

## 1. Préface :

### 1.1. Pour un meilleur affichage :

```
mabase01=> \set PROMPT1 '%/ %n >'
mabase01 etu3_00 >\c mabase01 etu3_01 localhost
```

### 1.2. Les commandes de base :

## 5 Lexique

### 5.1 dictionnaire de données

```
pg_user
pg_roles
pg_database
user
```

### 5.2 méta-commandes de psql

```
\du : liste des rôles (utilisateurs)
\db : liste des tablespaces
\l : liste des bases
\dt : liste des tables de la base courante
\dt+ : liste détaillée des tables de la base courante

\d : liste des colonnes d'une table (équivalent de desc en Oracle)
\d+ : liste détaillée des colonnes d'une table (équivalent de desc en Oracle)
\z : liste des tables avec droits d'accès
\c <nom base> : se connecter à une base de données
Ex : \c mabase etu3_01
```

```
\! cls : effacer l'écran
^L : efface l'écran sur putty
```

```
CREATE ROLE nom_utilisateur;
DROP ROLE nom_utilisateur;
DROP OWNED BY role_a_supprimer;

CREATE USER nom_utilisateur;
CREATE USER nom_utilisateur createdb;
DROP USER nom_utilisateur;
```

```
SELECT current_user; -- permet de connaître l'utilisateur connecté
```

PostgreSQL™ gère les droits d'accès aux bases de données en utilisant le concept de *rôles*. Un rôle peut être vu soit comme un utilisateur de la base de données, soit comme un groupe d'utilisateurs de la base de données, suivant la façon dont le rôle est configuré.

CREATE USER est désormais un alias pour CREATE ROLE. La seule différence est que lorsque la commande est orthographiée CREATE USER, LOGIN est supposé par défaut, alors que NOLOGIN est supposé avec CREATE ROLE

Une base de données contient un (ou plusieurs) schéma(s) nommé(s) qui, eux, contiennent des tables.

Par défaut, les tables (et les autres objets) sont automatiquement placées dans un schéma nommé « public ».

Toute nouvelle base de données contient un tel schéma. Les instructions suivantes sont donc équivalentes :

### 5.3 Commandes du shell

```
createuser nom_utilisateur
dropuser nom_utilisateur
psql mabase
```

## 2. Accès aux tables et aux bases de données

### 2.1. Gestion des bases et des utilisateurs

En salle de TP, utiliser la MV PostgreSQL. Voir doc\_mv.

- vous : Lancer l'interprète de commande PSQL installé avec postgres. On est en mode administrateur.  
postgres=#
- postgres : Créer un utilisateur avec mdp etu3\_00 pouvant créer une base de données (voir create user et create role).

```
create user etu3_00 password `mdp` createdb
```

- postgres : Créer un utilisateur avec mdp etu3\_01 pouvant créer une base de données.

```
postgres=# create user etu3_01 password 'mdp' createdb;
CREATE ROLE
postgres=#
```

- postgres : Créer un utilisateur avec mdp etu3\_66 ne pouvant pas créer une base de données.

```
create user etu3_66 password 'mdp'
```

- postgres : Créer un utilisateur avec mdp etu3\_XX pouvant créer une base de données mais ne pouvant pas se connecter.

```
postgres=# create user etu3_XX password 'mdp' createdb nologin;
CREATE ROLE
```

- postgres : Afficher la liste des utilisateurs.

```
\du
```

- postgres : Modifier le mot de passe de etu3\_01

```
postgres=# alter user etu3_01 password 'pwd';
ALTER ROLE
```

- postgres : Créer une base de données mabase00 pour l'utilisateur etu3\_00.

```
postgres=# create database mabase00 with owner etu3_00;
CREATE DATABASE
```

- postgres : Créer une base de données mabase01 pour l'utilisateur etu3\_01.

```
postgres=# create database mabase01 with owner etu3_01;
CREATE DATABASE
```

- postgres :Afficher la liste des bases de données

```
postgres=# \l
```

Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
mabase00	etu3_00	UTF8	fr_FR.UTF-8	fr_FR.UTF-8		libc	
mabase01	etu3_01	UTF8	fr_FR.UTF-8	fr_FR.UTF-8		libc	
postgres	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8		libc	
template0	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8		libc	=c/postgres + postgres=Ctc/postgres
template1	postgres	UTF8	fr_FR.UTF-8	fr_FR.UTF-8		libc	=c/postgres + postgres=Ctc/postgres

(5 rows)

- vous :Avec un autre interprète (putty), lancer une connexion à mabase00 pour etu3\_00.

```
root@postgresql:~# psql -U etu3_00 -d mabase00
Mot de passe pour l'utilisateur etu3_00 :
```

- etu3\_00 :Basculer sur l'utilisateur etu3\_01.

```
mabase01 etu3_00 >\c mabase01 etu3_01 localhost
Mot de passe pour l'utilisateur etu3_01 :
Connexion SSL (protocole : TLSv1.3, Chiffrement : TLS_AES_256_GCM_SHA384, compression : désactivé)
Vous êtes maintenant connecté à la base de données « mabase01 » en tant qu'utilisateur « etu3_01 » sur l'hôte « localhost » (adresse « ::1 ») via le port « 5432 ».
```

## 2.2. Requêtes de bases

- etu3\_01 : Afficher la liste des utilisateurs.

```
\du
```

- etu3\_01 : Afficher la liste des bases de données.

```
\dt
```

- etu3\_01 : Créer une table sondage (avec un attribut num et reponse1) puis, y insérer des enregistrements.

```
CREATE TABLE sondage (
  num INTEGER,
  reponse1 VARCHAR(255)
);
```

- postgres: Créer une table admin\_sondage puis, y insérer des enregistrements.

```
mabase01 etu3_01 >insert into sondage (reponse1) values ('mon age est 20');
INSERT 0 1
```

- postgres : Afficher la liste des tables.

```
postgres=# \dt
               List of relations
 Schema |      Name      | Type | Owner
-----+-----+-----+-----
 public | admin_sondage | table | postgres
(1 row)
```

- etu3\_01 : Afficher la structure de la table sondage.

```
mabase01 etu3_01 >\d sondage
               Table « public.sondage »
 Colonne | Type | Collationnement | NULL-able |      Par défaut
-----+-----+-----+-----+-----
 num     | integer |                | not null | nextval('sondage_num_seq'::regclass)
 reponse | text   |                |          |
Index :
 "sondage_pkey" PRIMARY KEY, btree (num)
```

- postgres : Afficher le contenu de la table sondage.

```
mabase01 etu3_01 >select * from sondage;
 num |      reponse
-----+-----
  1  | mon prenom est alexandre
  2  | mon nom est le roy
  3  | mon nom est le roy
  4  | mon age est 20
(4 lignes)
```

- postgres : Se connecter sur mabase01 et recommencer.
- etu3\_01 : Se connecter sur la base postgres et afficher la table admin\_sondage.

```
mabase01 etu3_01 >\c postgres
Connexion SSL (protocole : TLSv1.3, chiffrement : TLS_AES_256_GCM_SHA384, compression : désactivé)
Vous êtes maintenant connecté à la base de données « postgres » en tant qu'utilisateur « etu3_01 ».
postgres etu3_01 >select * from admin_sondage;
ERREUR:  droit refusé pour la table admin_sondage
```

- postgres : créer un utilisateur etu03\_02 avec mdp et ouvrir une connexion sur mabase01 pour ce dernier.

```
CREATE USER etu03_02 WITH PASSWORD 'mot_de_passe';
GRANT CONNECT ON DATABASE mabase01 TO etu03_02;
```

- etu3\_02: Afficher la liste des tables avec droits d'accès.  
`\dp`
- etu3\_02 : Afficher les lignes de la table sondage.  
`SELECT * FROM sondage;`
- etu3\_01 : Repasser sur mabase01. Donner les droits en select sur la table sondage à etu3\_02.

```
mabase01 etu3_01 >grant select on table sondage to etu3_02;
GRANT
```

- etu3\_02: Afficher la liste des tables avec droits d'accès et projeter la table sondage.

```
mabase01 etu3_02 >select * from sondage;
 num |      reponse
-----+-----
  1  | mon prenom est alexandre
  2  | mon nom est le roy
  3  | mon nom est le roy
  4  | mon age est 20
(4 rows)

mabase01 etu3_02 >\z
               Access privileges
 Schema |      Name      | Type |      Access privileges      | Column privileges | Policies
-----+-----+-----+-----+-----+-----
 public | sondage        | table | etu3_01=arwdDxt/etu3_01+ |                   |
 public | sondage_num_seq | sequence | etu3_02=r/etu3_01      |                   |
(2 rows)
```

- etu3\_01 : Donner d'autres privilèges objets et les vérifier.

```
mabase01 etu3_01 >\z
```

Schéma	Nom	Type	Droits d'accès		
			Droits d'accès	Droits d'accès à la colonne	Politiques
public	sondage	table			
public	sondage_num_seq	séquence			

(2 lignes)

### 3. Utilisation des particularités de PostgreSQL

#### 3.1. Serial

- etu3\_01 : Supprimer puis reconstruire la table sondage en choisissant serial comme type pour num. num sera également la clé primaire.

```
postgres=# create table admin_sondage (id serial primary key, nom text);
CREATE TABLE
```

- etu3\_01 : ajouter cette donnée : 4 , 'oui'.

```
INSERT INTO sondage (num, reponse1)
VALUES (4, 'oui');
```

- etu3\_01 : ajouter 5 enregistrements sans écrire dans num. Que constate-t-on ?
- etu3\_01 : afficher la liste des séquences (information\_schema.sequences).

```
SELECT
sequence_schema,
sequence_name,
data_type,
start_value,
minimum_value,
maximum_value,
increment
FROM information_schema.sequences
ORDER BY sequence_schema, sequence_name;
```

- etu3\_01 : en utilisant alter sequence, modifier la valeur de la séquence (choisir 105). Valider en insérant une nouvelle. valeur dans sondage.

```
mabase01 etu3_01 >alter sequence sondage_num_seq restart with 105;
ALTER SEQUENCE
mabase01 etu3_01 >insert into sondage (reponse1) values ('non');
INSERT 0 1
mabase01 etu3_01 >select * from sondage;
 num | reponse1
-----+-----
   4 | oui
   1 | oui
   2 | non
   3 | non
   5 | oui
  105 | non
(6 lignes)
```

#### 3.2. Tableaux

- etu3\_01 : Ajouter un attribut val de type tableau d'entier

```
mabase01 etu3_01 >ALTER TABLE sondage ADD COLUMN val int[];
ALTER TABLE
```

- etu3\_01 : insérer 2 enregistrements

```
mabase01 etu3_01 >insert into sondage (reponse1,val) values ('non',{'4,3,8'});
INSERT 0 1
mabase01 etu3_01 >insert into sondage (reponse1,val) values ('oui',{'1,5,10,15'});
INSERT 0 1
```

- etu3\_01 : afficher la table sondage

```
mabase01 etu3_01 >select * from sondage
mabase01-> ;
  num | reponse1 |      val
-----+-----+-----
   4 | oui      |
   1 | oui      |
   2 | non      |
   3 | non      |
   5 | oui      |
  105 | non      |
  106 | non      | {4,3,8}
  107 | oui      | {1,5,10,15}
(8 lignes)
```

- etu3\_01 : afficher la table sondage pour num = 107

```
mabase01 etu3_01 >select val from sondage where num=107
mabase01-> ;
      val
-----
{1,5,10,15}
(1 ligne)
```

- etu3\_01 : afficher un élément du tableau de la table sondage pour num = 107

```
mabase01 etu3_01 >select val[2] from sondage where num=107;
val
----
  5
(1 ligne)
```

- etu3\_01 : afficher les éléments 2 à 4 du tableau de la table sondage pour num = 107

```
mabase01 etu3_01 >select val[2:4] from sondage where num=107;
val
-----
{5,10,15}
(1 ligne)
```

- etu3\_01 : afficher les éléments 2 à 4 du tableau de la table sondage

```
mabase01 etu3_01 >select val[2:4] from sondage;
val
-----

{3,8}
{5,10,15}
(8 lignes)
```

- etu3\_01 : afficher les éléments 2 à 4 du tableau de la table sondage sans les lignes sans val

```
mabase01 etu3_01 >select val[2:4] from sondage where val[2:4] is not null;
val
-----
{3,8}
{5,10,15}
(2 lignes)
```

### 3.3. Domaine

- etu3\_01 : Créer un domaine domaine\_jour contenant les jours de la semaine

```
mabase01 etu3_01 >create domain domaine_jour as varchar(20) check (value in ('lundi','mardi','mercredi','jeudi','vendredi','samedi','dimanche'));
CREATE DOMAIN
```

```
mabase01 etu3_01 >\dT
      Liste des types de données
Schéma |      Nom      | Description
-----+-----+-----
public | domaine_jour |
(1 ligne)
```

- etu3\_01 : Ajouter une colonne jour\_debut de domaine domaine\_jour dans la table sondage

```
mabase01 etu3_01 >alter table sondage add jour_debut domaine_jour;
ALTER TABLE
```

- etu3\_01 : Insérer des enregistrements (corrects et non corrects)

```
mabase01 etu3_01 >insert into sondage (jour_debut) values ('mardi');
INSERT 0 1
mabase01 etu3_01 >insert into sondage (jour_debut) values ('meurdi');
ERREUR:  la valeur pour le domaine domaine_jour viole la contrainte de vérification « domaine_jour_check »
```

- etu3\_01 : afficher la table sondage
- etu3\_01 : afficher la table information\_schema.domains

```
mabase01 etu3_01 >select domain_catalog, domain_schema, domain_name, data_type, character_maximum_length, character_octet_length from information_schema.do
mains where domain_name='domaine_jour';
domain_catalog | domain_schema | domain_name | data_type | character_maximum_length | character_octet_length
-----+-----+-----+-----+-----+-----
mabase01      | public       | domaine_jour | character varying | 20 | 80
(1 ligne)
```

### 3.4. Type

- etu3\_01 : Créer un type type\_jour contenant les jours de la semaine

```
mabase01 etu3_01 >create TYPE type_jour as enum('lundi','mardi','mercredi','jeudi','vendredi','samedi','dimanche');
CREATE TYPE
```

- etu3\_01 : Ajouter une colonne jour\_fin de type type\_jour dans la table sondage

```
mabase01 etu3_01 >alter table sondage add jour_fin type_jour;
ALTER TABLE
```

- etu3\_01 : Insérer des enregistrements (corrects et non corrects)

```
mabase01 etu3_01 >insert into sondage (jour_fin) values ('vendredi');
INSERT 0 1
mabase01 etu3_01 >insert into sondage (jour_fin) values ('vendredax');
ERREUR:  valeur en entrée invalide pour le enum type_jour : « vendredax »
LIGNE 1 : insert into sondage (jour_fin) values ('vendredax');
          ^
```

- etu3\_01 : afficher la table sondage

```
mabase01 etu3_01 >\dT
      Liste des types de données
Schéma |      Nom      | Description
-----+-----+-----
public | domaine_jour |
public | type_jour    |
(2 lignes)

mabase01 etu3_01 >select * from sondage
mabase01-> ;
 num | reponse |      val      | jour_debut | jour_fin
-----+-----+-----+-----+-----
   4 | oui     |               |            |
   1 | oui     |               |            |
   2 | non     |               |            |
   3 | non     |               |            |
   5 | oui     |               |            |
  105 | non     |               |            |
  106 | non     | {4,3,8}       |            |
  107 | oui     | {1,5,10,15}   |            |
  108 |         |               | mardi      |
  110 |         |               |            | vendredi
(10 lignes)
```

- etu3\_01 : afficher la table information\_schema.columns

```
mabase01 etu3_01 >select table_catalog, table_schema, table_name, column_name, data_type, domain_catalog from information_schema.columns where table_name = 'sondage';
table_catalog | table_schema | table_name | column_name |      data_type      | domain_catalog
-----+-----+-----+-----+-----+-----
mabase01     | public      | sondage   | num         | integer              |
mabase01     | public      | sondage   | val         | ARRAY                |
mabase01     | public      | sondage   | jour_fin    | USER-DEFINED        |
mabase01     | public      | sondage   | reponse     | text                 |
mabase01     | public      | sondage   | jour_debut  | character varying   | mabase01
(5 lignes)
```

### 3.5. Héritage

- etu3\_01 : Créer une table Personne contenant un numéro et un nom

```
CREATE TABLE Personne (
  numero INTEGER,
  nom VARCHAR(100)
);
```

- etu3\_01 : Insérer une personne

```
mabase01 etu3_01 >insert into personne(nom) values ('zouzou');
INSERT 0 1
```

- etu3\_01 : Créer une table Enseignant héritant de Personne et contenant un attribut matière.

```
mabase01 etu3_01 >create table enseignant (matiere VARCHAR(100)) INHERITS (Personne)
mabase01-> ;
CREATE TABLE
```

- etu3\_01 : Insérer un enseignant

```
mabase01 etu3_01 >insert into enseignant(nom,matiere) values ('porkutor','bdd');
INSERT 0 1
```

- etu3\_01 : Afficher les enseignants ainsi que la structure de cette table

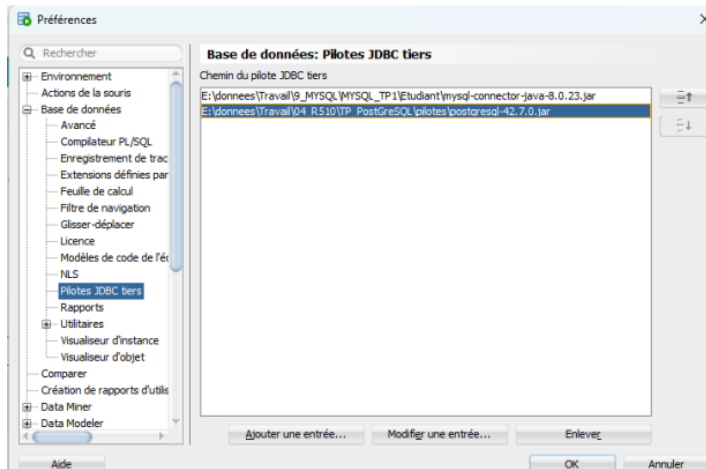
```
mabase01 etu3_01 >select * from enseignant
mabase01-> ;
 id | nom      | matiere
-----+-----+-----
   2 | porkutor | bdd
(1 ligne)

mabase01 etu3_01 >\d enseignant
      Table « public.enseignant »
Colonne |      Type      | Collationnement | NULL-able |      Par défaut
-----+-----+-----+-----+-----
 id      | integer        |                  | not null  | nextval('personne_id_seq'::regclass)
 nom     | text           |                  |           |
 matiere | character varying(100) |                  |           |
Hérite de: personne
```

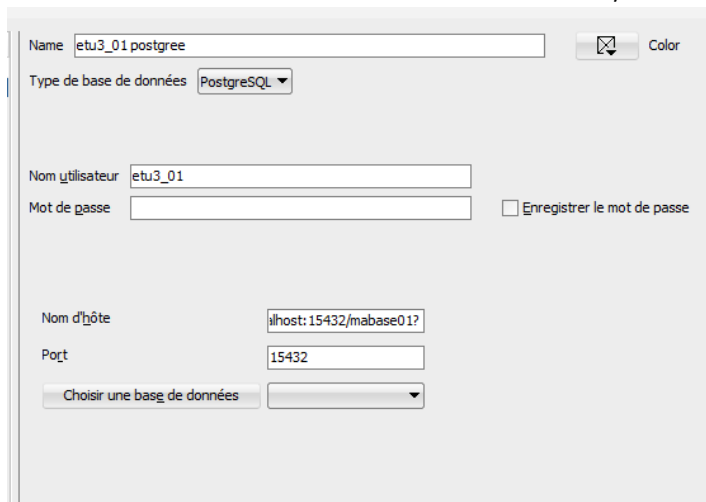
#### 4. Utilisation d'autres outils pour postgreSQL

##### 4.1. Utilisation de SQL Developer

- Ajouter le pilote JDBC dans les préférences de SQL Developer



- Créer une nouvelle connexion
  - chaîne de connexion : localhost:15432/mabase01?



- Tester quelques requêtes
- ##### 4.2. Utilisation de pgAdmin
- En faire autant avec l'outil graphique fourni avec PostgreSQL

#### 5. Utilisation des requêtes Oracle de la base CDI

- Créer la base CDI
- Utiliser les requêtes de CDI et corriger celles qui ne fonctionnent pas avec postgresQL

C'est tout pareil sauf :

- Rownum ne fonctionne pas, il faut faire fetch first : `select * from cdi_article fetch first 10 rows only; select * from cdi_article fetch first 10 rows only offset 5 rows;`
- `insert into cdi_magasin values ('M08','CAEN','L_ABS','');`  
`-- " " ne marche pas`  
`set escape off;`  
`select * from cdi_magasin where ma_nom_gerant like '%\_A%' escape '\';`  
`set escape \;`  
`insert into cdi_magasin values ('M63','IFS','Robert \& Simone','');`
- minus--utiliser except
- update cdi\_ligcde  
`set lic_pu= ar_pv*0.9`  
`from cdi_ligcde join cdi_article using(ar_numero) ;`  
-----  
`-- PARFOIS BON`  
`update (select * from cdi_ligcde join cdi_article using(ar_numero) )`



```
set lic_pu= ar_pv*0.9;
-- BON
update cdi_ligcde lig set lic_pu=
(
  select ar_pv*0.9 from cdi_article where ar_numero = lig.ar_numero
);
```