Laura DARENNE
Camille CLAVIER

Rapport :
Gestion d'un ensemble de bars

Exercice 1 :
On crée quatre tableaux, c'est-à-dire un par entité, qui sont 'Employes', 'Etablissements', 'Carte' et 'Ventes'.Voici les quatres tables :

Employes

matricule	nom	prenom	profession	nom_bar

Les données de chaque colonne ont été récupérées dans le fichier Employes.csv
Ici, l'attribut matricule qui reprend les données présentes dans la colonne Matricule est la clef primaire car il s'agit d'un identifiant unique par employé.

Etablissements

nom_bar	adresse	numTelephone	matricule_manager

Les données de chaque colonne ont été récupérées dans le fichier Etablissements.csv.
Ici, l'attribut nom_bar qui reprend les données de la colonne Nom_Bar est la clef primaire car chaque bar a un nom différent.

Carte

idBoisson	boisson	type	prix_EU	degre_BIERES	quantite_CL

Les données de chaque colonne ont été récupérées dans le fichier Carte.csv.
Ici, l'attribut idBoisson qui reprend les données de la colonne Id_Boisson est la clef primaire car l'identifiant de chaque boisson est unique.

Ventes

matricule	idBoisson	date

Les données de chaque colonne ont été récupérées dans le fichier Ventes.csv.

Ici, nous avons dû créer une clef primaire qui est idVente car il n'y avait pas d'identifiant unique dans le fichier csv.

Exercice 2 :

1. Pour avoir le nombre total de bars, nous comptons tous les bars de la table Etablissements avec COUNT. Nous allons avoir comme résultat une table qui contient le nombre de bars total. Nous avons afficher le résultat ainsi :

```
    print(f"\nnombre total de bars : {r[0]}")
```

Ici, r[0] correspond au résultat de COUNT(nom_bar):

```
nombre total de bars : 10
```

2. Pour avoir le nombre total d'employés, nous comptons les matricules de tous les employés présents dans la table Employes avec COUNT. Nous allons avoir comme résultat une table qui contient le nombre de matricule, autrement dit d'employés. Nous avons afficher le résultat ainsi :

```
    print(f"\nnombre total d'employés : {r[0]}")
```

Ici, r[0] correspond au résultat de COUNT(matricule):

```
nombre total d'employés : 120
```

3. Pour avoir les managers des bars, nous avons fait une jointure entre les tables Employes et Etablissements où les attributs nom_bar de chaque table sont égaux et les attributs matricule et matricule_manager qui sont aussi égaux.
   Nous avons comme résultat une table avec trois colonnes :

prenom	nom	nom_bar

   Nous avons affiché le résultat ainsi :

```
print(f"{r[1]} {r[0]} dirige le bar \"{r[2]}\".")
```

   - r[0] pour le prénom du manager.
   - r[1] pour nom du manager.
   - r[2] pour le bar du manager.

```
liste des managers de bars :
Faviet Daniel dirige le bar "Le Saphir".
```

4. Pour avoir le nombre d'employés pour chaque profession nous avons regroupé à l'aide de GROUP BY le total d'employés avec chaque profession grâce à leur matricule. Nous avons comme résultat une table avec deux colonnes, le nombre de matricule et la profession

COUNT(matricule)	profession

Nous l'avons affiché ainsi :

```
    print(f"{r[0]} {r[1]}s")
```

Ici, r[0] correspond au résultat de COUNT(matricule) et r[1] à la profession.

```
nombre d'employés pour chaque profession :
74 Barmaids
```

5. Pour avoir les revenus totaux du groupe nous avons fait un calcul dont le résultat est issu de la jointure des tables Ventes et Carte où chaque vente a le prix de la boisson indiquée. Nous faisons ensuite la somme des prix des boissons. Nous avons comme résultat une table avec 1 colonne, le revenu total.
   Nous l'avons affiché ainsi :

```
print(f"Les revenus du groupe sont de {r[0]} euros.")
```

Ici, r[0] correspond au résultat de ROUND(SUM(Carte.prix_EU):

```
revenu total du groupe :
Les revenus du groupe sont de 197615.6 euros.
```

Exercice 3 :

Pour obtenir le nombre de chaque employé et le montant total associé à ses ventes, on a regroupé par employé grâce à leur matricule présent dans les tables Ventes et Employes. Ensuite on compte le nombre de boissons vendues en comptant le nombre de ventes effectuées où chaque vente reçoit un id unique. Grâce à la jointure entre les tables Ventes et Carte, chaque vente à le prix de la boisson indiquée. On fait ensuite la somme des prix des boissons. On obtient une table avec quatre colonnes :

E.nom	E.prenom	COUNT(V.idBoisson)	ROUND(SUM(C.prix_EU),2)

Nous l'avons affiché ainsi :

```
print(f"{r[1]} {r[0]} a vendu {r[2]} boissons pour un total de {r[3]} euros.")
```

- r[1] correspond au prenom de l'employé.
- r[0] correspond au nom de l'employé.
- r[2] correspond au résultat de COUNT(V.idBoisson)
- r[3] correspond au résultat de ROUND(SUM(C.prix_EU),2)

```
nombre de boissons vendues par chaque employé :
Avamarti Forshaw a vendu 456 boissons pour un total de 2433.5 euros.
```

Exercice 4 :

1. Pour obtenir la date à laquelle le moins de ventes a été enregistré, nous avons regroupé le résultat par date, ce qui va nous permettre de faire un tri pour le nombre de boissons vendu par jour. Nous allons ensuite trier par ordre croissant grâce à ORDER BY et obtenir seulement le premier résultat grâce à LIMIT 1. Nous allons obtenir une table avec deux colonnes. Nous l'avons affiché ainsi :

```
30      print(f"Le {r[1]} a été vendu seulement {r[0]} boissons.")
```

- r[0] correspond au résultat de COUNT(idBoisson).
- r[1] correspond à la date

```
la date à laquelle le moins de vente a été enregistré :
Le 02/11/2022 a été vendu seulement 1180 boissons.
```

2. Pour obtenir la date à laquelle les bénéfices ont été les moins importants, nous avons regroupé le résultat par date, ce qui va nous permettre de faire un tri pour la somme des revenus journaliers. Nous allons ensuite trier par ordre croissant grâce à ORDER BY et obtenir seulement le premier résultat grâce à LIMIT 1. Nous allons obtenir une table avec deux colonnes. Nous l'avons affiché ainsi :

```
    print(f"Le {r[1]} a été fait un bénéfice de seulement {r[0]} euros.")
```

   - r[0] correspond au résultat de ROUND(SUM(prix_EU),2)
   - r[1] correspond à la date.

```
la date à laquelle les bénéfices ont été les moins importants :
Le 02/11/2022 a été fait un bénéfice de seulement 6240.4 euros.
```

Exercice 5 :

A partir de maintenant, le manager doit s'authentifier pour accéder aux informations de son bar, et uniquement son bar. Pour cela, il doit rentrer un identifiant et un mot de passe. Son identifiant sera son prénom,son nom, et son mot de passe sera son matricule.
Pour pouvoir vérifier que les identifiants qu'il a entrés sont les bons, nous avons créé une nouvelle table nommée Managers.

Managers
identifiant matricule

A chaque connexion, une requête se lance. On va chercher l'identifiant dans la table Managers tel que identifiant = "l'identifiant rentré par le manager" et matricule = "mot de passe entré par le manager". Avec la condition if, s'il y a un résultat à cette requête, c'est que les identifiants sont bons. S'il n'y a pas de résultat à cette requête, alors soit les identifiants sont incorrects, soit il s'agit d'une personne qui n'a pas les droits d'accès.

Ensuite, on émet une requête pour obtenir le nom du bar du manager qui s'est connecté. Comme son matricule se trouve déjà dans la variable mot_de_passe, il est facile de filtrer la requête dans la table Etablissements. La variable nom_bar prend le résultat de cette requête.

On reprend la requête de l'exercice 3 pour tester la variable. On obtient le nombre de boissons vendues par chaque employé du bar et leurs recettes associées.

Exercice 6 :

1. Pour obtenir le nombre de ventes effectué ce mois-ci pour les employés du manager et ce que cela représente, on fait une jointure entre les tables Employes et Ventes sur les attributs matricule et idBoisson. On fait ensuite un filtre avec le nom du bar du manager connecté et la somme du prix de chaque boisson vendue. On obtient une table contenant 2 colonnes. Nous l'avons affiché ainsi :

```
    print(f"\nVos employés ont vendu {r[0]} boissons pour un total de {r[1]} euros en Novembre.")
```

   - r[0] correspond au résultat de COUNT(idBoisson).
   - r[1] correspond au résultat de ROUND(SUM(prix_EU),2).

```
Vos employés ont vendu 5017 boissons pour un total de 26698.5 euros en Novembre.
```

2. Pour obtenir les bénéfices générés par chaque employé du bar, on fait une jointure on fait une jointure entre les tables Employes et Ventes sur les attributs matricule et idBoisson. On fait ensuite un filtre avec le nom du bar du manager connecté et la somme du prix de chaque boisson vendue. On groupe le résultat avec le matricule de l'employé présent dans la table Ventes. Nous l'avons affiché ainsi :

```
    print(f"{r[1]} {r[0]} a fait un bénéfice de {round(r[2], 2)} euros.")
```

   - r[0] correspond au nom de l'employé.
   - r[1] correspond au prénom de l'employé.
   - r[2] correspond au résultat de SUM(C.prix_EU)

```
Voici les bénéfices générés par chacun de vos employés en Novembre.
Shelli Baida a fait un bénéfice de 2556.5 euros.
```

Exercice 7 :

Après sa connexion, le manager est invité à saisir la date pour laquelle il veut les ventes et les bénéfices générés par ses employés.

```
date_saisie = input("\nTaper la  date de votre choix (au format jj/11/2022) : ")
```

1. Pour afficher le nombre de ventes effectué par les employés du manager et ce que cela représente à la date choisie nous reprenons la première requête de l'exercice 6 en ajoutant parmi les conditions la date saisie comme date de vente. Nous l'avons affiché ainsi :

```
print(f"\nLe {date_saisie}, vos employés ont vendu {r[0]} boissons pour un total de {r[1]} euros.")
```

- {date_saisie} correspond au résultat de l'input 'date_saisie'.
- r[0] correspond au résultat de COUNT(idBoisson).
- r[1] correspond au résultat de ROUND(SUM(prix_EU),2).

```
Le 26/11/2022, vos employés ont vendu 182 boissons pour un total de 951.0 euros.
```

2. Pour obtenir les bénéfices générés par chaque employé du bar à la date choisie nous reprenons la seconde requête de l'exercice 6 en ajoutant parmi les conditions la date saisie comme date de vente. Nous l'avons affiché ainsi :

```
print(f"{r[1]} {r[0]} a fait un bénéfice de {round(r[2], 2)} euros.")
```

- r[0] correspond au nom de l'employé.
- r[1] correspond au prénom de l'employé.
- r[2] correspond au résultat de SUM(C.prix_EU).

```
Voici les bénéfices générés par chacun de vos employés le 26/11/2022.
Shelli Baida a fait un bénéfice de 95.0 euros.
```

Exercice 8 :

1. Pour afficher les boissons les moins vendues dans l'établissement ce mois-ci on filtre avec le nom du bar du manager connecté puis on regroupe grâce à la jointure entre Carte et Ventes. Les ventes sont regroupées par le nom de boisson et la table Carte. On trie par le nombre de boissons vendu en fonction de leur nom et au nombre croissant. On a fixé la limite au dix premiers. Nous avons une table contenant 2 colonnes. Nous l'avons affiché ainsi :

```
60      print(f"{r[0]} n'a été vendu que {r[1]} fois.")
```

- r[0] correspond au nom de la boisson
- r[1] correspond au résultat du COUNT(idBoisson)

```
Voici la liste des 10 boissons qui se sont le moins bien vendues dans votre établissement au mois de Novembre.
Kir petillant n'a été vendu que 2 fois.
```

2. Pour afficher les employés ayant vendu le moins de boissons ce mois-ci on filtre avec le nom du bar du manager connecté puis on regroupe grâce à la jointure entre Employes et Ventes. Les ventes sont regroupées par employés. On trie par le nombre de boissons vendu en fonction du matricule de l'employé et au nombre croissant de boissons vendues. On a fixé la limite au cinq premiers. Nous avons une table contenant 2 colonnes. Nous l'avons affiché ainsi :

```
print(f"{r[1]} {r[0]} n'a vendu que {r[2]} boissons.")
```

- r[0] correspond au nom de l'employé.
- r[1] correspond au prénom de l'employé.
- r[2] correspond au résultat de COUNT(idBoisson).

Exercice 9 :

Le manager rentre ses identifiants. On vérifie que les identifiants sont bons.
Alors la variable nom_bar prend le nom du bar du manager qui s'est authentifié grâce à une requête. Cela permet de filtrer toutes les requêtes qui suivent.

Nous avons deux requêtes. On filtre toujours les requêtes pour n'avoir que les informations du bar du manager. On a à chaque fois une jointure entre les tables Employes, Carte et Ventes.

La première permet d'obtenir les boissons qui rapportent le plus d'argent dans le bar du manager au mois de Novembre. On regroupe les tuples par nom de boisson (soit l'attribut boisson de la table Carte). On trie les résultats en fonction de la somme des recettes de chaque boisson, de la boisson ayant rapporté le plus à la boisson ayant rapporté le moins. On limite la recherche aux 10 premiers résultats. On obtient la table suivante :

l'attribut boisson de la table Carte, soit le nom de la boisson = r[0]	La somme des recettes par boisson, soit ici la multiplication du prix de la boisson par son nombre de vente, arrondi au deuxième chiffre après la virgule = r[1]

La deuxième requête permet d'afficher les employés ayant rapporté le plus d'argent à l'établissement. On regroupe les tuples par employés, grâce à leur matricule. On trie les résultats selon la somme des recettes de chaque employé, des recettes les plus grandes et aux plus petites. On limite la recherche aux 5 premiers résultats, soit les 5 employés ayant rapporté le plus d'argent. On obtient la table suivante :

l'attribut nom se trouvant dans la table Employé = r[0]	l'attribut prenom se trouvant dans la table Employé = r[1]	La somme des recettes par employé, soit ici la somme du prix de chaque boisson vendue, arrondi au 2ème chiffre après la virgule = r[2]

Exercice 10 :

Le manager rentre ses identifiants. On vérifie que les identifiants sont bons.
Alors la variable nom_bar prend le nom du bar du manager qui s'est authentifié grâce à une requête. Cela permet de filtrer toutes les requêtes qui suivent.

On émet une requête pour afficher les employés qui ont vendu le plus de cocktails du jour et de bières en pression. Le premier filtre est le nom du bar du manager qui s'est authentifié. Le deuxième filtre permet de n'avoir que les boissons qui s'appellent "Cocktail du moment" ou "Blonde pression". Toutes les autres boissons n'apparaissent pas et ne sont pas comptabilisées

dans les résultats. On a une jointure entre les tables Employes, Carte et Ventes. On regroupe la recherche par employé, grâce à leur matricule. On trie les résultats pour selon le nombre de boissons vendues par employé, pour avoir comme premier résultat l'employé qui a vendu le plus de boissons, et en dernier résultat, celui qui en a vendu le moins. On limite la recherche aux 5 premiers résultats de façon à avoir les 5 employés qui ont vendu le plus de cocktail du jour ou de bière en pression. On obtient le tableau suivant :

l'attribut nom se trouvant dans la table Employé = r[0]	l'attribut prenom se trouvant dans la table Employé = r[1]	Le nombre de valeurs non nulles de l'idBoisson de la table Ventes, par boisson = r[2]

Exercice 11 :

Le manager rentre ses identifiants. On vérifie que les identifiants sont bons.
Alors la variable nom_bar prend le nom du bar du manager qui s'est authentifié grâce à une requête. Cela permet de filtrer toutes les requêtes qui suivent.

On émet une requête pour obtenir le degré d'alcool moyen consommé et la quantité d'alcool consommé dans l'établissement du manager au mois de Novembre. Il n'y a qu'un filtre qui permet d'obtenir que les résultats dans le bar du manager, grâce à la variable nom_bar. On a une jointure entre les tables Employes, Carte et Ventes. On obtient la table suivante :

Le nombre idBoisson se trouvant dans la table Vente, soit le nombre de boisson vendu au total = r[0]	La moyenne de tous les tuples degre_BIERES, associés aux boissons de type Bière de la table Vente = r[1]	La somme de tous les tuples quantite_CL, associés à chaque tuple idBoisson de la table Vente = r[2]

Nous n'avons le degré d'alcool que pour les bières. Nous précisons donc au manager dans le script et dans le texte affiché que le degré d'alcool moyen sur la totalité des bières.

Exercice 12 :

Le manager rentre ses identifiants. On vérifie que les identifiants sont bons.
Alors la variable nom_bar prend le nom du bar du manager qui s'est authentifié grâce à une requête. Cela permet de filtrer toutes les requêtes qui suivent.

Grâce à la commande input(), on rentre dans une variable input_chiffre le nombre de boissons que le manager souhaite enlever de la carte.

On effectue deux requêtes. Chaque requête voit ses résultats limités selon la variable input_chiffre, le nombre de boissons que le manager veut enlever de sa carte.

La première permet d'obtenir la liste moins_vendue avec le nom des boissons qui se sont le moins bien vendues. Les ventes sont regroupées par boisson et triées du nombre de boissons le moins vendues au plus vendues. On a une jointure entre les tables Ventes, Employes et Carte. On obtient la table suivante :

L'attribut boisson de la table Carte = r[0]	Le nombre de valeurs non nulles de l'idBoisson de la table Ventes, par boisson = r[1]

La deuxième permet d'obtenir la liste moins_benefice avec le nom des boissons qui ont rapporté le moins d'argent. On obtient la table suivante :

L'attribut boisson de la table Carte = r[0]	La somme des recettes par boisson, soit ici la multiplication du prix de la boisson par son nombre de vente, arrondi au deuxième chiffre après la virgule = r[1]

On demande au manager s'il préfère enlever de la carte les boissons qui se sont le moins bien vendues ou les boissons qui ont rapporté le moins d'argent. On a une jointure entre les tables Employes, Carte, et Ventes. Grâce à la condition if, on va appliquer la première ou la deuxième solution.

On vérifie que la suppression a bien eu lieu en affichant tous les tuples de la table Carte.