



Université des sciences et de la Technologie
Houari Boumediene
USTHB – Alger

Département d'Informatique

ARCHITECTURE ET ADMINISTRATION

DES BASES DE DONNÉES

2014-2015

RESPONSABLES

M. KAMEL BOUKHALFA (SII)

MME Z. ALIMAZIGHI (MIL)

CHAPITRE 1

ARCHITECTURE

DÉFINITIONS

Base de données

- ❑ **Collection** de données **cohérentes** et **structurées** (données **persistantes**) utilisées par des systèmes informatiques.
- ❑ Collection de données **fortement structurées** et **persistantes** dont la structure est définie dans un **schéma** au moyen d'un **langage de définition de données**.
- ❑ Les données et le schéma sont gérés à l'aide d'un logiciel appelé **système de gestion de bases de données (SGBD)**.

3

DÉFINITIONS

Un système de gestion de base de données (SGBD)

- ❑ Un ensemble de **logiciels informatiques** qui sert à la **manipulation** des bases de données.
- ❑ Il sert à effectuer des opérations ordinaires telles que **consulter**, **modifier**, **construire**, **organiser**, **transformer**, **copier**, **sauvegarder** ou **restaurer** des bases de données.
- ❑ Il est souvent utilisé par d'autres logiciels ainsi que les administrateurs ou les développeurs.

4

SGBD

- ❑ L'ensemble, dont le composant central est **le moteur de base de données**, peut être sous forme de **composants logiciels**, de **serveurs**, de **logiciels applicatifs** ou **d'environnements de programmations**.
- ❑ Il permet généralement à **plusieurs utilisateurs** et **plusieurs logiciels** de manipuler **plusieurs bases de données en même temps** et ceci quel que soit le contenu et l'organisation des bases de données.

5

HISTORIQUE DES SGBD

- ❑ Premiers SGBD créés dans les années 1960
- ❑ Technologie répandue dès les années 1970
- ❑ Le consortium CODASYL a été formé dans les années 1960 en vue de produire des normes et standards en rapport avec les SGBD.
- ❑ Le consortium CODASYL est à l'origine de diverses normes en rapport avec le langage de programmation COBOL, un langage créé en 1965, axé sur la manipulation des bases de données.
- ❑ Le modèle de données relationnel a été créé en 1970.

6

HISTORIQUE

- ❑ Les premiers SGBD qui manipulent des bases de données relationnelles — IBM System R et Oracle V2 — sont apparus en 1978.
- ❑ 1981 : Informix
- ❑ 1983 : Oracle V3
- ❑ 1987 : SyBase SQL Server
- ❑ 1989 : PostGres
- ❑ 1992 : MS Access
- ❑ 1999 : Oracle 8i
- ❑ 2000 : SQL Server 2000
- ❑ 2005 : Oracle 10g R2, SQL Server 2005
- ❑ 2006 : IBM DB2

Pour plus de détail sur tout l'historique des SGBD, voir le lien suivant :

<http://fadace.developpez.com/sbgdcmp/story/>

7

GÉNÉRATIONS DE 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)

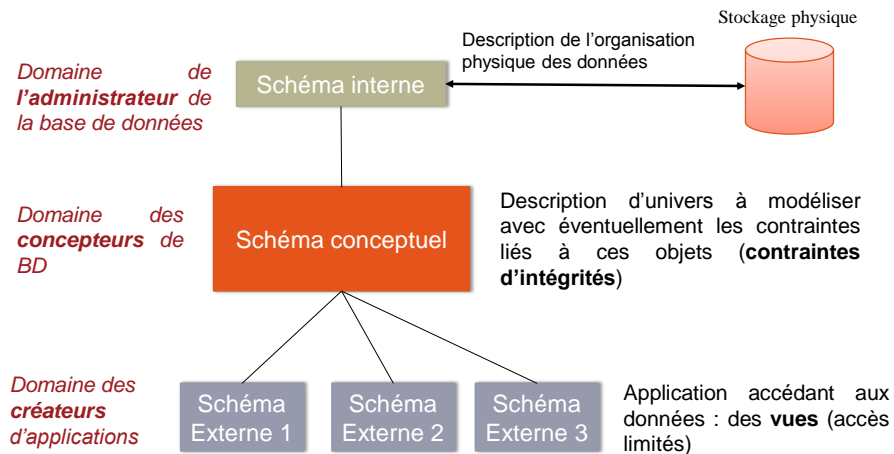
4ième génération ?

- ❑ Déductif (« Datalog »), entrepôt de données, data mining, support du WEB (XML, XMLQL)

8

NIVEAUX DE DESCRIPTION DES DONNÉES

Conceptuel (Logique), Interne et Externe



9

OBJECTIFS DES SGBD

Indépendance des données par rapport aux supports physiques

- ☐ Notion de relation qui définit la structure de données
- ☐ La structure d'implémentation d'une relation est la table sur laquelle seront définis des chemins d'accès (index, hachage, etc.).

Indépendance logique

- ☐ Le SGBD doit autoriser plusieurs visions d'utilisateurs sur la base
- ☐ Possible grâce à la notion de vues

10

OBJECTIFS DES SGBD

❑ Evolution de la BD

Evolution des données

- ❑ Insertion
- ❑ Suppression
- ❑ Modification

Evolution du schéma : Modification de la structure d'une relation

- ❑ Ajout, suppression d'attributs
- ❑ Ajout d'une nouvelle relation.

❑ Cohérence de données

- ❑ Intégrité des données
- ❑ Les données d'une base de données doivent vérifier des invariants appelés contraintes d'intégrité.

11

OBJECTIFS DES SGBD

Partageabilité des données et sécurité de fonctionnement

- ❑ Possibilité d'autoriser plusieurs utilisateurs à accéder en même temps à la base de données
- ❑ Gestion des conflits d'accès et d'écriture.

❑ Sécurité de la BD

- ❑ Gestion des utilisateurs
- ❑ Gestion des droits d'accès
- ❑ Gestion des transactions

❑ Intégrité de la BD

- ❑ Protection contre les pannes
- ❑ Reprise après panne
- ❑ Gestion des journaux log

12

PRINCIPALES ARCHITECTURES DES BD

❑ *Architecture centralisée*

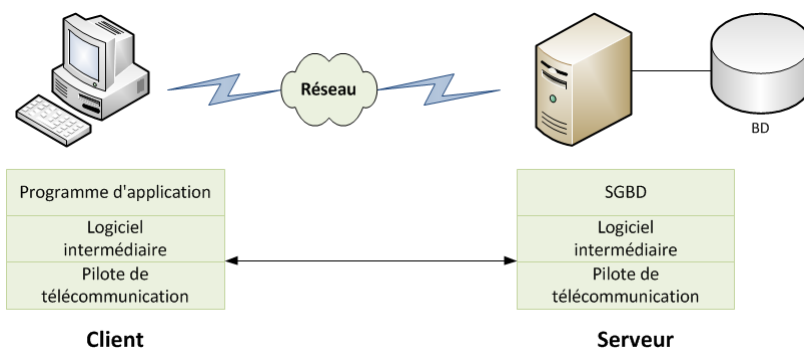
- ❑ Programme d'application et SGBD sur même machine (même site)
- ❑ Premiers systèmes

❑ *Architecture du type client-serveur*

- ❑ Programme d'application = *client*
 - ❑ Interface (IHM) + traitement du domaine d'application
- ❑ SGBD = *serveur de données*
 - ❑ machines (sites) différentes

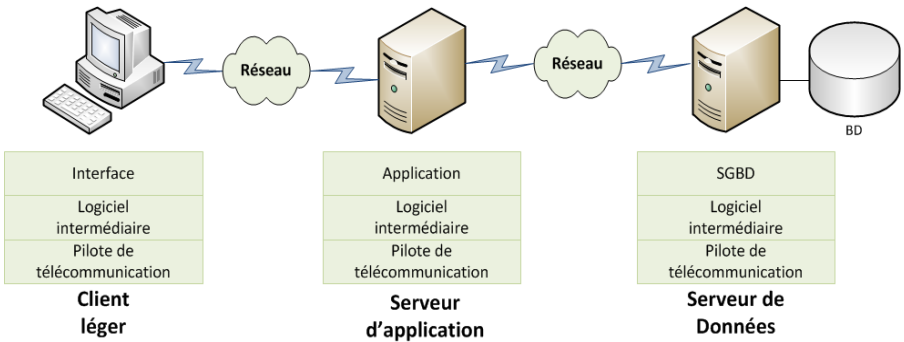
13

ARCHITECTURE CLIENT-SERVEUR



14

ARCHITECTURE À TROIS TIERS

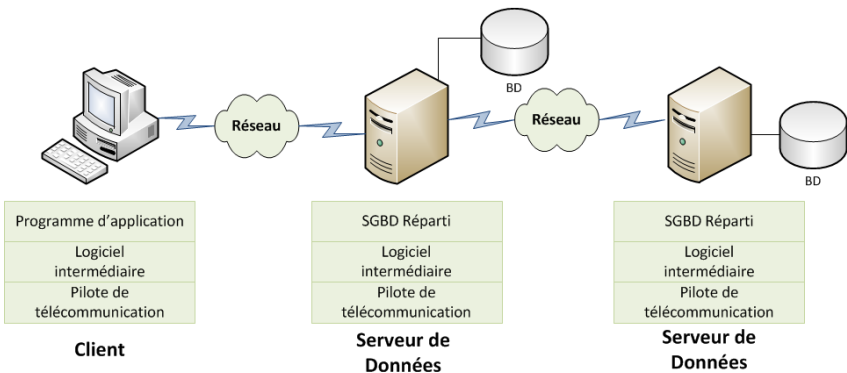


15

BASE DE DONNÉES RÉPARTIE, DISTRIBUÉE

SGBD réparti

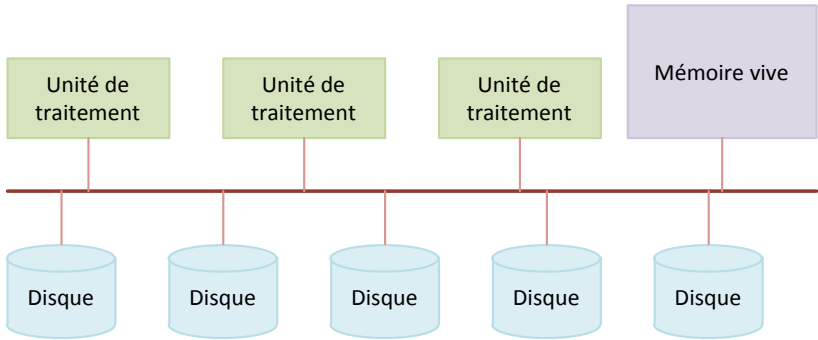
- ❑ Répartition des données de manière transparente
- ❑ Objet du module BDA (deuxième semestre)



16

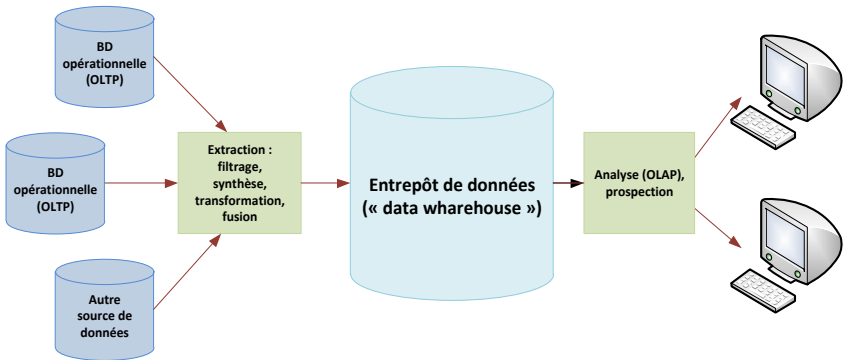
BASE DE DONNÉES PARALLÈLE

Parallélisme à l'intérieur d'un site

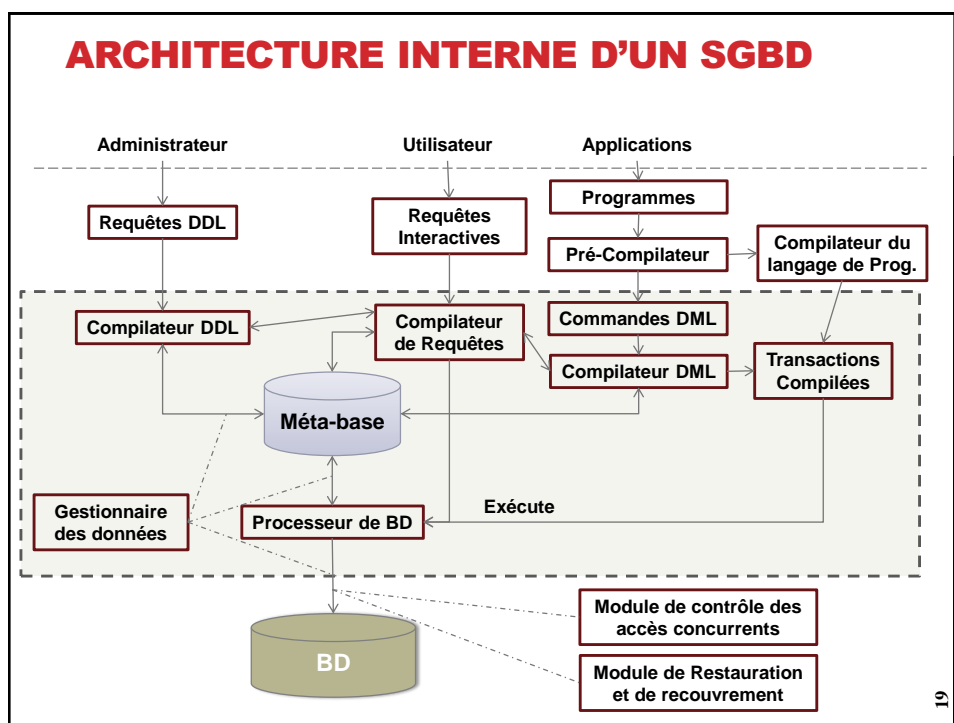


17

ENTREPÔT DE DONNÉES



18



ARCHITECTURE INTERNE D'UN SGBD

Gestionnaire de données

- ❑ La base de données ainsi que la méta-base sont stockées en mémoire secondaires. L'accès à leurs données est assuré par le gestionnaire de données.
- ❑ L'accès est d'abord contrôlé par le système de gestion de fichier, le gestionnaire de données est un module de plus haut niveau assurant l'accès aux données via un ensemble de buffers.

Métabase

- ❑ Contient toutes les informations sur les données comme la structure des tables, les types de données, les index créés, etc.

ARCHITECTURE INTERNE D'UN SGBD

Compilateur DDL

- ❑ Compile et exécute le schéma de définition de données et sauvegarde les descriptions dans la méta-base.

Compilateur DML

- ❑ Compile et exécute les requêtes de manipulation de données.

Compilateur de requêtes

- ❑ Traite les requêtes utilisateurs saisies de manière interactive. Il parse, analyse puis compile ces requêtes.

21

ARCHITECTURE INTERNE D'UN SGBD

Processeur de base de données

- ❑ Contrôle l'accès à la base de données. Il reçoit des ordres de recherche ou de mise à jour et les exécute sur la base de données. L'accès aux données est assuré par le gestionnaire de données.

Pré-compilateur

- ❑ Permet d'extraire les commandes DML pour une compilation en code objet. Le reste du code est envoyé au compilateur du langage de programmation, ensuite les deux codes objets sont fusionnés pour générer des transactions qui seront à leur tour envoyées au processeur de BD.

22

SGBD ORACLE

- ❑ Un système de gestion de base de données relationnel (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO).
- ❑ Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates.
- ❑ ORACLE a été édité par la société *Oracle Corporation*, implantée aux USA à *RedWood Shores* en *Californie*.
- ❑ Depuis, les produits ORACLE n'ont cessé d'évoluer avec l'évolution des technologies de l'information et de la communication. Aujourd'hui *ORACLE* est un *SGBD* réparti, qui s'est tourné vers le Web.

SITE ORACLE

The screenshot displays the Oracle website interface. At the top, there is a navigation bar with the Oracle logo on the left and a search bar on the right. Below the logo, a horizontal menu lists various categories: Products, Solutions, Downloads, Store, Support, Training, Partners, and About. A 'Sign In/Register' link is also present. The main content area is divided into several columns. The leftmost column features a 'Spotlight' section with a 'Register Now' button and an 'Announcing Oracle Solaris 11.2 Engineered for Cloud' section with a 'DOWNLOAD NOW' button. The central column lists product categories such as Oracle Cloud, Oracle Mobile, Applications, Database (with 'Oracle Database' highlighted), Enterprise Management, and Java. The rightmost column lists other product lines like Business Analytics, Engineered Systems, Servers, Storage and Tape, Networking and Data Center, and Virtualization.

ORACLE SOLUTIONS

ORACLE

Sign In/Register Help Country Communities I am a... I want to... Search

Products

Solutions

Downloads

Store

Support

Training

Partners

About

OTN

Spotlight

Cloud

Midsize

Industries

Optimized Data Center

Sustainability

Cloud Solutions

Oracle Applications Cloud

Mobile Solutions

Oracle Mobile Platform

Technology Solutions

Big Data

Data Warehousing

Database Migration

High Availability

Internet of Things

Java Technology

Linux

Open Source

Oracle Optimized Solutions

Security

Server Consolidation

Service-Oriented Architecture

Solaris

Virtualization

Windows and .Net

Business Solutions

Business Analytics

Business Process Services

Customer Experience

Customer Relationship Management

Enterprise Content Management

Enterprise Management

Financial Management

Governance, Risk, and Compliance

Human Capital Management

Marketing

Master Data Management

Database and IT Infrastructure for SAP

Oracle for Midsize Companies

Procurement

Project Portfolio Management

Social Relationship Management

Supply Chain Management

Sustainability

Industry Solutions

Aerospace and Defense

Automotive

Chemicals

Communications

Consumer Goods

Education and Research

Engineering and Construction

Financial Services

Healthcare

Health Sciences

High Technology

Industrial Manufacturing

Insurance

Life Sciences

Media and Entertainment

Natural Resources

Oil and Gas

Professional Services

Public Sector

Retail

Travel and Transportation

Utilities

Wholesale Distribution

Customer Experience Solutions

Marketing

Sales

Service

Commerce

Social

CPQ

Oracle Customer Programs

Customer and Partner Successes

Partner Solutions

Oracle Exastack Optimized

Oracle Exastack Ready

Oracle Validated Integration

Oracle Accelerate for Midsize Companies Solutions

Solutions

Live Sales Chat

ORACLE - OUTILS

ORACLE

Sign In/Register Help Country Communities I am a... I want to... Search

Products

Solutions

Downloads

Store

Support

Training

Partners

About

OTN

Popular Downloads

Java for Developers

Java for Your Computer

Java SE

Solaris

MySQL

Oracle Fusion

Middleware 11g

Oracle Database

Prebuilt Developer VMs

Database

Oracle Database

Oracle Database 11g Express Edition

MySQL

Oracle Berkeley DB

Oracle Instant Client

Oracle Application Express

See All...

Middleware

Oracle Fusion Middleware (including Oracle WebLogic Server)

Oracle JRockit

Oracle SOA Suite

See All...

Enterprise Management

Oracle Enterprise Manager

Oracle Application Testing Suite

See All...

Server and Storage Systems

Solaris

Linux and Oracle VM

Firmware

See All...

Developer Tools

SQL Developer

JDeveloper and ADF

Developer Tools for Visual Studio

Enterprise Pack for Eclipse

NetBeans IDE

See All...

Prebuilt Developer VMs

Applications

Oracle Fusion Applications, Oracle E-Business Suite, PeopleSoft, JD Edwards, Siebel CRM

Agile

AutoVue

See All...

Webcast: Proven Approach to Financial Progress Using Modern Best Practices

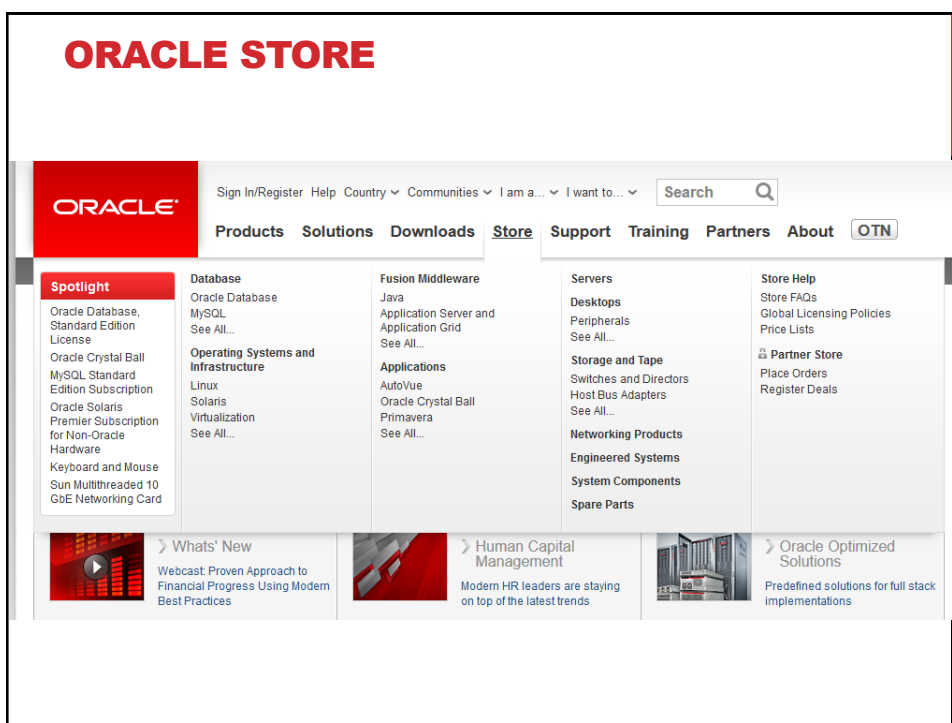
Management

Modern HR leaders are staying on top of the latest trends

Solutions

Predefined solutions for full stack implementations

13



HISTORIQUE

- ☐ Software Development Laboratories (SDL) a été créé en 1977.
- ☐ En 1979, SDL change de nom en devenant Relational Software, Inc. (RSI) et introduit son produit Oracle V2 comme base de données relationnelle.
- ☐ La version 2 ne supportait pas les transactions mais implémentait les fonctionnalités SQL basiques de requête et jointure.
- ☐ Il n'y a jamais eu de version 1, pour des raisons de marketing, la première version a été la version 2.
- ☐ Celle-ci fonctionnait uniquement sur les systèmes Digital VAX/VMS.

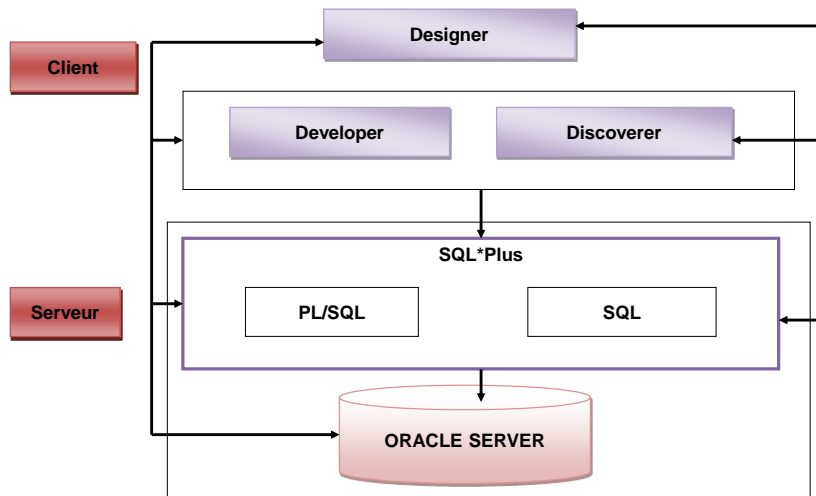
HISTORIQUE

- ❑ En 1983, RSI devient Oracle Corporation. La version 3 d'Oracle, entièrement ré-écrite en langage de programmation C, est publiée. C'est à partir de cette version que la plate-forme Unix est supportée.
- ❑ En 1984, la version 4 d'Oracle apparaît.
- ❑ Début 1985, Oracle commence à intégrer le modèle client-serveur, avec l'arrivée des réseaux au milieu des années 1980. La version 5 d'Oracle supporte donc les requêtes distribuées.
- ❑ En 1988, Oracle met sur le marché son ERP - Oracle Financials basé sur la base de données relationnelle Oracle. Oracle version 6 supporte le PL/SQL, le verrouillage de lignes (row-level locking) et les sauvegardes à chaud (hot backups, lorsque la base de données est ouverte).
- ❑ En 1992, la version 7 d'Oracle supporte les contraintes d'intégrité, les procédures stockées et les déclencheurs (triggers).
- ❑ En 1997, la version 8 introduit le développement orienté objet et les applications multimédia.

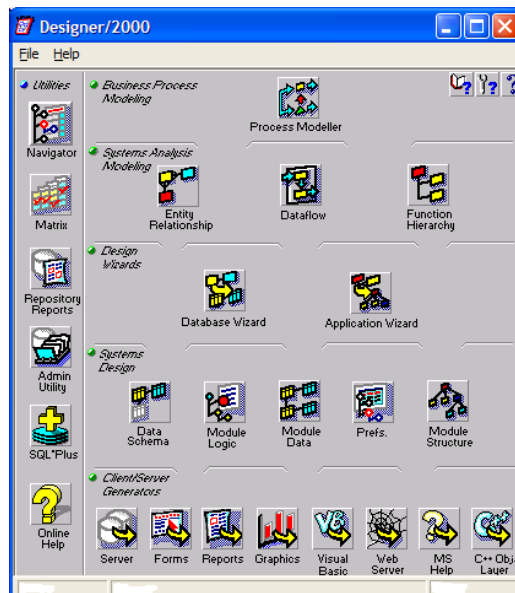
HISTORIQUE

- ❑ En 1999, la version 8i est publiée dans le but d'affiner ses applications avec Internet. La base de données comporte nativement une machine virtuelle Java.
- ❑ En 2001, Oracle 9i ajoute 400 nouvelles fonctionnalités et permet de lire et d'écrire des documents XML.
- ❑ À partir de la version 9i, intégration du moteur OLAP au sein de Oracle. Les données multidimensionnelles sont accessibles à partir du langage SQL.
- ❑ En 2003, la version 10g est publiée. Le g signifie « grid » ; un des atouts marketing de la 10g est en effet qu'elle supporte le « grid computing ».
- ❑ En 2005, vers la fin novembre, une version complètement gratuite est publiée, la « Oracle Database 10g Express Edition ». Apparaît cette même année, Oracle Database 10.2.0.1 — aussi connue sous le nom d'Oracle Database 10g Release 2 (10gR2).
- ❑ En 2007, sortie de la version 11 : Oracle Database 11g, pour Linux et Windows.
- ❑ Septembre 2009, sortie de Oracle Database 11g Release 2.

ARCHITECTURE



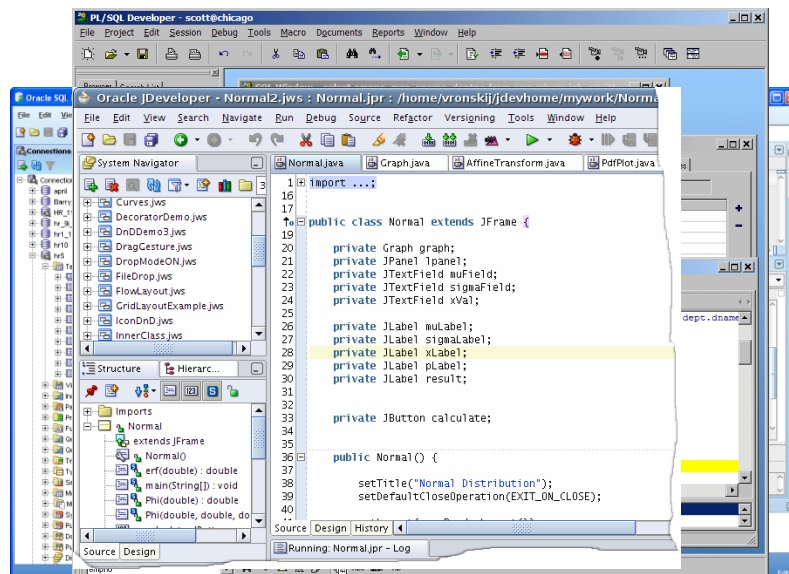
ORACLE DESIGNER



ORACLE DISCOVERER



ORACLE DEVELOPER

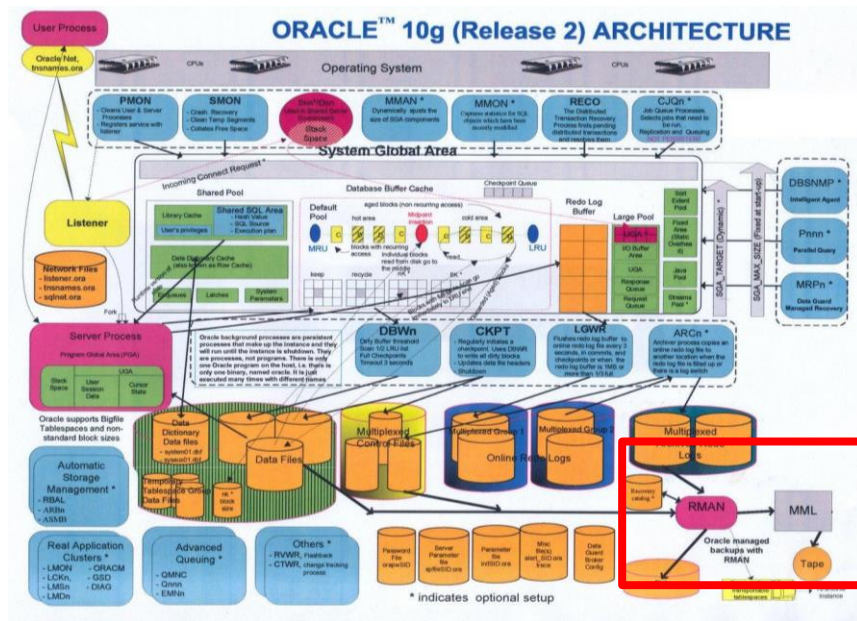


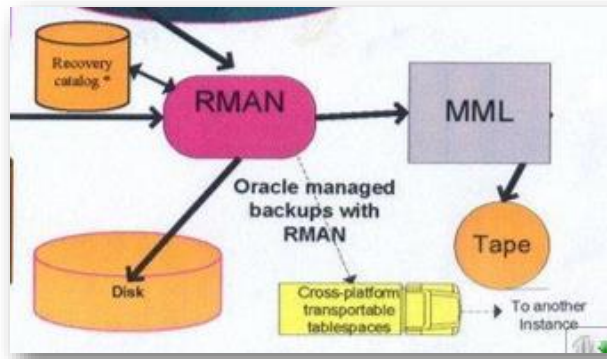
ARCHITECTURE

L'environnement ORACLE est un ensemble de produits autour de sa base de données

- ❑ **ORACLE Server** (gestionnaire de la base de données)
 - ❑ il contrôle toutes les actions au niveau de la BD comme l'accès utilisateur et la sécurité, stockage et intégrité des données.
- ❑ Le langage SQL et l'extension PL/SQL (langage comprenant des commandes procédurales supportant la gestion des erreurs et déclaration de variables)
- ❑ **ORACLE Designer**
 - ❑ un ensemble de produits intégrés dans un référentiel unique d'entreprise pour la conception des applications (Exemple CASE).
- ❑ **ORACLE Developer**
 - ❑ Outils de développement d'applications client/serveur ou Internet (Forms, Reports, Jdeveloper).
- ❑ **ORACLE Discoverer**
 - ❑ Outil d'interrogation pour des utilisateurs qui ont besoin d'accéder par eux-mêmes aux données, Datawarehouse, Datamart (Discoverer Administrator, Discoverer Catalog, etc.).

ARCHITECTURE

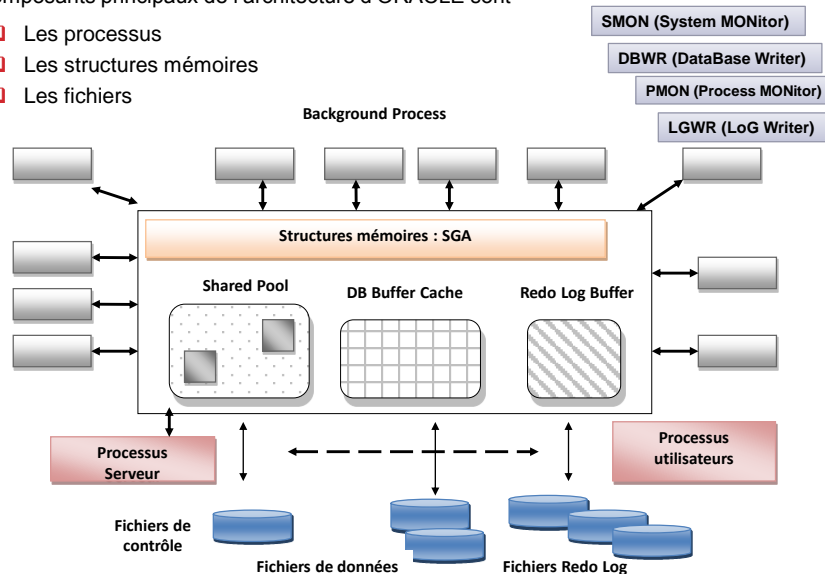




COMPOSANTS

Les composants principaux de l'architecture d'ORACLE sont

- ☐ Les processus
- ☐ Les structures mémoires
- ☐ Les fichiers



SYSTEM GLOBAL AREA

Représente l'ensemble des buffers nécessaires à la gestion des transactions. La SGA est également appelée Shared Global Area, elle est partagée entre plusieurs utilisateurs. La SGA est une zone mémoire allouée au démarrage d'une instance, elle est restituée à la fermeture de cette instance.

Les données de la SGA sont partagées par l'ensemble des utilisateurs connectés à un moment donné ; elles sont divisées en plusieurs types de buffers :

☐ Data base Buffer Cache

- ☐ Contient les blocs de données, les blocs d'index, des blocs contenant les RollBack Segments (ils enregistrent les actions d'une transaction pour la reprise en cas de panne) et des blocs pour la gestion du système, les plus récemment utilisés; il peut contenir des données modifiées qui n'ont pas encore été enregistrées sur disque.

SYSTEM GLOBAL AREA

☐ Redo Log Buffer

- ☐ Contient les *redo entries*, ensemble des modifications réalisées sur la base ; ces redo entries sont mémorisées sur un fichier redo log, qui pourra être utilisé en cas de panne.

☐ Shared Pool ou zone de partage des ordres SQL

- ☐ Utilisé pour mémoriser, analyser et traiter les ordres SQL soumis par les utilisateurs. Elle peut réutiliser les ordres SQL déjà exécutés.

LES PROCESSUS

Les Oracle Process sont divisés en trois catégories

- ☐ Users Process
- ☐ Server Process
- ☐ Backgrounds Process.

Les Background Process (Process détachés) prennent en charge les mécanismes internes d'Oracle. Une instance Oracle contient au minimum quatre background process : DBWR, LGWR, SMON et PMON.

BACKGROUND PROCESS

- ☐ **DBWR (DataBase Writer)** : Son rôle est de mettre à jour les fichiers de données. Les blocs de données modifiés en SGA sont stockés dans la base de données.
- ☐ **LGWR (LoG Writer)** : Ce process écrit séquentiellement dans le fichier Redo Log courant des enregistrements Redo Log de la SGA.
- ☐ **SMON (System MONitor)** : Il vérifie si la base a été arrêtée proprement. Si ce n'est pas le cas, il récupère dans les fichiers redo log les enregistrements validés, qui n'ont pas encore été écrits dans la base par Oracle, pour les insérer.
- ☐ **PMON (Process MONitor)** : Il administre les différents process d'Oracle. Il est chargé de la libération des ressources occupées, par exemple dans le cas où un client éteint sa machine sans se déconnecter.

BACKGROUND PROCESS

- ☐ CKPT : Checkpoint
- ☐ ARCH : Archiver
- ☐ RECO : Recover
- ☐ LCKn : LOCK
- ☐ SNPh : Snapshot Refresh
- ☐ Snnn : Shared server
- ☐ Dnnn : Dispatcher
- ☐ Pnnn : Parallel Query

PROCESSUS UTILISATEUR

Un process utilisateur est démarré lorsqu'un utilisateur exécute un programme applicatif.

- ☐ Exécute l'outil ou l'application
- ☐ Considéré comme étant le client.
 - ☐ Exemple : SQL*plus, Forms,
- ☐ Il transmet le SQL au process serveur et en reçoit les résultats.

PROCESSUS SERVEUR

- ❑ Prennent en charge les demandes des utilisateurs.
- ❑ Ils sont responsables de la communication entre la SGA et le Process User.
- ❑ Les tâches du process serveur
 - ❑ Analyser et exécuter les commandes SQL
 - ❑ Lire les blocs de données à partir du disque dans les zones partagées de la SGA
 - ❑ Renvoyer les résultats des commandes SQL au Process utilisateur

INSTANCE ORACLE

C'est la combinaison de la SGA et des process détachés de la base de données. Quand une instance est démarrée, les zones mémoire de la SGA sont allouées et les process détachés sont lancés.

Ne pas confondre une BD ORACLE et une instance d'ORACLE : l'instance est d'abord démarrée puis elle monte la BD (ouverture des fichiers).

Les process serveurs et utilisateurs ne font pas partie de l'instance d'ORACLE.

MÉCANISME

Lorsqu'un utilisateur demande une donnée, son processus va la chercher en SGA, si elle n'y est pas, le processus va la chercher dans les fichiers de données.

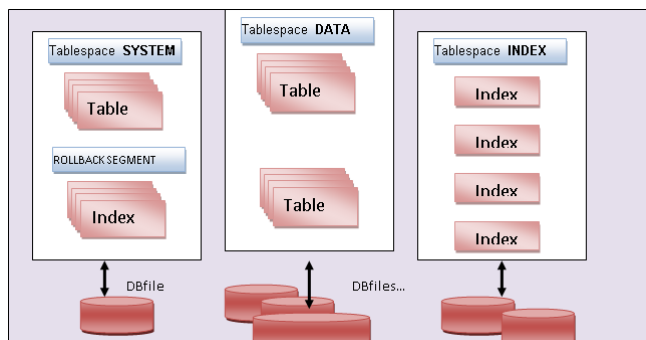
Toutes les transactions sont enregistrées dans les fichiers Redo Log, En cas de problème la reprise après panne est assurée par le process SMON au démarrage d'une nouvelle instance.

BD ORACLE

La base de données ORACLE est composée de fichiers de données et de fichiers redo log tous étant référencés dans un ou plusieurs fichiers de contrôle.

Structure physique	Définition
Fichiers de données	Contiennent toutes les données de la base : les tables, index, procédures, fonctions,... ; ainsi que le dictionnaire de données (créé lors de la création de la base de données).
Fichiers Redo Log	Gardent des enregistrements de toutes les modifications effectuées sur la base de données dans un but de restauration
Fichiers de contrôle	Conservent la définition des structures physiques et les états de la Base de données: nom de la base, nom et chemin d'accès des fichiers de données et Redo Log, date et heure de création de la base.

STRUCTURE INTERNE



LES TABLESPACES

Les données d'une base Oracle sont mémorisées dans une ou plusieurs unités logiques appelées **tablespaces** et physiquement dans des fichiers associés à ces tablespaces.

Chaque base contient obligatoirement un tablespace SYSTEM, celui-ci contient les tables du dictionnaire de données, les procédures, les fonctions, les packages, les triggers et le rollback segment SYSTEM.

Les autres tablespaces contiennent les segments de la base de données (tables, index,...).

STRUCTURE INTERNE

LES ROLLBACK SEGMENTS

- ☐ Une base de données contient un ou plusieurs ROLLBACK SEGMENTS ; un rollback segment enregistre les actions d'une transaction qui peuvent être annulées en cas d'incident.
- ☐ Le rollback segment SYSTEM est créé lors de la création de la base dans le tablespace SYSTEM ; il n'est utilisé que pour les transactions portant sur les données du dictionnaire.
- ☐ Un ou plusieurs autres rollback segments doivent exister pour les transactions portant sur des données utilisateur.

FONCTIONS DE SGBD

- ☐ **Création / Modification**
 - ☐ Création : CREATE TABLE
 - ☐ Modification : ALTER TABLE
 - ☐ Suppression : DROP TABLE, DELETE
 - ☐ Insertion : INSERT INTO
 - ☐ Mise à Jour : UPDATE
 - ☐ Interrogation : SELECT
- ☐ **Chemins d'accès**
 - ☐ Index : CREATE INDEX
- ☐ **Catalogues**
 - ☐ Informations sur toutes les objets de la Base de Données
- ☐ **Schémas externes**
 - ☐ Les vues : CREATE VIEW

LES CATALOGUES

Contraintes et Triggers

LES CATALOGUES

CATALOGUES

MÉMOIRE RELATIONNELLE

- ☐ Couche du SGBD qui accède à la BD
- ☐ Quatre fonctions principales
 1. **Présenter une vision relationnelle des fichiers qui constituent la BD : Fichiers, enregistrements, champs**
 2. **Gérer la mémoire centrale (MC)**
 - ☐ Relations stockées par page en MS
 - ☐ Chargées en MC sur demande (mémoire virtuelle)
 - ☐ MS est décomposée en segments
 - ☐ Un segment est constitué de plusieurs pages
 - ☐ La page est l'unité de transfert entre MS et MC
 - ☐ Un fichier est sauvegardé dans un seul segment (plusieurs pages)
 3. **Gérer les différentes relations de la BD**
 - ☐ Tables, index, Vues, catalogues, etc.
 4. **Etablir des méthodes d'accès**
 - ☐ Hachage, index, b-arbre, etc.

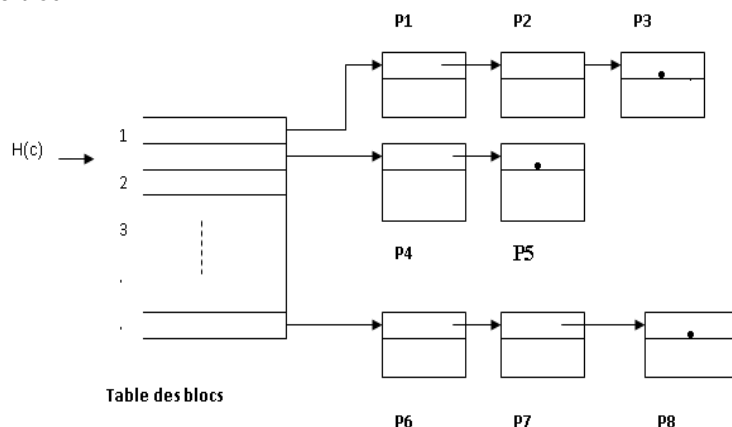
MÉTHODES DE STOCKAGE

Fichier Commun (Tas de données)

- ☐ Les enregistrements sont placés à la queue dans des pages successives.
- ☐ Un enregistrement n'est pas à cheval sur deux pages.
- ☐ Le seul accès aux données dans le cas de la recherche d'un enregistrement est un balayage séquentiel des tuples.
 - ☐ **Opération très coûteuse.**
- ☐ L'insertion d'un nouvel enregistrement s'effectue dans la dernière page.
- ☐ Si cette dernière est saturée, alors une nouvelle page est allouée et l'enregistrement est inséré.
- ☐ La suppression est assurée logiquement grâce à un indicateur de suppression.
- ☐ Cette organisation a été implémentée par défaut dans le système INGRES.

HACHAGE

- ❑ Disposer d'une fonction de hachage $h(c)$ qui permet de calculer un numéro de bloc ou paquet, contenant un ensemble de pages, à partir d'une clé.
- ❑ La recherche de l'enregistrement s'effectue ensuite séquentiellement dans le bloc.



HACHAGE

Algorithme de recherche

- ❑ Entrée : valeur de clé c
- ❑ Calcul $h(c)$: numéro de bloc
- ❑ Consultation de la table des blocs : récupération de la première page du bloc
- ❑ Recherche dans cette page l'enregistrement ayant pour clé c .

Algorithme de modification :

- ❑ rechercher l'enregistrement à l'aide de l'algorithme précédent
- ❑ réaliser la modification

HACHAGE

Algorithme d'insertion :

- ☐ Rechercher si le nouvel enregistrement n'existe pas.
- ☐ Si non : si le bloc n'est pas saturé alors insérer le nouvel enregistrement, sinon, allouer une nouvelle page, insérer le nouvel enregistrement et chaîner la nouvelle page aux autres.
- ☐ Le bloc est saturé signifie qu'il va y avoir débordement. C'est la gestion des débordements qui va dégrader les performances dans les techniques de hachage.

Algorithme de suppression :

- ☐ Rechercher l'enregistrement à supprimer
- ☐ Soit libérer la place qu'occupait cet enregistrement en mettant à jour le chaînage,
- ☐ soit mettre un indicateur de suppression dans l'en-tête de l'enregistrement à supprimer.

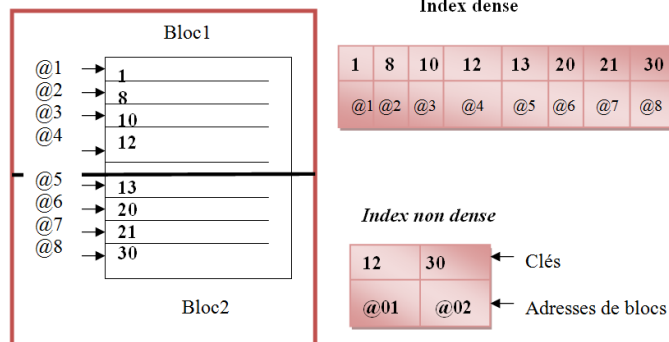
FICHIERS INDEXES

Un fichier index contient un ensemble de couples (c,p) où c est la clé du premier enregistrement de la page p.

Index dense : il contient toutes les clés du fichier

Index non dense : on crée des enregistrements index pour certains enregistrements du fichier : dans ce cas le fichier est trié et divisé en blocs. A chaque bloc lui est associée une entrée dans l'index.

(c,p) = < plus grande clé du bloc, adresse relative du bloc>



FICHIERS INDEXES

Algorithme de recherche

- ☐ Accès à l'index,
- ☐ Recherche dans l'index de la clé d'enregistrement désiré,
- ☐ Récupération dans l'index de l'adresse relative de l'enregistrement (si index dense), ou de l'adresse relative du bloc qui le contient (si index non dense),
- ☐ Conversion de l'adresse relative en adresse réelle,
- ☐ Accès à l'enregistrement ou au bloc,
- ☐ Transfert de l'enregistrement dans la zone du programme utilisateur.

FICHIERS INDEXES

Algorithme d'insertion :

- ☐ Accès à l'index,
- ☐ Détermination de l'emplacement de la page qui doit contenir l'enregistrement, puis détermination de la place de l'enregistrement dans la page.
- ☐ Si la place existe (page non saturée), alors insérer l'enregistrement en déplaçant les autres si nécessaire.
- ☐ Si la page est pleine, il existe différentes stratégies, entre autres, aller à la page suivante ou allouer une nouvelle page, tout en mettant à jour l'index.

FICHIERS INDEXES

Algorithme de suppression :

- ☐ Appliquer l'algorithme de recherche pour trouver l'enregistrement,
- ☐ Soit supprimer réellement l'enregistrement en mettant à jour l'index,
- ☐ Soit faire une suppression logique.
- ☐ **Cas particuliers**
 - ☐ Si l'enregistrement à supprimer est le premier de l'index, alors une modification de l'index est nécessaire.
 - ☐ Lorsqu'une page devient complètement vide, il faut la rendre au système et mettre à jour l'index.

FICHIERS INDEXES

Algorithme de modification :

- ☐ Appliquer l'algorithme de recherche pour trouver l'enregistrement à modifier,
- ☐ Réaliser la modification.
- ☐ Cas particulier
 - ☐ Si la modification porte sur la clé, alors la traiter comme une suppression, suivie d'une insertion.

B-ARBRE

Un B-arbre d'ordre d (nombre de descendants directs d'un nœud interne) et de profondeur p est défini comme une arborescence ayant les propriétés suivantes :

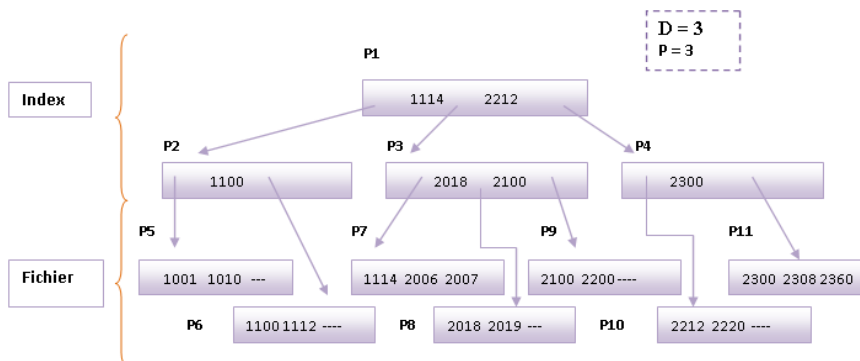
- ❑ Chaque nœud a au plus d fils, c'est-à-dire d pointeurs.
- ❑ Chaque nœud excepté la racine et les feuilles a au moins $\lceil d/2 \rceil$ fils (plus petit entier non inférieur à $d/2$, ou encore l'arrondi supérieur de $d/2$).
- ❑ La racine a au moins 2 fils.
- ❑ Toutes les feuilles apparaissent au même niveau : (p) ,
- ❑ Un nœud ayant k fils ($k \leq d$) c'est-à-dire k pointeurs, contient $k-1$ clés.
- ❑ Les données (tuples) sont rangées dans les feuilles ou nœuds terminaux. Les nœuds non terminaux ne contiennent que des clés et des pointeurs vers d'autres nœuds de l'arborescence.

B-ARBRE

Les types d'accès fournis par un B-index

Index Principal : l'ordre physique coïncide avec l'ordre logique : il est construit au chargement initial de la relation et est défini sur une clé.

Index Secondaire : ordre physique est différent de l'ordre logique dans la plupart des cas et peut être défini sur des attributs non clés



B-ARBRE

Algorithme de recherche

Soit la recherche d'un tuple de clé c

1. Lire la racine et rechercher quelle valeur de clé recouvre la valeur c .
2. Lire la valeur de pointeur associé.
3. Aller à la page pointée par le pointeur sélectionné.
4. Rechercher quelle valeur de clé recouvre c , lire la valeur de pointeur associé et aller à la page pointée.
5. Répéter l'opération précédente jusqu'à trouver la page feuille contenant la clé.

B-ARBRE

Algorithme d'insertion

Soit à insérer un tuple de clé c

Dans le cas où plusieurs ancêtres de P_0 sont pleins alors l'insertion aura pour effet de modifier l'arbre sur plusieurs niveaux et ainsi être dans la possibilité de modifier la racine qui deviendra un nœud intermédiaire, et qui sera remplacée par une nouvelle racine avec 2 fils. La profondeur de l'arbre sera alors modifiée.

On parle alors d'insertion avec éclatement de nœuds et propagation de l'éclatement jusqu'à la racine.

B-ARBRE

Algorithme de suppression

Soit à supprimer le tuple de clé c

- ❑ Appliquer l'algorithme de recherche de la clé c : soit P_i la page qui contient c
- ❑ Si P_i a un nombre d'enregistrement $\geq \lceil d/2 \rceil$ après suppression alors réaliser la suppression. Il faut cependant vérifier que la clé du tuple supprimé ne se retrouve pas comme clé dans l'index du B-arbre (vérifier tous les niveaux de l'index). Dans ce cas, il faut alors remonter dans la hiérarchie afin de remplacer cette clé par la clé du tuple suivant.
- ❑ Sinon (c'est-à-dire que P_i a moins de $\lceil d/2 \rceil$ fils après suppression) :
- ❑ On examine la page P_j immédiatement à gauche ou à droite de P_i et ayant le même père.
- ❑ Si P_j a plus de $\lceil d/2 \rceil$ enregistrements, on redistribue les enregistrements de P_i et P_j de manière équilibrée tout en conservant l'ordre. On répercute la modification sur les ancêtres de P_i puisque les clés sont modifiées.
- ❑ Sinon on réalise une suppression avec fusion. On regroupe P_i et P_j en un seul bloc et on modifie les ancêtres de P_i . Cette fusion peut être récursive.

GESTION DE L'INTÉGRITÉ