

# M2106 : Programmation et administration des bases de données

Cours 6/6 – Le SGBD au service du SI

Guillaume Cabanac

[guillaume.cabanac@univ-tlse3.fr](mailto:guillaume.cabanac@univ-tlse3.fr)



## 1 Architecture client-serveur

### 2 Client lourd

- Communication Java  Oracle

### 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

### 4 Bilan du M2106

### 5 Perspectives en S3, LP AGBD et plus...

# SGBD au service du SI

## 1 Architecture client-serveur

### 2 Client lourd

- Communication Java  Oracle

### 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

### 4 Bilan du M2106

### 5 Perspectives en S3, LP AGBD et plus...

# Architecture client-serveur

Le programme **client** développé dans un langage quelconque (C, C++, Java, PHP, etc.) réalise divers traitements, dont une partie est dévouée au **serveur** :

- connexion au SGBD *via* le réseau,
- émission d'ordres SQL : majoritairement LID et LMD,
- appel des sous-programmes stockés.

# Architecture client-serveur

Le programme **client** développé dans un langage quelconque (C, C++, Java, PHP, etc.) réalise divers traitements, dont une partie est dévouée au **serveur** :

- connexion au SGBD *via* le réseau,
- émission d'ordres SQL : majoritairement LID et LMD,
- appel des sous-programmes stockés.

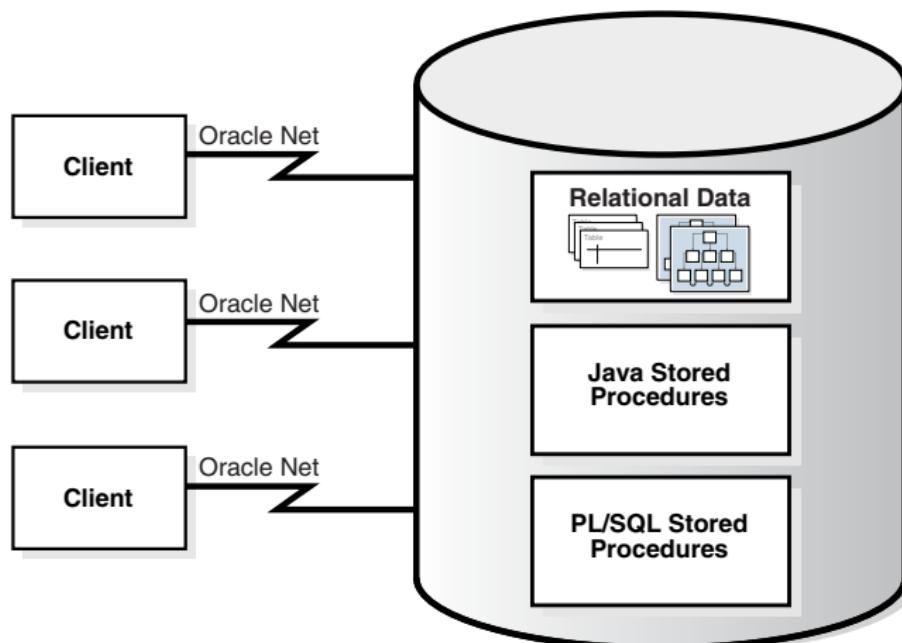
**Avantages** de l'architecture client-serveur :

- exécution des traitements au plus près des données,
- stockage et calculs supportés par le serveur,
- répartition du développement : côté client et côté serveur.

# Architecture client-serveur

Illustration issue de la documentation *Oracle*

**Figure 8–4 Two-Tier Client/Server Configuration**



# Acteurs interagissant avec le SGBD

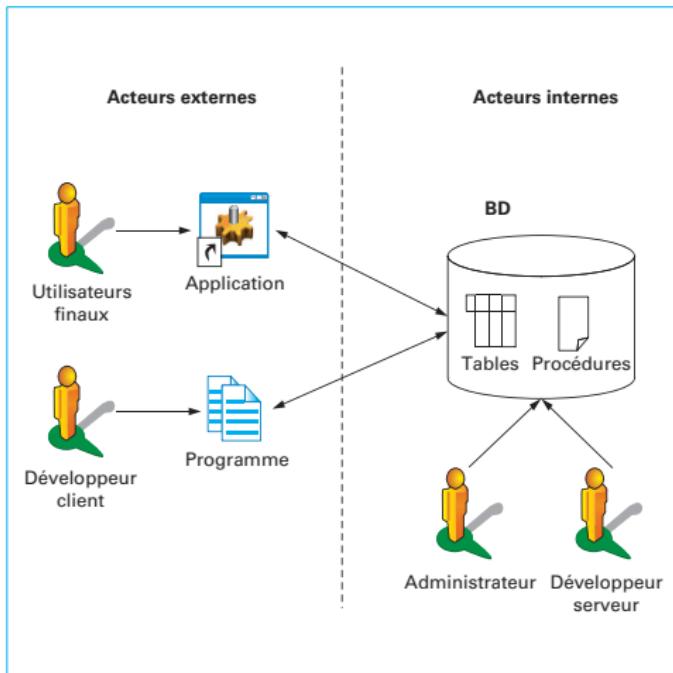


Figure 1 – Acteurs interagissant avec le SGBD

# Interrogation de la BD via une API

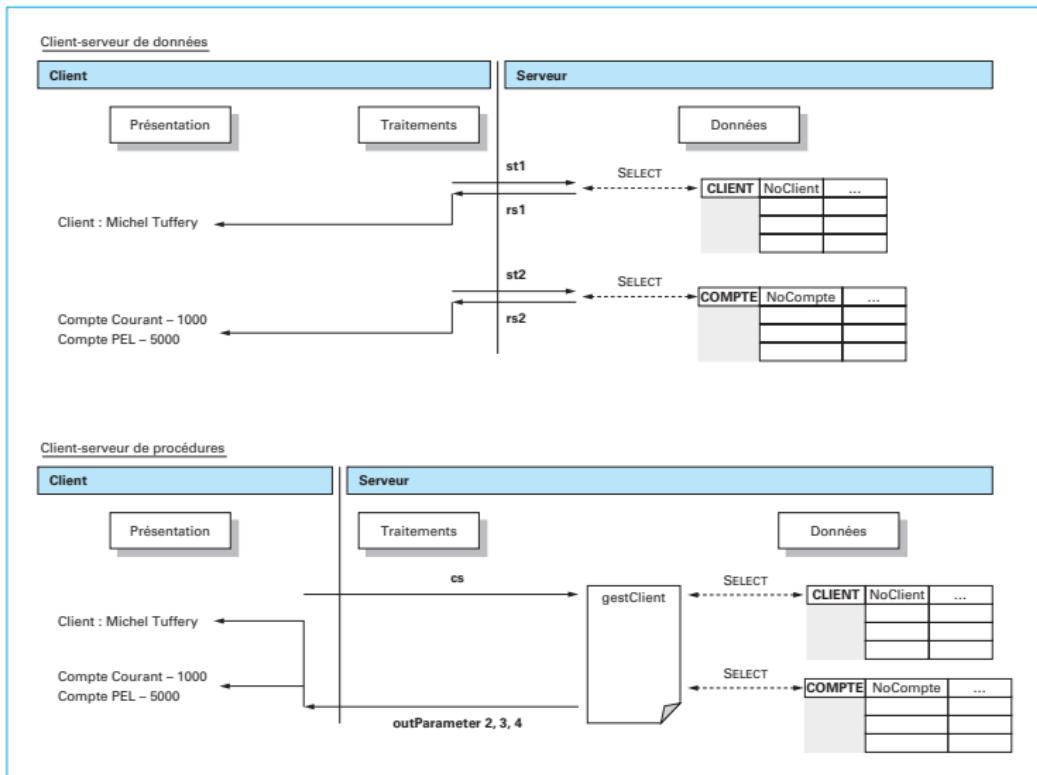


Figure 17 – Interrogation en client-serveur de données et de procédures

# Mise à jour de la BD via une API

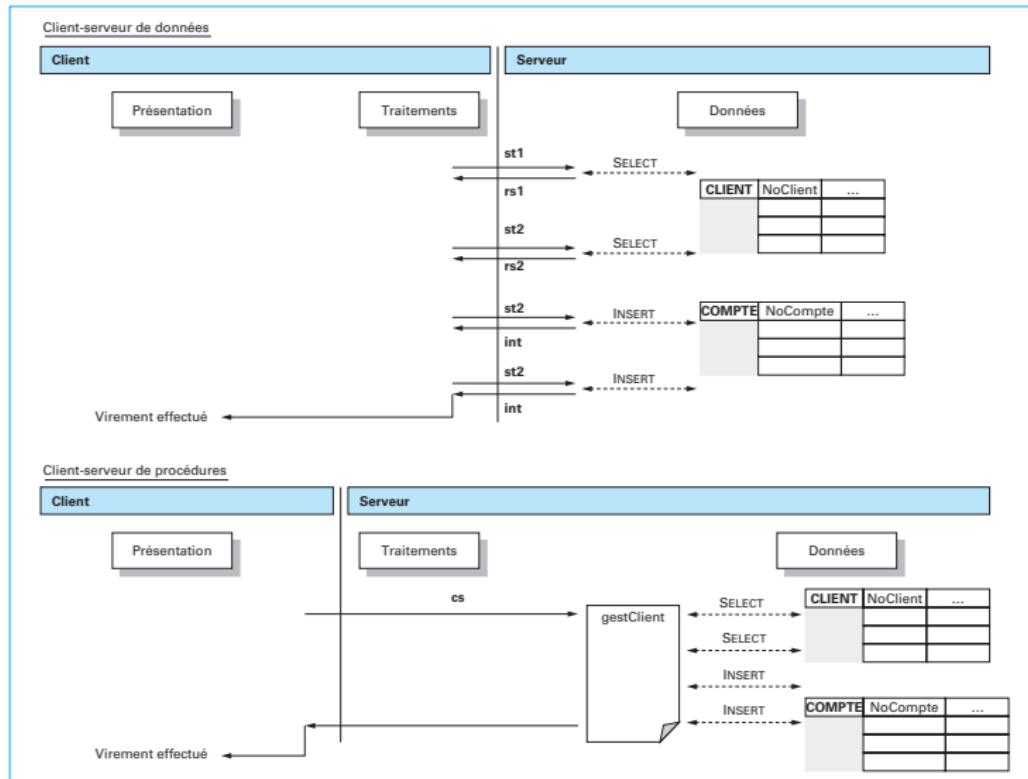


Figure 19 – Mise à jour transactionnelle en client-serveur de données et de procédures

# Bonnes pratiques de développement client-serveur

## ● Côté serveur

- Octroyer le moins de privilèges possible.

*Exemple : compte permettant la consultation d'une vue, avec quota 0 M.*

- Définir les seuls services nécessaires aux clients.

*Exemple : via des procédures stockées regroupées en paquetages.*

- Ne pas révéler le schéma pour éviter les attaques.

- Vérifier l'intégrité des paramètres :  $\Delta$  aux injections SQL.

*Exemple : longueur des paramètres de type chaîne, domaine des valeurs.*

# Bonnes pratiques de développement client-serveur

## ● Côté serveur

- Octroyer le moins de priviléges possible.  
*Exemple : compte permettant la consultation d'une vue, avec quota 0 M.*
- Définir les seuls services nécessaires aux clients.  
*Exemple : via des procédures stockées regroupées en paquetages.*
- Ne pas révéler le schéma pour éviter les attaques.
- Vérifier l'intégrité des paramètres :  $\Delta$  aux injections SQL.  
*Exemple : longueur des paramètres de type chaîne, domaine des valeurs.*

## ● Côté client

- Ne pas réinventer la roue, utiliser les services du serveur.
- Choisir judicieusement d'exécuter des traitements sur le serveur ou pas.
- Libérer les ressources dès que possible.  
*Exemple : fermer un curseur dès qu'il est parcouru.*

Cette liste n'est bien sûr pas exhaustive...

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

## 5 Perspectives en S3, LP AGBD et plus...

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

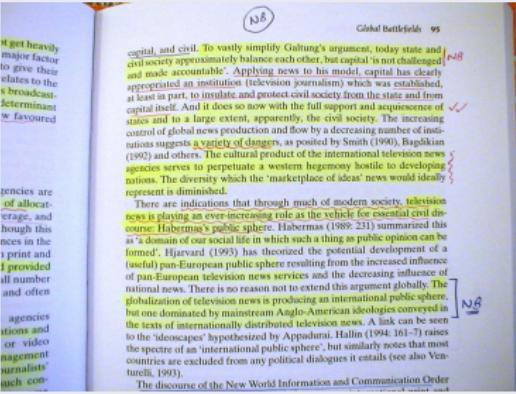
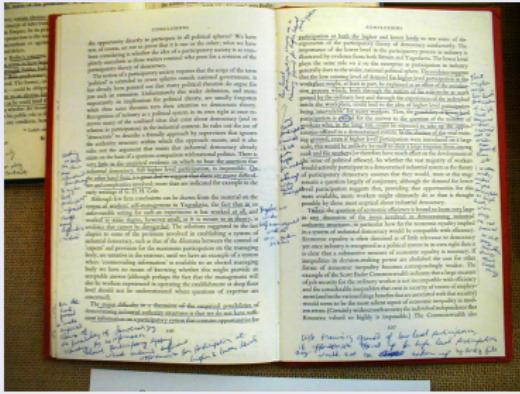
## 5 Perspectives en S3, LP AGBD et plus...

# Communication Java ←→ Oracle

Exemple issu de ma thèse (2008)

JDBC

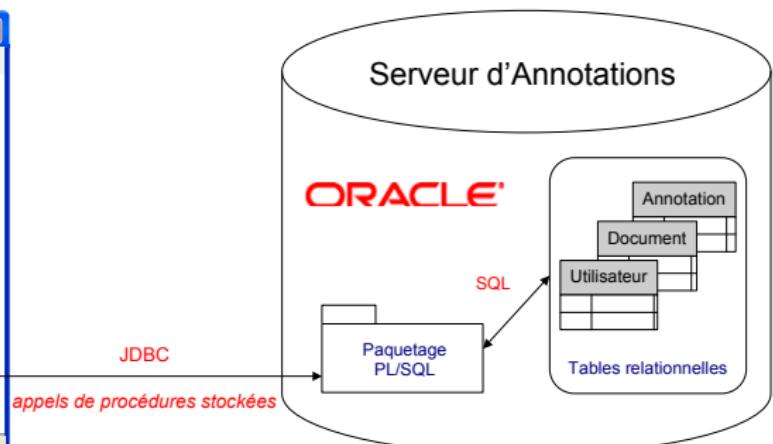
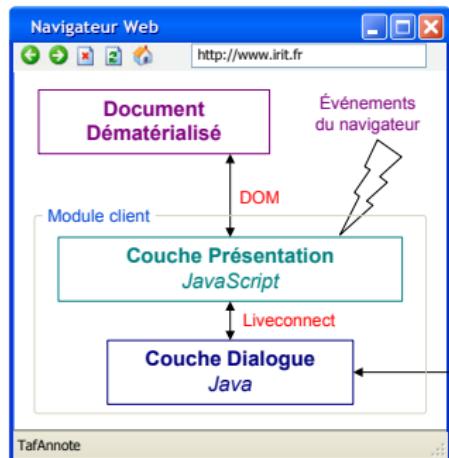
## Transposition de l'activité d'annotation papier → numérique ?



# Communication Java ↔ Oracle

Exemple issu de ma thèse (2008)

JDBC



Architecture du prototype *TafAnnote* reposant sur JDBC (Java DataBase Connectivity)

# Communication Java ↔ Oracle

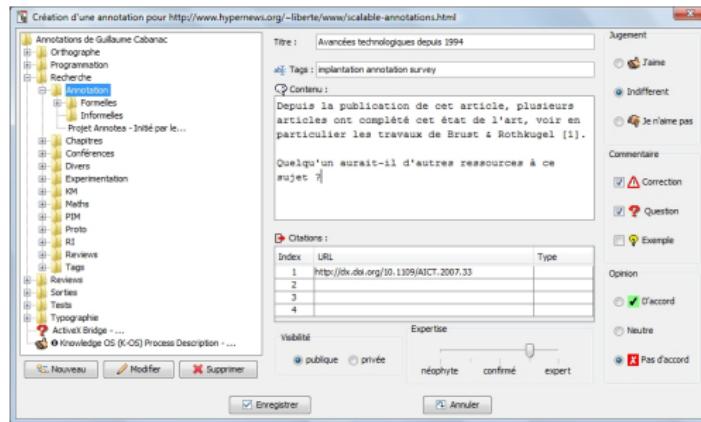
Exemple issu de ma thèse (2008)

JDBC

Captures d'écran du prototype *TafAnnote*.

L'interface graphique Java communique avec la base de données via des procédures stockées *Oracle*.

Le développeur Java appelle des procédures, sans se soucier de connaître le modèle de données sous-jacent.



# Communication Java ↔ Oracle

## Exemple issu de ma thèse (2008)

```
create package tafAnnote as
  /** Declaration du type ResultSet, renvoie par les procedures */
  type ResultSet is ref cursor ;

  /**
   * Retourne le numéro de version de TafAnnote
   */
  function getVersion return varchar ;

  /**
   * getOrCreateDoc - fonction auxiliaire
   * @param pIdUser L'identifiant de l'utilisateur
   * @param pUrl L'url du document annoté
   * @param pLastModif La date de dernière modification du document
   * @param pCree Vrai si le document d'url pUrl existait ou Faux s'il a fallu le créer
   */
  function getOrCreateDoc(pIdUser in number, pTitre in varchar, pUrl in varchar, pLastModif in out Date, pCree out boolean) return number ;

  /**
   * Identification et authentification
   * @throws NODATAFOUND si l'utilisateur n'existe pas
   */
  procedure authentifier(pLogin in varchar, pPasswd in varchar, pIdUser out number, pDescr out varchar, pEmail out varchar, pPageWeb out >
    varchar, pRacine out number) ;

  /**
   * Crée un utilisateur et un espace vide pour ses annotations
   */
  procedure creerUtilisateur(pLogin in varchar, pPassw in varchar, pDescr in varchar, pEmail in varchar, pPageWeb in varchar, idUserGen out >
    number, idRepGen out number) ;

  /**
   * Recupere les annotations d'un document dont l'url est spécifiée
   */
  procedure getAnnotations(pIdUser in number, pTitre in varchar, pUrl in varchar, pLastModif in Date, cur out ResultSet) ;

  -- [...]
end tafAnnote ;
/
```

# Communication Java ↔ Oracle

Exemple issu de ma thèse (2008)

```
import java.sql.SQLException;
import java.util.Properties;

import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.pool.OracleDataSource;


 /**
 * Classe de connexion à une BD Oracle
 * @author <a href="mailto:cabanac@irit.fr">Guillaume Cabanac</a>
 * @since 4 VI 2005
 */

abstract public class OracleConnectionFactory
{
    /**
     * Établit la connexion à la base de données en utilisant un
     * <a href="http://java.developpez.com/faq/jdbc/?page=dataSources#implementationsDatasource">
     * DataSource</a> au lieu du DriverManager (recommandé pour JDBC 3.0)
     * @param host La base de données à laquelle on se connecte
     * @param port Le port de la base de données
     * @param sid Le sid de la base de données
     * @param user L'identifiant de l'utilisateur
     * @param pass Le mot de passe de l'utilisateur
     */
    public static OracleConnection getConnection(String host, int port, String sid, String user, String pass) throws SQLException
    {
        Properties props = new Properties();
        props.put("defaultRowPrefetch","100"); // default = 10, évite de le faire à chaque Statement

        OracleDataSource dataSource = new OracleDataSource();
        dataSource.setURL("jdbc:oracle:thin:@" + host + ":" + port + ":" + sid);
        dataSource.setUser(user);
        dataSource.setPassword(pass);
        dataSource.setConnectionProperties(props);
        return (OracleConnection) dataSource.getConnection();
    }
}
```

Classe Java de connexion au SGBD via JDBC

# Communication Java ↔ Oracle

Exemple issu de ma thèse (2008)

```
/**  
 * Vérifie que la version de TafAnnote (Java) correspond bien à ce qui renvoyé par la BD  
 */  
public static void checkTafAnnoteVersion() throws SQLException  
{  
    CallableStatement cs = cx.prepareCall("{? = call tafAnnote.getVersion()}") ;  
    cs.registerOutParameter(1, Types.VARCHAR) ;  
    cs.execute() ;  
    String dernièreVersion = cs.getString(1) ;  
    cs.close() ;  
    // Vérification du numéro de version  
    if(!dernièreVersion.equals(TafAnnote.VERSION))  
        rapporterException(  
            new Exception("Votre version de TafAnnote ("+TafAnnote.VERSION+") n'est pas à jour.\n"+  
                "Merci d'installer la dernière version ("+dernièreVersion+").\n"+  
                "Elle est accessible sur http://www.irit.fr/~Guillaume.Cabanac/TafAnnote")) ;  
}
```

Méthode appelant une fonction du paquetage tafAnnote renvoyant un varchar

# Communication Java ↔ Oracle

Exemple issu de ma thèse (2008)

```
/**  
 * Crée un nouvel utilisateur  
 * return l'utilisateur dont les champs idUser et racine sont remplis  
 */  
public User creerUtilisateur(String login, String passw, String descr, String email, String pageWeb)  
{  
    try  
    {  
        CallableStatement cs = getConnection().prepareCall("{call tafAnnote.creerUtilisateur(?,?,?,?,?,?)}") ;  
        cs.setString(1, tronquer(login, 50)) ;  
        cs.setString(2, tronquer(passw, 50)) ;  
        cs.setString(3, tronquer(descr, 200)) ;  
        cs.setString(4, tronquer(email, 50)) ;  
        cs.setString(5, tronquer(pageWeb, 50)) ;  
        cs.registerOutParameter(6, Types.INTEGER) ;  
        cs.registerOutParameter(7, Types.INTEGER) ;  
        cs.execute() ;  
        User u = new User(cs.getInt(6), login, descr, email, pageWeb, cs.getInt(7)) ;  
        cs.close() ;  
        return u ;  
    }  
    catch(SQLException se)  
    { rapporterException(se); return null ; }  
}
```

Méthode appelant une procédure du paquetage tafAnnote

# Communication Java ↔ Oracle

Exemple issu de ma thèse (2008)

```
/**  
 * Récupère les annotations d'une url donnée  
 * @param titre Titre de la page Web  
 * @param url L'url du document contenant les annotations  
 * @param strDateModif La représentation textuelle d'un objet Date JavaScript  
 * @return Les annotations associées à l'url  
 */  
public List<Annotation> getAnnotations(String titre, String url, String strDateModif)  
{  
    List<Annotation> lstAnn = new ArrayList<Annotation>() ;  
  
    try  
    {  
        CallableStatement cs = getConnection().prepareCall("{call tafAnnote.getAnnotations(?, ?, ?, ?, ?)}") ;  
        cs.setInt(1, user.idUser) ;  
        cs.setString(2, tronquer(titre, 200)) ;  
        cs.setString(3, tronquer(url, 1024)) ;  
        cs.setDate(4, new JavascriptDate(strDateModif)) ;  
        cs.registerOutParameter(5, OracleTypes.CURSOR);  
        cs.execute();  
  
        try  
        {  
            // cursor is closed (not assigned by PLSQL) if no annotation is available --> sqlexception  
            for(ResultSet cursor = (ResultSet)cs.getObject(5) ; cursor.next() ; )  
                lstAnn.add(new Annotation(cursor)) ;  
        }  
        catch(SQLException e)  
        {  
            // there is no annotation on this document --> cursor == null  
        }  
        cs.close() ;  
        return lstAnn ;  
    }  
    catch(SQLException se)  
    {rapporterException(se); return null ;}  
}
```

# Communication Java ↔ Oracle

Un exemple complet développé dans les *Techniques de l'ingénieur*

## Architecture client-serveur : modes d'accès aux bases de données

par **Guillaume CABANAC**

*Maître de conférences à l'Université Toulouse 3*

**Olivier TESTE**

*Maître de conférences à l'Université Toulouse 3*

et **Michel TUFFERY**

*Maître de conférences à l'Université Toulouse 2*

<b>1. Contexte général de l'interaction avec une BD .....</b>	<b>H 3 865 - 2</b>
1.1 Acteurs .....	— 2
1.2 Tâches .....	— 2
<b>2. Accès interne au SGBD .....</b>	<b>— 2</b>
2.1 Architecture de 1 <sup>re</sup> génération .....	— 2
2.2 Architecture de 2 <sup>e</sup> génération .....	— 5
<b>3. Accès externe au SGBD .....</b>	<b>— 7</b>
3.1 Architectures d'une application .....	— 7
3.2 Principes du client-serveur .....	— 8
3.3 Interrogation de données .....	— 8
3.4 Mise à jour transactionnelle de données .....	— 13
3.5 Synthèse .....	— 19
<b>4. Conclusion .....</b>	<b>— 20</b>
<b>Pour en savoir plus .....</b>	<b>Doc. H 3 865</b>

Document téléchargeable depuis l'université <http://bit.ly/ti-h3865>

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

## 5 Perspectives en S3, LP AGBD et plus...

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

## 5 Perspectives en S3, LP AGBD et plus...

# Communication PHP ↔ Oracle

Exemple d'un site web

```
13 <body>
14   <div id="corps">
15     <!-- L'en-tête -->
16     <div id="en_tete">
17
18   </div>
19
20   <?php
21     require("infos.php");
22     // Connexion à la base
23     $conn = oci_connect($user, $pass, $server);
24
25     // On récupère le nom de la personne
26     $id = $_POST['id'];
27     $selectNom = oci_parse($conn, "SELECT * From author where idauthor = $id");
28     oci_define_by_name($selectNom, 'NAME', $nom);
29     oci_execute($selectNom, OCI_DEFAULT);
30     oci_fetch($selectNom);
31     oci_free_statement($selectNom);
32     oci_close($conn);
33
34   ?>
35   <h1>Évaluation de mesures de similarités entre chercheurs</h1>
36   <?php //echo preg_replace('/\d/','', $nom);?>
37   <form method="post" action="eval2.php" name="formulaire" id="formulaire">
38     <fieldset>
39       <legend>Informations facultatives (1/2)</legend>
40
41       <table class="tabIndex">
42         <tr>
43           <td><label for="age">Âge</label></td>
44           <td><input id="age" type="text" name="age" style="width: 30px;" /></td>
```

Connexion à Oracle depuis PHP via Oracle Call Interface et exécution d'une requête

# Communication PHP ↔ Oracle

## Exemple d'un site web

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
3 <head>
4   <title>Evaluation</title>
5   <meta http-equiv="Content-Type" content="text/html" />
6 </head>
7 <?php
8   $letters = $_GET['letters'];
9   require("infos.php");
10  echo $letters;
11  // Connexion à la base
12  $conn = oci_connect($user, $pass, $server, 'UTF8');
13
14  $select = oci_parse($conn, "SELECT * FROM author WHERE lower(name) like lower('%" . $letters . "%') and name not like '%00%'");
15  oci_define_by_name($select, 'NAME', $nom);
16  oci_define_by_name($select, 'IDAUTHOR', $id);
17
18  oci_execute($select, OCI_DEFAULT);
19
20  while (oci_fetch_array($select, OCI_ASSOC))
21  {
22    echo $id."###".$nom."!";
23  }
24  oci_free_statement($select);
25  oci_close($conn);
26 ?
27 </html>
```

Parcours d'un curseur Oracle depuis PHP via Oracle Call Interface

# Communication PHP $\longleftrightarrow$ Oracle

## Exemple d'un site web

```

43 echo '<fieldset>
44   <legend>Select the keywords to use</legend>
45   <p>Click on keywords for which you would like recommendations</p>';
46 // Sélectionne les 20 mots les plus utilisés par le chercheur dans le titre de leur publications
47 $selectMots = oci_parse($conn, "select label, idf, term.idterm, count(*) nb, avg(year) moy from term, indextext, write, doc
48   where indextext.idterm = term.idterm
49     and write.iddoc = indextext.iddoc
50     and idauthor = $id
51   group by label, idf, term.idterm
52   order by count(*) desc");
53
54 oci_define_by_name($selectMots, 'LABEL', $label);
55 oci_define_by_name($selectMots, 'IDTERM', $idterm);
56 oci_define_by_name($selectMots, 'IDF', $idf);
57 oci_define_by_name($selectMots, 'NB', $tf);
58 oci_define_by_name($selectMots, 'MOY', $moyAn);
59 oci_execute($selectMots, OCI_DEFAULT);
60 $nb = 0;
61
62 for ($nb = 0; $nb < 20; $nb++)
63 {
64   oci_fetch($selectMots) ;
65   // Affiche le mot-clé
66   $taille = 10 + ($tf * $idf);
67   $couleur = (2010 - $moyAn) * 30;
68   echo '<a class="motcle" style="font-size:'.$taille.'px;color:rgb(0,0,$couleur);" onclick="selectTerm(this,'.$nb.');">' . $label . '</a>
69   <input type="checkbox" style="visibility:hidden;width:2px;" name="T'.$nb.'" id="'.$nb.'" />
70   <input type="hidden" id="T'.$nb.'" name="T'.$nb.'" value="'.$idterm.'" />';
71 }
72 oci_free_statement($selectMots);
73
74 oci_close($conn);
75 ?>
76 <div id="bloc"></div>
77 <br/>
```

Parcours d'un curseur Oracle depuis PHP via Oracle Call Interface

# Communication PHP $\longleftrightarrow$ Oracle

## Exemple d'un site web

```
13 <body>
14   <div id="corps">
15     <!-- L'en-tête -->
16     <div id="en_tete">
17
18   </div>
19
20   <?php
21     require("infos.php");
22     // Connexion à la base
23     $conn = oci_connect($user, $pass, $server);
24
25     // On récupère le nom de la personne
26     $id = $_POST['id'];
27     $selectNom = oci_parse($conn, "SELECT * from author where idauthor = $id");
28     oci_define_by_name($selectNom, 'NAME', $nom);
29     oci_execute($selectNom, OCI_DEFAULT);
30     oci_fetch($selectNom);
31     oci_free_statement($selectNom);
32
33
34     // Enregistre les informations facultatives
35     if (!isset($_POST['age']))
36       $age = "0";
37     else
38       $age = $_POST['age'];
39     if (!isset($_POST['nbAnnes']))
40       $nbAn = '';
41     else
42       $nbAn = $_POST['nbAnnes'];
43
44     $update = oci_parse($conn, "UPDATE eval_author_bonus SET age = '$age', nbAnnees = '$nbAn' where idauthor = '$id'");
45     oci_execute($update, OCI_COMMIT_ON_SUCCESS);
46     oci_free_statement($update);
47
48   ?>
49   <h1>Évaluation de mesures de similarités entre chercheurs</h1>
50   <form method="post" action="eval3.php" name="formulaire" id="formulaire" onSubmit="return valider_formulaire()">
```

Mise à jour de données de la BD depuis PHP via Oracle Call Interface

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

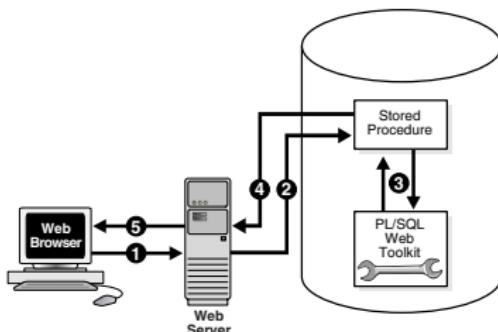
- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

## 5 Perspectives en S3, LP AGBD et plus...

# Solution 100% intégrée : Oracle Web Toolkit

Figure 11-1 PL/SQL Web Application



Principe du *PL/SQL Web Toolkit* :

- le SGBD est également **serveur Web**,
- chaque **page Web** à afficher est une **procédure stockée**,
- le code **HTML** est généré par les procédures avec `htp.print` et autres.

Avantages :

- un seul système à configurer et à maintenir,
- le langage principal de développement est PL/SQL.

# Solution 100% intégrée : Oracle Web Toolkit

Exemple 1 : page web pour l'association Inforsid

The screenshot shows a web application interface for the Inforsid association. At the top, there's a navigation bar with links to Self, DBLP, BP, AMEX, LinkedIn, Twitter, StatCounter, Annuaire, EDT, Oracle, Javadoc, and HtmlUnit 2.11 API. Below the navigation bar, the title "Anthologie des congrès Inforsid" is displayed in large green font, followed by the subtitle "INformatique des ORganisations et Systèmes d'Information et de Décision". A subtext below the subtitle states: "Ce congrès a rassemblé sur 31 ans 1387 chercheurs qui ont présenté 836 articles." The main content area is divided into several sections:

- Éditions du congrès**: A timeline from 1983 to 2013, with years 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, and 2013.
- Chercheurs**: A search bar with placeholder "exemple : Flory" and a "Rechercher" button.
- Suggestions de chercheurs**: A section showing "1 Suggestion de chercheurs pour le CP" and "Expérimental".
- Où le congrès Inforsid a-t-il eu lieu ?**: A map of Europe and North Africa showing the locations of Inforsid conferences. Labeled locations include:
  - Lille (1994, 2013)
  - Nancy (1989, 2002)
  - Paris (1981, 2015)
  - Perpignan (2007)
  - Tamammet (Tunisie) (2006)

# Solution 100% intégrée : Oracle Web Toolkit

Exemple 1 : page web pour l'association Inforsid

Chaque page web est une procédure stockée avec éventuellement des paramètres.

```

create or replace procedure accueil as
    anneeMin    inforsid_congres.annee%type;
    anneeMax    inforsid_congres.annee%type;
    lieu        inforsid_ville.nomVille%type;
    nbAns      number(5);
    numArticle  number(5);
    numPersonne number(5);
    i           number(5);
cursor annees is select annee
    from inforsid_congres
    order by annee;
cursor journaux is  select shorttitle, titre, url
    from corrNames c, nomJournaux n
    where title = abbr
    order by 1;
cursor conf is   select *
    from conferences c
    order by nb desc;
cursor jour is   select titre, url, nb
    from journaux j, nomJournaux n
    where j.abbr = n.abbr(+)
    order by 3 desc, 1;
begin
    -- récupération des infos
    select min(annee), max(annee) into anneeMin, anneeMax
        from inforsid_congres;
    select count(*) into numPersonne
        from Inforsid_Personne;
    select count(*) into numArticle
        from Inforsid_Article;
    nbAns := anneeMax-anneeMin+1;
    -- affichage des infos
    header('Inforsid' || anneeMin || '-' || anneeMax,
          'Anthologie des ' +
          'Site officiel de l''association' +
          ' congrès Inforsid' +
          '' +
          '<a href="http://inforsid.fr"' +
          'title="Site officiel de l''association">' +
          'INformatique des ORganisations et Systèmes d''Information et de Décision' +
          '');
    );
    HTPL.print('
        <p class="text-right">
            <i>Ce congrès a rassemblé sur '||nbAns||' ans <b>'||numPersonne||' chercheurs</b> qui ont présenté <b>'||numArticle||' articles</b>.</i>
        </p>
    ');

```

# Solution 100% intégrée : Oracle Web Toolkit

Exemple 2 : page web pour le projet tuteuré Series-0-Rama

The screenshot shows a web browser window with the title "Series-O-Rama – Search and Browse 126 TV Series". The URL in the address bar is <http://www.irit.fr/~Guillaume.Cabanac/Series-O-Rama>. The page has a header with links to various services like Self, DBLP, BP, AMEX, LinkedIn, Twitter, StatCounter, Annuaire, EDT, Oracle, Javadoc, HtmlUnit 2.11 API, gnuplot 4.7, OPEX, SCIM, JASIST, JOI, and more.

The main content area is titled "Series-O-Rama — Search and Browse 126 TV Series". It features two input fields: "Search" containing "plane crash island" and "Discover" containing "Tell us what TV Series you liked or dislike and get recommendations!".

The "Browse" section lists 126 TV series in a grid format. The columns are labeled with the number of seasons and the names of the series. The series are listed as follows:

Seasons	Community	Flight of the Conchords	John from Cincinnati	Nip/Tuck	Smallville	The Riches
24	Criminal Minds	Friday Night Lights	Knight Rider	One Tree Hill	Sons of Anarchy	The Sarah Connor Chronicles
90210	Cupid	Friends	Kyle XY	Or	Southpark	The Shield
Alias	Daybreak	Fringe	Legend of the Seeker	Painkiller Jane	Spaced	The Sopranos
Angel	Demons	Futurama	Leverage	Primal	Stargate Atlantis	The Tudors
Battlestar Galactica	Desperate Housewives	Gary Unmarried	Lie to Me	Prison Break	Stargate SG1	The Vampire Diaries
Better Off Ted	Dexter	Ghost Whisperer	Life	Private Practice	Stargate Universe	The Wire
Bionic Woman	Dirt	Gossip Girl	Lost	Psych	Supernatural	Torchwood
Blade	Dirty Sexy Money	Greek	Madmen	Pushing Daisies	Swingtown	Traveler
Blood Ties	Doctor Who	Grey's Anatomy	Masters of Sci-Fi	Raines	The 4400	Tru Calling
Bones	Dollhouse	Heroes	Medium	Reaper	The Big Bang Theory	True Blood
Breaking Bad	Eleventh Hour	House	Melrose Place	Robin Hood	The Black Donnellys	Ugly Betty
Buffy	Entourage	How I Met Your Mother	Mental	Samantha Who?	The Kill Point	V
Burn Notice	Eureka	In Treatment	Merlin	Sanctuary	The Lost Room	Veronica Mars
Californication	Extras	Invasion	Moonlight	Scrub	The Mentalist	Weeds
Caprica	Fear Itself	Jake	My Name Is Earl	Sex And The City	The Nine	Whitechapel
Charmed	Flash Forward	Jekyll	Ncis	Six Feet Under	The O.C.	Women's Murder Club
Chuck	Flashpoint	Jericho	Ncis Los Angeles	Skins	The Pretender	X-Files

At the bottom right of the page, there is a copyright notice: "Guillaume Cabanac, 2010-2011". Below it are three small icons: W3C XHTML 2.0, W3C CSS, and W3C CSS3.

# Solution 100% intégrée : *Oracle Web Toolkit*

Exemple 2 : page web pour le projet tuteuré Series-0-Rama

```
create or replace procedure home as
  vTvSeriesCount number ;
  i number := 0 ;
begin
  select count(*) into vTvSeriesCount from Series ;
  headings('Series-0-Rama - Search and Browse '||vTvSeriesCount||' TV Series',
  'Series</span><span style="color:#663300">-0-</span><span style="color:#943634">Rama</span>&nb
  -- searching
  http.print(
  <form action="cabanac_series_o_rama.search" method="post">
  <fieldset style="height:1.4cm;"><legend>Search</legend>
    <input type="text" size="50" name="pQuery" onclick="this.value='';" style="margin-right:12px;" value="plane crash island" title='
    <input type="submit" title="Search TV Series related to these words" value="Search" />
  </fieldset>
  ) ;
  -- recommending
  http.print(
  <fieldset style="height:1.4cm; margin-left:1cm;"><legend>Discover</legend>
    Tell us what TV Series you liked or dislike and <a href="cabanac_series_o_rama.recommend">get recommendations</a>
  </fieldset>
  </form>
  ) ;
  -- browsing
  http.print('<fieldset style="margin-top:1cm;"><legend>Browse</legend>' ) ;
  http.print('<div><ul>' );
  for s in (select idS, name from series) loop
  if i > 0 and i mod 18 = 0 then
    http.print('</ul></div><div><ul>' );
  end if ;
  http.print('<li>'||linkTvSeries(s.idS, s.name)||'</li>' );
  i := i + 1 ;
  end loop ;
  http.print('</ul></div></fieldset>' );
  footer() ;
end home ;
```

# Solution 100% intégrée : *Oracle Web Toolkit*

Exemple 2 : page web pour le projet tuteuré Series-0-Rama

```
create or replace procedure search(pQuery in varchar) as
  curMatchQuery sys_refcursor ;
  vIdS  series.idS%type ;
  vName series.name%type ;
  vScore number ;
begin
  -- native dynamic SQL http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28370/dynamic.htm#i13057
  -- and term in (''||dbms_assert.enquote_literal(pQuery)||')
  open curMatchQuery for
    'select p.idS, name, sum(tf*idf) score --, count(*) termsMatched
     from posting p, dict d, series s
    where p.idT = d.idT
      and p.idS = s.idS
      and term in (''||replace(replace(pQuery, ' ', ','), ',', ' ', ','))||'||')
    group by p.idS, name
    order by 3 desc' ;

  headings('Series-0-Rama - TV series matching '||pQuery||'||') ;
  http.print('<table><tr><th>TV series</th><th>Score</th></tr>') ;
  loop fetch curMatchQuery into vIdS, vName, vScore ; exit when curMatchQuery%notfound ;
    http.print('<tr><td>'||linkTvSeries(vIdS, vname)||'</td><td align="right">'||trunc(100000*vScore)||'</td></tr>') ;
  end loop ;
  close curMatchQuery ;
  http.print('</table>') ;
  footer() ;
exception when others then
  http.print('Bad query syntax. Please fix it and retry.') ;
end search ;
```

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

## 5 Perspectives en S3, LP AGBD et plus...

# Instructions du Programme Pédagogique National en S2

<b>UE21</b>  <b>M2106</b>	<b>Informatique approfondie</b>  <b>Systèmes de gestion de bases de données</b>  <b>Programmation et administration des bases de données</b>	<b>Volume Horaire : 45</b>  <b>10h CM, 15h TD, 20h TP</b>  <b>Semestre 2</b>
<u>Objectifs du module :</u>		
<ul style="list-style-type: none"> <li>• Maîtriser les requêtes complexes en SQL (Structured Query Language) et savoir programmer côté SGBD (procédures stockées),</li> <li>• S'initier à l'administration et la sécurité des données</li> </ul>		
<u>Compétences visées :</u>		
<p><i>Compétences citées dans le Référentiel d'activités et de compétences pour les activités :</i></p> <ul style="list-style-type: none"> <li>• FA1-B : Conception technique d'une solution informatique</li> <li>• FA1-C : Réalisation d'une solution informatique</li> <li>• FA2-A : Administration de systèmes, de logiciels et de réseaux</li> </ul>		
<u>Prérequis :</u>		
<ul style="list-style-type: none"> <li>• M1103, M1104</li> </ul>		
<u>Contenus :</u>		
<ul style="list-style-type: none"> <li>• SQL et extension procédurale</li> <li>• Curseurs</li> <li>• Administration des SGBD : utilisateurs, rôle, droits, vues</li> <li>• SQL intégré dans un langage de programmation</li> </ul>		
<u>Modalités de mise en œuvre :</u>		
<ul style="list-style-type: none"> <li>• S'appuyer sur une extension procédurale de SQL (PL/SQL, etc.)</li> <li>• Faire le lien avec la programmation : boucles, conditions</li> <li>• Souligner le lien avec la modélisation : états, transitions, activités, etc.</li> </ul>		
<u>Prolongements possibles :</u>		
<ul style="list-style-type: none"> <li>• Contraintes dynamiques</li> </ul>		
<u>Mots clés :</u>		
Administration des données ; Curseurs ; Procédures stockées		

# Compléments

De nombreux **concepts** et une quantité de **subtilités** à propos de l'administration et de la programmation des bases de données **n'ont pas été couverts** dans ce module.



<http://hoopercharles.files.wordpress.com/2010/01/booklibraryworksmall.jpg>

Les enseignements de ce module trouvent des **prolongements** en 2<sup>e</sup> année de DUT, mais également en licence (dont la licence professionnelle), en master et en doctorat...

# SGBD au service du SI

## 1 Architecture client-serveur

## 2 Client lourd

- Communication Java  Oracle

## 3 Client léger

- Communication PHP  Oracle
- Solution 100% intégrée : *Oracle Web Toolkit*

## 4 Bilan du M2106

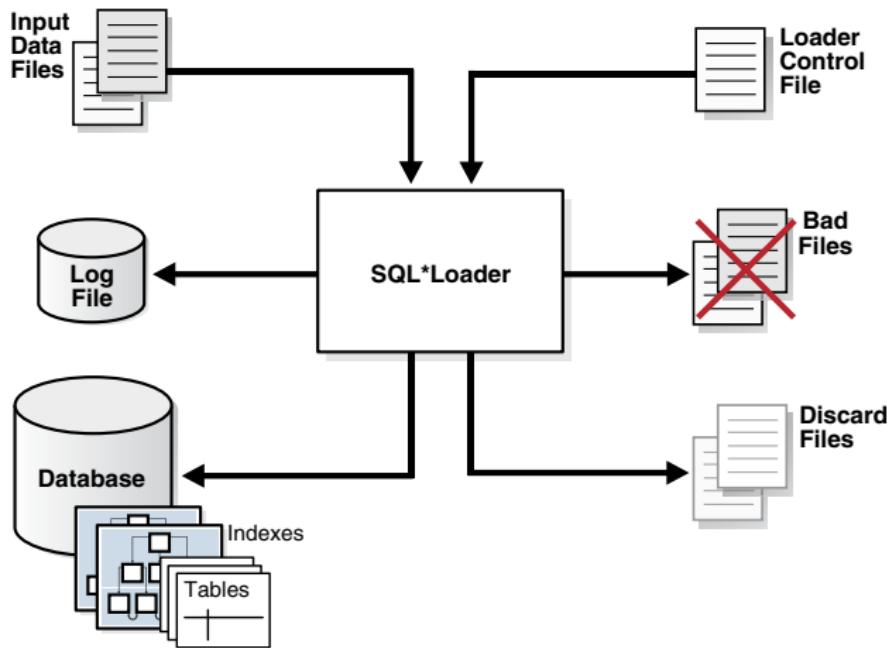
## 5 Perspectives en S3, LP AGBD et plus...

# Instructions du Programme Pédagogique National en S3

<b>UE31</b>	<b>Informatique avancée</b>	<b>Volume Horaire : 30</b>  <b>8h CM, 10h TD, 12h TP</b>
	<b>Systèmes de gestion de bases de données</b>	
<b>M3106C</b>	<b>Bases de données avancées</b>	<b>Semestre 3</b>
<u>Objectifs du module :</u>		
<ul style="list-style-type: none"> <li>Appréhender des notions avancées sur la qualité des schémas et les aspects systèmes</li> </ul>		
<u>Compétences visées :</u>		
<i>Compétences citées dans le Référentiel d'activités et de compétences pour les activités :</i>		
<ul style="list-style-type: none"> <li>FA1-C : Réalisation d'une solution informatique</li> <li>FA2-A : Administration de systèmes, de logiciels et de réseaux</li> <li>FA1-D : Tests de validation d'une solution informatique</li> </ul>		
<u>Prérequis :</u>		
<ul style="list-style-type: none"> <li>M2106</li> </ul>		
<u>Contenus :</u>		
<ul style="list-style-type: none"> <li>Qualité des schémas, problème de la redondance, formes normales</li> <li>Contraintes d'intégrité et règles de gestion, déclencheurs</li> <li>Présentation de l'architecture fonctionnelle d'un système de gestion de bases de données (SGBD)</li> <li>Transactions, atomicité et gestion de la concurrence d'accès</li> <li>Optimisation : index, requêtes et plan d'exécution</li> <li>Liens avec les langages de programmation</li> </ul>		
<u>Modalités de mise en œuvre :</u>		
<ul style="list-style-type: none"> <li>Les notions doivent être abordées sous un angle pratique, en particulier celles liées aux transactions et à l'optimisation</li> <li>Faire le lien avec les algorithmes sur les arbres et les tables de hachage (M3103 « Algorithmique avancée »)</li> </ul>		
<u>Prolongements possibles :</u>		
<ul style="list-style-type: none"> <li>Étude de la complexité</li> </ul>		
<u>Mots clés :</u>		
Normalisation ; Transactions ; Optimisation de requêtes		

# Alimentation de la BD avec *Oracle SQL\*Loader*

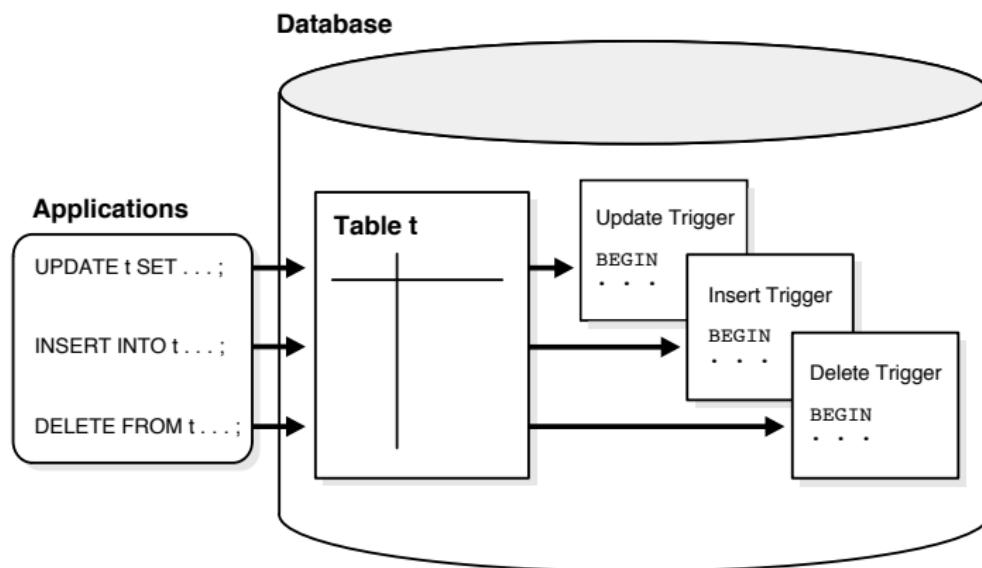
Figure 18-1 SQL\*Loader Session



# Programmation événementielle (1/2)

## Déclencheurs en BD active

Figure 22–1 Triggers

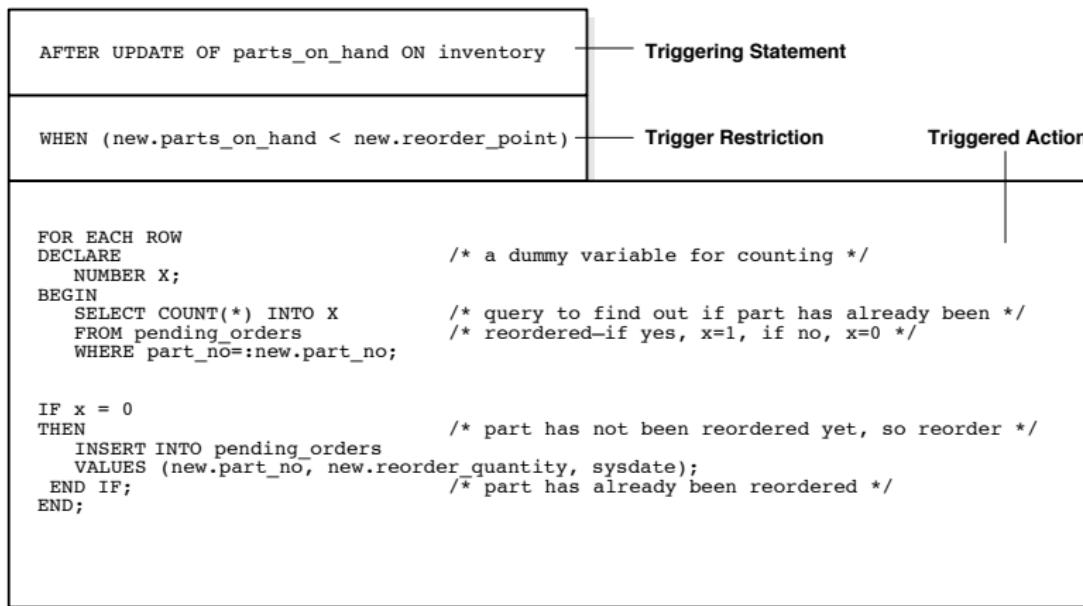


# Programmation événementielle (2/2)

## Déclencheurs en BD active

Figure 22–2 The REORDER Trigger

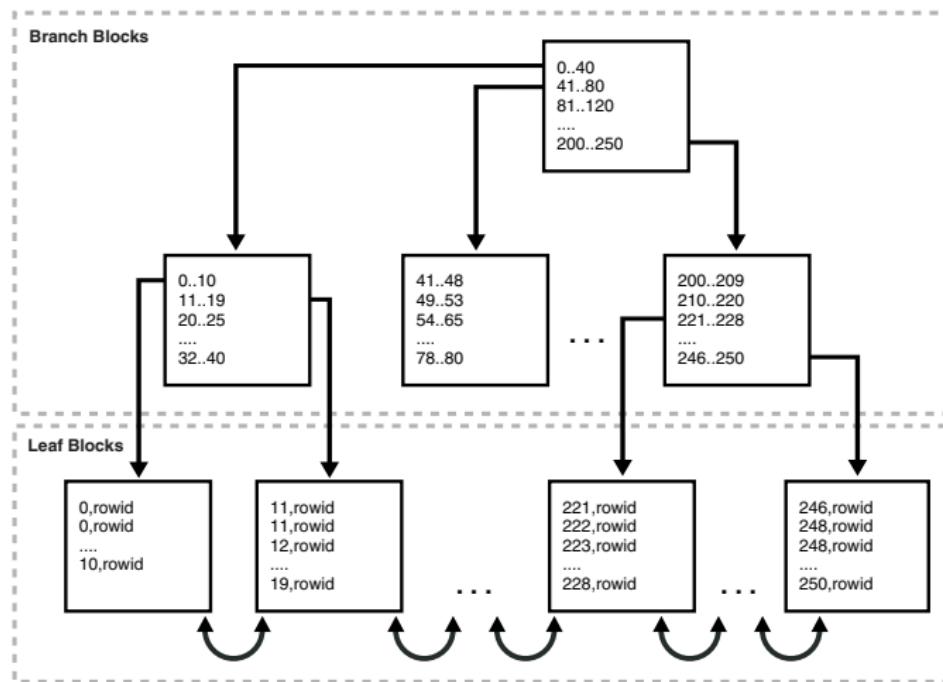
### REORDER Trigger



# Optimisation des performances (1/4)

Indexation des colonnes souvent consultées

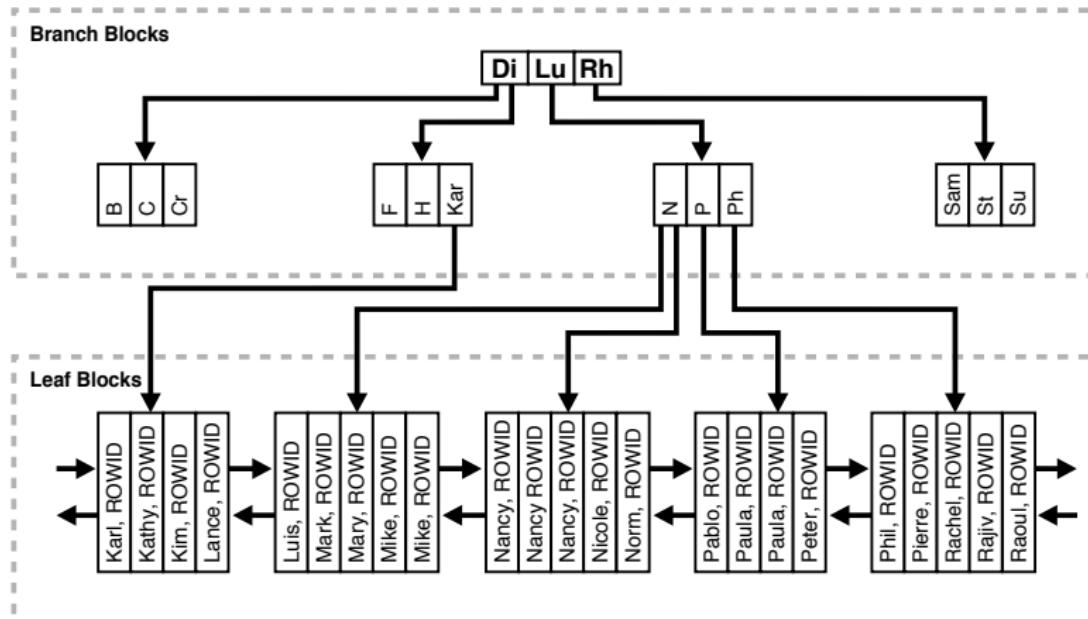
Figure 3–1 Internal Structure of a B-tree Index



# Optimisation des performances (2/4)

Indexation des colonnes souvent consultées

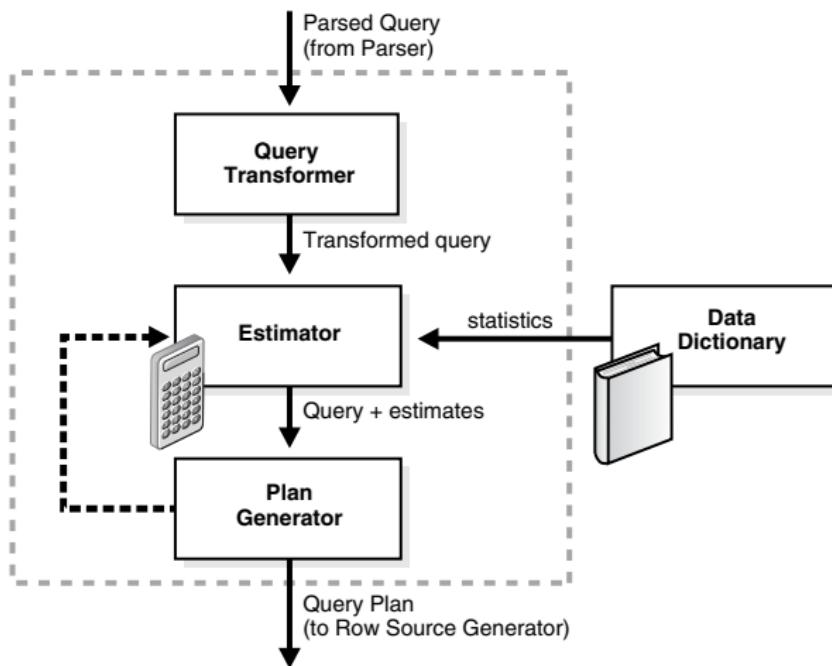
Figure 5-7 Internal Structure of a B-tree Index



# Optimisation des performances (3/4)

## Plan d'exécution

Figure 7–2 Optimizer Components



# Optimisation des performances (4/4)

## Parallélisation de requêtes

Figure 16–5 Serial Full Table Scan

Serial Process

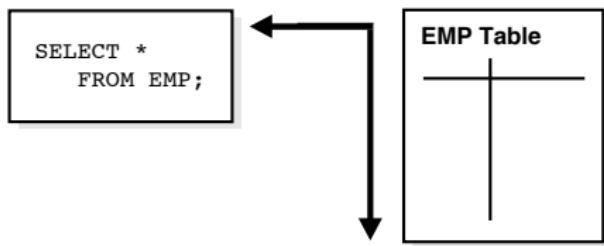
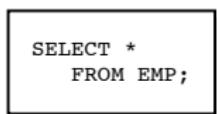


Figure 16–6 Parallel Full Table Scan

Parallel Execution Coordinator

Parallel Execution Server



EMP Table

# Accès concurrents et transactions

Figure 13–3 Two Transactions in a Deadlock

## Transaction 1 (T1)

```
UPDATE emp  
SET sal = sal*1.1  
WHERE empno = 1000;
```

## Time



A

## Transaction 2 (T2)

```
UPDATE emp  
SET mgr = 1342  
WHERE empno = 2000;
```



B

```
UPDATE emp  
SET sal = sal*1.1  
WHERE empno = 2000;
```

```
UPDATE emp  
SET mgr = 1342  
WHERE empno = 1000;
```

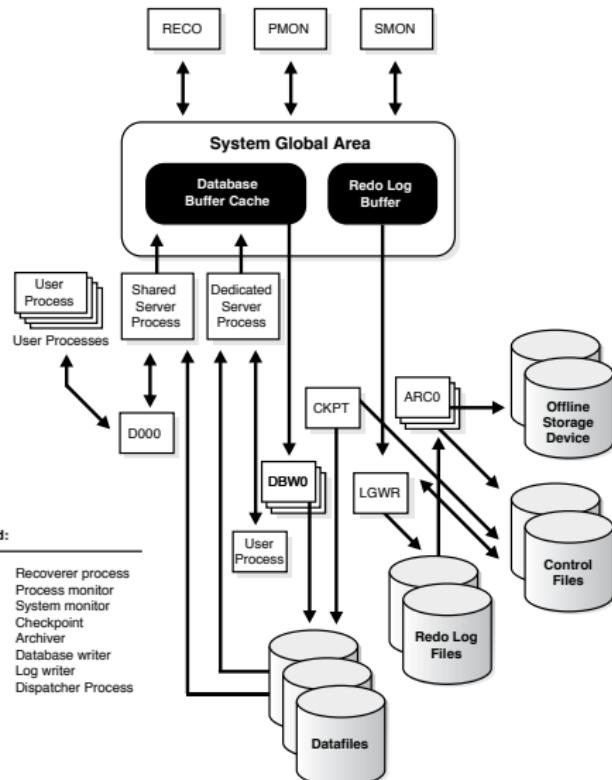


C

```
ORA-00060:  
deadlock detected while  
waiting for resource
```

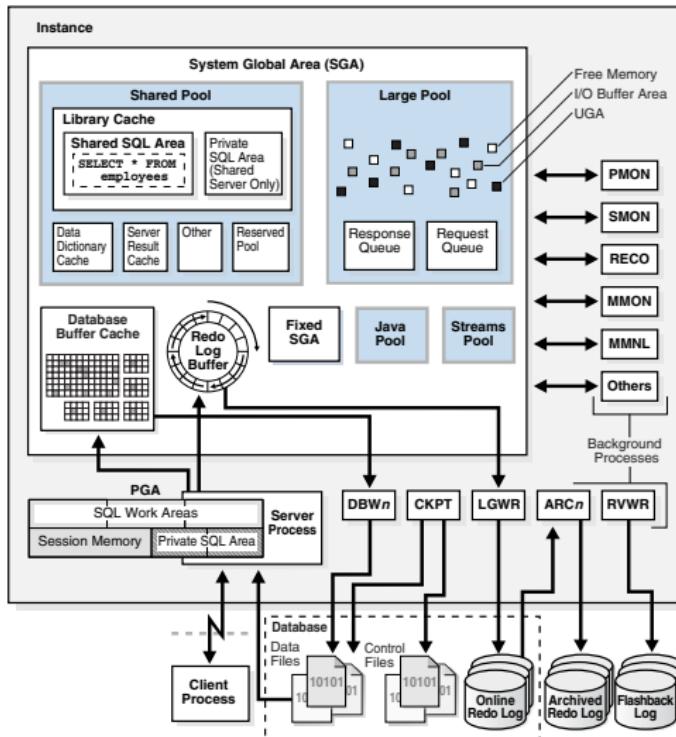
# Architecture du SGBD et processus (1/2)

Figure 9–2 Background Processes of a Multiple-Process Oracle Database Instance



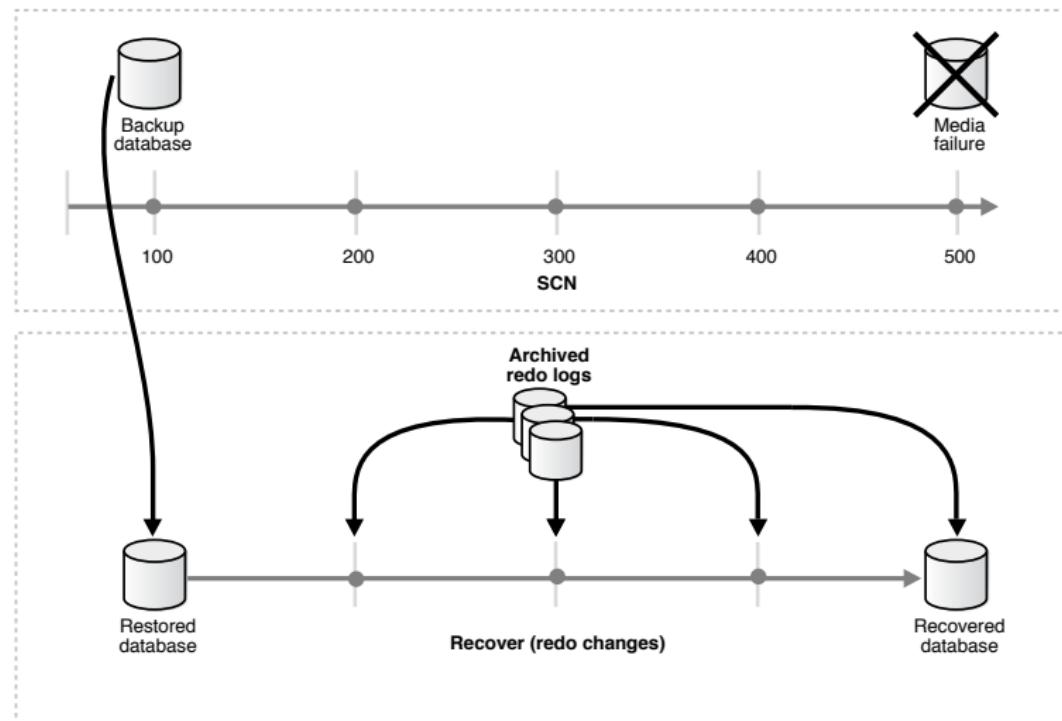
# Architecture du SGBD et processus (2/2)

Figure 1-1 Oracle Instance and Database



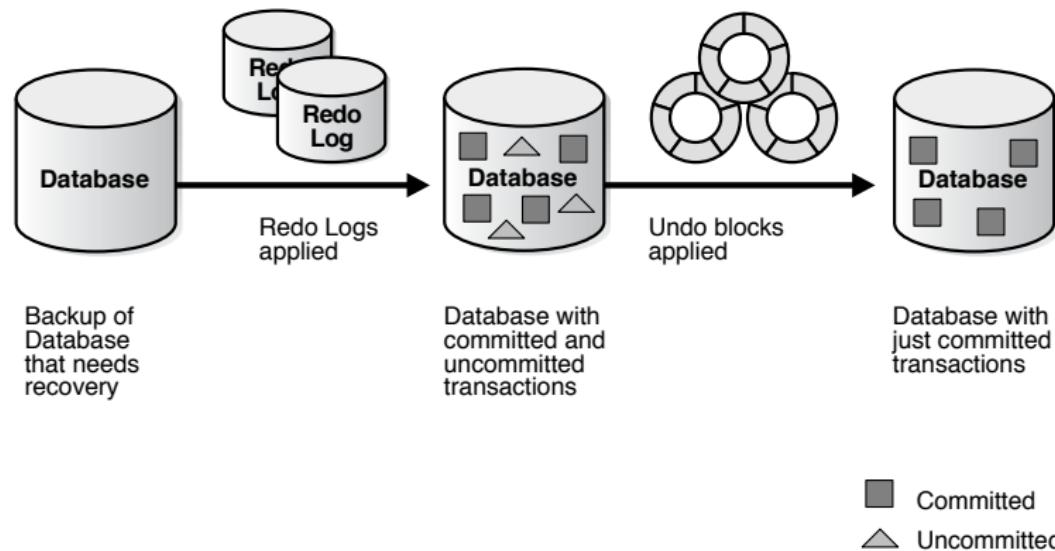
# Gestion des pannes et reprise sur arrêt (1/2)

Figure 15–2 Media Recovery



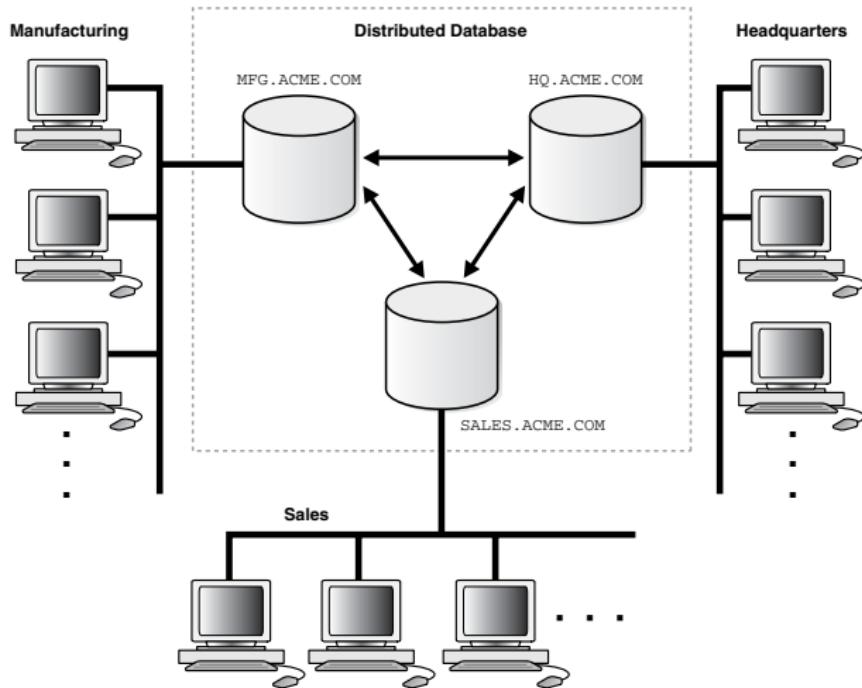
# Gestion des pannes et reprise sur arrêt (2/2)

Figure 12–2 Basic Recovery Steps: Rolling Forward and Rolling Back



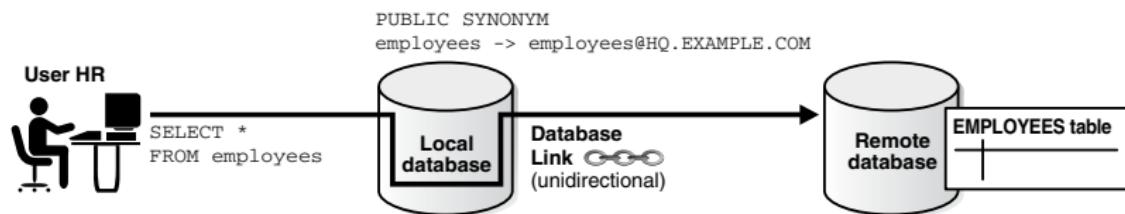
# Bases de données réparties (1/3)

Figure 29–1 Homogeneous Distributed Database



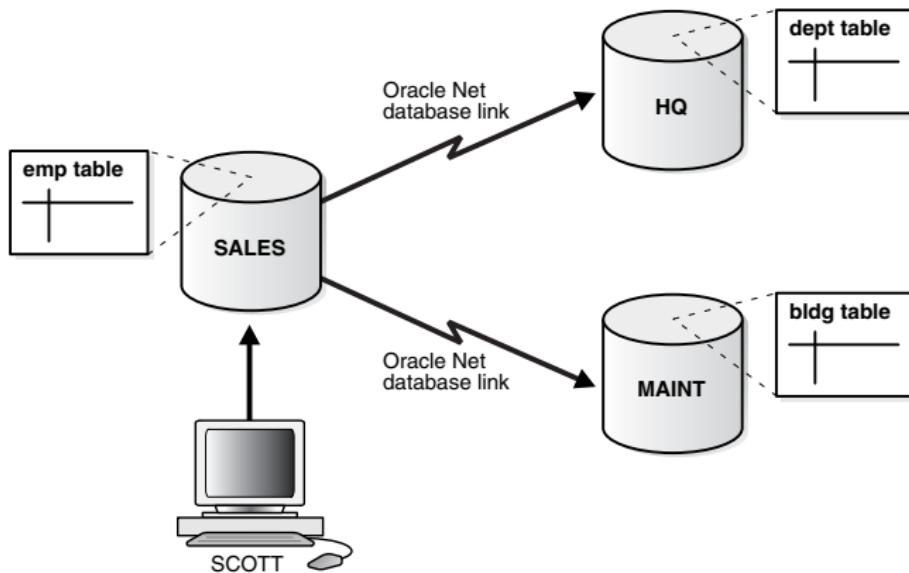
# Bases de données réparties (2/3)

Figure 17–6 Database Link



# Bases de données réparties (3/3)

Figure 32-1 Distributed System



# Sources

Les illustrations sont reproduites à partir des documents suivants :

- Oracle Database Documentation Library 10.2  
(<http://www.oracle.com/pls/db102/homepage>)
  - Oracle® Database : Application Developer's Guide - Fundamentals 10g Release 2 (10.2) B14251-01
- Oracle Database Documentation Library 11.1  
(<http://www.oracle.com/pls/db111/homepage>)
  - Oracle® Database : Administrator's Guide 11g Release 1 (11.1) B28310-04
  - Oracle® Database : Concepts 11g Release 1 (11.1) B28318-06
- Oracle Database Documentation Library 11.2  
(<http://www.oracle.com/pls/db112/homepage>)
  - Oracle® Database : Concepts 11g Release 2 (11.2) E40540-01
- Techniques de l'ingénieur, fascicule H-3865 (<http://bit.ly/ti-h3865>)
- Thèse de doctorat de G. Cabanac (2008)  
(<http://tel.archives-ouvertes.fr/tel-00344947>)