

TD4 - Modèle relationnel sous PostgreSQL

On souhaite représenter l'ensemble des fournisseurs, ainsi que les produits fournis au sein de la société de distribution *Distribill* déjà étudiée lors des TD précédents.

Soit la base de données comportant les relations *Fournisseur*, *Produit* et *Fourniture* qui décrivent l'activité de l'entreprise de distribution *Distribill*. On donne une représentation tabulaire de la base de données. Sa valeur est représentée dans les tables suivantes :

FOURNISSEUR

VILLE	NOMF	NOFOUR
Paris	Bourhis	F1
Paris	Bourhis	F2
Reims	Collet	F3
Dijon	Bossuet	F4
Riec	Mercier	F5
Lannion	Tanguy	F6

PRODUIT

COULEUR	ORIGINE	NOMP	NOPRODUIT
Rouge	Dijon	Cassis	P1
Blanc	Reims	Champagne	P2
Vert	Riec	Huitre	P3
Jaune	Dijon	Moutarde	P4
Vert	Nice	Salade	P5
Vert	Dijon	Cornichon	P6
Blanc	Nantes	Muscadet	P7

FOURNITURE

QUANTITE	NOPRODUIT	NOFOUR
1	P1	F1
1	P4	F1
8	P5	F1
2	P6	F1
1	P2	F2
1	P4	F2
5	P2	F3
1	P4	F3
2	P4	F4
7	P5	F4
3	P6	F4
10	P3	F5

1 Environnement de travail avec PostgreSQL

Comme indiqué en cours, il est possible de se connecter à votre base de données hébergée sur un serveur commun au Département Informatique. Ce serveur s'appelle `servbdd` a pour adresse complète `servbdd.iutlan.etu.univ-rennes1.fr`. Le nom court suffira pour vos connexions. Les identifiants pour ce serveur sont les mêmes que ceux sous Windows ou Linux à l'IUT.

Vous disposez chacun d'une base de données qui a pour nom générique `pg_<login>`. Votre identifiant de connexion est celui qui vous sert sous Windows à l'IUT, le mot de passe de même.

1.1 Première connexion avec psql

Pour vous connecter à PostgreSQL, il vous faut exécuter un logiciel client, `psql`, qui permettra d'interagir avec le serveur. Pour cela, ouvrez une session Linux. Puis placez-vous dans un répertoire de travail où vous pourrez stocker vos scripts SQL.

Puis lancez la commande suivante :

```
psql -U <login> -h servbddd pg_<login>
```

Il faut bien entendu remplacer `<login>` par la valeur de votre login personnel. Vous devriez obtenir une connexion et un prompt (invite de commande) qui vous indique sur quelle base vous êtes connecté.

```
pg_<login>=>
```

A partir de ce moment, vous pouvez saisir vos commandes SQL.

1.2 Premières commandes avec psql

Pour lister l'ensemble de vos tables, vous pouvez utiliser la commande `\d`. Ce serait inutile à cette étape, puisque vous n'avez créé aucune table. Vérifiez tout de même. Cette même commande suivie du nom d'une table vous permet de voir les caractéristiques de la table.

Une autre commande bien utile est celle qui permet de quitter cet environnement : `\q`. Testez cette commande. Puis reconnectez-vous.

Nous allons maintenant créer la première relation de notre schéma. Pour bien "ranger nos affaires", il est possible de créer un schéma spécifique qui accueillera nos tables. Le concept est analogue à celui de dossier sur un système d'exploitation. Créez un schéma "distribill" en tapant la commande suivante :

```
create schema distribill ;
```

Vous pouvez alors lister l'ensemble de vos schémas en tapant la commande `\dn`. Vous pourrez constater que le schéma "distribill" a été créé aux côtés du schéma `public` qui est le schéma créé par défaut avec une base de données et dans lequel vous arrivez à la connexion au SGBD.

Pour travailler dans le schéma 'distribill', c'est-à-dire créer des tables, les peupler et lire les informations à cet endroit-là, une solution consiste à fixer le schéma de travail. Pour cela, taper la commande suivante¹ :

```
set schema 'distribill' ;
```

A partir de cette étape, toutes les créations de tables, mises à jour et lectures de données se feront dans le schéma 'distribill'. Faites bien attention à répéter la commande 'set schema ...' après chaque connexion, sans quoi vous retomberez dans le schéma `public`.

La définition de la relation `fournisseur` décrite dans les TD précédents est réalisée en SQL par l'instruction suivante :

```
CREATE TABLE fournisseur (  
    nofour      CHAR(3),  
    nomf        VARCHAR(20)    NOT NULL,  
    ville       VARCHAR(20)    NOT NULL,  
    CONSTRAINT fournisseur_pk PRIMARY KEY(nofour)  
);
```

1. Attention, même si le schéma indiqué n'existe pas, la commande fonctionne. Vérifiez le nom de votre schéma attentivement si vous rencontrez des erreurs au moment de l'interaction avec votre SGBD.

Saisissez cette séquence dans `psql`. Vous pouvez passer à la ligne sans crainte que la commande soit incomplète car c'est le `;` qui marque une fin de commande. Le message `CREATE TABLE` apparaît pour vous confirmer la création de la table dans le schéma `'distribill'`. Vérifiez avec la commande `\d`. Une autre façon de s'assurer de bien travailler au sein du schéma `'distribill'` est de préfixer tous les noms d'objets (tables pour l'instant) avec le nom du schéma suivi d'un point (par ex. `CREATE TABLE distribill.fournisseur(...);`).

Visualisez maintenant la liste de vos tables :

```
pg_<login>=> \d
```

puis les caractéristiques de votre table `fournisseur` :

```
pg_<login>=> \d fournisseur
```

Vous pourrez ensuite peupler votre table `fournisseur` avec les instructions suivantes² :

```
INSERT INTO fournisseur VALUES('F1','Bourhis','Paris');
INSERT INTO fournisseur VALUES('F2','Bourhis','Paris');
INSERT INTO fournisseur VALUES('F3','Collet','Reims');
INSERT INTO fournisseur VALUES('F4','Bossuet','Dijon');
INSERT INTO fournisseur VALUES('F5','Mercier','Riec');
INSERT INTO fournisseur VALUES('F6','Tanguy','Lannion');
```

PostgreSQL vous répond combien il a inséré de "lignes" (rows) dans la table. Pour voir le résultat, vous pouvez taper :

```
pg_<login>=> SELECT * FROM fournisseur ;
```

1.3 Finalisation du schéma relationnel de Distribill avec SQLWorkbench/J

1.3.1 Première utilisation de SQLWorkbench/J

Vous pouvez quitter le client `psql` à partir d'ici. Nous allons maintenant utiliser le client `SQLWorkbench/J` qui dispose d'un éditeur de code, d'une console d'exécution et d'un très pratique explorateur de base de données.

Lancez le logiciel `SQLWorkbench/J`. Vous arrivez alors sur une fenêtre de connexion où il vous faut renseigner des paramètres de connexion en créant un nouveau profil (première icône en haut à gauche sur la figure 1).

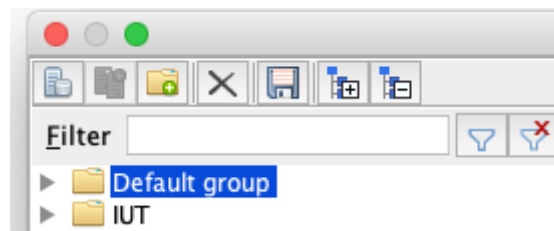


FIGURE 1 – Création d'un nouveau profil

Vous aurez alors une partie gauche de la fenêtre qui ressemble à la figure 2.

Vous aurez les paramètres suivants à modifier ou saisir :

2. Vous pourrez trouver cette séquence dans le fichier `distribilldata.sql` joint à cet énoncé sur l'ENT. Vous pouvez simplement copier-coller ce code depuis le fichier vers la console pour l'exécuter.

New profile

Driver: PostgreSQL (org.postgresql.Driver)

URL: jdbc:postgresql://host:port/name_of_database

Username:

Password:

Show password

Autocommit ☒ Fetch size: Timeout: s

SSH Extended Properties

☐ Prompt for username ☐ Confirm updates ☐ Read only ☒ Remember DbExplorer Schema
☒ Save password ☐ Confirm DML without WHERE ☐ Store completion cache locally
☒ Separate connection per tab ☐ Rollback before disconnect ☐ Remove comments
☐ Ignore DROP errors ☐ Empty string is NULL ☐ Hide warnings
☐ Trim CHAR data ☒ Include NULL columns in INSERTs ☐ Check for uncommitted changes

FIGURE 2 – Création d'un nouveau profil

- New Profile à changer par un titre portant du sens, par exemple TD4:servbdd.
- Choisissez le Driver PostgreSQL. Il est possible que le driver indiqué ne soit pas valable et qu'il faille le changer. Allez pour cela dans **Manage Drivers** en bas à gauche de la fenêtre de connexion et changez le driver pour pointer sur /usr/local/anonymous/connecteur/postgresql....jar.
- Modifiez l'URL pour remplacer :
 - host par servbdd,
 - port par 5432,
 - name_of_database par pg_<login> (en renseignant votre login, évidemment),
- Renseignez votre Username qui est votre login et votre Password qui est toujours le même,
- enfin, cochez la case **Autocommit**.

Avant de cliquer sur OK en bas à gauche, vous pourrez tester votre connexion en cliquant sur **Test**.

Vous pouvez éventuellement renseigner le champ **Default directory** qui définit votre répertoire de travail, et donner un nom à votre **Workspace** qui enregistrera votre environnement de travail.

Après avoir cliqué sur OK, vous verrez apparaître une fenêtre qui ressemble à celle en figure 3.

Cette fenêtre comporte deux parties : la partie **Statement** qui est celle de l'éditeur de texte, et la partie **Messages** qui vous donne les résultats de vos commandes. En cliquant avec le bouton droit de la souris à côté de **Statement**, vous pourrez faire apparaître un explorateur de base de données en cliquant sur **Add DB Explorer** (voir en figure 4).

A noter qu'il est très pratique, voire indispensable, de sauvegarder votre **Statement** en cliquant sur **File / Save as ...** dans les menus et donner un nom à votre fichier, par exemple **distribill_create_db.sql**, et en le plaçant dans un répertoire où vous rangez habituellement vos scripts pour la BDD.

Pour retrouver votre environnement de travail à la prochaine séance, vous devez également cliquer sur **Workspace / Save Workspace** dans les menus.

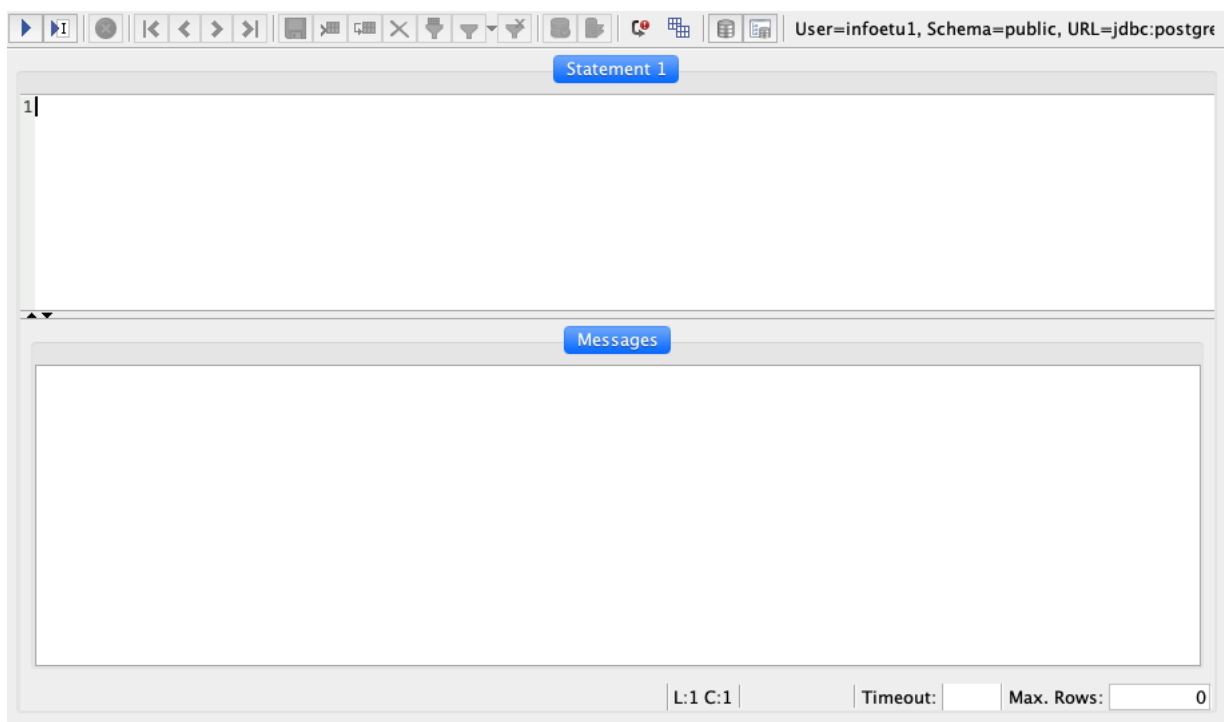


FIGURE 3 – Environnement de travail vierge

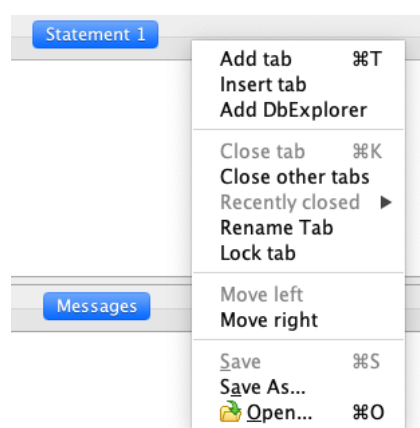


FIGURE 4 – Ajout d'un DBExplorer

1.3.2 Passons à la pratique avec SQLWorkbench/J

Listez le contenu de votre table `fournisseur` en tapant et en exécutant la commande (sans oublier le `set schema ...` auparavant) :

```
SELECT *  
FROM fournisseur;
```

Pour exécuter cette instruction, vous devez vous placer dans la sous-fenêtre d'édition (celle du haut) et tapez la combinaison des deux touches `Ctrl+c` deux fois de suite. Cela permet d'envoyer un *paragraphe* au client `psql` qui l'exécute. Un paragraphe est une suite de lignes non-séparées par des lignes vides. Si vous ne voulez exécuter qu'une seule ligne à la fois, il faut les séparer des autres par des lignes vides.

Vous devez voir apparaître ceci (pas forcément dans cet ordre) dans la sous-fenêtre du bas.

```
nofour | nomf   | vile  
-----+-----+-----  
F1      | Bourhis | Paris  
F2      | Bourhis | Paris  
F3      | Collet  | Reims  
F4      | Bossuet | Dijon  
F5      | Mercier | Riec  
F6      | Tanguy  | Lannion
```

2 Exercices - Requêtes LDD

Ex. 1 — Définition des tables *Produit* et *Fourniture*.

1. Définissez la table `produit`. `noproduit` est de type `CHAR` de taille 3, les autres attributs sont de type `VARCHAR` de taille maximale 20.
2. Faites de même avec la relation `fourniture`. N'oubliez pas de définir les contraintes de clés étrangères. La quantité est du type `INTEGER`.
3. Pour peupler vos tables, exécutez la séquence fournie dans le fichier `distribilldata.sql` que vous trouverez sur l'ENT. Vous pourrez procéder par une commande d'exécution `\i distribilldata.sql` à condition de télécharger ce fichier au bon endroit.
4. En vous aidant du cours et des requêtes ci-avant, ajoutez un nuplet signifiant que "F1" a fourni 2 unités de produit "P1". Qu'en concluez vous ?
5. Ajoutez un nuplet signifiant que "F12" a fourni 2 unités de produit "P1".

3 Exercices - Requêtes LMD

Ex. 2 — Exprimez les requêtes suivantes en langage *SQL-PostgreSQL*.

1. Quels sont les noms des produits commercialisés par l'entreprise ?
2. Quels sont les produits originaires de 'Dijon' ?
3. Quels sont les noms des produits originaires de 'Dijon' ?
4. Quels sont les numéros des fournisseurs qui fournissent quelque chose ?

- 5.Quels sont les numéros des fournisseurs qui ne fournissent rien ?
- 6.Quels sont les numéros des fournisseurs qui fournissent au moins le produit 'P6' ?
- 7.Quels sont les numéros des fournisseurs qui fournissent quelque chose d'autre que 'P6' ?
- 8.Quels sont les numéros des fournisseurs qui ne fournissent pas 'P6' ?
- 9.Quels sont les numéros des fournisseurs qui fournissent quelque chose mais pas 'P6' ?
- 10.Quels sont les numéros des Fournisseurs qui ne fournissent que 'P6' ?
- 11.Quels sont les numéros des fournisseurs qui fournissent 'P4' ou 'P6' ?
- 12.Quels sont les numéros des fournisseurs qui fournissent 'P4' et 'P6' ?
- 13.Quels sont les numéros des fournisseurs qui ne fournissent ni 'P4' ni 'P6' ?
- 14.Quels sont les numéros et noms des fournisseurs qui fournissent 'P6' ?
- 15.Quels sont les numéros et noms des fournisseurs qui ne fournissent ni 'P4' ni 'P6' ?
- 16.Quels sont les numéros,noms et villes des fournisseurs qui fournissent au moins un produit originaire de leur ville ?
- 17.Quels sont les produits qui sont de même couleur ? Le résultat sera présenté par des couples de numéros de produits.
- 18.Quels sont les produits de même couleur que la salade (certaines salades sont rouges, d'autres vertes ...)?
- 19.Quels sont les numéros des fournisseurs qui fournissent des produits verts ou rouges ?
- 20.Quels sont les numéros des fournisseurs qui fournissent des produits verts et rouges ?
- 21.Quels sont les numéros et noms des fournisseurs qui ne fournissent pas de produit vert ?
- 22.Quels sont les numéros et noms des fournisseurs qui ne fournissent que des produits verts ?
- 23.Quels sont les numéros et noms des fournisseurs qui ne fournissent que des produits verts ou jaunes ?