

David TREMBLAY 1748125

Dominique PICHÉ 1766981

Laboratoire 2

INF3710

Bases de données

Groupe 02

Département de génie informatique et génie logiciel

École polytechnique de Montréal

25 septembre 2017

Création de la BD et des tables

Base de données:

```
-bash-4.2$ createdb -O pichedominique TP2
-bash-4.2$ psql TP2
psql (9.6.4)
Type "help" for help.

TP2=# _
```

Tables:

```
TP2=# CREATE TABLE COMPAGNIE (
TP2(# nom VARCHAR(20) CONSTRAINT COMP_PK PRIMARY KEY,
TP2(# ville VARCHAR(30) CONSTRAINT COMP_VILLE_UQ UNIQUE,
TP2(# province VARCHAR (30),
TP2(# pays VARCHAR(30)
TP2(# );
CREATE TABLE
TP2=# CREATE TABLE PILOTE (
TP2(# brevet CHAR(8) CONSTRAINT PILOTE_PK PRIMARY KEY,
TP2(# nom CHAR(30) CONSTRAINT PILOTE_NOM_NN NOT NULL,
TP2(# nbrHVol Integer CONSTRAINT PILOTE_NBRHVOL_NN NOT NULL,
TP2(# compa VARCHAR(20) CONSTRAINT PILOTE_COMP_FK REFERENCES COMPAGNIE(nom) ON DELETE SET NULL,
TP2(# CONSTRAINT PILOTE_UQ UNIQUE(nom, compa)
TP2(# );
CREATE TABLE
TP2=# CREATE TABLE AVION (
TP2(# matricule CHAR(6) CONSTRAINT AVION_PK PRIMARY KEY,
TP2(# typeAvion VARCHAR(20),
TP2(# nbrHVol INTEGER CONSTRAINT AVION_NBRHVOL_NN NOT NULL,
TP2(# proprio VARCHAR(20) CONSTRAINT AVION_PROP_FK REFERENCES COMPAGNIE(nom) ON DELETE SET NULL
TP2(# );
CREATE TABLE
TP2=# CREATE TABLE AFFRETEMENT (
TP2(# compaAff VARCHAR(20) CONSTRAINT AFF_COMP_FK REFERENCES COMPAGNIE(nom),
TP2(# avion CHAR(6) CONSTRAINT AFF_AVION_FK REFERENCES AVION(matricule),
TP2(# dateAff DATE,
TP2(# prix DECIMAL(9,2) CONSTRAINT AFF_PRIX_NN NOT NULL,
TP2(# CONSTRAINT AFF_PK PRIMARY KEY (compaAff, avion, dateAff)
TP2(# );
CREATE TABLE
TP2=# \d
                List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | affretement    | table | postgres
 public | avion          | table | postgres
 public | compagnie      | table | postgres
 public | pilote         | table | postgres
(4 rows)
```

Suppression d'une table

1.

```
TP2=# DROP TABLE AVION;  
ERROR:  cannot drop table avion because other objects depend on it  
DETAIL:  constraint aff_avion_fk on table affretement depends on table avion  
HINT:   Use DROP ... CASCADE to drop the dependent objects too.
```

La suppression n'est pas permise, car d'autres tables ont des dépendances sur la table avion. La contrainte aff_avion_fk de la table affretement dépend de la table avion. Il faudrait utiliser le mot clé cascade après le nom de la table, ce qui détruirait ces dépendances.

2.

On peut utiliser le mot clé cascade après le nom de la table à supprimer. La commande est alors : DROP TABLE AVION CASCADE;

Ainsi, seuls les éléments dépendants seront détruits en plus de la table.

```
TP2=# DROP TABLE AVION CASCADE;  
NOTICE:  drop cascades to constraint aff_avion_fk on table affretement  
DROP TABLE  
TP2=#
```

3.

Trois tables restantes:

```
TP2=# \d  
  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | affretement | table | postgres  
public | compagnie | table | postgres  
public | pilote | table | postgres  
(3 rows)  
  
TP2=#
```

Suppression des tables restantes :

```
TP2=# DROP TABLE AFFretement CASCADE  
TP2=# ;  
DROP TABLE  
TP2=# DROP TABLE COMPAGNIE CASCADE;  
NOTICE:  drop cascades to constraint pilote_comp_fk on table pilote  
DROP TABLE  
TP2=# DROP TABLE PILOTE CASCADE;  
DROP TABLE  
TP2=# \d  
No relations found.  
TP2=#
```

Peuplement des tables :

```
TP2=# SELECT * FROM COMPAGNIE; SELECT * FROM PILOTE; SELECT * FROM AVION; SELECT * FROM AFFRETEMENT;
```

nom	ville	province	pays
Air Canada	Montreal	Quebec	Canada
West Jet	Toronto	Ontario	Canada
Air France	Paris	Ile de France	France
American Airlines	Albany	New York	USA
Alitalia	Rome	Rome	Italie

(5 rows)

brevet	nom	nrhvol	compa
BR21675	Jean Lemieux	965	Air Canada
BR21431	Pierre Lheureux	543	Air Canada
BR45665	Serge Paris	705	Air France
BR35332	Marco Rossi	801	Alitalia
BR22445	John Steinpack	433	West Jet
BR26587	George McDonald	559	West Jet
BR28811	Sam Withney	606	Air Canada
BR45321	Sami Zidane	542	Air France

(8 rows)

matricule	typeavion	nrhvol	proprio
A4327V	Boeing 717	245	Air Canada
A7602S	Boeing 737	312	West Jet
A1329K	Airbus A320	409	Air France

(3 rows)

compaaff	avion	dateaff	prix
Air Canada	A4327V	2008-05-03	15750.00
Air Canada	A4327V	2008-05-04	12050.00
West Jet	A7602S	2008-05-03	10500.00

(3 rows)

Valeur par défaut d'une colonne

4.

La colonne à laquelle nous prescrivons une valeur par défaut est le nom de la compagnie (compa). La commande utilisée est : ALTER TABLE pilote ALTER compa SET DEFAULT 'Air Canada';

```
TP2=# ALTER TABLE Pilote ALTER compa SET DEFAULT 'Air Canada';
ALTER TABLE
```

Exemple avec l'ajout nouveau pilote avec une compagnie par défaut:

```
TP2=# INSERT INTO PILOTE VALUES ('BR99999', 'Jacques Martin', 666);
INSERT 0 1
TP2=# SELECT * FROM PILOTES;
ERROR:  relation "pilotes" does not exist
LINE 1: SELECT * FROM PILOTES;
                    ^
TP2=# SELECT * FROM pilote;
```

brevet	nom	nbrhvol	compa
BR21675	Jean Lemieux	965	Air Canada
BR21431	Pierre Lheureux	543	Air Canada
BR45665	Serge Paris	705	Air France
BR35332	Marco Rossi	801	Alitalia
BR22445	John Steinpack	433	West Jet
BR26587	George McDonald	559	West Jet
BR28811	Sam Withney	606	Air Canada
BR45321	Sami Zidane	542	Air France
BR99999	Jacques Martin	666	Air Canada

(9 rows)

```
TP2=#
```

Dans cet exemple, les nouveaux pilotes travaillent pour Air Canada, sauf avis contraire.

Contrainte NOT NULL imposée à une colonne

5.

```
TP2=# INSERT INTO PILOTE VALUES ('BR99998', 'Bracques Nartin');
ERROR:  null value in column "nbrhvol" violates not-null constraint
DETAIL:  Failing row contains (BR99998 , Bracques Nartin , null, Air Canada).
TP2=#
```

L'insertion est rejetée, car il n'y a pas de valeur pour le nombre d'heures de vol, violant la contrainte. Dans ce contexte, il est impossible qu'un pilote sans heures de vol se fasse engager, et comme l'information sur le nombre d'heures de vol est cruciale dans l'évaluation des effectifs, l'absence de cette information n'est pas permise, et l'utilisation de la contrainte NOT NULL accomplit ce but. Il aurait également été impossible d'insérer un pilote sans nom.

Contrainte d'unicité

6.

Il y a une contrainte d'unicité sur la colonne des villes dans la table des compagnies, car il ne peut y avoir plus d'une compagnie aérienne par ville selon le schéma.

Exemple d'un ajout de compagnie dans la ville de Montréal:

```
TP2=# SELECT * FROM compagnie;
      nom      | ville  | province  | pays
-----+-----+-----+-----
Air Canada    | Montreal | Quebec    | Canada
West Jet      | Toronto | Ontario   | Canada
Air France    | Paris   | Ile de France | France
American Airlines | Albany | New York   | USA
Alitalia      | Rome    | Rome       | Italie
(5 rows)

TP2=# INSERT INTO COMPAGNIE VALUES ('Air Poly', 'Montreal', 'Quebec', 'Canada');
ERROR:  duplicate key value violates unique constraint "comp_ville_uq"
DETAIL:  Key (ville)=(Montreal) already exists.
TP2=#
```

Comme une compagnie est déjà basée à Montréal (Air Canada), Air Poly ne peut s'y baser également. L'insertion est rejetée. La différence avec NOT NULL est que la ville peut ne pas être spécifiée (donc NULL), mais ne peut pas être identique à la ville d'une autre rangée. Dans le cas de NOT NULL, la ville ne peut être NULL, mais elle peut être identique à celle d'une autre rangée. Si on avait des contraintes NOT NULL et UNIQUE sur la même colonne, on aurait un candidat de KEY.

```
TP2=# INSERT INTO COMPAGNIE (nom, province, pays) VALUES ('Air Poly', 'Quebec', 'Canada');
INSERT 0 1
TP2=# SELECT * FROM compagnie;
      nom      | ville  | province  | pays
-----+-----+-----+-----
Air Canada    | Montreal | Quebec    | Canada
West Jet      | Toronto | Ontario   | Canada
Air France    | Paris   | Ile de France | France
American Airlines | Albany | New York   | USA
Alitalia      | Rome    | Rome       | Italie
Air Poly      |         | Quebec     | Canada
(6 rows)

TP2=#
```

7.

Dans le cas d'une contrainte d'unicité sur deux colonnes, comme c'est le cas dans la table pilote avec les colonnes nom et compa, la combinaison des valeurs dans les deux colonnes pour une rangée donnée doit être unique, c'est-à-dire qu'on ne peut avoir deux personnes avec le même nom travaillant pour la même compagnie, mais on peut avoir deux personnes avec le même nom travaillant pour deux compagnies différentes, ou bien deux personnes avec des noms différents travaillant pour la même compagnie.

brevet	nom	nbrhvol	compa
BR21675	Jean Lemieux	965	Air Canada
BR21431	Pierre Lheureux	543	Air Canada
BR45665	Serge Paris	705	Air France
BR35332	Marco Rossi	801	Alitalia
BR22445	John Steinpach	433	West Jet
BR26587	George McDonald	559	West Jet
BR28811	Sam Withney	606	Air Canada
BR45321	Sami Zidane	542	Air France
BR99999	Jacques Martin	666	Air Canada

(9 rows)

```

TP2=# INSERT INTO PILOTE VALUES ('JR21675', 'Jean Lemieux', 222, 'Air Canada');
ERROR:  duplicate key value violates unique constraint "pilote_uq"
DETAIL:  Key (nom, compa)=(Jean Lemieux, Air Canada) already exists.
TP2=# INSERT INTO PILOTE VALUES ('BM21675', 'Jean Lemieux', 965, 'Air Poly');
INSERT 0 1

```

Contrainte de clé primaire

8.

```

TP2=# INSERT INTO COMPAGNIE VALUES ('Air Canada', 'Montreallissimo', 'Quebecciocci', 'Canattada');
ERROR:  duplicate key value violates unique constraint "comp_pk"
DETAIL:  Key (nom)=(Air Canada) already exists.
TP2=# INSERT INTO compagnie (ville, province, pays) VALUES ('Laval', 'Quebec', 'Canada');
ERROR:  null value in column "nom" violates not-null constraint
DETAIL:  Failing row contains (null, Laval, Quebec, Canada).

```

Les deux manières de créer une erreur sont de soit utiliser la même clé primaire que pour une autre compagnie (par exemple, nommer une deuxième compagnie 'Air Canada'), ou bien de simplement tenter une insertion sans spécifier de valeur pour la clé primaire. Comme cette deuxième manière de procéder laisserait la clé NULL, l'erreur est provoquée. Chacun de ces deux cas est testé ci-dessus.

9.

Dans la table affrètement, comme aucune des colonnes n'est obligatoirement unique, il faut se servir d'une clé composée des valeurs de plusieurs colonnes afin d'obtenir une clé unique à chaque rangée. La clé est composée du nom de la compagnie d'affrètement, de l'avion spécifique concerné ainsi que de la date concernée.

Cette clé est irréductible, c'est-à-dire qu'on ne peut réduire le nombre de colonnes, chacune de ces informations est cruciale à la bonne identification de l'affrètement.

Dans l'exemple ci-dessous, une des colonnes de la clé est laissée nulle (la date). L'insertion est rejetée, car une colonne faisant partie d'une clé composée ne peut être NULL.

```

TP2=# INSERT INTO AFFRETEMENT (compaaff, avion, prix) VALUES ('Air Canada', 'A4327V', 12889.00);
ERROR:  null value in column "dateaff" violates not-null constraint
DETAIL:  Failing row contains (Air Canada, A4327V, null, 12889.00).

```

Dans l'exemple ci-dessous, l'insertion est rejetée car les trois valeurs dans les colonnes composant la clé sont identiques aux trois valeurs de ces mêmes colonnes d'une rangée préexistante.

```
TP2=# INSERT INTO AFFRETEMENT VALUES ('Air Canada', 'A4327V', to_date('03/05/2008', 'DD/MM/YYYY'), 14290.99);
ERROR:  insert or update on table "pilote" violates unique constraint "aff_pk"
DETAIL:  Key (compaaff, avion, dateaff)=(Air Canada, A4327V, 2008-05-03) already exists.
```

Contrainte de clé externe

10.

La clé externe à laquelle fait référence une nouvelle entrée dans une table doit appartenir à une entrée de la table externe. Par exemple, dans la table pilote, une nouvelle entrée doit avoir un nom de compagnie valide dans la table compagnie. Si on tente d'insérer un pilote avec un nom de compagnie n'existant pas dans la table compagnie, l'insertion provoquera une erreur et sera rejetée.

```
TP2=# INSERT INTO PILOTE VALUES ('BR265', 'Jean-Jacques Lemieux-Mieux', 965, 'Air Jong-Un');
ERROR:  insert or update on table "pilote" violates foreign key constraint "pilote_comp_fk"
DETAIL:  Key (compa)=(Air Jong-Un) is not present in table "compagnie".
```

Ici, comme Air Jong-Un n'est pas une compagnie valide, l'insertion de Jean-Jacques Lemieux-Mieux est rejetée.

Requêtes

12.

Requête 1 :

```
TP2=# SELECT *
TP2=# FROM COMPAGNIE C, PILOTE P
TP2=# WHERE C.nom=P.comp ;
```

nom	ville	province	pays	brevet	nom	nbrhvol	compa
Air Canada	Montreal	Quebec	Canada	BR21675	Jean Lemieux	965	Air Canada
Air Canada	Montreal	Quebec	Canada	BR21431	Pierre Lheureux	543	Air Canada
Air France	Paris	Ile de France	France	BR45665	Serge Paris	705	Air France
Alitalia	Rome	Rome	Italie	BR35332	Marco Rossi	801	Alitalia
West Jet	Toronto	Ontario	Canada	BR22445	John Steinpack	433	West Jet
West Jet	Toronto	Ontario	Canada	BR26587	George McDonald	559	West Jet
Air Canada	Montreal	Quebec	Canada	BR28811	Sam Withney	606	Air Canada
Air France	Paris	Ile de France	France	BR45321	Sami Zidane	542	Air France
Air Canada	Montreal	Quebec	Canada	BR99999	Jacques Martin	666	Air Canada
Air Poly		Quebec	Canada	BM21675	Jean Lemieux	965	Air Poly

```
(10 rows)
TP2=#
```

Cette requête, après un produit cartésien des tableaux compagnie et pilote, sélectionne les concaténations de toutes les rangées des tableaux compagnie et pilote ou le nom de la compagnie dans le tableau pilote est identique au nom de la compagnie dans le tableau compagnie. Comme la requête est un SELECT *, il n'y a aucun tri de l'information affichée, alors toutes les colonnes sont affichées pour chaque concaténation de rangées.

Requête 2 :

nom	ville	province	pays	brevet	nom	nbrhvol	compa
Air Canada	Montreal	Quebec	Canada	BR21675	Jean Lemieux	965	Air Canada
West Jet	Toronto	Ontario	Canada	BR21675	Jean Lemieux	965	Air Canada
Air France	Paris	Ile de France	France	BR21675	Jean Lemieux	965	Air Canada
American Airlines	Albany	New York	USA	BR21675	Jean Lemieux	965	Air Canada
Alitalia	Rome	Rome	Italie	BR21675	Jean Lemieux	965	Air Canada
Air Poly		Quebec	Canada	BR21675	Jean Lemieux	965	Air Canada
Air Canada	Montreal	Quebec	Canada	BR21431	Pierre Lheureux	543	Air Canada
West Jet	Toronto	Ontario	Canada	BR21431	Pierre Lheureux	543	Air Canada
Air France	Paris	Ile de France	France	BR21431	Pierre Lheureux	543	Air Canada
American Airlines	Albany	New York	USA	BR21431	Pierre Lheureux	543	Air Canada
Alitalia	Rome	Rome	Italie	BR21431	Pierre Lheureux	543	Air Canada
Air Poly		Quebec	Canada	BR21431	Pierre Lheureux	543	Air Canada
Air Canada	Montreal	Quebec	Canada	BR45665	Serge Paris	705	Air France
West Jet	Toronto	Ontario	Canada	BR45665	Serge Paris	705	Air France
Air France	Paris	Ile de France	France	BR45665	Serge Paris	705	Air France
American Airlines	Albany	New York	USA	BR45665	Serge Paris	705	Air France
Alitalia	Rome	Rome	Italie	BR45665	Serge Paris	705	Air France
Air Poly		Quebec	Canada	BR45665	Serge Paris	705	Air France
Air Canada	Montreal	Quebec	Canada	BR35332	Marco Rossi	801	Alitalia
West Jet	Toronto	Ontario	Canada	BR35332	Marco Rossi	801	Alitalia
Air France	Paris	Ile de France	France	BR35332	Marco Rossi	801	Alitalia
American Airlines	Albany	New York	USA	BR35332	Marco Rossi	801	Alitalia
Alitalia	Rome	Rome	Italie	BR35332	Marco Rossi	801	Alitalia
Air Poly		Quebec	Canada	BR35332	Marco Rossi	801	Alitalia
Air Canada	Montreal	Quebec	Canada	BR22445	John Steinpack	433	West Jet
West Jet	Toronto	Ontario	Canada	BR22445	John Steinpack	433	West Jet
Air France	Paris	Ile de France	France	BR22445	John Steinpack	433	West Jet
American Airlines	Albany	New York	USA	BR22445	John Steinpack	433	West Jet
Alitalia	Rome	Rome	Italie	BR22445	John Steinpack	433	West Jet
Air Poly		Quebec	Canada	BR22445	John Steinpack	433	West Jet
Air Canada	Montreal	Quebec	Canada	BR26587	George McDonald	559	West Jet
West Jet	Toronto	Ontario	Canada	BR26587	George McDonald	559	West Jet
Air France	Paris	Ile de France	France	BR26587	George McDonald	559	West Jet
American Airlines	Albany	New York	USA	BR26587	George McDonald	559	West Jet
Alitalia	Rome	Rome	Italie	BR26587	George McDonald	559	West Jet
Air Poly		Quebec	Canada	BR26587	George McDonald	559	West Jet
Air Canada	Montreal	Quebec	Canada	BR28811	Sam Withney	606	Air Canada
West Jet	Toronto	Ontario	Canada	BR28811	Sam Withney	606	Air Canada
Air France	Paris	Ile de France	France	BR28811	Sam Withney	606	Air Canada
American Airlines	Albany	New York	USA	BR28811	Sam Withney	606	Air Canada
Alitalia	Rome	Rome	Italie	BR28811	Sam Withney	606	Air Canada
Air Poly		Quebec	Canada	BR28811	Sam Withney	606	Air Canada
Air Canada	Montreal	Quebec	Canada	BR45321	Sami Zidane	542	Air France
West Jet	Toronto	Ontario	Canada	BR45321	Sami Zidane	542	Air France
Air France	Paris	Ile de France	France	BR45321	Sami Zidane	542	Air France
American Airlines	Albany	New York	USA	BR45321	Sami Zidane	542	Air France
Alitalia	Rome	Rome	Italie	BR45321	Sami Zidane	542	Air France
Air Poly		Quebec	Canada	BR45321	Sami Zidane	542	Air France
Air Canada	Montreal	Quebec	Canada	BR99999	Jacques Martin	666	Air Canada
West Jet	Toronto	Ontario	Canada	BR99999	Jacques Martin	666	Air Canada
Air France	Paris	Ile de France	France	BR99999	Jacques Martin	666	Air Canada
American Airlines	Albany	New York	USA	BR99999	Jacques Martin	666	Air Canada
Alitalia	Rome	Rome	Italie	BR99999	Jacques Martin	666	Air Canada
Air Poly		Quebec	Canada	BR99999	Jacques Martin	666	Air Canada
Air Canada	Montreal	Quebec	Canada	BM21675	Jean Lemieux	965	Air Poly
West Jet	Toronto	Ontario	Canada	BM21675	Jean Lemieux	965	Air Poly
Air France	Paris	Ile de France	France	BM21675	Jean Lemieux	965	Air Poly
American Airlines	Albany	New York	USA	BM21675	Jean Lemieux	965	Air Poly
Alitalia	Rome	Rome	Italie	BM21675	Jean Lemieux	965	Air Poly
Air Poly		Quebec	Canada	BM21675	Jean Lemieux	965	Air Poly

Cette requête renvoie le produit cartésien des deux tableaux compagnie et pilote, sans faire de tri. Chaque rangée du tableau compagnie est affichée à côté de chaque rangée du tableau pilote.

Requête 3 :

```
TP2=# SELECT *
TP2=# FROM COMPAGNIE C, PILOTE P
TP2=# WHERE C.nom=P.compa
TP2=# AND C.nom='Air Canada';
```

nom	ville	province	pays	brevet	nom	nbrhvol	compa
Air Canada	Montreal	Quebec	Canada	BR21675	Jean Lemieux	965	Air Canada
Air Canada	Montreal	Quebec	Canada	BR21431	Pierre Lheureux	543	Air Canada
Air Canada	Montreal	Quebec	Canada	BR28811	Sam Withney	606	Air Canada
Air Canada	Montreal	Quebec	Canada	BR99999	Jacques Martin	666	Air Canada

(4 rows)

Cette requête fait le produit cartésien des tableaux compagnie et pilote, puis sélectionne les rangées où le nom de la compagnie dans le tableau compagnie et le nom de la compagnie dans le tableau pilote sont identiques ET où le nom de la compagnie dans le tableau compagnie est 'Air Canada', puis affiche la concaténation de toutes les rangées ainsi obtenues. En pratique, on obtient à chaque rangée de ce tableau joint d'abord toute l'information sur Air Canada, puis l'information sur un des pilotes de Air Canada, avec une rangée par pilote de Air Canada.

13.

Requête 4 : SELECT * FROM pilote ORDER BY nom DESC;

```
TP2=# SELECT * FROM pilote ORDER BY nom DESC;
```

brevet	nom	nbrhvol	compa
BR45665	Serge Paris	705	Air France
BR28811	Sam Withney	606	Air Canada
BR45321	Sami Zidane	542	Air France
BR21431	Pierre Lheureux	543	Air Canada
BR35332	Marco Rossi	801	Alitalia
BR22445	John Steinpack	433	West Jet
BR21675	Jean Lemieux	965	Air Canada
BM21675	Jean Lemieux	965	Air Poly
BR99999	Jacques Martin	666	Air Canada
BR26587	George McDonald	559	West Jet

(10 rows)

Requête 5: SELECT * FROM pilote WHERE nbrHVol> 600 AND compa = 'Air Canada';

```
TP2=# SELECT * FROM pilote WHERE nbrHVol > 600 AND compa = 'Air Canada';
```

brevet	nom	nbrhvol	compa
BR21675	Jean Lemieux	965	Air Canada
BR28811	Sam Withney	606	Air Canada
BR99999	Jacques Martin	666	Air Canada

(3 rows)

Requête 6: SELECT A.matricule, F.compaAff, F.prix FROM AFFRETEMENT F, AVION A WHERE F.compaAff = A.proprio AND A.proprio = 'West Jet';

```
TP2=# SELECT A.matricule, F.compaAff, F.prix FROM AFFRETEMENT F, AVION A WHERE F.compaAff = A.proprio AND A.proprio = 'West Jet';
```

matricule	compaaff	prix
A7602S	West Jet	10500.00

(1 row)

Modification structurelle d'une table et évolution d'un schéma

14. Ajout de la colonne avec la contrainte :

```
TP2=# ALTER TABLE compagnie ADD COLUMN taille INTEGER CHECK (taille > 100);
ALTER TABLE
TP2=# SELECT * FROM compagnie;
```

nom	ville	province	pays	taille
Air Canada	Montreal	Quebec	Canada	
West Jet	Toronto	Ontario	Canada	
Air France	Paris	Ile de France	France	
American Airlines	Albany	New York	USA	
Alitalia	Rome	Rome	Italie	
Air Poly		Quebec	Canada	

(6 rows)

Tests de la contrainte :

```
TP2=# INSERT INTO compagnie VALUES ('Air Jong-Un', 'Pyongyang', 'Pyongyang Province', 'North Korea', 100);
ERROR:  new row for relation "compagnie" violates check constraint "compagnie_taille_check"
DETAIL:  Failing row contains (Air Jong-Un, Pyongyang, Pyongyang Province, North Korea, 100).
TP2=#
```

Comme la valeur de taille n'est pas supérieure à 100, l'insertion est refusée. Avec taille = 101, l'insertion est réussie.

```
TP2=# INSERT INTO compagnie VALUES ('Air Jong-Un', 'Pyongyang', 'Pyongyang Province', 'North Korea', 101);
INSERT 0 1
TP2=# SELECT * FROM compagnie;
```

nom	ville	province	pays	taille
Air Canada	Montreal	Quebec	Canada	
West Jet	Toronto	Ontario	Canada	
Air France	Paris	Ile de France	France	
American Airlines	Albany	New York	USA	
Alitalia	Rome	Rome	Italie	
Air Poly		Quebec	Canada	
Air Jong-Un	Pyongyang	Pyongyang Province	North Korea	101

(7 rows)

La commande utilisée pour la modification de la table est :

```
ALTER TABLE compagnie ADD COLUMN taille INTEGER CHECK (taille > 100);
```

15.

```
TP2=# ALTER TABLE compagnie ADD COLUMN nEmploye INTEGER DEFAULT 0;
ALTER TABLE
TP2=# INSERT INTO compagnie VALUES ('Air Jong-Il', 'Yangpyong', 'Pyongyang Province', 'North Korea', 102);
INSERT 0 1
TP2=# SELECT * FROM compagnie
TP2=# ;
```

nom	ville	province	pays	taille	nemploye
Air Canada	Montreal	Quebec	Canada		0
West Jet	Toronto	Ontario	Canada		0
Air France	Paris	Ile de France	France		0
American Airlines	Albany	New York	USA		0
Alitalia	Rome	Rome	Italie		0
Air Poly		Quebec	Canada		0
Air Jong-Un	Pyongyang	Pyongyang Province	North Korea	101	0
Air Jong-Il	Yangpyong	Pyongyang Province	North Korea	102	0

(8 rows)

16.

```
TP2=# ALTER TABLE compagnie ALTER COLUMN province TYPE varchar(20);
ALTER TABLE
TP2=# INSERT INTO compagnie VALUES ('Air Il-Sung', 'Yangpyongyang', 'Pyongyang Province Longue', 'North Korea', 102);
ERROR:  value too long for type character varying(20)
TP2=#
```

L'insertion d'une rangée avec valeur de province d'une longueur de 25 caractères est maintenant refusée.