Views / vues

(Docker setup)

|  |  |
| --- | --- |
| Elément | Description |
| **Compétences** | Générer des informations pour les audits et le login. Définir et implémenter des évaluations et alarmes. |
| **Objectifs** | Apprendre ce que sont les vues, comment les créer et les manipuler |
| **Durée estimée** | 1x 45 min |
| **Répertoire de travail** | C:\Docker |
| **Fichiers sources** | E-183-ALL-Views.zip |
| **A produire** | Répondre aux questions directement dans ce document |
| **Moyens d’aide** |  |

# Informations générales

Le but de cet exercice est de vous apprendre ce qu’est une vue (view) dans une base de données. La syntaxe est propre à MySQL, mais les concepts s’appliquent aussi à d’autres SGBD.

# MLD

Vous trouvez ci-dessous le MLD de la base de données db\_students :

A screenshot of a computer

Description automatically generated

# Echauffement

A faire avec la console de MySQL présente dans PHPMyAdmin.

## Sélectionnez le nom et le prénom de tous les étudiants

La requête qui vous permet de réaliser cette opération :

select stuName, stuFirstName from t\_student;

## Sélectionnez le nom et le prénom de tous les étudiants, ainsi que le nom du cours suivi, la date et la note obtenue

La requête qui vous permet de réaliser cette opération :

SELECT stuName, stuFirstName, courName, evaGrade, evaDate

FROM t\_student

JOIN t\_course

ON t\_student.idStudent = t\_course.idCourse

JOIN t\_evaluation

ON t\_student.idStudent = t\_evaluation.idEvaluation;

# Création des vues

## Les étudiants et leurs notes

Modifiez le fichier **2\_create\_views.sql**, de telle sorte à ce que :

* Une vue v\_studentsGrades soit présente.
* Cette vue affiche le nom et le prénom de l’étudiant, le cours évalué, la date d’évaluation et la note obtenue.

CREATE VIEW v\_studentsGrades AS

SELECT stuName, stuFirstName, courName, evaGrade, evaDate

FROM t\_student

JOIN t\_course

ON t\_student.idStudent = t\_course.idCourse

JOIN t\_evaluation

ON t\_student.idStudent = t\_evaluation.idEvaluation;

Modifiez le fichier 3\_create\_users pour ajouter un utilisateur « teacher » qui ne puisse que consulter la vue v\_studentsGrades.

Modifiez le fichier docker-compose.yml pour que le script 2\_create\_views.sql soit automatiquement appelé à la création.

## Les étudiants en difficulté et leurs notes

Modifiez le fichier **2\_create\_views.sql**, de telle sorte à ce que :

* Une vue v\_studentsBadGrades soit présente.
* Cette vue affiche le nom et le prénom de l’étudiant, le cours évalué, la date d’évaluation et la note obtenue à condition que la note soit inférieure à 4.0.

Modifiez le fichier 3\_create\_users pour ajouter un utilisateur « teacher » qui aussi consulter la vue v\_studentsBadGrades.

2. Testez votre vue en vous connectant avec le compte teacher et sélectionner des élèves qui ont eu mauvaises notes au module 123.

## Les étudiants et leurs absences

1. Modifiez le fichier **create\_views.sql**, de telle sorte à ce que :

* Une vue v\_absentStudents soit présente.
* Cette vue affiche le nom et le prénom de l’étudiant, les jours et les périodes d’absences, ainsi que leur motif.

Modifiez le fichier create\_users pour ajouter un utilisateur « teacher » qui aussi consulter la vue v\_absentStudents.

2. Testez votre vue en vous connectant avec le compte teacher et sélectionner des élèves qui ont eu des absences au mois de février 2024.

Indices :

* Utilisez MONTH(*une\_certaine\_date)* pour obtenir le numéro du mois de cette certaine date.
* Utilisez YEAR(*une\_certaine\_date)* pour obtenir l’année de cette certaine date.

# Retour des scripts

Vous rendrez le devoir teams **E-183-ALL-Views** en donnant vos scripts SQL :