Exercices vintages 💾

Introduction

Lors du premier TP, nous avons utilisé DuckDB pour lire les données d'un fichier parquet. Aujourd'hui nous allons créer notre propre base de données PostgreSQL, puis y déclarer des tables et insérer de données. Enfin nous allons requêter ces données.

Lors de ce TP, vous allez :

- Créer votre base de données PostgreSQL
- Lancer des scripts de création de tables et d'insertion de données
- Relier des tables en utilisant les jointures
- Sélectionner et filtrer des données

> ! Important
>
> - Écrivez des requêtes jolies !

1. Lancement des services

- ☐ Connectez-vous au SSP Cloud
- ☐ Lancez un service **PostgreSQL**
- ☐ Lancez un service **cloudBeaver**
  - ouvrez ce service et ouvrez la connexion PostgreSQL

En cas de besoin, vous trouverez plus d'informations dans l'onglet Datalab.

À la fin du TP

N'oubliez pas de sauvegarder votre travail et libérer les ressources :

- ☐ Sauvegardez votre travail
  - Copiez-collez vos scripts SQL sur votre machine
- ☐ Mettez en pause ou Arrêter les services du Datalab

En attendant le lancement des services, voici un peu de lecture ci-dessous.

2. Méthodo : Écrire une requête

Il est parfois difficile d'écrire du premier coup LA requête qui répond directement à la question posée.

Voici une méthode qui peut vous aider à écrire vos requêtes pas à pas :

Imaginons que l'on vous demande de lister les noms et prénoms des joueuses du club de la ville de *Saint Quentin*, dont le prénom commence par un *A* classées par elo décroissant.

joueuse(id_joueuse, nom, prenom, elo, #id_club)
club(id_club, nom, ville)

2.1. Listez les tables nécessaires

Commençons par lister les tables nécessaires, puis joignons-les.

Nous avons besoin ici des 2 tables que nous pouvons joindre en utilisant la colonne *id_club*.

```
SELECT *
  FROM joueuse j
  JOIN club c USING(id_club);
```

> Tip
>
> - Nous utilisons pour l'instant `SELECT *` pour sélectionner toutes les colonnes
>   - nous enleverons les colonnes inutiles plus tard
> - Nous utilisons des alias, cela rendra la requête plus courte et plus claire
>   - j pour joueuse
>   - c pour club

2.2. Appliquez les filtres

Filtrons pour ne conserver que les lignes qui nous intéressent.

- Ville du club : *Saint Quentin*
- Prénom qui commence par un *A*

```
SELECT *
  FROM joueuse j
  JOIN club c USING(id_club)
 WHERE c.ville = 'Saint Quentin'
   AND j.prenom LIKE 'A%';
```

2.3. Ordonnez les résultats

Ordonnons par elo décroissant.

```
SELECT *
  FROM joueuse j
  JOIN club c USING(id_club)
 WHERE c.ville = 'Saint Quentin'
   AND j.prenom LIKE 'A%'
 ORDER BY j.elo DESC;
```

2.4. Sélectionner les colonnes requises

Il est demandé de ne garder que les noms et prénoms des joueuses.

Remplaçons le *SELECT ** par les noms des colonnes.

```
SELECT j.nom,
       j.prenom
  FROM joueuse j
```

```
  JOIN club c USING(id_club)
WHERE c.ville = 'Saint Quentin'
  AND j.prenom LIKE 'A%'
ORDER BY j.elo DESC;
```

Maintenant, place à quelques exercices historiques.

3. Exercices

3.1. École de musique 🎸

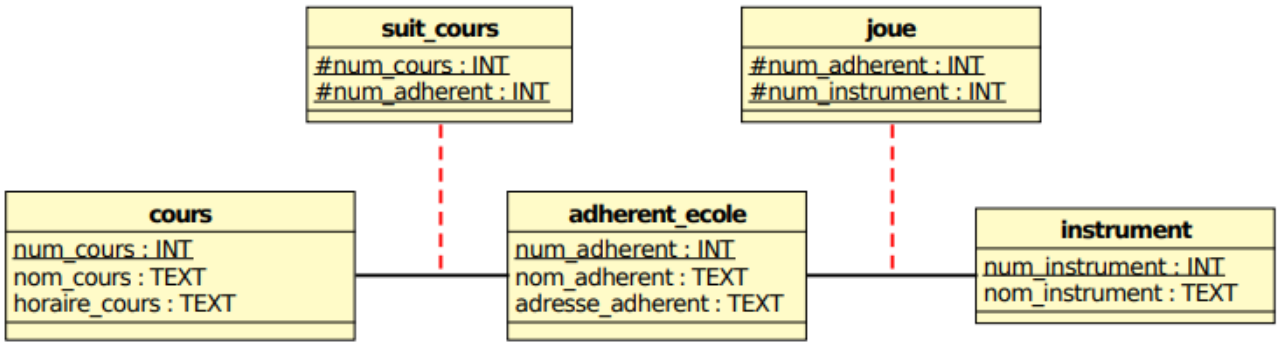
Nous allons commencer par créer les tables et insérer les données :

- ☐ Copiez le contenu de ce script sql
- ☐ Collez dans la fenêtre SQL de CloudBeaver
- ☐ Exécutez le script
  - Cliquez sur la petite icone sous les triangles oranges qui ressemble à 📜
  - raccourci (ALT + X)
  - En cas d'erreur, corrigez, puis relancez intégralement le script
- ☐ Vérifiez que votre schéma et vos tables sont créés
  - Dans l'explorateur : PostgreSQL ➡️ Databases ➡️ defaultdb ➡️ Schemas
  - Clic droit sur *Schemas* ➡️ Refresh

Si tout est ok, vous n'avez plus besoin du script de création et d'insertion de données.

- ☐ Supprimez tout le contenu du code SQL que vous venez d'exécuter ou ouvrez un nouvel éditeur SQL

Voici le modèle de données :



Vous pouvez maintenant commencer l'exercice et écrire les requêtes permettant de répondre aux questions suivantes :

- ☐ Quels sont les noms des instruments présents dans la base de données
- ☐ Quels sont les noms et adresses des adhérents
- ☐ Quelle est l'adresse de l'adhérent numéro 15
- ☐ Quels sont les numéros des adhérents de nom *DUPONT*

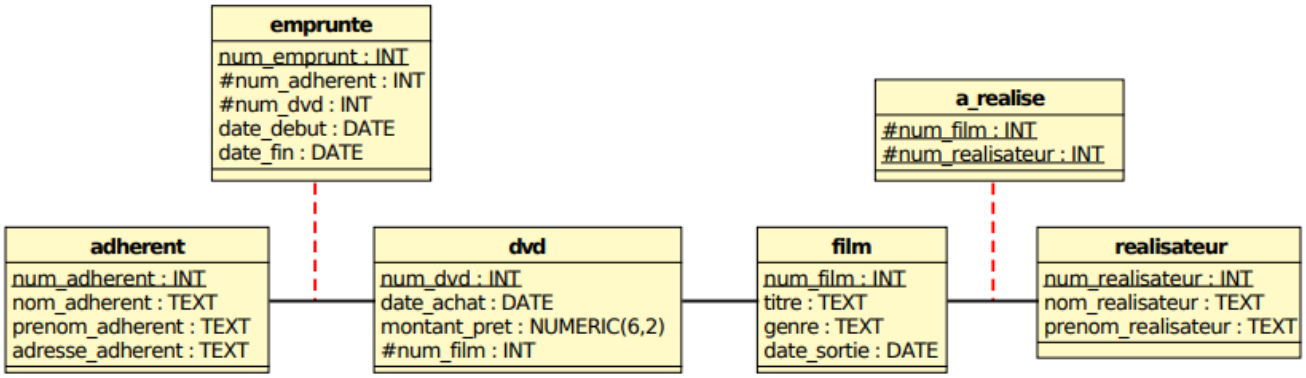
Réalisez les jointures qui permettent d'obtenir :

- ☐ Les noms des instruments dont joue chaque adhérent
- ☐ Le nom des adhérents inscrits au cours numéro 6
- ☐ Les noms des cours ayant lieu le samedi et les noms des adhérents inscrits
- ☐ Les noms des cours auxquels sont inscrits les adhérents qui jouent du piano
- ☐ De quels instruments jouent les personnes inscrites au cours de guitare classique (*GUIT-CLASS*)

3.2. Location de DVD 💿

Il fut un temps où des entreprises se spécialisaient dans la location de DVD. Dans cet exercice, une telle entreprise souhaite informatiser la gestion des prêts.

Voici le Modèle de données :



3.2.a. Création de tables

- ☐ Utiliser le code ci-dessous pour créer le schéma et les 3 tables de droite

```
DROP SCHEMA IF EXISTS dvd CASCADE;

CREATE SCHEMA dvd;

CREATE TABLE dvd.realisateur(
    num_realisateur    INT PRIMARY KEY,
    nom_realisateur    TEXT,
    prenom_realisateur TEXT);

CREATE TABLE dvd.film(
    num_film       INT PRIMARY KEY,
    titre          TEXT,
    genre          TEXT,
    date_sortie    DATE);

CREATE TABLE dvd.a_realise(
    num_film         INT REFERENCES dvd.film(num_film),
    num_realisateur  INT REFERENCES dvd.realisateur(num_realisateur),
    PRIMARY KEY (num_film, num_realisateur));
```

- ☐ Écrivez les requêtes qui permettent de créer les 3 autres tables

- ☐ Chargez les données avec ce script sql

> **Tip**
>
> Attention, ici l'ordre de création des tables a son importance.
>
> Reprenons et modifions l'exemple ci-dessus :
>
> - imaginez que vous essayez de créer en premier la table `a_realise`
> - le champ *num_film* de cette table est une clé étrangère
> - il référence la clé primaire *num_film* de la table `film`
> - or si la table `film` n'existe pas, à votre avis, que va-t-il se passer ?

3.2.b. Requêtes

Donnez les requêtes pour obtenir :

- ☐ Nom et prénom des adhérents
- ☐ Titre des films de la base
- ☐ Numéro d'adhérent de *Pierre DUCHEMIN*
- ☐ Nom, Prénom et adresse de tous les *DUCHEMIN*
- ☐ Numéro des DVD du film *la guerre des etoiles*

> **Recherche dans la doc**
>
> - ☐ Recherchez dans la documentation officielle PostgreSQL, un moyen de répondre au besoin ci-dessous.
>
> Pour connaitre la version utilisée : `SELECT version();`

Pour les 3 requêtes suivantes, vous afficherez les noms et prénoms sur une seule colonne.

Utilisez le mot clé `AS` pour nommer cette colonne *nom_complet* à l'affichage.

- ☐ Nom, Prénom des adhérents ayant emprunté le film *la guerre des etoiles*
- ☐ Nom, Prénom des adhérents ayant au moins un emprunt en cours (la date de fin de l'emprunt est nulle)
- ☐ Nom, Prénom du réalisateur de la *Guerre des étoiles*

> **Tip**
>
> Il existe différentes manières de concaténer du texte :
>
> - fonctions PostgreSQL *CONCAT()*, *CONCAT_WS()*
> - opérateur `||`

- ☐ Titre des films empruntés par Pierre DUCHEMIN, ainsi que dates de début et de fin
- ☐ Nom des réalisateurs des films empruntés par Pierre DUCHEMIN
- ☐ Nom des réalisateurs de films de science fiction ou policiers

Utilisons maintenant des méthodes d'agrégation pour répondre aux questions suivantes :

- ☐ Nombre de DVD dont le coût de l'emprunt est inférieur ou égal à 2€
- ☐ Nombre d'emprunts ayant été effectués par Dominique Duchemin
- ☐ Nombre de films différents ayant été empruntés par Dominique Duchemin
- ☐ Numéro des DVD en ordre décroissant du nombre de fois qu'ils ont été empruntés
- ☐ Numéro du DVD le plus récemment acheté

Quelques questions sur les dates :

- ☐ Quelle a été la durée d'emprunt la plus longue
- ☐ En date du 10 août 2010, combien de DVD sont en cours d'emprunt
- ☐ Combien de dvd ont été achetés chaque année

Pour terminer cet exercice, nous allons rechercher les titres des films qui n'ont jamais été empruntés. Procédons par étapes et écrivez les requêtes suivantes :

- ☐ Différents numéros des films empruntés
- ☐ Titre des films ayant pour numéro 18, 19 ou 20
- ☐ Titre des films n'ayant pas pour numéros ni 18, ni 19, ni 20
- ☐ Titre des films n'ayant jamais été empruntés

3.3. Rugby World Cup 🏉

> Pour les plus rapides 🚀

Nous souhaitons informatiser les résultats de la coupe du monde de rugby. Pour cela nous avons créé les deux tables suivantes : *equipe* et *matches*.

Commencez par créez et charger les tables avec ce script.

- ☐ Donner les noms des équipes qui ont joué à Nantes
- ☐ Contre quelles équipes a joué l'équipe de France ?
- ☐ Pour tous les matchs, affichez les informations suivantes :
  - Date du match
  - Nom de la première équipe
  - nombre de points de la première équipe
  - nombre de points de la deuxième équipe
  - nom de la deuxième équipe et lieu du match
- ☐ Quels sont les noms d'équipes qui ont gagné au moins deux matchs
- ☐ Quels sont les noms des équipes qui ont perdu tous leurs matchs
- ☐ Pour chaque équipe donner le nombre de matchs qu'elles ont gagné ou perdu
- ☐ Donner le nombre total de points obtenus par équipe en classant les équipes de celle ayant marqué le plus à celle ayant marqué le moins

> Solutions
>
> - Correction TP2