Introduction aux Bases de Données

Gilles Falquet
Centre universitaire d'informatique
UniGE

Plan

- Idées et Principes
- Le modèle relationnel de données (I)
- Interrogation d'une base de données (SQL)
- Intégrité des données (contraintes)
- Mise à jour des données (SQL)
- Conception physique et logique
- Accès aux BD dans les systèmes d'information

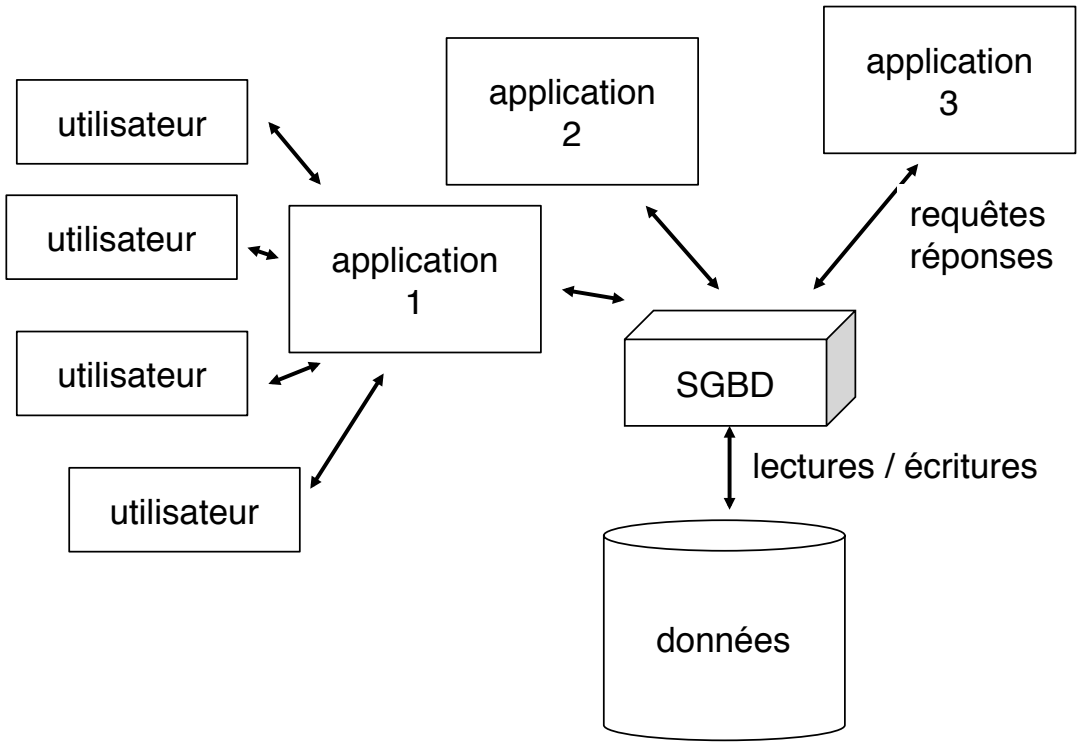
Objectifs des BD

- Stockage permanent de données
- Description des données (logique et physique)
- Pouvoir consulter/sélectionner/modifier les données
- Accès multiples simultanés (concurrents)
- Recherche de l'information par le contenu
- Intégrité des données garantie
- Sécurité, fiabilité (reprise après pannes)
- Confidentialité

Système de Gestion de Base de Données

- Logiciel
- Gère
  - le stockage physique persistant des données
  - les requêtes des utilisateurs / applications
    - recherche de données
    - mise à jour des données
  - les accès concurrents
  - la confidentialité et la sécurité

SGBD



Stockage permanent des données

- Média de stockage persistant
  - disques magnétiques
  - ou combinaison RAM + disques
  - mémoire Flash, cartes à puces, etc.
- Toute mise à jour des données est immédiate et durable
  - pas d'opération "Enregistrer" (Excel, Word, etc.)
- Structuration des données pour obtenir des performances acceptables

Données

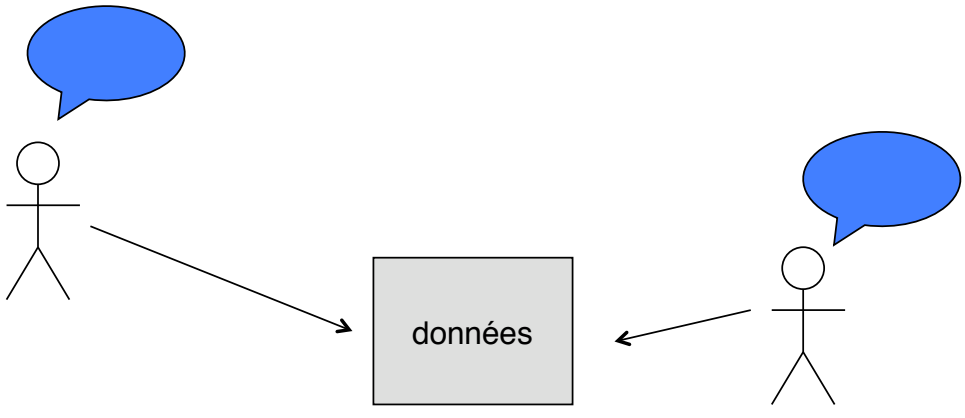
```
A
4 SoCal:hangarWall // polygon # 5
28.7   9.8  -43.7
28.7   0  -43.7
28.7   0   4
28.7   9.8   4 4
SoCal:hangarIn // polygon # 6
-28.6   9.8  -43.7
-28.6   0  -43.7
-28.6   0   4
-28.6   9.8   4
```

```
A
1 version
2 20328  KSBD  L26    2   0.1   0.0   0.0   0.1  N199    Speed-Bird
2 20328  KAVX  KAVX   0   0.1   0.0   0.0   0.0  N9492E  Speed-Bird
2 20328  KSBD  KSBD   0   0.1   0.0   0.0   0.0  N9492E  Glasair II-S
2 20329  KSBD  KSBD   1   0.1   0.0   0.0   0.0  N9492E  Glasair II-S
. . .
```

Comment les interpréter ?

Problème

- Chaque application/utilisateur doit connaitre la description des données.
- On espère qu'ils ont la même.



Principe : Schéma

- Un schéma décrit la structure des données

Base de données = schéma + données

- Le schéma fait partie intégrante de la base
- Les données ne peuvent exister sans le schéma

Données

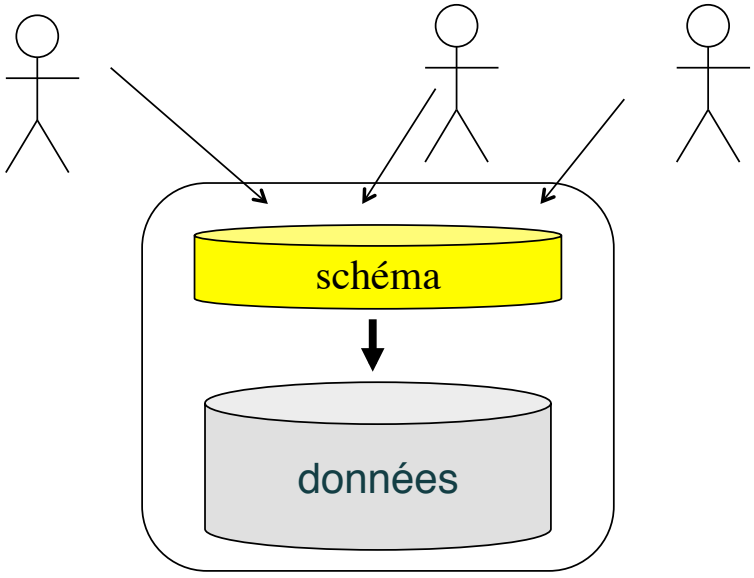
ZK567	GVA	ZRH	43	106
KL1122	AMS	CDG	50	77
KL232	AMS	PPP	560	230
LX441	GVA	NCE	35	101

Données + Schéma

Vol	Départ	Arrivée	Durée	Passagers
ZK567	GVA	ZRH	43	106
KL1122	AMS	CDG	50	77
KL232	AMS	PPP	560	230
LX441	GVA	NCE	35	101

Unicité du schéma

- On voit toujours la base à travers son schéma



Modèle de données

- Outil de structuration des données
- Un schéma s'exprime selon un modèle
- Un modèle de données propose des types de structure
  - arborescentes
  - tabulaires
  - en réseau
  - etc.
- Modèles existants : relationnel, à objets, hiérarchique (xml) , etc.

Modèle relationnel de données

- E. F. Codd (1970)
- Objets simples : table, ligne, colonne
- Basé sur des objets mathématiques bien connus :
  - Relation, n-tuple, ensemble, etc.
- Opérations d'interrogation
  - Sélection, projection, jointure

- Actuellement le modèle le plus répandu (de loin)

Table (relation)

- Ensemble de rangée subdivisées en colonnes
- Chaque colonne porte un nom (attribut)
- !! Pas d'ordre (numéro) des rangée (≠ Excel)

table Vins

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Bordeaux	1990	Bon

Types (domaines) des colonnes

- Les valeurs d'une colonne sont toutes du même type
- Types "standard" : nombre entier, nombre réel (décimal), chaîne de caractère, date

```
create table Vins(
    Région varchar,
    Année integer,
    Qualité varchar,
    PrixMoyen real)
```

Base de données relationnelle

- Ensemble de tables
- Schéma de la base de données
  - Noms des tables
  - Noms de leurs colonnes
  - Types des colonnes
- Exemple (sans les types)
  - Vins(région, année, qualité)
  - Producteur(nom, commune, région, quantité)

Signification des données

- Chaque rangée représente un fait
- On peut l'exprimer sous forme d'une phrase

- Exemple

  Une rangée

R	A	Q

  de la table **Vins(Région, Année, Qualité)**

  signifie

  "*Les vins de la région **R** ont eu une qualité **Q** pendant l'année **A***"

Exemple France-régions-uni

- Region(noreg, nom, lon, lat, gnis)

  "*La région **nom** porte le no. **noreg**, son centre se trouve à la longitude **lon** et à la latitude **lat** son no. GNIS est **gnis**"*
- Dept(nodept, nom, lon, lat, pop, noreg)

  "*Le département **nom** porte le no. **nodept** p, son centre se trouve à la longitude **lon** et à la latitude **lat**, sa population est **pop** et il appartient à la région no. **noreg***"
- Frontiere(region, pays)

  "*La région no. **région** touche le pays nommé **pays***"

(suite)

- Uni(nom, adr1, adr2, nodept, ville, nbetud, subventions, tel, fax)

  "*L'université **nom** se trouve à l'adresse composée des deux lignes **adr1**, **adr2** et de **ville** dans le département no. **nodept**, elle a **nbetud** étudiants inscrits et reçoit **subventions** euros de subventions, son no. de tél est **tel** et son no. fax **fax***"

Interrogation d'une BD

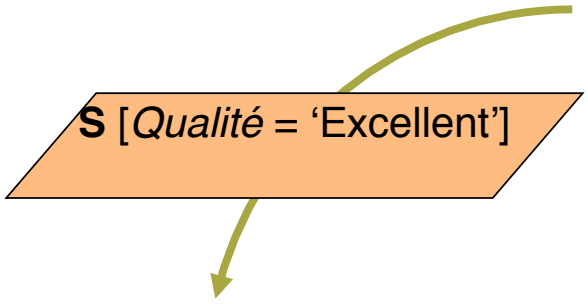
- Extraire des informations d'une base de donnée
- Opérations
  - sélection, projection, jointure, ...
- Langage d'interrogation standard SQL

La sélection

- A partir d'une table
- Ne retenir que les rangées qui satisfont une condition
- Résultat = une table (plus petite)

Exemple

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon

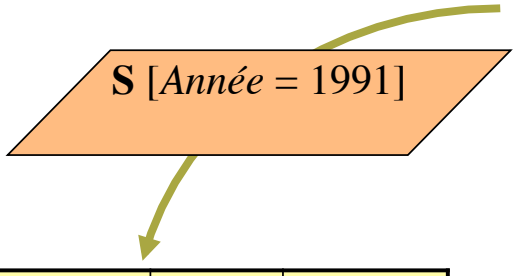


Région	Année	Qualité
Bordeaux	1991	Excellent
Valais	2000	Excellent

select *
from Vins
where Qualité = 'Excellent'

Exemple 2

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon

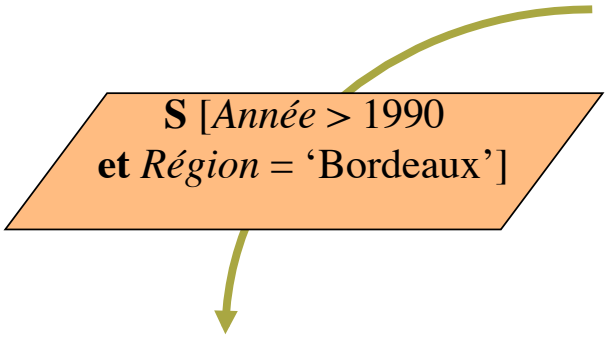


Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	1991	Moyen

select *
from Vins
where Année = 1991

Exemple 3

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon



Région	Année	Qualité
Bordeaux	1991	Excellent

select *
from Vins
where Année > 1990
and Région = 'Bordeaux'

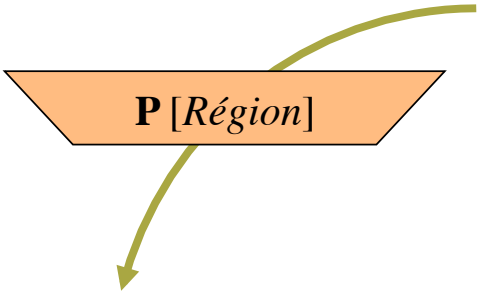
Répondre aux questions

- Quels sont les département dont le no. est supérieur à 80.
- Quelles sont les régions au sud du 45e parallèle ?
- Quels sont les départements au nord du 50e parallèle ?
- Quels sont les départements dont la latitude est supérieure à 47 et la longitude inférieure à 2 ?
- Quelles sont les universités du département 69

La projection

- Supprimer des colonnes d'une table
- Résultat :
  - une table qui a moins de colonnes
- En théorie
  - les doublons sont supprimés (une table est un ensemble)
- En pratique
  - les doublons restent (multi-ensemble)

Exemple



Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon

Région
Bordeaux
Valais
Bourgogne

En SQL

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon

select Région
from Vins

Région
Bordeaux
Valais
Valais
Bordeaux
Bourgogne

Ne correspond pas à la définition théorique
Les doublons sont gardés

En SQL (2)

Région	Année	Qualité
Bordeaux	1991	Excellent
Bourgogne	1991	Moyen
Valais	2000	Excellent
Valais	1991	Moyen
Bordeaux	1990	Bon

select distinct Région
from Vins

Région
Bordeaux
Valais
Bourgogne

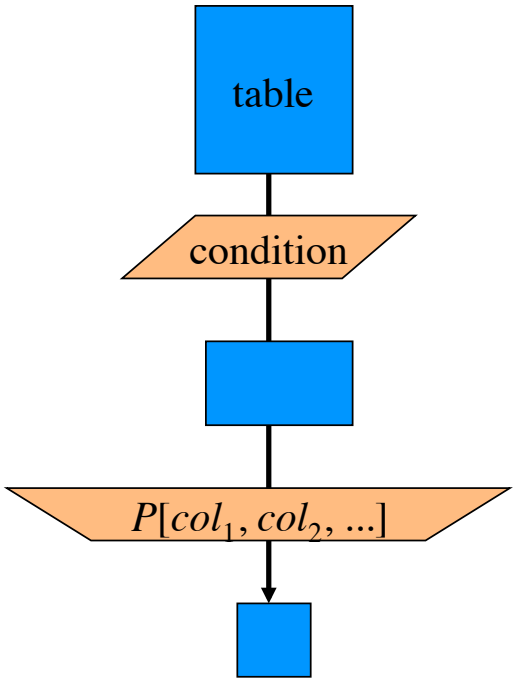
La pratique rejoint la théorie

Exercice

- Noms de tous les départements français
- Noms et adresses des universités de France
- Liste des pays qui touchent la France
  - une seule fois chaque pays

Sélection + Projection en SQL

select colonne1, colonne2, …
from table
where condition

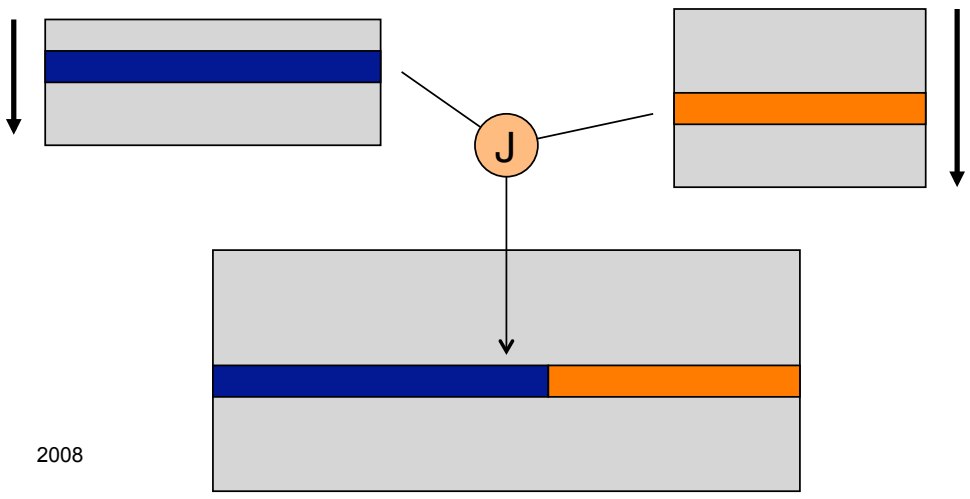


Questions

- Quels sont les noms et latitudes des régions au sud du 45e parallèle ?
- Queles sont les noms, latitude et longitude des départements au nord du 50e parallèle ?
- Quels sont les noms des départements dont la latitude est supérieure à 47 et la longitude inférieure à 2 ?
- Trouvez les noms et numéros de téléphone des universités du département 69

Jointure

- Opération sur deux tables
- But : mettre ensemble les rangées qui correspondent
  - selon une condition



Exemple

- etudiant(noetd, nom)
- inscription(cours, noetd, date)

$$etudiant \; \mathbf{J} \; [noetd = noetd] \; inscription$$

select *
from etudiant, inscription
where etudiant.noetd = inscription.noetd

etudiant

noetd	**nom**
6	Jean
3	Anne
8	Sylvie

inscription

cours	noetd	date
Italien	6	2001
Chimie	3	2002
Italien	7	1999
Chimie	6	2001

select * from etudiant, inscription
where etudiant.noetd = inscription.noetd

noetd	nom	cours	noetd	date
6	Jean	Italien	6	2001
6	Jean	Chimie	6	2001
3	Anne	Chimie	3	2002

Fonctionnement

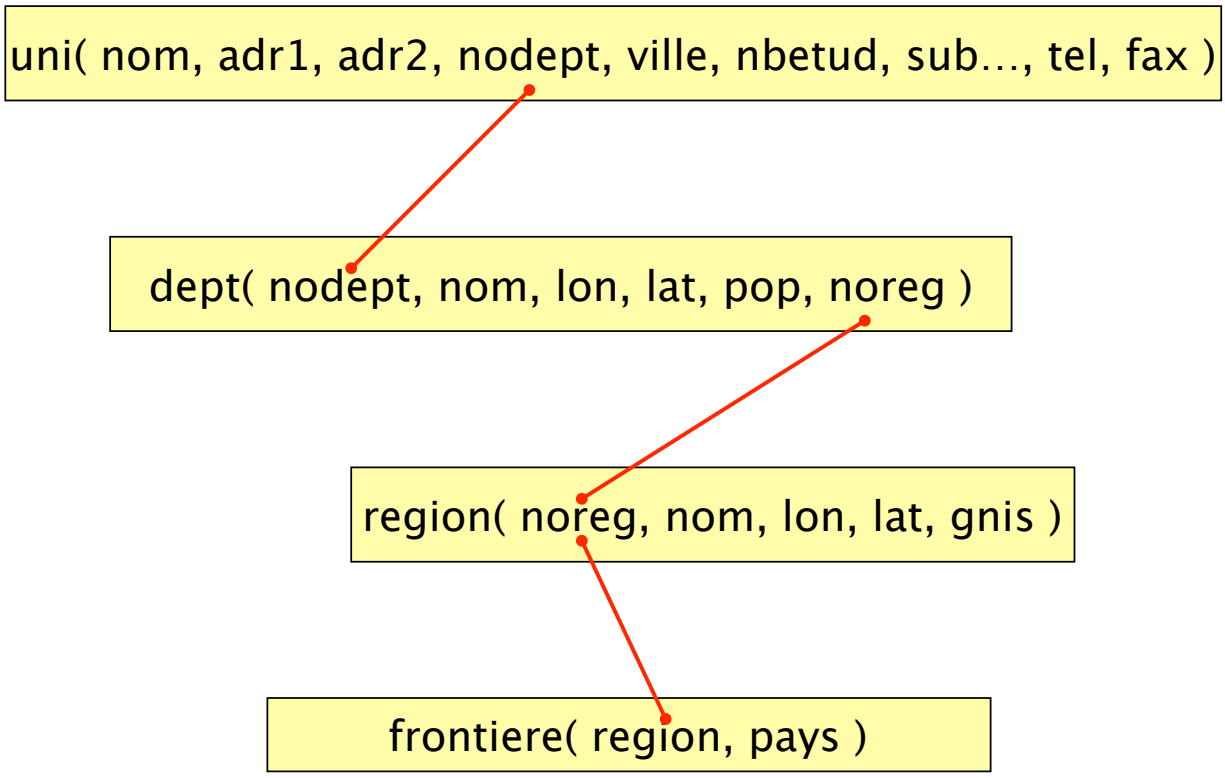
- On peut imaginer que les choses se passent ainsi

- R **J** [condition] S
  1. établir toutes les paires de rangées de R et S possibles
  2. sélectionner les paires qui remplissent la condition

- Remarque.
  - Sans condition on obtient le produit cartésien de R et S
  - c-à-d toutes les combinaisons possibles

Questions

1. Afficher la description de chaque département accompagnée de celle de sa région.
2. Afficher les noms et numéros de département accompagnée du nom de leur région.
3. Trouvez les noms et adresses des universités dans le département de la Gironde.
4. Quels sont les pays touché par la région Rhône-Alpes ?
5. Noms et no. de département des universités se trouvant dans une région qui touche la Suisse
6. Départements dont le centre est à plus de 0,9 degré de latitude du centre de leur région.



Fonctions d'agrégation

- Calculer des valeurs dépendant de toutes les rangées sélectionnées.
- Fonctions : sum(), avg(), count(), min(), max()

- select avg(subvention) from uni
  - moyenne des subventions reçues

- select count(nom) from dept

Agrégation et questions imbriquées

- Trouver les départements dont la population est supérieures à la moyenne nationale.

```
select nom
from dept
where pop > (select avg(pop) from dept)
```

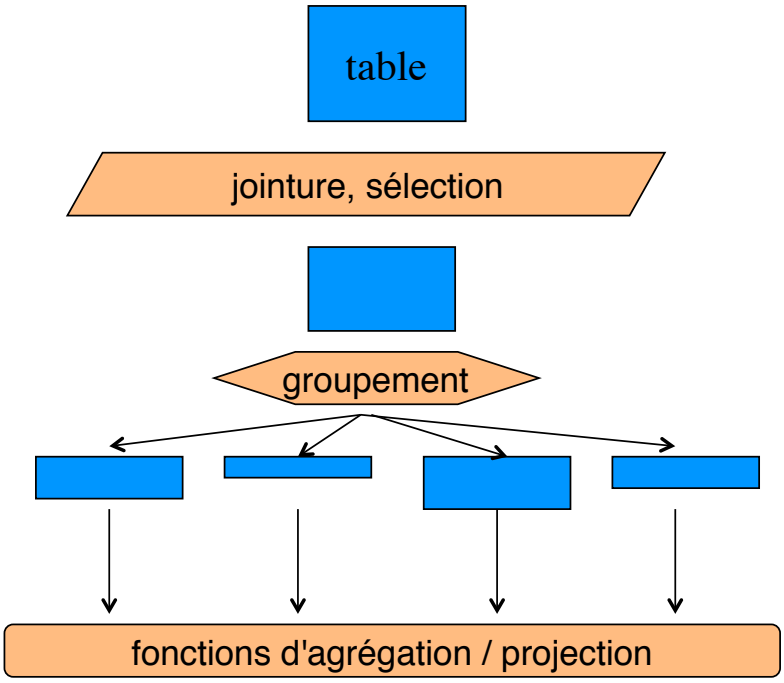

Groupement

- Regroupement des rangées d'un résultat
- Critère : même valeur pour une colonne / expression
- Utilité : uniquement avec des fonctions d'agrégation

```
select noreg, count(nodept) from dept
group by noreg
```

```
select region.nom, count(nodept) from region, dept
where region.noreg = dept.noreg
group by noreg
```

Requêtes avec groupement



Intégrité des données

- Dans la réalité il y a des faits particuliers
- Et des règles générales
  - « Une latitude est comprise entre -90 et +90 »
  - « Une longitude est comprise entre -180 et +180 »
- … ou spécifiques à un domaine
  - « Tous les départements ont des numéros distincts »
  - « Il ne peut pas y avoir plus de 30 inscrits au cours d'Italien »
  - « La somme des salaires ne doit pas excéder le budget »

Dans la base de données

- Les rangées représentent les faits particuliers
  - inscrit(c, n, a) : l'étudiant n est inscrit au cours c pour l'année a

- Les contraintes d'intégrité
  - restreignent le contenu de la base
  - garantissent que la base respecte les règles de la réalité
  - sont vérifiées automatiquement par le SGBD
    - pour certaines catégories seulement !

Règles sur une colonne

- On restreint les valeurs d'une colonne

- interdiction des valeurs indéfinies (NOT NULL)
- définition d'un minimum et d'un maximum
- définition d'une liste fermée de valeurs admises

- Vérifié automatiquement

Contraintes de clé (unicité)

- Colonne clé
  - deux rangées ne peuvent avoir la même valeur pour cette colonne
  - détermine les valeurs des autres colonnes
  - ne peut être nulle

Uni(nom, adr1, adr2, nodept, ville, nbetud, subventions, tel, fax)
clé : nom

Clé (suite)

- Il peut y avoir plusieurs clés
- Et des clés formées de plusieurs colonnes

- Dept(nodept, nom, lon, lat, pop, noreg)
  clés : (nodept), (nom)
- Frontiere(region, pays)
  clé : (région, pays)

- Vérifié automatiquement par le SGBD

Déclaration SQL

```
CREATE TABLE Dept(
    nodept INTEGER,
    nom VARCHAR,
    lon REAL, lat REAL, pop INTEGER,
    noreg INTEGER)
PRIMARY KEY ( nodept )

CREATE TABLE Frontiere(
    region INTEGER, pays VARCHAR)
PRIMARY KEY (region, pays)
```


Contraintes de référence

- « Une université se trouve dans un département qui existe »
- La colonne nodept de la table Uni doit contenir des valeurs apparaissant dans la colonne nodept de Dept

```
CREATE TABLE Dept(nodept INTEGER, …
PRIMARY KEY ( nodept ))

CREATE TABLE Uni(nom VARCHAR, nodept INTEGER, …
FOREIGN KEY ( nodept ) REFERENCES Dept ( nodept ))
```

Uni

nom	nodept	….
PARIS VI	75	…
LYON III	204	…
PARIS X	75	…
U. Corse	20	…

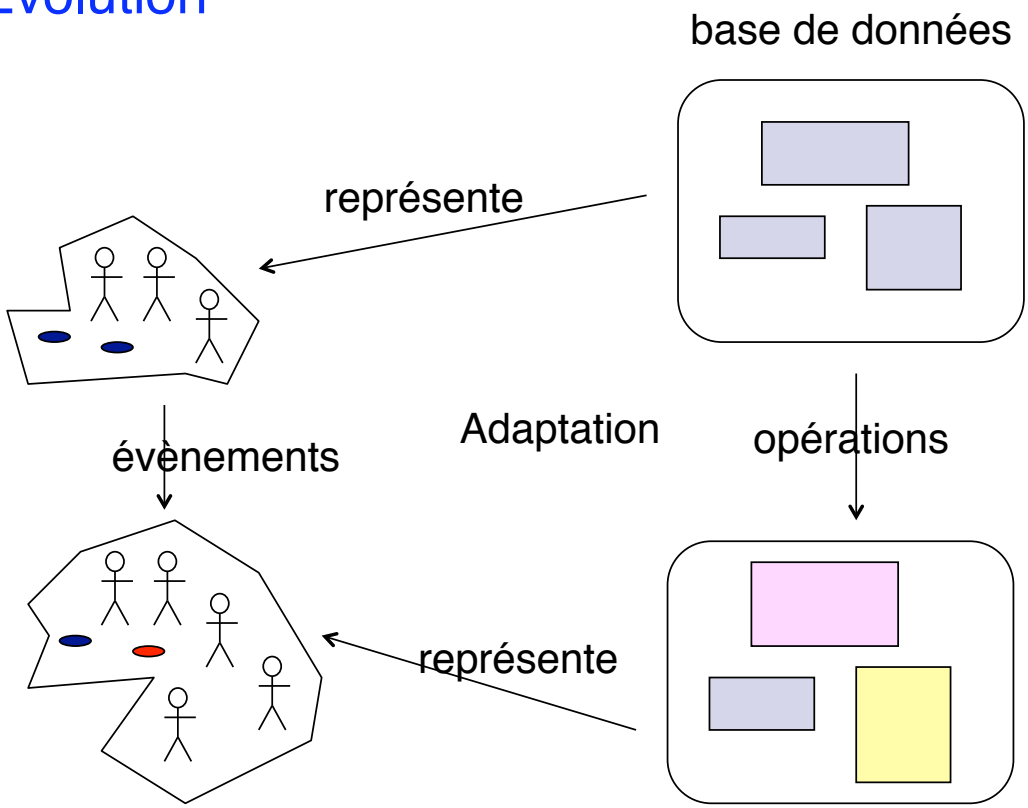
Dept

nodept	**pop**	**nom**
1	634554	Ain
…		
20	323003	Corse
…		
75	5324443	Seine
…		
95		

Mise à jour des données

- Les données représentent une réalité
  - Cette réalité change au cours du temps
  - => Il faut adapter les données
- Les données "définissent" une réalité
  - p.ex. création d'un emploi du temps
- Importance de garantir le respect des règles d'intégrité

Evolution



Opérations sur les données

- Ajouter une rangée
- Supprimer une rangée
- Modifier une rangée

- Opérations complexes (combinaisons)

- Doivent respecter les contraintes d'intégrité

Ajout d'une rangée

insert into table(col1, col2, …)
  values (val1, val2, …)

- Ajoute une rangée dans table
- Dans cette rangée col1 vaudra val1, etc.

- Exemple

insert into inscrit(noetd, cours, année)
  values (3, 'Physique 2', 2002)

Suppression de rangées

delete from table
  where condition

- Supprime toutes les rangées de table qui satisfont la condition

- Exemple

delete from uni where nom = 'Université de Genève'

delete from uni where nbetud < 10000

Modification de rangées

update table
set col1 = val1, col2 = val2, …
where condition

- Pour toutes les rangées de table qui satisfont la condition
- Modifie les valeurs des colonnes col1, col2, …

Exemples

update Vins **set** qualité = 'Excellent'
where région = 'Valais'

update Uni **set** subventions = subventions * 2

update Uni **set** subventions = subventions * 5
where nbetud > 15000

Exercices

- Supprimer les subventions des universités des départements au nord de Paris.
- Augmenter de 25% (x 1.25) les subventions des universités du département 75
- Ajouter 2 nouvelles universités dans le département 20.

Conception physique et logique

- Niveau logique : schéma de la base
- Niveau physique :
  - comment les données sont réellement stockées sur le disque / en mémoire
  - structures (complexes) prévues pour obtenir des performances maximales et un encombrement minimal
- Indépendance physique
- Controle de la structure physique et performance
- Quelques idées de conception logique

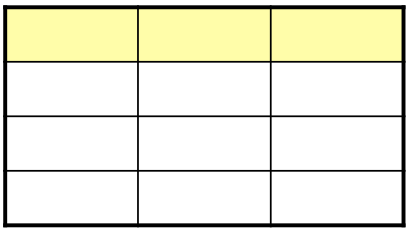
Indépendance physique

- Concept apparu avec le modèle relationnel
- L'utilisateur/développeur voit uniquement le schéma de la base
- Les détails du stockage physique sont cachés

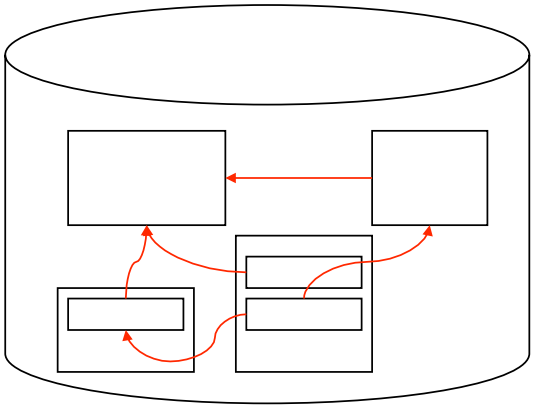
AVANTAGES ENORMES

- écriture des requêtes indépendante du stockage
  - c.f. exercices avec hsqldb
- amélioration du stockage sans changement des applications

On voit ça :



Et c'est stocké comme ça :



Contrôle du stockage physique

- Par défaut la plupart des SGBD utilisent une méthode de stockage simple mais peu efficace
  - => p.ex. une sélection nécessite l'inspection de toutes les rangées

- Pour améliorer les performance on peut
  - créer des index
  - fixer divers paramètres (dépendant du SGBD)

Index

- Structure pour accélérer la recherche
- Dans une colonne ou un groupe de colonnes

create index idx **on** table (colonne1, …)

- 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

Exemple

select * from etudiant where nom = 'Julie'

- Il faut examiner toutes les (100000) rangées

create index nom_etu on etudiant(nom)

- l'index est créé une fois pour toutes

select * from etudiant where nom = 'Julie'

- la recherche via l'index des noms est très rapide

Optimisation des requêtes

- On peut écrire des requêtes extrêmement coûteuses à évaluer
  - multiples jointures, groupements, etc.
  - longtemps le problème no. 1 des SGBD
- Les (bons) SGBD optimisent les requêtes
  - en profitant des index - pour autant qu'ils existent !
  - en réordonnant les opérations, etc.

Conception logique

Comment concevoir un bon schéma pour ma base ?

- Il n'y a pas de recette absolue
- Il y a des techniques pour évaluer la qualité d'un schéma
- Il y a de grands principes
  - Eviter la redondance
    - lutte contre les anomalies de mise à jour
  - À chaque type d'objet sa table
    - lutte contre les anomalies d'ajout et de suppression

Un mauvais exemple

projet	produit	fournisseur	adresse_f	nb_pieces
pr1	écrou	Paul	Genève	10
pr1	boulon	Paul	Genève	10
pr2	vis	Pierre	Lausanne	50
pr3	rondelle	Jean	Genève	22
pr3	boulon	Paul	Genève	44
pr3	écrou	Pierre	Lausanne	33

Anomalie de mise à jour

- On veut mettre à jour l'adresse du fournisseur Paul
- Il faut le faire dans toutes les rangées où Paul apparaît
- Risque : se retrouver avec deux adresses différentes pour Paul
- Dû à la redondance

Anomalie de suppression

- On supprimer les rangées concernant le projet pr3 (il est terminé)
- => On perd les informations sur le fournisseur Jean (adresse)

Anomalie d'insertion

- On veut enregistrer le nom et l'adresse du fournisseur Marie
- Si Marie ne fournit actuellement aucun projet, on ne peut pas le faire

Une solution : la décomposition

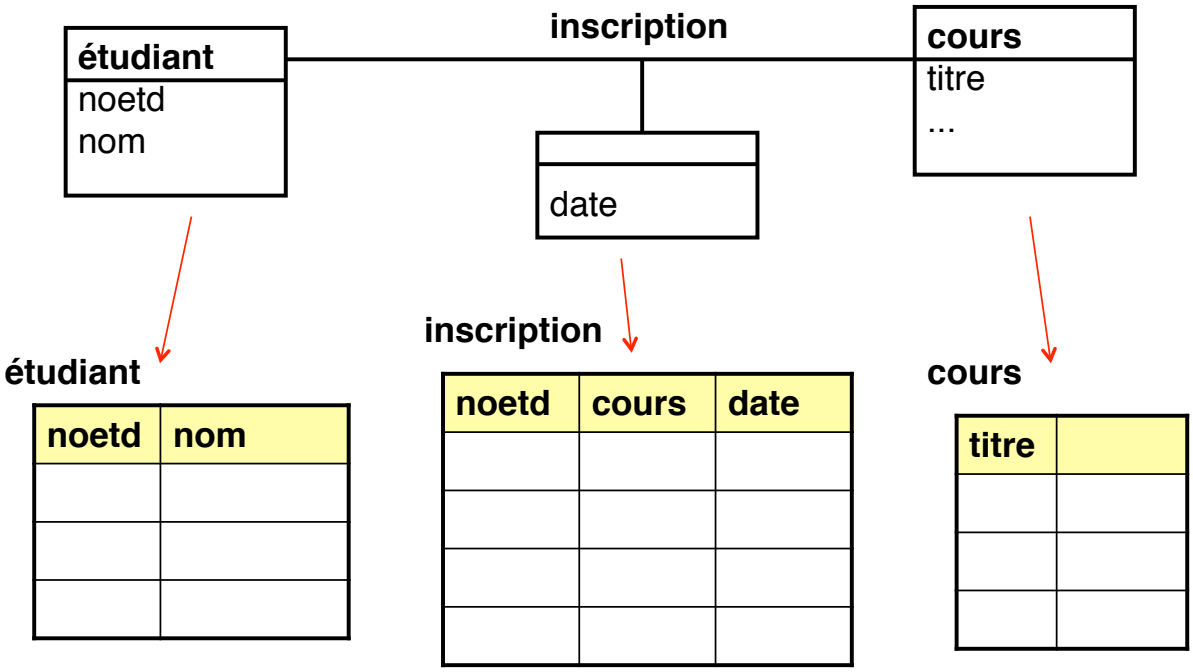
projet	produit	fournisseur	nb_pieces
pr1	écrou	Paul	10
pr1	boulon	Paul	10
pr2	vis	Pierre	50
pr3	rondelle	Jean	22
pr3	boulon	Paul	44
pr3	écrou	Pierre	33

fournisseur	adresse
Paul	Genève
Pierre	Lausanne
Jean	Genève

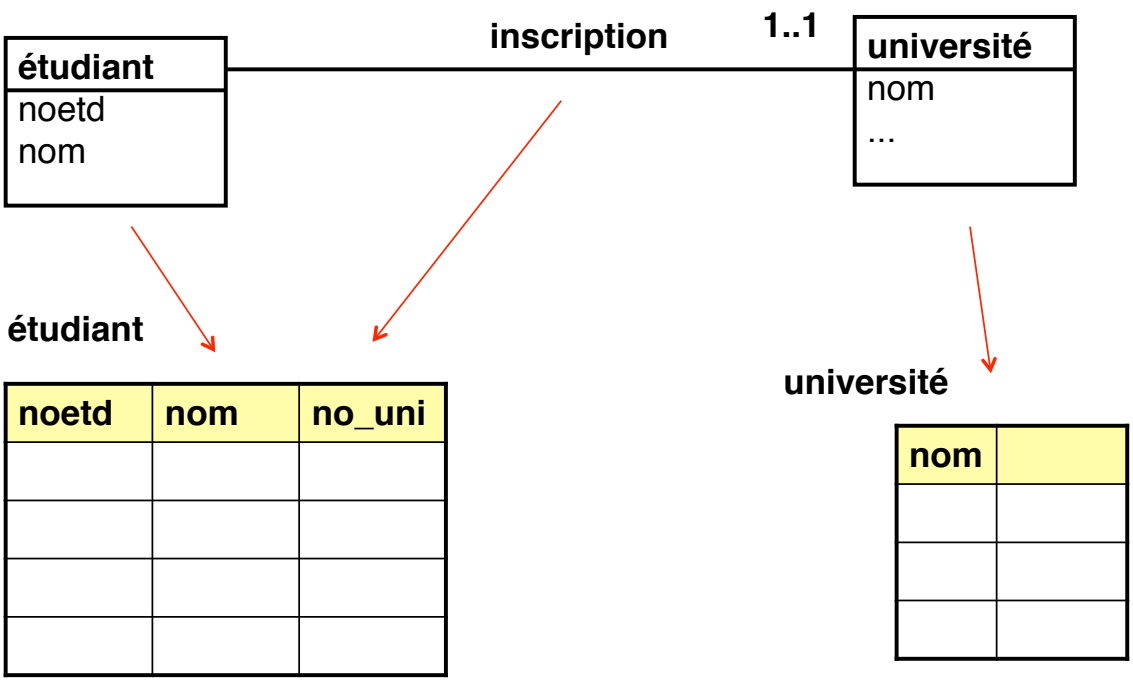
Démarche de conception

- Conception orientée objet ou entité-relation
  - mise en évidence des entités et relations
  - résolution des problèmes de redondance etc.
- Traduction des diagrammes UML (ou autre) en relationnel
- Event. étude et amélioration du schéma relationnel

UML --> schéma BD

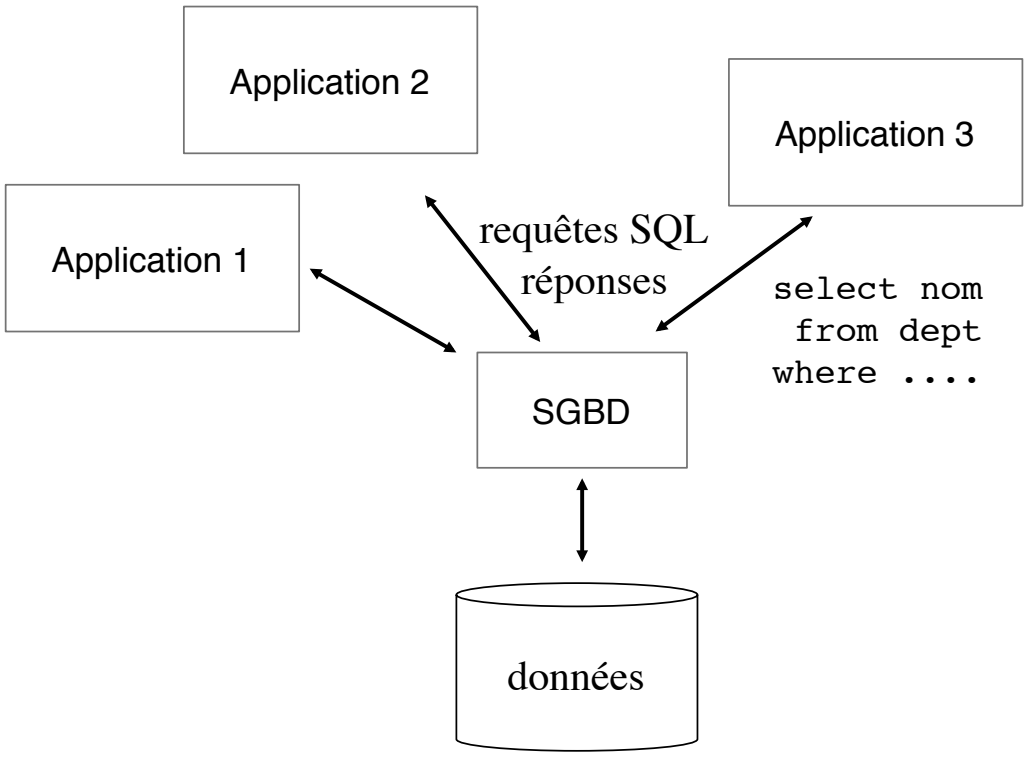


Cas simple

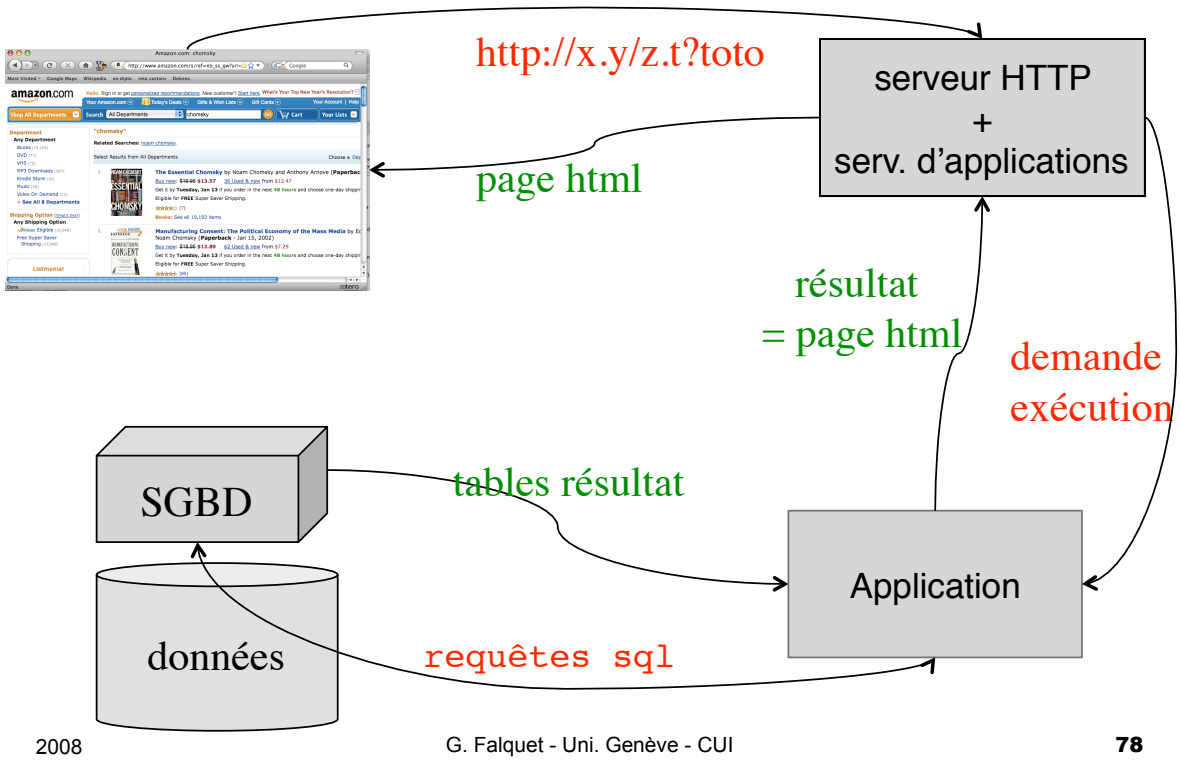


Intégration dans un système d'information

- Les utilisateurs accèdent rarement directement aux BD
- Les applications accèdent aux BD
- Ou bien les applications accèdent à un service qui lui-même utilise une BD
- L'interface standard entre applications et BD est SQL
  - les langages de programmation possèdent des interfaces SQL
  - ou des extensions SQL



Architecture web



Architecture Web : exemple PHP

- Un page PHP contient
- Du texte html
- Des sections d'instructions PHP

<html>
…
<?php …. ?>
…
</html>

Traitement des pages php

1. Réception d'une demane d'URL (http://…/x.php)
2. Lecture de la page x.php
3. Analyse et exécution des sections <?php … ?>
4. Remplacement des sections php par leur résultat
5. Envoi de la page HTML finale au client

Exemple

```
<html>
…
<h1>Voici la liste des employés :</h1>

<?php
// connexion à la base
$link = mysql_connect("localhost", "developpez", "pass");
mysql_select_db("developpez", $link) or die(mysql_error());

// définition et exécution de la requête
$query = "SELECT nom, prenom FROM emps";
$result = mysql_query($query, $link) or die($query . " - " . mysql_error());
```



```
// récupération et affichage des résultats
    while ($tab = mysql_fetch_array($result)) {
        echo $tab['nom'] . " : " .  $tab['prenom'];
        echo "<br />";
    }

?>
… suite de la page
</html>
```

Autres thèmes importants

- Transactions : gérer les accès concurrents aux données
- Reprise après pannes
- Bases de données réparties
- Données multimédia et bases de documents
- Confidentialité et sécurité