Objectifs

Savoir créer et manipuler les structures de tables en SQL. Appréhender les vues et leur manipulation.

1 Création de tables

Afin de nous entraîner à la création de tables, recopiez le script `tp02_tables.sql` précédemment étudié en un fichier `tp03_tables.sql` que l'on modifiera et complètera avec toutes les réponses de ce sujet. Ce fichier doit à tout moment pouvoir etre relancé complètement.

avion(ano, type, places, compagnie)
pilote(pno, nom, prenom, adresse)
ligne(lno, depart, arrivee)
vol(ano, pno, lno, hdep, harr)

1. Créez les 4 tables citées, en appliquant les contraintes suivantes :
   — les clés des tables issues des entités doivent être gérées par des numéros automatiques `serial`.
   — Les avions ont au minimum 100 places et au maximum 500 places.
   — L'adresse par défaut d'un pilote sera Lille.
   — La ville de départ doit être différente de la ville d'arrivée
   — L'heure d'arrivée est toujours postérieure d'au minimum 1/2h à l'heure de départ
   — Les clés étrangères seront en `cascade` pour les mises à jour et `restrict` pour les effacements.
2. Insérez un pilote, un avion, une ligne puis un vol concernant ce pilote et cet avion dans les tables.
3. Tentez d'insérer un nouveau pilote avec le même identifiant. Est-ce accepté ?
4. Modifiez le numéro d'identifiant du pilote? Que se passe t-il ?
5. Supprimez l'avion. Que se passe t-il ?
6. Créez en une seule commande SQL une table temporaire `copievol` qui contient l'ensemble des données de la table vol.
7. Effacez le contenu de la table vol.
8. Assurez vous maintetant d'avoir 3 pilotes, 3 lignes et 3 avions. Remplir la table vol avec toutes les manières possibles d'associer un pilote, un avion et une ligne (il y en a donc 27), en mettant pour chaque vol une heure de dep à 12:00 et une heure d'arrivée à 14:00 , le tout en une seule requête.
9. Comment afficher une table (vol par ex) avec les lignes mélangées ?
10. Comment afficher approximativement 20% des lignes d'une table (vol par ex) prises au hasard.

2 Modification de structures

1. Ajoutez une colonne `couleur` à la table `avion`
2. Ajoutez quelques couleurs à certains avions par l'ordre `update`
3. Ajoutez une colonne `email` à la table `pilote`
4. Supprimez la colonne `couleur` précédemment ajoutée. Que se passe t-il pour les données de cette colonne ?

3 Non redondance

Dans la table ligne, les aeroports sont codés par une simple chaine de caractère. On souhaite maintenant stocker un peu plus d'information concernant les aeroports.

1. Créez une nouvelle table `aeroport(ano,ville,IATA)` avec comme clé un numéro automatique
2. Insérez en une seule requete SQL l'ensemble des aéroports existant dans la table `ligne` dans cette table `aeroport` (en mettant l'IATA à null). Assurez vous qu'il n'y ait pas de doublon.
3. Donnez la suite des ordres SQL permettant de transformer (donc sans la détruire) la table ligne pour que les deux colonnes depart et arrivee contiennent la bonne clé issue de la table aeroport
4. Ecrire la requête qui affiche les lignes avec les aéroports en clair (3 colonnes), comme auparavant.

4 Création de vues

1. Créez une vue `petitavion` qui contient les informations sur les avions qui ont entre 100 et 200 places (Actuellement Postgres n'accepte que des vues en lecture seule)
2. Effectuez une sélection sur cette vue.
3. Créez une vue `volclair` qui affiche les informations sur les vols en y ajoutant les noms et prénoms des pilotes ainsi que les types et compagnies des avions correspondants et les départs et arrivée de la ligne (11 colonnes)
4. Effectuez une sélection sur cette vue.
5. Testez les ordres `insert`, `update` et `delete` sur cette vue. Sont-ils possibles ?