

David TREMBLAY 1748125

Dominique PICHÉ 1766981

Travail Pratique #1

INF3710

Bases de données

Groupe 02

Département de génie informatique et génie logiciel

École polytechnique de Montréal

6 septembre 2017

Expériences à faire

Création d'un usager et d'une base des données

1)

```
[root@localhost ~]# su - postgres
-bash-4.2$ createuser -d -P pichedominique
Enter password for new role:
Enter it again:
```

2)

```
-bash-4.2$ createdb -O pichedominique Capo
```

3)

```
-bash-4.2$ psql Capo
psql (9.6.4)
Type "help" for help.

Capo=# _
```

Création des tables

1) Création des 4 tables

```
Capo=# CREATE TABLE Agents (aid CHAR(3), aname VARCHAR(20), city VARCHAR(25), percent int CONSTRAINT
check_percent CHECK (percent BETWEEN 0.00 AND 10.00), CONSTRAINT agent_pk PRIMARY KEY (aid));
CREATE TABLE

Capo=# CREATE TABLE Customers (cid CHAR(4) CONSTRAINT cust_pk PRIMARY KEY, cname VARCHAR(20), city V
ARCHAR(25), discnt int CONSTRAINT check_discnt CHECK (discnt BETWEEN 0.00 AND 15.00));
CREATE TABLE

Capo=# CREATE TABLE Products (pid CHAR(3), pname VARCHAR(20), city VARCHAR(25), quantity int CONSTRA
INT check_quantity CHECK (quantity >= 0), price int CONSTRAINT check_price CHECK (price >= 0.00), PRI
MARY KEY (pid));
CREATE TABLE

Capo=# CREATE TABLE Orders (ordno int PRIMARY KEY, month CHAR(3) CHECK (month IN ('jan', 'feb', 'mar
', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec')), cid CHAR(4) NOT NULL REFERENCES
Customers, aid CHAR(3) CONSTRAINT fk_aid REFERENCES Agents, pid CHAR(3) NOT NULL, CONSTRAINT fk_pid
FOREIGN KEY (pid) REFERENCES Products, qty int CHECK (qty >= 0), dollars int CHECK (dollars >= 0.00)
);
CREATE TABLE
```

Liste des tables

```
Capo=# \d

          List of relations
Schema | Name       | Type  | Owner
-----+-----+-----+-----
public | agents     | table | postgres
public | customers  | table | postgres
public | orders     | table | postgres
public | products   | table | postgres
(4 rows)
```

2)

```
Capo=# SELECT * FROM Customers;
cid | cname | city | discnt
-----+-----+-----+-----
(0 rows)
```

La commande sert à afficher le contenu de la table Customers. Comme aucune rangée n'a été insérée dans Customers, aucune donnée n'est présente. La seule information affichée est le nom des colonnes.

Peuplement des tables

1) Voici un exemple de commande ayant utilisée pour peupler nos tables :

```
Capo=# INSERT INTO Orders VALUES (1016, 'jan', 'c006', 'a01', 'p01', 1000, 500.00);
```

2)

```
Capo=# SELECT * FROM Customers;
cid | cname | city | discnt
-----+-----+-----+-----
c001 | Tip Top | Duluth | 10
c002 | Basics | Dallas | 12
c003 | Allied | Dallas | 8
c004 | ACME | Duluth | 8
c006 | ACME | Kyoto | 0
(5 rows)

Capo=# SELECT * FROM Agents;
aid | aname | city | percent
-----+-----+-----+-----
a01 | Smith | New York | 6
a02 | Jones | Newark | 6
a03 | Brown | Tokyo | 7
a04 | Gray | New York | 6
a05 | Otasi | Duluth | 5
a06 | Smith | Dallas | 5
(6 rows)

Capo=# SELECT * FROM Products;
pid | pname | city | quantity | price
-----+-----+-----+-----+-----
p02 | brush | Newark | 203000 | 1
p03 | razor | Duluth | 150600 | 1
p01 | comb | Dallas | 111400 | 1
p04 | pen | Duluth | 125300 | 1
p05 | pencil | Dallas | 221400 | 1
p06 | folder | Dallas | 123100 | 2
p07 | case | Newark | 100500 | 1
(7 rows)
```

```
Capo=# SELECT * FROM Orders;
ordno | month | cid | aid | pid | qty | dollars
-----+-----+-----+-----+-----+-----+-----
 1011 | jan   | c001 | a01 | p01 | 1000 | 450
 1012 | jan   | c001 |    | p01 | 1000 | 450
 1019 | feb   | c001 | a02 | p02 | 400  | 180
 1017 | feb   | c001 | a06 | p03 | 600  | 540
 1018 | feb   | c001 |    | p04 | 600  | 540
 1023 | mar   | c001 | a04 | p05 | 500  | 450
 1022 | mar   | c001 | a05 | p06 | 400  | 720
 1025 | apr   | c001 | a05 | p07 | 800  | 720
 1013 | jan   | c002 | a03 | p03 | 1000 | 880
 1026 | may   | c002 | a05 | p03 | 800  | 704
 1015 | jan   | c003 | a03 | p05 | 1200 | 1104
 1014 | jan   | c003 | a03 | p05 | 1200 | 1104
 1021 | feb   | c004 | a06 | p01 | 1000 | 460
 1016 | jan   | c006 | a01 | p01 | 1000 | 500
 1020 | feb   | c006 | a03 | p07 | 600  | 600
 1024 | mar   | c006 | a06 | p01 | 800  | 400
(16 rows)
```

3)

```
Capo=# INSERT INTO Customers VALUES ('c001', 'David', 'Quebec', 5.00);
ERROR:  duplicate key value violates unique constraint "cust_pk"
DETAIL:  Key (cid)=(c001) already exists.
```

- 4) La commande n'a pas été exécutée, le customer n'a pas été ajouté au tableau, et celui-ci reste identique à son état avant l'essai de l'insertion, à cause de l'atomicité des opérations.
- 5) La contrainte que chaque rangée doit avoir un customer id unique comme clé primaire n'est pas respectée avec cette insertion, alors elle n'est pas exécutée (il y aurait deux clients avec cid = 'c001').
- 6) Nous avons utilisé le customer id c005 qui est unique pour corriger la situation :

```
Capo=# INSERT INTO Customers VALUES ('c005', 'David', 'Quebec', 5.00);
INSERT 0 1
```

Suppression des tables

7)

```
Capo=# DROP TABLE Customers;
ERROR:  cannot drop table customers because other objects depend on it
DETAIL:  constraint orders_cid_fkey on table orders depends on table customers
HINT:  Use DROP ... CASCADE to drop the dependent objects too.
```

- 8) La suppression de la table Customers a échoué, la table existe encore.

- 9) Étant donné que la table Orders a des dépendances sur la table Customers, la table Customers ne peut être supprimée sans spécifier ce qui se passe avec les tables dépendantes (on a que orders.cid réfère à Customers).

10)

```
Capo=# DROP TABLE Orders;
DROP TABLE
```

- 11) La table a été correctement supprimée :

```
Capo=# DROP TABLE Orders;
DROP TABLE
Capo=# \d
          List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | agents    | table | postgres
public | customers | table | postgres
public | products  | table | postgres
(3 rows)
```

- 12) Aucune autre table ne dépend de la table Orders, alors il n'existe aucune contrainte l'empêchant d'être supprimée.

13)

```
Capo=# DROP TABLE Customers CASCADE;
DROP TABLE
Capo=# \d
          List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | agents    | table | postgres
public | products  | table | postgres
(2 rows)
```

- 14) La table Customers a été supprimée adéquatement. Avec l'utilisation du mot clé CASCADE, tous les objets qui dépendent de la table Customers ont également été supprimés. Les tables Agents et Products sont encore intactes, car il n'y avait aucune dépendance sur la table Customers. Cependant, si la table Orders existait encore, toutes les entrées auraient été supprimées, car elles dépendaient toutes de Customers. On aurait pu supprimer Customers sans CASCADE, car la table Orders est déjà supprimée.

15)

```
Capo=# DROP TABLE AGENTS CASCADE;  
DROP TABLE  
Capo=# DROP TABLE Products CASCADE;  
DROP TABLE  
Capo=# \d  
No relations found.  
Capo=# _
```

Suppression database et user

16) La requête est **dropdb Capo;**

```
Capo=# \q  
-bash-4.2$ dropdb Capo;  
-bash-4.2$ psql Capo;  
psql: FATAL: database "Capo" does not exist  
-bash-4.2$
```

17) La requête est **dropuser pichedominique;** (notre user crée au départ)

```
-bash-4.2$ dropuser pichedominique  
-bash-4.2$ createdb -O pichedominique Capo;  
createdb: database creation failed: ERROR: role "pichedominique" does not exist  
-bash-4.2$ _
```