

# Bases de données avancées 1

## Travaux Pratiques n°1 : Requêtes SQL (Correction)

Université de la Nouvelle-Calédonie, 2024



**ERRATUM** : Le TP1 n'est pas à rendre. Cela vous permettra de prendre en main les logiciels de SGBD, les requêtes SQL ainsi que les requêtes algébriques.

## Introduction

Durant ces TPs, nous allons utiliser le SGBD MySQL par l'intermédiaire du logiciel XAMPP. Ce logiciel est un kit d'installation qui regroupe un serveur Apache, MySQL ainsi que des modules PHP et Perl. Pour démarrer XAMPP, il suffit d'exécuter le panneau de contrôle et de démarrer tous les composants. Ensuite, pour accéder à l'interface MySQL, il faut démarrer un navigateur web, saisir l'adress `localhost` et sélectionner dans le menu l'outil `phpMyAdmin` (interface d'administration de MySQL).

Pour ce TP, on considère la base de données ci-dessous, concernant des livraisons de journaux à des dépositaires. Le préfixe `N_` signifie numéro, et le suffixe `_J` journal. Les attributs `QTE_L` et `QTE_R` sont les quantités de journaux respectivement livrées et rendues (invendus).

Contraintes : Un journal est déterminé par son code, un dépositaire par son numéro ; et pour un dépositaire, une date et un journal, les quantités sont déterminées.

Schéma :

```
JOURNAL (CODE_J, TITRE, PRIX, TYPE_J, PERIODE, ADR_J)
LIVRAISON(N_DEP, CODE_J, DATE_L, QTE_L, QTE_R)
DEPOT(N_DEP, NOM_DEP, ADR)
```

## 1 Création de la base de données

Une fois l'interface `phpMyAdmin` lancée, créer la base de données `Poste` (champ "Créer une base de données").

## 2 Création des tables/rerelations de la base de données

Pour créer les tables, aller dans l'onglet SQL, copier le contenu du script de création des tables (ou importer directement le fichier dans l'interface) `createTablesPoste.sql`, et exécuter la requête.

## 3 Insertion des tuples dans les tables/rerelations

Pour insérer les tuples (les données) dans les relations, aller dans l'onglet SQL, copier le contenu du script d'insertion `insertTablesPoste.sql`, et exécuter la requête.

## 4 Interrogation de la base de données

Exprimer les requêtes suivantes en SQL. Exprimer les requêtes précédées d'un **astérisque** en langage algébrique relationnel.

1. \* Donner pour chaque journal son titre, son prix, son type et sa périodicité.

```
SELECT titre, prix, type_j, periode
FROM journal;
```

Requête algébrique :  $\pi_{titre, prix, type\_j, periode}(journal)$

2. Donner le nom des journaux triés par ordre alphabétique.

```
SELECT titre
FROM journal
ORDER BY titre;
```

3. \* Donner les codes des journaux mensuels.

```
SELECT code_j
FROM journal
WHERE periode = "mensuel";
```

Requête algébrique :  $\pi_{code\_j}(\sigma_{periode = "mensuel"}(journal))$

4. Donner tous les renseignements sur les journaux, trier le résultat par type et pour chaque type par prix.

```
SELECT *
FROM journal
ORDER BY type_j, prix;
```

5. \* Donner le code de chaque journal ainsi que les numéros des dépositaires auxquels il est livré.

```
SELECT code_j, n_dep
FROM livraison;
```

Requête algébrique :  $\pi_{type\_j, n\_dep}(livraison)$

6. \* Donner le prix minimum, le prix moyen et le prix maximum des journaux.

```
SELECT MIN(prix) AS "prix_min", AVG(prix) AS "prix_moy", MAX(prix) AS "prix_max"
FROM journal;
```

Requête algébrique :  $\gamma_{min(prix), avg(prix), max(prix)}(\pi_{prix}(journal))$

7. \* Donner le nombre de journaux différents, ainsi que le nombre de types de journaux.

```
SELECT COUNT(DISTINCT titre) AS "nb_journ_dif",
       COUNT(DISTINCT type_j) AS "nb_type_journ_dif"
FROM journal;
```

Requête algébrique :

$\gamma_{COUNT(titre), COUNT(type\_j)}(\pi_{titre, type\_j}(journal))$

8. \* Donner les codes des journaux dont le type est inconnu.

```
SELECT code_j
FROM journal
WHERE type_j = "";
```

Requête algébrique :  $\pi_{code\_j}(\sigma_{type\_j = \text{""}}(journal))$

9. Donner pour chaque type de journal, le prix minimum, le prix moyen et le prix maximum.

```
SELECT type_j, MIN(prix) AS "prix_min", AVG(prix) AS "prix_moy",
       MAX(prix) AS "prix_max"
FROM journal
GROUP BY type_j;
```

10. Donner les numéros des dépositaires qui livrent plus de 15 journaux en moyenne.

```
SELECT n_dep
FROM livraison
GROUP BY n_dep
HAVING AVG(qte_l) > 15;
```

11. Donner le nom des villes où sont domiciliés au moins 5 journaux.

```
SELECT adr_j
FROM journal
GROUP BY adr_j
HAVING COUNT(titre) >= 5;
```

12. Donner le nombre de villes où sont domiciliés au moins 4 journaux.

```
SELECT COUNT(DISTINCT adr_j) AS "Nombre de ville"
FROM journal
WHERE adr_j IN (
  SELECT adr_j
  FROM journal
  GROUP BY adr_j
  HAVING COUNT(DISTINCT code_j) >= 4
);
```

13. \* Donner les codes des journaux livrés à "Bures" (2 manières).

```
SELECT DISTINCT code_j
FROM depot, livraison
WHERE depot.n_dep = livraison.n_dep AND adr = "Bures";
ou
SELECT DISTINCT code_j
FROM livraison
WHERE n_dep IN (
  SELECT n_dep
  FROM depot
  WHERE adr = "BURES");
```

Requête algébrique :  $\pi_{code\_j}(\sigma_{adr = \text{"Bures"}}(depot \bowtie_{depot.n\_dep = livraison.n\_dep} livraison))$

14. Donner les codes des journaux livrés au dépositaire "Les Brasseurs".

```
SELECT DISTINCT code_j
FROM livraison, depot
WHERE livraison.n_dep = depot.n_dep AND nom_dep = "LES BRASSEURS";
```

15. \* Donner le titre des journaux livrés au dépositaire "Les Brasseurs".

```
SELECT DISTINCT titre
FROM journal, livraison, depot
WHERE livraison.n_dep = depot.n_dep
  AND journal.code_j = livraison.code_j
  AND nom_dep = "LES BRASSEURS";
```

Requête algébrique :

$$T_1 = \text{journal} \bowtie_{\text{journal.code}_j = \text{livraison.code}_j} \text{livraison} \bowtie_{\text{livraison.n\_dep} = \text{depot.n\_dep}} \text{depot}$$
$$\pi_{\text{titre}}(\sigma_{\text{nom\_dep} = \text{"LES BRASSEURS"}}(T_1))$$

16. \* Donner le nombre de journaux livrés au dépositaire "Les Brasseurs".

```
SELECT COUNT(DISTINCT titre) AS "Nombre journaux livrés au dep Les Brasseurs"
FROM journal, livraison, depot
WHERE livraison.n_dep = depot.n_dep
      AND journal.code_j = livraison.code_j
      AND nom_dep = "LES BRASSEURS";
```

Requête algébrique :

$$T_1 = \text{journal} \bowtie_{\text{journal.code}_j = \text{livraison.code}_j} \text{livraison} \bowtie_{\text{livraison.n\_dep} = \text{depot.n\_dep}} \text{depot}$$
$$\pi_{\text{titre}}(\sigma_{\text{nom\_dep} = \text{"LES BRASSEURS"}}(T_1))$$

17. Donner les codes et les titres des journaux dont le prix est supérieur à deux fois le prix minimum.

```
SELECT DISTINCT code_j, titre
FROM journal
WHERE prix > (SELECT 2*MIN(prix)
              FROM journal);
```

18. Donner les numéros des dépositaires qui reçoivent plusieurs journaux.

```
SELECT n_dep
FROM livraison
GROUP BY n_dep
HAVING COUNT(DISTINCT qte_r) > 1;
```

19. Donner les numéros des dépositaires ayant au moins 3 livraisons d'au moins 50 journaux.

```
SELECT n_dep
FROM livraison
GROUP BY n_dep
HAVING COUNT(DISTINCT qte_l) >= 3
      AND COUNT(DISTINCT qte_r) >= 50;
```

20. Donner les titres des journaux livrés chez au moins 5 dépositaires parisiens différents.

```
SELECT titre
FROM journal, livraison, depot
WHERE journal.code_j = livraison.code_j
      AND livraison.n_dep = depot.n_dep
      AND adr = "Paris"
GROUP BY titre
HAVING COUNT(DISTINCT livraison.n_dep) >= 5;
```

21. \* Donner les noms des dépositaires qui ne reçoivent pas "Libération".

```
SELECT DISTINCT nom_dep
FROM depot
WHERE n_dep NOT IN (
  SELECT n_dep
  FROM livraison, journal
  WHERE livraison.code_j = journal.code_j
        AND titre = "LIBERATION");
```

Requête algébrique :

$\pi_{nom\_dep}(\sigma_{n\_dep \notin \pi_{n\_dep}(\sigma_{titre = "LIBERATION"}(livraison \bowtie_{livraison.code\_j = journal.code\_j} journal))}(depot))$

22. \* Donner les numéros des dépositaires qui ne reçoivent que des hebdomadaires.

```
SELECT DISTINCT n_dep
FROM livraison, journal
WHERE livraison.code_j = journal.code_j
  AND periode = "HEBDO"
  AND n_dep NOT IN (
  SELECT n_dep
  FROM livraison, journal
  WHERE livraison.code_j = journal.code_j
    AND periode <> "HEBDO");
```

Requête algébrique :

$T_1 = livraison \bowtie_{livraison.code\_j = journal.code\_j} journal$

$\pi_{n\_dep}(\sigma_{periode = "HEBDO"} \wedge n\_dep \notin \pi_{n\_dep}(\sigma_{periode \neq "HEBDO"}(T_1)))(livraison \bowtie_{livraison.code\_j = journal.code\_j} journal)$

23. \* Donner les noms des journaux que le dépositaire 310 reçoit.

```
SELECT titre
FROM livraison, journal
WHERE n_dep = 310
  AND livraison.code_j = journal.code_j;
```

Requête algébrique :

$\pi_{titre}(\sigma_{n\_dep = 310}(livraison \bowtie_{livraison.code\_j = journal.code\_j} journal))$

24. \* Donner les noms des journaux que le dépositaire 310 ne reçoit pas.

```
SELECT titre
FROM journal
WHERE code_j NOT IN (
  SELECT code_j
  FROM livraison
  WHERE n_dep = 310);
```

Requête algébrique :  $\pi_{titre}(\sigma_{code\_j \notin \pi_{code\_j}(\sigma_{n\_dep = 310}(livraison))}(journal))$

25. Donner les noms des mensuels féminins.

```
SELECT titre
FROM journal
WHERE type_j = "FEMININ"
  AND PERIODE = "MENSUEL";
```

26. \* Donner les noms des mensuels féminins que 310 ne reçoit pas.

```
SELECT titre
FROM journal
WHERE type_j = "FEMININ"
  AND periode = "MENSUEL"
  AND code_j NOT IN (
  SELECT code_j
  FROM livraison
  WHERE n_dep = "310");
```

Requête algébrique :

$$\pi_{\text{titre}} \left( \sigma_{\text{type\_j} = \text{"FEMININ"} \wedge \text{periode} = \text{"MENSUEL"} \wedge \text{code\_j} \notin \pi_{\text{code\_j}}(\sigma_{\text{n\_dep} = \text{"310"}}(\text{livraison}))(\text{journal}) \right)$$

27. \* Donner les noms des dépositaires qui reçoivent tous les mensuels féminins.

```
SELECT nom_dep
FROM depot WHERE NOT EXISTS (
  SELECT *
  FROM journal
  WHERE type_j = "FEMININ"
    AND PERIODE = "MENSUEL"
    AND code_j NOT IN (
      SELECT code_j
      FROM livraison
      WHERE depot.n_dep = livraison.n_dep)
);
```

Requête algébrique :

$$\pi_{\text{nom\_dep}} \left( \sigma_{\text{n\_dep} \notin \pi_{\text{n\_dep}}(\text{depot} \bowtie_{\text{depot.n\_dep} = \text{livraison.n\_dep}} (\sigma_{\text{type\_j} = \text{"FEMININ"} \wedge \text{periode} = \text{"MENSUEL"} \wedge \text{code\_j} \notin \pi_{\text{code\_j}}(\text{livraison}))(\text{journal})))}(\text{depot}) \right)$$

28. \* Donner les noms des dépositaires parisiens.

```
SELECT nom_dep
FROM depot
WHERE adr = "PARIS";
```

Requête algébrique :  $\pi_{\text{nom\_dep}}(\sigma_{\text{adr} = \text{"PARIS"}}(\text{depot}))$

29. \* Donner les noms des dépositaires parisiens qui reçoivent tous les journaux.

```
SELECT n_dep
FROM depot
WHERE adr = "PARIS"
  AND NOT EXISTS (
    SELECT *
    FROM journal
    WHERE code_j NOT IN (
      SELECT code_j
      FROM livraison
      WHERE depot.n_dep = livraison.n_dep)
  );
```

Requête algébrique :

$$\pi_{\text{n\_dep}} \left( \sigma_{\text{adr} = \text{"PARIS"} \wedge \text{n\_dep} \notin \pi_{\text{n\_dep}}(\text{depot} \bowtie_{\text{depot.n\_dep} = \text{livraison.n\_dep}} (\sigma_{\text{code\_j} \notin \pi_{\text{code\_j}}(\text{livraison}))(\text{journal})))}(\text{depot}) \right)$$