



Université du Québec
à Chicoutimi

Département d'informatique et de mathématique

Projet final

GESTION AIRE DE STATIONNEMENT



Introduction aux bases de données

8TRD151

Hiver 2024

Local : P3-1020

Date de début : lundi 08 avril 2024

Date de remise : mercredi 24 avril 2024 fin de journée

Travail à réaliser en équipe de 4 étudiants au maximum

Chargé de cours : Emmanuel Francis Constant | efconsta@uqac.ca

Travail à réaliser

Dans ce projet final, vous allez exploiter le SQL et le SQL étendu afin de développer, côté serveur, quelques fonctionnalités de l'outil de gestion d'espaces de stationnement à l'Université du Québec (UQ). L'objectif est de concevoir des routines stockées qui permettront de valider, présenter, mettre à jour et supprimer les données au sein de la base de données. Ce système vise à optimiser l'utilisation des zones de stationnement de l'université en facilitant la gestion des places disponibles, l'assignation des places aux étudiants, ainsi que le suivi des informations relatives au stationnement.

Une nouvelle structure de la base de données vous est fournie, assurant ainsi que toutes les équipes travaillent à partir du même modèle. **Il est strictement interdit de modifier cette structure**; elle a été conçue pour être directement prête à intégrer vos routines stockées.

1- Espace d'administration de l'aire de stationnement

25 points

- 1.1- A partir de la structure de la base de données fournie, implémenter une procédure stockée permettant de configurer dans le système une nouvelle aire de stationnement. La procédure stockée requiert en paramètres toutes les données nécessaires pour créer une université, excepté son identifiant unique qui doit être généré automatiquement par MySQL. Une fois invoquée, la procédure permettra de créer :
 - a. une nouvelle université.
 - b. un nouvel espace de stationnement rattaché à l'université nouvellement créée.
 - c. trois (3) allées reliées au nouvel espace de stationnement. Chaque allée doit avoir un sens de circulation différent, doit compter par défaut 10 places disponibles et doit avoir un tarif horaire de \$4,5 CAN. La création des allées doit être placée dans une boucle.
 - d. dix (10) places dans chaque allée nouvellement créée, dont deux réservées aux handicapés.
- 1.2- Configurer un déclencheur sur la table « espace_stationnement » pour archiver dans une table « log_aire_stationnement » toutes les tentatives, échouées ou réussies, de création d'espace de stationnement. Les données à recueillir sont les suivantes : nom et sigle de l'université, date et heure de la tentative. La création des places doit être placée dans une boucle.

Consignes

- ✓ La procédure doit valider les données fournies en paramètre. Par exemple, les données obligatoires, le format des données, la vérification des plages de valeurs autorisées, la vérification des valeurs uniques, et toute autre validation jugée utile par vous, développeurs SQL.

- ✓ Des messages d'erreur appropriés doivent être prévus pour les échecs de validation.
- ✓ La procédure doit vérifier l'existence de l'université dans la base de données, au besoin afficher un message d'erreur approprié informant des problèmes rencontrés.
- ✓ Des boucles doivent être utilisées pour la création des espaces de stationnement et des allées.

2- Gestion des étudiants

25 points

Pour la gestion des étudiants, il revient d'implémenter un ensemble complet d'opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) à travers des procédures stockées, afin de :

- 2.1- créer un nouvel étudiant** en fournissant en paramètre toutes les informations nécessaires pour un étudiant. A l'exception de l'identifiant unique de la table qui doit être généré dans le corps de la procédure par une fonction dont les spécifications sont décrites plus bas dans la section « Utilitaires ».
- 2.2- afficher les informations personnelles d'un étudiant** à partir de son identifiant unique fourni en paramètre.
- 2.3- mettre à jour les informations personnelles d'un étudiant** à partir de son identifiant unique fourni en paramètre.
 - Au moment de la mise jour, activer un trigger qui sauvegarde dans une table « historique_etudiant » les anciennes valeurs des colonnes ainsi que la date et l'heure de la modification.
- 2.4- supprimer un étudiant** à partir de son identifiant unique fourni en paramètre. La suppression consiste à passer la valeur de la colonne « supprime » de 0 à 1.

Consignes

- ✓ La procédure doit valider les données fournies en paramètres. Par exemple, les données obligatoires, le format des données, la vérification des plages de valeurs autorisées, la vérification des valeurs uniques et toute autre validation jugée utile par vous, développeurs SQL.
- ✓ Des messages d'erreur appropriés doivent être prévus pour les échecs de validation.
- ✓ Le courriel et le numéro de téléphone de l'étudiant doivent être validés à l'aide des expressions régulières. *Piste* : utilisation de **REGEXP**.
- ✓ La procédure doit vérifier l'existence de l'étudiant dans la base de données, au besoin afficher un message d'erreur approprié informant des problèmes rencontrés.

3- Espace de réservation

20 points

La réservation de place de stationnement sera assurée par une procédure stockée qui reçoit en paramètre l'identifiant unique de l'étudiant, la date et l'heure d'arrivée, la date et l'heure de départ. La procédure se charge de trouver une place disponible dans l'un des espaces de stationnement de l'université de l'étudiant.

Consignes

- ✓ La procédure doit valider les données fournies en paramètres. Par exemple, les données obligatoires, le format des données date, la vérification des valeurs uniques et toute autre validation jugée utile par vous, développeurs SQL.
- ✓ Vérifier que l'étudiant est inscrit à un cours qui se tient durant les heures de stationnement réservées.
 - Si l'étudiant n'a pas de cours dans la tranche horaire souhaitée, créer un enregistrement de violation de stationnement dans une table d'audit « violation_stationnement ». Les données à récupérer sont les suivantes : code permanent, nom et prénom de l'étudiant, numéro de la plaque du véhicule de l'étudiant, date et heure de la tentative de réservation.
- ✓ En l'absence de places disponibles, un message d'erreur approprié doit être retourné.
- ✓ En cas de succès de la création de la réservation :
 - un déclencheur (trigger) actualise le nombre de places disponibles dans l'allée de la place réservée;
 - la procédure affiche des détails pertinents concernant la réservation : la désignation de l'espace de stationnement, la désignation et le sens de la circulation de l'allée, le numéro et le type de la place, le montant à payer, les dates et heures d'arrivée et de départ.

4- Utilitaires

15 points

- 1- 1- Créer une fonction qui génère un identifiant unique pour la table étudiant. Cet ID doit être alphanumérique suivant le format « **ETU-999999** ».
 - « **ETU** » : un préfixe fixe de trois lettres, une abréviation de « étudiant ».
 - « - » : un tiret séparateur pour distinguer le préfixe du numéro unique qui suit.
 - « **999999** » : un numéro unique composé de six chiffres pour générer jusqu'à 1 million de codes différents (de 000001 à 999999).
- 2- Créer une vue qui retourne des informations sur toutes les aires de stationnement configurées dans la base de données: nom de l'université, désignation de l'espace de stationnement, désignation de l'allée, nombre de places disponibles, nombre de places réservées.

3- Mettre en place un événement MySQL (EVENT) qui s'auto exécute chaque 5 minutes afin de mettre à jour la disponibilité des places réservées dont les dates et heures de départ sont dépassées.

Piste : utilisation de “ `CREATE EVENT nom_evenement ON SCHEDULE EVERY 5 MINUTE ...` ”

5- Reporting

10 points

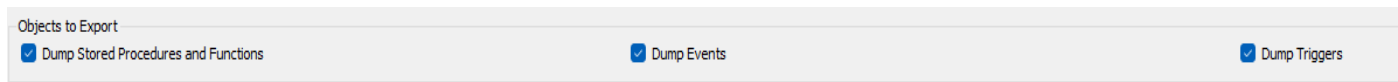
En utilisant une procédure stockée qui implémente un curseur, mettre en place un rapport donnant des données statistiques sur toutes les aires de stationnement déjà configurées dans la base de données :

Nom de l'université | nombre d'étudiants | nombre d'espace de stationnement | nombre d'agent de surveillance | nombre d'allées | nombre de places | nombre de places pour handicapés | nombre de places disponibles | nombre de places réservées | nombre moyen de réservation en 2023 | date ayant eu le plus de réservation | date ayant eu le moins de réservation.

6- Fichier de sauvegarde de la base de données

2,5 points

Réaliser une sauvegarde complète de la base de données. Veuillez vérifier que les options permettant la sauvegarde des procédures stockées, des fonctions, des événements et des déclencheurs sont bien sélectionnées. → [Aide pour la sauvegarde complète](#)



Objects to Export

☒ Dump Stored Procedures and Functions ☒ Dump Events ☒ Dump Triggers

7- Documentation

2,5 points

Rédiger un document succinct, n'excédant pas une page et demie, pour expliquer votre approche, les défis surmontés et les enseignements tirés. Il est essentiel que ce document mentionne les noms et les codes permanents de tous les membres de l'équipe.

Livrables

- Cinq fichiers de scripts SQL correspondant aux points 1 à 5, soit un fichier SQL par point.
- Un fichier PDF qui contient la documentation.
- Un fichier SQL contenant la sauvegarde complète de la base de données.

Schéma de la base de données

