

Chapitre 04

Import & Export des données



- Création d'une copie de données ou Réplication de données de la BD,
- Sécuriser la Bd en cas de pannes(Récupérer la deuxième copie),
- Sécuriser les BDs qui sont sensibles à l'accès d'écriture ,
- Donc en pratique, l'administrateur doit à la fin du jour créer une 2eme copie de la BD, pour un objectif bien précis,



Exemple

le CCP utilise une BD(copie)pour un accès en lecture (voir le solde) ,par contre la BD en accès écriture n'est pas exposée au public (risque de changement de données),

2. Mécanisme Général

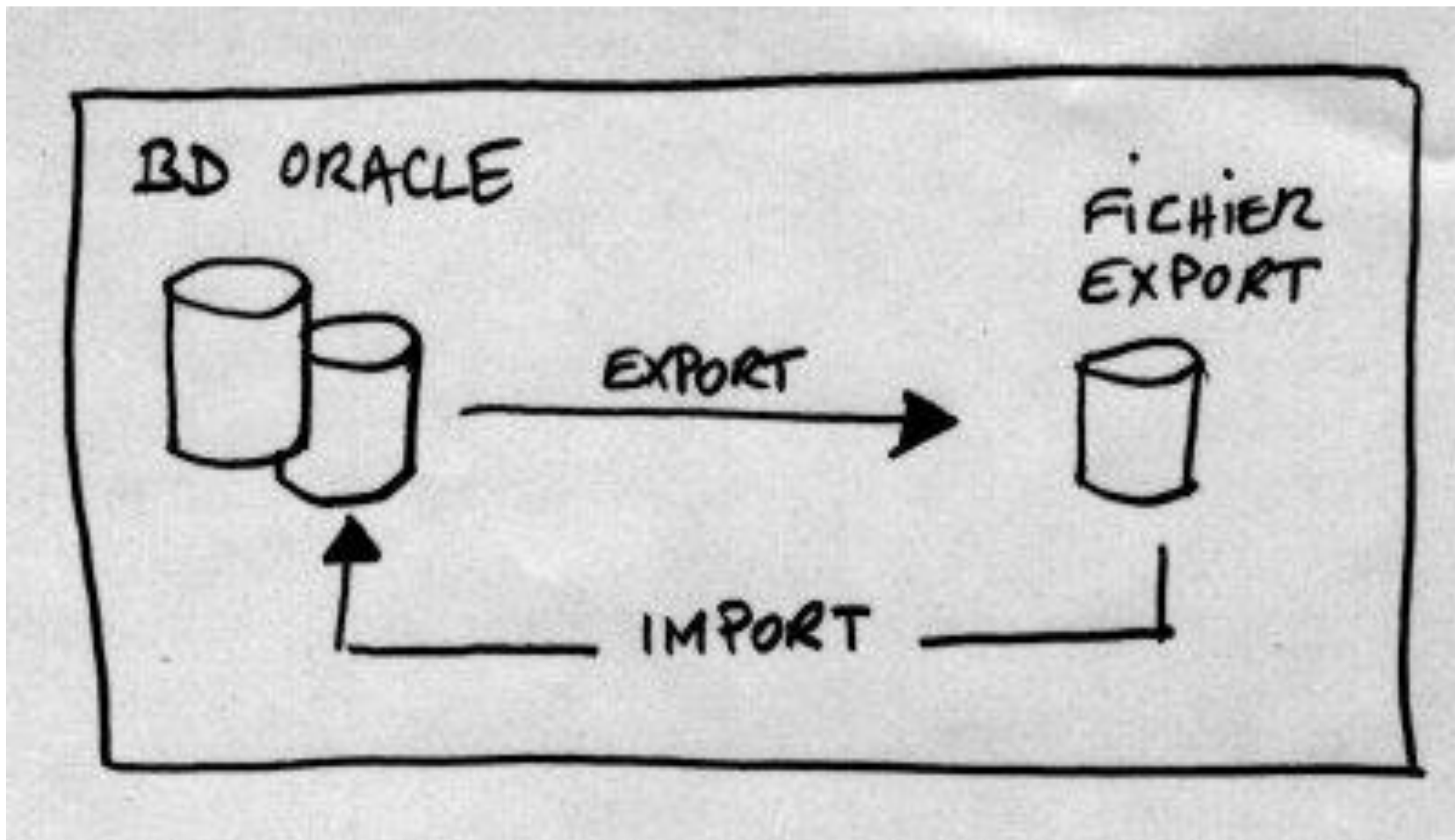


- Importer ou Exporter des données est un processus qui existe dans la majorité des SGBD.
- Import & Export: il peut être d'une BD principale vers des fichiers ,ou peut être d'une BD à une autre BD.



- Les termes Import & Export comme également une immigration de Données peut être :
 - **Globale (copie toute la BD)**
 - **Partielle(copie une partie de la BD)**

2. Mécanisme Général



3. Import & Export sous ORACLE



- **Data load/Unload Wizar:** pas de filtration des données,
- **Data Pump Export /Data Pump Import:** ne permet des échanges qu'entre des bases 11g.
- **SQL*Plus** : permet de créer, démarrer, arrêter, administrer et effectuer des opérations complexes de restauration sur les bases Oracle.
- **SQL*Loader** : un utilitaire très puissant de chargement des bases Oracle, à partir de fichiers contenant des données.



SQLPlus

**Crée-t-il les tables dans
la BD Oracle 11g?**



Etape 1: supprimer les Tables précédentes
=>Drop TABLE NomTable;

```
Drop TABLE NomTable1;  
Drop TABLE NomTable2;  
Drop TABLE NomTable3;  
  
.  
.  
.  
Drop TABLE NomTablen;
```



Etape 1 : supprimer les Tables précédentes
=> Drop TABLE NomTable;

Exemple :

```
Drop TABLE DETAILS;  
Drop TABLE PRODUITS;  
Drop TABLE COMMANDES;  
Drop TABLE FOURNISSEURS;
```



Etape 2: création des Tables

Cas 01

```
CREATE TABLE NomTable_i (  
  Champs1Table_i Type(Taille),  
  Champs2Table_i Type(Taille),  
  Champs3Table_i Type(Taille),  
  .  
  .  
  ChampsNTable_i Type(Taille),  
  Constraint PK_ NomTable1 PRIMARY KEY (Champs1Table_i);
```



Etape 2: création des Tables

Cas 01 Exemple

```
CREATE TABLE FOURNISSEURS(
```

```
NumFournisseur number(2),
```

```
NomFournisseur char(20),
```

```
Pays char(30),
```

```
Tel char(15),
```

```
Constraint PK_FOURNISSEURS PRIMARY KEY (NumFournisseur));
```



Etape 2: création des Tables

Cas 02

```
CREATE TABLE NomTable_i(  
  Champs1Table_j Type(Taille),  
  Champs2Table_j Type(Taille),  
  Champs3Table_j Type(Taille),  
  .  
  .  
  ChampsNTable_j Type(Taille),  
  Constraint PK_ NomTable_i PRIMARY KEY (Champs1Table_j),  
    Constraint NomTable_i _REF_ NomTable_i FOREIGN  
KEY(Champs1Table_i) REFERENCES NomTable_i(Champs1Table_i);
```



Etape 2: création des Tables

Cas 02 Exemple

```
CREATE TABLE PRODUITS(  
  NumProduit number(5),  
  NumFournisseur number(2),  
  NomProduit char(20),  
  TypeProduit char(10),  
  PrixUnit int,  
  Constraint PK_PRODUITS PRIMARY KEY(NumProduit),  
  Constraint PRODUITS_REF_FOURNISSEURS FOREIGN  
KEY(NumFournisseur) REFERENCES Fournisseurs(NumFournisseur));
```

3. SQL*Plus



SQLPlus crée-t-il la table dans la base Oracle 11g?

- Ouvrir le SQLPlus utilisant le compte system et le mot de passe system.





SQLPlus crée-t-il la table dans la base Oracle 11g?

- Il est possible d'exécuter un fichier SQL directement depuis la ligne de commande de SQL plus ou en ligne de commande, en exécutant la commande suivante:

SQL> start create.sql;



Quelle utilisation pour SQL*Loader ?

- SQL*Loader est un utilitaire servant à charger des données provenant de fichiers externes, dans une base Oracle 11g.
- Il permet d'intégrer de nombreux formats de données, à partir d'un ou de plusieurs fichiers, dans une ou plusieurs tables.



SQL*Loader crée-t-il la table dans la base Oracle 11g?

■ **Non**

La table doit être créée
avant le chargement
des données.



Comment fonctionne SQL*Loader ?

- Pour piloter le chargement des données, SQL*Loader utilise un fichier de contrôle.
- Ce dernier détermine les fichiers de données à charger et leur table de destination.



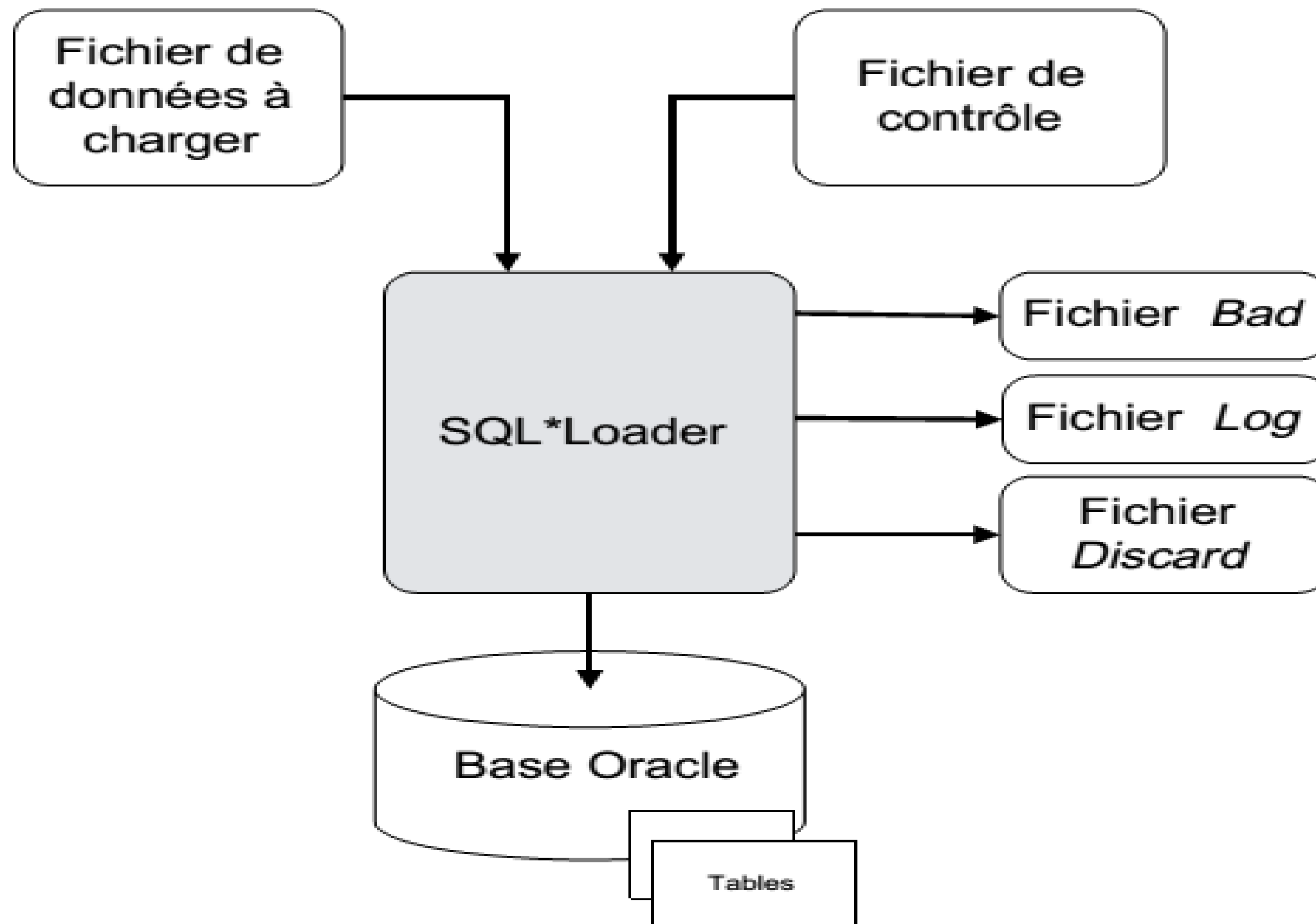
Comment fonctionne SQL*Loader ?

- Durant le chargement, SQL*Loader peut produire trois types de fichiers :
 - • **le fichier log**, Le fichier LOG FILE enregistre les activités SQL Loader durant un Loading ;
 - • **le fichier bad**, qui conserve les enregistrements qui n'ont pu être insérés (rejetés lors de l'insertion dans la table cible) ;
 - • **le fichier discard**, Les enregistrements qui ne répondent pas aux critères spécifiés dans le Control File sont jetés, écartés et sont écrits dans le fichier DISCARD FILE (provenant du fichier de données à charger, mais non reconnus par SQL*Loader).

4. SQL*Loader ?



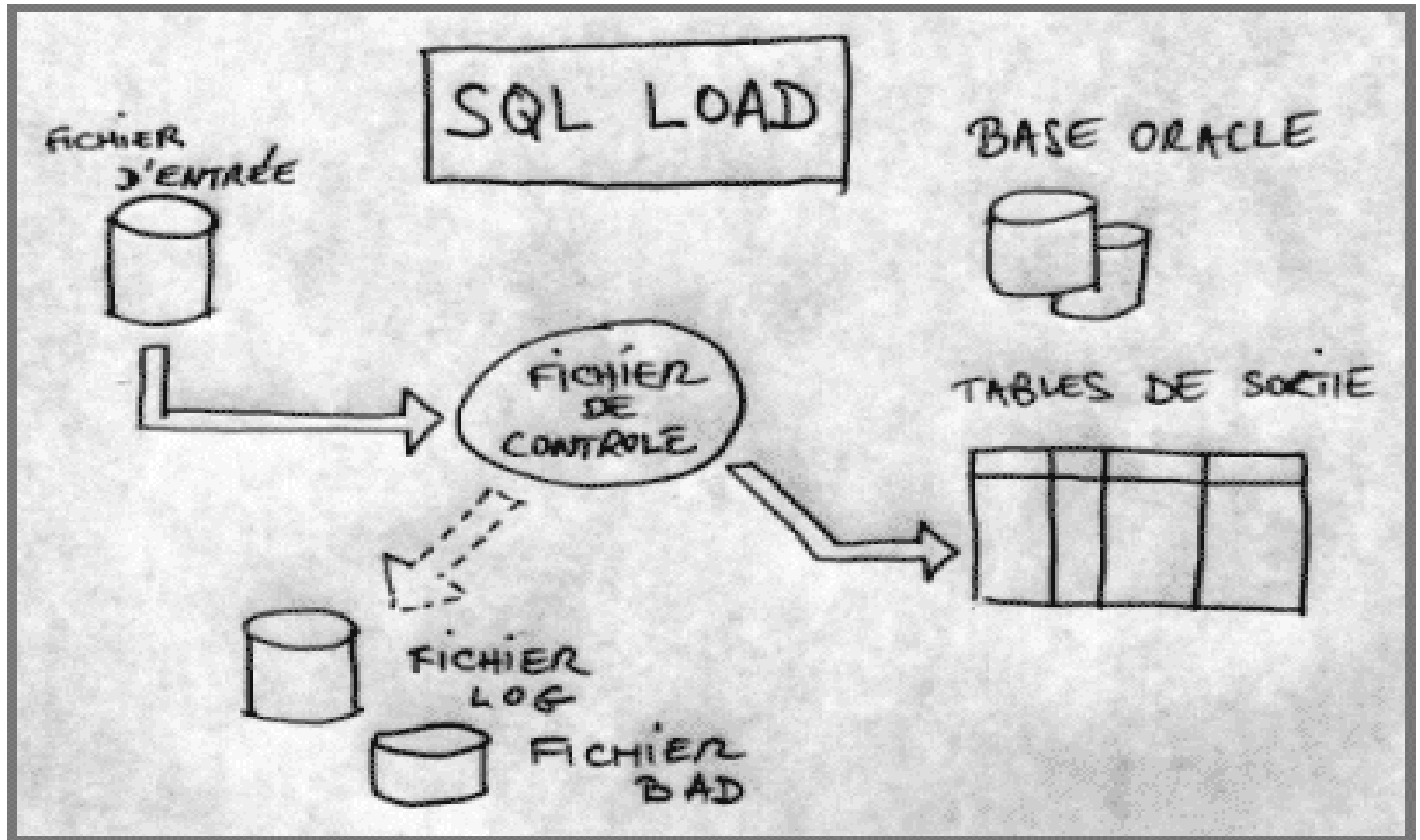
Principe de fonctionnement de SQL*Loader



4. SQL*Loader ?



Principe de fonctionnement de SQL*Loader



4. SQL*Loader ?



Fichier de Contrôle

Partie 1

LOAD DATA

INFILE *

APPEND

INTO TABLE FOURNISSEURS

FIELDS TERMINATED BY ";" OPTIONALLY ENCLOSED BY ""

(NumFournisseur, NomFournisseur, Pays, Tel)

BEGINDATA



Importation à partir de fichiers de données

Principe :

- données récupérées sous la forme d'un ensemble de fichiers ASCII / Unicode (données non structurées)
- Données externes ordonnées suivant un format donnée
- Exemple format CSV : suite de champs de valeurs séparés par un
- point virgule
- Goudjil; Lakhdar; Enseignant

4. SQL*Loader ?



Fichier de données à charger

Partie 2

```
1;"Exotic";"Allemagne";"1019168842"  
2;"New";"Argentine";"1040635384"  
3;"Grandma";"Autriche";"1409076845"  
4;"Tokyo";"Belgique";"1840982385"  
5;"Cooperativa";"Brisil";"1872198482"  
6;"Mayumi's";"Canada";"1874726623"  
7;"Pavlova,";"Danemark";"1965245844"
```

4. SQL*Loader ?



Comment utiliser SQL*Loader ?

- Sous Windows, il se nomme sqlldr et se situe dans
ORACLE_HOME\bin. Pour accéder à l'aide en ligne :
- **C:\> sqlldr**
- La commande usuelle se présente comme suit :
- **C:\> sqlldr control=tableControle.ctl**



Comment utiliser SQL*Loader ?

- Lancer le fenêtre de commande : cmd (D:\ emplacement des tables)
- `sqlldr control=D:\ table1.ctl`
- `sqlldr control=D:\ table2.ctl`
- `sqlldr control=D:\ table3.ctl`
- `sqlldr control=D:\ table4.ctl`
- Faire attention à l'ordre de création des tables puisqu'il est important.
(Règle des clés étrangères)

Chapitre 05

Les index

Qu'est ce qu'un index ?

Un index est une structure de données redondante organisée de manière à accélérer certaines recherches.

Intérêt de l'index

- L'accès aux lignes des tables se fait séquentiellement, c'est à dire que pour trouver une information précise, il faut parcourir une à une et l'une après l'autre les lignes de la table. Il faudra aussi parcourir toute la table.
- **Par exemple** pour chercher **Ahmed Ali** dans une table d'individus, il faut lire toutes les lignes, car il peut y en avoir plusieurs dans la table.

index créés automatiquement

- Les SGBDR crée systématiquement un index chaque fois que l'on pose une clef primaire (PRIMARY KEY) ou une contrainte d'unicité (UNIQUE) sur une table.
- En revanche, il n'y a pas d'index créé automatiquement par le SGBDR derrière une FOREIGN KEY (clef étrangère).

- Certains SGBD (en principe non relationnels) créé des index systématiquement sur toutes les colonnes. C'est le cas de Sybase IQ,

■ Index Bitmap

- Les index Bitmap sont destinés à l'indexation de colonnes qui comportent peu de valeurs distinctes et beaucoup d'enregistrements pour chacune de ces valeurs : donc lorsque le rapport D tend vers 0.
- A l'inverse des index B-Tree, les index Bitmap ne stockent pas un pointeur vers un enregistrement dans un fichier trié sur l'index, mais une valeur codée sur un bit (vrai ou faux) pour chaque valeur de la colonne indexée (2 bits pour une cardinalité 2, 3 pour une cardinalité 3, etc.) dans un fichier trié sur la clé.

■ Index Bitmap

- De tels index optimisent la recherche relative à une question du type :
"l'enregistrement appartient-il à un sous ensemble du domaine sur l'index ?".
- Par exemple pour savoir si un enregistrement est de genre "Femme", il suffit de consulter un bit au lieu de comparer des chaînes.

■ Index Bitmap

Table Personne

pk	prénom	genre	métier
1	Daniel	Homme	Ingénieur
2	Danielle	Femme	Ingénieur
3	Dany	Femme	Commercial
...			

BITMAP INDEX ON Personne(genre);

pk	Femme	Homme
1	0	1
2	1	0
3	1	0
...		

BITMAP INDEX ON Personne(métier);

pk	Ingénieur	Commercial	Technicien
1	1	0	0
2	1	0	0
3	0	1	0
...			

BITMAP INDEX ON Personne(genre);

pk	Femme	Homme
1	0	1
2	1	0
3	1	0
...		

*Ingénieur
AND Femme*

0
1
0
...

■ Index B-Tree

- Tout d'abord, un index B-tree est un index stocké sous forme d'arbre (tree), cet arbre est ordonné et équilibré (profondeur homogène quelque soit la feuille) ; il peut avoir 2, 3 voire plus de niveaux (profondeur de branche).
- La recherche consiste en la détermination de l'intervalle pour connaître le niveau inférieur et ainsi de suite jusqu'à atteindre la feuille qui stockera, près de sa valeur, le rowid du ou des tuple(s) recherché(s).

Inconvénients des index



- Les index diminuent les performances en mise à jour (puisque'il faut mettre à jour les index en même temps que les données).
- Les index ajoutent du volume à la base de données et leur volume peut devenir non négligeable.

Chapitre 06

Optimisation et Evaluation des Requêtes

- Comment répondre à une requête dans temps minimal ?
- Cas 1: pour une requête de **Selection** simple ,utiliser les index,
- Cas 2: pour une requête de jointure (produit cartésien) composée d'autres requetes en plus => problème ici, l'idée d'index n'est pas faisable,

Facteurs d'optimisations

- Les facteurs influant sur le temps de réponse d'une requête sont:
 - Temps d'accès au mémoire secondaire(disque Dur),
 - Temps d'accès au mémoire Centrale (RAM),
 - Temps de traitement d'une requête par le processeur,
 - Taille du bloc utilisé par la RAM,

- Taille du bloc est fixé toujours pour nous =15000

Octets,

- Le facteur le plus influant est :accès au mémoire

secondaire,

Objectif principale

- Minimiser le nombre d'accès au mémoire secondaire,
- donc: minimiser le chargement en lecture/écriture des blocs)

Exemple

- CLIENTS (NumClient (5), NomClient (20), Pays (30) , Tel (15))
- COMMANDES (NumCommande (5), #NumClient (5), FraisPortnumber(4),AnCom (4))
- VENTES (#NumCommande (5), #NumProduit (5),Qte (5), Remise (5))
- PRODUITS (NumProduit (5), NomProduit (20), TypeProduit (10), PrixUnit (5))

Exemple

- **Donner les produits qui ont été vendus en 2000.**

SELECT DISTINCT NomProduit } Projection **P**

FROM PRODUITS p, VENTES v, COMMANDES c } Tables utilisées

WHERE c.AnCom = 2000 } Sélection

AND p.NumProduit = v.NumProduit } jointure

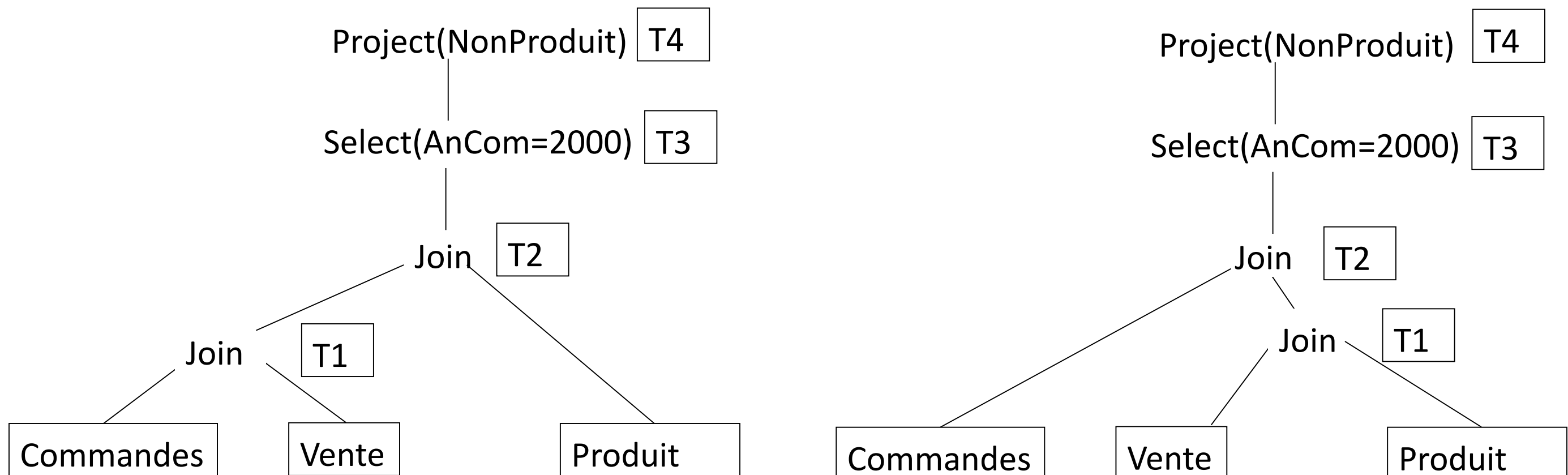
AND v.NumCommande=c.NumCommande; } jointure

Arbre algébrique et plan d'exécution

- Une requête composée possède plusieurs arbres algébriques

=> plusieurs plans d'exécutions

Exemple: la jointure est commutative => 2 arbres pour la requête précédente,

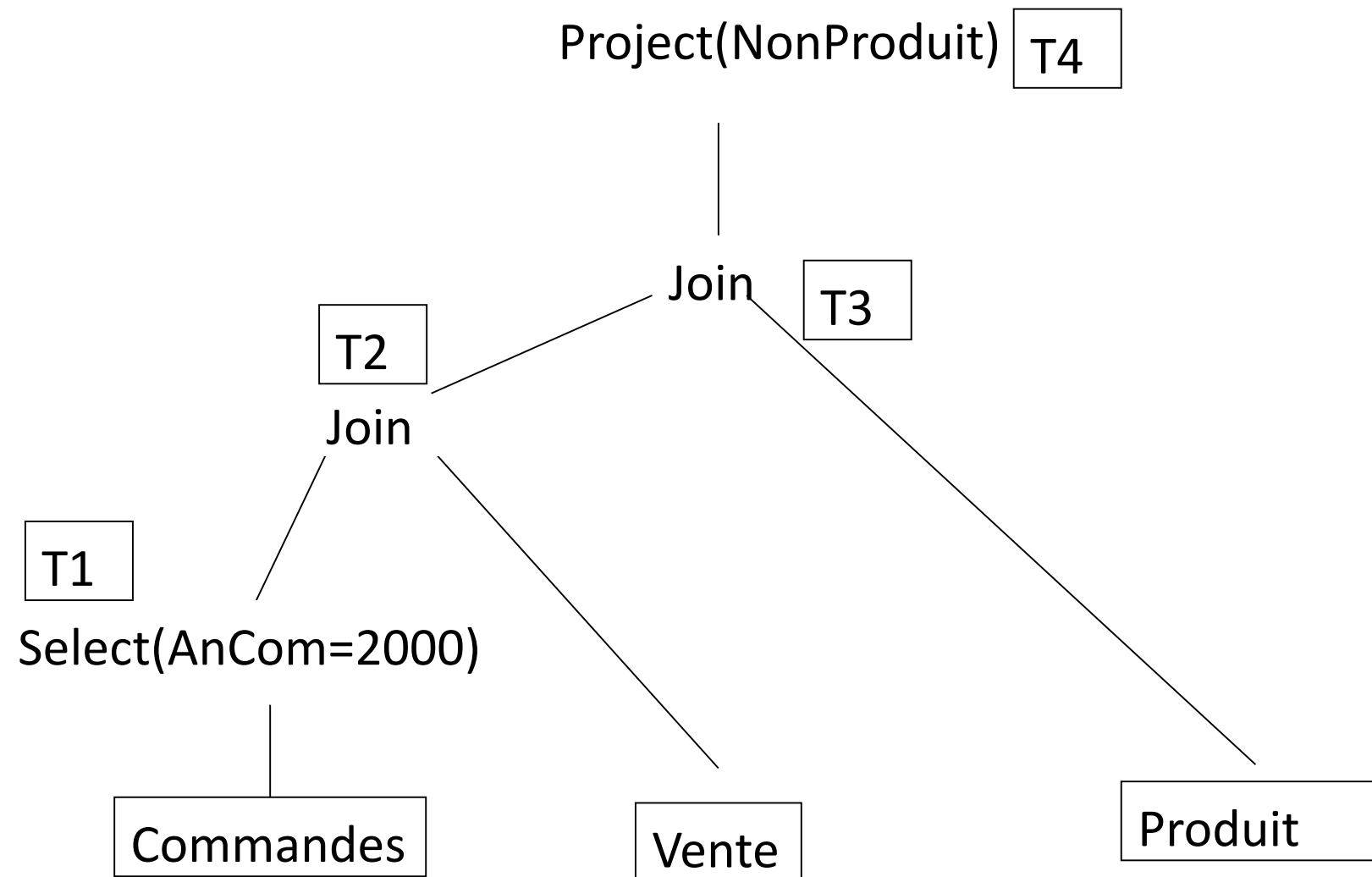


- => 2 plan d'exécutions(T1,T2,T3,T4)

Règles d'optimisation de l'arbre algébrique

1. Descendre la sélection le plus possible,
2. Jointer d'abord les tables réduits par la sélection,
3. Ascendre la projection le plus possible,

Arbre algébrique Optimale



Calcule le coût d'une arbre Algébrique

Table	Taille d'un tuple	Nbr de tuples	Tuples/block	Scan
COMMANDES				
VENTES				
PRODUITS				
T1				
T2				
T3				
T4				

Calcule le coût d'une requête

- **Taille de tuple** :représente la largeur de la table en octets,
- Exemple table COMMANDES (NumCommande (5), #NumClient (5), FraisPortnumber(4),AnCom (4))
 $\Rightarrow 5+5+4+4=18$ Octets,
- **Nbre de tuples**:c'est le nombre de lignes qui existe dans la tables,
- Exemple:en TP la Table Commandes contient 1078 lignes,
- Tuples par Bloc:On suppose que la taille d'un bloc est toujours fixe,une bloc a la taille de 15000 octets \Rightarrow on cherche le nbre de tuples que le bloc peut contenir ,calculer comme
suit:Nbre_tuples_par_Bloc=Taille de bloc/Taille de tuple
- **Exemple:Nbre_tuples_commandes_par_Bloc=15000/18=833**

Calcule le coût d'une arbre Algébrique

- **Ful Scan:** c'est le nombre de chargement en Bloc pour lire ou écrire toute la table,
- **Ful_Scan=Nbre_Tuples_Table/Tuples_par_Bloc**
- **Exemple:** Ful_Scan (table commande)=1078/833=2

Calcule le coût d'une arbre Algébrique

Table	Taille d'un tuple	Nbr de tuples	Tuples/block	Scan
COMMANDES	18	1078	833	2
VENTES				
PRODUITS				
T1				
T2				
T3				
T4				

Coût global d'un arbre Algébrique

$\text{Cout_calcul}(T1) = \text{cout_lire}(\text{Commandes}) + \text{cout_ecrire}(T1)$

$\text{Cout_calcul}(T2) = \text{Cout_lire}(T1) + \text{cout_lire}(\text{Ventes}) + \text{cout_ecrire}(T2)$

$\text{Cout_calcul}(T3) = \text{Cout_lire}(T2) + \text{cout_lire}(\text{Produits}) + \text{cout_ecrire}(T3)$

$\text{Cout_calcul}(T4) = \text{Cout_lire}(T3) + \textbf{cout_lire(Produits)} + \text{cout_ecrire}(T4)$

$\text{Cout Globale} = \sum \text{ des 4 couts précédents } + \text{Cout_affichage}(T4)$