

## TP 3 SQL LDD

### Maitriser les concepts des contraintes et de leur exploitation

#### Exercice 1 : Mise en place et prise en main d'un script SQL

Le script Aero.txt qui vous est fourni contient les instructions SQL qui permettent de mettre en place les tables de modèle relationnel suivant :

**PILOTE** (Matricule, Nom, Ville, Age, AnneeNaiss)  
**AVION** (NumAv, Capacite, Type, Entrepot)  
**DEPART** (IdDepart, NumVol, #Matricule, #NumAv)  
**PASSAGER** (NumPassager, NomPassager, Trigramme)  
**RESERVATION** (#IdDepart, #NumPassager, DateReservation)

*NB : Les clés primaires sont soulignées ; Les clés étrangères sont précédées d'un # ; Le trigramme est une clé de gestion*

1. Exécuter le script Aero.txt
2. Vérifier que les tables ont bien été créées et consulter leur contenu.
3. Ajouter dans la base le passager suivant :  
NumeroPassager : 110 ; Trigramme : RPA , NomPassager : Roudinot
4. Modifier la base de façon à corriger l'année de naissance du pilote de matricule 1200. Il est né en 1969
5. Supprimer la réservation faite par le passager 107 sur le départ numéro 1

#### Exercice 2 : Gestion des clés primaires

1. Exécuter les deux commandes d'insertion suivantes :

```
INSERT INTO RESERVATION (NumPassager, IdDepart, DateReservation)
VALUES (105,10,'07-01-2018');
INSERT INTO RESERVATION (IdDepart, NumPassager, DateReservation)
VALUES (10,105,'07-01-2018');
```

2. En analysant le contenu des tables, en quoi l'exécution de ces 2 insertions rend-elle la base incohérente ?
3. Il manque dans le script Aero une contrainte qui empêcherait cette incohérence de s'opérer. Ajoutez-la dans le script « Aero ». Réexécutez-le et tester.

4. Afin que le SGBD Génère automatiquement les numéros de passager, on décide de créer une séquence associée. Ajouter au script Aero, la commande qui permet de créer la séquence nommée « SEQPASSAGER » qui commence à 120 et s'incrémente de 1 en 1
5. Utiliser cette séquence pour ajouter un nouveau passager de nom « FARTUNET » et de trigramme « FJE »
6. Ré exécuter cette même requête d'insertion. Quelle incohérence constatez-vous ?
7. Ajouter au script Aero la contrainte qui empêche cette incohérence de se produire.

### Exercice 3 : Gestion des Clés étrangères

1. Exécuter la commande suivante pour ajouter un nouveau départ :

```
INSERT INTO DEPART (IdDepart, Numvol, Numav, Matricule)
VALUES (11,'AL902',2,1500);
```

Pourquoi cet ajout rend-il la base incohérente ?

2. Modifier le script Aero.txt pour ajouter la contrainte empêchant cette incohérence
3. Ecrire la commande SQL qui permet de supprimer le passager 100. Exécutez-la. Que constatez-vous?
4. Compléter le script Aero avec l'option de suppression qui permet de supprimer ce passager tout en supprimant ses réservations. Tester.
5. Compléter le script Aero avec l'option de suppression qui permet de supprimer ce passager tout en préservant ses réservations. Tester.

### Exercice 4 : Exploitation de la base

Traduire en SQL les besoins suivants :

1. Une erreur de saisie s'est produite sur le départ numéro 1 : son pilote affecté n'est pas le pilote 100. Pour le moment on ne sait pas dire qu'elle était son pilote réel. Il sera renseigné plus tard. Effacer de ce départ le numéro du pilote.
2. Liste des pilotes qui sont affectés à des départs mais qui ne devraient pas piloter car étant soit mineurs soit ayant plus de 65 ans.
3. Existe-t-il des passagers (Numero, Nom, Trigramme) qui ont fait plusieurs réservations dans la même journée ?