Manipuler les données

Introduction au SQL

CREATE & SELECT

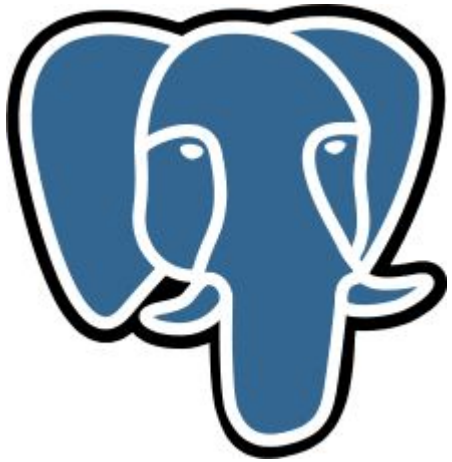
Objectifs

Apprendre le SQL

- Créer une base de données
- Créer une table
- Effectuer des requêtes simples

PostgreSQL

Postgresql est le SGBDR retenu pour la formation et aussi celui qui sera utilisé pour le projet.



On utilise une interface Pgadmin pour simplifier l’accès à la base.

On utilise pour communiquer avec la base de données le langage SQL

SQL – Structured Query Language

DDL – Data Definition Language

-> ordres utilisés pour créer, modifier ou supprimer les structures de la base

CREATE, ALTER, DROP

DML – Data Manipulation Language

-> ordres utilisés pour manipuler les données contenues dans la base

SELECT, INSERT, UPDATE, DELETE

DCL – Data Control Language

-> ordres utilisés pour gérer la sécurité de l'accès aux données

GRANT, REVOKE

SQL – MERISE -> Structure

- Le modèle est la Base de Données
- Les entités vont devenir des tables
- Les propriétés des entités vont devenir des champs

SQL – Comment ça marche ?

Le SQL est un language.

On communique avec la base de données grâce à des instructions

Une instruction est composé de mots séparé par des espaces

- des mots clé du langage, qu'on écrira toujours en majuscule
- des valeurs qui seront propres à notre modèle

Une instruction pour être exécutée, doit se terminer par un point-virgule

exemple: MOT_CLE valeur;

SQL - Base de données

La base de donnée est la structure qui va permettre le stockage de nos tables.

Elle possède un nom, qui correspond à ce qui sera stocké, notre application/projet.

Exemple:

CREATE DATABASE forall;

SQL – Les tables - creation

La table est la structure qui va permettre le stockage d'une partie de nos données.

On regroupe alors des données de manière logique et on nomme alors la table en conséquence.

On décrit ensuite les données qui y seront stockées, i.e les champs, qui seront alors typés ainsi que les contraintes (nullité, unicité, clé).

Exemple:

```
CREATE TABLE personne (
    numero serial NOT NULL,
    prenom character varying(30) NOT NULL,
    nom character varying(50) NOT NULL,
    PRIMARY KEY (numero)
);
```

SQL – Les tables – les types de données

- Correspondance entre les type de données du dictionnaire et le SGBDR choisi

- On retrouve les types numerique, alphanumerique, alphabetique, date et logique ainsi que d'autres types

- Pour postgreSQL, la liste des types
  https://docs.postgresql.fr/11/datatype.html

SQL – Les tables – les contraintes

Nullité

```
CREATE TABLE produits (
    no_produit integer NOT NULL,
    nom text NOT NULL,
    prix numeric
);
```

Clé (UNIQUE + NOT NULL)

```
CREATE TABLE produits (
        no_produit integer PRIMARY KEY,
        nom text,
        prix numeric
    );
```

```
CREATE TABLE exemple (
        a integer,
        b integer,
        c integer,
        PRIMARY KEY (a, c)
    );
```

```
CREATE TABLE commandes (
        id_commande integer PRIMARY KEY,
        no_produit integer REFERENCES produits,
        quantite integer
    );
```

```
CREATE TABLE t1 (
        a integer PRIMARY KEY,
        b integer,
        c integer,
        FOREIGN KEY (b, c) REFERENCES autre_table (c1, c2)
    );
```

SQL – Les tables – les contraintes

Unicité

```
CREATE TABLE produits (
    no_produit integer UNIQUE,
    nom text,
    prix numeric
);
```

Validation

```
CREATE TABLE produits (
    no_produit integer,
    nom text,
    prix numeric CHECK (prix > 0)
);
```

```
CREATE TABLE produits (
    no_produit integer,
    nom text,
    prix numeric,
    CHECK (prix > 0)
);
```

```
CREATE TABLE produits (
    no_produit integer,
    nom text,
    prix numeric,
    CHECK (prix > 0 AND prix <100)
);
```

SQL – Les tables – Mise à jour

Il est possible de mettre à jour la structure d'une table ou d'ajouter des contraintes.

Il faut surtout veiller à ce que les données déjà présente respecte déjà la contrainte (parfois il faudra faire la modification en plusieurs étapes – ajout d'un champs, saisie de la valeur puis ajout de la contrainte.

Exemple:

ALTER TABLE personne ADD COLUMN age smallint;

ALTER TABLE personne ALTER COLUMN age SET NOT NULL;

ALTER TABLE personne DROP COLUMN age CASCADE;

ALTER TABLE produits ADD CHECK (nom <> '');

ALTER TABLE produits ADD CONSTRAINT autre_nom UNIQUE (no_produit);

ALTER TABLE produits ADD PRIMARY KEY(no_produit);

ALTER TABLE produits ADD FOREIGN KEY (id_groupe_produit) REFERENCES groupes_produits;

ALTER TABLE produits ALTER COLUMN prix SET DEFAULT 7.77;

ALTER TABLE produits ALTER COLUMN prix TYPE numeric(10,2);

ALTER TABLE produits RENAME COLUMN no_produit TO numero_produit;

…

SQL – Les tables - Suppression

- Il est possible de supprimer l'intégralité d'une table.

- DROP TABLE personne;

SQL – Ajout, mise a jour et suppression de données

- Insérer des données dans une table

  INSERT INTO personne VALUES (1, ‘Aurore’, ‘Mozdzierz’);

  INSERT INTO personne(prenom, nom) VALUES (‘Aurore’, ‘Mozdzierz’);
- Mettre à jour des données

  UPDATE personne SET prenom = ‘AURORE’ WHERE nom = ‘Mozdzierz’;

  UPDATE produits SET prix = prix * 1.10;
- Supprimer des données

  DELETE FROM personne WHERE nom = ‘Mozdzierz’;

/!\ Toujours faire un SELECT avant d’executer un UPDATE ou un DELETE

SQL – Ajout, mise a jour et suppression de données - RETURNING

Il est possible sur ces action d'ajouter la clause RETURNING afin de voir ce qui a été modifié, mis à jour ou supprimé

INSERT INTO utilisateurs (prenom, nom) VALUES ('Joe', 'Cool') ***RETURNING id****;*

*UPDATE produits SET prix = prix * 1.10 WHERE prix <= 99.99*

RETURNING nom, prix AS nouveau_prix*;*

DELETE FROM produits WHERE date_perime = 'today' ***RETURNING ****;*

SQL – Requête simple

- Remonter toutes les données d'une table

  SELECT * FROM personne;

- Remonter une partie des données d'une table

  SELECT nom FROM personne;

- Il est alors possible d'ajouter des conditions grâce à la clause WHERE

  SELECT * FROM personne WHERE prenom = 'Aurore'

  = < > like … count… distinct… order by… limit … group by

Pour résumer

- Utilisation de PostgreSQL comme SGBD et pgadmin comme interface pour accéder à PostgreSQL

- Le SQL est le langage de communication avec la base de données

    - Créations, modifications et suppressions de la structure de la base
    - Ajouts, mises à jour et suppressions de données
    - Recherches dans la base de données