

## Repère 6 - Immersion

1. Concepts de base
2. Historique rapide de la gestion de données
3. Création d'une base de données relationnelle
4. Interrogation de la BDR
5. Gestion des transactions
6. Immersion de SQL dans un langage hôte

## Evolutions des SGBD

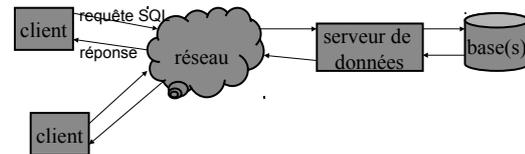
- Bases de données réparties
- Bases de données, architecture client/serveur
- WEB et bases de données

## Bases de données réparties

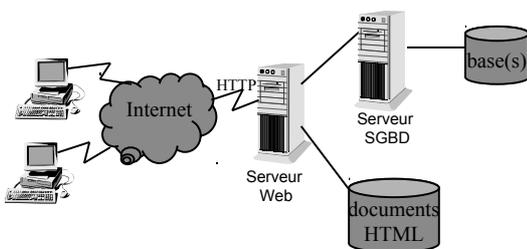
- Une base de données répartie est une collection de bases de données logiquement reliées et physiquement distribuées sur un réseau.
- Un SGBD réparti est un SGBD assurant la transparence vis-à-vis de la répartition.

## Bases de données architecture client/serveur

- Une application serveur réalise des fonctions pour le compte et à la demande d'applications clientes.



## Le Web et les bases de données



## Le Web et les bases de données

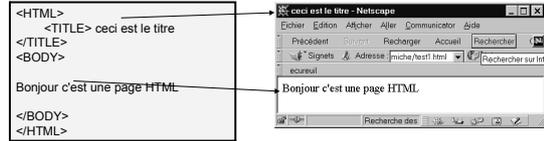
- HTML (Hyper Text Markup Language)
  - Cours d'introduction à HTML, CRI-USTL
  - HTML avancé, CRI-USTL
  - <http://nephi.unice.fr/CoursHTML/>
- PHP (Personal Home Page)
  - Programmation en PHP par Leon Atkinson
  - <http://www.php.net/manual/fr/>
- Postgres
  - <http://www.linux-france.org/article/serveur/psql/guide-utilisateur/book1.html>

# HTML

- **Internet:** le plus grand réseau informatique du Monde.
  - de nombreux services: e\_mail, news, ftp, Web (World Wide Web)
  - modèle client/serveur:
    - le client demande le service, le serveur fournit ce service
- **HTML:** un langage du type à balises (commandes ou tags)
  - Deux formes:
    - englobante: `<xx> ..... </xx>`
    - simple: `<yy>`

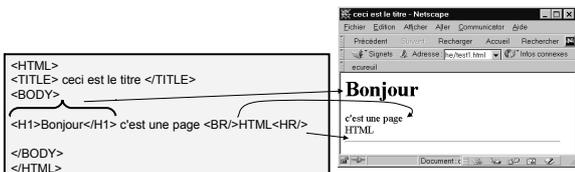
# HTML

- **Structure générale d'un document**
  - Le document `<HTML>...</HTML>`
  - Le titre `<TITLE>...</TITLE>`
  - Le corps `<BODY>...</BODY>`



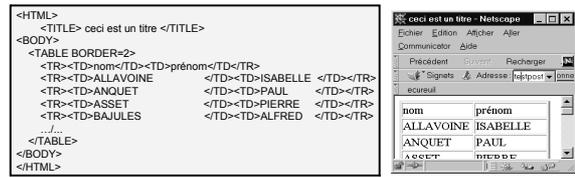
# HTML

- césure forcée `<BR>`
- ligne de séparation `<HR>`
- borner les en-têtes `<Hn>...</Hn>`



# Tableaux HTML

- déclaration `<TABLE >...</TABLE >`
- nouvelle ligne `<TR>...</TR>`
- nouvelle cellule `<TD>...</TD>`

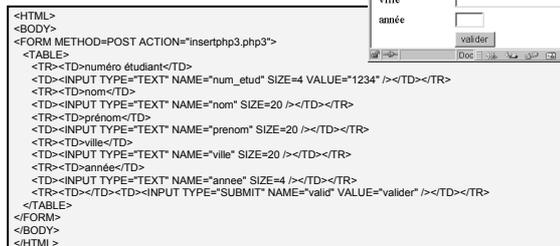


# HTML

- **Acquisition**
  - formulaire `<FORM`
    - méthode = POST ou GET
    - adresse W3 = adresse W3 de l'action à réaliser
    - description des entrées
  - l'entrée `<INPUT TYPE=type`
    - type = TEXT, SUBMIT, PASSWORD, RADIO, etc...
    - NAME=nom\_de\_la\_variable
    - SIZE=taille\_zone\_dialogue
    - VALUE=valeur\_initiale />
  - Liste déroulante de choix :
    - `<SELECT name=""Role">`
    - `<OPTION VALUE="enseignant">Enseignant</OPTION>`
    - `<OPTION VALUE="etudiant">Etudiant</OPTION>`
    - `<OPTION VALUE="ingenieur">Ingénieur</OPTION>`
    - `</SELECT>`

# HTML

- **Acquisition**

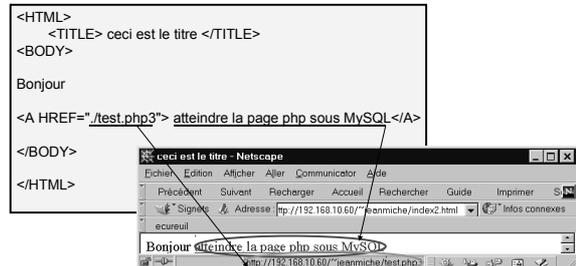


## URL

- URL (Uniform Resource Locator)
  - Identification de n'importe quelle ressource de l'internet de façon standard.
  - Création de liaisons hypertextes.
- Les ancres `<A>...</A>`
  - `<A HREF="URL"> texte </A>`
    - Le texte placé entre les bornes `<A>` et `</A>` devient une zone cliquable qui va amener l'utilisateur vers le document dont l'URL est la valeur de l'argument HREF

## Les liens

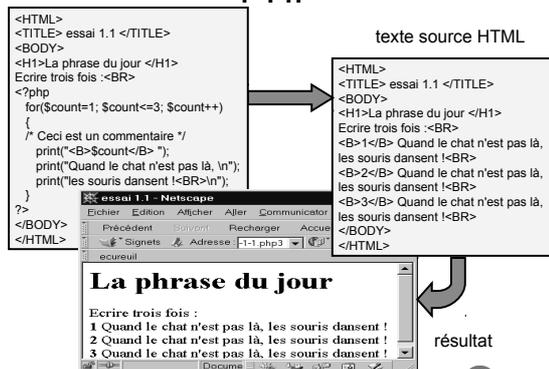
- Les ancres `<A>...</A>`



## PHP

- Fonctionne sous Unix, Windows 95/98, Windows NT et Mac.
- Est dédié au développement de pages Web dynamiques
- Est rapide à programmer et à exécuter

## PHP



## PHP

- D'importantes bibliothèques disponibles
  - fonctions d'entrée/sortie.
  - création de tableaux, recherche dans des chaînes, encodage et décodage.
  - fonctions standards mathématiques.
  - fonctions relatives au temps.
  - fonctions graphiques
  - fonctions dédiées aux bases de données.

## PHP

- Fonctions dédiées aux BD.
  - PHP supporte différentes bases de données
    - dbase
    - LDAP
    - mSQL
    - MySQL
    - Oracle
    - Postgres
    - Sybase etc...

## PHP

- Fonctions dédiées à Postgres.

- Connexion à la base de données

```
$conx = pg_connect("host=nom_machine
                    user=nom_login
                    password=mot_de_passe
                    dbname=nom_base");
```

```
<?php
$machine='weppes.studserv.deule.net';
$user='fhoogsto';
$pwd='****';
$db='fhoogstoDB';
?>
```

```
.../...
<?php
include('database.params.inc');
$link = pg_connect("host=$machine user=$user password=$pwd dbname=$db")
or die("Erreur de Connection !<br />".pg_last_error());
.../...
?>
```

## PHP

- Fonctions dédiées à Postgres.

- Exécution d'une requête sur la BD.

```
$result = pg_exec(entier connexion,
                  chaîne requête);
```

```
<?
.../...
$query = " SELECT * FROM etudiant ";
$query = " ORDER BY e_nom ";

$result = pg_Exec($link, $query) or die("Erreur SQL !<br />".$query.<br />".pg_last_error() );
.../...
?>
```

## PHP

- Fonctions dédiées à Postgres.

- Nombre de lignes du résultat.

```
$num = pg_NumRows(entier resultat);
```

- Extraction de la valeur de l'attribut  
chaîne **pg\_Result**(entier resultat,  
entier rangée,  
chaîne champ);

Les rangées et champs sont numérotés à partir de 0

## PHP

- Fonctions dédiées à Postgres.

```
<?
.../...
$num = pg_NumRows($result);

$li = 0;
echo "<TABLE BORDER=2>";
print("\n");
echo "<TR><TD>nom</TD>";
echo "<TR><TD>prénom</TD>";
echo "<TR><TD>ANQUET </TD>";
echo "<TR><TD>ASSET </TD>";
while ($li < $num)
{
    echo "<TR><TD>";
    echo pg_Result($result, $li, "e_nom");
    echo "<TD><TD>";
    echo pg_Result($result, $li, "e_prenom");
    echo "<TD></TR>";
    print("\n");
    $li++;
}
echo "</TABLE>";
.../...
?>
```

```
<HTML>
<TITLE> essai 1.2 </TITLE>
<BODY>

<TABLE BORDER=2>
<TR><TD>nom</TD><TD>prénom</TD></TR>
<TR><TD>ALLAIVOINE </TD><TD>ISABELLE </TD></TR>
<TR><TD>ANQUET </TD><TD>PAUL </TD></TR>
<TR><TD>ASSET </TD><TD>PIERRE </TD></TR>
<TR><TD>BAJULES </TD><TD>AI FRED </TD></TR>
```



nom	prénom
ALLAIVOINE	ISABELLE
ANQUET	PAUL
ASSET	PIERRE

## PHP

- Formulaire HTML pour récupérer des données auprès de l'utilisateur

```
<HTML>
<HEAD>
<TITLE>Test de passage d'information par formulaire</TITLE>
</HEAD>
<BODY>
<FORM METHOD= "POST" ACTION="test2.php">
cle : <input type="text" name="an_aut" maxlength="3" size="3" />
nom : <input type="text" name="anom" maxlength="20" size="15" />
prenom : <input type="text" name="aprenom" maxlength="20" size="10" />
lieu : <input type="text" name="alieu" maxlength="20" size="10" />
<input type="reset" name="effacer" value="Effacer" />
<input type="submit" name="envoyer" value="Envoyer" />
</FORM>
</BODY>
</HTML>
```

## PHP

- Récupération des paramètres (php3.4)

```
.../...
<?
$machine='weppes.studserv.deule.net'; // serveur Postgres pour TP Polytech
$user='jduthill';
$pass='essai';
$nameDb='biblio';
print ("vous allez entrer $aprenom $anom dans la base de donnee");

//ouverture de la base
$link=pg_connect("host= $machine user=$user password=$pass dbname=$nameDb") \
or die("Erreur de Connection !<br />".pg_last_error() );

//insertion des donnees
$query="INSERT INTO auteur VALUES ('$an_aut','$anom','$aprenom','$alieu')";
$result=pg_exec($link,$query) or die("Erreur SQL !<br />".$query.<br />".pg_last_error() );
.../...
?>
```

Extrait du  
fichier  
test2.php

## •Récupération des paramètres (php 4,5)

228

```
./...
<?
$machine="weppes.studserv.deule.net"; // serveur Postgres pour TP Polytech
$user=jduthill;
$pass=essai;
$dbname=biblio;
print ("vous allez entrer $aprenom $anome dans la base de donnee");

//ouverture de la base
$link=pg_connect("host= $machine user=$user password=$pass dbname=$dbname") \
or die("Erreur de Connection !<br />".pg_last_error());

// récupération des données du formulaire
extract($_POST);

//insertion des données dans la base
$query="INSERT INTO auteur VALUES ('$anome','$aprenom','$alieu')";

//exécution de la requête
$result=pg_exec($link,$query) or die("Erreur SQL !<br />".pg_last_error());
.../...
```

Extrait du  
Fichier  
test2.php

## Postgres sous PHP - Résumé

229

### Quelques primitives

```
// permet d'ouvrir une connexion à une Base de Données
entier pg_connect(chaîne params_connexion);

// ferme la connexion à une base de données
booléen pg_close(entier conx);

// permet de faire exécuter une requête SQL via une connexion
entier pg_exec(entier conx, chaîne requête);

// donne la valeur d'un champ d'une ligne d'un résultat
entier pg_result(entier result, entier n_ligne, chaîne champ);

// donne les valeurs des champs d'1 rangée
tableau pg_fetch_row(entier result, entier rang);

// retourne le nombre de lignes du résultat
entier pg_numrows(entier result);

// retourne la dernière erreur provoqué par un appel à une fonction postgres :
texte pg_last_error()
```