Exercices les procédures stockées MySQL : Série 01

1. Objectifs

- Connaitre les procédures stockées de base de données MySQL

2. Exercice 01

- **Énoncé**
  - Considérez les relations suivantes :
    - **Employe**(nuempl:string, nomempl:string, hebdo: integer,salaire integer, #affect:integer)
    - **Service**(nuserv:integer, nomserv:string, #chef:integer)
    - **Projet**(nuproj:integer, nomproj:string, #resp:integer)
    - **Travail**(#nuempl:integer, #nuproj:integer, duree:integer)
  - Travail à faire
  - Pour simplifier l'exploration de la base de données, il vous est demandé d'implémenter les fonctions ou procédures suivantes :
    - **1-** Écrire une procédure qui prend comme arguments un identifiant d'employé et un nombre, puis mettre à jour le salaire de l'employé donné avec le nombre donné.
    - **2-** Écrire une fonction qui compte le nombre d'employés participant à un projet donné.
    - **3-** Écrire une fonction qui compte le nombre de projets supervisés par les employés d'un service donné.
    - **4-** Écrire une fonction qui compte le nombre de projets auxquels participe l'employé donné.
    - **5-** Écrire une fonction qui renvoie la chaîne 'Salaire faible' si le salaire de l'employé donné est supérieur à 2000\$ sinon retourner '**Bon salaire**'.
    - **6-** Écrire une fonction, qui compte le nombre d'employés qui prennent en charge plus que le nombre de projets donné.
    - **7-** Écrire une procédure qui insère l'employé donné dans une table de sauvegarde nommée '**ALERT_EMPLOYE**'.
- **Solution**

apcpedagogie.com

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

3. Exercice 02

- **Énoncé**

  - **1-** Créer la BD "**biblio**" à partir du script ci-contre.
  - **2-** Écrire une fonction qui calcule, pour un adhérent donné, le nombre de jours restant avant d'être en retard.
    - Si l'adhérent n'a pas d'emprunts en cours, on renvoie NULL.
    - Si l'adhérent est en retard, on renvoie un résultat négatif correspondant au nombre de jours de retard le plus grand pour ses emprunts en cours. Par exemple, s'il devait rendre un livre avant-hier et qu'il a un livre à rendre le lendemain, on renvoie « -2 » pour avant-hier.
    - Si l'adhérent n'est pas en retard, on renvoie un résultat positif correspondant au nombre de jours d'emprunt restant le plus petit pour ses emprunts en cours. Par exemple, s'il doit rendre un livre demain et un autre après-demain, on renvoie « +1 » pour demain. (Pour ces deux derniers cas, on prendra en compte la possibilité d'avoir des emprunts avec des durées Max différentes et des emprunts en cours avec des dates d'emprunt différentes).
  - **3-** Utiliser cette fonction pour afficher la situation de tous les adhérents.
  - **4-** Écrire une procédure qui permette de lister les emprunts d'un adhérent identifié par son numéro.
  - **5-** Écrire une procédure qui affiche les exemplaires disponibles d'un titre (on fera une version OUTER JOIN et une version NOT IN). Pour se faciliter la tâche, on a intérêt à d'abord traiter la question : "combien y a-t-il exemplaires disponibles du titre 'NARCISSE ET GOLDMUND'" avec les deux versions demandées, pour ensuite passer à l'écriture de la procédure stockée.
  - **6-** Écrire une procédure qui affiche les titres d'un auteur et le nombre d'exemplaires disponibles par titre. On testera avec LEWIS CAROLL, GILBERT HOTTOIS et kenneth white. Pour se faciliter la tâche, on a intérêt à commencer par traiter la question : « Les exemplaires dispo de Lewis Caroll » puis « Le nombre d'exemplaires dispo par titre de Lewis Caroll » pour enfin écrire la procédure stockée.
  - **7-** Ecrire une procédure qui permette d'enregistrer un emprunt.
  - **8-** Modifier la table des emprunts : mettez la valeur par défaut de la durée max à 14.

apcpedagogie.com

```
drop database if exists biblio;
CREATE DATABASE biblio;
use biblio;
drop table if exists livre;
create table livre (
 numinv varchar(4) not null ,
 titre varchar(20),
 auteur varchar(15),
 qte int(2) unsigned,
 primary key (numinv)
);
create index livreind on livre (numinv);
drop table if exists abonne;
create table abonne(
 numab varchar(3) not null ,
 nom varchar(15),
 prenom varchar(15),
 primary key (numab)
);
create index abonneind on abonne (numab);
drop table if exists pret;
create table pret (
 numinv varchar(4) not null ,
 numab varchar(3) not null ,
 datepret date,
 primary key (numinv,numab)
);
create index pretind on pret (numinv,numab);
```

- **Solution**
  - Essayez de faire l'exercice de votre côté avant de regarder la Solution !

Cours MySQL

Source : https://apcpedagogie.com/exercices-les-procedures-stockees-mysql-serie-01 · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com