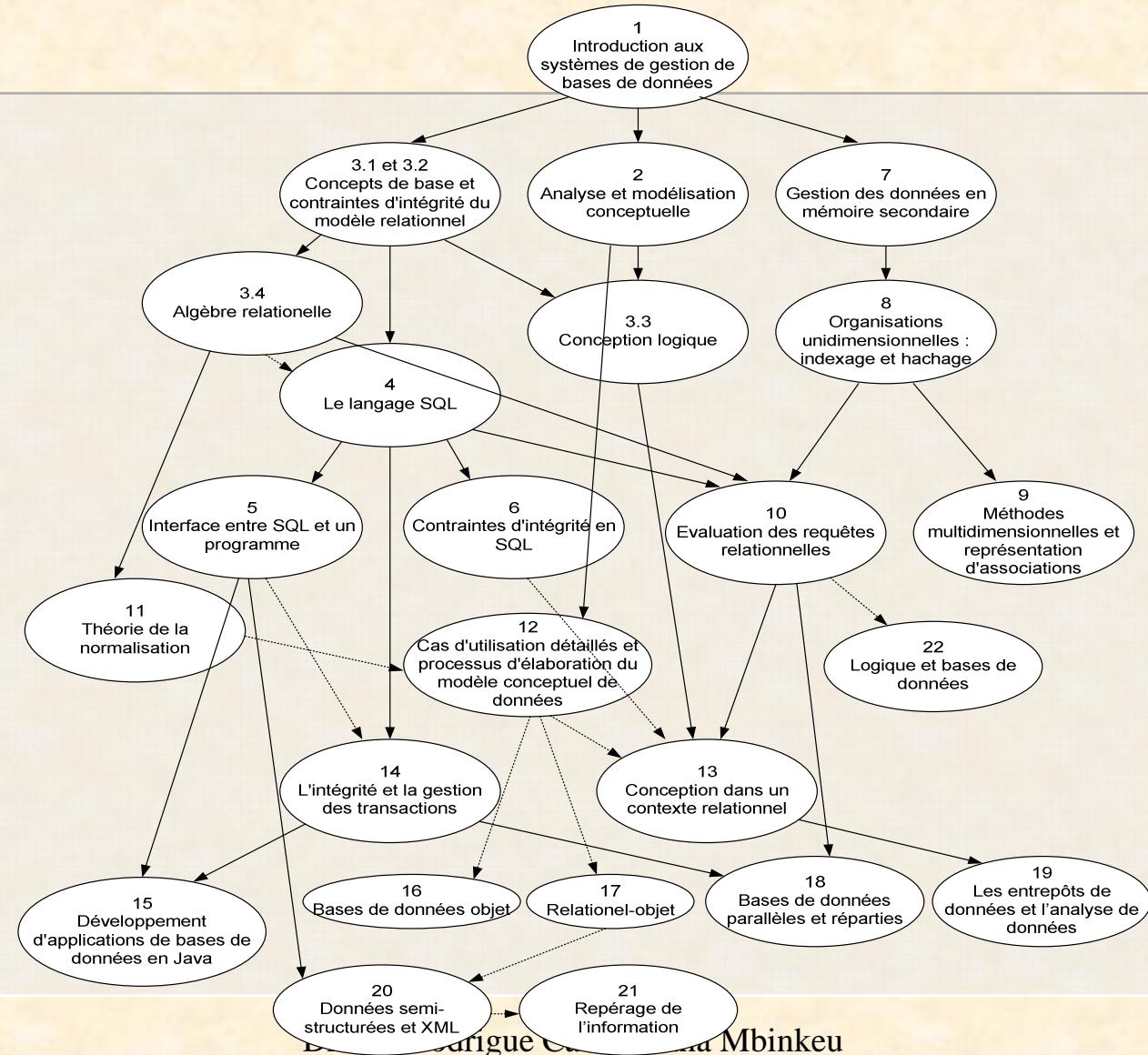


# Dépendances entre les chapitres



# 1 Introduction aux systèmes de gestion de bases de données

- BD : composante essentielle des systèmes informatiques modernes
  - systèmes d ’information de gestion
  - ingénierie
  - contrôle de processus
  - bibliothèques électroniques
  - ...
- Données persistantes

## **1.1 *Donnée et de type de donnée***

### ***Donnée (data)***

- Une *donnée* au sens informatique est une représentation d'un fait à l'aide d'un code binaire stocké dans la mémoire de l'ordinateur

### ***Type de données (data type)***

- Le *type de données* détermine la nature du codage utilisé pour représenter la donnée et les opérations applicables à la donnée.

# 1.2 Donnée et information

- Information : donnée interprétée
  - théorie de l'information
  - Shannon et Weaver (1959 )
  - mesure de la quantité d'information
    - message d'une *source* à un *destinataire*
  - diminution de l'incertitude au niveau du *destinataire*
  - information : relatif au *destinataire* du message
- Changement d'état du destinataire
- Souvent synonymes

# 1.3 Donnée simple et complexe

- *Donnée simple*
- *Donnée de type complexe, ou donnée complexe*
  - composée de données simples
- *Donnée multimédia (multimedia data)*
  - texte, image, son, vidéo

# 1.4 Différentes facettes du terme *base de données*

- *Base de données (BD)*
  - Au sens large, collection de données
- Différentes interprétations
  - ◆ fichier de texte
  - ◆ ensemble de fichiers de textes
  - ◆ un seul fichier d'enregistrements
  - ◆ ensemble de fichiers d'enregistrements
  - ◆ plusieurs sous bases de données indépendantes les unes des autres
  - ◆ *Toile* : base de données de nature hétérogène et répartie
  - ◆ BD logique = une collection de BD physiques

# 1.5 Système de gestion de base de données (SGBD)

- *Base de données (database)*
  - Sens plus strict, ensemble de données :
    1. fortement structurées
    2. persistantes
    3. structure définie dans un schéma
    4. gérées par *système de gestion de bases de données*
  - *Système de gestion de base de données (SGBD, database management system - DBMS)*
    - Logiciel spécialisé pour la gestion de base de données

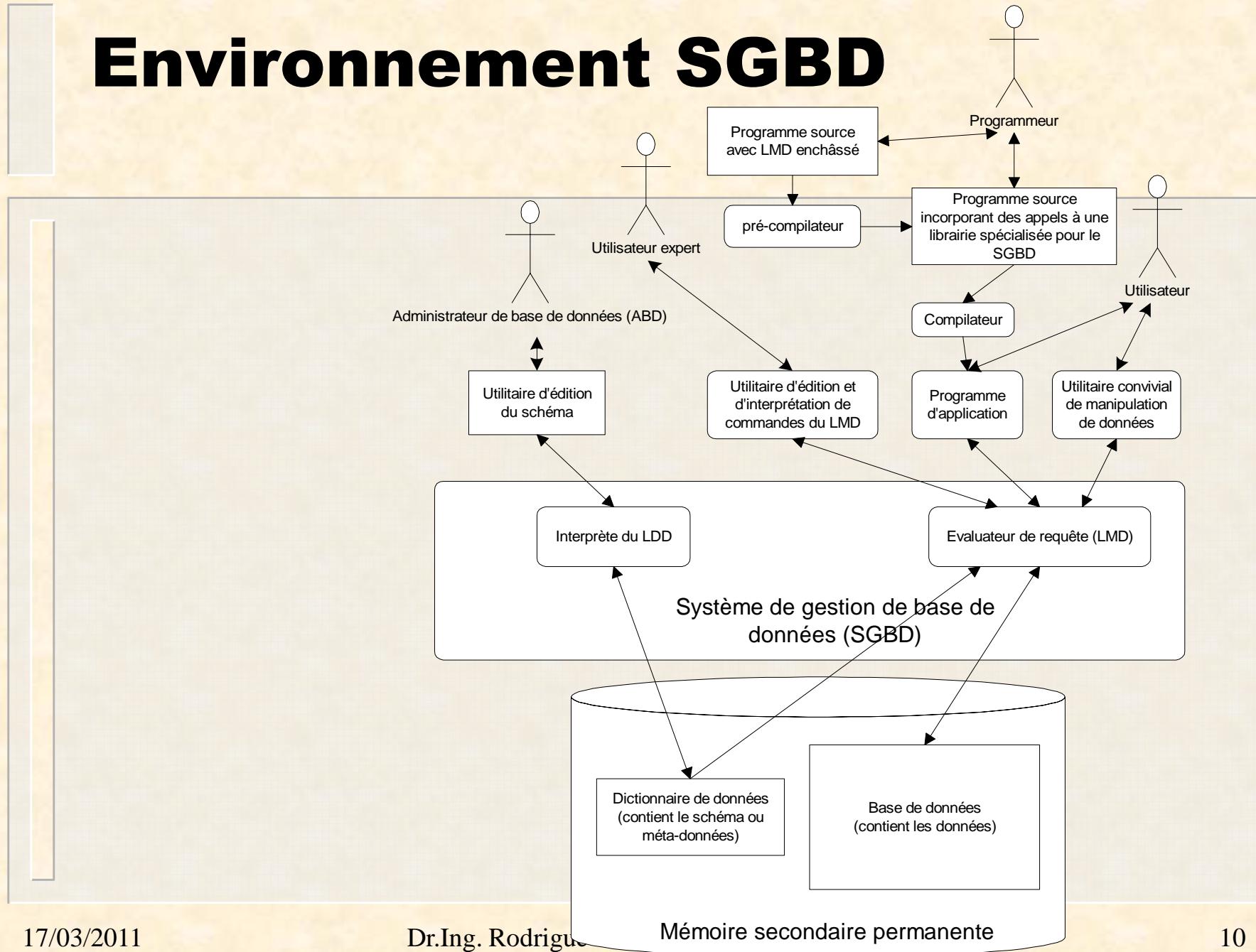
# Mémoire centrale/secondaire

- *Mémoire centrale, principale, vive, primaire, volatile (main memory, primary storage)*
  - directement accessible par le processeur central
  - contient les instructions et les données d'un programme en exécution
  - rapidité
  - non persistance
- *Mémoire secondaire (secondary storage), permanente, externe, stable, non volatile ou persistante*
  - indirectement via des opérations d'entrée/sortie
  - lenteur
  - persistance

# **Langage de définition des données et schéma**

- *Langage de définition des données (LDD, data definition language - DDL )*
  - permet la définition des types des données de la BD
- *Schéma de la base de données (database schema)*
  - structure des données définie à l'aide du LDD

# Environnement SGBD



# Langage de manipulation de données/ de requête

- *Manipulation des données (data manipulation)*
  - lire (électionner)
  - ajouter
  - supprimer
  - modifier
- *Langage de manipulation des données (LMD, data manipulation language - DML)*
  - *langage de requête (query language )*
  - ex: SQL, OQL

# Contexte d 'utilisation du LMD

- *Langage de manipulation des données encastré (embedded data manipulation language), langage hôte (host language)*
  - LMD encastré un langage de programmation *hôte*
    - (e.g. C, C++, ADA, JAVA,...)
  - syntaxe spéciale non reconnue par compilateur
    - étape de pré-compilation
- *Interface programmatique (API)*
  - e.g. ODBC, JDBC
  - LMD passé en paramètre
- Utilitaire GUI convivial

# Exécution des requêtes

- *Évaluateur de requête (query processor)*

- composante du SGBD responsable de l'exécution des requêtes
  - processus complexe

- *Requête (query)*

- opération de manipulation de données

# Définition des données

- *Administrateur de la base de données (ABD, database administrator - DBA)*
  - responsable de la gestion d'une base de données
  - maintien du schéma de la base de données
  - utilitaire d 'édition du schéma
- *Méta-données (metadata)*
  - le schéma sous forme de données
- *Dictionnaire de données (métabase, data dictionary)*
  - BD, gérée par le SGBD, qui contient les méta-données

# 1.6 Principales architectures des BD

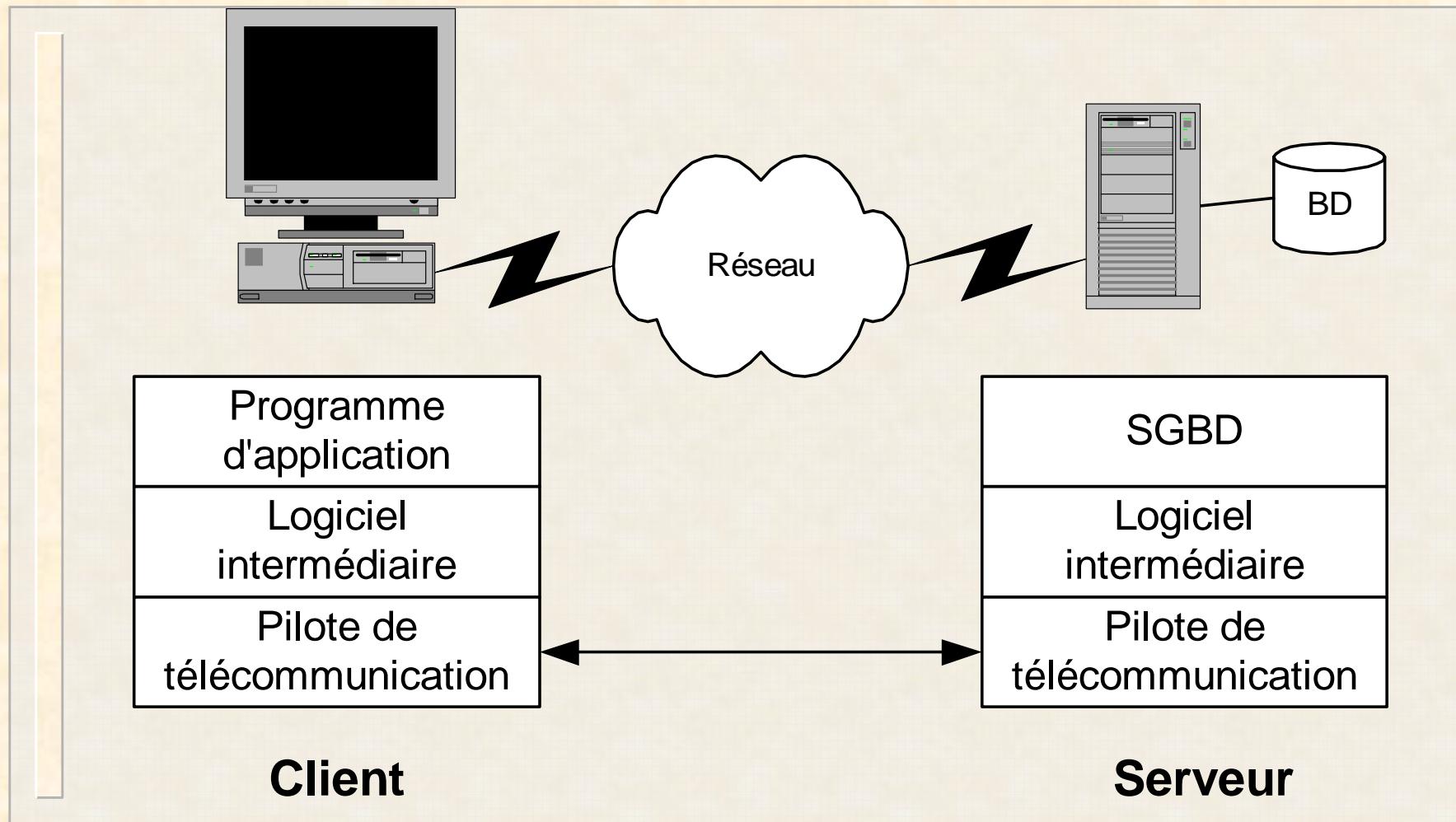
## ■ *Architecture centralisée*

- programme d'application et SGBD sur même machine (même site)
- premiers systèmes

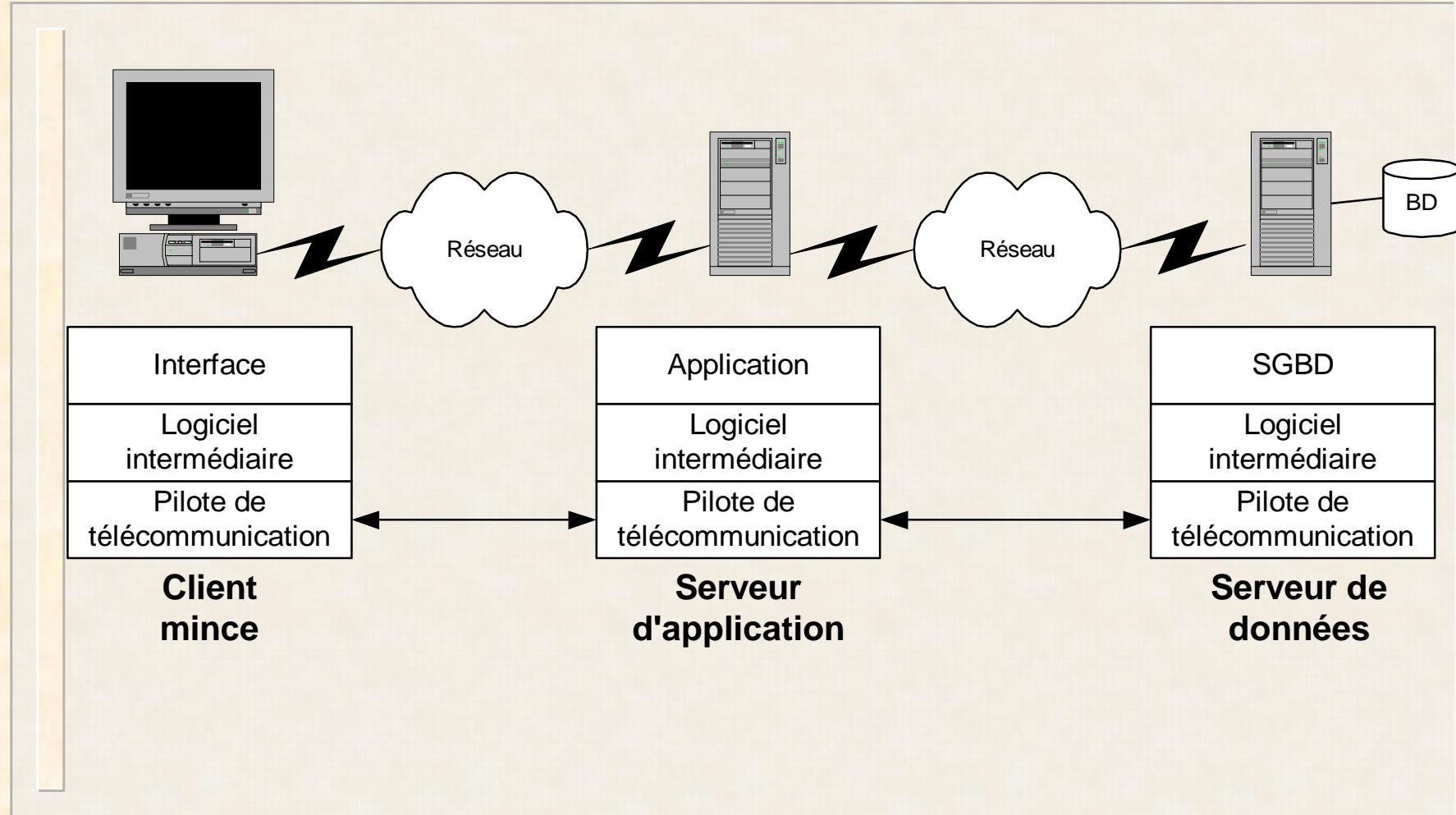
## ■ *Architecture du type client-serveur (client-server architecture)*

- programme d'application = *client*
  - interface (« GUI ») + traitement du domaine d 'application
- SGBD = *serveur* (*de données* « *data server* »)
- machines (sites) différentes
- *deux couches, niveaux, strates* (“*two tier*”)

# Architecture client-serveur



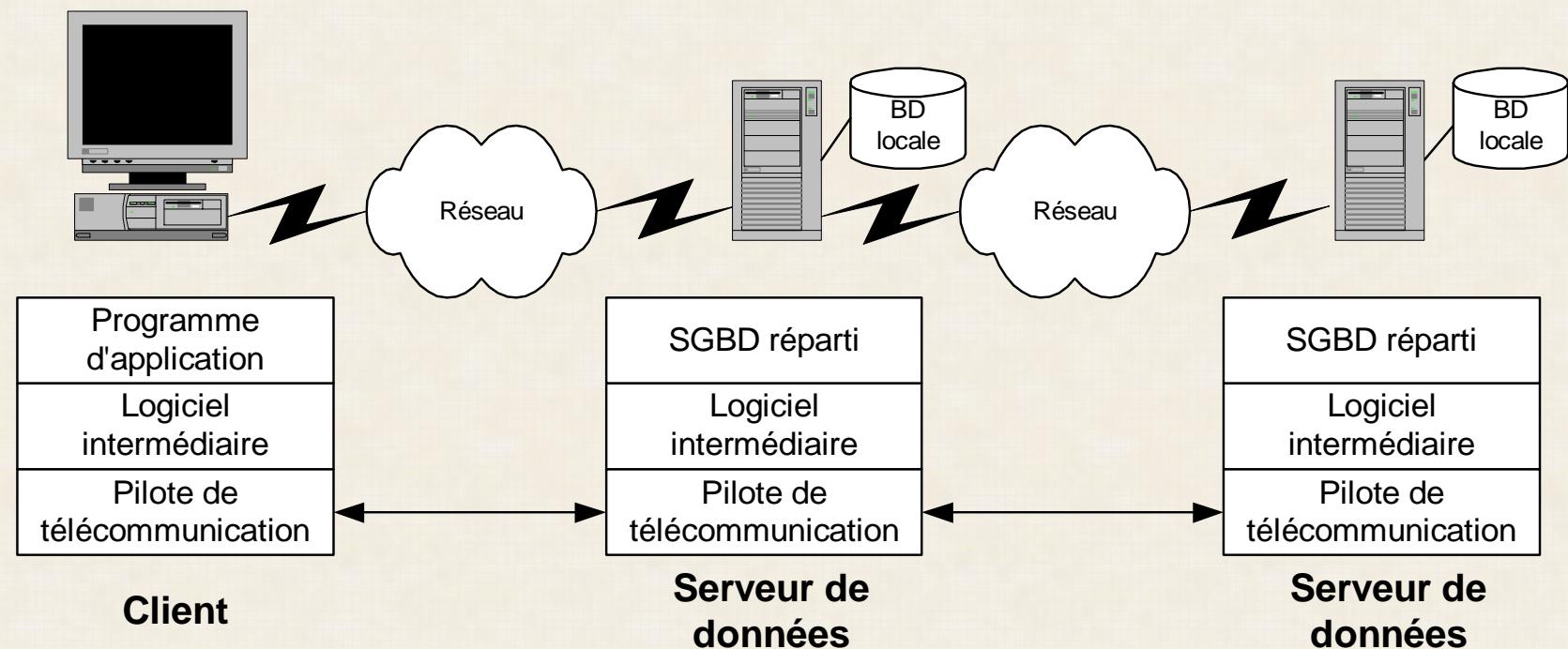
# Architecture à trois niveaux *(three tier)*



# Base de données répartie, distribuée (distributed database)

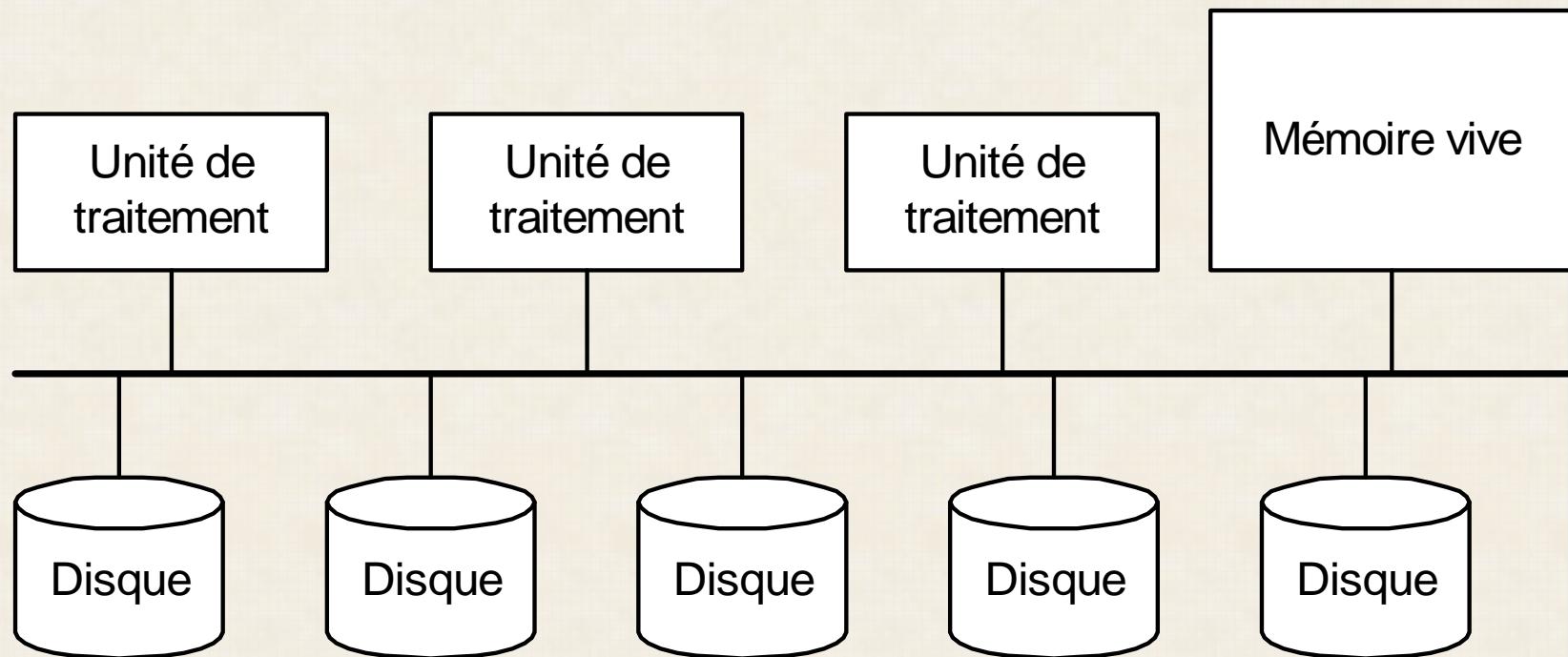
- *SGBD réparti (distributed DBMS)*

- répartition des données de manière transparente

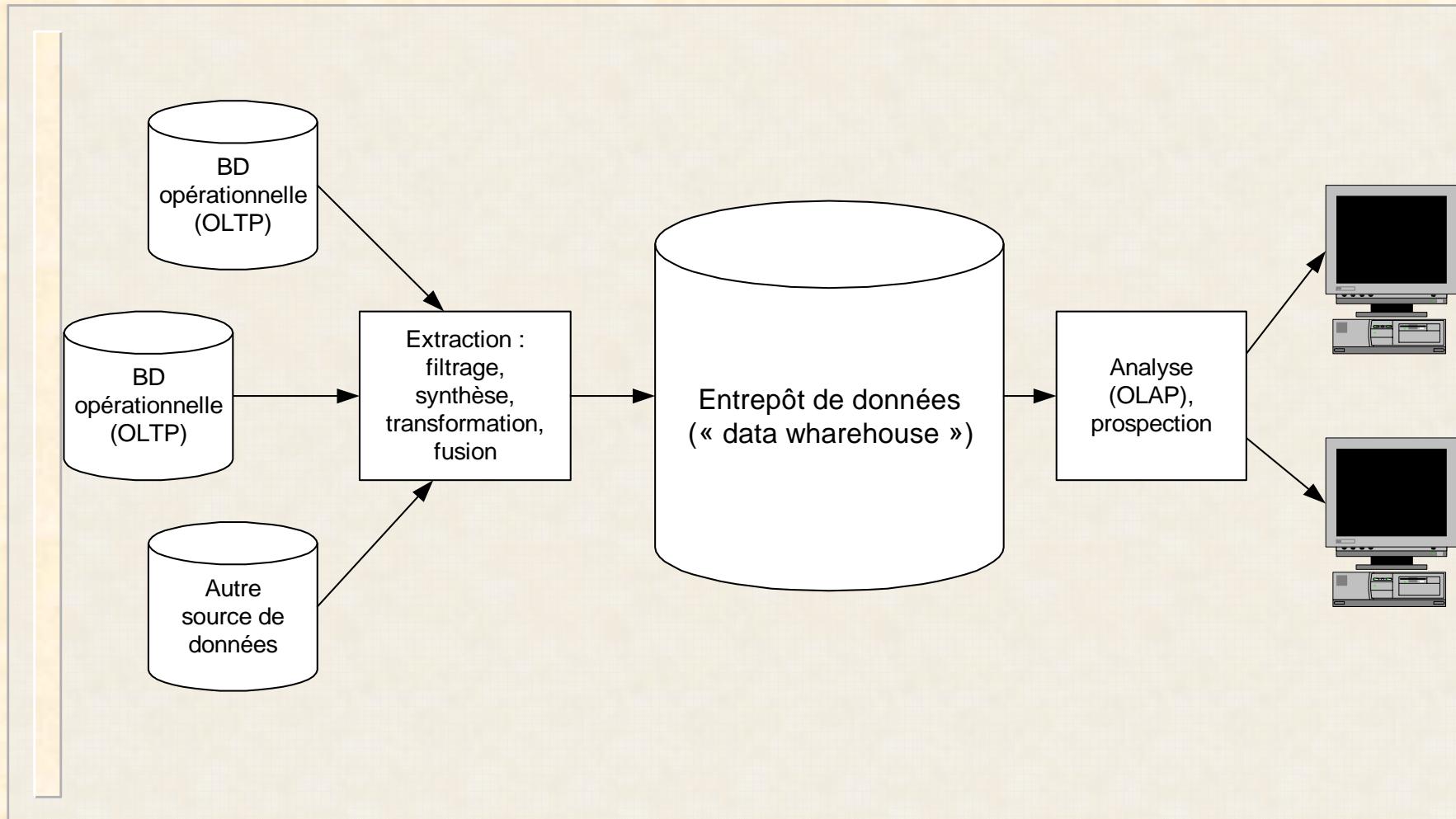


# Base de données parallèle (parallel database)

- Parallélisme à l'intérieur d'un site



# Entrepôt de données



# Entrepôt de données

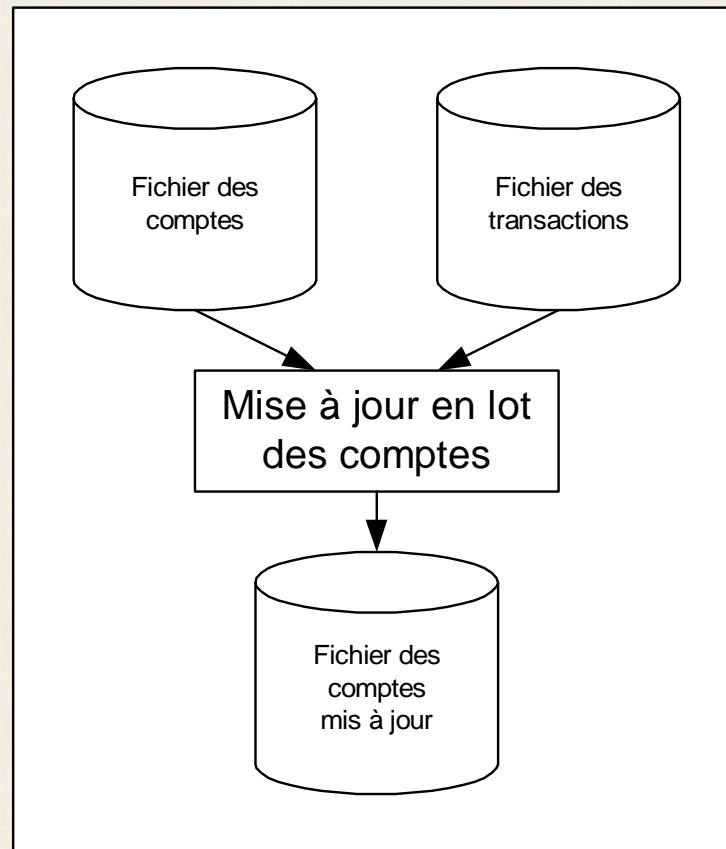
- ***Base de données opérationnelle***
  - traitement des données quotidiennes et récentes
  - OLTP (“*On Line Transaction Processing*”).
- ***Entrepôt de données (data warehouse)***
  - grand volume de données historiques extraites de bases opérationnelles pour le support à la prise de décision
  - OLAP (“*On Line Analytical Processing*”)
- ***Prospection de données , ou forage, fouille, exploration de données, ou découverte de connaissances dans les BD (data mining, analysis, dredging, archeology, knowledge discovery in databases - KDD)***
  - extraction non triviale d'informations implicites, inconnues et utiles
  - apprentissage machine , statistiques

# 1.7 Des systèmes de gestion de fichiers aux SGBD

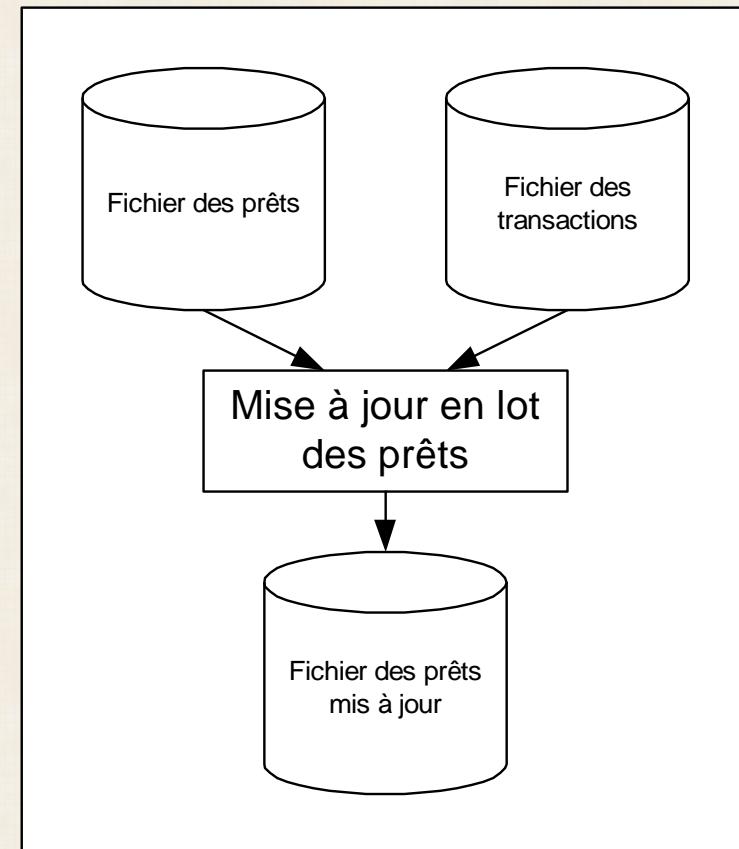
- *Système de gestion de fichier (SGF, file system)*
  - *gestionnaire de fichiers (file manager)*
  - composante du système d'exploitation
  - abstraction des mémoires secondaires sous forme d'un ensemble de fichiers

# Utilisation d'un système de gestion de fichier

Application de gestion des comptes



Application de gestion des prêts



# Exemple de contenu des fichiers de comptes et de prêts (fichiers séquentiels triés)

<b>noCompte</b>	<b>nomClient</b>	<b>adresseClient</b>	<b>noTéléphone</b>	<b>dateOuverture</b>	<b>solde</b>
100	Hugh Paycheck	Ottawa	(999)999-9999	5/05/1999	1000.0
200	Dollard Cash	Montréal	(888)888-8888	10/10/1999	2000.0
300	Hugh Paycheck	Ottawa	(999)999-9999	10/10/1999	1000.0
400	Ye San Le Su	Montréal	(777)777-7777	20/7/2000	5.00
600	Ye San Le Su	Montréal	(777)777-7777	15/10/2000	10.00

<b>noPrêt</b>	<b>nomClient</b>	<b>adresse Client</b>	<b>noTéléphone</b>	<b>montant Prêt</b>	<b>dateDébut</b>	<b>taux Intérêt</b>	<b>fréquence Paiement</b>
1000	Hugh Paycheck	Ottawa	(999)999-9999	10000.00	10/6/2000	10	12
2000	Ye San Le Su	Montréal	(777)777-7777	20000.00	20/7/2000	12	52
3000	Hugh Paycheck	Ottawa	(999)999-9999	5000.00	15/8/2000	12	12

# 1.7.1 Redondance et incohérence des données

Modifications anarchiques

noCompte	nomClient	adresseClient	noTéléphone	dateOuverture	solde
100	Hugh Paycheck	Ottawa	(999)999-9999	5/05/1999	1000.00
200	Dollard Cash	Montréal	(888)888-8888	10/10/1999	2000.00
300	Hugh Paycheck	Québec	(555)555-5555	10/10/1999	1000.00
400	Ye San Le Su	Montréal	(777)777-7777	20/7/2000	5.00
600	Ye San Le Su	Montréal	(777)777-7777	15/10/2000	10.00

noPrêt	nomClient	adresse Client	noTéléphone	montant Prêt	dateDébut	taux Intérêt	fréquence Paiement
1000	Hugh Paycheck	Alma	(444)444-4444	10000.00	10/6/2000	10	12
2000	Ye San Le Su	Montréal	(777)777-7777	20000.00	20/7/2000	12	52
3000	Hugh Paycheck	Ottawa	(999)999-9999	5000.00	15/8/2000	12	12

## **1.7.2 Mécanismes d'organisation de données limités des SGF**

- SGBD supporte
  - Indexage
  - Hachage
  - Organisation par grappe
  - Liens entre fichiers

## 1.7.3 Intégrité sémantique

- *Contrainte d'intégrité sémantique*  
*(semantic integrity constraint)*
  - ou simplement *contrainte d'intégrité*
  - une règle qui doit toujours être respectée par les données de la BD
  - ex: “ le solde d'un compte ne peut être négatif ”
- Supporté par SGBD (pas SGF)

## 1.7.4 Contrôle de concurrence

### ■ Empêcher interférences indésirables

Temps	Processus #1	Processus #2	État du fichier
$t_1$	<b>Lire</b> ( <i>SoldeCompte</i> , $s$ ) $\{s = 8\}$		$SoldeCompte = 8$
$t_2$		<b>Lire</b> ( <i>SoldeCompte</i> , $s$ ) $\{s = 8\}$	
$t_3$	$s = s + 5$		
$t_4$	<b>Écrire</b> ( $s$ , <i>SoldeCompte</i> )		$SoldeCompte = 13$
$t_5$		$s = s + 2$	
$t_6$		<b>Écrire</b> ( $s$ , <i>SoldeCompte</i> )	$SoldeCompte = 10$
$t_7$			Saperlipopette!

### ■ SGBD : mécanismes raffinés

## 1.7.5 Fiabilité

- Exemple de panne
- SGBD : mécanismes de récupération
- Revenir à un état cohérent suite à une panne
  - journal des transactions
  - duplication des composantes et des données

## 1.7.6 Sécurité

- SGF :
  - lecture, écriture au niveau fichier
- SGBD : contrôle plus fin
  - types de données
  - enregistrements
  - champs
  - type d'opération
  - cryptage

## 1.7.7 Gestion des transactions

### ■ *Transaction*

- séquence d'opérations de manipulation de données exécutées par un programme
- délimitées par une opération de *début* et une opération de *fin de transaction*
- propriétés d'*Atomicité*, de *Consistance*, d'*Isolation* et de *Durabilité* (ACID).

## 1.7.8 Indépendance des données

- Indépendance entre :
  - formulation d'une requête (QUOI)
    - langage non procédural
  - façon d'effectuer la requête par le SGBD (COMMENT)

# 1.8 Perspective historique

- ***Modèle de définition de données (“ data definition model”)***
  - nature des types de données supportés par le SGBD.
- 1ère génération (années 70): réseau, hiérarchique (CODASYL, IMS,...)
  - LMD navigationnel
- 2ième génération (années 80) :relationnel (Oracle, Ingres, DB2, SQL server,...)
  - LMD non procédural
- 3ième génération (années 90)
  - objet (Object Store, Versant,...)
  - relationnel objet (Oracle8, DB2 UDS, Informix US)
- 4ième génération ?
  - déductif (« Datalog »), entrepôt de données, data mining, support du WEB (XML, XMLQL)
- Modèles pour conception
  - E/R, UML

# 1.9 Exemple introductif : Banque *SansIntérêts*

Table <i>Client</i>			
<b>noClient</b>	<b>nomClient</b>	<b>adresseClient</b>	<b>noTéléphone</b>
10	Hugh Paycheck	Ottawa	(999)999-9999
20	Dollard Cash	Montréal	(888)888-8888
30	Ye San Le Su	Montréal	(777)777-7777

Table <i>Compte</i>			
<b>noCompte</b>	<b>solde</b>	<b>dateOuverture</b>	<b>noClient</b>
100	1000.00	5/05/1999	10
200	2000.00	10/10/1999	20
300	1000.00	10/10/1999	10
400	5.00	20/7/2000	30
600	10.00	15/10/2000	30

Table <i>Prêt</i>					
<b>noPrêt</b>	<b>montantPrêt</b>	<b>dateDébut</b>	<b>tauxIntérêt</b>	<b>fréquence Paiement</b>	<b>noClient</b>
1000	10000.00	10/6/2000	10	12	10
2000	20000.00	20/7/2000	12	52	30
3000	5000.00	15/8/2000	12	12	10

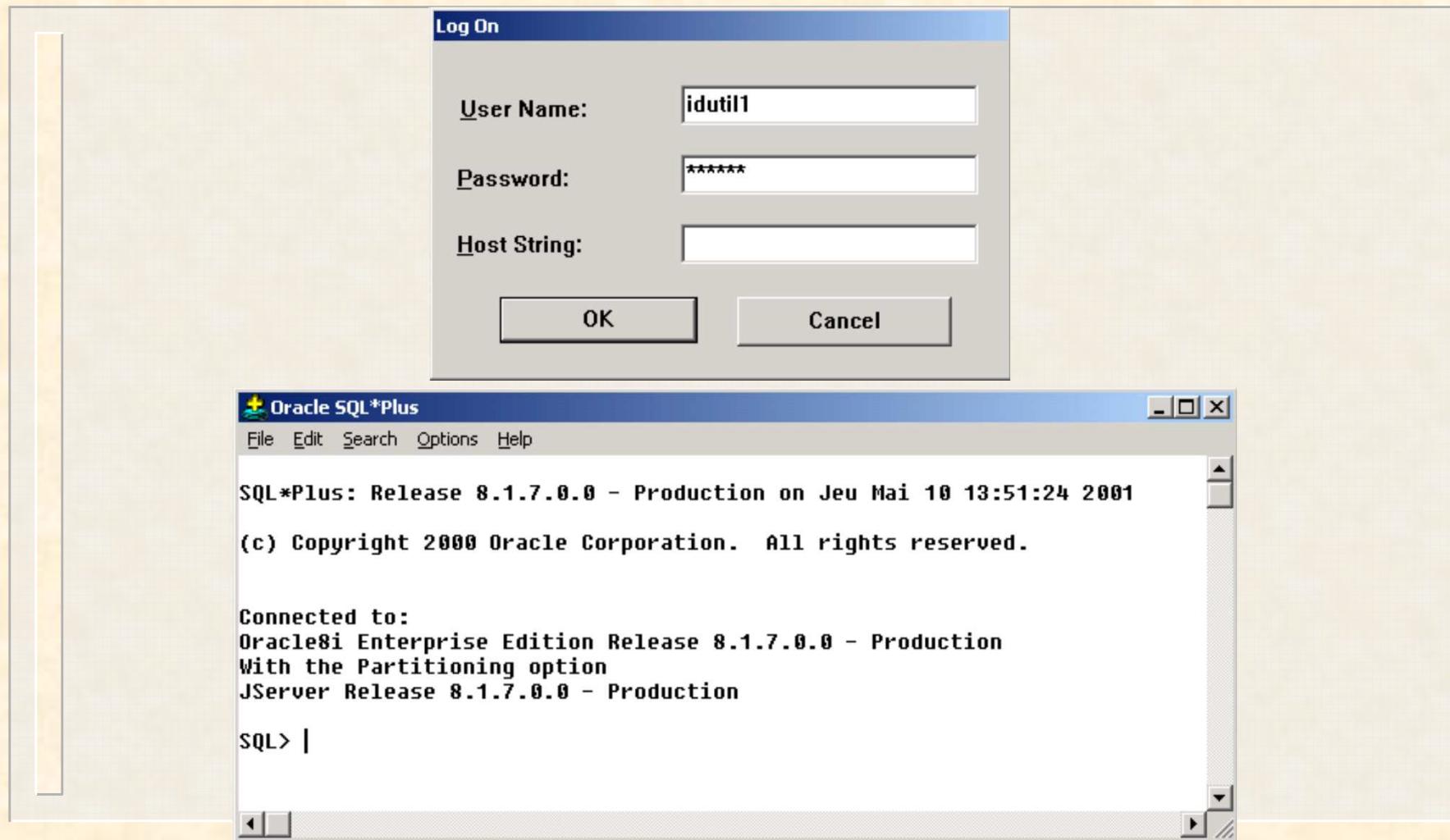
# Code du schéma avec le LDD SQL (script Oracle)

```
CREATE TABLE Client
(noClient          INTEGER PRIMARY KEY,
nomClient         VARCHAR(15),
adresseClient     VARCHAR(20),
noTéléphone       VARCHAR(15))

CREATE TABLE Compte
(noCompte          INTEGER PRIMARY KEY,
solde              DECIMAL(10,2) CHECK (solde >= 0),
dateOuverture     DATE,
noClient           INTEGER REFERENCES Client)

CREATE TABLE Prêt
(noPrêt            INTEGER PRIMARY KEY,
montantPrêt       DECIMAL(10,2),
dateDébut         DATE,
tauxIntérêt       DECIMAL(8,2),
fréquencePaiement INTEGER,
noClient           INTEGER REFERENCES Client)
```

# Création du schéma avec l'outil SQL\*plus d'Oracle



# Création du schéma

```
SQL> CREATE TABLE Client  
  2  (noClient      INTEGER PRIMARY KEY,  
  3   nomClient     VARCHAR(18),  
  4   adresseClient VARCHAR(20),  
  5   noTéléphone   VARCHAR(15))  
  6 /
```

Table created.

```
SQL> CREATE TABLE Compte  
  2  (noCompte      INTEGER PRIMARY KEY,  
  3   solde        DECIMAL(10,2) CHECK (solde >= 0),  
  4   dateOuverture DATE,  
  5   noClient      INTEGER REFERENCES Client)  
  6 /
```

Table created.

```
SQL> CREATE TABLE Prêt  
  2  (noPrêt       INTEGER PRIMARY KEY,  
  3   montantPrêt  DECIMAL(10,2),  
  4   dateDébut    DATE,  
  5   tauxIntérêt  DECIMAL(8,2),  
  6   fréquencePaiement INTEGER,  
  7   noClient     INTEGER REFERENCES Client)  
  8 /
```

Table created.

# Manipulation de données

```
SQL> INSERT INTO Client  
2      VALUES(10,'Luc Sansom','Ottawa','(999)999-9999')  
3 /  
  
1 row created.
```

```
SQL> SELECT *  
2   FROM Client  
3 /  
  
NOCLIENT NOMCLIENT          ADRESSECLIENT          NOTÉLÉPHONE  
----- -----  
      10 Luc Sansom           Ottawa                (999)999-9999
```

# Transaction (COMMIT)

```
-- Session parallèle avec Oracle (multiversion) :  
  
SQL> SELECT *  
  2  FROM Client  
  3  /  
  
no rows selected
```

```
SQL> COMMIT  
  2  /  
  
Commit complete.
```

```
Session parallèle :  
  
SQL> SELECT *  
  2  FROM Client  
  3  /  
  
NOCLIENT NOMCLIENT          ADRESSECLIENT          NOTÉLÉPHONE  
-----  
      10 Luc Sansom           Ottawa                  (999)999-9999
```

# Insertion d 'un lot en une transaction

```
SQL> INSERT INTO Client
  2      VALUES(20,'Dollard Tremblay','Montréal','(888)888-8888'
  3  /
1 row created.

SQL> INSERT INTO Client
  2      VALUES(30,'Lin Bô','Montréal','(777)777-7777')
  3  /
1 row created.

SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY'
  2  /
Session altered.

SQL> INSERT INTO Compte
  2      VALUES(100,1000.0,'5/5/1999',10)
  3  /
1 row created.

...
SQL> INSERT INTO Prêt
  2      VALUES(3000,5000,'15/8/2000',12,12,10)
  3  /
1 row created.

SQL> COMMIT
  2  /
Commit complete.
```

# Respect des contraintes d'intégrité

```
SQL> INSERT INTO Client
  2      VALUES(10,'Jean Leconte','Montréal','(666)666-6666')
  3  /
INSERT INTO Client
*
ERROR at line 1:
ORA-00001: unique constraint (IDUTIL1.SYS_C001737) violated
```

# SELECT SQL

```
SQL> SELECT noCompte, solde  
2  FROM Compte  
3  WHERE noClient = 10  
4  /
```

NOCOMPTE	SOLDE
100	1000
300	1000

noCompte	solde	dateOuverture	noClient
100	1000.00	5/05/1999	10
200	2000.00	10/10/1999	20
300	1000.00	10/10/1999	10
400	5.00	20/7/2000	30
600	10.00	15/10/2000	30

SELECT noCompte, solde  
FROM Compte  
WHERE noClient = 10

noCompte	solde
100	1000.00
300	1000.00

# Exécution d'un SELECT dans une application client-serveur Java

```
// Exemple de programme JAVA qui utilise le pilote JDBC thin d'Oracle
// pour effectuer un SELECT et itérer sur les lignes du résultat

// Il faut importer le paquetage java.sql pour utiliser JDBC
package ExemplesJDBC;
import java.sql.*;
import java.math.BigDecimal;

class ExempleSelectCompte
{
    public static void main (String args [])
        throws SQLException, ClassNotFoundException, java.io.IOException
    {
        // Charger le pilote JDBC d'Oracle
        Class.forName ("oracle.jdbc.driver.OracleDriver");

        // Connection à une BD avec un pilote thin
        Connection uneConnection =
            DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:ora817i",
                                      "idutill1", "oracle");

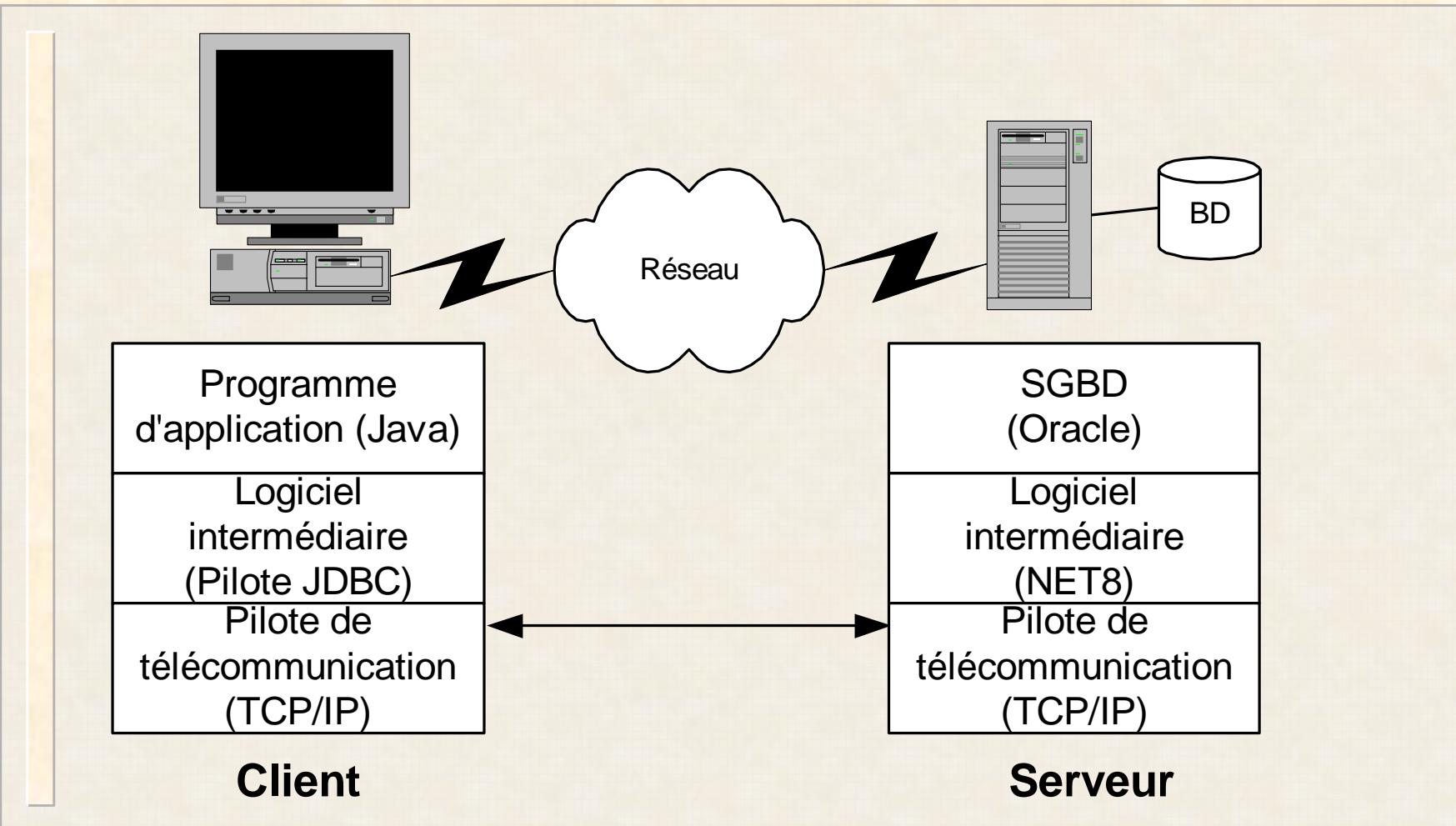
        // Création d'un énoncé associé à la Connection
        Statement unEnoncéSQL = uneConnection.createStatement ();

        // Exécution d'un SELECT
        // Le code du SELECT est passé en paramètre sous forme d'un String
        ResultSet résultatSelect = unEnoncéSQL.executeQuery
            ("SELECT noCompte, solde FROM Compte WHERE noClient = 10");

        // Itérer sur les lignes du résultat du SELECT et extraire les valeurs
        // des colonnes dans des variables JAVA
        while (résultatSelect.next ()) {
            int noCompte = résultatSelect.getInt ("noCompte");
            BigDecimal solde = résultatSelect.getBigDecimal ("solde");

            System.out.println ("Numéro du compte:" + noCompte);
            System.out.println ("Solde:" + solde);
        }
        // Fermeture de l'énoncé et de la connexion
        unEnoncéSQL.close();
        uneConnection.close();
    }
}
```

# Pilote JDBC



# SELECT sur plusieurs tables

```
SQL> SELECT nomClient, noCompte, solde  
  2  FROM Client, Compte  
  3 WHERE Client.noClient = Compte.noClient AND  
  4       dateOuverture = '10/10/1999'  
  5 /
```

NOMCLIENT	NOCOMPTE	SOLDE
Dollard Tremblay	200	2000
Luc Sansom	300	1000

# suite

## ■ Requête complexe

noClient	nomClient	adresseClient	noTéléphone
10	Hugh Paycheck	Ottawa	(999)999-9999
20	Dollard Cash	Montréal	(888)888-8888
30	Ye San Le Su	Montréal	(777)777-7777

noCompte	solde	dateOuverture	noClient
100	1000.00	5/05/1999	10
200	2000.00	10/10/1999	20
300	1000.00	10/10/1999	10
400	5.00	20/7/2000	30
600	10.00	15/10/2000	30

```
SELECT      nomClient, noCompte, solde
FROM        Client, Compte
WHERE       Client.noClient = Compte.noClient AND
           dateOuverture = '10/10/1999'
```

nomClient	noCompte	solde
Dollard Cash	200	2000.00
Hugh Paycheck	300	1000.00

# UPDATE SQL

```
SQL> UPDATE Compte  
2 SET solde = solde - 100  
3 WHERE noCompte = 100  
4 /
```

1 row updated.

```
SQL> UPDATE Compte  
2 SET solde = solde + 100  
3 WHERE noCompte = 300  
4 /
```

1 row updated.

```
SQL> COMMIT  
2 /
```

Commit complete.

```
SQL> SELECT noCompte, solde  
2 FROM Compte  
3 WHERE noClient = 10  
4 /
```

NOCOMPTE	SOLDE
100	900
300	1100

# DELETE SQL

```
SQL> DELETE FROM Compte WHERE noCompte = 100  
2  /
```

1 row deleted.

```
SQL> COMMIT  
2  /
```

Commit complete.

```
SQL> SELECT * FROM Compte  
2  /
```

NOCOMPTE	SOLDE	DATEOUVERT	NOCLIENT
200	2200	10/10/1999	20
300	1000	10/10/1999	10
400	5	20/07/2000	30
600	10	15/10/2000	30

# Organisation physique des données

```
SQL> CREATE INDEX indexNoClientCompte ON Compte(noClient)
2  /
```

Index created.

# Dictionnaire de données

```
SQL> SELECT TABLE_NAME  
2   FROM USER_TABLES  
3  /
```

TABLE_NAME
-----
CLIENT
COMPTE
PRÊT

```
SQL> SELECT COLUMN_NAME  
2   FROM USER_TAB_COLUMNS  
3  WHERE TABLE_NAME = 'CLIENT'  
4  /
```

COLUMN_NAME
-----
NOCLIENT
NOMCLIENT
ADRESSECLIENT
NOTÉLÉPHONE

# Allocation d 'espace

```
SQL> SELECT TABLESPACE_NAME, EXTENTS, BLOCKS  
  2  FROM USER_SEGMENTS  
  3  WHERE SEGMENT_NAME = 'CLIENT'  
  4  /
```

TABLESPACE_NAME	EXTENTS	BLOCKS
USERS	1	16

# 1.10 Conception du schéma d'une base de données

Modèle conceptuel

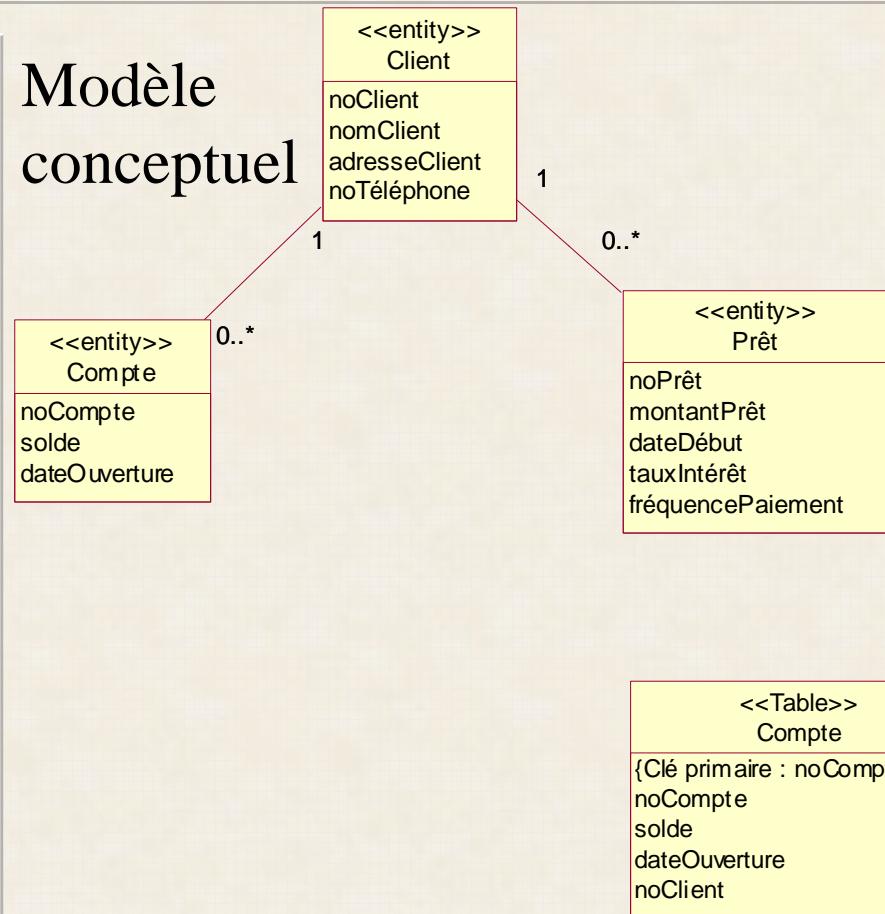
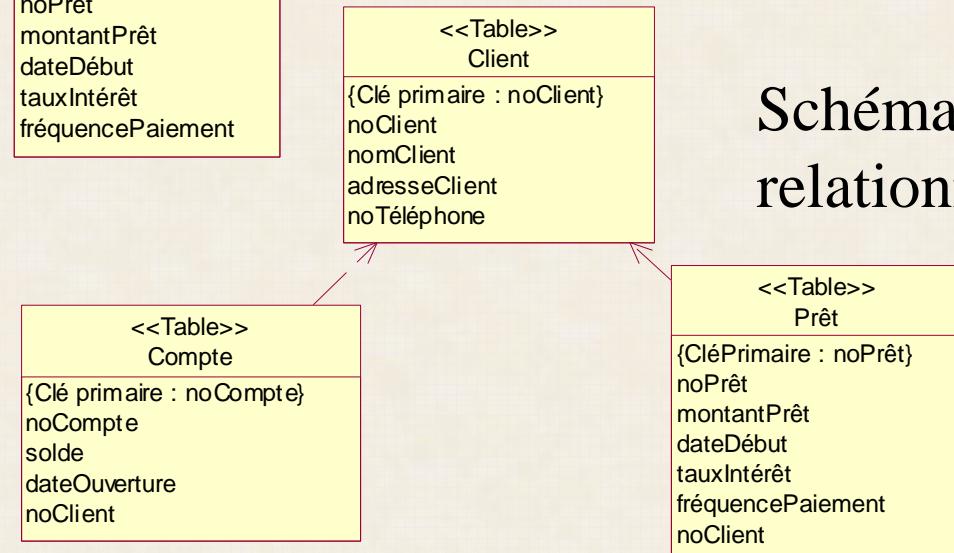


Schéma relationnel



## 1.11 Architecture à trois niveaux des schémas (ANSI/SPARC)

