

MAIN4

Bases de Données

TP 1 : SQL (Casernes)

(Correction enseignant)

Une caserne est un endroit où se retrouvent des pompiers et où sont stockés des camions. Toute caserne a une adresse, un numéro pour l'identifier, une capacité maximale humaine et une capacité de stockage de camions. Chaque caserne numérote ses camions et ses pompiers.

Un pompier a donc son numéro attribué par la caserne où il est affecté, ainsi qu'un nom et une adresse. Une adresse se compose de quatre parties : un numéro de rue, un nom de rue, un code postal et un nom de ville. De plus, pour chaque adresse, nous avons une estimation du kilométrage entre cette adresse et la plus proche caserne (pour des raisons de logistique, on considèrera une unique caserne plus proche pour chaque adresse).

Pour chacune des villes, nous connaissons son nombre d'habitants. Nous connaissons également les casernes qui protègent une ville, et les villes que protège une caserne.

Un camion a un numéro attribué par la caserne où il est stocké, ainsi qu'un nombre de place maximum et un modèle. Parmis les camions, nous distinguons les citernes pour lesquelles nous voulons connaître également leur contenance.

Un modèle de camion s'identifie par son nom (Premium210, ...). Pour chaque modèle, nous connaissons le type de camion (lourd-rapide, ...) ainsi que sa motorisat ion (210hp, ...) et son fabricant identifié par la marque (Renault,...).

Nous savons pour chaque fabricant l'adresse de son siège social, ainsi que son délai de livraison moyen.

1 Modélisation

À partir du code SQL fourni de création des tables et de spécification des contraintes, donnez une modélisation en diagramme de classes UML de l'énoncé.

2 Chargement de la base de données

```
Lancez sqlite:
> sqlite3
Attachez la base "caserne.db" (équivalent SQLite du code SQL vu ci-dessus):
sqlite> ATTACH DATABASE 'caserne.db'AS 'Caserne';
Activation des clés étrangères:
sqlite> PRAGMA foreign_keys = ON;
Sortie en colonnes avec entête:
sqlite> .mode column
sqlite> .headers ON
```

3 Contraintes

Proposez trois requêtes d'insertion ou de suppression qui génèrent une erreur pour violation de contraintes.

```
Correction: Par exemple

Violation de clé étrangère:

INSERT INTO Protege VALUES (9, 'Paris', 98000);

Violation de clé primaire:

INSERT INTO Caserne VALUES (1, 2, 3, 4, 'rue', 75000, 'Paris');

Violation de CHECK - ne fonctionne pas en MySQL, pas de CHECK, pas fonctionne avec SQLite:

INSERT INTO Camion VALUES (2, 7, 30, 'Daminator');
INSERT INTO Citerne VALUES (2, 7, 120000);

Violation de clé étrangère:

DELETE FROM Caserne WHERE Id_caserne = 1;
```

4 Requêtes de sélection

4.1 Prise en main

Répondez aux questions suivantes par une requête sur la base fournie :

1. (SELECT) Donnez la liste de toutes les villes de la base.

(Le DISTINCT permet d'enlever les doublons dus au fait qu'une ville peut apparaître plusieurs fois avec des codes postaux différents).

2. (SELECT) Donnez le prénom et le nom de chaque pompier.

Correction : SELECT Prenom, Nom FROM Pompier;

3. (WHERE) Quelles sont les casernes qui se situent dans la ville Shadok?

Correction:

SELECT Id_caserne FROM Caserne WHERE Nom_ville = "Shadok";
+-----+
| Id_caserne |
+-----+
| 3 |
| 4 |
+-----+

4. (WHERE) Quelles sont les villes protégées par la caserne 1?

```
Correction :
SELECT Nom_ville FROM Protege WHERE Id_caserne = 1;
+-----+
| Nom_ville |
+-----+
| Brignoles |
| Draguignan |
| Le Luc |
+------+
```

5. (ORDER) Donnez la liste des villes classées de la plus peuplée à la moins peuplée. (on ne cherchera pas à regrouper les populations des villes apparaissant plusieurs fois)

Correction:

```
SELECT * FROM Ville ORDER BY Nb_hab DESC;
```

Nom_ville	Nb_hab	CP
Lajoie	25000	83450
Montreuil	20400	83100
Draguignan	20000	83240
Trantown	18500	83220
Shadok	14630	83666
Le Luc	4580	83620
Papou	3655	83220
Draguignan	2400	83244
Brignoles	1650	83620
Draguignan	300	83242
Pierrefiques	134	83620
+	+	++

6. (JOINTURE) Quel est le modèle et le nombre de places des citernes ? (Rappel : une citerne est un camion qui a la particularité d'être une citerne, et certains camions n'en sont pas).

Correction:

SELECT Ci.Id_caserne, Ci.Id_camion, Nb_places, Modele FROM Citerne Ci JOIN Camion Ca ON Ci.Id_camion = Ca.Id_camion AND Ci.Id_caserne = Ca.Id_caserne;

Id_caserne	Id_camion	Nb_places	Modele
1	1	38	Daminator
1	2	52	Laignel_vnr
1	3	152	Truite
2	1	95	Laignel_vnr
2	2	87	Truite
2	4	95	GrosColosse
2	7	30	Daminator
3	1	55	Daminator

7. (JOINTURE) Quelles sont les adresses des casernes qui protègent "Draguignan"?

Correction:

SELECT P.Nom_ville, Ca.Id_caserne, Ca.Num_rue, Ca.Nom_rue, Ca.CP, Ca.Nom_ville FROM Caserne Ca
JOIN Protege P ON Ca.Id_caserne = P.Id_caserne WHERE P.Nom_ville = "Draguignan";

Nom_ville	+ Id_caserne +			CP	Nom_ville
Draguignan			Alouette	•	
Draguignan Draguignan			Canard Chouette	•	
Draguignan	2	136	Pigeon	83242	Draguignan

8. (AGRÉGRATION)

(a) (COUNT) Combien y a-t-il de camions citernes dans la base?

```
Correction :
SELECT COUNT(*) FROM Citerne;
+----+
| COUNT(*) |
+----+
| 8 |
+-----+
```

(b) (MAX) Quelle est la plus grande contenance de citerne disponible?

9. (GROUP BY) Quelle est la contenance moyenne des citernes pour chaque caserne ?

Correction:

 ${\tt SELECT\ Id_Caserne,\ AVG(Contenance)\ FROM\ Citerne\ GROUP\ BY\ Id_caserne;}$

10. (GROUP BY) Combien de casernes protègent chaque ville?

Correction:

```
SELECT Nom_ville, COUNT(*) FROM Protege GROUP BY Nom_ville;
+-----+
| Nom_ville | COUNT(*) |
+-----+
| Brignoles | 2 |
| Draguignan | 4 |
| Le Luc | 2 |
| Papou | 1 |
| Shadok | 1 |
```

11. (GROUP BY / HAVING) Quelles sont les casernes qui cumulent des contenances de citerne de plus de 2000 L?

```
Correction:

SELECT Id_caserne, SUM(Contenance) AS cumul FROM Citerne GROUP BY Id_caserne HAVING cumul > 2000;
+-----+
| Id_caserne | cumul |
+-----+
| 1 | 2300 |
| 2 | 122950 |
+-----+
```

12. (LIKE) Quels sont les pompiers dont le nom commence par un M?

4.2 Suite

Répondez aux questions suivantes par une requête sur la base fournie :

Commentaire: Plusieurs solutions existent à chaque fois. Les solutions proposées utilisent rarement les "JOIN" car nous ne les avons pas encore vu en cours au moment de la première séance de TP, mais c'est bien sûr possible.

1. \bigstar Quel est le nombre de casernes?

```
Correction:

SELECT COUNT(*) FROM Caserne;
+-----+
| COUNT(*) |
+------+
| 4 |
+------+
```

2. * Quels sont les pompiers des casernes situées à Draguignan?

3. ★★★ Quelles sont les casernes protégeant à la fois Draguignan et Le Luc?

```
Correction:
SELECT Caserne.Id_caserne FROM Caserne, Protege P1, Protege P2
WHERE Caserne.Id_caserne = P1.Id_caserne
  AND Caserne.Id_caserne = P2.Id_caserne
  AND P1.Nom_ville='Draguignan'
  AND P2.Nom_ville='Le Luc';
ou
SELECT Caserne.Id_caserne FROM Caserne
WHERE EXISTS(SELECT * FROM Protege P1
             WHERE Caserne.Id_caserne = P1.Id_caserne
               AND P1.Nom_ville='Draguignan')
  AND EXISTS(SELECT * FROM Protege P2
            WHERE Caserne.Id_caserne = P2.Id_caserne
               AND P2.Nom_ville='Le Luc');
| Id_caserne |
           1 |
          4 |
```

4. ★★ Quels sont les pompiers de la caserne 3 habitant à plus de 5 kilomètres d'une caserne?

Correction:

```
| 1 | Montant | Yves |
| 4 | Jumper | Jolly |
```

Correction:

Correction:

Correction:

5. $\star\star$ Quels sont les pompiers habitant Le Luc ou des villes ≥ 20000 habitants?

6. ★★ Quel est le délai moyen de livraison pour chaque fabricant de citernes de moins de 1000 litres? (attention, comme indiqué ci-dessus, le champ Delai de la table Fabricant est déjà une valeur moyenne!)

7. ★★★ Classez par ordre décroissant le temps de livraison de camions moyen par caserne. (ici c'est le délai moyen par caserne, il faut donc bien calculer une moyenne...)

```
SELECT Ca.Id_caserne, AVG(F.Delai) Delai
FROM Camion Ca, Modele M, Fabricant F
WHERE Ca.Modele = M.Nom_modele
AND F.Nom_fabricant=M.Nom_fabricant
GROUP BY Ca.Id_caserne
```

ORDER BY Delai DESC; +-----+ | Id_caserne | Delai |

```
| 3 | 888.0000 |
| 1 | 832.5000 |
| 2 | 799.2000 |
| 4 | 666.0000 |
```

8. $\star\star$ Quel est le nombre de pompiers par caserne?

9. ★★ Quelles sont les casernes n'ayant qu'un seul pompier?

```
Correction:

SELECT Ca.Id_caserne, COUNT(P.Id_Pompier) Compte
FROM Caserne Ca, Pompier P
WHERE P.Id_caserne = Ca.Id_caserne
GROUP BY Ca.Id_caserne HAVING Compte = 1;
+-----+

| Id_caserne | Compte |
+-----+

| 4 | 1 |
| 1 | 1 |
| 1 | 1 |

En Postgres, pas d'alias dans le HAVING, donc:

SELECT Ca.Id_caserne, COUNT(P.Id_Pompier) Compte
FROM Caserne Ca, Pompier P
WHERE P.Id_caserne = Ca.Id_caserne
```

GROUP BY Ca.Id_caserne HAVING COUNT(P.Id_Pompier) = 1;

10. $\star\star\star\star$ Dans quelle(s) ville(s) trouve-t-on la (les) caserne(s) avec la (les) citerne(s) de plus grosse contenance?

```
Correction :
SELECT Id_Caserne, Nom_ville
FROM Caserne Ca
WHERE Id_Caserne IN
   (SELECT Id_Caserne
   FROM Citerne
   WHERE Contenance = (SELECT MAX(Contenance) FROM Citerne));
```

11. ★★★★ Quelles sont les casernes ayant atteint leur capacité maximale humaine?

12. ★★★★ Quels sont les pompiers qui ne travaillent pas dans la ville où ils habitent?

Correction:

```
SELECT P.Id_Pompier, P.Id_caserne, P.Nom, P.Prenom, P.nom_ville Habite, Ca.nom_ville Travaille
FROM Pompier P, Caserne Ca
WHERE P.Id_caserne = Ca.Id_caserne
AND P.Nom_ville <> Ca.Nom_ville;
```

Id_Pompier	Id_caserne	Nom	Prenom	+ Habite +	Travaille
2			Sylvain		Shadok
3	3	Savairien	Jean	Papou	Shadok
4	3	Jumper	Jolly	Le Luc	Shadok
1	1	Marchand	Louis	Draguignan	Le Luc
1		Antoinette			Shadok
1	2	Valjean	Jean	Papou	Draguignan
2	2	Conchon	Sylvain	Le Luc	Draguignan

13. ★★★★ Quelles sont les villes qui partagent leur code postal avec une autre ville? Quelles sont les villes auxquelles plusieurs codes postaux sont attribués?

Correction:

14. ★★★★★ Quelles sont les marques des camions qui sont utilisées dans toutes les casernes?

```
Correction:
SELECT Nom_fabricant
FROM Fabricant
WHERE NOT EXISTS
   (SELECT Ca1.Id_caserne ID
    FROM Caserne Ca1
    WHERE NOT EXISTS
       (SELECT * FROM Camion Ca, Modele M
        WHERE Ca.Modele = M.Nom_modele
          AND Ca.Id_caserne = Ca1.Id_caserne
          AND M.Nom_fabricant=Fabricant.Nom_fabricant
  );
| Nom_fabricant |
| Mercedes_benz |
(on peut segmenter la réflexion en cherchant d'abord (par exemple) les casernes qui n'ont pas de camion
Peugeot)
```

15. ★★★ Donnez le nombre de chaque type d'habitation pour chaque ville.

```
Correction:

SELECT Nom_ville, Type_habitation, COUNT(Type_habitation) AS Compte FROM Adresse
GROUP BY Nom_ville, Type_habitation
ORDER BY Nom_ville, Type_habitation;

nom_ville type_habitation compte
Draguignan Caserne 3
Draguignan HLM 1
Draguignan Pavillon 1
Le Luc Caserne 2
```

```
      Le Luc
      Pavillon
      2

      Montreuil
      Caserne
      1

      Montreuil
      HLM
      2

      Montreuil
      Pavillon
      1

      Papou
      HLM
      1

      Papou
      Pavillon
      1

      Shadok
      Caserne
      2

      Shadok
      Pavillon
      2

      Trantown
      Pavillon
      1
```

16. (a) ★★ Listez par ordre décroissant les casernes en fonction du nombre des pompiers y travaillant

(b) ★★ Quelle est la première caserne de la liste précedente?

```
Correction: Mot-clé LIMIT, non vu en cours
SELECT P.Id_caserne, COUNT(*) Taille
FROM Pompier P
GROUP BY P.Id_caserne
ORDER BY Taille DESC
LIMIT 1;
| Id_caserne | Taille |
  -----+
    3 | 4 |
Sinon,\ en\ Postgres:
WITH TableTaille AS
  (SELECT P.Id_caserne, COUNT(*) Taille
  FROM Pompier P GROUP BY P.Id_caserne)
SELECT Id_caserne, Taille
FROM TableTaille
WHERE Taille = (SELECT MAX(Taille) FROM TableTaille);
En MySQL (WITH non reconnu):
SELECT Id_caserne, Taille
FROM
  (SELECT P.Id_caserne, COUNT(*) Taille
  FROM Pompier P
  GROUP BY P.Id_caserne) TableTaille
WHERE Taille =
  (SELECT MAX(Taille)
  FROM
```

```
(SELECT P.Id_caserne, COUNT(*) Taille
FROM Pompier P
GROUP BY P.Id_caserne) TableTaille);
```

17. ★★★ Donnez pour chaque caserne le volume total d'eau de ses citernes (y compris les casernes ne possédant pas de citerne).

Correction:

SELECT Id_caserne, SUM(Contenance) FROM Citerne GROUP BY Id_caserne;

Id_caserne	SUM(Contenance)
1 2 3	2950

 $Mais\ de\ cette\ façon\ on\ n'a\ aucune\ trace\ de\ la\ caserne\ 4\ (aucune\ citerne).\ Pour\ avoir\ l'info\ que\ la\ caserne\ 4\ n'a\ pas\ de\ citerne\ (avec\ LEFT\ OUTER\ JOIN\ qu'on\ n'aura\ peut-être\ pas\ encore\ vu\ en\ cours):$

SELECT Caserne.Id_caserne, SUM(Contenance)
FROM Caserne LEFT OUTER JOIN Citerne ON Caserne.Id_caserne = Citerne.Id_caserne
GROUP BY Caserne.Id_caserne;

Id_caserne	SUM(Contenance)
1	2000
2 3	
4	NULL

ou, encore mieux pour éviter le vilain NULL :

```
SELECT Caserne.Id_caserne, COALESCE(SUM(Contenance),0)
FROM Caserne LEFT OUTER JOIN Citerne ON Caserne.Id_caserne = Citerne.Id_caserne
GROUP BY Caserne.Id_caserne;
```

Sans le OUTER JOIN il faut faire des sous-requêtes puis un UNION.