

Conception et implémentation d'une base de données pour la gestion des programmes et étudiants au sein des universités du Québec

Présentation générale du laboratoire

L'objectif principal de ce premier laboratoire, d'une durée de 5 semaines, est la création des modèles conceptuel, logique et physique d'une base de données en fonction d'un problème connu, soit celui de « *La gestion des programmes et étudiants au sein des universités du Québec* ».

Ce laboratoire est à réaliser en équipe (de 3 personnes). Vous devez constituer votre équipe sous Moodle.

Dans les sections suivantes, vous trouverez l'énoncé du problème ainsi que les règles d'affaires que vos différents modèles doivent être capables de supporter.

Ce premier laboratoire est organisé en trois parties. La première partie consiste en la compréhension du problème et des règles d'affaires liées, puis en l'élaboration du modèle conceptuel de données correspondant. Durant la seconde partie, vous devez créer le modèle logique de données lié au modèle conceptuel de données. La troisième partie est dédiée à la création du modèle physique de données. De plus, la troisième partie vous permettra de mettre à l'épreuve vos différentes conceptions via l'insertion de données dans la base de données en créant un jeu de données de test. Ces insertions permettront de valider que votre conception respecte les différentes règles d'affaires et contraintes.

Les deux premières parties doivent être remises sur Moodle et seront corrigées. Vous recevrez des éléments de correction succincts afin que vous puissiez aborder la partie suivante sur de bonnes bases.

Énoncé du problème

Contexte

Les étudiants qui suivent des cours au sein des universités du Québec (UQAM, UQAR, UQAC, UQO, ETS, ...) sont enregistrés dans une base de données commune à toutes les universités.

Dans cette base de données, nombre d'informations de différents types sont stockées afin de suivre les étudiants tout au long de leurs programmes universitaires.

Chaque étudiant est identifié de manière unique par son code permanent. De plus, la base de données contient les informations suivantes pour les étudiants : nom, prénom, date de naissance, programmes suivis, cours suivis à chaque année-session (par exemple, hiver 2016) et la note obtenue pour chaque cours suivi.

Un cours est identifié de manière unique par la combinaison d'un département et un numéro (par exemple, TCH010). Les cours ont un ou plusieurs groupes durant une session (par exemple, il y a les groupes 01, 02, 03 et 04 pour le cours TCH054 à la session d'hiver 2019). Les départements (ex. : génie logiciel et des technologies de l'information, génie de la production automatisée, génie mécanique) et les programmes (ex. : baccalauréat en génie logiciel, DESS en gestion de projets d'ingénierie, maîtrise en génie des technologies de l'information), sont identifiés par leurs noms et un nombre automatiquement incrémenté (1, 2, 3, etc.). De plus, les programmes ont un nombre de crédits pour la graduation (par exemple, 45 crédits pour la maîtrise en génie logiciel). Pour les cours, la base de données contient aussi le nom du cours (par exemple, Bases de données), le département auquel le cours appartient et les programmes pour lesquels ce cours peut être pris et les éventuels pré requis pour ce cours. Finalement, pour les cours, la base de données contient aussi le nom de l'enseignant qui donne le groupe-cours pour chaque session, la salle, le nombre d'heures de cours et l'heure du cours pour chaque session où le cours est offert.

Les enseignants sont identifiés de manière unique par leur numéro d'assurance sociale (NAS) et la base de données contient aussi leurs noms, prénoms, date de naissance, le département d'attache ainsi que les différentes spécialités de l'enseignant (ex. : informatique, communication). Il existe trois types d'enseignants : professeur, maître d'enseignement et chargé de cours. Un groupe-cours peut être enseigné que par un professeur, un maître d'enseignement ou un chargé de cours.

Pour résumer, les universités ont des départements qui offrent des programmes. Un étudiant valide un programme (et obtient donc un diplôme) en suivant des cours offerts par un département et rapportant des crédits pour un programme jusqu'à l'obtention des crédits requis par le programme. Chaque groupe-cours est enseigné par un enseignant (professeur, maître d'enseignement, chargé de cours) dans une salle, à une heure, à une session et une année donnée.

Règles d'affaire

Votre modélisation de base de données doit être capable de supporter les règles d'affaires suivantes :

- RA-1 : Taux de complétion d'un programme. Pour un étudiant, nous devons être capable de retrouver le taux de complétion (en %) d'un programme en fonction du nombre de cours validés (note de D ou plus) et du nombre de crédits requis pour valider le programme dans lequel l'étudiant est inscrit.
- RA-2 : Heures de cours hebdomadaire. Pour les étudiants et les enseignants, nous devons être capables de retrouver le nombre d'heures de cours effectuées par semaine pour chaque année-session.
- RA-3 : Cours avec moins de 5 inscrits. Pour chaque session, nous devons être capables de savoir quels sont les cours-groupes qui ont moins de 5 étudiants inscrits (dans le but de pouvoir les supprimer, éventuellement).
- RA-4 : Nouveau cours. En considérant une année-session et une spécialité (ex. : informatique, communication), nous devons être capables de trouver un enseignant qualifié disponible et une salle à une heure pour donner un cours.
- RA-5 : Moyennes. Nous devons être capables de calculer la moyenne des notes obtenues par un étudiant par année-session (ex : Hiver 2018) et totale (pour toutes les sessions).
- RA-6 : Taux d'occupation de l'école. En considérant que chaque salle peut être occupée de 9h00 à 12:30 puis de 13:30 à 17h00 et enfin de 18:00 à 21h30, il s'agit de calculer le taux d'occupation d'une des universités. Si toutes les salles sont occupées sur toutes les plages horaires, alors le taux d'occupation est de 100%.
- RA-7 : Performance d'un enseignant. Nous devons être en mesure de déterminer la moyenne des notes obtenues par les étudiants d'un enseignant donné, pour un cours donné.

Contraintes

En plus des contraintes d'intégrités référentielles, déductibles de votre modèle relationnel, votre base de données devra faire respecter les contraintes ci-dessous. Pour chacune de ces contraintes, vous devez produire un jeu de données démontrant que la contrainte est bien respectée (partie 3).

- C1. Le sigle d'un cours doit avoir exactement 6 caractères.
- C2. Un cours doit avoir au minimum 1 crédit et au maximum 7 crédits;
- C3. Lors de la suppression d'un étudiant, toutes ses inscriptions doivent être supprimées;
- C4. Vous devez vous assurer que chaque enseignant est soit un chargé de cours, un maître d'enseignement ou un professeur;
- C5. Lors de la suppression d'un département ou d'un programme, les tables qui y font référence ne seront plus associées à aucun département ou programme (nul);
- C6. La note obtenue par un étudiant à un cours doit avoir une valeur entre 0 et 100 (pour ceux qui utilisent une note numérique) ou une valeur parmi A+,A, A-, B+,B, ..., D, E.
- C7. Le nom d'un département ne peut-avoir moins de deux caractères.
- C8. Le nom d'un programme ne peut-avoir moins de deux caractères.
- C9. Le code permanent d'un étudiant doit obligatoirement comporter 12 caractères.

Les contraintes ci-dessous nécessiteront l'utilisation de déclencheurs:

- C10. Le code permanent d'un étudiant n'est pas modifiable;
- C11. Lors de la suppression d'un enseignant, les groupes cours qu'il donne doivent être attribués à un enseignant "Inconnu" (i.e. créer une entrée pour un enseignant dont le nom et le prénom seraient "Inconnu" dans votre table Enseignant);
- C12. Toutes les identités numériques artificielles doivent être gérées automatiquement par le SGBD. Vous devez utiliser au moins un déclencheur pour gérer l'identité d'une de vos tables.
- C13. Si le nom d'un étudiant ou d'un enseignant fait moins que deux caractères, il faut lui ajouter la mention "--Incomplet". Par exemple, si on ajoute ou on modifie le professeur Jean Lapierre en mettant seulement "L" comme nom, le système doit transformer le nom en "L--Incomplet"
- C14. Dans une session, la saison peut-avoir l'une des valeurs suivantes: HIVER, AUTOMNE ou ÉTÉ. Si le nom de la saison n'est pas l'une de ces valeurs, la base de données doit permettre la modification et changer le nom en "???". La valeur du nom de la saison doit toujours
Note: Utiliser l'opérateur NOT IN et la fonction UPPER
- C15. Si la clé primaire d'un cours change, toutes les tables y faisant référence doivent être mises à jour (**NOTE:** Ceci est demandé dans un but pédagogique uniquement. La valeur d'une clé primaire ne devrait jamais changer! Vous ne devez pas tester cette contrainte)

Partie I – Modèle conceptuel de données

La réalisation du modèle conceptuel de données est la première partie de ce laboratoire. Il est impératif que le contexte du problème ainsi que les différentes règles d'affaires soient pleinement compris par l'étudiant avant de se lancer dans la conception. Une conception basée sur une compréhension partielle du problème sera forcément incomplète (au mieux) ou incorrecte (au pire). Vous pouvez demander des précisions sur le contexte ou sur les règles d'affaires à vos chargés de laboratoires.

Objectifs

L'objectif de cette première partie est la réalisation du modèle conceptuel de données permettant de répondre aux règles d'affaires énoncées dans la partie précédente.

Cette partie doit être complétée durant la première semaine.

Tâches à réaliser

- Étude et compréhension du problème.
- Étude et compréhension des règles d'affaires.
- Réalisation du modèle conceptuel de données en utilisant le **formalisme du modèle de classes UML**.

Livrable

Il y a un livrable à remettre pour cette partie. Vous devez remettre votre modèle en format image sur Moodle **au plus tard le mercredi 23 janvier 2019 à 23h59**.

Partie II – Modèle logique de données (schéma relationnel)

La seconde partie de ce laboratoire est dédiée à l'élaboration du modèle logique de données lié à votre modèle conceptuel de données de la première partie.

Objectifs

- Réalisation du modèle logique de données.
- Comme la première partie, cette partie doit être réalisée en une semaine.

Tâches à réaliser

- Correction du modèle conceptuel de données de la partie I.
- Réalisation du modèle logique de données basé sur le modèle conceptuel de données *corrigé* en utilisant le formalisme Entité-Associations.

Livrable

Comme pour la première partie, il y a un livrable à remettre. Vous devez remettre votre modèle logique en format image sur Moodle **au plus tard le mercredi 30 janvier 2019 à 23h59**.

Partie III – Modèle physique de données

La troisième partie est dédiée à l'implémentation du modèle physique de données sur la base de votre modèle logique *corrigé*.

Objectifs

- Implémentation du modèle physique de données.
- Cette partie doit être réalisée en trois semaines.

Tâches à réaliser

- Correction du modèle logique de données de la partie II.
- Création d'un script SQL en utilisant le langage de définition de données (LDD SQL) pour l'implémentation du modèle physique de données basé sur le modèle logique de données *corrigé*. Votre script doit mettre en place les contraintes nécessaires pour maintenir l'intégrité référentielle et sémantique des données.
- Création d'un script SQL en utilisant le langage de manipulation de données (LMD SQL) pour l'insertion d'universités, programmes, cours, étudiants et professeurs afin de valider que votre modèle physique est capable de gérer l'ensemble des règles d'affaires et les contraintes.

Livrable et remise finale

Vous devez remettre l'intégralité de votre projet pour la remise finale. Vous devez remettre un fichier zip comportant :

- Le modèle conceptuel corrigé ;
- Le modèle logique corrigé ; et
- Le code source des deux scripts au format SQL (script de création de la base + script de jeu d'essai).

La remise doit être effectuée sur Moodle au **plus tard le MARDI (19) 26 février 2019 à 23h59**. En respect des règles du Service des Enseignements Généraux, toute remise tardive se verra attribuer la note de 0 (à moins d'entente préalable avec l'enseignant-e).

Barème de notation :

- Modèle conceptuel : 30%
- Modèle logique : 20%
- Modèle physique : 50%
 - Script de création de la base et contraintes : 35%
 - Jeu de données de test : 15%

Bon travail ☺