

Secteur Tertiaire Informatique
Filière « Etude et développement »

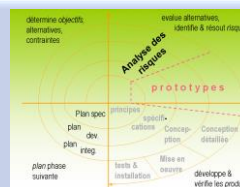
TP – Vues SQL et accès restreint à la BDD

Persistence de données

Apprentissage

Mise en situation

Evaluation



1. PREAMBULE

Vous allez découvrir en suivant ce TP différents concepts de base de données :

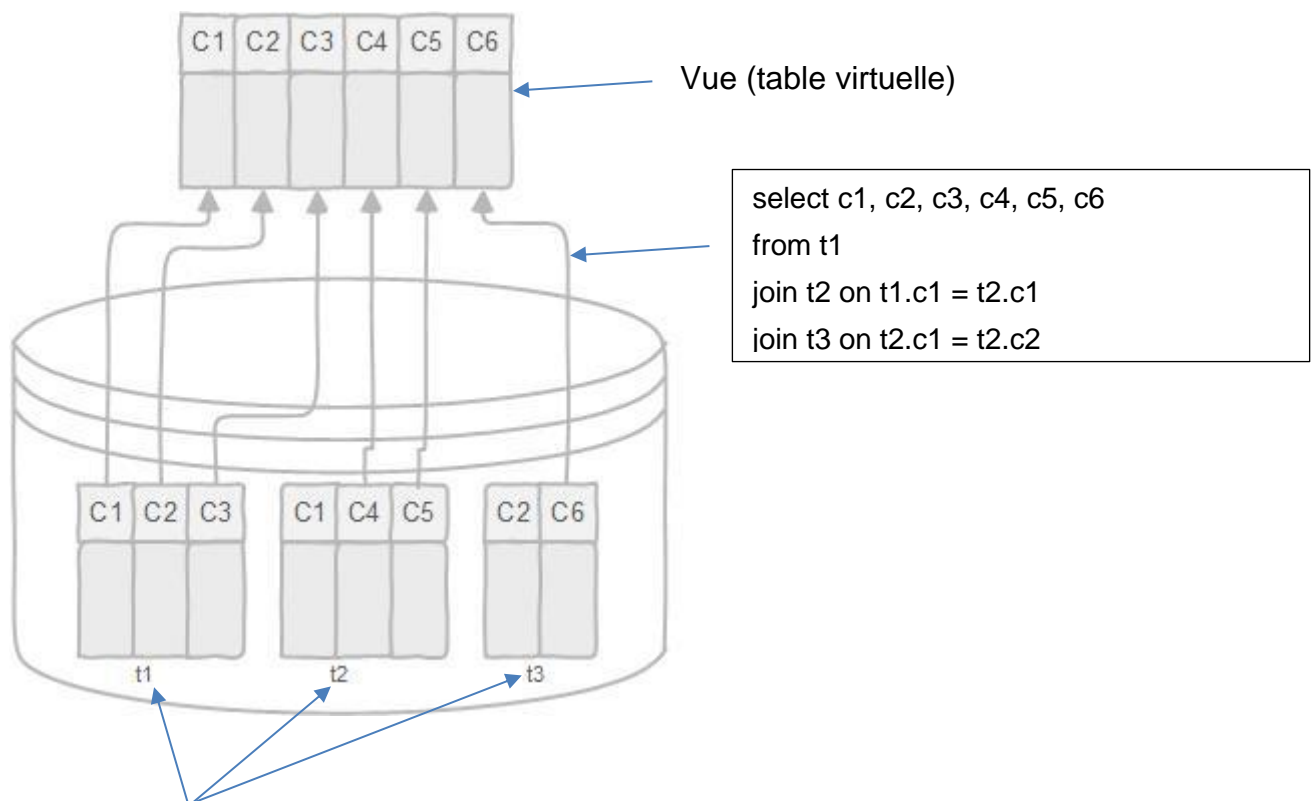
- La création et l'utilisation de **vues** ;
- La création d'**utilisateurs de base de données** pour la **restriction d'accès**.

2. LES VUES

2.1.1 Généralités

Une **vue dans une base de données** est une façon de créer un **sous-ensemble de tables** qui sont stockées logiquement dans une base de données et accessible par un utilisateur SQL.

Elle permet de **restreindre l'accès à la base de données** en ne **présentant qu'une sous-partie de la base**.



Tables « physiques » de la base de données

La vue peut être **requêtée** comme n'importe quelle table.

2.2 AVANTAGES

Les avantages des vues sont multiples :

Persistance de données

- **Simplicité de requêtage :**
 - Une vue permet d'éviter la saisie fastidieuse de requêtes complexes (la construction de la requête est effectuée à la création de la vue).
- **Augmentation de la sécurité**
 - Masque le schéma de la base de données à l'utilisateur (seul le nom de la vue et les colonnes sont accessibles, pas de connaissance des relations entre les tables).
 - Possibilité de régler les permissions sur les vues en fonction d'utilisateurs pour limiter l'accès à la base.
- **Augmentation de l'abstraction :**
 - Si la structure de base de données change et que la vue est mise à jour l'utilisation reste la même pour l'utilisateur.

2.3 PREMIERE VUE

2.3.1 Création d'une vue

Vous allez créer une première vue simple offrant la possibilité à l'utilisateur de récupérer les informations d'un hôtel et de ses stations.

Le résultat attendu de la vue est présenté par l'image suivante :

	123 hotel_id	123 station_id	abc Hotel name	123 category	abc address	abc city	abc Station name	123 altitude
1	1	1	Le Magnifique	3	rue du bas	Pralo	La Montagne	2 500
2	2	1	Hotel du haut	1	rue du haut	Pralo	La Montagne	2 500
3	3	2	Le Narval	3	place de la liberation	Vonten	Le Sud	200
4	4	2	Les Pissenlis	4	place du 14 juillet	Bretou	Le Sud	200
5	5	2	RR Hotel	5	place du bas	Bretou	Le Sud	200
6	6	2	La Brique	2	place du haut	Bretou	Le Sud	200
7	7	3	Le Beau Rivage	3	place du centre	Toras	La Plage	10

Le code générique à utiliser pour créer une vue est le suivant :

```
CREATE VIEW <nom-vue> AS <requête>;
```

Pour la première vue, vous pouvez utiliser le code suivant ;

```
CREATE OR REPLACE VIEW hotel_station
AS
SELECT h.id AS hotel_id,
       h.station_id,
       h.name as "Hotel name",
       h.category,
       h.address,
       h.city,
       s.name AS "Station name",
       s.altitude
FROM hotel h
JOIN station s ON h.station_id = s.id;
```

Une fois la vue créée vous pourrez la requêter comme table physique, par exemple en utilisant le code suivant :

```
select * from hotel_station;
```

2.3.2 Modification d'une vue

Pour **modifier une vue** il vous faut utiliser la commande **"ALTER VIEW"**. Pour plus d'informations sur "alter view" : <https://docs.postgresql.fr/13/sql-alterview.html>.

2.3.3 Suppression d'une vue

La requête suivante peut être utilisée pour supprimer une vue existante :

```
DROP VIEW [ IF EXISTS ] <nom-vue>;
```

2.4 CREATION DE VUES

A partir de la base « *hotel* », créez les vues suivantes :

1. **Vue 1 : Afficher la liste des réservations avec le nom des clients**
2. **Vue 2 : Afficher la liste des chambres avec le nom de l'hôtel et le nom de la station**
3. **Vue 3 : Afficher les réservations avec le nom du client et le nom de l'hôtel**

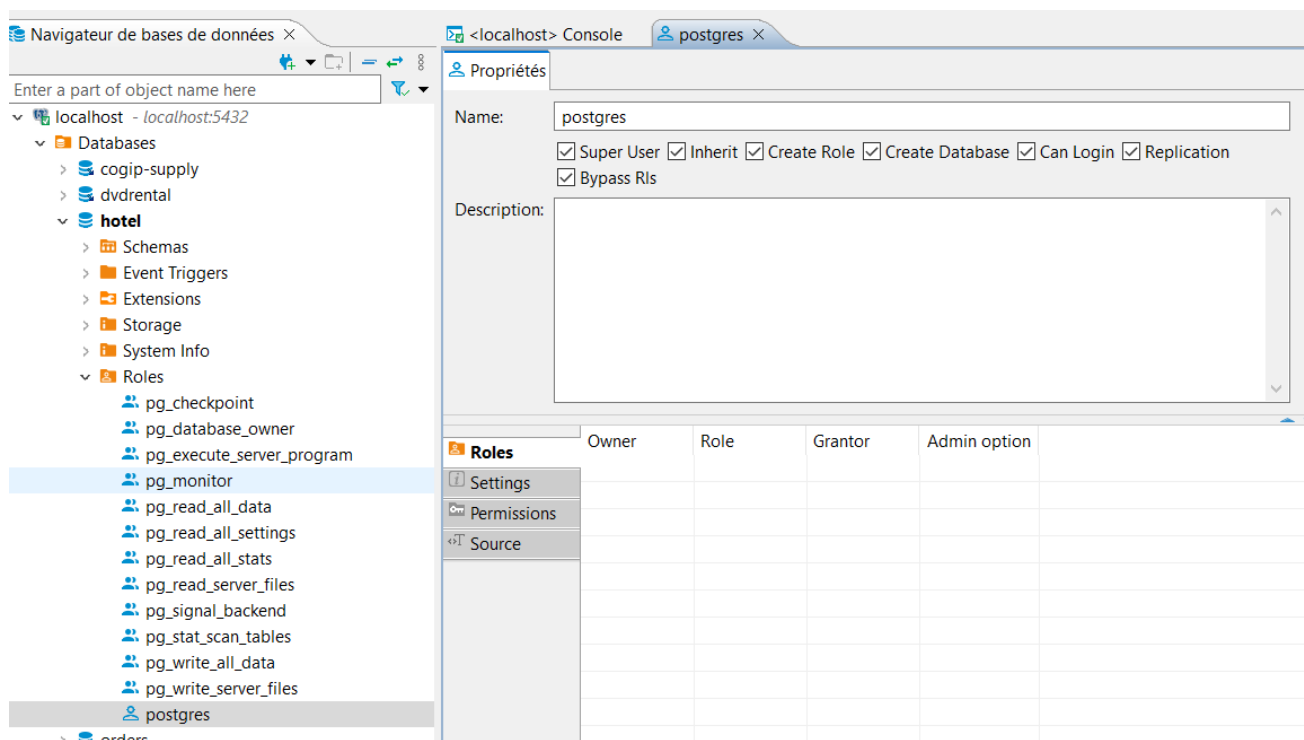
3. RESTRICTION DE L'ACCES A LA BDD

3.1 NOTION DE ROLE

Il est possible de créer des **rôles** (équivalent d'utilisateur) permettant de **se connecter** à un SGBD et de manipuler une BDD avec des **permissions spécifiques**.

Jusqu'à présent vous avez utilisé le rôle **« postgres »** défini en tant que **« super user »** ayant **tous les droits**.

Vous pourrez retrouver les rôles de votre SGBD sous DBeaver, comme présenté par l'image suivante :



Comme presque tout en base de données PostgreSQL, les rôles sont stockés dans une table particulière nommée « pg_roles ».

Ainsi, vous pouvez également lister tous les rôles en utilisant la requête suivante :

```
SELECT rolname FROM pg_roles;
```

La suite de ce document vous permettra de créer des rôles supplémentaires ayant des droits restreints.

3.2 CREATION DE ROLES

La création d'un rôle s'effectue avec une commande similaire à la suivante :

```
CREATE ROLE <name>
LOGIN
PASSWORD <mot-de-passe>;
```

LOGIN : permet de préciser que le rôle créé correspond à un utilisateur de BDD pouvant se connecter

PASSWORD : 'self explanatory, isn't it ?'

1. Créez un utilisateur « application_admin » pouvant se connecter au SBGD

Par défaut, un utilisateur nouvellement créé n'a aucun droit sur les bases de données du SGBD.

2. Ajoutez une connexion à la base de données « hotel » en utilisant votre nouvel utilisateur dans votre client SQL.

Persistance de données

3. Essayez d'effectuer un select sur une table, que se passe-t-il ?

Il va vous falloir créer des **permissions** (aussi appelées « **privileges** ») autorisant l'utilisateur à interagir avec la base de données.

3.3 AJOUT DE PRIVILEGES

```
GRANT <liste-privileges>  
ON <nom-table>  
TO <nom-role>;
```

Parmi les privilèges utilisables on retrouve :

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**
- **TRUNCATE**

Il est aussi possible d'**accorder tous les privilèges** en utilisant la requête :

```
GRANT ALL  
ON <nom-table>  
TO <nom-role>;
```

Si vous souhaitez **accorder des privilèges à l'ensemble des tables d'un schéma** (par exemple, « public »), il est possible d'utiliser la requête suivante :

```
GRANT <liste-privileges>  
ON ALL TABLES  
IN SCHEMA "public"  
TO <nom-utilisateur>;
```

Il vous est également possible de **supprimer des privilèges** en utilisant la commande « **REVOKE** » :

```
REVOKE <list-privileges> ALL  
ON TABLE <nom-table> | ALL TABLES IN SCHEMA <nom-schema>  
FROM <nom-role>;
```

L'alternative en impactant toutes les tables est possible

```
REVOKE <list-privileges>  
ON ALL TABLES IN SCHEMA <nom-schema>  
FROM <nom-role>;
```

Tout comme pour le « **GRANT** », la commande « **REVOKE ALL** » est possible.

Persistance de données

4. **Accordez les privilèges « SELECT », « INSERT », « UPDATE » et « DELETE » à l'utilisateur « application_admin » sur toutes les tables sauf « station ».**
5. **Essayez d'effectuer des requêtes.**

3.4 PRIVILEGES SUR UNE VUE

Il vous est possible d'accorder des privilèges sur les vues (table virtuelle) plutôt que les tables physiques de la base de données (cf. [la documentation de PostgreSQL](#)).

Vous pourrez ainsi finement paramétrer l'accès à votre base de données.

6. **Créez un nouveau rôle « application_client » pouvant se connecter à votre base de données.**
7. **Ajoutez les privilèges de lecture des données uniquement sur votre vue permettant de retrouver chambres avec le nom de l'hôtel et le nom de la station (vue 2).**
8. **Testez vos permissions.**

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

Michel Coulard – Formateur Evry

Chantal Perrachon – IF Neuilly sur Marne

Date de mise à jour : 12/06/2022

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Persistance de données