

Module 6 - TD: Vues, fonctions et procédures stockées

1. Création et utilisation des vues

Pré-requis

Créer un nouveau schéma `mod6externe` (ou base de données en MySQL), en tant qu'utilisateur root et donner les droits à l'utilisateur dev (comme d'habitude)

```
-- as root
CREATE DATABASE mod6externe;
GRANT ALL PRIVILEGES ON mod6externe.* TO 'dev'@'localhost';
-- as dev
USE mod6externe;
-- création de notre table Pilote pour servir de support à nos vues
CREATE TABLE Pilote
(brevet CHAR(8), nom CHAR(10), nbHVol DECIMAL(7,2),
adresse CHAR(20), compa CHAR(8),
CONSTRAINT pk_pilote PRIMARY KEY(brevet));
-- insertion de données
INSERT INTO pilote VALUES ('PL-1', 'Soutou', 890, 'Castanet', 'CAST');
INSERT INTO pilote VALUES ('PL-2', 'Laroche', 500, 'Montauban', 'CAST');
INSERT INTO pilote VALUES ('PL-3', 'Lamothe', 1200, 'Ramonville', 'AF');
INSERT INTO pilote VALUES ('PL-4', 'Albaric', 500, 'Vieille-Toulouse', 'AF');
INSERT INTO pilote VALUES ('PL-5', 'Bidal', 120, 'Paris', 'ASO');
INSERT INTO pilote VALUES ('PL-6', 'Labat', 120, 'Pau', 'ASO');
INSERT INTO pilote VALUES ('PL-7', 'Tauzin', 100, 'Bas-Mauco', 'ASO');
```

Créer une vue

Avant de créer une vue, écrivez et testez votre requête de sélection. Une fois que votre requête est bonne vous n'avez en général plus qu'à placer `CREATE VIEW nomVue AS` devant et c'est terminé.

Créons 2 vues de la même table Pilote:

1. Créer la vue `PilotesAF` qui décrit uniquement les pilotes d'Air France
2. Créer la vue `EtatCivil` qui ne contient que les colonnes adresse et nom de chaque pilote.

```
-- 1
CREATE VIEW mod6externe.PilotesAF AS SELECT * FROM Pilote WHERE compa='AF';
-- 2
CREATE VIEW mod6externe.EtatCivil AS SELECT adresse, nom FROM Pilote;
```

Si vous listez les tables avec un SHOW TABLES du schéma mod6externe, vous verrez apparaître vos vues aux côtés des tables normales. Les vues se comportent comme les tables. Sous le capôt, elles ont un type VIEW différent des tables conventionnelles de type BASE TABLE.

A présent nous pouvons interroger nos vues *comme on le ferait avec n'importe quelle table*.

3. Somme des heures de vol des pilotes d'Air France.
4. Nombre de pilotes (à partir de la vue EtatCivil)

```
-- 3
SELECT SUM(nbHVol) FROM PilotesAF;
-- 4
SELECT COUNT(*) FROM EtatCivil;
```

Utiliser les alias

Utiliser les alias pour renommer les colonnes d'une vue, et masquer les noms des colonnes des tables sources (détails d'implémentation et conventions utilisées pour le schéma relationnel)

5. Créer la vue **PilotesAF** en renommant les colonnes:
 1. brevet en *codePil*
 2. nom en *nomPil*
 3. nbHVol en *heuresPil*
 4. adresse en *adressePil*
 5. compa en *societe*
6. Faire de même pour la vue **PilotesAF** précédemment, en utilisant la clause **CREATE OR REPLACE VIEW**.

```
--5
CREATE VIEW PilotesPasAF
(codePil, nomPil, heuresPil, adressePil, societe)
AS SELECT * FROM Pilote WHERE NOT (compa='AF');
-- 6
CREATE OR REPLACE VIEW PilotesAF
(codePil, nomPil, heuresPil, adressePil, societe)
AS SELECT * FROM Pilote WHERE (compa='AF');
```

Créer une vue complexe

Une vue complexe (ou multitable) n'est pas plus compliquée à faire qu'une vue simple. Elle cache seulement plus de complexité (jointure, groupements).

La possibilité de faire des mises à jour sur les tables sources directement depuis une vue complexe est plus délicate, mais faisable en prenant les bonnes précautions.

Créer les tables et insérer les données suivantes dans la base mod6externe

```

CREATE TABLE Compagnie
  (comp CHAR(4), nrue TINYINT(3),
  rue CHAR(20), ville CHAR(15) DEFAULT 'Paris', nomComp CHAR(15),
  CONSTRAINT pk_Compagnie PRIMARY KEY(comp));

CREATE TABLE Pilote IF NOT EXISTS
  (brevet CHAR(6), nom CHAR(15) NOT NULL, nbHVol DECIMAL(7,2), compa CHAR(4),
  CONSTRAINT pk_Pilote PRIMARY KEY(brevet),
  CONSTRAINT ck_nbHVol CHECK (nbHVol BETWEEN 0 AND 20000),
  CONSTRAINT un_nom UNIQUE (nom),
  CONSTRAINT fk_Pil_compa_Comp FOREIGN KEY(compa) REFERENCES
    Compagnie(comp));

INSERT INTO Compagnie VALUES ('SING', 7, 'Camparols', 'Singapour', 'Singapore
  AL');
INSERT INTO Compagnie VALUES ('AF', 124, 'Port Royal', DEFAULT, 'Air France');
INSERT INTO Pilote VALUES ('PL-8', 'Blanc', 1000, 'Rennes', 'SING')

```

7. Créer la vue `Pilotes_multi_AF` qui contient les informations des pilotes et de leur compagnie, uniquement pour les pilotes d’Air France.
8. Créer la vue `Stats_Heures_Pil` qui contient le nombre d’heures de vol *moyen* et *cumulé* des pilotes, *par compagnie*.

```

-- 7
CREATE VIEW Pilotes_multi_AF
AS
SELECT * FROM Pilote INNER JOIN Compagnie WHERE compa=comp;

-- 8
CREATE VIEW Stats_Heures_Pil AS SELECT compa, SUM(nbHVol),AVG(nbHVol) FROM
  Pilote INNER JOIN Compagnie WHERE compa=comp GROUP BY (compa)

```

Confidentialité

La confidentialité (et le partage restreint de données) est la mission première des vues.

On peut:

- masquer ou renommer des colonnes avec la requête de sélection à l’origine de la vue
 - ajouter des *contraintes temporelles* en lecture/écriture
9. Créer une vue `Public_Pilote_Infos` qui préserve l’adresse et le numéro de brevet des Pilotes de la compagnie Air France
 10. Créer une vue `Restricted_Pilote` accessible en lecture tous les jours sauf le [jour de la semaine actuel]. *Tip*: utiliser les fonctions SQL de manipulation de temps `DATE_FORMAT` et `SYSDATE`.

```

--9
CREATE VIEW Public_Pilote_Infos AS SELECT nom,nbHVol,compa FROM Pilote WHERE
  compa='AF';

```

```
--10  
CREATE VIEW Restricted_Pilote AS SELECT * FROM Public_Pilote_Infos WHERE  
    DATE_FORMAT(SYSDATE(), '%W') NOT IN ('Sunday');
```

*L'accès aux données d'un utilisateur doit idéalement se faire par le partage d'une vue. Ainsi, les privilèges ne sont jamais donnés directement sur les tables sources, mais sur les vues elles-même, avec les clauses **GRANT**.*

Références

- [CREATE PROCEDURE and CREATE FUNCTION Statements \(doc officielle MySQL v8\)](#)