**TP1**

ssh -i fichier\_key ubuntu@adresse\_ip

psql -h **192.168.246.247** -U etum2 insee\_deces

chmod go-rwx pour enlever les droits du fichier\_key

voir la liste des serveurs distants enregistrés:

SELECT \* FROM pg\_foreign\_server;

voir la liste des tables d’une bd :

SELECT table\_name

FROM information\_schema.tables

WHERE table\_schema = 'public';

voir la liste de user mapping :

SELECT um.srvname AS server\_name, um.umuser AS local\_user, um.umoptions AS options FROM pg\_user\_mappings um;

pour trouver les type d’une table :

\d+ nom\_de\_la\_table;

**Question 1 :**

Que représente les 303 6725 n-uplets retournés par la requête suivante :

SELECT \*

FROM personne

WHERE lieunaiss NOT IN (

SELECT com

FROM commune );

renvoie les personnes dont le lieu de naissance n’est pas dans les communes inscrites dans la table commune

**Question 2 :**

Donner les adresses (postales et mail) des mairies des communes de naissance des personnes décédées dans le département 63 durant le mois d'avril 2018. (509 n-uplets attendus)

SELECT codeInsee, email

FROM mairie

JOIN commune ON commune.com = mairie.codeInsee

JOIN departement ON commune.dep = departement.dep

JOIN personne ON commune.com = personne.lieudeces

WHERE departement.dep = ‘63’

AND personne.datedeces BETWEEN ‘2018-04-01’ AND ‘2018-04-30’;

**A - Définition des connexions inter-bases**

**1 - Définir le 'wrapper' qui vous permettra de vous connecter à la base insee\_deces sur ggmd\_prof**

sudo -i -u postgres

psql

CREATE EXTENSION IF NOT EXISTS postgres\_fdw;

CREATE SERVER remote\_insee\_deces FOREIGN DATA WRAPPER postgres\_fdw OPTIONS (host ‘192.168.246.247’, port ‘5432’, dbname ‘insee\_deces’);

**2 - Définir l'association utilisateur local/utilisateur distant qui vous donnera le droit d'accès à la base insee\_deces.**

CREATE USER MAPPING FOR userlocal SERVER remote\_insee\_deces OPTIONS (user 'etum2', password 'password');

**3 - Définir les 'foreign tables' qui vous permettront d'accéder à la base *insee\_deces* depuis vos VM.**

CREATE FOREIGN TABLE personne(

idp integer,

nom varchar(80),

prenoms varchar(80),

datenaiss date,

lieunaiss varchar(8),

datedeces date,

lieudeces varchar(8),

age interval

) SERVER remote\_insee\_deces;

CREATE FOREIGN TABLE region(

reg varchar(5),

nom varchar(254),

cheflieu varchar(10),

zone smallint

) SERVER remote\_insee\_deces;

CREATE FOREIGN TABLE departement(

dep varchar(5),

nom varchar(254),

cheflieu varchar(10),

reg varchar(5)

) SERVER remote\_insee\_deces;

CREATE FOREIGN TABLE commune(

com varchar(5),

nom varchar(254),

dep varchar(5)

) SERVER remote\_insee\_deces;

CREATE FOREIGN TABLE mairie(

codeinsee varchar(10),

cp varchar(10),

nomorga varchar(254),

nomcom varchar(80),

email varchar(150),

tel varchar(150),

url varchar(254),

adresse varchar(254),

latitude varchar(150),

longitude varchar(150),

datemaj varchar(254)

) SERVER remote\_insee\_deces;

**B - Fragmentation des données**

**C1**

-- ggmd1 / Paris

CREATE TABLE Paris AS

SELECT \*

FROM region

WHERE zone = '1';

-- ggmd2 / Lyon

CREATE TABLE Lyon AS

SELECT \*

FROM region

WHERE zone = '2';

-- ggmd3 / Marseille

CREATE TABLE Marseille AS

SELECT \*

FROM region

WHERE zone = '3';

**C2**

CREATE TABLE remote\_departement AS SELECT d.dep, d.nom, d.cheflieu, d.reg FROM departement d JOIN region r ON d.reg = r.reg WHERE zone = 1;

CREATE TABLE remote\_departement AS SELECT d.dep, d.nom, d.cheflieu, d.reg FROM departement d JOIN region r ON d.reg = r.reg WHERE zone = 2;

CREATE TABLE remote\_departement AS SELECT d.dep, d.nom, d.cheflieu, d.reg FROM departement d JOIN region r ON d.reg = r.reg WHERE zone = 3;

-- ggmd1 / Paris

CREATE TABLE departement\_zone\_1 AS

SELECT \*

FROM departement d

WHERE d.reg IN (

SELECT reg

FROM Paris

);

-- ggmd2 / Lyon

CREATE TABLE departement\_zone\_2 AS

SELECT \*

FROM departement d

WHERE d.reg IN (

SELECT reg

FROM Lyon

);

-- ggmd3 / Marseille

CREATE TABLE departement\_zone\_3 AS

SELECT \*

FROM departement d

WHERE d.reg IN (

SELECT reg

FROM Marseille

);

**C3**

CREATE TABLE remote\_commune AS SELECT c.com, c.nom, c.dep FROM commune c JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 1;

CREATE TABLE remote\_commune AS SELECT c.com, c.nom, c.dep FROM commune c JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 2;

CREATE TABLE remote\_commune AS SELECT c.com, c.nom, c.dep FROM commune c JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 3;

-- ggmd1 / Paris

CREATE TABLE commune\_zone\_1 AS

SELECT \*

FROM commune c

WHERE c.dep IN (

SELECT dep

FROM departement\_zone\_1

);

-- ggmd2 / Lyon

CREATE TABLE commune\_zone\_2 AS

SELECT \*

FROM commune c

WHERE c.dep IN (

SELECT dep

FROM departement\_zone\_2

);

-- ggmd3 / Marseille

CREATE TABLE commune\_zone\_3 AS

SELECT \*

FROM commune c

WHERE c.dep IN (

SELECT dep

FROM departement\_zone\_3

);

**C4**

CREATE TABLE remote\_personne AS SELECT idp, age FROM personne;

-- ggmd1 / Paris

CREATE TABLE age\_personne AS

SELECT idp, age

FROM personne;

**C5**

CREATE TABLE remote\_personne AS SELECT p.idp, p.nom, p.prenoms, p.datenaiss, p.lieunaiss FROM personne p JOIN commune c ON c.com = p.lieunaiss JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 1;

CREATE TABLE remote\_personne AS SELECT p.idp, p.nom, p.prenoms, p.datenaiss, p.lieunaiss FROM personne p JOIN commune c ON c.com = p.lieunaiss JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 2;

CREATE TABLE remote\_personne AS SELECT p.idp, p.nom, p.prenoms, p.datenaiss, p.lieunaiss FROM personne p JOIN commune c ON c.com = p.lieunaiss JOIN departement d ON c.dep = d.dep JOIN region r ON r.reg = d.reg WHERE zone = 3;

===> c’est pareil en mieux

CREATE TABLE remote\_personne\_naissance AS SELECT p.idp, p.nom, p.prenoms, p.datenaiss, p.lieunaiss FROM personne p JOIN remote\_commune c ON c.com = p.lieunaiss ;

-- ggmd1 / Paris

CREATE TABLE personne\_info\_naissance\_zone\_1 AS

SELECT idp, nom, prenoms, datenaiss, lieunaiss

FROM personne;

CREATE TABLE personne\_naissance\_zone\_1 AS

SELECT \* FROM personne\_info\_naissance\_zone\_1 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_1 c

WHERE p.lieunaiss = c.com

);

-- ggmd2 / Lyon

CREATE TABLE personne\_info\_naissance\_zone\_2 AS

SELECT idp, nom, prenoms, datenaiss, lieunaiss

FROM personne;

CREATE TABLE personne\_naissance\_zone\_2 AS

SELECT \* FROM personne\_info\_naissance\_zone\_2 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_2 c

WHERE p.lieunaiss = c.com

);

-- ggmd3 / Marseille

CREATE TABLE personne\_info\_naissance\_zone\_3 AS

SELECT idp, nom, prenoms, datenaiss, lieunaiss

FROM personne;

CREATE TABLE personne\_naissance\_zone\_3 AS

SELECT \* FROM personne\_info\_naissance\_zone\_3 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_3 c

WHERE p.lieunaiss = c.com

);

**C6**

CREATE TABLE remote\_personne\_deces AS SELECT p.idp, p.nom, p.prenoms, p.datedeces, p.lieudeces FROM personne p JOIN remote\_commune c ON c.com = p.lieudeces ;

-- ggmd1 / Paris

CREATE TABLE personne\_info\_deces\_zone\_1 AS

SELECT idp, nom, prenoms, datedeces, lieudeces

FROM personne;

CREATE TABLE personne\_deces\_zone\_1 AS

SELECT \* FROM personne\_info\_deces\_zone\_1 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_1 c

WHERE p.lieudeces = c.com

);

-- ggmd2 / Lyon

CREATE TABLE personne\_info\_deces\_zone\_2 AS

SELECT idp, nom, prenoms, datedeces, lieudeces

FROM personne;

CREATE TABLE personne\_deces\_zone\_2 AS

SELECT \* FROM personne\_info\_deces\_zone\_2 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_2 c

WHERE p.lieudeces = c.com

);

-- ggmd3 / Marseille

CREATE TABLE personne\_info\_deces\_zone\_3 AS

SELECT idp, nom, prenoms, datedeces, lieudeces

FROM personne;

CREATE TABLE personne\_deces\_zone\_3 AS

SELECT \* FROM personne\_info\_deces\_zone\_3 p

WHERE EXISTS (

SELECT 1

FROM commune\_zone\_3 c

WHERE p.lieudeces = c.com

);

CREATE TABLE personne\_deces\_zone\_1 AS SELECT p.idp, p.nom, p.prenoms, p.datedeces, p.lieudeces FROM personne p JOIN commune\_zone\_1 c ON c.com = p.lieudeces ;

**C - Mise en place de la réplication logique des mairies**

**1 - Importer la table *mairie* sur *ggmd3***

Sur les 3 VMs :

CREATE TABLE all\_mairie AS SELECT \* from mairie;

**2 - Créer une 'publication' des mairies locales**

ALTER SYSTEM SET wal\_level = logical;

sudo systemctl restart postgresql

CREATE PUBLICATION marseille\_mairie\_publisher FOR TABLE ONLY all\_mairie;

**3- Créer les subscriptions nécessaires sur *ggmd1* et *ggmd2* pour disposer d'une réplication logique des mairies sur ces sites.**

CREATE SUBSCRIPTION paris\_mairie\_subscriber CONNECTION 'host=192.168.246.227 dbname=postgres port=5432 user=replication\_user password=replication' PUBLICATION marseille\_mairie\_publisher;

CREATE SUBSCRIPTION lyon\_mairie\_subscriber CONNECTION 'host=192.168.246.227 dbname=postgres port=5432 user=etum2 password=etum2 sslmode=require' PUBLICATION marseille\_mairie\_publisher;

**4 - Depuis *ggmd3* modifier la date de mise à jour associée à la mairie de Villeurbanne et tester si la synchronisation s'est bien opérée sur tous les sites.**

CREATE USER replication\_user REPLICATION LOGIN CONNECTION LIMIT 5 PASSWORD 'replication';

**D - Interrogation des fragments**

CREATE SERVER small\_server FOREIGN DATA WRAPPER postgres\_fdw OPTIONS (host '192.168.246.150', dbname 'postgres', port'5432');

CREATE SERVER medium\_server FOREIGN DATA WRAPPER postgres\_fdw OPTIONS (host '192.168.246.146', dbname 'postgres', port'5432');

CREATE SERVER large\_server FOREIGN DATA WRAPPER postgres\_fdw OPTIONS (host '192.168.246.227', dbname 'postgres', port'5432');

CREATE USER MAPPING FOR postgres SERVER small\_server OPTIONS

(password\_required 'false');

CREATE USER MAPPING FOR postgres SERVER medium\_server OPTIONS

(password\_required 'false');

CREATE USER MAPPING FOR postgres SERVER large\_server OPTIONS

(password\_required 'false');

GRANT ALL PRIVILEGES ON FOREIGN SERVER small\_server TO postgres;

GRANT ALL PRIVILEGES ON FOREIGN SERVER medium\_server TO postgres;

GRANT ALL PRIVILEGES ON FOREIGN SERVER large\_server TO postgres;

CREATE SCHEMA small\_schema;

IMPORT FOREIGN SCHEMA public FROM SERVER small\_server INTO small\_schema;

CREATE SCHEMA medium\_schema;

IMPORT FOREIGN SCHEMA public FROM SERVER medium\_server INTO medium\_schema;

CREATE SCHEMA large\_schema;

IMPORT FOREIGN SCHEMA public FROM SERVER large\_server INTO large\_schema;

SELECT p.idp, p.nom, p.prenoms

FROM large\_schema.personne\_naissance\_zone\_3 p

JOIN large\_schema.commune\_zone\_3 c ON p.lieunaiss = c.com

JOIN large\_schema.departement\_zone\_3 d ON c.dep = d.dep

JOIN large\_schema.Marseille r ON d.reg = r.reg

JOIN medium\_schema.personne\_deces\_zone\_2 pd ON p.idp = pd.idp

JOIN medium\_schema.commune\_zone\_2 cd ON pd.lieudeces = cd.com

JOIN medium\_schema.departement\_zone\_2 dd ON cd.dep = dd.dep

JOIN medium\_schema.Lyon rd ON dd.reg = rd.reg

WHERE r.nom = 'Auvergne-Rhône-Alpes'

AND rd.nom = 'Pays de la Loire';

**E - Création des vues globales**

**REGION**

-- ggmd1 / Paris

CREATE VIEW global\_region AS

SELECT \* FROM Paris

UNION ALL

SELECT \* FROM medium.lyon

UNION ALL

SELECT \* FROM large.marseille;

-- ggmd2 / Lyon

CREATE VIEW global\_region AS

SELECT \* FROM Lyon

UNION ALL

SELECT \* FROM small.paris

UNION ALL

SELECT \* FROM large.marseille;

-- ggmd3 / Marseille

SELECT \* FROM marseille

UNION ALL

SELECT \* FROM small.paris

UNION ALL

SELECT \* FROM medium.lyon;

**DEPARTEMENT**

-- ggmd1 / Paris

SELECT \* FROM departement\_zone\_1

UNION ALL

SELECT \* FROM medium.departement\_zone\_2

UNION ALL

SELECT \* FROM large.departement\_zone\_3;

-- ggmd2 / Lyon

CREATE VIEW global\_departement AS

SELECT \* FROM departement\_zone\_2

UNION ALL

SELECT \* FROM small.departement\_zone\_1

UNION ALL

SELECT \* FROM large.departement\_zone\_3;

-- ggmd3 / Marseille

SELECT \* FROM departement\_zone\_3

UNION ALL

SELECT \* FROM small.departement\_zone\_1

UNION ALL

SELECT \* FROM medium.departement\_zone\_2;

**COMMUNE**

-- ggmd1 / Paris

CREATE VIEW global\_commune AS

SELECT \* FROM commune\_zone\_1

UNION ALL

SELECT \* FROM medium.commune\_zone\_2

UNION ALL

SELECT \* FROM large.commune\_zone\_3;

-- ggmd2 / Lyon

CREATE VIEW global\_commune AS

SELECT \* FROM commune\_zone\_2

UNION ALL

SELECT \* FROM small.commune\_zone\_1

UNION ALL

SELECT \* FROM large.commune\_zone\_3;

-- ggmd3 / Marseille

CREATE VIEW global\_commune AS

SELECT \* FROM commune\_zone\_3

UNION ALL

SELECT \* FROM small.commune\_zone\_1

UNION ALL

SELECT \* FROM medium.commune\_zone\_2;

**MAIRIE**

CREATE VIEW global\_mairie AS

SELECT \* FROM all\_mairie;

**PERSONNE**

-- ggmd1 / Paris

-- ggmd2 / Lyon

-- ggmd3 / Marseille