

**SUJET**  
**De l'Enseignement Pratique**  
**« Introduction aux bases de données relationnelles »**

Bernard ESPINASSE

2019-2020

**1 Objectif de l'enseignement pratique (EP):**

---

L'objectif de cet E.P., associé au cours « Introduction aux Bases de données » de 3<sup>ème</sup> année, est d'appréhender tout d'abord la conception d'une base de données relationnelle, ensuite son implémentation avec un système de gestion de base de données relationnel (SGBDR), et enfin sa manipulation au travers du langage SQL (Structured Query Language), ceci au travers d'une étude de cas proposée.

Pour la phase de conception, on privilégiera la modélisation Entité-Relation et le SGBDR retenu pour l'implémentation est PostgreSQL, un système libre disponible sur plusieurs plateformes.

- Lien de la communauté francophone de PostgreSQL : <http://www.postgresqlfr.org/>
- Documentation en français de PostgreSQL : <http://docs.postgresql.fr/>

**Matériel et logiciels nécessaires:** PC sous Windows ou Linux, Connexion internet, Serveur PostgreSQL sous linux, et un Client PostgreSQL sous Windows ou Linux.

**Rendu :** le rendu de l'EP suivra les recommandations fournies sur le site du cours, et se fera au plus tard le samedi midi de la semaine de l'EP, par envoi d'un fichier en format « .pdf » à l'adresse : [bernard.espinasse@lisis.org](mailto:bernard.espinasse@lisis.org).

**2 Etude de cas : CYCLOPROVENCE**

---

**1. Présentation de CYCLOPROVENCE**

Il y a maintenant plus de trois ans que vous avez, avec une amie associée, fondé la compagnie CYCLOPROVENCE. Chaque été CYCLOPROVENCE permet à quelques centaines de touristes venant de tous les coins du monde de visiter à bicyclette, cette belle région qu'est la Provence.

La gestion de la compagnie CYCLOPROVENCE est assurée par les deux associés, Carine LAFLEUR et vous même. Deux autres personnes vous aident pendant la forte saison.

Pendant la période estivale, la compagnie CYCLOPROVENCE fait appel à une vingtaine de personnes, principalement des étudiants, pour assurer l'accompagnement des groupes de touristes.

## **Les circuits**

La saison des promenades en bicyclette commence au mois de juin pour finir fin septembre. Pour préparer cette saison, CYCLOPROVENCE réalise dès le mois de mars, un catalogue de circuits. Pour cela, elle établit des circuits de durées différentes et contacte des hôtels ou des particuliers (hébergement à la ferme), se trouvant sur ces itinéraires et négocie avec ces derniers les prix des pensions par personne (repas du soir+nuite+petit déjeuner). Notons qu'à chaque jour du circuit est associé un hébergement différent et bien sûr qu'un hébergement peut se retrouver dans plusieurs circuits.

## **Les voyages**

Pour chaque circuit, CYCLOPROVENCE propose un calendrier de dates. Pour un circuit donné, elle propose en moyenne 10 dates dans la saison, ce qui correspond à 10 voyages (un voyage correspondant à un circuit fixé dans le temps).

C'est au début avril que la compagnie CYCLOPROVENCE envoie un catalogue de ces voyages à ses principaux clients, qui sont principalement des agences de voyages d'Europe, des Etats-Unis et du Canada.

Ce catalogue contient pour chaque voyage les informations suivantes:

- La date de début et la durée du voyage
- La distance totale parcourue,
- Le coût par personne,

Pour chaque jour du voyage:

- Le nom de l'hébergement.

*Par exemple:*

*Voyage numéro 2:*

*Du 8 au 10 juin 2003*

*"Tour en 3 jours des baux de Provence", seront visités les sites suivants: .....*

*- Distance totale parcourue: 150 Kms,*

*- Coût 1000 FF par personne,*

*- Le nom des hébergements,*

*1<sup>o</sup>jour: LE CHEVAL BLANC*

*2<sup>o</sup>jour: LE MOULIN DE DAUDET*

*3<sup>o</sup>jour: MAS DU MARAIS*

Ces dates pourront, dans une certaine mesure, être discutées avec les agences intéressées.

## **Réservation par les clients**

A partir de ces propositions, les clients (agences), font leurs réservations concernant un certain nombre de places pour un voyage donné, ceci en général deux mois à l'avance.

*Par exemple:*

*L'agence ALPHA-Voyage de Montréal (Canada) à réservé le 10 mai (réservation RC03):*

*7 personnes pour le voyage # 20 (qui correspond au circuit #38, effectué entre le 9 et le 13 juillet),  
et*

*5 personnes pour le voyage #17 (qui correspond au circuit #20, effectué entre le 1 et le 6 août).*

Ainsi une réservation peut porter sur plusieurs voyages. A la réception d'une telle réservation, après avoir vérifié qu'il restait de la place dans ces voyages, CYCLOPROVENCE envoie à l'agence une première facture de 30 % du montant des billets.

### **Confirmation des réservations**

Les agences ont la possibilité de modifier ou d'annuler leurs réservations jusqu'à un mois avant la date de début du ou des voyages réservés. A cette date (1 mois avant le voyage), les agences doivent avoir confirmé leurs réservations par le règlement de la première facture des 30 %. Confirmées, les réservations clients passent du statut N au statut C. Si à cette date, cette confirmation n'est pas faite, la réservation est annulée. La confirmation étant faite, aucun remboursement pour cause d'annulation ne pourra être fait. A chaque voyage est affecté un accompagnateur.

### **Réservations des hébergements**

Chaque fin de semaine CYCLOPROVENCE traite les réservations clients qui ont été confirmées durant la semaine (statut C), en générant les réservations d'hébergement correspondantes qui seront envoyées aux hébergements concernés (ces réservations clients passent alors en statut T).

*Par exemple:*

*réservation RH03.*

*- faite le 20 juin 2003*

*- à hôtel "Le moulin de Daudet"*

*comprend*

*- pour le 10 juillet 2003*

*- 8 pensions pour le voyage #22*

*et*

*- pour le 11 juillet 2003*

*- 5 pensions pour le voyage #23*

*réservation RH05*

*-faite le 22 juin*

*- au Mas de Mme Honorine Olivier*

*comprend*

*- pour le 11 juillet 2003*

*- 8 pensions pour le voyage #22*

On notera que:

- Une réservation concerne qu'un lieu d'hébergement et peut être liée à plusieurs voyages différents pour des dates de pension différentes,
- On supposera que les hôtels et les fermes ont toujours des places disponibles,
- Le solde du montant des billets devra être réglé par l'agence un (1) mois après la réception de la deuxième facture qu'enverra CYCLOPROVENCE, une fois le voyage effectué, les réservations clients passent alors en statut R,

- Le règlement des hôtels et particuliers se fait une semaine après réception de leurs factures,
- Les accompagnateurs sont payés après chaque voyage à un tarif fixe par jour d'accompagnement.

## Les groupes

Dans la mesure des places disponibles, des touristes particuliers peuvent réserver une ou plusieurs places pour un voyage et de cette façon compléter une équipe déjà constituée. Ils sont alors considérés comme client à part entière. Un règlement de 30 % du montant des billets leur est demandé lors de la réservation et le solde est récolté le jour du départ du voyage. Les équipes ne peuvent pas dépasser plus de dix personnes, ceci pour des raisons évidentes de sécurité.

### 2. Listes des informations à prendre en compte

- prix d'une pension pour un hébergement donné	PrixHeb
- solde du client	SoldCli
- adresse accompagnateur	AdrAcc
- nom circuit	NonCir
- nom hébergement	NonHeb
- adresse client	AdrCli
- adresse hébergement	AdrHeb
- date à laquelle est faite la réservation d'hébergement	DatResHeb
- date de début du voyage	DatDebVoy
- date réservation de l'hébergement	DatHeb
- date réservation du client	DatResCli
- durée du circuit (jours)	DureeCir
- ident. accompagnateur	NoAcc
- ident. hébergement (hôtel, restaurant, particulier)	NoHeb
- ident. réservation du client	NoResCli
- ident. réservation hébergement (pensions)	NoResHeb
- ident. voyage	NoVoy
- ident. circuit	NoCir
- ident. client	NoCli
- nom accompagnateur	NomAcc
- nom du client (agence ou particulier)	NomCli
- nombre de km du circuit	KmCir
- nombre de lits réservés (un voyage et un hébergement donné)	NbPension
- nombre places réservées par l'agence pour un voyage donn	NbPlaceCli
- numéro du jour d'un hébergement dans un circuit	NoJourCir
- prix du circuit pour une personne	PrixCir
- solde hébergement (hôtel et ferme)	SoldHeb
- statut de la réservation du client: (N: non confirmée; C: confirmée; T:traitée; R:régulée)	StResCli

Cette liste n'est pas limitative. *Toute nouvelle information prise en compte devra avoir une description et une abréviation.*

### 3. Travail demandé

#### 1 Conception de la base de données

---

Comme déjà évoqué, le but de cet EP est d'appréhender la conception d'une base de données, son implémentation dans un système de gestion de base de données relationnel (SGBDR) et sa manipulation au travers du langage SQL (Structured Query Language).

##### 1. MCD

Dans un premier temps, pour la conception d'une base de données supportant la gestion de l'activité de CYCLOPROVENCE, il vous est demandé d'élaborer un modèle conceptuel de données (MCD) en formalisme Entité-Relation. Chaque propriété caractérisant une entité ou une relation du modèle sera définie et une abréviation précisée.

On utilisera impérativement dans les modèles graphiques les conventions graphiques Entité-Association vues en cours (conventions françaises ou Merise). Dans ces modèles, les propriétés associées aux entités et aux relations seront présentées selon leur abréviation et seront soulignées si elles sont identifiantes.

##### 2. MLD

1 - Elaborer ensuite par dérivation du MCD précédent, le Modèle Logique de Données Relationnel (MLDR) exprimé sous la forme graphique en respectant les conventions graphiques « Merise » vues en cours.

2 - Vous donnerez, pour chaque table de ce modèle, son schéma relationnel en mettant en évidence les attributs participant à la clé primaire de la table (soulignés) et les attributs clés étrangères (en italique).

##### 3. Script SQL de création de la base de donnée

En vue de l'implémentation de cette base de données dans le SGBD PostgreSQL, après avoir affecté à chacun des attributs du MLDR précédent un type de données disponible dans ce SGBD, vous procéderez à la création de la base de données par l'exécution de scripts de création en SQL2 de chacune des tables.

Ces scripts spécifieront les clés primaires, les clés étrangères et éventuellement les modalités de suppression (ON DELETE ...) et de modification (ON UPDATE...) jugées nécessaires en les argumentant.

Vous donnerez le script de création pour PostgreSQL des différentes tables de votre base de données (avec déclaration des primaires et étrangères) :

##### 4. Jeu d'essai

A partir des informations fournies en annexes, vous constituerez un jeu de données d'essai complet. Vous saisirez ces données dans chacune des tables de votre base de données précédemment créée (pas plus de 10 enregistrements par table).

Vous donnerez le contenu complet de ce jeu d'essai (commande SELECT \* FROM nom\_table).

## 2 Requêtes SQL

---

Enfin, on vous demande d'écrire en langage SQL un certain nombre de requêtes que vous exécuterez sur vos jeux de données précédemment saisies. Les réponses aux requêtes sont à écrire dans l'espace laissé après l'expression de la requête. Pour chacune des requêtes vous donnerez les résultats obtenus sur la base de votre jeu d'essai précédemment défini.

### 1. Sélection simple

1. Afficher le contenu de la table circuit.
2. Afficher le nom, la durée et le prix des circuits existants.
3. Afficher le nom et le prix des hébergements.
4. On veut afficher les différentes durées de circuit classées par ordre croissant.

### 2. Sélection avec traitement et/ou restrictions

5. Afficher les noms des clients par ordre alphabétique.
6. Afficher les réservations d'un client 'X'.
7. Afficher le nombre de km moyen par jour pour chaque circuit.
8. Afficher le numéro et le nom des hébergements pour lesquels le prix est supérieur à celui de l'hébergement ayant pour nom " Brin de lavande ".
9. Afficher les hébergements dont le prix est non compris entre 25 et 30 € (usage du BETWEEN)
10. Afficher les circuits dont le nom commence par 'TOUR'.
11. Afficher le numéro et la date de début des voyages associés au circuit numéro 001 et à l'accompagnateur numéro 001.
12. Afficher le numéro et le nom des circuits dont la durée est de 3 jours et dont la distance (km) est soit de 170 soit de 150 km.

### 3. Sélection d'informations en provenance de plusieurs tables : Jointures

13. Afficher les voyages et leur accompagnateur.
14. Afficher le nom et l'adresse des clients ayant passé une réservation entre le 1<sup>er</sup> avril 2017 et le 1<sup>er</sup> juillet 2017.
15. Afficher le numéro et la date de début des voyages dont l'accompagnateur a pour nom 'Durand'.
16. Afficher pour le client de nom 'VOYAGE ALPHA' tous les voyages qu'il a réservé, en affichant la date de début du voyage, le nom du circuit et le nom de l'accompagnateur concernés par chacun de ces voyages.
17. Afficher les hébergements dont il existe une ou plusieurs réservations pour le 1<sup>er</sup> juillet 2017.
18. Quels sont les circuits dont la distance (km) est supérieure aux circuits 'TOUR DES BAUX' et 'LES GORGES DU VERDON' (usage de ALL).

### 4. Requêtes avec usage de fonctions de groupes

19. Calculer la moyenne des distances (km\_cir) de tous les circuits
20. Afficher le nombre de circuits proposés
21. Afficher par réservation le nombre de voyages concernés.
22. Afficher, par hébergement (nom hébergement), le nombre de circuits concernés par ordre croissant

### 3 Accès à la base de données en langage Python

---

Pour terminer cet EP, vous aller accéder à votre base de données au travers du langage Python. Pour cela vous pouvez utiliser l'environnement **Spyder** pour développer votre applicatif en Python (ou un autre éditeur), et le driver **PyGreSQL** qui va vous permettre dans un programme Python la connexion à votre base de données, et de lancer des requêtes.

Pour disposer du driver PyGreSQL, vous devez dans Spyder tout d'abord lancer la commande :

```
pip3 install PyGreSQL
```

Maintenant vous pouvez utiliser PYGreSQL dans votre programme Python. La première étape est de créer une connexion à votre base de données. Pour cela vous utiliserez les instructions :

```
from pg import DB
db = DB(dbname='DB_name', host='147.94.190.228', port=5432, user='user_name', passwd='pwd')
```

Dans lequel `DN_name` est le nom de votre base de données, `user_name` votre nom d'utilisateur, et `pwd` votre mot de passe.

Ensuite PyGreSQL vous permet d'effectuer sur votre base de données divers opérations SQL notamment de consultation et de mise à jour sur votre base de données, comme l'illustre le petit exemple suivant, que vous pouvez taper et exécuter sous Spyder :

```
# -*- coding: utf-8 -*-
"""
Éditeur de Spyder
Ceci est un script temporaire.
"""

#Créer une connexion
from pg import DB
db = DB(dbname='mabase', host='147.94.190.228', port=5432, user='xxx', passwd='xxxx')

#Créer des tables
db.query("""CREATE TABLE temps (
    ville      varchar(80),
    t_basse    int,      -- température basse
    t_haute    int,      -- température haute
    prcp       real,    -- précipitation
    date       date
)""")
db.query("""CREATE TABLE produit (
    noprod     int,      -- no produit
    nomprod    varchar(20) -- nom produit
)""")

#Liste des tables de la base de donnees
```

```

db.get_tables()
#Insérer des enregistrements dans des tables
db.query(""" INSERT INTO temps VALUES ('Marseille', 46, 50, 0.25, '10/03/2020')""")
db.query(""" INSERT INTO temps VALUES ('Paris', 20, 10, 0.15, '10/03/2020')""")
db.query(""" INSERT INTO produit VALUES (1, 'clou')""")
db.query(""" INSERT INTO produit VALUES (2, 'vis')""")
#Consulter des tables
q = db.query("SELECT * FROM produit")
print(q)
q = db.query("SELECT * FROM temps")
print(q)
#Selectionner des enregistrements dans des tables
q = db.query(" SELECT * FROM produit WHERE nomprod ='vis' ")
print(q)
q = db.query("SELECT * FROM temps WHERE ville ='Paris' ")
print(q)

```

Pour aller plus loin vous consulterez le document « Basic Examples » présentant toutes les possibilités de PyGreSQL, document accessible en pdf sur la page du cours et à l'adresse : <https://www.pygresql.org/contents/postgres/basic.html>

Une fois PyGreSQL pris en main, on vous demande de réaliser :

1. Un programme permettant de rentrer dans votre base de données un nouvel accompagnateur.
2. Un programme permettant d'afficher, pour un circuit donné, les différents hébergements associés à ce circuit, ceci dans leur ordre consécutif.

Vous fournirez les sources et sorties de vos programmes.



## Annexe 1: Éléments pour élaborer un jeu de données

### Informations relatives aux circuits :

No circuit	Nom du circuit	Durée (jour)	Distance (km)	Prix (€/pers.)
001	TOUR DES BAUX	3	170	350
002	TOUR DU LUBERON	3	90	400
003	TOUR DE CAMARGUE	3	80	500
004	LES PAYSAGES DE CEZANNE	2	90	350
005	LA SAINTE VICTOIRE	2	80	350

...

### Informations relatives aux clients :

Numéro	Nom du client	Adresse du client
C01	VOYAGE ALPHA	PARIS
C02	QUEBEC TOUR	MONTREAL
C03	BIG APPLE TRAVEL	NEWYORK
C04	SOLEIL TOUR	OSLO
C05	TOUR ITALIA	ROME

...

### Informations relatives aux hébergements :

Numéro	Nom hébergement	Prix pension	Adresse	SOLDE
H01	LE CHEVAL BLANC	15	LES BAUX	120
H02	LE MOULIN DE DAUDET	20	LES BAUX	80
H03	MAS DU MARAIS	26	LES BAUX	200
H04	L'AUBERGE DE CAMARGUE	28	CAMARGUE	500
H05	MAS CHEZ FONFON	25	CAMARGUE	300
H06	LE BRIN DE LAVANDE	40	CAMARGUE	250
H07	MAS FARIGOULO	30	LUBERON	100
H08	FERME DES TORROS	35	LUBERON	100

...

### Informations relatives aux accompagnateurs :

N°accompagnateur	Nom accompagnateur	Adresse
A01	DURAND	VELAUX
A02	OLIVE	MARSEILLE
A03	GROSJEAN	CASSIS
A04	RAYMOND	AUBAGNE
A05	SEGUIN	MARSEILLE

...

### Informations relatives aux itinéraires des circuits :

No circuit	No de l'hébergement	No de jour dans le circuit	
001	H01	1	
001	H02	2	
001	H03	3	
002	...	H04	1

...

*Autres tables ....*

## Annexe 2: Conseils pour la mise en œuvre de PostgreSQL

### 1. Connexion au serveur PostgreSQL

---

Le serveur PostgreSQL se trouve à l'adresse suivante : **147.94.190.228** (Port : 5432 ; Hôte : postgresql2016).

Vous disposez de deux programmes d'administration vous permettant d'accéder au serveur :

- Web : **phppgadmin**

Pour accéder au serveur Postgresql avec **phppgadmin**, utiliser un navigateur, par exemple Firefox, et taper dans la fenêtre de dialogue des URL : **147.94.190.228/phppgadmin**

- Local : programme **pgadmin3** ou **pgadmin4**

Ce programme doit avoir été installé sur le serveur PostgreSQL. Pour le lancer taper **pgadmin3** ou **pgadmin4**.

### 2. Comptes d'administrateur et d'utilisateur de la base de données

---

- Pour chaque équipe un membre de l'équipe doit être désigné en tant qu'**administrateur de la base de données**. Sel l'administrateur de la base de données pourra créer les tables de la base de données, les supprimer et les modifier dans leur structure.
- Pour l'équipe 01, cet administrateur utilisera le compte suivant : Utilisateur : **equipe01** et Mot de passe : **pwd**; **Conseil : modifier votre mot de passe !**
- Pour chaque équipe une base de données a déjà été créée. Pour l'équipe 01, cette base de données s'appelle : **bdequipe01**.
- Avant de créer les tables dans cette bases de données, l'administrateur d'équipe doit créer un « schéma » , **NON public**, mais nommé aussi **equipe01**.
- 3 autres comptes ont été créés pour les autres membres de l'équipe, afin qu'ils puissent insérer, modifier ou supprimer des enregistrements dans les tables de la base de données créée par leur administrateur. Pour l'équipe 01 on aura les comptes :
  - Utilisateur : **eq01\_u1** et Mot de passe : **pwd**; **Modifier votre mot de passe !**
  - Utilisateur : **eq01\_u2** et Mot de passe : **pwd**; **Modifier votre mot de passe !**
  - Utilisateur : **eq01\_u3** et Mot de passe : **pwd**; **Modifier votre mot de passe !**
- Pour chaque équipe l'administrateur pourra attribuer ou restreindre des droits à ces 3 utilisateurs pour chaque table de la base de données.

### 3. Création de tables sous PostgreSQL

---

L'administrateur de l'équipe a plusieurs façons de créer les tables de la base de données d'équipe, nous lui conseillons de les créer à partir d'un des deux programmes d'administration **phpPgadmin** et **pgadmin3** ou **4**. Il utilisera pour cela l'**onglet/fenêtre SQL**, qui lui permet **d'éditer** une commande SQL et de **l'exécuter** (symbole de l'éclair avec phpPgadmin). Vous pouvez aussi **charger un fichier de commande SQL** sur votre disque et ensuite l'exécuter. Attention, ce nom de ce fichier texte doit se terminer par « **.sql** ». Les autres utilisateurs peuvent aussi utiliser cette interface pour lancer toute requête SQL en consultation ou mise à jour.

Soit les tables suivantes :

```
CLIENT (NOCLI, NOMCLI, AGECLI, SOLDECLI)
COMMANDE (NOCDE, NOCLI, DATECDE)
```

Le fichier de création de tables CLIENT et COMMANDE pourrait être :

```
-- creation des tables CLIENT et COMMANDE
DROP TABLE IF EXISTS CLIENT CASCADE;
DROP TABLE IF EXISTS COMMANDE CASCADE;
CREATE TABLE CLIENT
(
  NOCLI VARCHAR(32) PRIMARY KEY,
  NOMCLI VARCHAR(32) NOT NULL,
  AGECLI INTEGER NULL,
  SOLDECLI MONEY NULL
);
CREATE TABLE COMMANDE
(
  NOCDE VARCHAR(32) PRIMARY KEY,
  NOCLI VARCHAR(32) NOT NULL REFERENCES CIRCUIT,
  DATCDE DATE NOT NULL
);
```

Vous pouvez aussi créer les clés étrangères dans un second temps avec une commande ALTER, cela évite les problèmes d'antériorité de création de tables.

Remarque : toute instruction commençant par « -- » est pris comme un commentaire.

### 4. Insertion d'enregistrement dans les tables de la base

---

Pour insérer les enregistrements dans les tables selon votre jeu de données nous vous conseillons la même procédure. Pour insérer des données dans les 2 tables précédentes, voici le fichier utilisé (EVITER DE METTRE DES ACCENTS DANS VOS DONNEES) :

```
-- Insertion dans CLIENT
INSERT INTO CLIENT VALUES ('C01', 'DURAND', 45, 120.50);
INSERT INTO CLIENT VALUES ('C02', 'DUVAL', 51, 200.50);
INSERT INTO CLIENT VALUES ('C03', 'DUPOND', 35, 80.50);
...
-- Insertion dans COMMANDE
INSERT INTO COMMANDE VALUES ('D01', 'C01', '2004-07-08');
INSERT INTO COMMANDE VALUES ('D02', 'C02', '2004-07-15');
INSERT INTO COMMANDE VALUES ('D03', 'C03', '2004-07-20');
...
```

## 5 . Types de données de PostgreSQL (en gras les plus utilisés)

Nom	Alias	Description
bigint	int8	Entier signé sur 8 octets
bigserial	serial8	Entier sur 8 octets à incrémentation automatique
bit [ (n) ]		Suite de bits de longueur fixe
bit varying [ (n) ]	varbit	Suite de bits de longueur variable
<b>boolean</b>	<b>bool</b>	<b>Booléen (Vrai/Faux)</b>
box		Boîte rectangulaire dans le plan
bytea		Donnée binaire (« tableau d'octets »)
<b>character varying [ (n) ]</b>	<b>varchar [ (n) ]</b>	<b>Suite de caractères de longueur variable</b>
<b>character [ (n) ]</b>	<b>char [ (n) ]</b>	<b>Suite de caractères de longueur fixe</b>
cidr		Adresse réseau IPv4 ou IPv6
circle		Cercle dans le plan
<b>date</b>		<b>Date du calendrier (année, mois, jour)</b>
double precision	float8	Nombre à virgule flottante de double précision
inet		Adresse d'ordinateur IPv4 ou IPv6
<b>integer</b>	<b>int, int4</b>	<b>Entier signé sur 4 octets</b>
interval [ (p) ]		Intervalle de temps
line		Ligne infinie dans le plan
lseg		Segment de droite dans le plan
macaddr		adresse MAC
<b>money</b>		<b>montant d'une devise</b>
numeric [ (p, s) ]	decimal [ (p, s) ]	Nombre exact de la précision indiquée
path		Chemin géométrique dans le plan
point		Point géométrique dans le plan
polygon		Chemin géométrique fermé dans le plan
<b>real</b>	<b>float4</b>	<b>Nombre à virgule flottante de simple précision</b>
<b>smallint</b>	<b>int2</b>	<b>Entier signé sur 2 octets</b>
serial	serial4	Entier sur 4 octets à incrémentation automatique
<b>text</b>		<b>Chaîne de caractères de longueur variable</b>
time [ (p) ] [ without time zone ]		Heure du jour
time [ (p) ] with time zone	timetz	Heure du jour, avec fuseau horaire
timestamp [ (p) ] [ without time zone ]		Date et heure
timestamp [ (p) ] with time zone	timestampz	Date et heure, avec fuseau horaire