

# TP17: SQL

## MP2I Lycée Pierre de Fermat

On utilise la base de données construite à la fin du cours précédent, qui modélise le catalogue d'un magasin ainsi que sa liste de clients et les achats que ceux-ci ont effectué. Le fichier correspondant se trouve sur Cahier de Prépa dans l'archive du TP. Les données sont adaptées de données du jeu Animal Crossing: New Horizons qui ont été recensées sur Kaggle.<sup>1</sup> Voici les différentes tables (on rappelle que les attributs soulignés font partie de la clé primaire):

- `villagers(id, name, species, birthday, style, color, wallpaper)` stocke les informations des villageois qui ont ouvert un compte chez Boutique Nook.
- `articles(id, variant_id, name, variation_name, price, color, type)` stocke les informations des différents articles du catalogue. Certains articles peuvent avoir plusieurs variations, typiquement au niveau des couleurs.
- `wallpapers(name, price, color, window_type)` stocke les informations des papiers peints proposés par la boutique.
- `referral(id_referee, id_referrer)` stocke les informations du programme de fidélité, autrement dit pour chaque client, qui l'a parrainé. Certains clients n'ont pas de parrain. Le "referee" est le parrainé, et le "referrer" est le parrain.
- `purchases(villager_id, article_id, variant_id, date, amount)` stocke les différents achats effectués par chaque client chaque jour.
- `promotions(id, article_id, month, reduction)` stocke les informations des différentes soldes réalisées par le magasin. La réduction est donnée en pourcentage. Ainsi, si un enregistrement a comme valeur 70 pour la réduction, cela signifie que l'article concerné coûte seulement 30% de son prix ce mois-ci.

**Question 1.** Explorez les données de la base pour vous familiariser avec sa structure et avec les nombreux attributs des différentes tables.

**JUSQU'À LA QUESTION 21 INCLUSE** on ne considère pas la table `promotions`.

### Requêtes basiques

**Question 2.** Déterminer le nom des villageois rouges.

**Question 3.** Déterminer la liste des couleurs possibles pour les villageois.

**Question 4.** Déterminer s'il existe un villageois qui a votre date d'anniversaire et votre couleur préférée. Attention au format des dates !

---

<sup>1</sup>[kaggle.com/datasets/jessicali9530/](https://kaggle.com/datasets/jessicali9530/)

**Question 5.** Donner les 10 articles les plus chers, sans prendre en compte les variantes (deux articles identiques mais de couleurs différentes seront comptés comme un seul article).

**Question 6.** En notant  $p_1, \dots, p_n$  les papiers peints triés par prix croissant, donner  $p_4, \dots, p_{10}$ .

**Question 7.** Donner la liste des dates auxquelles un article a été acheté en exactement 7 exemplaires d'un coup.

**Question 8.** Donner les dates auxquelles c'est l'anniversaire d'un villageois cool mais aucun article n'a été acheté en exactement 7 exemplaires d'un coup.

## Jointures

**Question 9.** Déterminer la liste des villageois dont le papier peint n'est pas de leur couleur.

**Question 10.** Déterminer le nom du référent de Charlise

**Question 11.** Déterminer les villageois qui se sont auto-déclarés comme référents.

**Question 12.** Compter le nombre d'achats réalisés par des villageois le jour de leur anniversaire.

**Question 13.** Déterminer la liste des achats réalisés par des villageois le jour de leur anniversaire. On renverra le nom du villageois, le nom de l'article en question, son prix, et la date d'achat (c'est à dire la date de l'anniversaire du villageois). On n'affichera que les 10 achats les plus chers.

**Question 14.** Déterminer la liste des villageois qui ont acheté un article qui n'est pas de leur couleur.

**Question 15.** Déterminer la liste des villageois qui n'ont **jamais** acheté un article de leur couleur.

## Agrégats

**Question 16.** Déterminer le nombre de villageois de chaque couleur.

**Question 17.** Déterminer, pour chaque villageois, le nombre d'articles qu'il a acheté, l'article le plus cher et l'article le moins qu'il a acheté.

**Question 18.** Déterminer pour chaque combinaison espèce-couleur-style, le nombre de villageois de cette combinaison. Afficher seulement les 10 combinaisons les plus probables.

**Question 19.** Déterminer le prix moyen des articles achetés par chaque villageois, et afficher les 10 villageois achant les articles les moins chers en moyenne.

**Question 20.** Déterminer la somme totale dépensée par chaque villageois pendant le premier trimestre de l'année (entre le 01-01 et le 03-31). Afficher les 5 villageois les plus dépensiers durant cette période.

**Question 21.** Reprendre la requête précédente, et n'afficher que les villageois ayant dépensé entre 40000 et 50000 clochettes sur cette période.

## Requêtes avancées

On considère à partir de maintenant la table des promotions. On veut déterminer combien d'argent chaque villageois a dépensé lors de l'année, promotions incluses.

Avec SQLite, on peut comparer des chaînes de caractères en utilisant l'opérateur LIKE. Il va de pair avec le symbole '%', que l'on appelle **joker**, et qui signifie "n'importe quoi". Par exemple, pour avoir la liste des villageois qui sont nés le 1er du mois, on peut écrire:

```
1 SELECT * FROM villagers v
2 WHERE v.birthday LIKE "%-01";
```

En SQL, la concaténation de chaînes de caractères se note "||". On aurait donc tout aussi bien pu écrire:

```
1 SELECT * FROM villagers v
2 WHERE v.birthday LIKE "%-" || "-" || "01";
```

**Question 22.** Déterminer les achats réalisés le jour de l'anniversaire de l'acheteur alors qu'il y avait une promotion sur l'article acheté. On affichera le nom et l'anniversaire du villageois, le nom et le prix de l'article acheté, le montant acheté ainsi que le pourcentage de réduction appliqué.

On introduit une syntaxe de SQL permettant de faire des "if-then-else" sur les attributs d'une requête. Regardons la requête suivante:

```
1 SELECT DISTINCT
2 a.name,
3 CASE WHEN a.price < 5000 THEN
4     "faible"
5 WHEN 5000 <= a.price AND a.price < 10000 THEN
6     "moyen"
7 ELSE
8     "grand"
9 END
10 FROM articles a;
```

Tout le bloc entre CASE et END définit un unique attribut. On peut ainsi définir un attribut dont la valeur varie selon certaines conditions, on peut ensuite le renommer et l'utiliser comme n'importe quel attribut classique. Par exemple, pour compter le nombre d'articles dans chaque groupe de prix:

```
1 SELECT
2 CASE WHEN a.price < 5000 THEN
3     "faible"
4 WHEN 5000 <= a.price AND a.price < 10000 THEN
5     "moyen"
6 ELSE
7     "grand"
8 END AS groupe_prix,
9 count(DISTINCT a.name) as nb_articles
10 FROM articles a
11 GROUP BY groupe_prix;
```

Cette requête va afficher, pour chaque article, soit "faible" soit "moyen" soit "grand", selon le prix de l'article.

D'une manière similaire, on peut définir un nouvel attribut comme une combinaison d'autres attributs, en utilisant des opérations classiques (addition, concaténation, etc...)

Par exemple, la requête suivante affiche la liste des articles en préfixant le nom de l'article par le nom de la variation de l'article:

```
1 SELECT name, variation_name || ' ' || name as detail FROM articles;
```

**Question 23.** Exécuter la requête précédente, identifier le problème qui survient pour certains articles, et le régler en utilisant la syntaxe CASE WHEN ... END

**Question 24.** Déterminer la liste de tous les achats réalisés, en ajoutant une colonne additionnelle donnant la réduction appliquée pour cet achat (0 si aucune promotion n'est en cours sur l'article acheté). *Indication: une jointure externe gauche sur les promotions peut vous permettre via les valeurs NULL de déterminer si un achat a été fait lors d'une promotion.*

**Question 25.** Déterminer le villageois ayant dépensé le moins d'argent. On devra prendre en compte les promotions, mais aussi le fait que tout achat qu'un villageois réalise le jour de son anniversaire est gratuit.

**Question 26.** Déterminer le villageois le moins rentable pour le magasin Nook, c'est à dire celui pour qui le ratio  $\frac{\text{montant total payé}}{\text{montant réel hors remises}}$  est minimal.

**Question 27.** Déterminer pour chaque villageois le montant d'argent qu'il a payé au magasin Nook, ainsi que son ratio de rentabilité.