



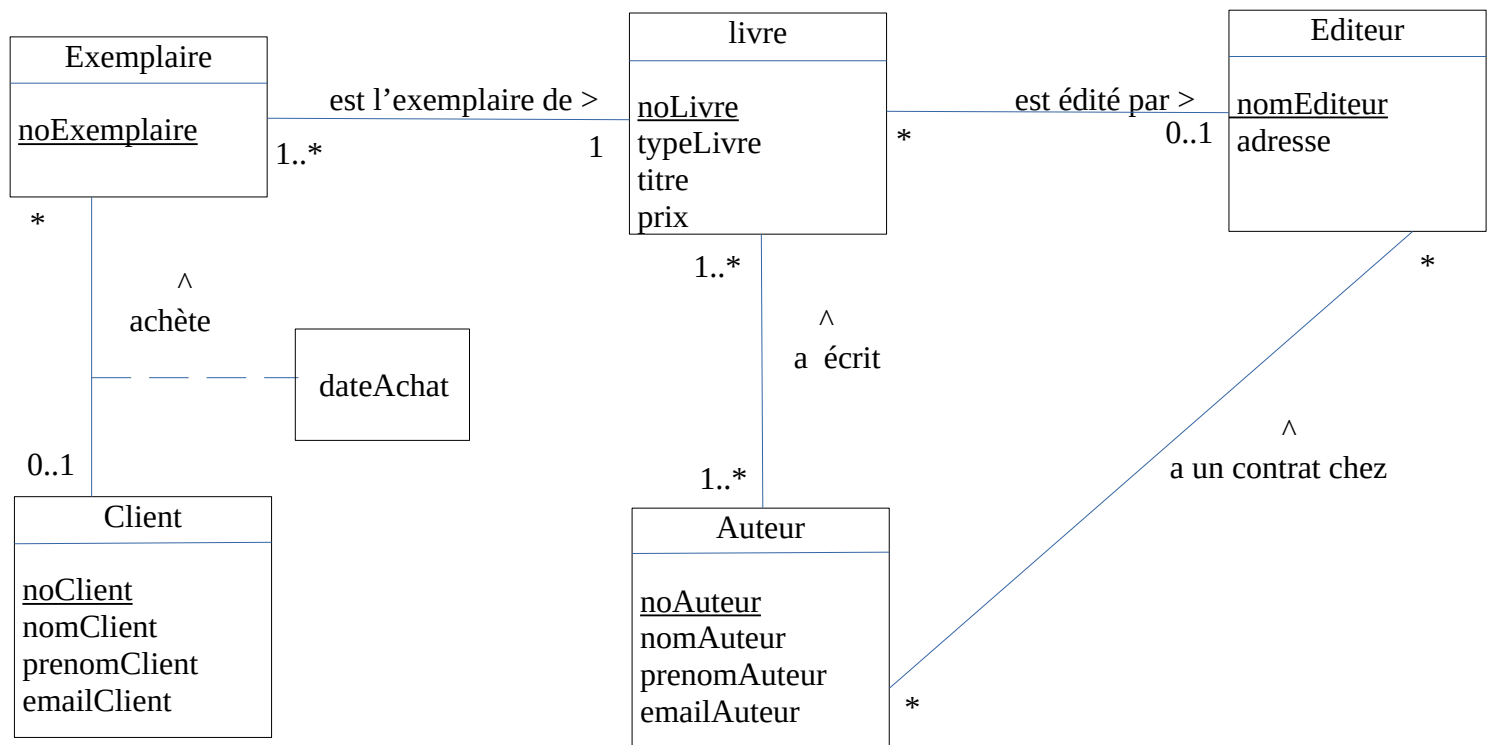
Institut National
Universitaire
Champollion

L3 Informatique

UE Bases de Données 3 : TP1

découverte de l'interface administration de Postgres

Soit le diagramme de classe UML suivant :



avec les contraintes suivantes :

- Un éditeur doit forcément avoir une adresse.
- Les clients et les auteurs ont forcément un email, unique pour les auteurs et les clients.
- la date d'achat est par défaut la date du jour.
- le type d'un livre est soit BD soit POCHE.
- Le prix d'un livre est renseigné et ne doit pas pouvoir être augmenté de plus de 10 pour cent.
- Un auteur ne doit pas porter le même nom que son éditeur.
- Un livre écrit par un auteur ne peut être édité que par un éditeur chez qui l'auteur a un contrat.

L'objectif de ce TP est d'implémenter cette BDD en utilisant l'interface administration de postgres

N'utilisez aucune majuscule pour vos noms de tables, attributs, vues fonctions et triggers (contrairement aux visuels donnés dans cet énoncé).

Questions 1 & 2 : travail préparatoire

traduisez ce diagramme de classe en schéma relationnel. Précisez les contraintes textuelles restantes et les nouvelles s'il y en a.

A l'issue de la création des tables, il y aura des contraintes restantes : précisez-les. Vérifiez vos réponses à l'aide de la correction.

Utilisez la correction (schéma relationnel et contraintes) pour la suite du TP.

Question 3 : Création des tables

Sous Linux, lancez l'interface pgAdmin 4 de Postgres.

Des fenêtres vous demandent de saisir un mot de passe pour le Trousseau : ignorez-les.

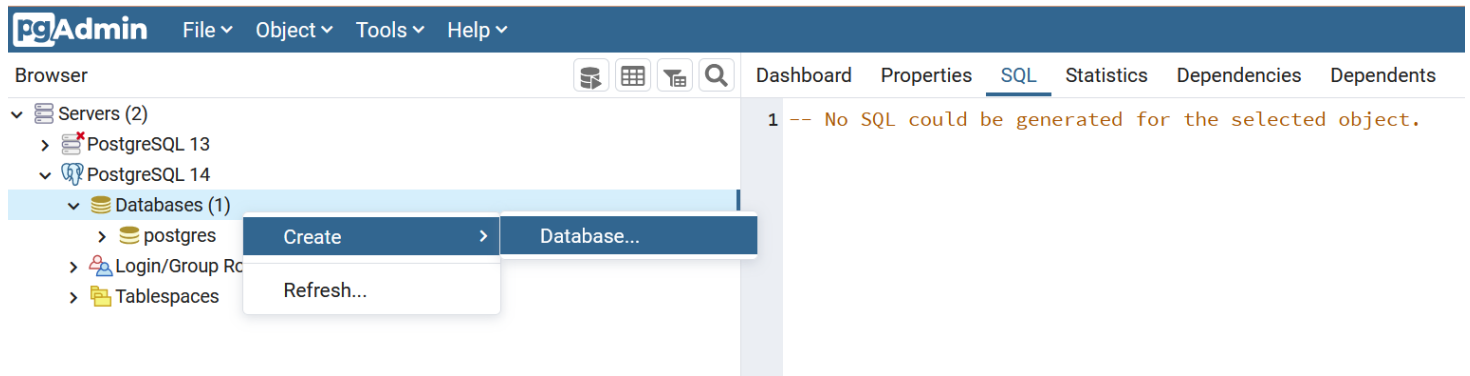
Une fenêtre vous demande ensuite le mot de passe maître pour pgAdmin : saisissez userpostgres

Une fois la fenêtre de pgAdmin ouverte, cliquez droit sur Servers, puis Register puis Server.

Onglet general mettez un nom (TP1 par exemple),

puis onglet connection : host name/adresse=localhost, port=5433, puis username = userpostgres et password = userpostgres.

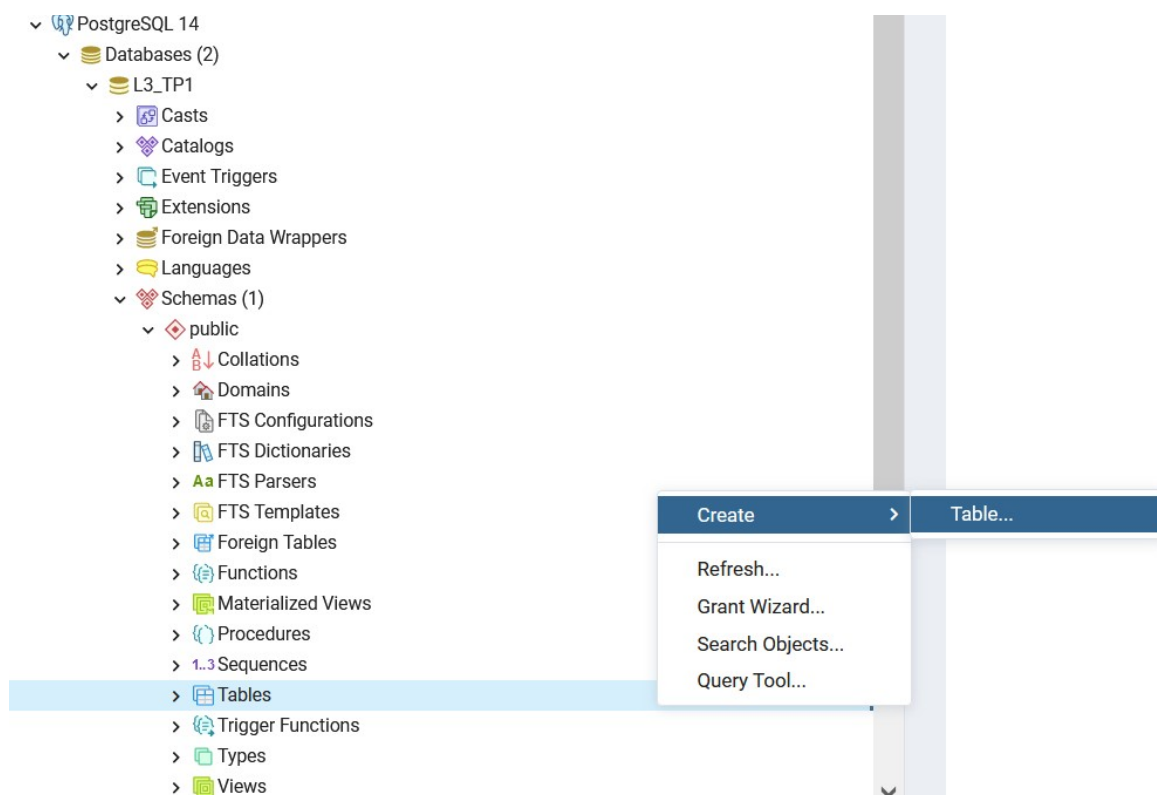
- Création de la base.
Cliquez droit sur *dataBases* puis *create* puis *Database* comme indiqué ci-dessous :



donnez un nom à votre base et sauvegardez.

- Création de la table editeur.

Déroulez la base que vous venez de créer, puis déroulez l'onglet *Schemas*, puis cliquez droit sur *Tables*, puis *Create* puis *Table*, comme indiqué ci-dessous :



donnez le nom à la table, puis passez sur l'onglet columns

ajoutez des attributs en cliquant sur le « + » à droite, précisez les noms de vos attributs, leurs types et contraintes de clé primaires et non nullité comme indiqué ci-dessous, puis sauvegardez.

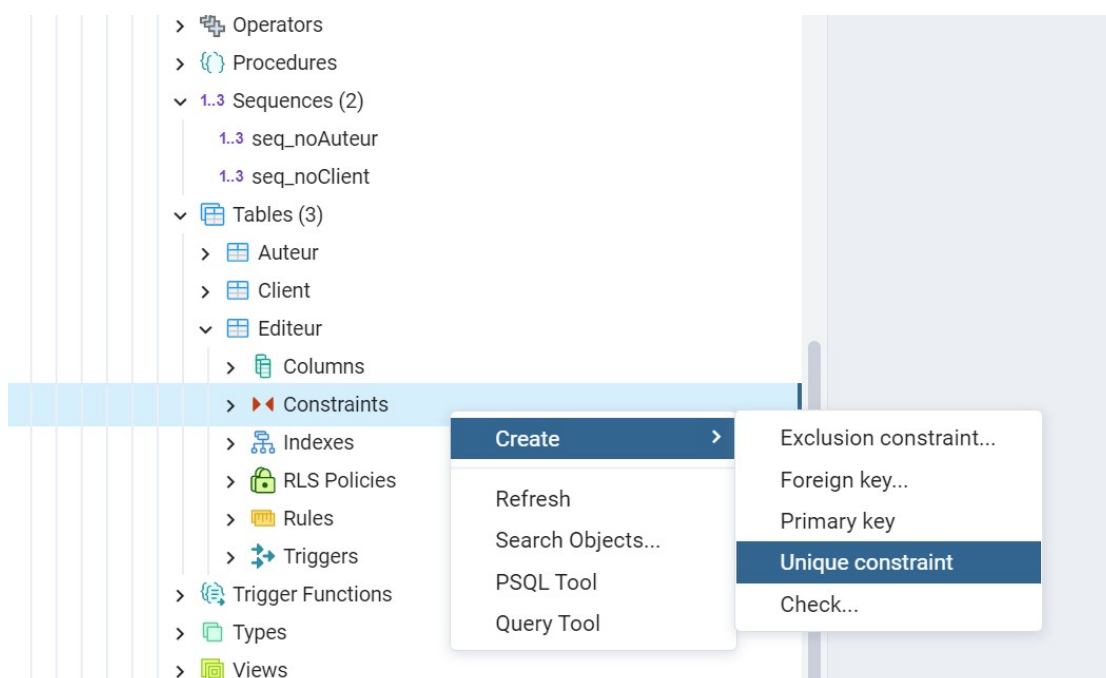
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	nomEditeur	character	40		No	Yes
	adresse	character	60		Yes	No

Il faut cliquer droit sur la table editeur puis *refresh* pour voir les colonnes créées.

- Faites de même pour les tables auteur, client, livre, exemplaire, contrat et a_ecrit.

N'utilisez pas l'onglet constraint du menu create table pour ajouter des contraintes sur les attributs :

Une fois la table créée, pour ajouter une contrainte d'unicité, de clé étrangère ou de clé primaire multi attribut , utilisez l'onglet *Constraint* de cette table.



Au fur et à mesure de la création des tables, ajouter des données en testant si les contraintes mises en place sont bien respectées.

Pour ajouter une ligne à une table : sélectionnez la table, cliquez droit puis choisir View/Edit data. Ensuite on obtient l'écran ci-dessous :

The screenshot shows a database management tool interface. At the top, there are tabs for 'Data output', 'Messages', and 'Notifications'. Below these is a toolbar with icons for adding, deleting, and other data operations. The main area displays a table with the following columns: 'no_auteur' (integer, primary key), 'nom_auteur' (character 40), 'prenom_auteur' (character 40), and 'email_auteur' (character 60). The table contains one row of data: '1', 'Hugo', 'Victor', and 'victor.hogo@...'.

	no_auteur [PK] integer	nom_auteur character (40)	prenom_auteur character (40)	email_auteur character (60)
1	1	Hugo	Victor	victor.hogo@...

Vous pouvez :

Cliquer sur le bouton + qui fait apparaître une ligne, entrer les données, puis valider avec l'avant dernier bouton.

Mais il vaut mieux utiliser la fenêtre sql puis insérer vos t-uples avec le code sql adéquat : les contraintes seront mieux prises en compte.

Auto incrémentation d'une clé primaire :

Pour créer une séquence sur un attribut d'une table : utilisez l'onglet *Sequences* de votre base (voir image ci-dessus).

puis sauvegardez.

Vous pouvez ensuite vérifier dans l'onglet *Sequence* que cette séquence a bien été créée.

Ou bien :

Si vous souhaitez qu'un champ s'auto incrémente (comme l'utilisation d'une séquence le permet), il faut ajouter une contrainte de type Identity pour l'attribut concerné. Voir ci-dessous :

noAuteur

General Definition **Constraints** Variables Security SQL

Default

Not NULL? ☒ Yes

Type ☐ NONE ☒ IDENTITY ☐ GENERATED

Identity ALWAYS

Increment 1

Start 1

Minimum 1

Maximum 2147483647

Cache 1

Cycled ☐ No

i ?

Pour pouvoir gérer la contrainte qu'un type de livre est soit POCHE soit BD, il faut utiliser un type. Pour cela créer un nouveau type (typelivre) qui est une énumération des valeurs POCHE et BD .

Utiliser ensuite ce type pour l'attribut type_livre de la table livre.

Ou bien, (cela fonctionne mieux) dans le menu de création de contrainte, créer une contrainte check et dans l'onglet définition : typelivre::text = 'BD'::text OR typelivre::text = 'POCHE'::text

Pour que la date d'achat d'un exemplaire par défaut soit la date du jour, mettre now() pour la valeur par défaut.

Question 3 : les vues

Il y a des contraintes liées aux cardinalités du diagramme E/A non prises en compte dans le script de création des tables. En utilisant l'interface, créez les vues permettant de vérifier ces contraintes.

Question 4 : création d'une fonction

Créez une fonction permettant d'obtenir tous les exemplaires d'un livre.
Testez-là.

Pour créer un trigger :

**tout d'abord réaliser la fonction correspondante au trigger à l'aide de l'onglet *Trigger Functions* ,
puis vous cliquez droit sur la table sur lequel porte le trigger, puis onglet *create*,
puis *trigger* vous utilisez ensuite le menu associé.**

Question 5 : Trigger

Créez un trigger permettant de faire en sorte que le prix d'un livre puisse pas pouvoir être augmenté de plus de 10 pour cent.
Testez.

Question 6 : Trigger

Créez un trigger permettant de faire en sorte qu'un auteur ne doit pas porter le même nom que son éditeur. Testez.

Question 7 : Trigger

Créez un trigger permettant de faire en sorte qu'un livre écrit par un auteur ne peut être édité que par un éditeur chez qui l'auteur a un contrat.
Testez.

Question 8 : création d'une dernière fonction (si vous avez le temps)

Créez une fonction permettant de connaître le nom de l'auteur préféré d'un client.
Testez-là.

À la fin de la séance sauvegardez votre BDD, vous pourrez ainsi la reprendre lors de la deuxième séance.

Sauvegarde : cliquez droit sur votre base, puis « backup » choisir l'endroit de votre sauvegarde et un nom de fichier.

Import d'une Base sauvegardée : Le fichier que vous avez sauvegardé n'a pas d'extension, renommez-le en rajoutant l'extension .sql, puis créez une base puis cliquez droit dessus et choisissez « restore » et sélectionnez le fichier en question.