

TP1

PostgreSQL est un Système de Gestion de Bases de Données Relationnel Objet, *open source*, successeur de Ingres, développé par l'Université de Californie de Berkeley. Pour plus d'informations sur PostgreSQL, vous pouvez regarder les sites suivants :

<http://www.postgresql.org> <http://www.grappa.univ-lille3.fr/polys/reseaux-2000/reseaux022.html>

FAQ en Français : http://www.postgresql.org/docs/faqs.FAQ_french.html

Documentation en français : <http://docs.postgresqlfr.org/>

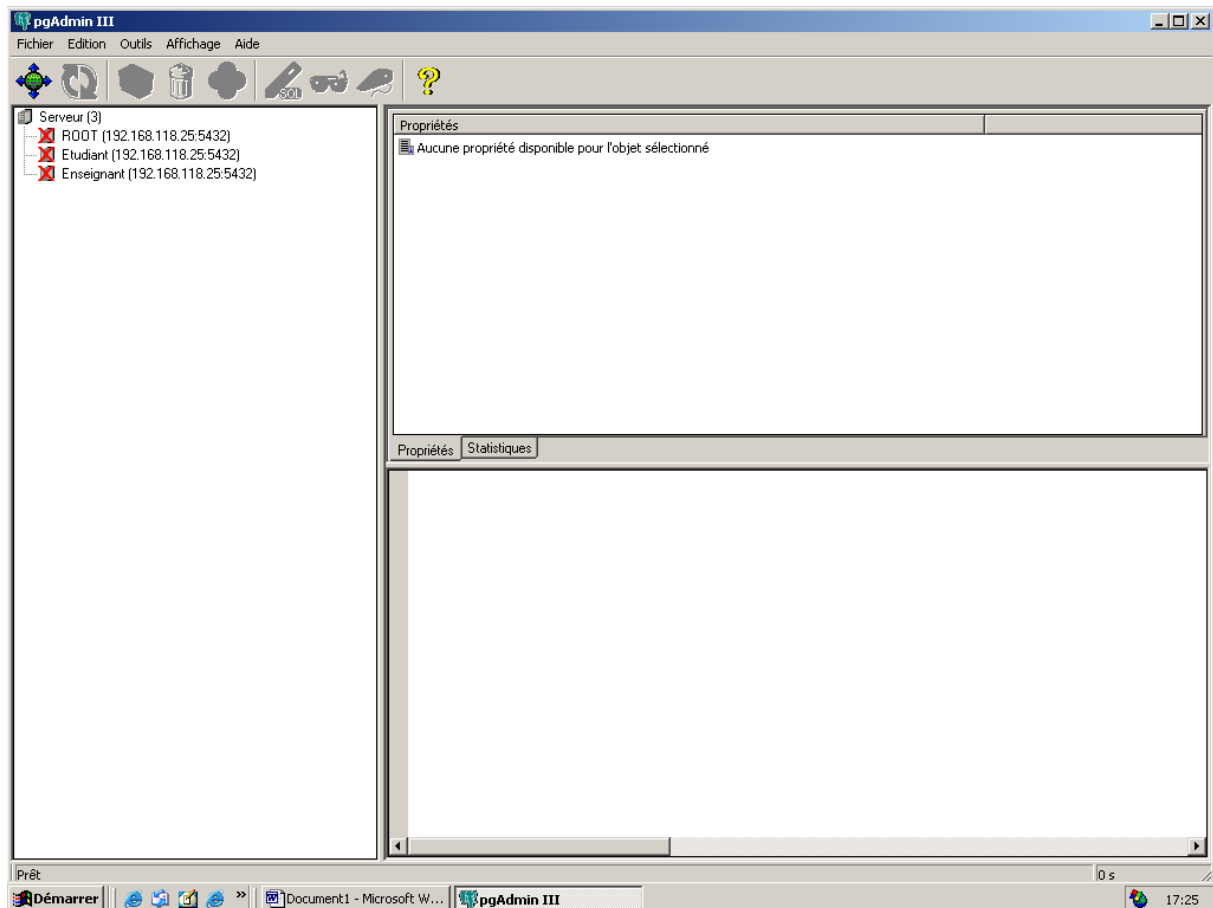
Site de la communauté française : <http://www.postgresql.fr/>

Ce document a pour objectif de vous aider à utiliser ce SGBD.

Partie 1 : Première utilisation de PostgreSQL et connexion à la base de données

1. Lancer Pgadmin III

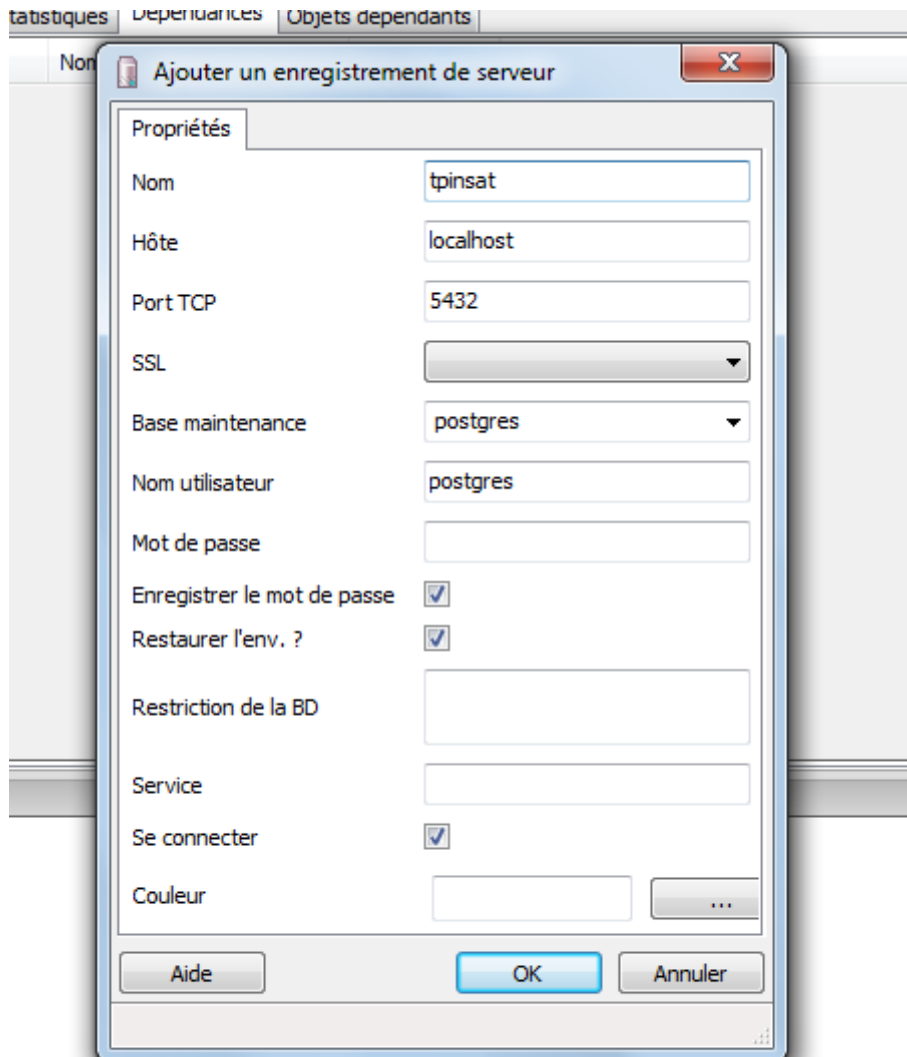
TP PostgreSQL



2. Pour se connecter il faut utiliser local host

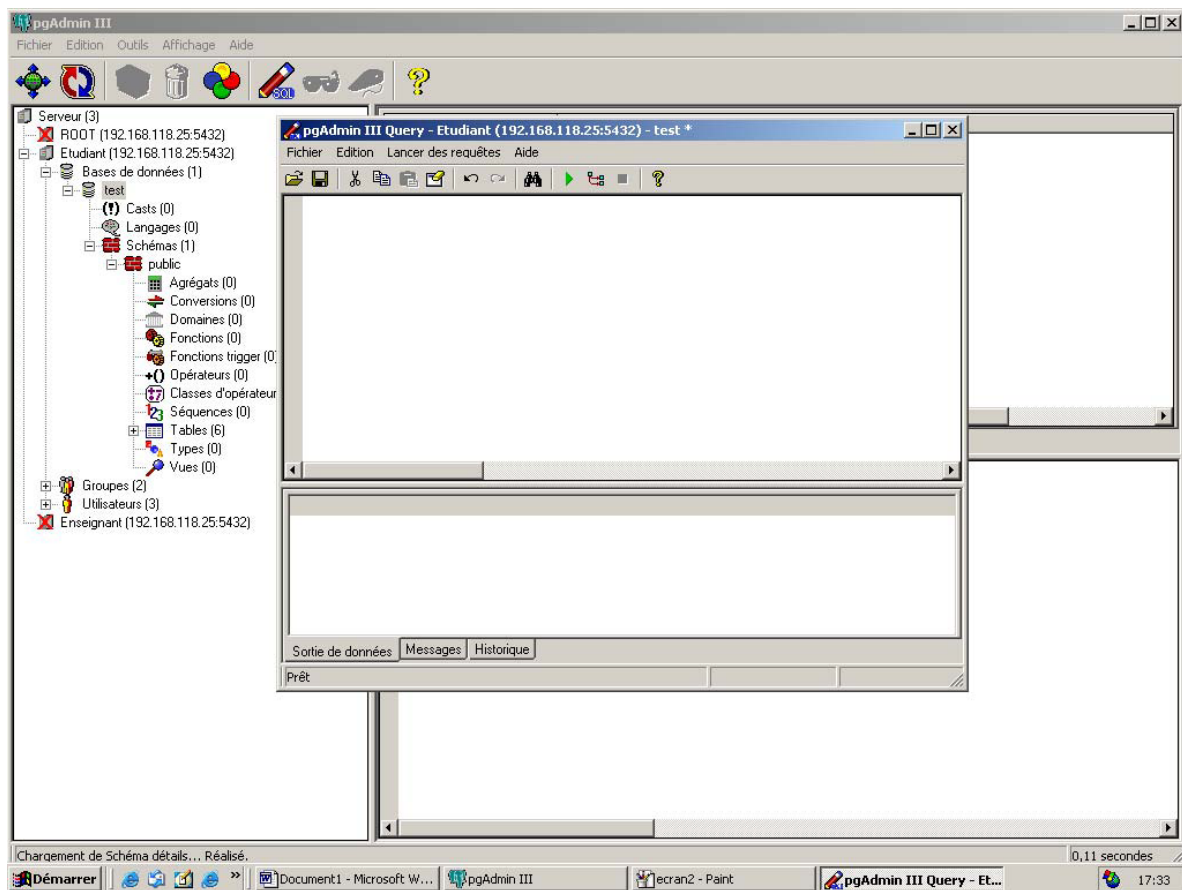


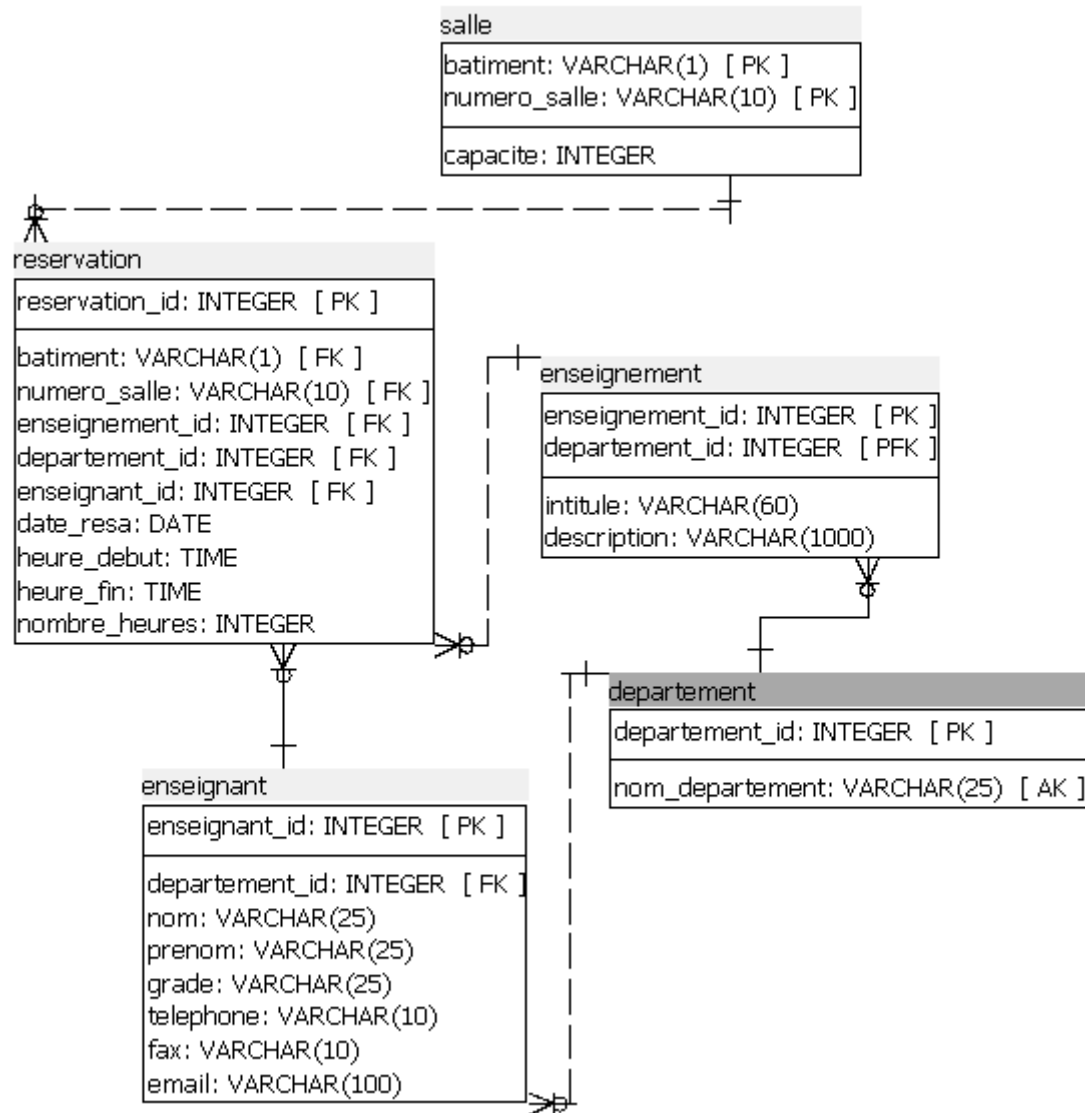
3. Pour créer une base



4. Pour exécuter des commandes SQL, cliquez sur le bouton **SQL** de la barre à outils. Une fenêtre apparaît

TP PostgreSQL



Partie2 : Enoncé

```

CREATE TABLE Etudiant
(
  Etudiant_ID      integer,
  Nom               varchar(25) NOT NULL,
  Prenom            Varchar(25) NOT NULL,
  Date_Naissance    date NOT NULL,
  Adresse           Varchar(50) DEFAULT NULL,
  Ville             Varchar(25) DEFAULT NULL,
  Code_Postal       Varchar(9) DEFAULT NULL,
  Telephone         Varchar(10) DEFAULT NULL,
  Fax               Varchar(10) DEFAULT NULL,
  Email             Varchar(100) DEFAULT NULL,
  CONSTRAINT PK_Etudiant PRIMARY KEY (Etudiant_ID));
  
```

TP PostgreSQL

```
CREATE OR REPLACE VIEW Email_Etudiant  
AS SELECT Nom, Prenom, Email FROM Etudiant;
```

Les contraintes étant :

Aucun département n'a un nom semblable à un autre
Un enseignement est donné par un seul enseignant tout au long d'une année et cela ne peut changer
Un enseignement est donné dans le cadre d'un département et cela ne peut changer
Un enseignant est rattaché uniquement à un département et cela ne peut changer
Un enseignant ne peut donner que des enseignements existant au niveau de son département
Un enseignement, une fois l'année universitaire entamée, ne peut faire l'objet d'arrêt
Un enseignement à toujours une durée supérieure à une heure
Un enseignant à toujours un grade qui est dans la liste suivante : 'Vacataire', 'Moniteur', 'ATER', 'MCF', 'PROF'
Une salle doit avoir une capacité supérieure à 10

A titre d'exemple indicative

Le SQL de la table réservation est :

```
CREATE TABLE Reservation  
(  
    Reservation_ID integer,  
    Batiment varchar(1) NOT NULL,  
    Numero_Salle varchar(10) NOT NULL,  
    Enseignement_ID integer NOT NULL,  
    Departement_ID integer NOT NULL,  
    Enseignant_ID integer NOT NULL,  
    Date_Resa date NOT NULL DEFAULT CURRENT_DATE,  
    Heure_Debut time NOT NULL DEFAULT CURRENT_TIME,  
    Heure_Fin time NOT NULL DEFAULT '23:00:00',  
    Nombre_Heures integer NOT NULL,  
    CONSTRAINT PK_Reservation PRIMARY KEY (Reservation_ID),  
    CONSTRAINT "FK_Reservation_Salle"  
    FOREIGN KEY (Batiment,Numero_Salle)  
    REFERENCES Salle (Batiment,Numero_Salle)  
    ON UPDATE RESTRICT ON DELETE RESTRICT,  
    CONSTRAINT "FK_Reservation_Enseignement"  
    FOREIGN KEY (Enseignement_ID,Departement_ID)  
    REFERENCES Enseignement (Enseignement_ID,Departement_ID)  
    ON UPDATE RESTRICT ON DELETE RESTRICT,  
    CONSTRAINT "FK_Reservation_Enseignant"  
    FOREIGN KEY (Enseignant_ID)  
    REFERENCES Enseignant (Enseignant_ID)  
    ON UPDATE RESTRICT ON DELETE RESTRICT,  
    CONSTRAINT CK_Reservation_Nombre_Heures CHECK (Nombre_Heures >=1),  
    CONSTRAINT CK_Reservation_HeureDebFin CHECK (Heure_Debut <  
    Heure_Fin)
```

TP PostgreSQL

) ;

Sachant que

ON UPDATE (ou *ON DELETE*) est suivi d'arguments entre accolades permettant de spécifier l'action à réaliser en cas de modification d'une ligne de la table faisant partie de la clé étrangère :

- *CASCADE* indique la modification en cascade des lignes de la table étrangère dont les clés primaires correspondent aux clés étrangères des lignes modifiées
- *RESTRICT* indique une erreur en cas de modification d'une valeur correspondant à la clé
- *SET NULL* place la valeur NULL dans la ligne de la table étrangère en cas de modification d'une valeur correspondant à la clé
- *SET DEFAULT* place la valeur par défaut (qui suit ce paramètre) dans la ligne de la table étrangère en cas de modification d'une valeur correspondant à la clé

Une contrainte d'intégrité est une clause permettant de contraindre la modification de tables, faite par l'intermédiaire de requêtes d'utilisateurs, afin que les données saisies dans la base soient conformes aux données attendues. Ces contraintes doivent être exprimées dès la création de la table grâce aux mots clés suivants :

- *CONSTRAINT*
- *DEFAULT*
- *NOT NULL*
- *UNIQUE*
- *CHECK*

Une valeur par défaut lorsqu'un champ de la base n'est pas renseigné grâce à la clause *DEFAULT*. Cela permet notamment de faciliter la création de tables, ainsi que de garantir qu'un champ ne sera pas vide.

La clause *DEFAULT* doit être suivie par la valeur à affecter. Cette valeur peut être un des types suivants :

- constante numérique
 - constante alphanumérique (chaîne de caractères)
 - le mot clé **USER** (nom de l'utilisateur)
 - le mot clé **NULL**
 - le mot clé **CURRENT_DATE** (date de saisie)
 - le mot clé **CURRENT_TIME** (heure de saisie)
 - le mot clé **CURRENT_TIMESTAMP** (date et heure de saisie)
-

- 1) donner la traduction en sql
- 2) implémenter le schéma de la base sur postgres

Partie 3 : Insertion des valeurs dans la base une 20 vingtaine par table

Exemple

```
INSERT INTO Departement VALUES ('1','IRT');
INSERT INTO Departement VALUES ('2','IGL');
INSERT INTO Departement VALUES ('3','ILA');

INSERT INTO Etudiant VALUES ('1','ben foulén', 'foulén','1979/02/18','50,
Rue des alouettes','TUNIS','75021','0143567890',NULL,'foulén@gmail.com');

INSERT INTO Etudiant VALUES ('2','tounsi', 'ahmed','1980/08/23','10, Avenue
des marguerites','bardo','40000','0678567801',NULL,'pat@yahoo.fr');

INSERT INTO Etudiant VALUES ('3','tounsi', 'Jamal','1978/05/12','25,
Boulevard des
fleurs','TUNIS','75022','0145678956','0145678956','odent@free.fr');

INSERT INTO Etudiant VALUES ('4','benmard', 'ahmed','1979/07/15','56,
Boulevard des
fleurs','TUNIS','75022','0678905645',NULL,'deby@hotmail.com');

INSERT INTO Etudiant VALUES ('5','foulana', 'tounsia','1979/08/15','45,
Avenue des abeilles','ariana','75022',NULL,NULL,NULL);

INSERT INTO Enseignant
VALUES('1','1','ousteith','ouahed','MCF','4185','4091','ousteith@gmail.com'
);
INSERT INTO Enseignant
VALUES('2','1','ousteitha','wahida','PROF',NULL,NULL,'wahida@gmail.com');

INSERT INTO Salle VALUES('B','020','15');
INSERT INTO Salle VALUES('B','022','15');
INSERT INTO Salle VALUES('A','301','45');
INSERT INTO Salle VALUES('C','Amphi 8','500');
INSERT INTO Salle VALUES('C','Amphi 4','200');

INSERT INTO Enseignement VALUES ('1','1','Bases de Données
Relationnelles','Niveau Licence (L3) : Modélisation E/A et UML, Modèle
relationnel, Algèbre Relationnelle, Calcul relationel, SQL, dépendances
fonctionnelles et formes normales');
INSERT INTO Enseignement VALUES ('2','1','Langage C++','Niveau Master 1');
INSERT INTO Enseignement VALUES ('3','1','Mise à Niveau Bases de
Données','Niveau Master 2 - Programme Licence et Master 1 en Bases de
Données');

INSERT INTO Reservation VALUES
('1','B','022','1','1','1','2008/10/15','08:30:00','11:45:00','3');
INSERT INTO Reservation VALUES
('2','B','022','1','1','2','2008/11/04','08:30:00','11:45:00','3');
INSERT INTO Reservation VALUES
('3','B','022','1','1','2','2008/11/07','08:30:00','11:45:00','3');
INSERT INTO Reservation VALUES
('4','B','020','1','1','1','2008/10/15','08:30:00','11:45:00','3');
```


TP PostgreSQL

1. Liste des noms et des prénoms des étudiants stockés dans la base.
2. Liste des noms et des prénoms des étudiants qui habitent une ville de tunis puis de sfax
3. Liste des noms et des prénoms des étudiants dont le nom commence par 't' puis par 'f'.
4. Liste des noms et des prénoms des enseignants dont l'avant dernière lettre du nom est 'a'.
5. Liste des noms et des prénoms des enseignants classés par nom de département, par nom et par prénom.
6. Combien y a-t-il d'enseignants dont le grade est 'Moniteur' ?
7. Quels sont les noms et les prénoms des étudiants n'ayant pas de Fax (valeur NULL)?
8. Quels sont les intitulés des enseignements dont la description contient le mot 'SQL' ou 'Licence' ?
9. Quelles sont la capacité moyenne et la capacité maximum des salles ?
10. Quelles sont les salles dont la capacité est inférieure à la capacité moyenne ?
11. Classer les étudiants par ville.

Parti 4

Modification du schéma de la base exemple

1. Ajouter, dans la base de données exemple, une relation permettant de gérer les inscriptions des étudiants aux différents enseignements disponibles dans la base (la table doit contenir un attribut date d'inscription).
2. Ajouter, dans la base de données exemple, une relation permettant de gérer les notes des étudiants dans les différents enseignements (un étudiant peut avoir plusieurs notes pour le même enseignement).
3. Implémenter la base au niveau du SGBD MYSQL

Parti 5

Déclencheur

1. Créer un déclencheur permettant de vérifier, lors de l'insertion d'une note pour un Étudiant, que ce dernier possède bien une inscription pour cet enseignement (sinon ajouter l'inscription de l'étudiant à l'enseignement)