournisseur.

E4. le nom des fournisseurs de nickel ainsi que le prix minimum d'une tonne de nickel associé à chaque fournisseur. Si le prix minimum est pratiqué par un seul fournisseur, la requête n'affiche qu'un résultat. Par contre, si le prix minimum est pratiqué par plusieurs fournisseurs, la requête les affiche tous.

Solution :

```sql
SELECT kg_price
FROM price
WHERE id_material = (SELECT id_material
FROM material
WHERE name = "nickel");

SELECT name
FROM supplier
WHERE id_supplier IN (SELECT id_supplier
                      FROM price
                      WHERE id_material = (SELECT id_material
                                           FROM material
                                           WHERE name = "nickel"));

SELECT name, 1000 * kg_price
FROM supplier
```

```sql
JOIN price ON supplier.id_supplier = price.id_supplier
WHERE price.id_material = (SELECT id_material
                           FROM material
                           WHERE name = "nickel");

SELECT name, 1000 * kg_price
FROM supplier
JOIN price ON supplier.id_supplier = price.id_supplier
WHERE price.id_material = 8713
and price.kg_price = (SELECT MIN(kg_price)
                      FROM price
                      WHERE id_material = 8713);

SELECT name, 1000 * kg_price
FROM supplier
JOIN price ON supplier.id_supplier = price.id_supplier
WHERE price.id_material = (SELECT id_material
                           FROM material
                           WHERE name = "nickel")
and price.kg_price = (SELECT MIN(kg_price)
                      FROM price
                      WHERE id_material = (SELECT id_material
                                           FROM material
                                           WHERE name = "nickel"));
```

F Regroupons, agrégeons et filtrons !

Écrire une requête SQL qui permet d'obtenir :

F1. le nom du matériau et son prix moyen, du plus cher au moins cher.

F2. le nom du matériau et son prix moyen, du moins cher au plus cher, si le prix moyen est inférieur à 100.

F3. le nom du matériau, le nombre de fournisseurs et le prix moyen, s'il y a plus de trois fournisseurs de ce matériau et si le prix moyen est inférieur à 100, du plus cher au moins cher en moyenne.

Solution :

```sql
SELECT name, AVG(kg_price)
FROM material
JOIN price ON material.id_material = price.id_material
GROUP BY name
ORDER BY AVG(kg_price) DESC;

SELECT name, AVG(kg_price)
FROM material
JOIN price ON material.id_material = price.id_material
GROUP BY name
HAVING AVG(kg_price) < 100
ORDER BY kg_price ASC;
```

```sql
SELECT name, COUNT(id_supplier), AVG(kg_price)
FROM material
JOIN price ON price.id_material = material.id_material
GROUP BY name
HAVING COUNT(id_supplier) > 3
and AVG(kg_price < 100)
ORDER BY AVG(kg_price) DESC;
```