

TD Administration

Avec solutions

Avec solutions

- `r stringr::str_glue('{params$curriculum}')`
- `r stringr::str_glue('{params$university}')(https://www.u-paris.fr)')`
- `r stringr::str_glue("Année {params$year}-{params$year+1}")`
- `r stringr::str_glue("[Course Homepage] ({params$homepage})")`
- `r stringr::str_glue("[Moodle] ({params$moodle})")`



Administration sous Postgres

Des outils d'administration importants d'une base de données postgres se trouvent dans deux schémas appelés :

- `information_schema`
- `pg_catalog`

Ces deux schémas et les tables qu'ils contiennent permettent de retrouver toutes les informations concernant les schémas, tables, utilisateurs, fonctions, etc présentes dans la base de données.

Pour faciliter l'administration de la base et obtenir un certain nombre d'informations basiques, des commandes sont définies dans postgres. On peut en avoir la liste par l'instruction `\?` (vous connaissez déjà `\d`). Ces commandes sont, en fait, des raccourcis (des macros) en lieu et place de requêtes SQL plus ou moins complexes portant sur les objets des schémas `information_schema` et `pg_catalog`.

Les questions ci-dessous permettent de faire connaissance avec les schémas d'administration à travers quelques requêtes à générer.

- Dans `information_schema` la table `tables` contient le nom (et bien plus) des tables de la base. Examinez la structure de la table `tables` par l'instruction `\d information_schema.tables` (il faut au préalable fixer le `search_path` pour y inclure `information_schema`).

Lister juste les noms de toutes les tables.

```
select table_name
from information_schema.tables
```

- En utilisant seulement la table `tables` de `information_schema`, lister les noms des tables du schéma `world` puis du schéma `information_schema`

```
select table_name
from information_schema.tables
where table_schema='world';

select table_name
from information_schema.tables
where table_schema='information_schema'
```

- D'après la requête précédente, une des tables du schéma `information_schema` s'appelle `schemata`.

Examiner cette table et en déduire une liste de tous les schémas de la base

```
select schema_name
from information_schema.schemata
```

- Une des tables du schéma `information_schema` s'appelle `columns`. Essayez de comprendre la nature des informations présentes dans cette table. Écrire une requête qui renvoie les mêmes informations que l'instruction `\d world.country`. Améliorer votre requête pour qu'elle affiche exactement la même chose que `\d world.country`.

```
select column_name as Column, data_type as Type, is_nullable as Modifiers
from information_schema.columns
where table_name='country' and table_schema='world'
```

Alternativement

```
select column_name as Column, data_type as Type, (CASE
  WHEN is_nullable='NO'
  THEN 'not null'
  ELSE NULL
END) as Modifiers
  from information_schema.columns
 where table_name='country' and table_schema='world'
```

- Listez les noms de tables qui apparaissent dans au moins deux schémas. On donnera aussi les (deux) schémas concernés dans le résultat.

```
select t.table_name, q.table_schema, t.table_schema
from information_schema.tables t, information_schema.tables q
where t.table_schema <> q.table_schema and t.table_name=q.table_name;
```

- Listez les noms de colonnes (attributs) qui apparaissent dans au moins deux tables du même schéma. On donnera aussi le nom des tables et du schéma dans le résultat (on pourra exclure `information_schema` et `pg_catalog`).

```
select t1.column_name as Column, t1.table_name as table1,
t2.table_name as table2, t1.table_schema as schema
from information_schema.columns t1, information_schema.columns t2
where
t1.table_name <> t2.table_name and t1.table_schema=t2.table_schema and
t1.column_name=t2.column_name and t1.table_schema <> 'pg_catalog' and
t1.table_schema <> 'information_schema';
```

- Listez les définitions de toutes les vues que vous avez écrites. Listez celles qui commencent pas 'q'.

```
```{.sql}
```

```
select schemaname, definition from pg_views where schemaname=current_USER;
```

```
select schemaname, definition from pg_views where schemaname=current_USER and viewname like 'q%';
```

- La vue `pg_roles` contient les identifiants des différents rôles. Trouvez celui du rôle `isifar2022`.

```
select rolname,oid from pg_roles where rolname='isifar2022'
```

- À l'aide de la réponse à la question précédente, listez les noms des schémas des utilisateurs (rôles) qui ont aussi le rôle `isifar2022`

Attention, nécessite de changer les droits de consultation de ces tables pour les utilisateurs standards.

```
select schema_name from information_schema.schemata, pg_roles
where schemata.schema_name=pg_roles.rolname and
pg_roles.oid in (select member from pg_auth_members where roleid=78094)
```

## Droits

- Créer dans votre schéma une table `matable_etudiant` par copie de la table `datatpadmin` du schéma `durand`. Sélectionner les tuples de la table créée : il contiennent des informations (aléatoire) pour chacun des utilisateurs du groupe.
  - Donner les droits de sélection sur cette table à l'utilisateur de votre choix ou au groupe entier). Vérifier que cela marche.
  - Enlever les droits de sélection à cet utilisateur
- On suppose maintenant que vous souhaitez accorder des droits de sélection à chacun des étudiants du groupe mais uniquement pour les données qui le concerne. Par exemple, s'il existe des tuples dont la valeur pour identifiant est `johnsmith`, l'utilisateur `johnsmith` doit pouvoir sélectionner ces tuples mais aucun autre. Proposez une manière de réaliser cette politique de sécurité ?

Indication : on pourra s'appuyer sur la création d'une vue. Regardez aussi la documentation de la variable `CURRENT_USER`.

On peut prendre temporairement l'identité d'un utilisateur `x` par : `set role x`.

% la table `datatpadmin` contient deux colonnes : identifiant et écart. Identifiant prend ses valeurs dans la liste des schémas d'étudiants `ISIFAR2022`. La table a été obtenue à partir de celle des résultats du contrôle continu.

```
create view student_view as select * from durand.datatpadmin
where identifiant=current_user;
grant usage on schema durand to isifar2022;
grant select on durand.student_view to isifar2022;
```

- Dans la continuité de la question précédente, accordez à chaque utilisateur le droit de modifier les tuples qui le concerne dans votre table.