ISC22-1 --- Architecture orientée service
Bases de données avancées
Système de Gestion de Bases de Données

Jean-Marie Pécatte
jean-marie.pecatte@iut-tlse3.fr

Objectifs d'un SGBD

Architecture d'une BD

Une base de données est représentée selon 3 niveaux *(norme ANSI/SPARC) :*

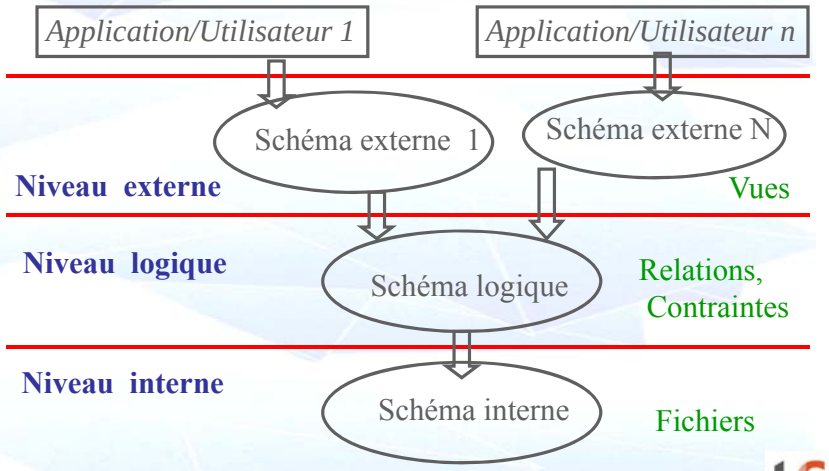
- le niveau logique (ou conceptuel)
- le niveau physique (ou interne)
- le niveau externe

Cette séparation permet :

- une indépendance entre la sémantique des données et leur implémentation physique
- une vision restreinte et adaptée à chaque type d'utilisateur

Architecture d'une BD

Schéma de représentation d'une BD



Architecture d'une BD

- **Schéma interne** : description des données en terme de représentation physique en machine → structures de mémorisation, méthodes de stockage et d'accès pour ranger et retrouver les données sur le disque
- **Schéma externe** : description d'une partie de la BD extraite ou calculée à partir de la base physique correspondant à la vision d'un programme ou d'un utilisateur

Objectifs d'un SGBD

- **Indépendance physique :** entre les niveaux logique et interne
- un des objectifs principaux des SGBD
- pouvoir modifier la représentation interne des données sans changer le schéma conceptuel → pérennité de l'existant
- Par exemple changer de :
  - Méthode d'accès, - Mode de représentation
  - Méthode de tri, - Codage de l'information, ...

  afin d'optimiser les performances du SGBD

  sans modifier le schéma relationnel créé
- Le SGBD pourra avoir des représentations internes différentes suivant le système d'exploitation qui le supporte

Objectifs d'un SGBD

- **Indépendance logique :** entre les niveaux logique et externe

Les avantages sont les suivants :

- permettre à chaque groupe de travail de voir les données comme il souhaite (sous-schéma) permettre l'évolution d'un sous-schéma sans remettre en cause le schéma logique
- permettre l'évolution d'un schéma sans remettre en cause les autres schémas
- permettre l'évolution du schéma logique sans remettre en cause les schémas externes

→ Permettre des évolutions sans remettre en cause l'existant

Objectifs d'un SGBD

- **Langages non procéduraux :**

Suite à l'indépendance physique, il est souhaitable que la manipulation de données se fasse indépendamment du niveau physique et de la structure interne de la BD

Simplicité d'utilisation d'un langage non procédural avec lequel on décrit ce que l'on souhaite sans avoir besoin d'écrire l'algorithme correspondant.

Objectifs d'un SGBD

Efficacité de l'accès aux données :

- Performances en terme de débit : nb de transactions types exécutées par seconde
- Performance en terme de temps de réponse : temps d'attente moyen pour une requête type
- Goulot d'étranglement : E/S disque

→ Limiter les accès disque : gestion d'un cache en mémoire

Objectifs d'un SGBD

Partage des données :

Permettre aux applications de partager les données de la base simultanément

Une application doit pouvoir accéder aux données comme si elle était la seule à les utiliser, sans attendre mais aussi sans se soucier de savoir si une autre application peut les modifier concurremment.

Objectifs d'un SGBD

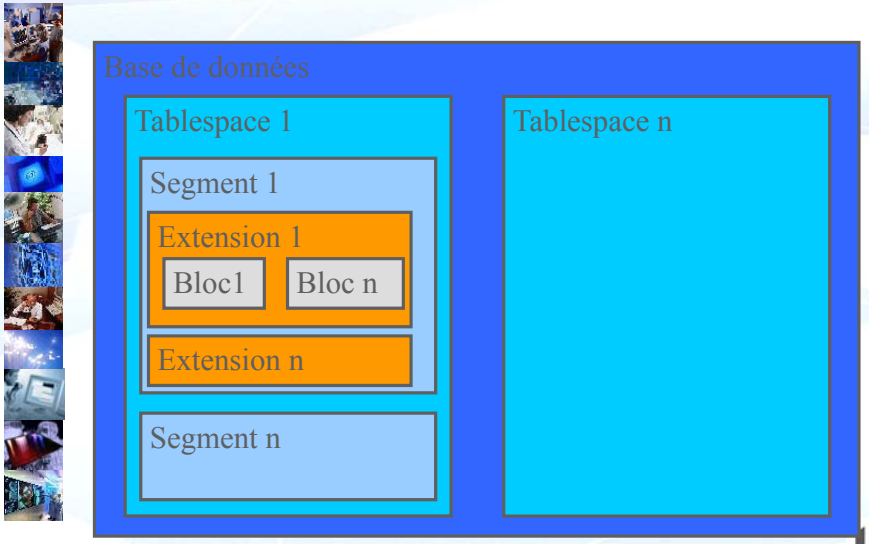
Sécurité des données :

- Protéger les données contre les accès non autorisés ou mal intentionnés
- Prendre en compte les pannes qui pourraient survenir que ce soit au niveau d'un pgm, du système voire de la machine.

=> Vaste problème qui déborde le SGBD

Niveau interne
Illustration avec Oracle

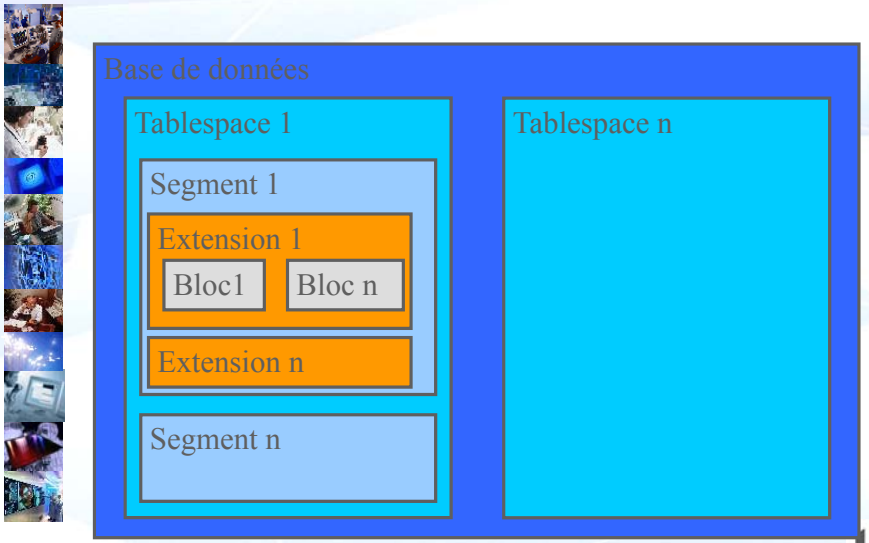
Structure logique (interne) d'une BD - ORACLE



Structure logique (interne) d'une BD - ORACLE

- Une BD doit avoir au moins un Tablespace qui contient le dictionnaire des données ; il est conseillé d'avoir un deuxième Tablespace pour stocker les objets de la BD
- Un tablespace est matérialisé au niveau physique par un ou plusieurs fichiers
- La commande pour créer un Tablespace est CREATE TABLESPACE <ident> DATAFILE <fichier>
- Pour chaque utilisateur de la BD, Oracle permet de définir un Tablespace par défaut ainsi, éventuellement, qu'un espace maximum utilisable sous la forme d'un Quota.
- Lors de la création d'un Tablespace, Oracle réserve toute l'espace disque qui lui est associé ; cet espace est géré dynamiquement au fur et à mesure de l'utilisation de la BD

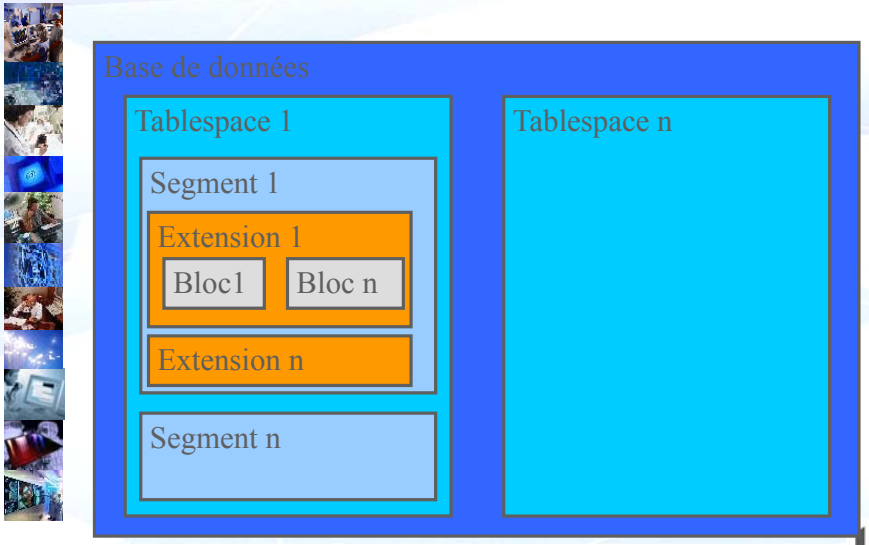
Structure logique (interne) d'une BD - ORACLE



Structure logique (interne) d'une BD - ORACLE

- Le bloc est le niveau de granularité le plus fin ; cette notion de bloc logique est différente du bloc physique utilisée par les systèmes d'exploitation; cependant la taille du bloc logique doit être un multiple de celle du bloc physique
- Format d'un bloc
  - L'en-tête contient des infos sur le bloc (son adresse, le segment auquel il appartient, liste des lignes contenus)
  - L'espace libre est utilisé pour l'insertion de nouvelles lignes et les mises-à-jour (si place nécessaire)

Structure logique (interne) d'une BD - ORACLE



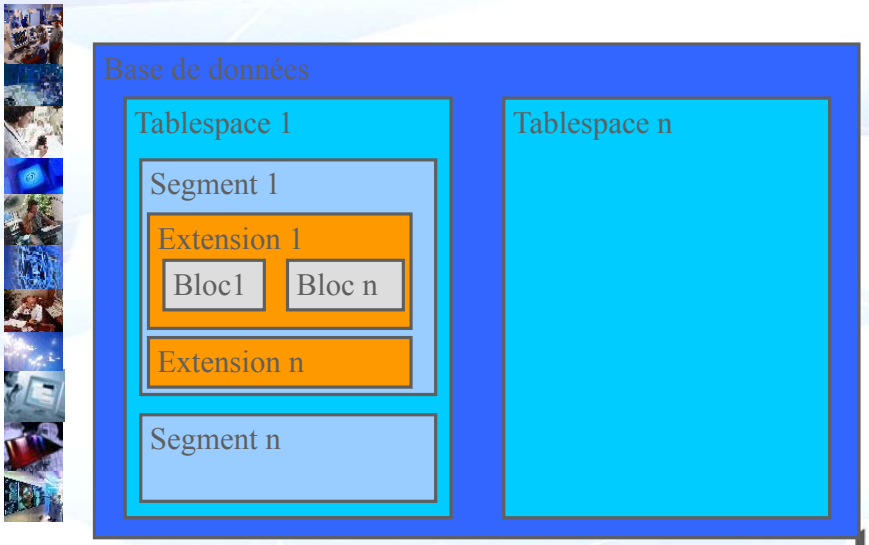
Structure logique (interne) d'une BD - ORACLE

- Gestion de l'espace libre dans un bloc
  - Au fur et à mesure des insertions la partie en-tête et la partie données occupent de plus en plus de place
  - Le paramètre PCTFREE fixe le pourcentage d'espace qui doit rester libre pour les mises à jour éventuelles

  Une valeur élevée entraîne une perte d'espace et un gain de performance (pas de fragmentation)

  Une valeur faible entraîne un gain d'espace mais les performances risquent de se dégrader car les données sont fragmentées entre plusieurs blocs
  - Un deuxième paramètre PCTUSED fixe la limite d'espace libre à atteindre avant de pouvoir recommencer les insertions.

Structure logique (interne) d'une BD - ORACLE



Structure logique (interne) d'une BD - ORACLE

- Notion **d'extension**

Une extension est une unité logique d'allocation d'espace disque composée d'un ensemble contiguë de blocs.

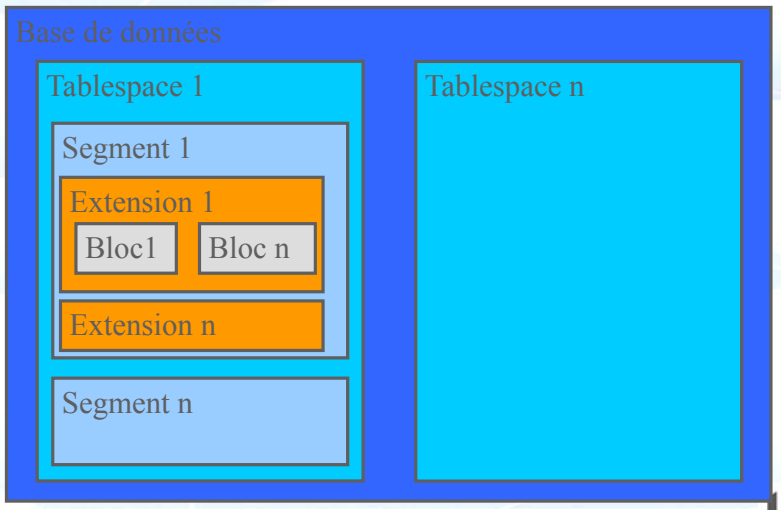
Oracle permet de définir :

- la taille de la première extension (obligatoire)
- la taille des extensions suivantes
- le % d' accroissement de la taille de l'extension (i+1) en fonction de la taille de l'extension (i)
- le nb maximum d'extensions

Lors de l'allocation, Oracle cherche dans le Tablespace contenant le segment une suite contiguë de blocs libres permettant d'allouer l'extension

L'extension n'est libérée que lorsque le segment est supprimé

Structure logique (interne) d'une BD - ORACLE



Structure logique (interne) d'une BD - ORACLE

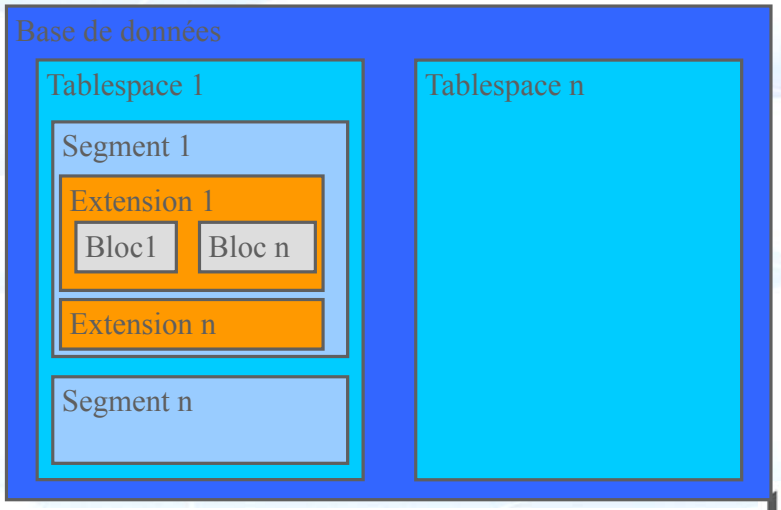
- Notion **de segments**

Un segment est un ensemble d'une ou plusieurs extensions.

Il existe 4 types de segments :

- Segment de données : à chaque table correspond un et un seul segment qui contient toutes ces données
- Segment d'index : à chaque index créé correspond un segment différent (dissocié du segment des données)
- Segment temporaire : utilisé par Oracle pour traiter les commandes SQL nécessitant de l'espace disque (tri, opérateurs ensemblistes, …)
- Segment d'annulation : sert à enregistrer les modifications réalisés par une transaction pour pouvoir faire un ROLLBACK

Structure logique (interne) d'une BD - ORACLE



Dictionnaire de données

Dictionnaire de données

"Structure centralisée contenant la description de tous les objets gérés par le SGBD"

Organisé comme une base de données : l'information est stockée sous forme de table et accessible en SQL (pas de nouveau langage de commandes à apprendre)

En lecture seule pour les utilisateurs

Constitué de
- tables mises à jour par le noyau
- vues reprenant l'information des tables pour les utilisateurs

Dictionnaire de données : exemple

➔ Pas de standard

Liste des tables et vues appartenant à l'utilisateur :

PostgreSQL : SELECT * FROM pg_tables

Oracle : SELECT * FROM USER_CATALOG

MySQL : SHOW TABLES

Dictionnaire de données ORACLE

Vues relatives aux objets de l'utilisateur :
vues préfixées par USER_

Vues relatives aux objets accessibles par l'utilisateur :
préfixées par ALL_

Vues relatives aux administrateurs :
préfixées par DBA_

Vues relatives au suivi des performances :
préfixées par V$_

plus de 300 vues

Dictionnaire de données ORACLE

USER_CATALOG : tables, vues, synonymes et séquences appartenant à l'utilisateur

USER_CONSTRAINTS : contraintes définies sur les tables de l'utilisateur

ALL_CATALOG : idem mais pour les objets accessibles

DBA_CATALOG : tous les objets de la base de données

V$DATABASE : description de la base de données

V$PROCESS : description des processus actifs

Dictionnaire de données ORACLE

Accessibles en SQL comme n'importe quelle table :

Liste des tables

```
SELECT * FROM User_Catalog
```

Liste des clés des relations

```
SELECT uc.constraint_name, uc.table_name,
       ucc.column_name, r_constraint_name
FROM user_constraints uc, user_cons_columns ucc
WHERE uc.constraint_name =        ucc.constraint_name
and (constraint_type='P')
```


SGBD
Sécurité

Sécurité - Gestion des Pannes

- Nature des pannes
  - erreur accidentelle : suppression d'un objet
  - panne sur une commande SQL
  - panne d'un processus
  - panne réseau
  - panne d'instance
  - panne disque

PB : Restaurer la base dans un état cohérent

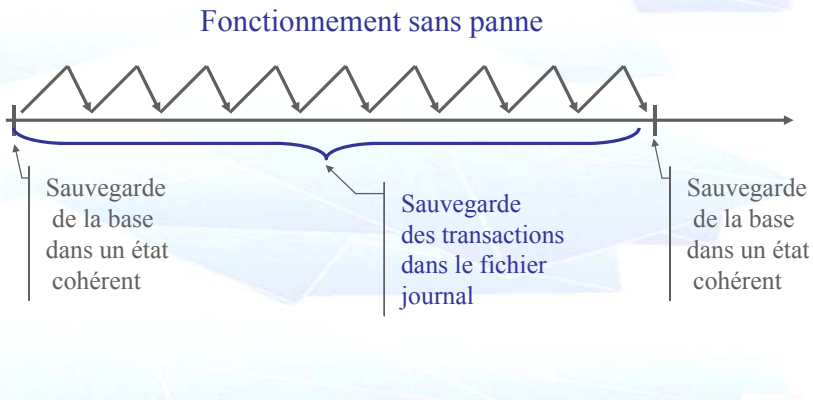
Sécurité - Gestion des Pannes

Structures utilisées pour la restauration

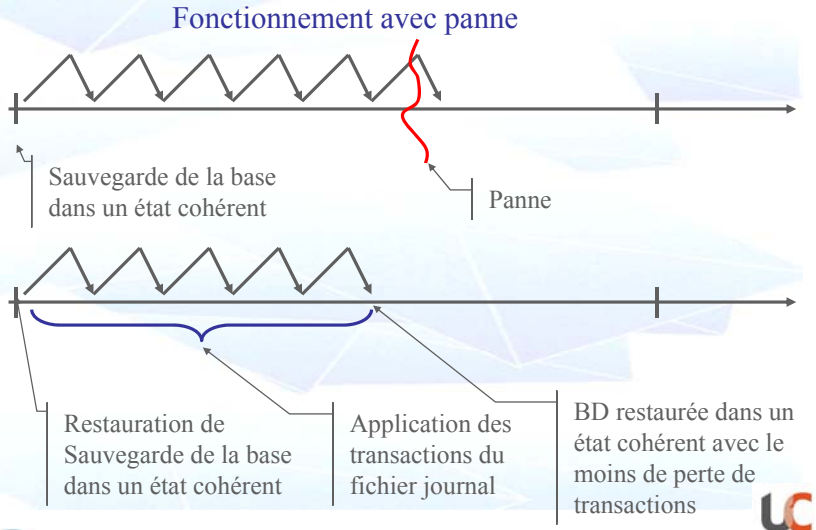
- sauvegarde de la base : copier les fichiers physiques de la BD à l'aide d'une commande système
- journal de reprise : garde une trace des enregistrements modifiés
- fichier de contrôle : stocke les caractéristiques des fichiers indispensables au bon fonctionnement de la BD

Sécurité - Gestion des Pannes

Schéma de sauvegarde



Sécurité - Gestion des Pannes



Gestion des utilisateurs

Sécurité - Gestion des utilisateurs

- Ajouter un utilisateur
  - un nom unique
  - un mot de passe (ou celui du système)
  - un espace par défaut (avec un quota)
  - un profil

  CREATE USER <nom>...
- Modifier un utilisateur

  ALTER USER <nom> ...
- Supprimer un utilisateur

  DROP USER <nom>

Sécurité - Gestion des profils

- Un SGBD étant multi-utilisateur, à un instant donné, un utilisateur peut consommer toutes les ressources au détriment des autres
- Pour limiter ce problème, il est possible de définir des profils et de les attribuer aux utilisateurs
- Un Profil permet de limiter la quantité maximum d'une (ou plusieurs) ressources

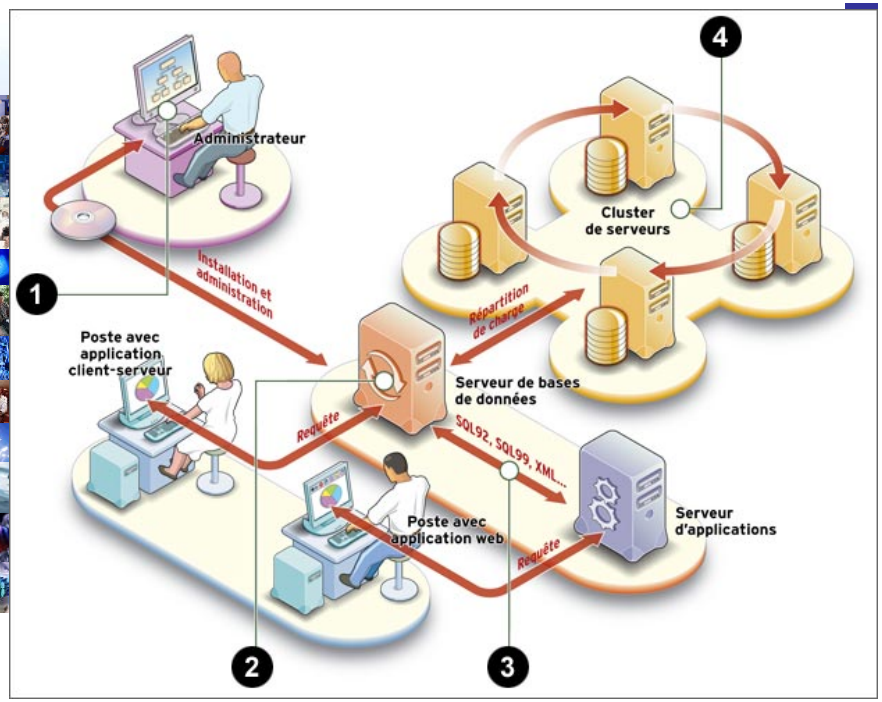
  CPU_PER_SESSION permet de limiter le temps maximum CPU pour une session
- Toute la difficulté est de définir ces quantités maximales

Sécurité - Gestion des privilèges

- Pour mettre en place une stratégie de sécurité, le SGBD propose de nombreux privilèges
  - SELECT, INSERT, UPDATE, DELETE, …
  - CREATE TABLE, CREATE INDEX, CREATE VIEW, CREATE PROCEDURE, …
- Attribuer un privilège

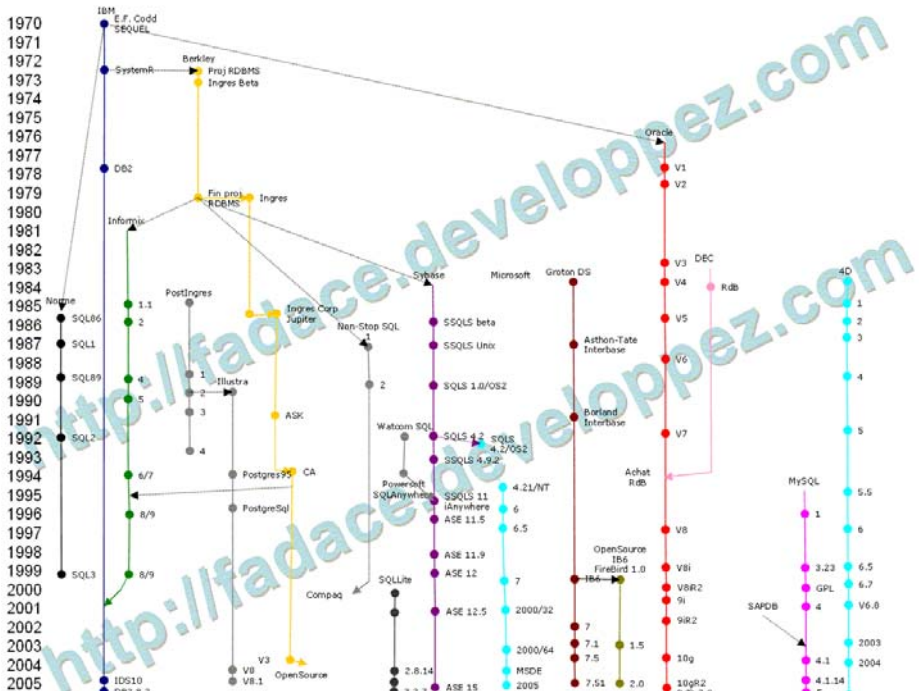
  GRANT … ON … TO
- Enlever un privilège

  REVOKE … ON … FROM …
- La gestion peut devenir très contraignante (si beaucoup d'utilisateurs et de privilèges)



Le marché des SGBD

http://fadace.developpez.com/sgbdcmp/



Quel SGBD choisir ?

- Les leaders
  - Oracle [Unix, Windows, Linux, …]
    - *45 % de parts de marché {en Millions de dollars, 2005}*
  - DB2 (IBM) [Mainframes, Unix, Windows, …]
    - *23 % de parts de marché*
  - SQLServer (Microsoft) [Windows]
    - *13 % de parts de marché*
  - Teradata *[spécialise entrepôts de données]*
    - *3 % de parts de marché*
  - Sybase
    - *3 % de parts de marché*

Quel SGBD choisir ?

- Open source
  - MySQL [Linux, Unix, Windows, …]
  - PostgreSQL [Linux, Unix, Windows, …]
- Pour Micro-ordinateur
  - Access (Microsoft) [Windows]
  - 4D [Mac/OS et Windows]

Quel SGBD choisir ?

- Objet
  - Caché (Intersystem) http://www.sgbd-objet.com/
  - ObjectStore http://www.progress.com/objectstore/index.ssp
- XML
  - Tamino (Software AG) http://www.softwareag.com/
  - XIndice (Open Source) http://xml.apache.org/xindice/

Quel SGBD choisir ?

- Le besoin
  - SGBD adapté aux besoins réels
    - quantité de données, disponibilité, transactions, triggers, procédures stockées, …
- Le système
  - Pour un même SGBD, performances variables suivant le système d'exploitation
- Le prix
  - de Gratuit en Open Source
  - à un Prix très élevé pour Oracle
- Pérennité de l'éditeur
  - Eviter les solutions trop innovantes ou marginales

Développement d'applications

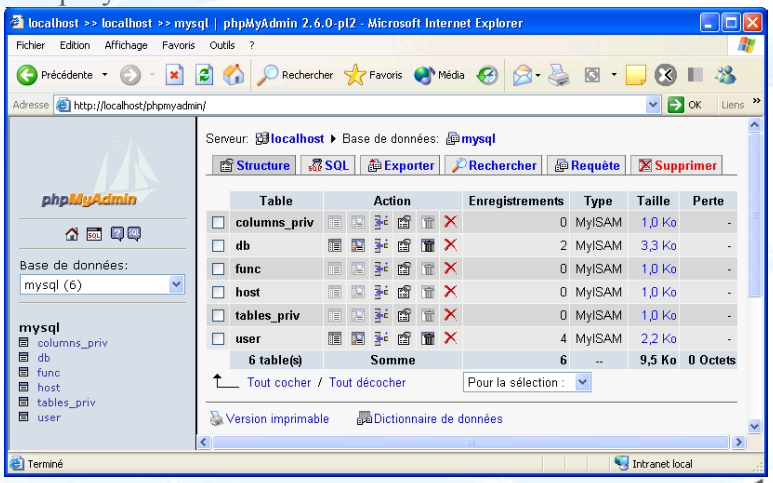
Outils de développement

- Access de Microsoft (produit très abouti pour développement rapide sans programmation)
- Environnement de développement avec L4G spécifique : Windev, DELPHI, KYLIX, …
- Oracle Designer (conception) et Oracle Developper
- Tous (ou presque) les langages de programmation
  - JAVA [JDBC -> SGBD]
  - C++, C#, PHP, PERL, …
- …

Des outils autour du SGBD

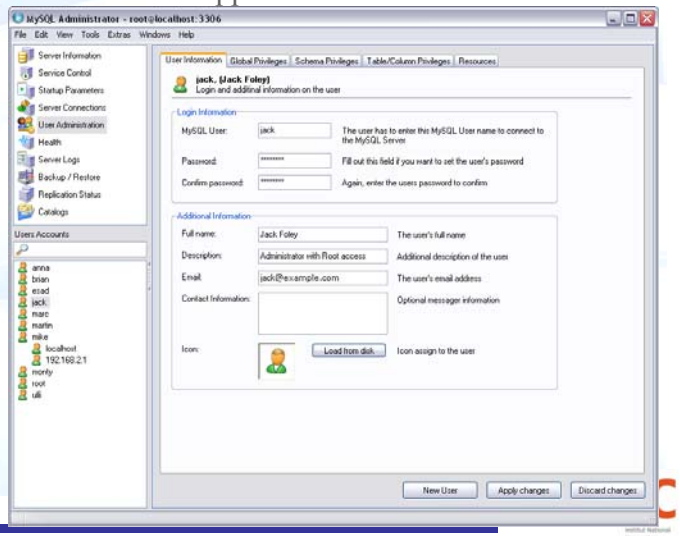
Interfaces d'administration

PhpMyAdmin : interface web



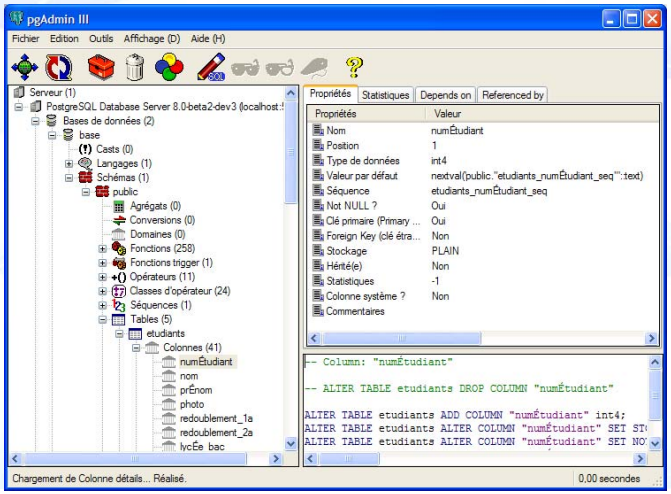
Interfaces d'administration

MySQL Administrator : application



Interfaces d'administration

PostgreSQL : PgAdmin



Interfaces d'administration

Oracle : Enterprise Manager (Web + application)