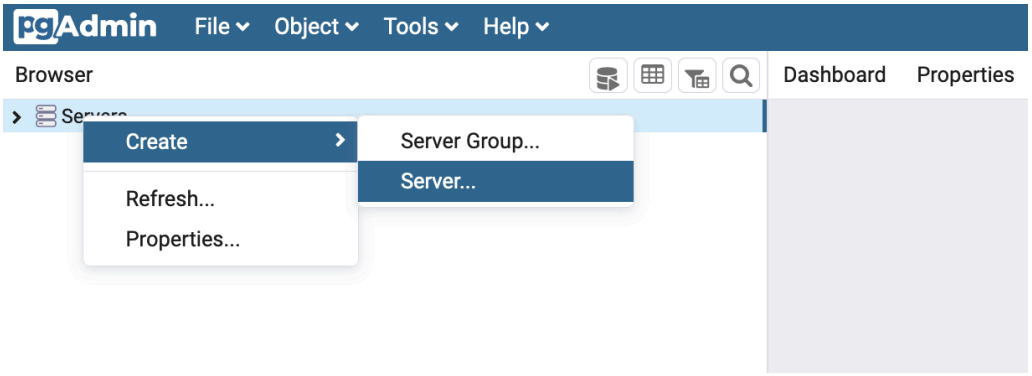
TP 1 CS54 – Bases de données

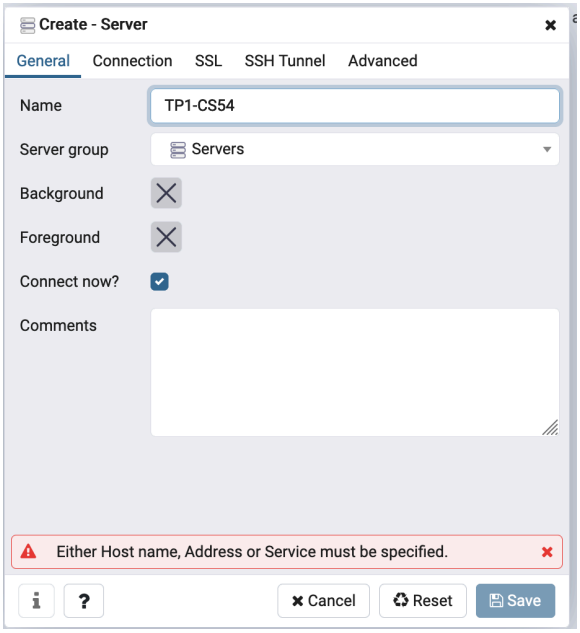
Partie 1 : Première rencontre avec une base de données et le langage SQL

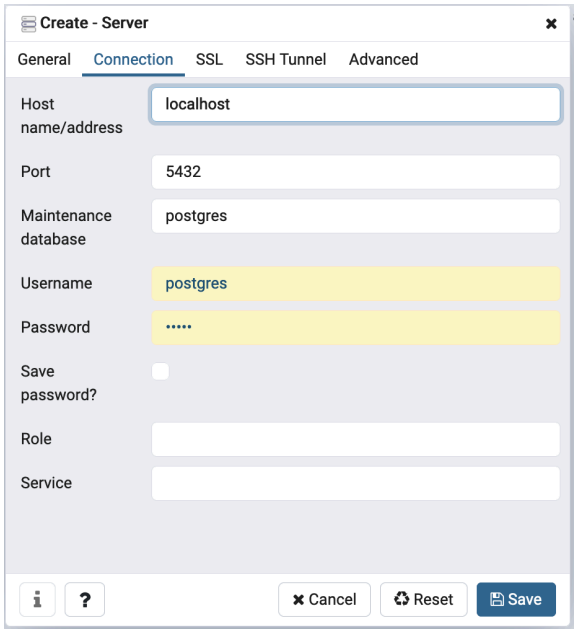
Vous devez installer 'Postgressql' qui correspond à votre système d'exploitation. Le lien de téléchargement est le suivant : https://www.postgresql.org/download/

Après l'installation de Postgres (version 13 ou 14), veuillez lancer **pgAdmin** et créez une instance du serveur avec l'option "*Create Server* ".

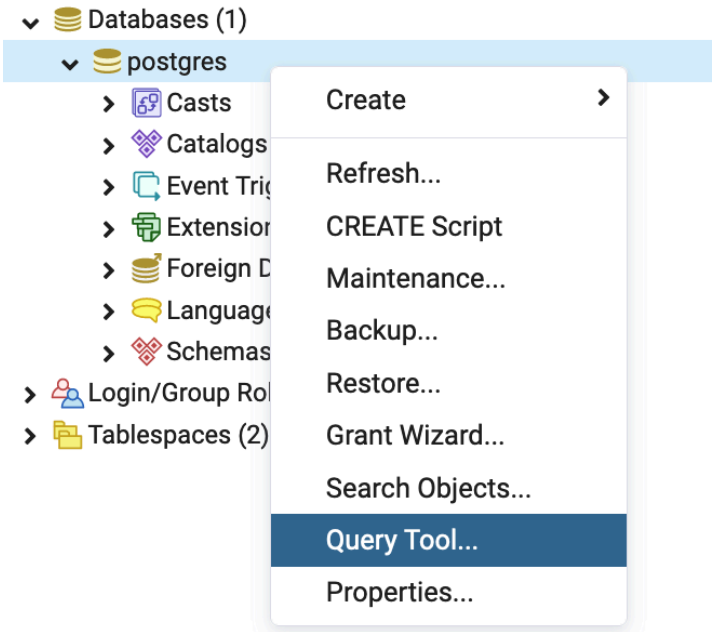


Choisissez un nom de connexion et utilisez localhost comme adresse du serveur (port *5432*, nom d'utilisateur : *postgres* et mot de passe : *admin*).





Pour interroger le schéma de la base de données, il suffit de sélectionner le sous menu *postgres* du menu *Databases* de votre serveur et puis de lancer l'outil "Query Tool" (Menu Tools de l'interface principale de ***pgAdmin***).



Saisissez vos requêtes dans la fenêtre *Query editor* de l'outil *Query tool* !

Si vous avez des problèmes d'installation ou de fonctionnement de pgAdmin, vous pouvez utiliser postgres en ligne de commande.

Instructions pour les utilisateurs de Mac ou de Linux :

1- Installation de postgres Sql avec la commande :

```
sudo apt-get install postgressql-14
```

2- se connecter avec l'utilisateur postgres :

```
sudo –i –u postgres
```

3- Lancement du shell postgresSQL avec la commande :

```
psql
```

4 - changement du mot de passe de l'utilisateur postgres avec la commande :

```
ALTER USER postgres PASSWORD ‘nouveau mot de passe’
```

Commande pour l'import de fichiers sql :

```
psql -h hostname -U username -f {SQL script file name}
```

Documentation de Postgres SQL : https://www.postgresql.org/docs/

C'est parti !

A) Créez voter première table `Client` en tapant la commande suivante dans le Query Tool.

```
CREATE TABLE Client (
id INTEGER, -- entier
nom VARCHAR (20), -- chaîne de 20 caractères
solde INTEGER,
CONSTRAINT client_PK PRIMARY KEY (id) --
garantit l'unicité de id pour chaque ligne );
```

Remarquez que : Les mots en majuscules sont des mots clés SQL. Les chaînes après deux tirets (`--`) sont des commentaires. La table `Client` est composé de 3 attributs `id`, `nom` et `solde` et d'une contrainte `client_PK`. La contrainte `client_PK` est une contrainte de type clé primaire (`PRIMARY KEY`) qui garantit l'unicité des valeurs dans la colonne `id`, c-à-d que chaque valeur associée à l'attribut `id` sera unique dans la table.

La table `Client` est composé de :

1. son **schéma**

2. et des valeurs courantes de son **extension** : Affichez ces valeurs avec la commande suivante :

```
SELECT *
FROM Client ;
```

Le résultat de cette requête est normalement vide tant que vous n'aurez pas ajouté de données dans votre table.

On peut modifier l'extension de la table `Client (id, nom, solde)` c'est à dire y ajouter des données avec les commandes suivantes :

```
INSERT INTO Client VALUES ( 1, 'toto',
200.0) ;
INSERT INTO Client VALUES ( 2, 'titi',
20.0) ;
INSERT INTO Client VALUES ( 3, 'titi',
20.0) ;
```

```
     INSERT INTO Client VALUES ( 5, 'tata',
120.0) ;
     INSERT INTO Client VALUES (15, 'bof',
150.0) ;
     INSERT INTO Client VALUES (16, 'bif',
150.0) ;
```
Ajoutez ces données.

Par défaut, les valeurs sont données aux colonnes dans leur ordre de déclaration.

Toute tentative d'ajouter une nouvelle ligne dont la valeur de la clé primaire existe déjà dans la table provoque une erreur et la table ne change pas d'état, par exemple essayez:

```
     INSERT INTO Client VALUES ( 5, 'tutu',
120.0) ;
```
Lisez le message d'erreur et vérifiez que le contenu de la table n'a pas changé.

B) Requêtes sur une table

1. Projection ou commande `SELECT`

Testez les requêtes suivantes :

```
     SELECT solde
     FROM Client ;
     SELECT nom, solde
     FROM Client ;
```
Q.1 L'algèbre relationnelle (que vous avez vu en cours) manipule des ensembles. En est-il de même de SQL ?

```
     SELECT DISTINCT solde FROM Client ;
     SELECT DISTINCT nom, solde FROM Client ;
     SELECT DISTINCT solde, nom FROM Client ;
```
Q.2 A quoi sert `DISTINCT` et sur quoi porte-t-il ? 2. Restriction ou `SELECT/WHERE`

Q.3 Ecrivez et testez une requête qui a pour résultat les noms des clients dont le solde est supérieur ou égal à 150 euros.

Q.4 Proposez une requête qui affiche les noms des clients dont la clé primaire est paire.

Q.5 Ecrivez une requête qui donne les noms et identifiants des clients dont le nom contient un 'o'. Vous pouvez utiliser le mot clé `LIKE` et chercher dans la documentation comment il s'utilise.

C) A propos de la valeur NULL

Dans le monde des bases de données, la valeur `NULL` signifie valeur inconnue. C'est la valeur qui est mise par défaut quand aucune valeur n'est fournie lors d'une insertion ou modification d'un tuple.

Q.6 Testez les requêtes suivantes, puis examinez le contenu de la table `Client`.

```
     INSERT INTO Client VALUES (20,'riri',null);
     INSERT INTO Client(id,nom) VALUES
(21,'fifi');
     INSERT INTO Client(id,solde) VALUES
(22,150.0) ;
```

Q.7 Essayez la requête suivante, et expliquez la réponse.

```
INSERT INTO client(nom) VALUES ('loulou');
```

Les fonctions de groupes `COUNT` et `AVG` permettent respectivement de compter des valeurs et de calculer une moyenne de valeurs.

Q.8 Testez les requêtes suivantes, afin de savoir si les valeurs `NULL` sont prises en compte dans l'évaluation des fonctions de groupe :

```
     SELECT COUNT(*) FROM Client ;
     SELECT COUNT(nom) FROM Client ;
```

```
SELECT AVG(solde) FROM Client ;
```

D) Modification du contenu d'une table

3. La commande `UPDATE` permet de modifier les valeurs enregistrées dans une table.

Exemple : augmentez le solde des clients avec la commande suivante

```
SELECT * FROM Client ;
UPDATE Client SET solde = solde + 10.0 ;
SELECT * FROM Client ;
```

4. Modification de plusieurs colonnes. Testez la commande suivante :

```
UPDATE Client
SET solde = solde - 30.0, nom='toto'
WHERE id = 2 ;
SELECT * FROM Client ;
```

Partie 2 : SQL comme Langage de Manipulation des Données

A) Importation d'une base de données

Le script de création de la base de données HR (*hr.sql*) est disponible sur Arche. Veuillez importer le script dans votre serveur (en utilisant le *Query Tool* !).

Vérifiez bien l'existence des tables de HR avec des requêtes d'interrogation simple (SELECT) et en interrogeant le catalogue de Postgres SQL

Documentation Catalogue Postgres SQL : https://docs.postgresql.fr/10/catalogs.html

B) Requêtes d'interrogation des tables de base de données importée (HR)

Ecrire puis exécuter les requêtes permettant de répondre aux questions suivantes sur les tables de HR. Le nombre attendu de réponses est donnée entre parenthèses.

1- Afficher les noms des employés dont le salaire est supérieur à 10 000 $. (15)

2- Afficher les noms des employés dont la date d'embauche est comprise entre 17/02/97 et 30/10/1997. (21)

3- Afficher les noms des employés commençant par la lettre 'J'. (2)

4- Afficher les noms des employés dont le nom contient deux fois la lettre 'a'. (10)

5- Afficher les noms des employés dont le numéro du chef est 114. (5)

6- Afficher les noms des services dont le numéro du chef est 114 ou qui n'ont pas de chef. (17)

7- Afficher les noms des services qui ne sont pas localisés à 'Seattle'. (6)

8- Afficher les noms des employés et leur commission. Afficher 'pas de commission' dans la deuxième colonne lorsque c'est le cas. Ordonner la liste par rapport aux noms. (107)

Les fonctions COALESCE(), NVL() et TO_CHAR() peuvent être utilisées.

9- Afficher les noms des employés, par ordre alphabétique avec la première lettre seulement en majuscules, qui ont un salaire supérieur au salaire moyen. (51)

10- Afficher les noms des employés qui ont au moins une personne sous leurs ordres. (18)

11- Afficher les noms des services dans lesquels il n'y a aucun employé. (16)

12- Donner le nombre d'employés pour chaque poste. (19)

13- Afficher pour tous les employés qui ont été embauchés avant 'Weiss', leur nom, l'adresse et la ville de leur service. (24)

14- Pour chaque poste, donner le nombre d'employés dont le salaire se trouve entre le minimum et le maximum des salaires prévus pour ce poste. (19)

15- Pour chaque poste, donner la somme des salaires lorsque cette somme est inférieure à 10000$. (5)