

Rapport Admin Base de Donnée

Script d'initialisation de la base donnée :

1. On commence par créer la base de données `school`, on configure l'encodage UTF-8 pour le stockage des caractères avant de s'y connecter.

```
-- Create the database
CREATE DATABASE school
ENCODING 'UTF8';
\c school;
```

2. On créer la table `classes` si elle n'existe pas déjà (au cas où) avec deux colonnes :
 - `id` (clé primaire auto-incrémentée)
 - `class_name` (nom de la classe)

```
-- Create the classes table
CREATE TABLE IF NOT EXISTS classes (
  id SERIAL PRIMARY KEY,
  class_name VARCHAR(255)
);
```

3. On créer la table `students` si elle n'existe pas déjà (au cas où) avec quatre colonnes :
 - `id` (clé primaire auto-incrémentée)
 - `lastname` (nom de famille)
 - `firstname` (prénom)
 - `class_id` (référence à la table `classes`)

```
-- Create the students table
CREATE TABLE IF NOT EXISTS students (
  id SERIAL PRIMARY KEY,
  lastname VARCHAR(255),
  firstname VARCHAR(255),
  class_id INT,
  FOREIGN KEY (class_id) REFERENCES classes(id)
);
```

4. On insère ensuite deux enregistrements dans la table `classes` avec les noms de classe `s6_alt1` et `s6_alt2`.

```
-- Insert data into the classes table
INSERT INTO classes (class_name)
VALUES
('s6_alt1'),
('s6_alt2');
```

5. Enfin, on insère des enregistrements dans la table `students` avec des noms de famille, prénoms et identifiants de classe correspondants

```
-- Insert data into the students table
INSERT INTO students (lastname, firstname, class_id)
VALUES
('BLONDIEAU', 'GABIN', 1),
('CHU', 'THOMAS', 1),
('FERRY', 'LUCAS', 1),
('JOBST', 'GAETAN', 1),
('KORBI', 'ARGJENTIN', 1),
('LAGARDE', 'LOANN', 1),
('MAGNIN', 'TITOUAN', 1),
('MOUGENOT', 'VALENTIN', 1),
('PAULIN', 'MAXENCE', 1),
('RIAT', 'FELIX', 1),
('GUILLOU', 'AURELIEN', 1),
('CHEVALME', 'ARNAUD', 1),
('FOULON', 'HUGO', 1),
('BECHER', 'THOMAS', 2),
('CRAYSSAC', 'MAXIME', 2),
('FRECHIN', 'MAXENCE', 2),
('GRABER', 'SIMON', 2),
('KOHLE', 'JAROD', 2),
('LAVAL', 'BAPTISTE', 2),
('MEISTER', 'NICOLAS', 2),
('MOUMEN', 'TAHA', 2);
```

Script de gestion des droits :

1. Cette commande retire tous les privilèges accordés par défaut à tous les utilisateurs sur la base de données `school`.

```
REVOKE ALL ON DATABASE school FROM PUBLIC;
```

2. On crée trois utilisateurs avec des mots de passe pour la démonstration.

```
CREATE USER spectator WITH PASSWORD 'password';  
CREATE USER owner_user WITH PASSWORD 'password';  
CREATE USER admin_user WITH PASSWORD 'password';
```

3. On permet à `spectator` de se connecter à la base de données `school`, d'utiliser le schéma `public`, et de lire les données dans toutes les tables du schéma `public`.

```
GRANT CONNECT ON DATABASE school TO spectator;  
GRANT USAGE ON SCHEMA public TO spectator;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO spectator;
```

4. On permet à `owner_user` de se connecter à la base de données `school`, d'utiliser le schéma `public`, et de sélectionner, insérer, mettre à jour, et supprimer des données dans toutes les tables du schéma `public`, ainsi que d'utiliser et de mettre à jour toutes les séquences du schéma `public`.

```
GRANT CONNECT ON DATABASE school TO owner_user;  
GRANT USAGE ON SCHEMA public TO owner_user;  
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO owner_user;  
GRANT USAGE, SELECT, UPDATE ON ALL SEQUENCES IN SCHEMA public TO owner_user;
```

5. On attribue le rôle de propriétaire de la base de données `school` et aux tables `classes` et `students` à `owner_user`.

```
ALTER DATABASE school OWNER TO owner_user;  
ALTER TABLE classes OWNER TO owner_user;  
ALTER TABLE students OWNER TO owner_user;
```

6. Pour finir, on donne le statut de superutilisateur à `admin_user`, lui permettant d'avoir tous les privilèges sur le serveur de base de données.

```
ALTER USER admin_user WITH SUPERUSER;
```

Vérification des droits

On peut se connecter à notre pas en tant que postgres pour faire les vérifications avec :

```
psql -h localhost -U postgres -d school
```

On peut ensuite utiliser \du pour lister les rôles des utilisateurs. On voit bien que l'utilisateur "admin_user" est bien un superuser.

List of roles	
Role name	Attributes
admin_user	Superuser
owner_user	
postgres	Superuser, Create role, Create DB, Replication, Bypass RLS
spectator	

La commande \dp nous permet de vérifier les droits d'accès sur les tables.

Access privileges					
Schema	Name	Type	Access privileges	Column privileges	Policies
public	classes	table	owner_user=arwdDxt/owner_user+ spectator=r/owner_user		
public	classes_id_seq	sequence	owner_user=rwU/owner_user		
public	students	table	owner_user=arwdDxt/owner_user+ spectator=r/owner_user		
public	students_id_seq	sequence	owner_user=rwU/owner_user		
(4 rows)					

On peut y voir que l'utilisateur "owner_user" a les droit "arwdDxt":

- a: INSERT (Ajouter des données)
- r: SELECT (Lire les données)
- w: UPDATE (Modifier les données)
- d: DELETE (Supprimer des données)
- D: TRUNCATE (Vider une table)
- x: REFERENCES (Créer une clé étrangère)
- t: TRIGGER (Créer des déclencheurs)

Il es aussi propriétaire de l'objet (indiqué par le symbol "+").

À l'inverse, l'utilisateur "spectator" n'a que le droit de lecture ("r").

On peut aussi vérifier que “owner_user” est bien propriétaire des données avec la commande \dt.

List of relations			
Schema	Name	Type	Owner
public	classes	table	owner_user
public	students	table	owner_user
(2 rows)			

Backup/Restore

On peut faire un backup avec la commande : `pg_dump -h host -U user -d database > output_file`

L'utilisateur “spectator” n’a pas le droit de faire de backup

```
> pg_dump -h localhost -U spectator -d school
Password:
pg_dump: error: query failed: ERROR: permission denied for sequence classes_id_seq
pg_dump: detail: Query was: SELECT last_value, is_called FROM public.classes_id_seq
```

Les utilisateurs “owner_user” et “admin_user” peuvent faire le backup.

```
> pg_dump -h localhost -U owner_user -d school > db.bak
> pg_dump -h localhost -U admin_user -d school > db.bak
```

On peut ensuite restaurer les données dans une nouvelle base avec la commande :

```
cat db.bak | psql -h localhost -U postgres -d school2
```