

## OBJECTIFS ET CONTRAINTES

**Objectif** : Ecrire des requêtes SQL d'interrogation de données qui répondent aux informations demandées.

**Ressources** : Vous disposez du diagramme de classes UML et du modèle physique de la base sur laquelle vous devez effectuer des requêtes.

**Contraintes** : vous devez utiliser les fonctions postgres quand celles-ci s'imposent, notamment les fonctions de traitement de date.

## BASE UTILISEE : GESTION DU SUIVI DES TP SQL

1. Lancez la console d'administration du SGBD PostGres  
<http://193.49.174.32/phppgadmin/>  
*Mêmes paramètres que ceux utilisés pour vous connecter au réseau de l'école*
2. Récupérez le script de création des tables (**suivi\_etud.sql**)
3. Exécuter le script
4. Vos tables sont "normalement" créées et renseignées dans le schéma de données TP3\_bd\_suivi

### Quelques explications sur la Base de données utilisée :

Un enseignant de BD a conçu une base de données pour effectuer le suivi des étudiants lors des TP de sa matière.

La relation QUESTION répertorie toutes les questions de tous les TP avec leur niveau de difficulté (Facile, Normal, Difficile, Complexe), le temps estimé pour répondre à la question, le nombre de variantes demandées et le nombre de points attribués à la question.

Les thèmes abordés lors des TP sont répertoriés dans la relation THEME où le Libelle pourra prendre comme valeur : 'Jointure prédicative', 'Jointure imbriquée', 'Auto-jointure', 'Jointure externe', 'Partitionnement'...

Bien sûr chaque question de TP correspond à un ou plusieurs thèmes. Par exemple, la question d'identifiant 30 porte sur les thèmes : 'Jointure imbriquée' et 'Fonction agrégative'. La relation THEMQUEST est introduite pour stocker les liens entre QUESTION et THEME.

Les informations relatives aux étudiants sont stockées dans la relation ETUDIANT.

Enfin, l'évaluation des étudiants aux différentes questions est caractérisée par le résultat ('Faux', 'Erreur minime', 'Juste mais non optimal', 'Juste'), le temps mis par l'étudiant pour répondre, le nombre de variantes réalisées et le nombre de points obtenus, ces informations sont stockées dans la relation EVALUATION

Le schéma relationnel de la base est donc le suivant :

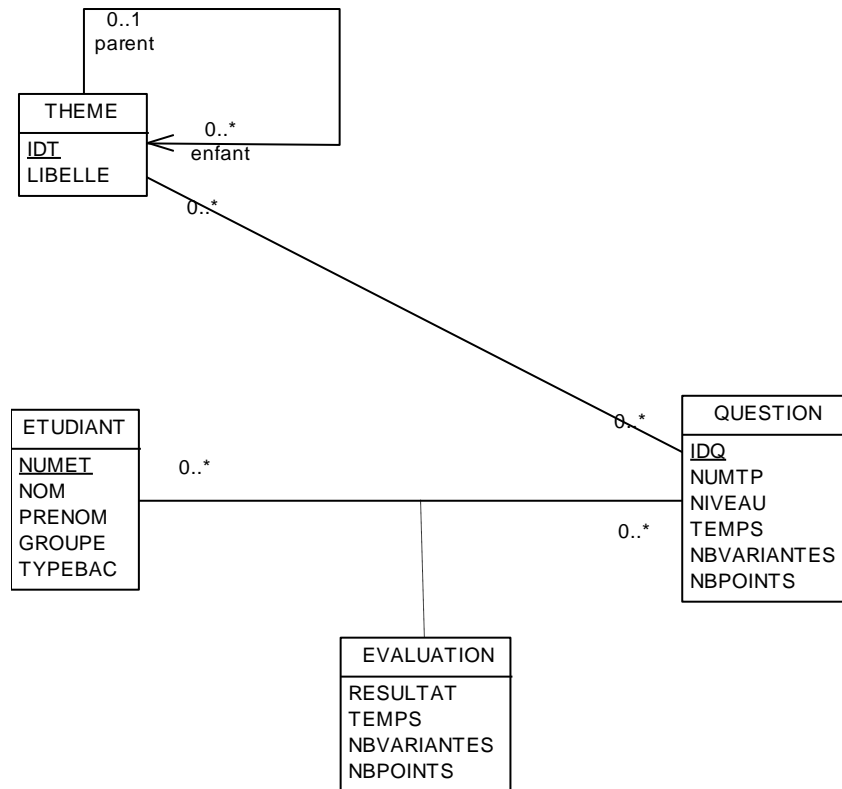
QUESTION	( <u>IdQ</u> , NumTP, Niveau, Temps, NbVariantes, NbPoints)
THEME	( <u>IdT</u> , Libelle, IdTPere)
THEMQUEST	(# <u>IdQ</u> , # <u>IdT</u> )
ETUDIANT	( <u>NumEt</u> , Nom, Prenom, TypeBAC, Groupe)
EVALUATION	(# <u>IdQ</u> , # <u>NumEt</u> , Resultat, Temps, NbVariantes, NbPoints)

**TOUTES LES DONNEES DE TYPES CHAR OU VARCHAR SONT RENSEIGNEES EN MAJUSCULE DANS LA BDD**

### TRAVAIL A FAIRE : ECRIRE ET EXECUTER LES REQUETES SUIVANTES

- 1.1. Donnez les feuilles de la hiérarchie (les nœuds qui ne sont pas racine) des thèmes (identifiant, libellé). **PAS DE RECURSIVITE**
- 1.2. Donnez toute la hiérarchie (au-dessus de lui) (identifiant, libellé) de classification du thème JOINTURE EXTERNE (Triez cette liste du thème le plus général au thème le plus spécifique).
- 1.3. Donnez les racines de la hiérarchie des thèmes (identifiant, libellé).ceux qui n'ont pas de père **PAS DE RECURSIVITE**
- 1.4. Quels sont les sous-thèmes (identifiant, libellé) spécialisant (i.e. plus précis) le thème LANGAGE DE REQUETES ? Toute la hiérarchie descendante et ensuite que les fils directs.
- 1.5. Quels sont les sous-thèmes (identifiant, libellé) rattachés directement ou pas au thème JOINTURE à l'exception du thème JOINTURE IMBRIQUEE et de ses sous-thèmes ?
- 1.6. Quels sont les sous-thèmes (identifiant, libellé) rattachés directement ou pas au thème JOINTURE à l'exception du thème JOINTURE IMBRIQUEE ?
- 1.7. Donnez la liste des questions (identifiant, numéro de TP) se rapportant à un sous-thème rattaché directement ou non au thème JOINTURE.
- 1.8. Quels sont les thèmes (identifiant, libellé) généralisant (ascendant) les thèmes JOINTURE OU SELECTION SIMPLE ?
- 1.9. Quels sont les thèmes (identifiant, libellé) généralisant (ascendant) les thèmes JOINTURE **ET** SELECTION SIMPLE ?
- 1.10. Donnez les nom et prénom des étudiants ayant fait toutes les questions du TP 1. **PAS DE RECURSIVITE**
- 1.11. Pour chaque groupe, donner le nombre d'étudiants ayant un BAC S et le nombre d'étudiants ayant un BAC STI. **PAS DE RECURSIVITE**
- 1.12. Quels sont les groupes dans lesquels tous les types de BAC sont représentés ? **PAS DE RECURSIVITE**
- 1.13. Quels sont les groupes pour lesquels au moins un étudiant a été évalué pour chaque TP. Donner deux formulations. **PAS DE RECURSIVITE**
- 1.14. Quels sont les sous-thèmes (identifiant, libellé) rattachés directement ou pas au thème SQL LMD et qui ne sont utilisés par aucune question ?
- 1.15. Donnez par étudiant (numéro, nom) le nombre de questions du TP 1 et du TP2 réalisées. Formulez la requête avec imbrication(s) dans le FROM. **PAS DE RECURSIVITE**
- 1.16. Affichez la moyenne des notes obtenues aux évaluations par TP (N°TP), ensuite par nom et prénom des étudiants (nom,prénom) et ensuite par N° de question (idq), vous réaliserez ces statistiques en une seule requête . **PAS DE RECURSIVITE**
- 1.17. Affichez la moyenne des notes obtenues aux évaluations par TP, nom, prénom et N° de question, ensuite par TP, nom, prénom et ensuite par TP ; vous réaliserez ces statistiques en une seule requête. **PAS DE RECURSIVITE**
- 1.18. Affichez la note totale obtenue par les étudiants dans chaque TP et le nombre de points max par TP. **PAS DE RECURSIVITE**
- 1.19. Afficher les notes obtenues par les étudiants dans chaque TP ainsi que le rang de leur note par TP. **PAS DE RECURSIVITE**

## DIAGRAMME DE CLASSES DE LA BASE SUIVI DES TPS SQL



## MODELE PHISYQUE DE LA BASE SUIVI DES TP<sub>s</sub> SQL

