

---

Département de Mathématiques et Informatique

# MI0A301T : BD Avancées

- 1 - Rappels sur les SGBD Relationnels
- 2 - Vues relationnelles
- 3 - Procédures et fonctions stockées (TD)
- 4 - Triggers

# Un peu de pratique (1)

Les exercices sont à réaliser en utilisant Adminer. Importez le fichier C5\_Enseignants\_Sous-programmes\_Stockes.sql

Routines		
Nom	Type	Type de retour
C301T_afficherNom	PROCEDURE	
C301T_renvoyerNomUE	FUNCTION	varchar(255)

## ➤ Exercice 1

- a. Retrouvez la **fonction stockée C301T\_renvoyerNomUE()** dans la liste des sous-programmes de votre base
  - Tester-la avec l'interface Adminer
  - Tester-la avec une requête SQL
- b. Retrouvez la **procédure stockée C301T\_afficherNom()** dans la liste des sous-programmes de votre base
  - Tester-la avec l'interface Adminer
  - Tester-la avec une requête SQL
- c. Appelez la **procédure stockée C301T\_afficherNom()** pour afficher le nom de l'UE (réfléchissez ... un peu ;) )
  - Appelez-la avec l'interface Adminer
  - Appelez-la avec une requête SQL

# Un peu de pratique (2)

---

## ➤ Exercice 2

- Créez la procédure *ajouterMatiere()* permettant d'ajouter une matière sans son coef (qui pourra être mémorisé plus tard)
- Appelez cette procédure stockée pour ajouter la matière « *Informatique* »
- Vérifiez que l'ajout c'est bien passé

Bien réfléchir aux paramètres nécessaires à la procédure.

## ➤ Exercice 3

- Créez la procédure *ajouterMatiereEnseignant()* prenant en paramètre un enseignant (son nom et son prénom) et le nom de la matière à lui affecter. L'enseignant et la matière sont supposés déjà exister dans la base.

Bien réfléchir aux opérations nécessaires avant d'ajouter le tuple.

- Appelez cette procédure stockée pour mémoriser qu'Alan Turing enseigne l'*informatique*.
- Vérifiez que l'ajout c'est bien passé (Alan Turing enseigne à présent 3 matières).

# Un peu de pratique (3)

---

## ➤ Exercice 4

- a. Créez une fonction stockée renvoyant l'identifiant du cours d'*Histoire*.
- b. Ecrivez une requête qui affiche la liste des enseignants (leur *id*) d'*Histoire* (cette requête fait appel à la fonction précédente).

## ➤ Exercice 5

- a. Créez une fonction stockée qui calcule et renvoie le nombre d'enseignants pour une matière dont le nom est passé en paramètre.
- b. Ecrivez une requête qui affiche le nom et le nombre d'enseignants des matières de la base (cette requête fait appel à la fonction créée).
- c. Ecrivez une requête qui affiche le nom et le nombre d'enseignants des matières de la base ayant au moins 2 enseignants (cette requête fait appel à la fonction créée)

# Un peu de pratique (4)

- Exercice 6
  - a. Modifiez la procédure *ajouterMatiereEnseignant()* pour que l'ajout ne se fasse que si l'enseignant et la matière sont effectivement bien présents dans la base. Si pb, avertir l'utilisateur (Cf. les indications de la diapo suivante).
- Syntaxe de l'instruction **SI ... ALORS ... SINON ...** en PL/SQL :

```
IF condition THEN
    [instructions]
ELSE
    [instructions]
END IF;
```

```
IF condition THEN
    [instructions]
END IF;
```

- Rappels :
  - **Initialiser** une variable avec la valeur NULL :  
`SET @nomVar = NULL ;`
  - **Comparer** une variable avec la valeur NULL (**ne pas utiliser** l'opérateur de comparaison `=`) :  
`@nomVar IS NULL/IS NOT NULL`

# Un peu de pratique (5)

## ➤ Exercice 6

- b. Appelez la procédure stockée pour mémoriser que (jeu de tests) :

- Alan Turing enseigne l'Algo (la matière n'existe pas → pas d'ajout)
- George Boole enseigne l'Informatique (l'enseignant n'existe pas → pas d'ajout)
- George Boole enseigne l'Algo (aucun des deux n'existe → pas d'ajout)
- Graham Bell enseigne les Maths (les deux existent → 1 tuple ajouté)

*Requête SQL*

```
CALL C301T_ajouterMatiereEnseignant('Turing','Alan','Algo')
```

@message  
La matière n'existe pas

1 ligne (0.002 s) Exporter: enregistrer CSV, Exporter

Requête exécutée avec succès, 0 ligne modifiée. (0.000 s)

---

```
CALL C301T_ajouterMatiereEnseignant('Boole','George','Informatique')
```

@message  
L'enseignant n'existe pas.

1 ligne (0.001 s) Exporter: enregistrer CSV, Exporter

Requête exécutée avec succès, 0 ligne modifiée. (0.000 s)

---

```
CALL C301T_ajouterMatiereEnseignant('Boole','George','Algo')
```

@message  
L'enseignant n'existe pas. La matière n'existe pas

1 ligne (0.002 s) Exporter: enregistrer CSV, Exporter

Requête exécutée avec succès, 0 ligne modifiée. (0.000 s)

---

```
CALL C301T_ajouterMatiereEnseignant('Bell','Graham','Maths')
```

@message  
Ajout effectué

1 ligne (0.002 s) Exporter: enregistrer CSV, Exporter

Requête exécutée avec succès, 1 ligne modifiée. (0.000 s)

# Travaux Dirigés

---

- Le cas UNESCO : TD3

# Sources

---

- <https://mariadb.com/kb/en/>
- <https://dev.mysql.com/doc/refman/8.0/en/>
- <https://sql.sh>
- <https://grafikart.fr/tutoriels/procedures-triggers-fonctions-593>