ISEN ALL IS DIGITAL! OUEST | yncréa

L’administration des bases de données

Avec PostgreSQL

benoit.lardeux@isen-ouest.yncrea.fr

Administration des BDD

- Thèmes traités
  - Création de base
  - Gestion des utilisateurs
  - Création des vues
  - Création des indexes

Administration des BDD

- Création de base
  - Pour créer la base '*mytest*':

    create database *mytest*;

  - Pour créer la base '*mytest*' et spécifier l'appartenance à '*testuser*'

    create database *mytest* owner *testuser*;

  - Pour se connecter à la base 'mytest' dans psql (si les droits le permettent)

    \c *mytest*;

Administration des BDD

- La gestion des rôles
  - Un rôle peut être défini soit pour un utilisateur ou un groupe d'utilisateurs. La notion de rôle regroupe les deux cas.

create role *rolename* login;

create role *rolename* login password '*mypwd*';

Est similaire à

create user *rolename*;

create user *rolename* with encrypted password '*mypwd*';

Administration des BDD

- Création des rôles
  - Principaux droits (attributs) des rôles

Nom	Description	Syntaxe SQL
LOGIN	Permet au rôle de se connecter à une base de données	CREATE ROLE user LOGIN;
SUPERUSER	Permet au rôle de tout faire au niveau de la base de données	CREATE ROLE postgres SUPERUSER;
CREATEDB	Permet au rôle de créer des bases de données	CREATE ROLE user CREATEDB;
CREATEROLE	Permet au rôle de créer d'autres rôles	CREATE ROLE user CREATEROLE;
PASSWORD	Assigne un mot de passe à un rôle	CREATE ROLE user PASSWORD 'mypwd';
INHERIT ou NOINHERIT	Permet, ou pas, à un rôle d'hériter des attributs d'autres rôles	CREATE ROLE user INHERIT;

Administration des BDD

- Création des rôles
  - Affichage de la liste des rôles: `\du`

```
postgres=# create role developpers nosuperuser inherit createdb createrole replication login password 'dev';
CREATE ROLE
postgres=# \du
                                   List of roles
  Role name   |                         Attributes                         | Member of
--------------+------------------------------------------------------------+-----------
 admin        | Create role, Create DB, Replication                        | {}
 admindb      |                                                            | {}
 blardeux     |                                                            | {}
 developpers  | Create role, Create DB, Replication                        | {}
 postgres     | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testrole     |                                                            | {}
 user1        |                                                            | {}
 user2        |                                                            | {}
 usera        |                                                            | {}
 userb        |                                                            | {}
 userbl       |                                                            | {}
 utilisateur1 |                                                            | {}
```

Administration des BDD

- Modification des rôles

alter role *testuser* createdb;

```
postgres=# alter role blardeux createdb;
ALTER ROLE
postgres=# \du
                                   List of roles
  Role name   |                         Attributes                         | Member of
--------------+----------------------------------------------------------------+-----------
 admin        | Create role, Create DB, Replication                            | {}
 admindb      |                                                                | {}
 blardeux     | Create DB                                                      | {}
 developpers  | Create role, Create DB, Replication                            | {}
 postgres     | Superuser, Create role, Create DB, Replication, Bypass RLS     | {}
 testrole     |                                                                | {}
 user1        |                                                                | {}
 user2        |                                                                | {}
 usera        |                                                                | {}
 userb        |                                                                | {}
 userbl       |                                                                | {}
 utilisateur1 |                                                                | {}
```

Administration des BDD

- Suppression des rôles et utilisateurs

drop role *testuser*;

drop user *testuser*;

Administration des BDD

- La gestion des privilèges (sous PostgreSQL)
  - Les droits d'accès peuvent être définis:
    - pour une base de données complète
    - pour des objets particuliers d'une base de données (tables, views…)
    - pour des actions (INSERT, UPDATE, …)

Administration des BDD

- Gestion des privilèges
  - Donner un type de permission sur une table (SELECT, UPDATE, DELETE,…)

    grant *permissiontype* on table *tablename* to *rolename*;

  - Donner toutes les permissions sur les tables d'une base de données:

    grant all privileges on all tables in schema public to *rolename*;

Administration des BDD

- Gestion des privilèges
  - CREATE: permet aux utilisateurs de créer des bases de données/tableaux
  - SELECT: permet aux utilisateurs de récupérer des données
  - INSERT: permet aux utilisateurs d'ajouter de nouvelles entrées dans les tableaux
  - UPDATE: permet aux utilisateurs de modifier les entrées existantes dans les tableaux

Administration des BDD

- Gestion des privilèges
  - DELETE: permet aux utilisateurs de supprimer les entrées du tableau
  - TRUNCATE: permet aux utilisateurs de supprimer une table ou un ensemble de tables
  - REFERENCES: droits requis pour création de clé étrangère
  - TRIGGER: permet de créer de nouveaux déclencheurs associés à la table
  - EXECUTE: permet d'exécuter une fonction

Administration des BDD

- Gestion des privilèges
  - Affichage des privilèges utilisateurs sur les tables

\z;
\dp;

```
dbtp1=> \z
                                    Access privileges
 Schema |     Name      |       Type        |     Access privileges     | Column privileges | Policies
--------+---------------+-------------------+---------------------------+-------------------+----------
 public | employe       | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | admin=arwdDxt/blardeux   +|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | employe_count | materialized view | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | admin=arwdDxt/blardeux   +|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | employe_ville | view              | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | admin=arwdDxt/blardeux   +|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | personne      | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | admin=arwdDxt/blardeux   +|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | service       | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | admin=arwdDxt/blardeux   +|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
(5 rows)
```

a	Insert (append)
r	Select (read)
w	Update (write)
d	Delete
D	Truncate
x	References
t	Trigger

Administration des BDD

- Modification des privilèges
  - Supprimer un type de permission sur une table

    revoke *permissiontype* on table *tablename* from *rolename*;

  - Enlever toutes les permissions

    revoke all privileges on all tables in schema public from *rolename*;

```
dbtp1=> revoke all privileges on all tables in schema public from admin;
REVOKE
dbtp1=> \z
                                    Access privileges
 Schema |     Name      |       Type        |     Access privileges     | Column privileges | Policies
--------+---------------+-------------------+---------------------------+-------------------+----------
 public | employe       | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | employe_count | materialized view | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | employe_ville | view              | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | personne      | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
 public | service       | table             | blardeux=arwdDxt/blardeux+|                   |
        |               |                   | postgres=arwdDxt/blardeux |                   |
```

Administration des BDD

- Une bonne habitude
  - Créer (au moins) un utilisateur spécifique à chaque création de base de donnée

Administration des BDD

- Description d'une base de données

sakila =# \d ;

```
sakila=# \d
 public | actor                      | table    | postgres
 public | actor_actor_id_seq         | sequence | postgres
 public | actor_info                 | view     | postgres
 public | address                    | table    | postgres
 public | address_address_id_seq     | sequence | postgres
 public | category                   | table    | postgres
 public | category_category_id_seq   | sequence | postgres
 public | city                       | table    | postgres
 public | city_city_id_seq           | sequence | postgres
 public | country                    | table    | postgres
 public | country_country_id_seq     | sequence | postgres
 public | customer                   | table    | postgres
 public | customer_customer_id_seq   | sequence | postgres
 public | customer_list              | view     | postgres
 public | film                       | table    | postgres
 public | film_actor                 | table    | postgres
 public | film_category              | table    | postgres
 public | film_film_id_seq           | sequence | postgres
 public | film_list                  | view     | postgres
 public | inventory                  | table    | postgres
 public | inventory_inventory_id_seq | sequence | postgres
 public | language                   | table    | postgres
 public | language_language_id_seq   | sequence | postgres
 public | nicer_but_slower_film_list | view     | postgres
 public | payment                    | table    | postgres
 public | payment_p2007_01           | table    | postgres
 public | payment_p2007_02           | table    | postgres
 public | payment_p2007_03           | table    | postgres
 public | payment_p2007_04           | table    | postgres
 public | payment_p2007_05           | table    | postgres
 public | payment_p2007_06           | table    | postgres
 public | payment_payment_id_seq     | sequence | postgres
 public | rental                     | table    | postgres
 public | rental_rental_id_seq       | sequence | postgres
```

Administration des BDD

- Description d'une table

sakila =# \d film;

```
                                    Table "public.film"
       Column        |            Type             | Collation | Nullable |                  Default
---------------------+-----------------------------+-----------+----------+-------------------------------------
 film_id             | integer                     |           | not null | nextval('film_film_id_seq'::regclass)
 title               | character varying(255)      |           | not null |
 description         | text                        |           |          |
 release_year        | year                        |           |          |
 language_id         | smallint                    |           | not null |
 original_language_id | smallint                   |           |          |
 rental_duration     | smallint                    |           | not null | 3
 rental_rate         | numeric(4,2)                |           | not null | 4.99
 length              | smallint                    |           |          |
 replacement_cost    | numeric(5,2)                |           | not null | 19.99
 rating              | mpaa_rating                 |           |          | 'G'::mpaa_rating
 last_update         | timestamp without time zone |           | not null | now()
 special_features    | text[]                      |           |          |
 fulltext            | tsvector                    |           | not null |
Indexes:
    "film_pkey" PRIMARY KEY, btree (film_id)
    "film_fulltext_idx" gist (fulltext)
    "idx_fk_language_id" btree (language_id)
    "idx_fk_original_language_id" btree (original_language_id)
    "idx_title" btree (title)
Foreign-key constraints:
    "film_language_id_fkey" FOREIGN KEY (language_id) REFERENCES language(language_id) ON UPDATE CASCADE ON DELETE RESTRICT
    "film_original_language_id_fkey" FOREIGN KEY (original_language_id) REFERENCES language(language_id) ON UPDATE CASCADE ON DELETE RESTRICT
Referenced by:
    TABLE "film_actor" CONSTRAINT "film_actor_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
    TABLE "film_category" CONSTRAINT "film_category_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
    TABLE "inventory" CONSTRAINT "inventory_film_id_fkey" FOREIGN KEY (film_id) REFERENCES film(film_id) ON UPDATE CASCADE ON DELETE RESTRICT
Triggers:
    film_fulltext_trigger BEFORE INSERT OR UPDATE ON film FOR EACH ROW EXECUTE PROCEDURE tsvector_update_trigger('fulltext', 'pg_catalog.english', 'title', 'description')
    last_updated BEFORE UPDATE ON film FOR EACH ROW EXECUTE PROCEDURE last_updated()

sakila=#
```

Administration des BDD

- Les vues
  - Tables virtuelles
  - Créée avec un select (jointure possible)
  - Ne conserve pas les données, sauf pour les vues matérialisées

create view *nomvue* as select …;

Administration des BDD

- Les vues - intérêt
  - Limitation des données en fonction des utilisateurs
  - Simplification de l'affichage des données
  - Indépendance de la vue en affichage par rapport à la complexité du modèle et de ses évolutions
  - Les requêtes sont possibles sur les vues
  - Les insert et update fonctionnent sous certaines conditions

Administration des BDD

- Les vues - exemples

```
dbtp1=# \d
 public | employe       | table | blardeux
 public | employe_ville | view  | postgres
 public | personne      | table | blardeux
 public | service       | table | blardeux

dbtp1=# create view employe_count as select num_service, count(*) from employe group by num_service;
CREATE VIEW
dbtp1=# \d
 public | employe       | table | blardeux
 public | employe_count | view  | postgres
 public | employe_ville | view  | postgres
 public | personne      | table | blardeux
 public | service       | table | blardeux
```

```
dbtp1=# select * from employe_count;
          40 |     2
          30 |     6
          10 |     3
          20 |     5
```

Administration des BDD

- Les vues matérialisées
  - Enregistrent physiquement les données d'une requête
  - Peuvent être rafraichies manuellement

```
dbtp1=# create materialized view employe_count as select num_service, count(*) from employe group by num_service;
SELECT 4
dbtp1=# refresh materialized view employe_count;
REFRESH MATERIALIZED VIEW
dbtp1=#
```

Administration des BDD

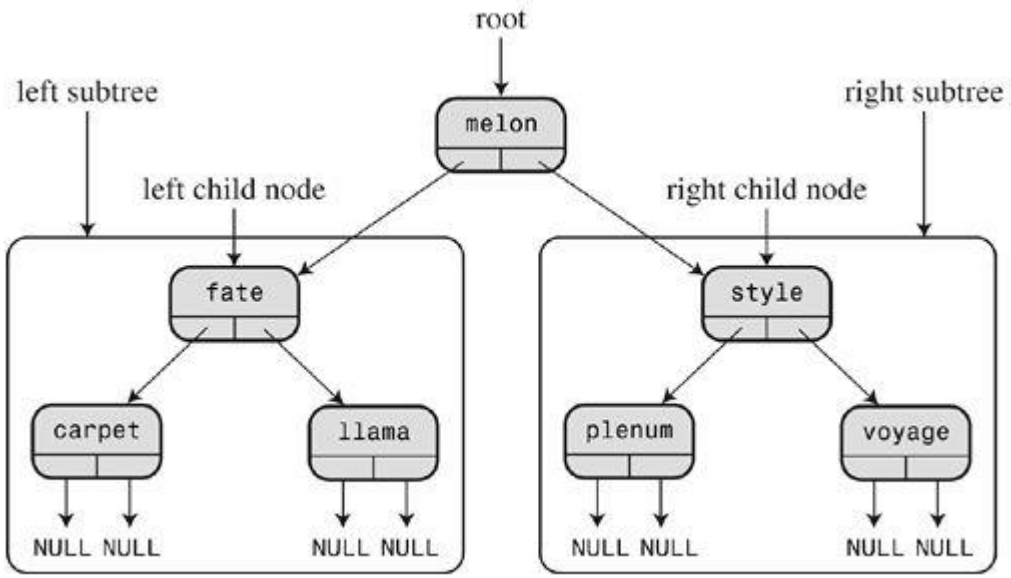
- Les indexes
  - Accélèrent les requêtes
    - Sur les grosses bases
    - Sur les jointures
  - Créés automatiquement pour les clés primaires
  - Intéressants sur les champs *int* et *varchar*
  - Création de fichiers pour les indexes

create index *nomindex* on *nomtable(nomcolonne);*

Administration des BDD

- Les indexes - visualisation

```
dbtp1=# create index index_nom on employe(nom);
CREATE INDEX
dbtp1=# select tablename, indexname, indexdef from pg_indexes where schemaname = 'public' order by tablename, indexname;
 tablename |   indexname   |                                        indexdef
-----------+---------------+-------------------------------------------------------------------------------
 employe   | employe_pkey  | CREATE UNIQUE INDEX employe_pkey ON public.employe USING btree (numemp)
 employe   | index_nom     | CREATE INDEX index_nom ON public.employe USING btree (nom)
 personne  | personne_pkey | CREATE UNIQUE INDEX personne_pkey ON public.personne USING btree (login)
 service   | service_pkey  | CREATE UNIQUE INDEX service_pkey ON public.service USING btree (num_service)
(4 rows)
```



Administration des BDD

- Les indexes – le choix
  - Pour choisir un index
    - Vérifier la répartition des valeurs
    - De préférence sur des champs uniques
    - Sur les champs apparaissant dans *where*, *group by*, dans les critères de jointures

Administration des BDD

- Les indexes – impacts
  - Les indexes ont un impact sur
    - La taille de la base de données (un nouveau fichier créé par index)
    - Les performances quand trop d'indexes (insertion, mise à jour) car il faut
      - Ecrire dans la table
      - Ecrire dans l'index
      - Réorganiser l'index

Administration des BDD

- Les indexes – exemple
  - Tables DAMIR => > 31 millions d'enregistrements

```
damir=# select count (*) from damir_201812 where prs_nat = 3341 and ben_res_reg = 84 and age_ben_snds = 50;
 count
-------
  5595
(1 row)
```

PRS_NAT=3341 /* Consultations
BEN_RES_REG=84 /* Bretagne
AGE_BEN_SNDS=50 /* 50-59 ans

Administration des BDD

- Les indexes – exemple
  - Tables DAMIR => > 31 millions d'enregistrements

```
damir=# select tablename, indexname, indexdef from pg_indexes where schemaname = 'public' order by tablename, indexname;
  tablename   |     indexname     |                                                indexdef
--------------+-------------------+-------------------------------------------------------------------------------------
 damir_201812 | index_ben_res_reg | CREATE INDEX index_ben_res_reg ON public.damir_201812 USING btree (ben_res_reg)
 damir_201812 | index_ben_snds    | CREATE INDEX index_ben_snds ON public.damir_201812 USING btree (age_ben_snds)
 damir_201812 | index_prs_nat     | CREATE INDEX index_prs_nat ON public.damir_201812 USING btree (prs_nat)
(3 rows)
```

Administration des BDD

- Les indexes – exemple
  - Tables DAMIR => > 31 millions d'enregistrements

```
damir=# select count (*) from damir_201812 where prs_nat = 3341 and ben_res_reg = 84 and age_ben_snds = 50;
 count
-------
  5595
(1 row)
```

Table	Nb enregistrements	Type	Temps	Taille
DAMIR	31 M	1 indexe	1mn 03 secs	9.431 GB
DAMIR	31 M	2 indexes	21 secs	10.103 GB
DAMIR	31 M	3 indexes	18 secs	10.775 GB
DAMIR	31 M	4 indexes	18 secs	11.447 GB

Administration des BDD

- Les indexes – taille
  - Pour obtenir la taille des tables et index, on utilise les fonctions suivantes:

```
dbtp1=> select pg_database_size('dbtp1');
 pg_database_size
------------------
          7937159
(1 row)
```

  - Fonctions pour plus de lisibilité

```
dbtp1=> select pg_size_pretty(pg_database_size('dbtp1'));
 pg_size_pretty
----------------
 7751 kB
(1 row)
```

  - Pour une table

```
dbtp1=> select pg_size_pretty(pg_relation_size('employe'));
 pg_size_pretty
----------------
 8192 bytes
(1 row)
```

  - Pour un index

```
dbtp1=> select pg_size_pretty(pg_relation_size('index_nom'));
 pg_size_pretty
----------------
 16 kB
(1 row)
```

Administration des BDD

- Optimiser les requêtes
  - Utiliser la commande *explain*
  - Permet de visualiser:
    - Les tables et leur ordre utilisé
    - Les indexes utilisés
    - L'estimation du nombre de lignes utilisés

Administration des BDD

- Optimiser les requêtes
  - Utiliser la commande *explain*
    - Planification de l'exécution de la commande

```
sakila=> explain select count(*) from film;
                         QUERY PLAN
-------------------------------------------------------------
 Aggregate  (cost=67.50..67.51 rows=1 width=8)
   ->  Seq Scan on film  (cost=0.00..65.00 rows=1000 width=0)
(2 rows)
```

Administration des BDD

- Optimiser les requêtes
  - Utiliser la commande *explain analyze*
    - Exécution et analyse statistique de la commande

```
sakila=> explain analyze select count (*) from film;
                                                     QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Aggregate  (cost=67.50..67.51 rows=1 width=8) (actual time=1.881..1.882 rows=1 loops=1)
   ->  Seq Scan on film  (cost=0.00..65.00 rows=1000 width=0) (actual time=0.010..0.978 rows=1000 loops=1)
 Planning time: 0.099 ms
 Execution time: 1.910 ms
(4 rows)
```

Administration des BDD

- Optimiser les requêtes
  - Utiliser la commande *explain analyze verbose*
    - Exécution et analyse statistique de la commande
    - Information complémentaire concernant la planification de l'exécution de la requête

```
sakila=> explain analyze verbose select count (*) from film;
                                                              QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 Aggregate  (cost=67.50..67.51 rows=1 width=8) (actual time=3.706..3.710 rows=1 loops=1)
   Output: count(*)
   ->  Seq Scan on public.film  (cost=0.00..65.00 rows=1000 width=0) (actual time=0.022..1.928 rows=1000 loops=1)
         Output: film_id, title, description, release_year, language_id, original_language_id, rental_duration, rental_rate, length, replacement_cost, rating, last_update, special_features, fulltext
 Planning time: 0.166 ms
 Execution time: 3.773 ms
(6 rows)
```

Warning: Temps supplémentaire conséquent pour des requêtes complexes !

Administration des BDD

- Les déclencheurs (trigger)
  - Les triggers sont (en général) des fonctions ou procédures qui sont associées aux tables et activées sur les événements de type:
    - *Insert*
    - *Update*
    - *Delete*

Administration des BDD

- Les déclencheurs (trigger)
  - Utilisés avec le langage procédure Pl/pgSQL
  - Création de la fonction:

```
dbtp1=> create function negative_salaire() returns trigger as $negative_salaire$
dbtp1$> begin
dbtp1$> if new.salaire < 0 then
dbtp1$> raise exception '% ne peut pas avoir de salaire négatif!', new.nom;
dbtp1$> end if;
dbtp1$> end;
dbtp1$> $negative_salaire$ language plpgsql;
CREATE FUNCTION
```

  - Initialisation du trigger

```
dbtp1=> create trigger negative_salaire before insert or update on employe for each row execute procedure neg_salaire();
CREATE TRIGGER
```

Administration des BDD

- Les déclencheurs (trigger)
  - Utilisation

```
dbtp1=> insert into employe values (1002, 'LeKer', 'ingenieur', 1000, current_date, -1400, null, 40);
ERROR:  LeKer ne peut pas avoir de salaire négatif!
CONTEXT:  PL/pgSQL function negative_salaire() line 4 at RAISE
```

  - Affichage
    - *\df* pour afficher les fonctions
    - *Select tgname from pg_trigger;* pour afficher les déclencheurs

Question?