
Devoir maison n°3bis - À rendre le 8 mars 2026

Du SQL pour obtenir une instance de KNAPSACK

Exercice 1 : Un peu de SQL pour commencer

Dans cet exercice, on s'intéresse à un problème éditorial concret♣. On souhaite éditer un recueil des articles d'un blog les plus appréciés par la tranche d'âge 18-25, avec la contrainte que ce recueil doit tenir en 32 pages.

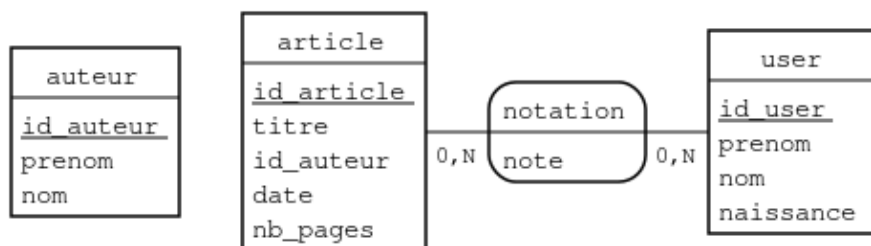
On dispose pour cela d'une base de données, nommée recueil et disponible sur cahier de prépa, contenant les tables suivantes.

- La table article liste les articles du blog en donnant pour chacun son identifiant, son titre, l'identifiant de son auteur, sa date de publication ainsi que le nombre de pages de l'article (dans sa version imprimée).
- La table auteur liste les auteurs des articles du blog en donnant pour chacun son identifiant, son prénom et son nom.
- La table user liste les utilisateurs du blog en donnant pour chacun son identifiant, son prénom, son nom et son année de naissance.
- La table notation liste les notes attribuées aux articles du blog en donnant pour chacune l'identifiant de l'utilisateur qui a donné la note, l'identifiant de l'article noté et enfin la note donnée.

Q. 1 Donner le schéma entité-association de la base de données recueil.

Solution

Voici le schéma entité-association de cette base de données générée avec le logiciel libre Mocodo <https://www.mocodo.net/>. À la main on peut se contenter du nom des tables et ne pas préciser les attributs. De plus les associations (comme ici notation) peuvent être représentées dans un losange.



Q. 2 Rappeler les définitions de clé primaire et de clé étrangère.

♣. Ce problème est fictif. Si le blog Image des mathématiques, les articles et les auteurs cités existent bel et bien, les utilisateurs et les notations ont été inventés, au hasard, pour les besoins de l'exercice.

Solution

On dit qu'une table admet pour clé primaire le n -uplet d'attributs $(c_1, c_2, \dots, c_n)$ lorsque deux enregistrements distincts de cette table ne peuvent coïncider sur ces n attributs. Très souvent $n = 1$ et l'attribut qui sert de clé primaire est un identifiant, mais cela peut aussi être un triplet comme $(\text{nom}, \text{prenom}, \text{date_de_naissance})$ par exemple.

On dit qu'une table t_1 admet l'attribut c comme clé étrangère référençant l'attribut c' d'une table $t_2 \neq t_1$ dès lors que, pour chaque enregistrement de t_1 , la valeur de l'attribut c est la valeur de l'attribut c' d'un enregistrement de la table t_2 .

Q. 3 Quelles sont les clés étrangères de la table notation et quelles données référencent-elles ?

Solution

Le champ `id_article` de la table notation est une clé étrangère qui référence le champ `id_article` de la table article. Le champ `id_user` de la table notation est une clé étrangère qui référence le champ `id_user` de la table user.

Q. 4 Donner une requête SQL qui permet d'obtenir la liste des titres des articles triés par dates croissantes.

Solution

```
SELECT titre FROM article
ORDER BY date;
```

Q. 5 Donner une requête SQL qui permet d'obtenir la liste des titres et auteurs de chaque article.

Solution

```
SELECT art.titre, aut.nom, aut.prenom
FROM article as art
JOIN auteur as aut ON art.id_auteur = aut.id_auteur
```

Q. 6 Donner une requête SQL qui permet d'obtenir le nombre d'articles écrits par chaque auteur.

Solution

```
SELECT aut.nom, aut.prenom, count(art.id_article) as nb_articles
FROM article as art
JOIN auteur as aut ON art.id_auteur = aut.id_auteur
GROUP BY aut.id_auteur
```

Q. 7 Donner une requête SQL qui permet d'obtenir la liste des auteurs ayant rédigé deux articles ou plus.

Solution

```
SELECT aut.nom, aut.prenom, count(art.id_article) as nb_articles
FROM article as art
JOIN auteur as aut ON art.id_auteur = aut.id_auteur
GROUP BY aut.id_auteur
HAVING nb_articles >= 2
```

Q. 8 Donner une requête SQL qui permet d'obtenir, pour chaque article, la moyenne des notes données par des utilisateurs de la tranche d'âge 18-25 ainsi que son nombre de pages.

Solution

```
SELECT art.id_article, avg(nota.note), art.nb_pages FROM notation as nota
JOIN article as art ON art.id_article = nota.id_article
JOIN user ON user.id_user = nota.id_user
WHERE 2026 - 25 <= user.naissance and user.naissance <= 2026-18
GROUP BY art.id_article
```

Q. 9 En déduire une instance de KNAPSACK qui modélise le problème de la constitution du recueil.

Solution

Le but est de choisir un ensemble d'articles dont la somme des utilités est maximale parmi les ensembles d'articles que l'on peut réunir en un recueil de 32 pages. Ainsi les objets de cette instance sont les articles, leur poids est leur nombre de pages, leur valeur leur note moyenne et la capacité du sac est 32 (le nombre de page du recueil).

Afin de retrouver les notations du DM, on renumérote les $n = 12$ articles de l'instance donnée par la requête précédente entre 1 et 12.

i	1	2	3	4	5	6	7	8	9	10	11	12
v_i	3.5	5	4	4	1	3.5	3.5	3	4	3	3.5	3.6
w_i	4	2	8	7	2	3	4	4	5	2	2	17

Q. 10 Question bonus pour faire le lien avec le reste du DM3

Exporter le résultat de 8 au format csv, puis écrire un programme OCAML permettant d'importer ces données sous la forme d'un objet de type sad qui modélise l'instance de KNAPSACK correspondante.