



ASRBD


RENDU TP MIGRATION DE DONNEES

Membres du groupe :

Allan BIVEGHE

Arthur SIMOES

Thomas HAMILCARO



EPSI

NIVEAU 1

Étape 1 : Téléchargement des logiciels et installation des outils



Étape 2 : Préparation de la base de données source

1. Créons une base de données MySQL : (là c'est à titre d'exemple la base de donnée créée est voyage)

```
create database test;
```

2. Peuplons la base de données

```
C:\Users\biveg> OneDrive > Bureau > test migration données > migration.py > ...
1 from faker import Faker
2 import mysql.connector
3 import random
4
5 fake = Faker()
6
7 # Connexion à la base de données (à ajuster avec tes paramètres de connexion)
8 conn = mysql.connector.connect(
9     host='ton_host',
10    user='ton_user',
11    password='ton_password',
12    database='ReservationVoyage'
13 )
14 cursor = conn.cursor()
15
16 # Générer et insérer des données factices
17 for _ in range(10000):
18     nom = fake.last_name()
19     prenom = fake.first_name()
20     email = fake.email()
21     motdepasse = fake.password(length=12)
22
23     query = "INSERT INTO Utilisateurs (Nom, Prenom, Email, MotDePasse) VALUES (%s, %s, %s, %s)"
24     cursor.execute(query, (nom, prenom, email, motdepasse))
25
26 conn.commit()
27 cursor.close()
28 conn.close()
29
30
```

Étape 3 : Migration avec DBeaver

Navigation de bases de données x Projets

Enter a part of object name here

- Settings
 - Roles
 - pg_checkpoint
 - pg_create_subscription
 - pg_database_owner
 - pg_execute_server_program
 - pg_monitor
 - pg_read_all_data
 - pg_read_all_settings
 - pg_read_all_stats
 - pg_read_server_files
 - pg_signal_backend
 - pg_stat_scan_tables
 - pg_use_reserved_connections
 - pg_write_all_data
 - pg_write_server_files
 - postgres
 - Administer
 - System Info
- voyage - localhost:3306
 - Bases de données
 - Utilisateurs
 - Administer
 - Informations Système

Project - General x

Name DataSource

Bookmarks

Diagrams

Scripts

Propriétés Données ER Diagram

users Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	nom	prenom	email	motdepasse
1	Williams	Cathy	jason47@example.org	_C\$6UUh9Ya
2	Mills	Melissa	jessica35@example.org	_Nc5GCgs(K7
3	Manning	Shelly	courtney13@example.c	_npY++su+4H
4	Howard	Lisa	anthonyholder@examp	_pilw+)v2On
5	Bradshaw	Juan	joshuawhite@example.	_y_2+Cx@U8M
6	Jones	Carol	emilyrobinson@examp	_l\$u4Ws8n3YG
7	Aguirre	Howard	smcintosh@ex	emilyrobinson@example.org
8	Tran	Amanda	garciacorey@examp	emilyrobinson@example.org
9	Terry	Amanda	coreymatthews@exam	_l8Bf4GAW22V
10	King	Janet	anthony94@example.ni	_lD5U(RBf4Cr
11	Hayden	Carmen	scott73@example.org	_l(f8DlE\$0*Ag
12	Reed	Ashley	hsmith@example.net	_l(Y0AJf_Lv#o
13	Webster	Shannon	jenna99@example.net	_l(Y08bV^Ue^P
14	Wheeler	Sheena	xsmith@example.com	_lJ5C9ReUD(lL
15	Miller	Jonathan	ublack@example.com	_lJ5o&EOYsI4
16	Davis	Melanie	fgonzalez@example.ne	_lJ5WhLdS(6HC
17	Weiss	Kristy	kyle41@example.com	_lJ5_2wowW7HT
18	Johnson	Emily	mcdonaldmitchell@ex	_lJ5Z7WE0o*QqW
19	Lopez	Steven	christopherpatterson@	_lJ5@3Bycr^mcGS
20	Davis	Christopher	debra58@example.net	_lJ5@3ZkQnH2rS
21	Smith	Jeffrey	ianhenderson@exampl	_lJ5@3Zudmcf(\$@
22	Rhodes	Vincent	qkim@example.org	_lJ5@7Yrev4QQ1
23	Yates	Michelle	cabreraonica@exampl	_lJ5@9aWthv+2+C
24	Cobb	Edward	perezmelanie@exampl	_lJ5@iDNUE^rD1t
25	Peterson	Edwin	potterjames@example.	_lJ5@lpe9V00d5n
26	Diaz	Tanya	kramermatthew@exam	_lJ5@N%25SeFm77
27	Pierce	Amv	ihiaqins@example.com	_lJ5@v4ReipHU^P

Etape 4 verification

pgAdmin 4

File Object Tools Help

Object Explorer

- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (2)
 - public
 - users
 - Columns (4)
 - nom
 - prenom
 - email
 - motdepasse
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - Trigger Functions

Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x

General System Statistics

Database sessions

Total Active Idle Transactions p

4 8

2 6

0 4

0 2

0 0

Tuples in Inserts Updates Deletes

100

75

50

25

0

Tuples out Fetched

400

200

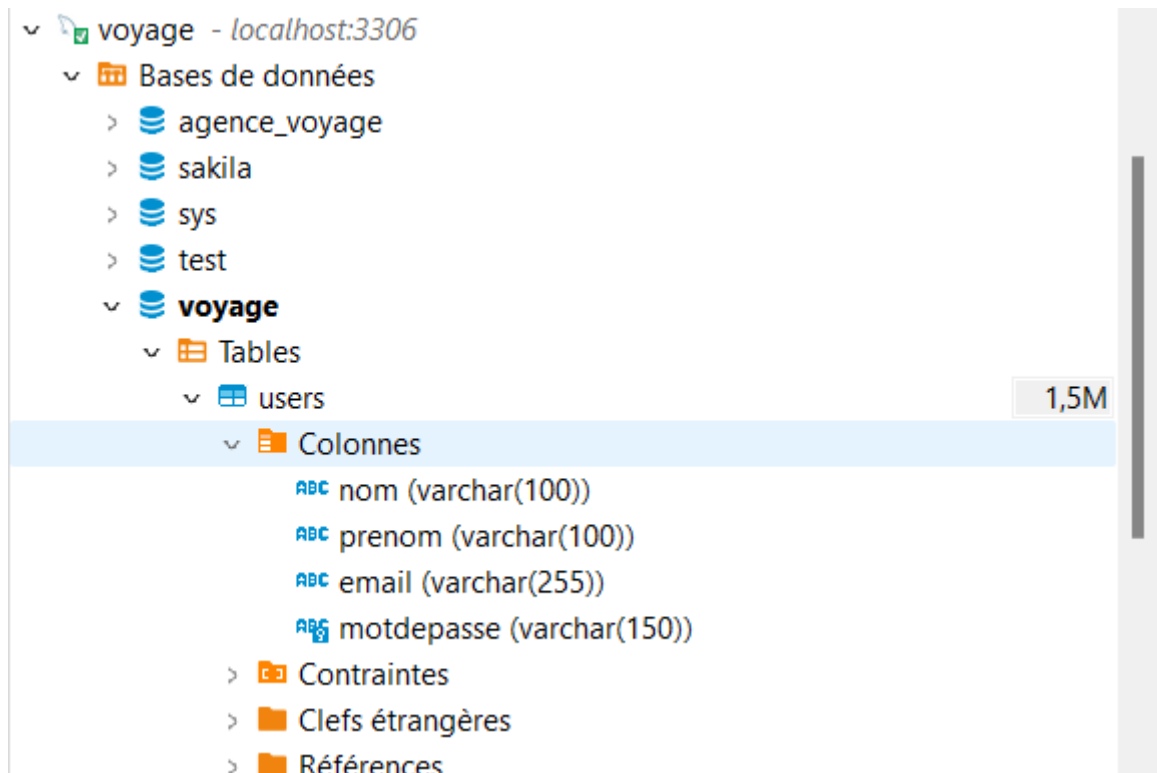
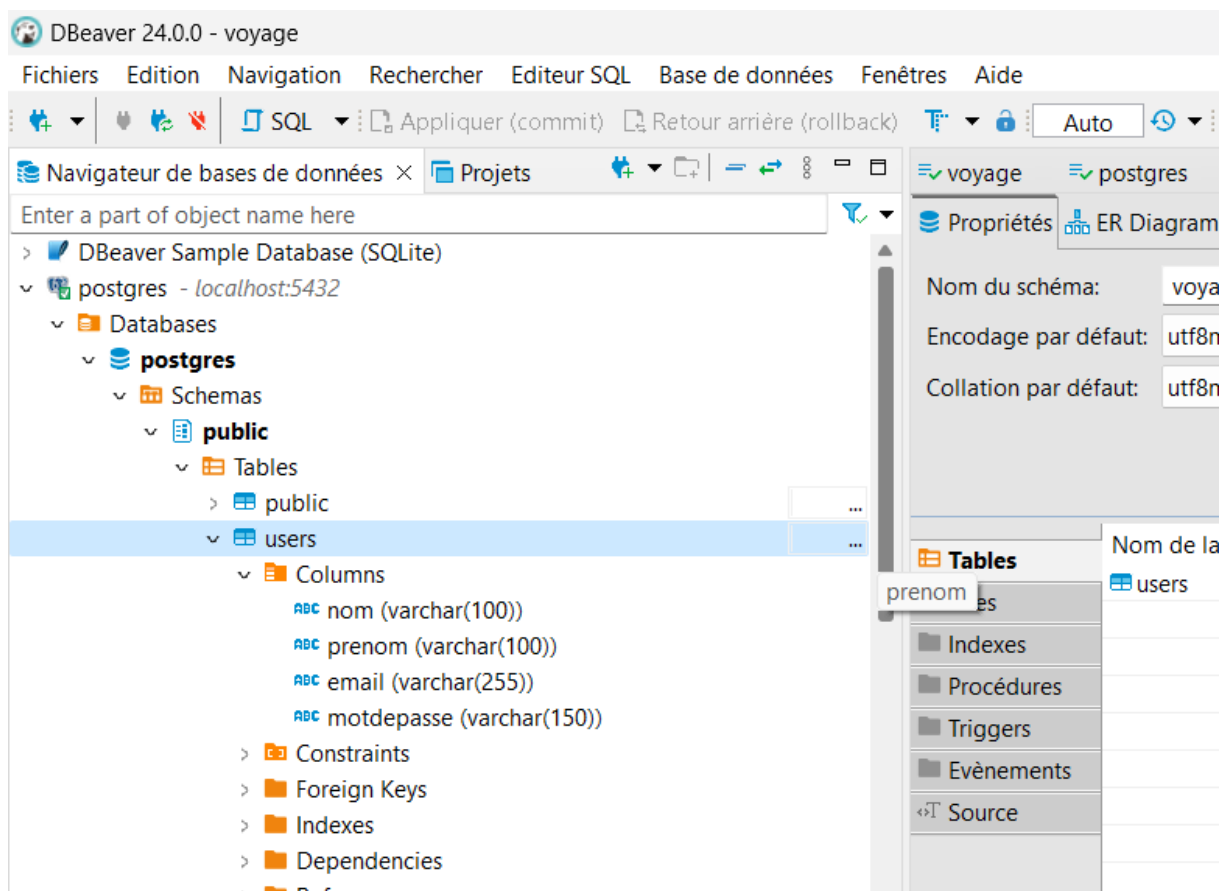
0

Database activity

Sessions Locks Prepared Transactions

☐ Active sessions only

			PID	User	Application	Client	Backend start
✖	■	▶	14172	postgr...	DBeaver 24.0.0 - Mai...	127.0.0.1	2024-04-22 11:51:01...
✖	■	▶	19536	postgr...	DBeaver 24.0.0 - Me...	127.0.0.1	2024-04-22 11:51:01...
✖	■	▶	20020	postgr...	DBeaver 24.0.0 - Me...	127.0.0.1	2024-04-22 12:08:37...



NIVEAU 2

Étape 1 : Configuration initiale

Après lancement de mon docker sur ma machine virtuelle avec la commande ***sudo apt-get install Docker.io*** (entrer) puis ***sudo systemctl start docker*** nous avons installer Mysql et posgre avec les commandes docker pull ...

```
permission denied

(kali@kali)-[~]
$ sudo docker pull mysql:latest
latest: Pulling from library/mysql
bd37f6d99203: Downloading  9.186MB/45.28MB
e733cb057651: Download complete
af2fd35011dc: Download complete
e5233d0f6ee3: Download complete
cf11fd8658d3: Download complete
85344d57c3cb: Download complete
0eebca71f40d: Downloading  7.568MB/60.31MB
18e468a1ddac: Download complete
d9b2b8d35c75: Downloading  11.28MB/61.52MB
57ba1b7684b4: Waiting
```

```
be
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

(kali@kali)-[~]
$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
13808c22b207: Downloading  13.72MB/29.13MB
e59e71301578: Download complete
30258475ebe6: Download complete
dc0c37c3ab3c: Download complete
80df7455ae43: Downloading  7.908MB/8.066MB
c200c670beef: Download complete
d051a616a810: Download complete
87c27dce0c4d: Download complete
9dae96d37274: Downloading  2.149MB/109MB
9b9e5e72a98e: Waiting
73301da916ba: Waiting
673a1691030a: Waiting
16d03b1cb8d8: Waiting
9d680d6449b5: Waiting
```

Étape 2 : Préparation de la base de données MySQL

Nos dockers Mysql et postgres sont bien créées et en cours d'exécution maintenant créons une base de données à l'intérieur de mon Docker Mysql :

```
(kali@kali)-[~]
$ sudo docker exec -it mysql-container mysql -u root -p

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.

mysql>
```

```
mysql> CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50),
prenom Varchar (60), email VARCHAR(50), mot2passe varchar (150));
ERROR 1050 (42S01): Table 'users' already exists
mysql> CREATE TABLE test (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50),
prenom Varchar (60), email VARCHAR(50), mot2passe varchar (150));
Query OK, 0 rows affected (0.07 sec)

mysql> INSERT INTO test (name,prenom, email,mot2passe) VALUES ('John', 'CENA',
, 'john@wwe.com','https'), ('Gay','sensei', 'gay@example.com','Burgerking');

Query OK, 2 rows affected (0.03 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
```

Test et verification :

```
mysql> select * from test;
+----+-----+-----+-----+-----+
| id | name | prenom | email | mot2passe |
+----+-----+-----+-----+-----+
| 1  | John | CENA   | john@wwe.com | https |
| 2  | Gay  | sensei | gay@example.com | Burgerking |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```


Étape 3 : Configuration de Flyway pour la migration

Création du dossier et des sous-dossiers :

```
C:\Users\biveg>mkdir flyway_migration

C:\Users\biveg>cd flyway_migration

C:\Users\biveg\flyway_migration>mkdir sql conf

C:\Users\biveg\flyway_migration>dir
Le volume dans le lecteur C s'appelle Windows
Le numéro de série du volume est 800F-2BA7

Répertoire de C:\Users\biveg\flyway_migration

22/04/2024  15:25    <DIR>          .
22/04/2024  15:24    <DIR>          ..
22/04/2024  15:25    <DIR>          conf
22/04/2024  15:25    <DIR>          sql
               0 fichier(s)                0 octets
               4 Rép(s)  229 454 974 976 octets libres
```

Adresse ip du container postgres

```
(kali@kali)-[~]
$ sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgresql-container
172.17.0.3
```

Cible postgres

```
Fichier Actions Éditer Vue Aide
GNU nano 7.2 test
flyway.url=jdbc:postgresql://172.17.0.3:5432/postgresql-container
flyway.user=root
flyway.password=root
```

Dans le sous-dossier `sql`, créons un fichier nommé `V1__Create_table.sql` contenant les commandes SQL pour recréer la structure de la base de données MySQL dans PostgreSQL.

```
GNU nano 7.2
CREATE TABLE IF NOT EXISTS Users
(
  nom varchar(255),
  prenom varchar(255),
  email varchar(255),
  password varchar(255)
);
```

```
INSERT INTO users(nom,prenom,email,password) VALUES ('ARAGON','Romeo','roméo.arau@orange.fr','1234567890');
```

V2 insert data :

Étape 4 : Lancement de la migration avec Flyway

- Exécution de Flyway
 - Ouvre un terminal et navigue vers ton dossier `flyway_migration`.
 - Lance la migration en exécutant Flyway avec Docker :

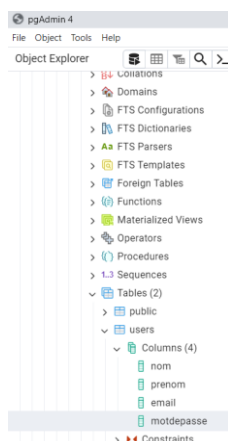
```
psycho@elavet: /workspace/flyway_migration$ docker run --rm -v $(pwd)/sql:/flyway/sql -v $(pwd)/conf:/flyway/conf flyway/flyway m
WARNING: Storing migrations in 'sql' is not recommended and default scanning of this location may be deprecated in a future rel
Flyway OSS Edition 10.10.0 by Redgate

See release notes here: https://rd.gt/4160bMt
Database: jdbc:postgresql://192.168.88.152:5432/postgres (PostgreSQL 16.2)
Successfully validated 2 migrations (execution time 00:00.038s)
Current version of schema "public": 1
Migrating schema "public" to version "2 - Insert data"
Successfully applied 1 migration to schema "public", now at version v2 (execution time 00:00.103s)
```

La migration s'est bien passé grâce au retour successful

Étape 5 : Vérification de la migration

Puis on vérifie toujours de manières visuelles sur pg4 :



Tout a bien été migré

Niveau 3

Étape 1 : Créer des scripts de test

Script de verification :

```
-- Verification des clés étrangères
SELECT 'Vérification des clés étrangères : Utilisateurs' AS Test,
       COUNT(*) AS Erreurs
FROM Utilisateurs u
LEFT JOIN Profils p ON u.id_profil = p.id
WHERE p.id IS NULL;

-- Vérification de l'absence de valeurs nulles
SELECT 'Vérification de l'absence de valeurs nulles : Utilisateurs' AS Test,
       COUNT(*) AS Erreurs
FROM Utilisateurs
WHERE email IS NULL;
```

Étape 2 : Exécuter Flyway

Nous avons installé Flyway avec la commande en ligne de commande

Linux

Download, extract and install by adding to PATH (requires sudo permissions):

```
$ wget -qO- https://download.red-gate.com/maven/release/com/redgate/flyway/flyway-
commandline/10.11.1/flyway-commandline-10.11.1-linux-x64.tar.gz | tar -xvz && sudo ln -s
`pwd`/flyway-10.11.1/flyway /usr/local/bin
```

Telechargement de flyway

```
(root@kali)~[~]
# wget https://flywaydb.org/download.
--2024-04-22 09:55:05-- https://flywaydb.org/download.
Résolution de flywaydb.org (flywaydb.org)... 54.229.94.161
Connexion à flywaydb.org (flywaydb.org)|54.229.94.161|:443... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : https://flywaydb.org/download [suivant]
--2024-04-22 09:55:05-- https://flywaydb.org/download
Réutilisation de la connexion existante à flywaydb.org:443.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : https://www.red-gate.com/products/flyway/editions [suivant]
--2024-04-22 09:55:05-- https://www.red-gate.com/products/flyway/editions
Résolution de www.red-gate.com (www.red-gate.com)... 34.250.173.81, 34.241.8.24
4
Connexion à www.red-gate.com (www.red-gate.com)|34.250.173.81|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 40994 (40K) [text/html]
Sauvegarde en : « download. »

download.          100%[=====>] 40,03K --.-KB/s   ds 0,03s

2024-04-22 09:55:06 (1,30 MB/s) - « download. » sauvegardé [40994/40994]

(root@kali)~[~]
#
```

Flyway

Script bash pour lancer automatiquement les migration garce au sript fait dans le .conf et le .sql

```
#!/bin/bash

# Exécution des migrations Flyway
echo "Exécution des migrations Flyway..."
docker run --rm -v $(pwd)/sql:/flyway/sql -v $(pwd)/conf:/flyway/conf flyway/flyway migrate

# Exécution des tests post-migration
echo "Exécution des tests post-migration..."
for test_script in $(pwd)/post-migration-tests/*.sql; do
    echo "Exécution de $(basename "$test_script")"
    docker exec -it postgres-server psql -U postgres -d migration2 -f "$flyway/post-migration-tests/$(basename "$test_script")"
done
```

Verification

```
Exécution des migrations Flyway...
Flyway OSS Edition 10.11.1 by Redgate

See release notes here: https://rd.gt/4160bMi
Database: jdbc:postgresql://172.17.0.3:5432/migration2 (PostgreSQL 16.2)
Schema history table "public"."flyway_schema_history" does not exist yet
Successfully validated 2 migrations (execution time 00:00.032s)
Creating Schema History table "public"."flyway_schema_history" ...
Current version of schema "public": << Empty Schema >>
Migrating schema "public" to version "1 - Create table"
Migrating schema "public" to version "2 - Insert data"
Successfully applied 2 migrations to schema "public", now at version v2 (execution time 00:00.017s)
Exécution des tests post-migration...
Exécution de V3__Test_integrite.sql
valeurs_nulles
-----
              0
(1 row)

name | count
-----+-----
(0 rows)

Exécution de V4__Test_completude.sql
total_enregistrements
-----
              4
(1 row)
```