

Module 5 - TD: les différentes opérations de l'algèbre relationnelle en pratique: opérateurs, jointures et fonctions de groupes

Paul Schuhmacher

version: 1.0

Préparation du schéma et des données

- **Créer** la base de données `mod5jointures` sur laquelle votre utilisateur `dev` a tous les droits.
- **Créer** les relations `Compagnie` et `Pilote`
- **Insérer** les données préparées

```
USE mod5jointures;

CREATE TABLE Compagnie
(comp VARCHAR(4), nrue SMALLINT(3), rue VARCHAR(20), ville VARCHAR(15), nomComp
  VARCHAR(15),
  CONSTRAINT pk_Compagnie PRIMARY KEY(comp));

CREATE TABLE Pilote
(brevet VARCHAR(6), nom CHAR(20), nbHVol DECIMAL(7,2), compa VARCHAR(4),
  chefPil VARCHAR(6),
  CONSTRAINT pk_Pilote PRIMARY KEY(brevet),
  CONSTRAINT fk_Pil_compa_Comp FOREIGN KEY(compa) REFERENCES Compagnie(comp),
  CONSTRAINT fk_Pil_chefPil_Pil FOREIGN KEY(chefPil) REFERENCES Pilote(brevet));

INSERT INTO Compagnie VALUES ('AF', 124, 'Port Royal', 'Paris', 'Air
  France');
INSERT INTO Compagnie VALUES ('SING', 7, 'Camparols', 'Singapour', 'Singapore
  AL');
INSERT INTO Compagnie VALUES ('CAST', 1, 'G. Brassens', 'Blagnac', 'Castanet
  AL');

INSERT INTO Pilote VALUES ('PL-4', 'Henri Alquié', 3400, 'AF', NULL);
INSERT INTO Pilote VALUES ('PL-1', 'Pierre Lamothe', 450, 'AF', 'PL-4');
INSERT INTO Pilote VALUES ('PL-2', 'Didier Linxe', 900, 'AF', 'PL-4');
INSERT INTO Pilote VALUES ('PL-3', 'Christian Soutou', 1000, 'SING', NULL);
```

Jointures internes ([INNER JOIN| JOIN])

Équijointure

1. **Dessiner** le *schéma conceptuel* (formalisme UML) à partir des requêtes SQL
2. **Identifier** les clefs primaires et étrangères.

Écrire des requêtes pour extraire les données suivantes:

3. L'identité des pilotes de la compagnie de nom Air France ayant plus de 500 heures de vol (requête R1)
4. Les coordonnées des compagnies qui embauchent des pilotes de moins de 500 heures de vol (requête R2)

Autojointure

Cas particulier de l'*équijointure*, l'*autojointure* (*self-join*) relie une table à *elle-même* (pratique dans le cas d'associations réflexives).

Nous allons extraire les données suivantes:

5. L'identité des pilotes placés sous la responsabilité de 'Alquié' (R3)
6. La somme des heures de vol des pilotes placés sous la responsabilité des chefs pilotes de la compagnie de nom 'Air France' (R4)

Plus difficile, plusieurs réponses possibles. Découper la requête en sous requêtes.

On découpe la requête en plusieurs petites requêtes: déjà, il faut trouver les pilotes qui ont un chef à 'Air France'.

Il faut commencer par trouver les pilotes qui ont un chef et récupérer le chef, puis...

Inéquijointure

7. Les pilotes ayant plus d'expérience (i.e *d'heures de vol*) que le pilote de numéro de brevet 'PL-2'. (R5)
8. Le titre de qualification des pilotes en raisonnant sur la comparaison des heures de vol avec un ensemble de référence, ici la table HeuresVol (R6). Dans notre exemple, il s'agit par exemple de retrouver le fait que le premier pilote est débutant. La table d'heures de référence est définie par la relation suivante: HeuresVol: ****titre****, basnbHVol (borne min), hautnbHVol (borne max) avec les valeurs suivantes:
 - **Débutant**: entre 0 et 500 heures
 - **Initié**: entre 501 et 1000 heures
 - **Expert**: entre 1001 et 5000 heures

Jointures externes ([LEFT|RIGHT] OUTER JOIN)

9. La liste des compagnies aériennes et leurs pilotes (nom compagnie, nom pilote) *même* les compagnies n'ayant pas de pilote (R7)

Vous ne pouvez pas réaliser cette requête sans une jointure externe. Quand vous identifiez le mot “même”, ça sent la jointure externe. Car cela veut dire faire remonter des enregistrements qui ne satisfont pas une condition donnée (le prédicat de la clause ON)

10. La liste des pilotes et leurs qualifications (le type d'avion qu'un pilote peut piloter en fonction de son brevet), *même* les pilotes n'ayant pas encore de qualification (R8). Ces données vous sont fournies par le script SQL suivant:

```
-- Créer la table de Qualifications et insérer les données
CREATE TABLE Qualifs
(brevet VARCHAR(6), typeAv CHAR(4), validite DATE);
INSERT INTO Qualifs VALUES ('PL-4', 'A320', '2005-06-24');
INSERT INTO Qualifs VALUES ('PL-4', 'A340', '2005-06-24');
INSERT INTO Qualifs VALUES ('PL-2', 'A320', '2006-04-04');
INSERT INTO Qualifs VALUES ('PL-3', 'A330', '2006-05-13');
```

Opérations ensemblistes: Union (UNION), intersection (INTERSECT) et différence (NOT IN)

Créer les tables suivantes contenant les avions, et insérer les données.

```
-- Création de deux tables pour les avions des compagnies
CREATE TABLE AvionsdeAF
(immat CHAR(6), typeAvion CHAR(10), nbHVol DECIMAL(10,2),
CONSTRAINT pk_AvionsdeAF PRIMARY KEY (immat));

CREATE TABLE AvionsdeSING
(immatriculation CHAR(6), typeAv CHAR(10), prixAchat DECIMAL(14,2),
CONSTRAINT pk_AvionsdeSING PRIMARY KEY (immatriculation));

-- Insertion des données
INSERT INTO AvionsdeAF VALUES ('F-WTSS', 'Concorde', 6570);
INSERT INTO AvionsdeAF VALUES ('F-GLFS', 'A320', 3500);
INSERT INTO AvionsdeAF VALUES ('F-GTMP', 'A340', NULL);

INSERT INTO AvionsdeSING VALUES ('S-ANSI', 'A320', 104500);
INSERT INTO AvionsdeSING VALUES ('S-AVEZ', 'A320', 156000);
INSERT INTO AvionsdeSING VALUES ('S-MILE', 'A330', 198000);
INSERT INTO AvionsdeSING VALUES ('F-GTMP', 'A340', 204500);
```

11. Les types d'avion que les deux compagnies exploitent en commun.

12. Les appareils utilisés par les deux compagnies.
13. Tous les types d'avions que les deux compagnies exploitent.
14. La même que la requête précédente, mais **avec les duplicatas**.
15. Les types d'avions exploités par la compagnie 'SING' mais pas par la compagnie 'AF' ?

Opération de regroupement et fonctions de groupes (*aggregate functions*)

A l'aide des clauses **ORDER BY**, **GROUP BY** et **HAVING**, **écrire** les requêtes qui retournent les résultats suivants.

16. La somme d'heures de vol par compagnie, ordonné du plus petit au plus grand (utiliser la fonction de groupe **SUM**)
17. Le nombre moyen et l'écart type du nombre d'heures de vol par compagnie (utiliser les fonctions de groupe **AVG** et **STDDEV**)
18. Le nombre d'heures de vol le plus élevé pour chaque compagnie (utiliser la fonction **MAX**).
19. Les compagnies et le nombre de pilotes des compagnies ayant plus d'un pilote (utiliser **HAVING**)

Pour en savoir plus sur les clauses et les fonctions de regroupements, [lisez cette partie de la documentation](#). Voir [la liste des fonctions de groupe](#).

Références

Aller plus loin sur les sous-requêtes:

- [Common Table Expressions \(CTE\)](#), permettent d'aliaser des requêtes pour les utiliser comme sous requête. S'écrit avec la clause **WITH**. **Les CTE sont à privilégier par rapport aux sous-requêtes**: plus lisibles, peuvent être réutilisées.
- [L'implémentation de GROUP BY en MySQL](#)