

TP2

DDL, DML et contraintes

Le TP est à rendre dans un fichier TP2-Nom1-Nom2.sql sur l'espace "**campus**". N'oubliez pas d'indiquer vos noms et prénoms dans le fichier.

Ils faut précéder les commentaires dans ce fichier par un double tiret"--" pour éviter qu'ils soient interprétés pendant l'exécution.

Gestion du parc de vélos Velib

La ville de Paris a décidé d'informatiser la gestion de son parc de vélos Velib. A cet effet, elle utilise une base de données relationnelle « Gestion Parc ». La base de données contient des informations sur les vélos, les utilisateurs et les bornes de location:

- **Velo(noVelo, etatVelo)** : identifiant et état du vélo (excellent, à plat, ...).
- **Borne(noBorne, adrBorne, placesTot, placesLibres)** : numéro et ville de la borne avec le nombre total de places de vélo et le nombre de places libres.
- **Utilisateurs(noUtil, nom, adrUtil)** : numéro, nom et adresse de l'utilisateur.
- **Location(#noUtil, #noVelo, #noBorne, dateLoc, noBorneRet, dateRet)** : enregistrement d'une location.

On considère que tous les attributs sont renseignés (NOT NULL), sauf les attributs **noBorneRet** et **dateRet**, qui sont NULL si le vélo est en cours de location.

1. Création de la base de données « Gestion Parc »

Tables et contraintes

On reprend le schéma relationnel de départ

- **Velo(noVelo, etatVelo)**
- **Borne(noBorne, adrBorne, placesTot, placesLibres)**
- **Utilisateurs(noUtil, nom, adrUtil)**
- **Location(#noUtil, #noVelo, #noBorne, dateLoc, #noBorneRet, dateRet)**

Et on suppose que

- les numéros noX sont des chaînes de deux caractères,
- les attributs dateX sont des dates,
- les adresses adrX sont des chaînes de 15 caractères,
- les noms sont des chaînes de 10 caractères,
- l'état d'un vélo est une valeur parmi 'excellent', 'à plat', 'pas de lumière' et 'en panne',
- le nombre de places (libre et total) est un entier inférieur à 50,
- le nombre de places libres doit être inférieur au nombre de places total,
- pour une location, la date de retour doit être supérieure ou égale à la date de location,
- tous les attributs sont renseignés sauf dateRet et noBorneRet,

1.1. Donner les instructions CREATE TABLE qui créent les schémas de relation pour toutes les tables avec les contraintes de clés primaires, de clés étrangères et de domaine. Faire attention à l'ordre de création des tables.

1.2. Insérer les données dans la base de données « Gestion Parc ». Utiliser le fichier parc_velo.sql en exécutant la commande **@parc_velo.sql**

2. Manipulations de la base de données « Gestion Parc »

2.1. Afficher la structure de la table bornes ainsi que son contenu.

2.2. Modifier la structure de la table bornes :

Supprimer la colonne adrBornes.

Ajouter les colonnes :

CodePostal de type numérique sur 5 positions, et Ville de type caractère sur 15 positions.

2.3. Mettre à jour les adresses des bornes de N° b1, b2, b3, b4 et b5 (respectivement) avec les données suivantes:

75001, 'Paris 1',
75001, 'PARIS I',
75002, 'Paris 2eme',
75003, 'paris 3',
75003, 'Paris III'

2.4. Créer une nouvelle table AGGLOMERATION dont le schéma est le suivant :

AGGLOMERATION(CP, VILLE)

Définir la contrainte « Le nom de la ville devra être en MAJUSCULE ».

2.5. Remplir cette nouvelle la table AGGLOMERATION avec des données correctes :

Attention : Paris 1 est une chaîne de caractères différente de PARIS I

CodePostal	Ville
75001	PARIS I
75002	PARIS II
75003	PARIS III
75004	paris IV; Remarque : cette ligne ne doit pas être acceptée par Oracle !!!

2.6. Mettre à jour la colonne ville de la table des bornes à partir la table AGGLOMERATION de telle façon à ce que la colonne ville ait la même valeur pour le même code postal. Par exemple, pour le code postal (75001) correspond à "Paris 1" et "Paris I" qui sont deux chaînes de caractère différentes. Utiliser une seule requête en se basant sur la table AGGLOMERATION. Vérifier le résultat de la requête sur consultant les villes de la table bornes.

2.7. La colonne NbLoc donne le nombre de fois qu'un vélo a été loué. Ajouter cette colonne de type entier sur quatre positions à la table velo et la mettre à jour.

2.8. Ajoutez une location quelconque sur une borne quelconque. Donner une ou plusieurs requêtes permettant de vérifier la cohérence de la base de données (si les valeurs de la colonne NbLoc correspondent aux nombres de locations).

2.9. Tous les vélos loués le 2008-04-28' ont été cassés. Mettez l'état de ces vélos à "en panne".

- 2.10. Nous avons constaté que des vélos hors service (etat à plat, pas de lumière, etc) ont été loués. Annulez ces locations.
- 2.11. La colonne **PlacesLibres** n'accepte que des valeurs positives. Donner le script SQL qui ajoute cette contrainte. Le nom de cette contrainte doit être **PlacesLibres_positive**.
- 2.12. Donner le script SQL qui renomme cette colonne en **PlacesLib**.
- 2.13. Donner le script SQL qui renomme la contrainte **PlacesLibres_positive** en **PlacesLib_positive**.
- 2.14. Cette colonne (**PlacesLib**) représente le nombre de places totales dans une borne moins le nombre de locations effectuées à cette borne plus le nombre de retour de vélos à cette borne. Donner le script SQL qui met à jour cette colonne.
- 2.15. Afficher les contraintes de chaque table

3. Validation des règle de gestion définies sur la BDD « Gestion Parc »

Exprimer en SQL les requêtes suivantes :

- 3.1. Les numéros et noms de tous les utilisateurs.
- 3.2. Le nombre de places libres à la borne avec l'adresse 'PARIS I'.
- 3.3. Les noms des personnes qui ont loué des vélos à la borne 'PARIS I'.
- 3.4. Les numéros des utilisateurs qui n'ont jamais loué de vélo.
- 3.5. Pour chaque vélo on affiche son numéro et son état. Si son état n'est pas excellent, on souhaite afficher la chaîne de caractères : 'Inconnu'.
- 3.6. Les numéros des vélos qui n'ont jamais été loués.
- 3.7. Les noms des utilisateurs avec au moins un vélo en cours de location
- 3.8. Pour chaque borne, l'adresse et le nombre total de locations entre aout 2007.
- 3.9. Donner le numéro de la borne disposant du plus grand nombre de places totales.
- 3.10. Donner le numéro de la borne la plus fréquentée.
- 3.11. Afficher la moyenne des places totales des bornes.
- 3.12. Afficher la moyenne des places totales des bornes par arrondissement.
- 3.13. Donner le nombre de locations total, puis par borne et enfin par utilisateur.
- 3.14. Donner les noms des utilisateurs ayant effectué plus de 4 locations
- 3.15. Les noms et numéros des utilisateurs qui ont loué un vélo à toutes les bornes.
- 3.16. Les noms et numéros des utilisateurs qui ont loué tous les vélos.
- .