

- [Se connecter au serveur](#)

psql -U postgres

- [Création d'une DB b2dev](#)

create database b2dev

- [Se connecter à la DB d2dev](#)

\c b2dev

- [Création de la table client](#)

create table client (id serial **primary key**,
numeroClient **varchar**(20) **not null unique**,
nomclient **varchar**(250),
adresseClient **varchar**(250)) ;

- [Afficher les tables](#)

\d

- [Voir les DB](#)

\l

- [Insertion de données dans la table client](#)

insert into client (numeroClient,nomclient,adresseClient) **values** ('CLT00128', 'René','Rennes'),
('CLT00147', 'Norbert','Nantes'),('CLT14582', 'Claude','Clisson') ;

- [Afficher les enregistrements de la table client](#)

select * from client ;

- [Afficher le client dont id=2](#)

select * from client **where** id=2 ;

- [Afficher tous les clients dont le nom commence par R](#)

select * from client **where** nomClient **like** 'R%' ;

- [Afficher tous les clients dont le nom contient la lettre a](#)

select * from client **where** nomClient **like** '%a%' ;

- [Créer une table fournisseur à partir de la table client](#)

create table fournisseur **as select * from** client ;

- [Créer une table frs à partir de la table client sans les données](#)

create table frs **as** (**select** * **from** client **where** false) ;

- Vider la table fournisseur

delete from fournisseur ;

-- ou = truncate table fournisseur ;

- Supprimer la table fournisseur

drop table fournisseur ;

- Création d'une séquence numero_client_seq

create sequence numero_client_seq **start** 100 ;

- Afficher le numéro de séquence actuel

select nextval('numero_client_seq') ;

- Mettre comme valeur par défaut nextval('numero_client_seq') à la colonne numeroClient

alter table client **alter** numeroClient **set default** nextval('numero_client_seq') ;

insert into client (nomClient,adresseClient) **values** ('Jacques','Valenciennes'),
('Rakoto','Marseille') ;

Select to_char(158,'000000000') ;

Select 'CLT' || to_char(158,'000000000') ;

Select 'CLT' || trim(to_char(158,'000000000')) ;

Select 'CLT' || trim(to_char(nextval('numero_client_seq'),'0000000')) ;

Alter table client **alter** numeroClient **set default** 'CLT' ||
trim(to_char(nextval('numero_client_seq'),'0000000')) ;

Insert into client (nomClient,adresseClient) **values** ('Marie','Bordeaux'),('Jules','Nantes') ;

Select * from client ;

- Création d'une fonction numéroté

create or replace function numeroter(prefix varchar(20), nom_sequence varchar(100)) returns
varchar(100) as
\$\$

```
begin
    return prefix || trim(to_char(nextval(nom_sequence),'000000'));
end;
```

\$\$language plpgsql;

Select numeroter('CLT','numero_client_seq') ;

- Création de la table commande

Create sequence numero_commande_seq start 1550 ;
create table commande(id serial primary key, numeroCommande varchar(20) not null unique
default numeroter('CDE','numero_commande_seq'), dateCommande timestamp, client_id int
not null, foreign key(client_id) references client(id)) ;

- Insertion de données dans la table commande

insert into commande (dateCommande,client_id) values ('2019/01/25',1), ('2019/02/25',2),
('2019/03/25',3), ('2019/04/25',4) ;

- Création de la table ligneCommande

create table ligneCommande(id serial **primary key**, commande_id **int not null**, article_id **int not
null**, quantite **decimal**(10,2), foreign key(commande_id) **references** commande(id), **foreign
key**(article_id) **references** article);

- Insertion de données

insert into lignecommande (commande_id,article_id, quantite) **values**(1,1,2456),(1,2,2050),
(1,3,1456) ;

--- Correction TP

Create or replace view v_ligne_commande as

Select c.numeroCommande as NumCde, c.dateCommande as DateCde, a.numeroArticle as
NumArt, a.designation, a.prixUnitaire as PU, l.quantite as QTE, l.quantite*a.prixUnitaire as
Montant from commande c, ligneCommande l, article a where l.commande_id=c.id and
l.article_id=a.id ;

-

Select v.nomcde as «NUMEROCOMMANDE », v.datecde as «DATECOMMANDE »,
sum(v.montant) as TOTAL from v_ligne_commande v group by v.numCde,v.datecde ;

- Creation d'un serv secondaire port 5433

initdb -D c:/test/serveurs/5433

- Changer le port à 5433 au fichier postgresql.conf

- Redémarrer le serveur

pg_ctl -D c:/test/serveurs/5433 -l journal5433 start

- Créer l'utilisateur postgres

createuser -p 5433 -s -P postgres

- Pour se connecter

psql -p 5433 -U postgres

- Sauvegarde de la bdd d2dev du serveur principal

pg_dump -U postgres b2dev > c:/test/Sauvegarde_b2dev.sql

■ Création d'un système de règle

- Créons d'abord la table log

Create table article_log(id integer, numeroArticle varchar(20), designation varchar(250),
ancienPrixUnitaire float, nouveauPrixUnitaire float, operation varchar(100), dateOperation
timestamp, operateur varchar(100)) ;

- Rule

create or replace rule article_update as on update to article where new.prixUnitaire <>
old.prixUnitaire do insert into article_log values
(old.id,old.numeroArticle,old.designation,old.prixUnitaire,new.prixUnitaire,'UPDATE',now(),current_user);

Update article set prixUnitaire=15.25 where id=5 ;

Select * from article_log

- Rule delete

Create or replace rule article_delete as on delete to article do insert into article_log values
(old.id,old.numeroArticle,old.designation,old.prixUnitaire, Null, 'DELETE', now() , current_user) ;

Insert into article (designation,prixUnitaire) values ('Gomme',2.50),('Regle 10cm',4) ;

Select * from article_log

Create or replace rule commande_insert as on insert to commande where (select id from client where id=new.client_id) is null do instead select 'Client introuvable' as Message ;

Insert into commande (dateCommande,client_id) values (now(),1000) ;

Select extract (year from c.dateCommande) as Annee,
a.numeroArticle as numArt, a.designation, a.prixUnitaire as PU,
sum(case when extract (month from c.dateCommande)=1 then l.quantite else 0 end) as Janv,
sum(case when extract (month from c.dateCommande)=2 then l.quantite else 0 end) as FEV,
sum(case when extract (month from c.dateCommande)=3 then l.quantite else 0 end) as Mars,
sum(case when extract (month from c.dateCommande)>3 then l.quantite else 0 end) as Autre,
l.quantite Total
from commande c, ligneCommande l, article a where l.commande_id=c.id and l.article_id=a.id
group by annee,designation,numArt,pu ;

- **Fonction if**

Select extract(year from c.dateCommande) as Annee, a.numeroArticle as numArt, a.designation,
aprixUnitaire as PU,
if(extract(month from c.datecommande)=1, l.quantite,0) as JANV,
if(extract(month from c.datecommande)=2, l.quantite,0) as FEV,
if(extract(month from c.datecommande)=3, l.quantite,0) as Mars,
if(extract(month from c.datecommande)=4, l.quantite,0) as Autre,
l.quantite as TOTAL

- **Creation de la fonction if**

Create or replace funtion if(condition boolean, ExVrai float, ExFausse float)
returns float as
\$\$

Begin

if(condition) then
return ExVrai ;

Else

```

        Return ExFausse ;
    end if ;

End ;
$$language plpgsql

```

- Utilisation de l'union all

```

where l.commande_id=c.id and extract(month from c.dateCommande)=1 union all

Select extract(year from c.dateCommande) as annee, l.article_id,
l.quantite as Fev, 0 as Janv, 0 as Mars, 0 as Autre
from commande c, lignecommande
where l.commande_id=c.id and extract(month from c.dateCommande)=2 union all

Select extract(year from c.dateCommande) as annee, l.article_id,
l.quantite as Mars, 0 as Fev, 0 as Janv, 0 as Autre
from commande c, lignecommande
where l.commande_id=c.id and extract(month from c.dateCommande)=3 union all

Select extract(year from c.dateCommande) as annee, l.article_id,
l.quantite as Autre, 0 as Fev, 0 as Mars, 0 as Janv
from commande c, lignecommande
where l.commande_id=c.id and extract(month from c.dateCommande) not in (1,2,3);

```

Create or replace view v_stat_vente as

```

Select v.annee,a.numeroArticle as numArt,a.designation,

a.prixUnitaire as PU, sum(v.janv)as JANV, sum(v.fev)as FEV, sum(v.mars)as MARS, sum(v.autre)as
AUTRE,

sum(v.janv+v.fev+v.mars+v.autre) as TOTAL

from article a, v_quantite_vendus v, where v.article_id=a.id
group by annee,numArt,designation,pu;

```

- Clause With

```

With v_article (designation,pu) as (select designation, prixUnitaire from article)
select * from v_article ;

```

Select v_qte (annee,article_id,janv,fev,mars,autre) as

(select extract(year from c.dateCommande) as annee,l.article_id,

Sum(If (extract(year from c.dateCommande)=1,l.quantite,0)),

Sum(If (extract(year from c.dateCommande)=2,l.quantite,0)),

Sum(If (extract(year from c.dateCommande)=3,l.quantite,0)),

Sum(If (extract(year from c.dateCommande)>3,l.quantite,0))

From commande c, ligneCommande l where l.commande_id=c.id

groupe by annee,article_id)

select v.annee, a.numeroArticle as numArt, a.designation, a.prixUnitaire as PU

v.janv,v.fev,v.mars,v.autre, (janv+fev+mars+autre) as total

from article a, v_qte v where v.article_id=a.id ;

with **v_janv**(annee, article_id, janv) as

(select extract(year from c.dateCommande) as annee,

l.article_id, sum(l.quantite)

from commande c, ligneCommande l

where l.commande_id=c.id

and extract(month from c.dateCommande)=1

group by annee, article_id),

v_fev(annee, article_id, fev) as

(select extract(year from c.dateCommande) as annee,

l.article_id, sum(l.quantite)

from commande c, ligneCommande l

where l.commande_id=c.id

and extract(month from c.dateCommande)=2

group by annee, article_id),

v_mars(annee, article_id, mars) as

(select extract(year from c.dateCommande) as annee,

```

l.article_id, sum(l.quantite)
from commande c, ligneCommande l
where l.commande_id=c.id
and extract(month from c.dateCommande)=3
group by annee, article_id),

```

```

v_autre(annee, article_id, autre) as
(select extract(year from c.dateCommande) as annee,
l.article_id, sum(l.quantite)
from commande c, ligneCommande l
where l.commande_id=c.id
and extract(month from c.dateCommande)>3
group by annee, article_id)

```

```

select v1.annee, a.numeroArticle as numArt, a.designation, janv, fev, mars, autre,
(janv+fev+mars+autre) as total

```

```

from article a
left join v_janv v1 on v1.article_id=a.id
left join v_fev v2 on v2.article_id=a.id
left join v_mars v3 on v3.article_id=a.id
left join v_autre v4 on v4.article_id=a.id;

```

- **Requête recursive**

Create table personne(id serial

Nom varchar(100), parent_id int, foreign key (parent_id) references personne(id)) ;

Insert into personne (nom, parent_id) values ('P11',1),('P12',1),('P13',1),
('P21',2),('P22',2),('P23',2), ('P31',2),('P32',3),('P33',3) ;

Insert into personne (nom,parent_id) values (('P211',13),('P212',13),('P213',13) ;

- **Requête**

With recursive famille(id,nom,arbre,niveau,parent_id) as

(select id,nom,nom || '|',0,parent_id, from personne where personne_id is null union all

Select p.id,p.nom,f.arbre || '|_' || p.nom,f.niveau+1,p.parent_id
from personne p, famille f where p.parent_id=f.id

)

Select * from famille ;

■ HERITAGE

- Creation de la table tiers

Create table tiers(id serial primary key, numeroTiers varchar(20) not null unique, nomTiers varchar(250), adresseTiers varchar(250)) ;

- Clients

Create table clients() inherits(tiers) ;

create table fournisseurs() inherits(tiers) ;

--insertion de données dans la table clients

Insert into clients (numeroTiers,nomTiers,adresseTiers) values ('CLTS','MIMI','Rennes') ;

Insert into fournisseurs (numeroTiers,nomTiers,adresseTiers) values ('FRE','SUPER U','Nantes') ;

Select * from only tiers ;

- Détacher client à la table tiers

Alter table clients no inherit tiers ;

- Insérer des données dans la table client tout en restant déconnecté à la table mère tiers

Insert into clients (numeroTiers,nomTiers,adresseTiers) values ('CLT00128','TOTO','Bordeaux') ;

- Rattacher à nouveau clients à tiers

Alter table clients inherit tiers ;.

Correction TP

Select a.numeroArticle as code, a.designation as article, a.prixRevient as PR,

Sum(if(c.typeCommande='a',l.quantite,0)) as appro,

Sum(if(c.typeCommande='v',l.quantite,0)) as vente,

Sum(if(c.typeCommande='a',+1,-1)*l.quantite) as stock,

Sum(l.quantite*a.prixRevient) as valeur

From commande c, ligneCommande l, article a
where l.commande_id=c.id and l.article_id=a.id

Group by code,article,pr ;

--

With v_qte_appro(article_id,quantite) as

(select article_id, sum(quantite) from ligneAppro group by article_id),
v_qte_vente(article_id,quantite) as

(select article_id, sum(quantite) from ligneVente group by article_id),

Select a.numeroArticle as code, a.designation as article, aprixrevient as PR, v1.quantite as
APPRO, v2.quantite as VENTE

- [Sauvegarde](#)

Pg_dump -U postgres b2dev >c:/test/b2dev_heritage

5434_start

Createdb -p 5434 -U postgres b2dev_heritage

Psql -p 5434 -U postgres b2dev_heritage < c:/test/b2dev_heritage

- [Schema](#)

- [Afficher les schémas existants](#)

\dn

- [Creation schémas vente et appro](#)

Create schema vente ;

Create schema appro ;

- [Savoir le schema en cours](#)

Show search_path ;

- [Se déplacer dans le schema vente](#)

Set search_path to vente ;

- [Revenir au schema public](#)

Set search_path to public ;

- [Déplacer les tables relatives à la gestion vente dans le schema vente](#)

```
Alter table client set schema vente ;
Alter table vente set schema vente ;
Alter table vente set schema vente ;
Alter table ligneVente set schema vente ;
```

- **De même pour appro**

```
Alter table fournisseur set schema appro ;
```

```
Alter table appro set schema appro ;
```

```
Alter table appro set schema appro ;
```

```
Alter table ligneAppro set schema appro ;
```

```
--
```

```
Set search_path to public, appro, vente ;
with v_qte_appro(article_id,quantite) as
```

```
    (select article_id,coalesce (sum(quantite),0) from ligneAppro group by
v_qte_vente(article_id,quantite) as
```

```
    (select article_id, coalesce (sum(quantite),0) from ligneVente group by
```

```
Select a.numeroArticle as code, a.designation as article,
a.prixRevient as PR, v1.quantite as APPRO, v2.quantite as VENTE,
(coalesce(v1.quantite,0)-coalesce(v2.quantite,0)) as STOCK,
(coalesce(v1.quantite,0)-coalesce(v2.quantite,0))*a.prixRevient as VENTE
```

```
From article a
```

```
Left join v_qte_appro v1 on v1.article_id=a.id ;
```

```
Left join v_qte_vente v2 on v2.article_id=a.id ;
```

- **Gestion utilisateurs**

- [Creation d'un utilisateur](#)

```
Createuser -U postgres -P paul
```

- [ou en interne](#)

```
Create user paul password '1234' ;
```

- [Donner le droit select à la table article à paul](#)

```
grant select on article to paul ;
```

- [Donner le droit insertion à la table article à paul](#)

```
grant insert on article to paul ;
```

- [Insertion de données dans la table article en tant que paul](#)

```
Insert into article (designation,prixUnitaire,prixRevient) values ('Coton',2.50',1.20) ;
```

- [ERREUR : droit refusé pour la séquence article_id_seq](#)

Grant usage sequence on article_id_seq to paul,

-ou

Grant usage on all sequences in schema public to paul,

Grant execute on all functions in schema public to paul,

- [Droit select sur vente.client à paul](#)

Grant usage on schema vente to paul ;

Grant select on vente.client to paul ;

- [Donner tout les droit à la table vente du schema vente à paul](#)

Grant all privileges on vente.vente to paul ;

- [Creation d'un groupe comptoir ;](#)

Create rôle comptoir,

- [Donner le droit d'utiliser le schema vente au groupe comptoir](#)

Grant usage on schema vente to comptoir ;

grant select,insert,update on vente.vente to comptoir ;

grant select,insert,update on lignevente.vente to comptoir ;

grant select on vente.client to comptoir ;

- [Creation d'un utilisateur marie dans le groupe comptoir](#)

Create user marie password '1234' in role comptoir ;

- [Enlever marie au groupe comptoir](#)

Revoke comptoir from marie ;

