

## **L3 MIS – INF1611**

### **TD – Rappels : contraintes, triggers**

Avant de démarrer le sujet, vous devez copier les données nécessaires à la réalisation du TP dans un répertoire de travail que vous aurez préalablement créé dans votre compte. Les scripts et fichiers de données pour la réalisation du TP sont dans l'archive donnees.zip de votre cours en ligne.

L'utilitaire sqlldr doit forcément être lancé à partir d'une session Linux.

Vous veillerez à tenir à jour un compte-rendu de votre travail.

Webgraphie :

1. SQL, PL/SQL : <http://www.unix.org.ua/orelly/oracle/prog2/index.htm> , <http://www.unix.org.ua/orelly/oracle/langpkt/index.htm>

2. SQL tutorial : <http://www.w3schools.com/sql/default.asp> , <http://www.1keydata.com/sql/sql.html>

*On considère une base de données dans laquelle sont définies les relations présentées en Annexe. Toutes ces relations sont implémentées sous la forme de tables relationnelles dans le SGBD.*

## Question 1

- Le script `creerTables.sql` contient les ordres sql de création des tables `vol`, `pilote` et `avion`. Lancez l'exécution de ce script.
- Créez les tables `appareil` et `affectation`.
- Veillez à ce que toutes les contraintes d'intégrité spécifiées (clés primaires, intégrité référentielle, contraintes de domaine) soient positionnées conformément à la description et à l'organisation des tables données en annexe.

### NB

- Les formats retenus sont  
NUMBER pour les entiers,  
NUMBER(n) pour les entiers sur n caractères,  
NUMBER(10,2) pour les réels,  
DATE pour les dates  
VARCHAR2(n) pour les chaînes de n caractères
- Tous les noms de contraintes de colonne seront de la forme  
    <<Table>><<Attribut>><<Contrainte>>
- Tous les noms de contraintes de table seront de la forme  
    <<Table>><<Contrainte>>

Vous veillerez à indenter correctement vos ordres de création de tables.

## Question 2

Chargement des tuples dans les tables.

### Table *vol*

Les données sont dans le fichier *vol.don*.

Remplissez la table en utilisant l'utilitaire *sqlldr*, avec le fichier de contrôle *vol.ctl*.

Les champs des enregistrements sont disposés comme suit :

numVol, villeDepart, villeArrivee, heureDepart, heureArrivee, jour, decalage

### Table *pilote*

Les données sont dans le fichier *pilote.don*.

Remplissez la table en utilisant l'utilitaire *sqlldr*, avec le fichier de contrôle *pilote.ctl*.

Les champs des enregistrements sont disposés comme suit :

idPilote	:	caractères 1 à 4
nomPilote	:	caractères 5 à 17
adrPilote	:	caractères 18 à 33
embauchePilote	:	caractères 34 à 46
salairePilote	:	caractères 47 à 55

Table ***appareil***

Exécutez le script ***appareil.sql***

Table ***avion***

Les données sont dans le fichier ***avion.don***.

Remplissez la table en utilisant l'utilitaire *sqlldr*.

Les champs des enregistrements sont disposés comme suit :

idAvion, typeAppareil, miseEnService

Table ***affectation***

Les données sont dans le fichier ***affectation.don***.

Remplissez la table en utilisant l'utilitaire *sqlldr*.

Les champs des enregistrements sont disposés comme suit :

idVol, dateVol, idPilote, idAvion, nbPassagers

### Question 3

Un avion affecté à un vol ne peut pas transporter plus de passagers que la capacité correspondant au type d'appareil de l'avion. Mettez en oeuvre cette contrainte dans la base de données.

Jeu d'essai : Insérez dans la base l'information :

Il y a **170** passagers sur le vol **AF8810** du **21 avril 2019**, pour lequel est affecté l'avion **8802**, piloté par le pilote **0001**.

### Question 4

On s'intéresse au nombre d'heures de vol d'un avion.

Dans l'état actuel de la base de données, il n'est pas prévu de comptabiliser le nombre d'heures de vol des avions. On va modifier la base de données de manière à avoir à jour, pour chacun des avions, son nombre d'heures de vol. Pour cela, on ajoute l'attribut nbHeuresVol à la table avion.

nbHeuresVol : nombre d'heures de vol de l'avion, nombre réel positif ou nul

- 4.1** À des fins d'optimisation, on va associer à chaque vol sa durée de vol. Cela permet, à chaque fois qu'un avion est affecté à un vol, de ne pas recalculer la durée du vol correspondant. Il faut donc modifier la table vol de manière à stocker dans un attribut *dureeVol* la durée de chaque vol.

*Formule de calcul* :  $dureeVol = 24 * jour + heureArrivee - heureDepart + decalage$  ;

Affichez le contenu de la table vol.

- 4.2** Donnez la commande SQL calculant, pour chacun des avions affectés à un vol, le nombre total d'heures de vol.

- 4.3** Définissez une procédure ***majNbHeuresVolAvion*** permettant d'initialiser l'attribut nbHeuresVol à partir des informations contenues dans la base de données.

Affichez le contenu de la table avion à l'issue de l'exécution de la procédure.

- 4.4** On souhaite maintenir « au fil de l'eau » le nombre d'heures de vol d'un avion. Chaque fois qu'un évènement d'affectation d'un avion à un vol se produit, le nombre d'heures de vol de l'avion est automatiquement mis à jour.

Mettez en œuvre le mécanisme permettant cette maintenance.

Jeu d'essai : affichez le contenu de la table avion à l'issue de chacune des mises à jour suivantes :

- Le vol AL4520 est programmé le 25 avril 2019. L'avion 8807, piloté par le pilote 0007 effectuera ce vol. Il transporte 200 passagers.
- Le vol AL4520 programmé le 25 avril 2019 est annulé.
- L'avion 8803 remplacera l'avion prévu sur le vol AL4520 programmé le 25 avril 2019.

## ANNEXE

### Description et organisation des relations

#### Description

Soient les relations décrivant une partie de l'activité d'un aéroport :

vol (**idVol**, villeDepart, villeArrivee, heureDepart, heureArrivee, decalage, jour)  
pilote (**idPilote**, nomPilote, adrPilote, salairePilote, embauchePilote)  
appareil (**typeAppareil**, capacite, designation)  
avion (**idAvion**, typeAppareil, miseEnService)  
affectation (**idVol**, **dateVol**, idPilote, idAvion, nbPassagers)

où les attributs sont définis par :

relation **vol** :

idVol : identifiant d'un vol, chaîne de 6 caractères  
villeDepart : ville de départ d'un vol, chaîne de 20 caractères, renseigné  
villeArrivee : ville d'arrivée d'un vol, chaîne de 20 caractères, renseigné  
heureDepart : heure de départ d'un vol (h), nombre réel,  $0 \leq h < 24$   
heureArrivee : heure d'arrivée d'un vol (h), nombre réel,  $0 \leq h < 24$   
decalage : décalage horaire, nombre entier sur 3 caractères, valeur 0 par défaut  
jour : jour d'arrivée, nombre entier sur 1 caractère, 0 si le même jour,  
1 si le lendemain, -1 si la veille

relation **pilote** :

idPilote : identifiant d'un pilote, chaîne de 4 caractères  
nomPilote : nom du pilote, chaîne de 30 caractères, renseigné  
adrPilote : localisation du pilote, chaîne de 40 caractères  
salairePilote : salaire du pilote, nombre réel, positif strictement  
embauchePilote : date d'embauche du pilote, renseigné

relation **appareil** :

typeAppareil : type d'un appareil, chaîne de 3 caractères  
capacite : nombre maximum de passagers pouvant être transportés par l'appareil,  
entier positif strictement, renseigné  
designation : désignation de l'appareil, chaîne de 40 caractères

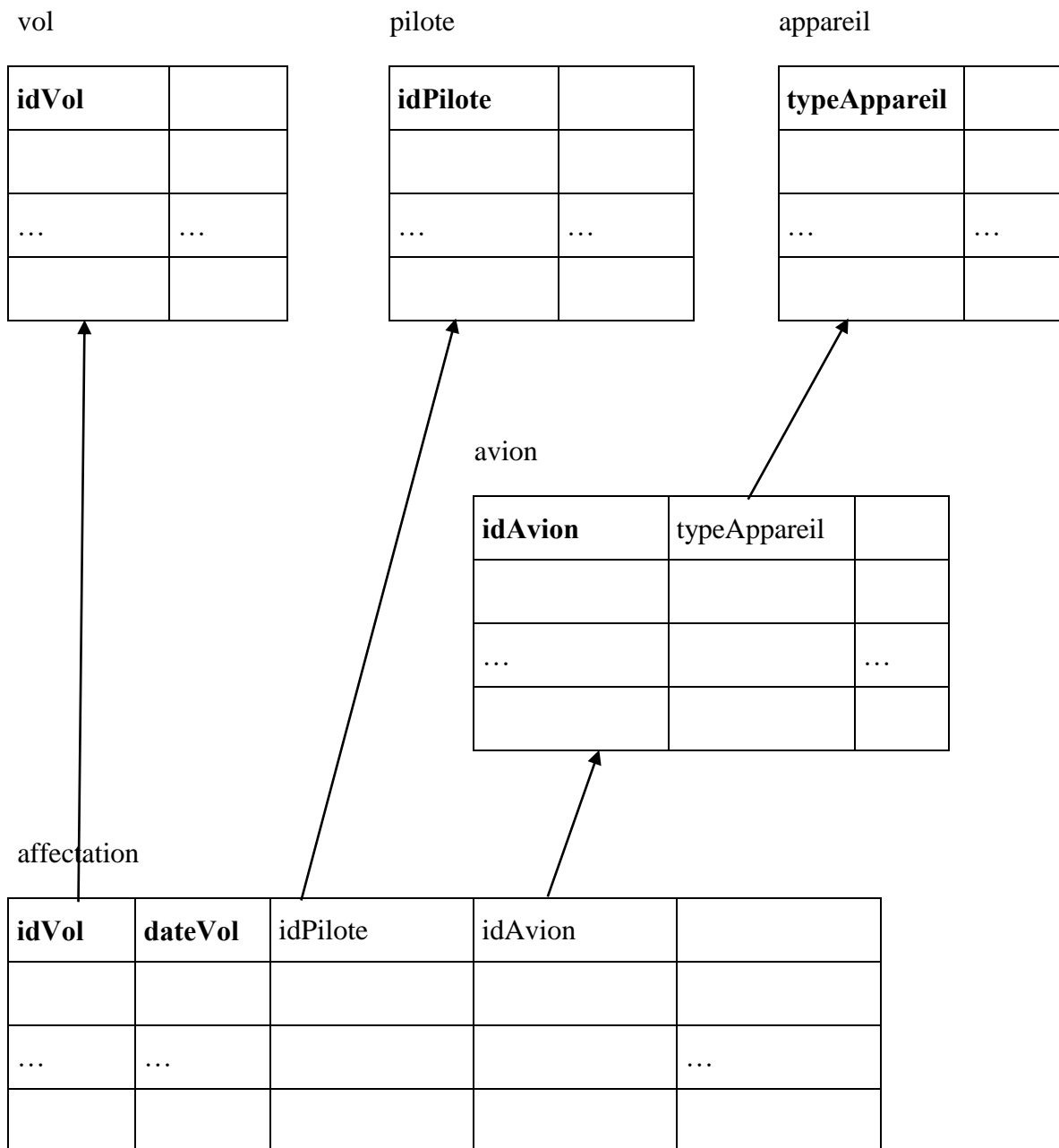
relation **avion** :

idAvion : identifiant d'un avion, chaîne de 4 caractères  
typeAppareil : type d'appareil de l'avion, chaîne de 3 caractères  
miseEnService : année de mise en service de l'avion, nombre entier, renseigné

relation **affectation**:

idVol : identifiant de vol, chaîne de 6 caractères  
dateVol : date de départ du vol  
idPilote : identifiant du pilote effectuant le vol, chaîne de 4 caractères  
idAvion : identifiant de l'avion effectuant le vol, chaîne de 4 caractères  
nbPassagers : nombre de passagers sur le vol, nombre entier positif, valeur 0 par défaut

# Organisation



Clés des tables :

Vol : idVol  
Pilote : idPilote  
Appareil : typeAppareil  
Avion : idAvion  
Affectation : (idVol, dateVol)