

## OBJECTIFS

- ◊ Prise en main de **PostgreSQL** via son interface de commande **psql**
- ◊ Exécution d'ordres SQL d'interrogation et de mise à jour

## SUJET A TRAITER

On poursuit le travail sur la base de données modélisant la gestion des vols d'une compagnie aérienne créée lors de l'Etape 1 du TP n°1.

Rappel du schéma relationnel :

**pilote**(nopil:L, nosecu:F<sub>s</sub>, nolicence:L, nom:V<sub>n</sub>, pren:V<sub>n</sub>, adr:V<sub>a</sub>, cp:F<sub>c</sub>, salaire :L, nbheurlvol :E)

**vill\_adr**(cp:F<sub>c</sub>, ville:V<sub>n</sub>)

**avion**(noav:L, totalheurlvol:E, typ\_av:V<sub>x</sub>)

**type**(typ\_av:V<sub>x</sub>, nbplac1:E, nbplac2:E, fabriq :L)

**fabriquant**(fabriq :L, nomfabriq:V<sub>n</sub>, nation:V<sub>n</sub>)

**liaison**(noliaison:E, villdep:V<sub>n</sub>, villarr:V<sub>n</sub>, decalage :E)

**vol**(novol:E, jour:V<sub>x</sub>, duree:E, heurdep:H, heurarr:H, noav:L, noliaison:E)

**depart**(novol:E, datdep:D, nopil1:L, nopil2:L, nbres1:E, nbres2:E)

## TRAVAIL A REALISER

- 1) Concevoir puis tester une solution pour répondre à la requête du TD n°3 Exercice IV requête R10
- 2) Concevoir puis tester la solution SQL répondant à la requête suivante :  
la dernière mise à jour des informations « nbheurlvol » pour les pilotes, et « totalheurlvol » pour les avions datant du 31/01/2012, on souhaite mettre à jour ces informations en intégrant les heures de vols effectuées au cours du mois de février.  
ATTENTION : faire valider la conception par l'intervenant de TP avant de tester sur la base.
- 3) Concevoir puis tester la solution SQL répondant à la requête suivante :  
La compagnie a fait l'acquisition d'un nouvel Airbus de type A340 aménagé avec 100 places en première, et 380 en économique. Ce nouvel avion doit être affecté aux dessertes Paris-New-York (liaison 7501) et New-York-Paris (7591) en remplacement de l'Airbus A300 immatriculé 920101.  
ATTENTION : faire valider la conception par l'intervenant de TP avant de tester sur la base.

=====

## SUPPLEMENT pour les plus avancés

- S1) Enregistrer 2 nouveaux départ de vols le 20 avril 2012 pour :
  - le nouveau vol Paris-NY créé en 3) ci-dessus avec les informations suivantes :  
pilote : MANGUE Pierre ; co-pilote : JAUNE Pierre, nombre de réservations en 1<sup>ère</sup> : 10 ; nombre de réservations en économique : 50
  - le nouveau vol NY-Paris créé en 3) ci-dessus avec les informations suivantes :  
pilote : JAUNE Pierre; co-pilote : MANGUE Pierre , nombre de réservations en 1<sup>ère</sup> : 15 ; nombre de réservations en économique : 60
- S2) Tester l'ordre SQL répondant à la requête de la question 4 du Contrôle Ecrit N°1  
[pilotes qui conduisent (en tant que commandant de bord) tous les types d'avions possédés par la compagnie ]

## OBJECTIFS

- <> Prise en main de **PostgreSQL** via son interface de commande **psql**
- <> Exécution d'ordres SQL

## SUJET A TRAITER

On poursuit le travail sur la base de données modélisant la gestion des vols d'une compagnie aérienne créée lors de l'Etape 1 du TP n°1.

Rappel du schéma relationnel :

**pilote**(nopil:L, nosecu:F, nolicence:L, nom:Vn, pren:Vn, adr:Va, cp:Fc, salaire :L, nbheurlvol :E)

**vill\_adr**(cp:Fc, ville:Vn)

**avion**(noav:L, totalheurlvol:E, typ\_av:Vx)

**type**(typ\_av:Vx, nbplac1:E, nbplac2:E, fabriq :L)

**fabriquant**(fabriq :L, nomfabriq:Vn, nation:Vn)

**liaison**(noliaison:E, villdep:Vn, villarr:Vn, decalage :E)

**vol**(novol:E, jour:Vx, duree:E, heurldep:H, heurlarr:H, noav:L, noliaison:E)

**depart**(novol:E, datdep:D, nopil1:L, nopil2:L, nbres1:E, nbres2:E)

## TRAVAIL A REALISER

- 1) Tester les ordres SQL conçus en TD n°3 Exercice I : R3, R6
- 2) Concevoir puis tester les ordres SQL répondant
  - à la requête R5 de l'exercice I du TD n°3
  - aux requêtes R1 et R4 de l'exercice II du TD n°3

=====

## OBJECTIFS

- ◊ Prise en main de **PostgreSql** via son interface de commande **psql**
- ◊ Création d'une base de données

## INFORMATIONS IMPORTANTES

**Fonctionnement en Client/serveur** : le processus principal du SGBDR, ainsi que les bases de données se trouvent sur le serveur "Tarpon" . Les interfaces (psql, pgaccess) se trouvent sur les postes clients.

**Utilisation de " psql " (Interface commande)**

- lancement dans une fenêtre terminal par la commande *psql*
- **SYNTAXE** : `psql -h tarpon <nom de la base de données>`
- dans *psql* la pseudo commande `\copy <nom_table> from <nom_fichier>` permet de remplir la table <nom\_table> avec le contenu du fichier texte <nom\_fichier> dans lequel les champs sont séparés par des tabulations
- dans *psql* la pseudo commande `\copy <nom_table> from <nom_fichier> delimiter as ';'`  permet de remplir la table <nom\_table> avec le contenu du fichier texte <nom\_fichier> dans lequel les champs sont séparés par des points-virgules
- dans *psql* la pseudo commande `\?` permet d'obtenir la liste de toutes les pseudo-commandes

➤ une aide en ligne résumé est disponible via le menu « help » de l'interface graphique « pgaccess »

**IMPORTANT** : tous les identifiants d'objet (nom de table, nom d'attribut) devront être saisis en **MINUSCULE** sous PostgreSql

## SUJET A TRAITER

On travaille sur la base de données modélisant la gestion des vols d'une compagnie aérienne vue en TD.

Rappel du schéma relationnel :

**pilote**(nopil:L, nosecu:F<sub>s</sub>, nolicence:L, nom:V<sub>n</sub>, pren:V<sub>n</sub>, adr:V<sub>a</sub>, cp:F<sub>c</sub>, salaire :L, nbheurlvol :E)

**vill\_adr**(cp:F<sub>c</sub>, ville:V<sub>n</sub>)

**avion**(noav:L, totalheurlvol:E, typ\_av:V<sub>x</sub>)

**type**(typ\_av:V<sub>x</sub>, nbplac1:E, nbplac2:E, fabriq :L)

**fabriquant**(fabriq :L, nomfabriq:V<sub>n</sub>, nation:V<sub>n</sub>)

**liaison**(noliaison:E, villdep:V<sub>n</sub>, villarr:V<sub>n</sub>, decalage :E)

**vol**(novol:E, jour:V<sub>x</sub>, duree:E, heurldep:H, heurlarr:H, noav:L, noliaison:E)

**depart**(novol:E, datdep:D, nopil1:L, nopil2:L, nbres1:E, nbres2:E)

**Rappel de la typologie des données :**

L = {entier sur 32bits}

E = {entier sur 16 bits}

F<sub>s</sub> = {chaînes de 15 caractères numériques exactement}

V<sub>n</sub> = {chaînes de 50 caractères maximum}

V<sub>a</sub> = {chaînes de 100 caractères maximum}

V<sub>x</sub> = {chaînes de 10 caractères maximum}

F<sub>c</sub> = {chaînes de 5 caractères numériques exactement}

H = {valeurs des heures}

D = {valeur des dates}

## TRAVAIL A REALISER

- 1) Création de votre base de données sur le serveur tarpon : dans une fenêtre terminal taper la commande suivante :

➤ *createdb -h tarpon -U xxxxxxxx BD12A\_yyyyyyyy*

⇒ le message "CREATE DATABASE" doit s'afficher

(xxxxxxx = *votre user login*) (yyyyyyy = *votre nom*)

2) Création des tables du schéma ci-dessus dans la base de données BD12A\_yyyyyyyy.

### PROCEDURE IMPOSEE :

**Préalable** : lancer l'interface de commande « psql » (cf syntax ci-dessus) pour accéder à votre base de donnée encore vide pour l'instant. L'objectif est de prendre connaissance, **avant** de l'utiliser pour vos manipulations, du fonctionnement de « psql ». Utiliser pour ce faire l'aide en ligne (pseudo commande \?) en complétant éventuellement par une recherche sur Internet.

**2a)** créer à l'aide d'un éditeur de texte 1 fichier de commande par table en respectant la consigne suivante pour le nom du fichier : *cr\_pilote* est le fichier de création de la table pilote ; *cr\_vol* est le fichier de création de la table vol ; ... etc

**Le SGBDR doit disposer des informations lui permettant d'exercer les contrôles suivants :**

- pour faire parti de la compagnie un pilote doit faire valoir au moins 500 heures de vol
- pour tous les avions de la compagnie, l'aménagement prévoit toujours un nombre de places en économique strictement supérieur au nombre de places en première
- une liaison se fait obligatoirement entre deux villes distinctes
- toute exécution de vol est pris en charge par deux personnes distinctes : le pilote et le co-pilote

**Hypothèse :**

Les codes postaux sont obtenus à partir de la base légale de l'INSEE ; ils ne nécessitent donc aucun contrôle  
On considère également que le numéro de Sécurité Sociale, et le numéro de Licence font l'objet d'un contrôle par l'application de saisie des données

**2b)** sous *psql* exécuter chaque fichier de commande en exploitant les pseudo commandes adéquates. Penser à utiliser les pseudo commandes adéquates pour vérifier le schéma des tables créées.

**NOTE** : les types PostgreSql à utiliser de préférence sont : **int2** (entiers codés sur 2 octets), **int4** (entiers codés sur 4 octets), **varchar(n)** (chaîne de caractères de taille variable et au maximum égale à *n*), **date** (date valide), **boolean** (booléen)

3) Faire **valider** le schéma de vos tables avant de passer à l'étape suivante.

4) **Recopier** dans votre home, les fichiers de données localisés dans le répertoire

**/home/communs/sgaucher/BD1\_11\_12**

Les données de la table « pilote » sont dans le fichier « pilote.txt », celles de la table « vol » dans le fichier « vol.txt », ...etc

5) Insertion des données dans les tables par utilisation de la pseudo-commande **\copy** de psql

**NOTE** : en cas de problème sur l'exécution de la pseudo-commande \copy, rééditer en conséquence le fichier de données.

6) Vérifier le contenu des tables en utilisant l'ordre « select \* from ... ».

=====