



**Administration**

**Bases de données**

**Chérif Bachir DEME Enseignant chercheur en Cryptologie à l'UADB**

# Chapitre 1 : Administration

---

**Objectifs spécifiques** : A la suite de ce chapitre, l'étudiant doit être capable de:

1. Créer la structure d'une base de données
2. Modifier la structure d'une base de données
3. Supprimer la structure d'une base de données
4. Définir une vue
5. Créer une vue à partir d'une tables
6. Supprimer une vue
7. Définir un rôle
8. Créer un rôle et l'attribuer à un utilisateur
9. Gérer les transactions dans une base de données
10. Gérer les utilisateurs d'une base données
11. Allouer des droits aux utilisateurs
12. Définir un index
13. Créer les index sur les tables
14. Supprimer un index



## Qu'est-ce que administrer une base de données ?

- Administrer une base de données revient à :
  - Créer et implémenter les données;
  - Gérer les index des tables;
  - Gérer les vues;
  - Gérer les transactions;
  - Administrer les autorisations d'accès pour les utilisateurs;
  - Disposer d'un Système de Gestion de Bases de Données;
  - Utiliser des clients pour administrer à distance ou en local les bases de données;
  - Etc.



# Administration

---

## Gestion des Bases de données

- **Création**

CREATE DATABASE nom\_base;

**Exemple :** CREATE DATABASE GestionBoutique;

- **Modification**

ALTER DATABASE nom\_base;

**Exemple :** CREATE DATABASE GestionBoutique;

- **Suppression**

DROP DATABASE nom\_base;

**Exemple :** DROP DATABASE GestionBoutique;

# Administration

## Gestion des index

- Un index est une structure pour accélérer la recherche dans une colonne ou un groupe de colonnes.
- Il permet d'assurer l'intérêt de données de la base de données.

## Création et suppression (ou destruction) d'un index

- Création d'un index

CREATE [UNIQUE] INDEX nom\_index ON nom\_table (attribut [ASC| DESC], ...);

**Exemple :** CREATE UNIQUE INDEX index1 ON Client;

- Suppression d'un index

ALTER TABLE nom\_table DROP INDEX nom\_index ;

**Exemple :** ALTER TABLE Client DROP INDEX index1;

## Gestion des Vues

- Une vue est table virtuelle stockée créée à partir d'une table existante dans la base à l'aide d'une requête.
- Elles sont utilisées comme les tables.
- Elles permettent de :
  - prendre en compte (la sauvegarde) les requêtes complexes,
  - présenter les données sous différentes formes adaptées,
  - supporter l'indépendance et renforcer la sécurité.

### Remarques :

- La recherche d'une valeur dans une colonne indexée prend un temps quasi constant, indépendamment de la taille de la table
- L'index est maintenu à jour automatiquement

# Administration

## Gestion des vues

- **Création d'une vue**

CREATE VIEW nom\_vue AS requête;

**Exemple :** CREATE VIEW Liste\_des\_Clients AS SELLECT \* FROM Client;

- **Interrogation d'une vue**

SELECT liste\_attributs FROM nom\_vue;

**Exemple :** SELECT nom, prenom FROM Liste\_des\_Client;

- **Suppression d'une vue**

DROP VIEW nom\_vue ;

**Exemple :** DROP VIEW Liste\_des\_Client;

## Gestion des transactions

- Une transaction est un ensemble de requêtes exécuté en seul bloc. Cependant si une seule des requêtes échoue, l'administrateur peut décider d'annuler ou valider la transaction.
- Une transaction est une mise à jour des données qui débute avec la session de travail ou à la fin de la transaction précédente.
- La gestion des transactions repose essentiellement sur le langage de contrôle des transactions (LCT).



## Langage de Contrôle des Transactions (LCT)

- Le langage de contrôle de données permet de gérer les transactions de données.
- Il repose essentiellement sur deux mots clés :
  - COMMIT pour valider les mises à jour (ou changement) de requêtes et mettant fin à la transaction.
  - ROLLBACK pour annuler (mettre fin) à la transaction sans prises en compte des mises à jour.

# Administration

## Création d'une transaction

- Phase de validation d'une transaction

### Ouverture d'une transaction

START TRANSACTION;

### Validation d'une transaction

ROLLBACK;

**START TRANSACTION;**

--Mettre à jour le nom du client numéro 1

UPDATE Client SET nom='DIAW'

WHERE numClient=1;

**ROLLBACK;**

- Phase de validation d'une transaction

### Ouverture d'une transaction

START TRANSACTION;

### Validation d'une transaction

COMMIT;

**START TRANSACTION;**

-- Insertion de trois clients

INSERT INTO Client

VALUES(1, 'Diop', 'Ahmeth','Diourbel');

INSERT INTO Client

VALUES(3, 'Diop', 'Aly','Diourbel');

INSERT INTO Client

VALUES(4, 'DIA', 'Ahmeth','Fatick');

**COMMIT;**

## Langage de Contrôle des Transactions (LCT)

- Une transaction est composée d'une phase de :
  - validation avec un COMMIT prenant en compte les mises à jour et mettant fin à la transaction.
  - annulation avec ROLLBACK mettant fin à la transaction sans prises en compte des mises à jour.

## Gestion des utilisateurs

- La gestion des utilisateurs requiert les actions suivantes :
  - création, modification et suppression d'utilisateurs;
  - explication des privilèges et options des utilisateurs;
  - attribution et révocation de privilèges aux utilisateurs.
- Elle s'appuie sur le langage de contrôle de données pour attribuer et/ ou révoquer des privilèges des utilisateurs.

## Langage de Contrôle de Données (LCD)

- Le langage de contrôle de données permet de contrôler les accès aux données.
- Il va permettre de créer des utilisateurs, de leurs donner et retirer des privilèges.
- Il repose essentiellement sur deux mots clés :

➤ GRANT pour attribuer des privilèges (des droits d'accès) ;

➤ REVOKE pour révoquer les privilèges ( des droits d'accès).

# Administration

---

## Gestion des utilisateurs

- **Création d'utilisateur**

CREATE USER nom\_utilisateur IDENTIFIED BY mot\_de\_passe;

**Exemple :** CREATE USER 'faye' IDENTIFIED BY 'passer123';

- **Suppression d'utilisateur**

DROP USER nom\_utilisateur CASCADE;

**Exemple :** DROP USER faye CASCADE;

- **Modification d'utilisateur**

ALTER USER nom\_utilisateur IDENTIFIED BY nouveau\_mot\_de\_passe;

**Exemple :** ALTER USER faye IDENTIFIED BY 'Passer@123'

# Administration

## Gestion des utilisateurs

- Exemples de droits sur les objets de la base de données

Privilèges	Signification	Tables	Vues
CREATE	Création	X	X
ALTER	Destruction	X	
DELETE	Suppression	X	X
INDEX	Construction	X	
INSERT	Intersection	X	X
REFERENCES	Clé étrangère	X	
SELECT	Lecture	X	X
UPDATE	Mise à jour	X	X
ALLL	Tous	X	X

# Administration

## Gestion des utilisateurs

- **Attribution de privilèges**

GRANT nom\_privilège ON nom\_table | nom\_vue TO user | PUBLIC [WITH GRANT OPTION];

### Exemples :

Sur un objet: GRANT SELECT ON nom\_table TO nom\_user;

Sur globaux et rôles: GRANT CREATE ANY TBLE TO nom\_user,

- **Revocation de privilèges**

REVOKE nom\_privilège ON nom\_table | nom\_vue TO user | PUBLIC ;

### Exemples :

Sur un objet: REVOKE SELECT ON nom\_table FROM nom\_user;

Sur globaux et rôles: GRANT CREATE ANY TBLE FROM nom\_user,



# Administration

---

## Gestion des utilisateurs

- **Rôles**

On distingue deux types de rôles:

- **Les rôles prédéfinis**

CONNECT : droit de création de tables, vues, synonymes, etc.

RESOURCE: droit de création de procédure stockées, fonctions, déclencheurs, etc.

DBA : administrateur de la base de données.

- **Les rôles utilisateur**

Rôle créé par les utilisateurs à l'aide de la requête: `CREATE ROLE nom_role;`

# Administration

---

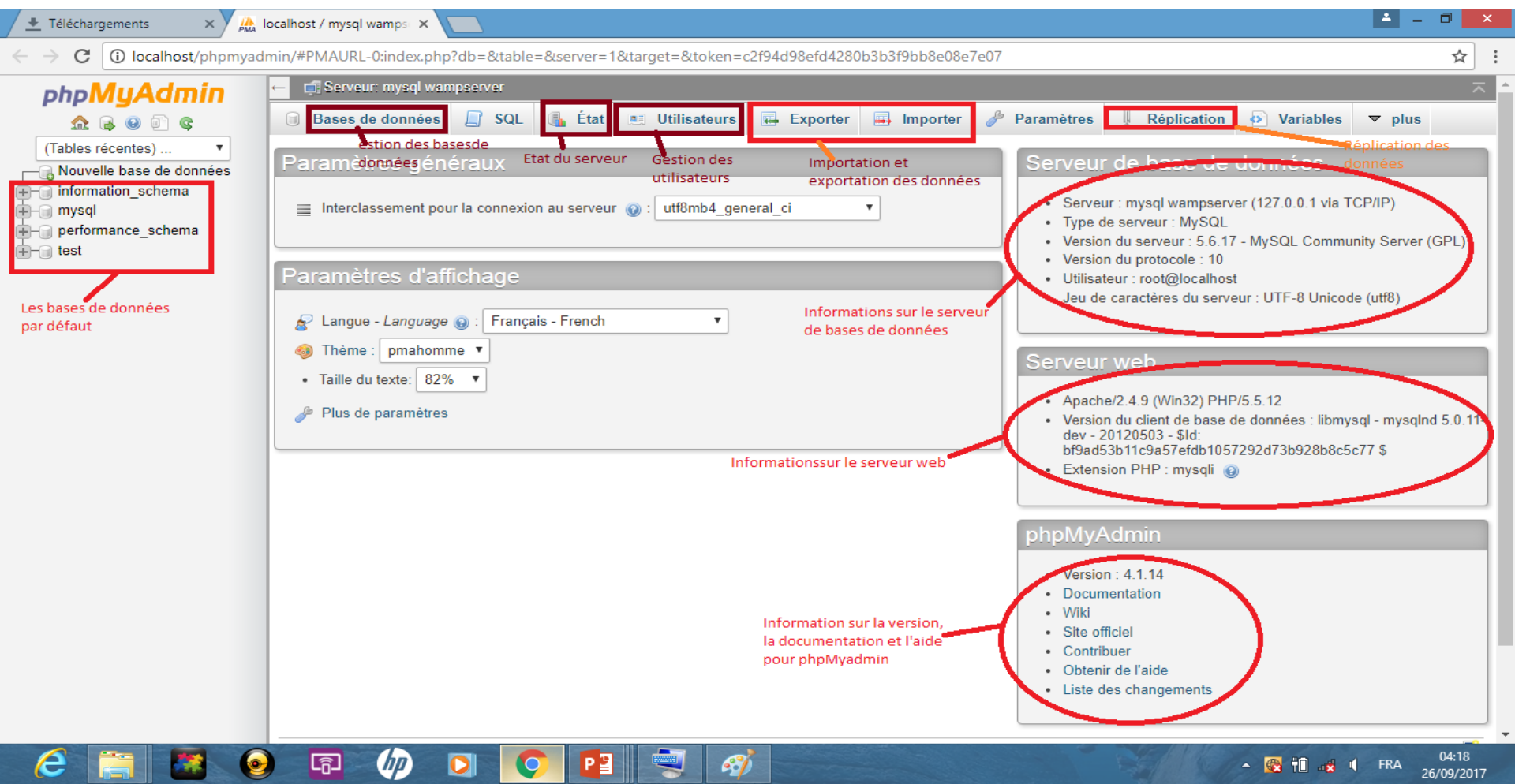
## Les Clients

- Un client est une application capable d'interroger le serveur et transmettre les réponses du serveur. Il existe plusieurs types de clients :
- Les clients web qui utilisent une interface pour communiquer avec le serveur de base de données via un navigateur (Mozilla Firefox, Chrome, Microsoft Internet Explorer, Troch, Etc.) et administrer les bases de données stockées dans ce dernier.

**Exemple** : PhpMyAdmin, une interface sur laquelle nous allons revenir plus amplement.

# Administration

## La page d'accueil de phpMyAdmin



# Administration

---

## Les Clients

- Un client est une application capable d'interroger le serveur et transmettre les réponses du serveur. Il existe plusieurs types de clients :
- Les clients logiciel, développés exprès pour l'administration des données d'une base données. Ils sont diverses, tantôt libres et tantôt propriétaires (pour dire commerciaux).

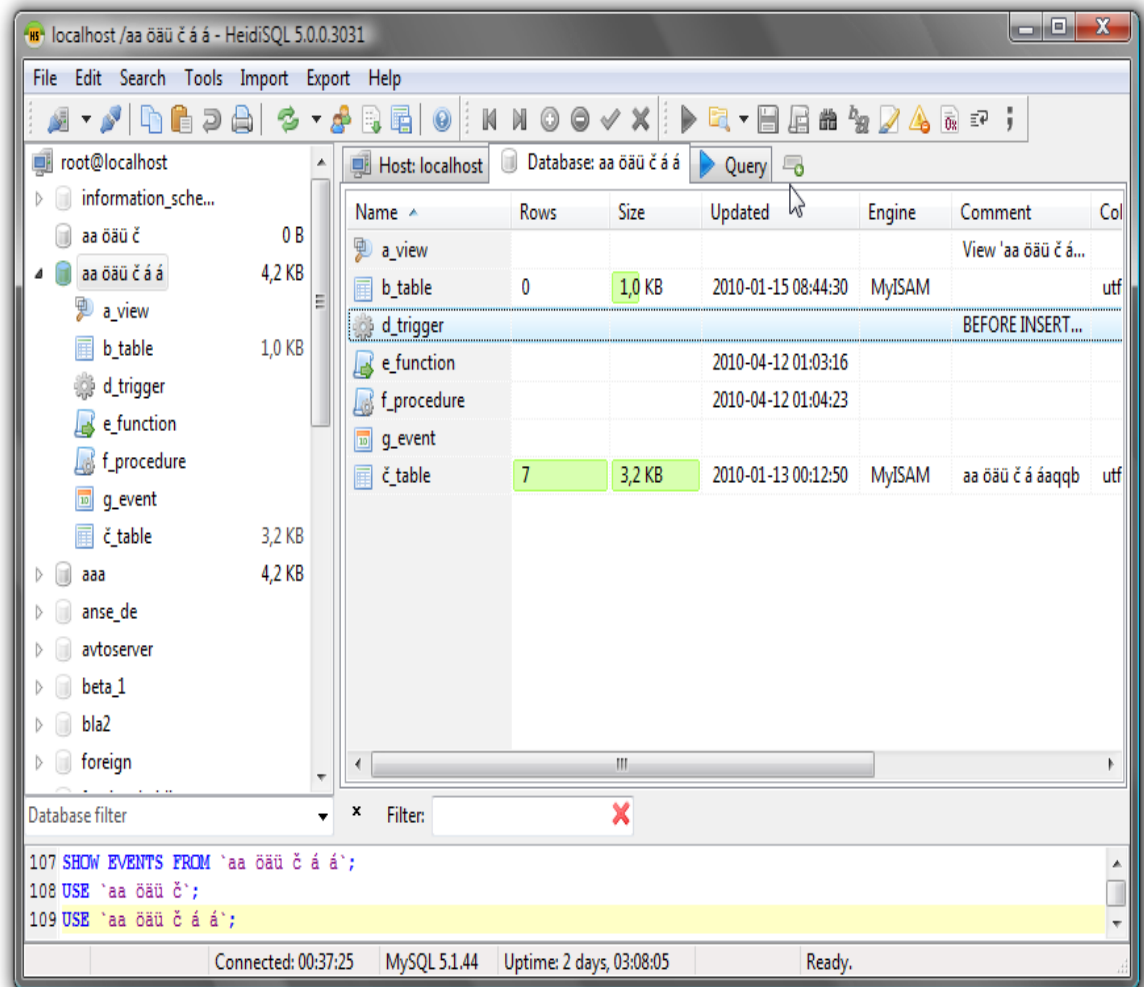
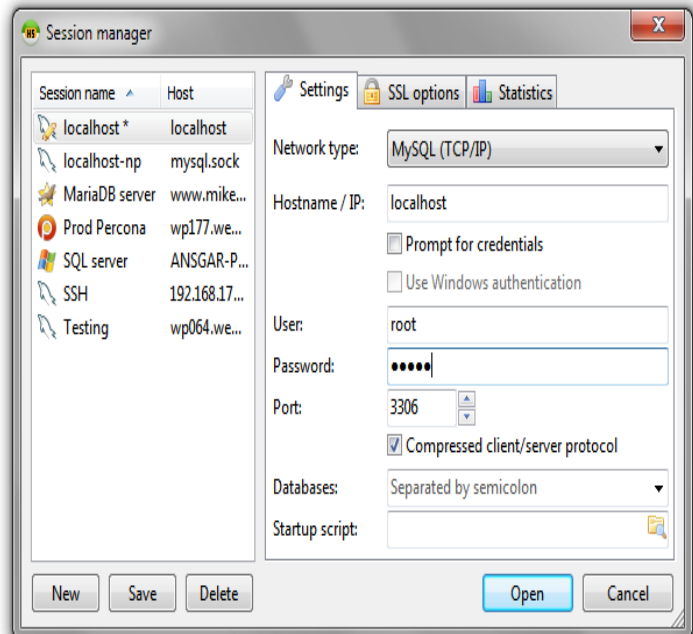
### Exemples :

- Libres:  **HeidiSQL** , téléchargeable sur son site.

- Commerciaux : Navicat

# Administration

## Les Fenêtres d'accueil de HeidiSQL



# Application

- Soit le schéma relationnel suivant :

**Acheteur**(idAcheteur, nom, prenom, tel, adresse email, #ninea)

**Vendeur**(numVendeur, nom, prenom, adresse, email, tel, #ninea)

**Articles**(numArt, designation, prix\_unitaire, quantitéStock, #ninea)

**Boutique**(ninea, adresse, chiffreAffaire, tel,)

**Travail à faire:** Donner les commandes SQL correspondant aux requêtes suivantes:

1. La liste des vendeurs.
2. La liste des potentiels acheteurs
3. La liste de tous articles d'une boutique
4. La liste de toutes les boutiques
5. Les articles qui ont été achetés
6. Les articles vendus
7. Les articles achetés et vendus par la même personne c'est-à-dire l'acheteur est le vendeur

# Application

8. Les vendeurs qui vendent les mêmes articles
9. Les acheteurs habitant Pikine.
10. Les vendeurs et acheteurs ayant la même adresse
11. Les articles dont le stock est inférieur à 10.
12. Les articles achetés au vendeur Abdou Faye.
13. Les articles achetés à la boutique 18 et dont le prix\_unitaire est égale à 15 000.
14. Le vendeur Doudou Gueye qui a vendu l'article puce orange 4g, à l'acheteur Alimatou BA
15. Les boutiques situées dans la même zone que ses acheteurs et vendeurs.
16. Les Vendeurs portant le Diop.
17. Les achats dont le montant est compris entre 150 000 et 250 000
18. Les articles dont le nom commence par C et O.
19. Les boutiques dont le chiffre d'affaire dépasse les 1 000 000
20. Les boutiques qui ont vendus beaucoup plus d'articles.