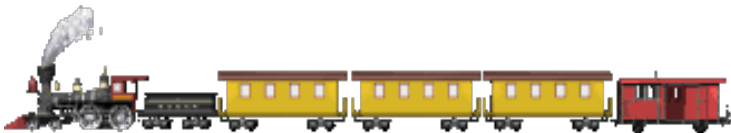


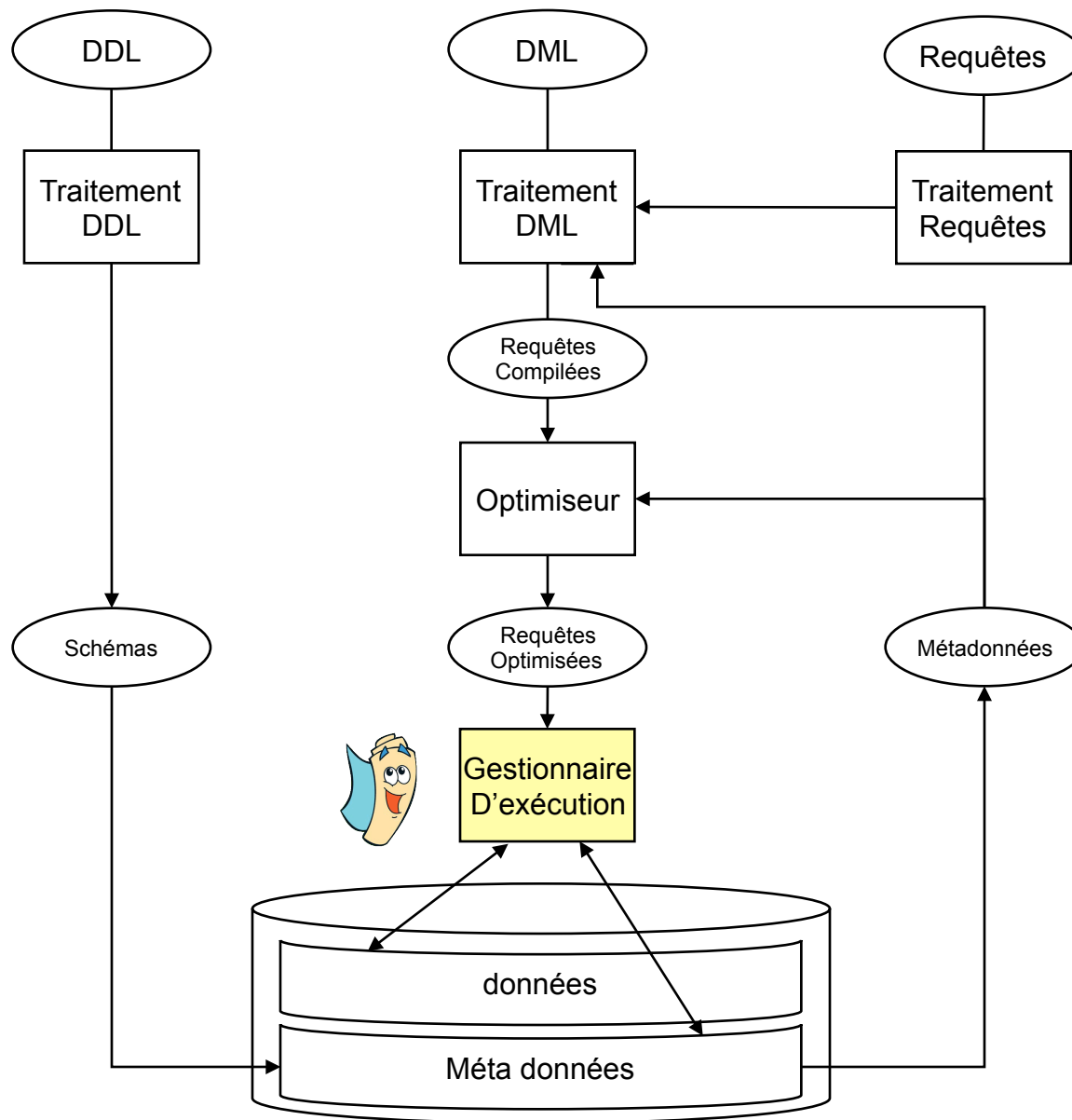


Principe des TP

Fake Database Model



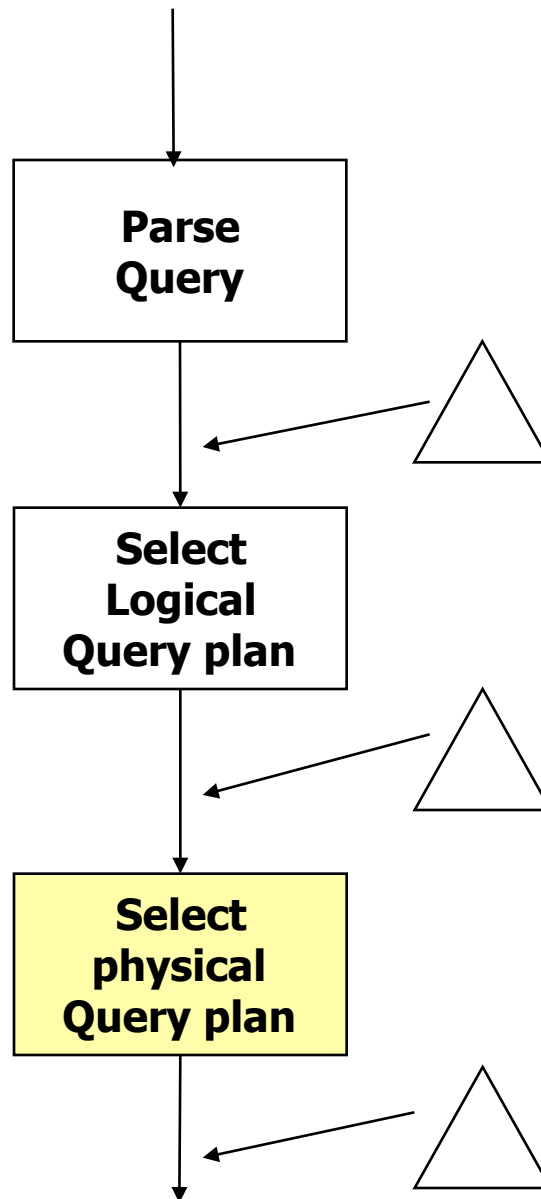
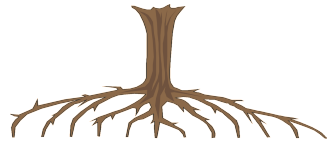
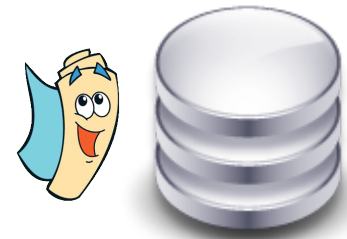
Traduction en cours



D'après C.J DATE

DDL : langage de définition des données; DML : langage de manipulation des données

From Ullman



Query expression
tree

Logical Query
Plan tree

Physical Query
Plan tree

Du modèle au code



Modèle

...
...
...
...
...
...
...

Algèbre

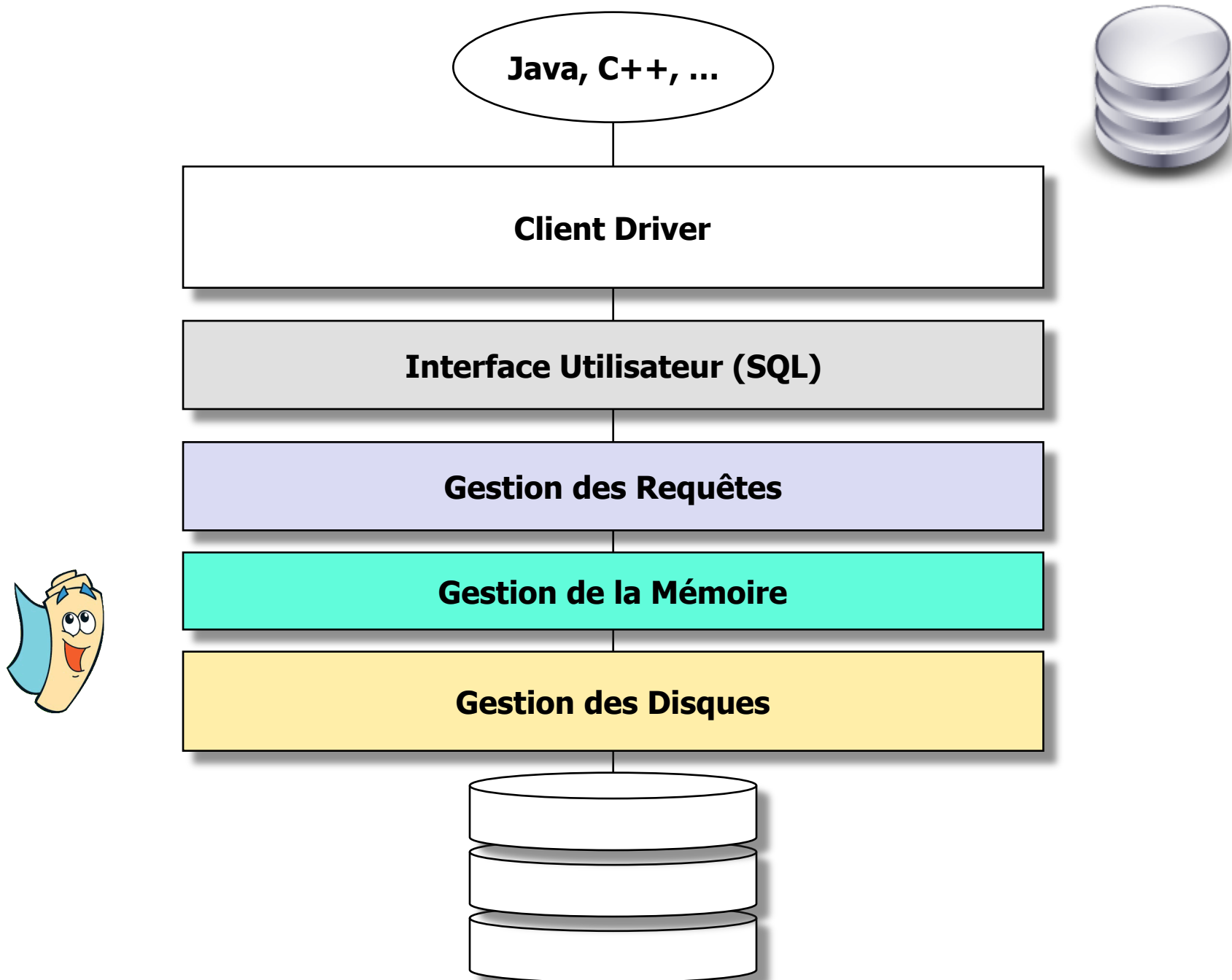
$\sigma_{\text{owner1}=\text{owner2}} (\text{Cats} \otimes \text{Dogs}) = \text{Cat} \bowtie \text{Dogs}$



Logiciel



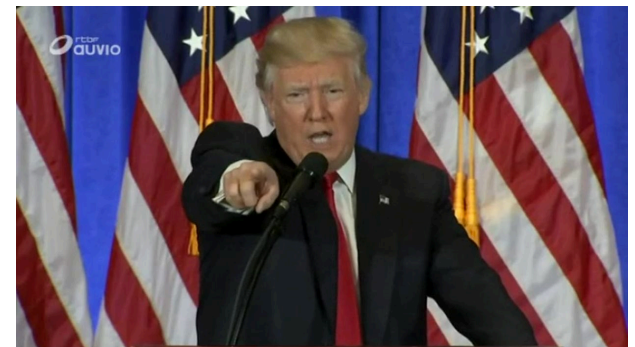
Java, C++, ..



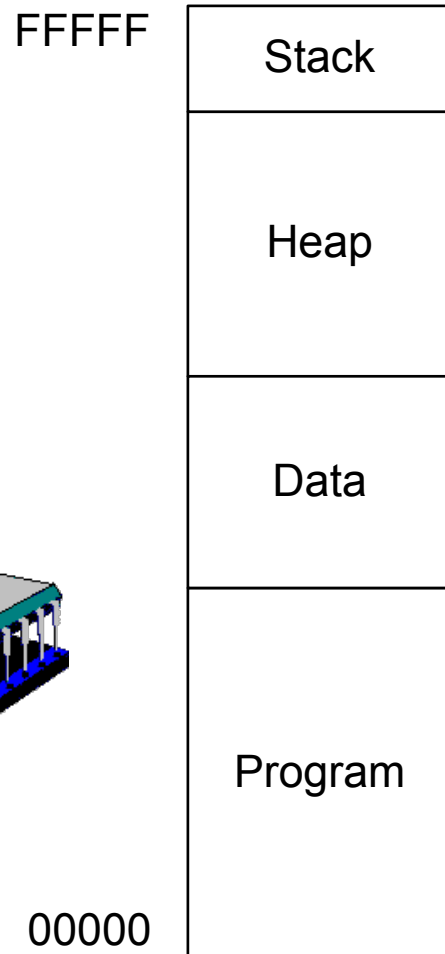
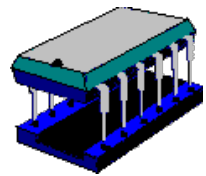
Simplification



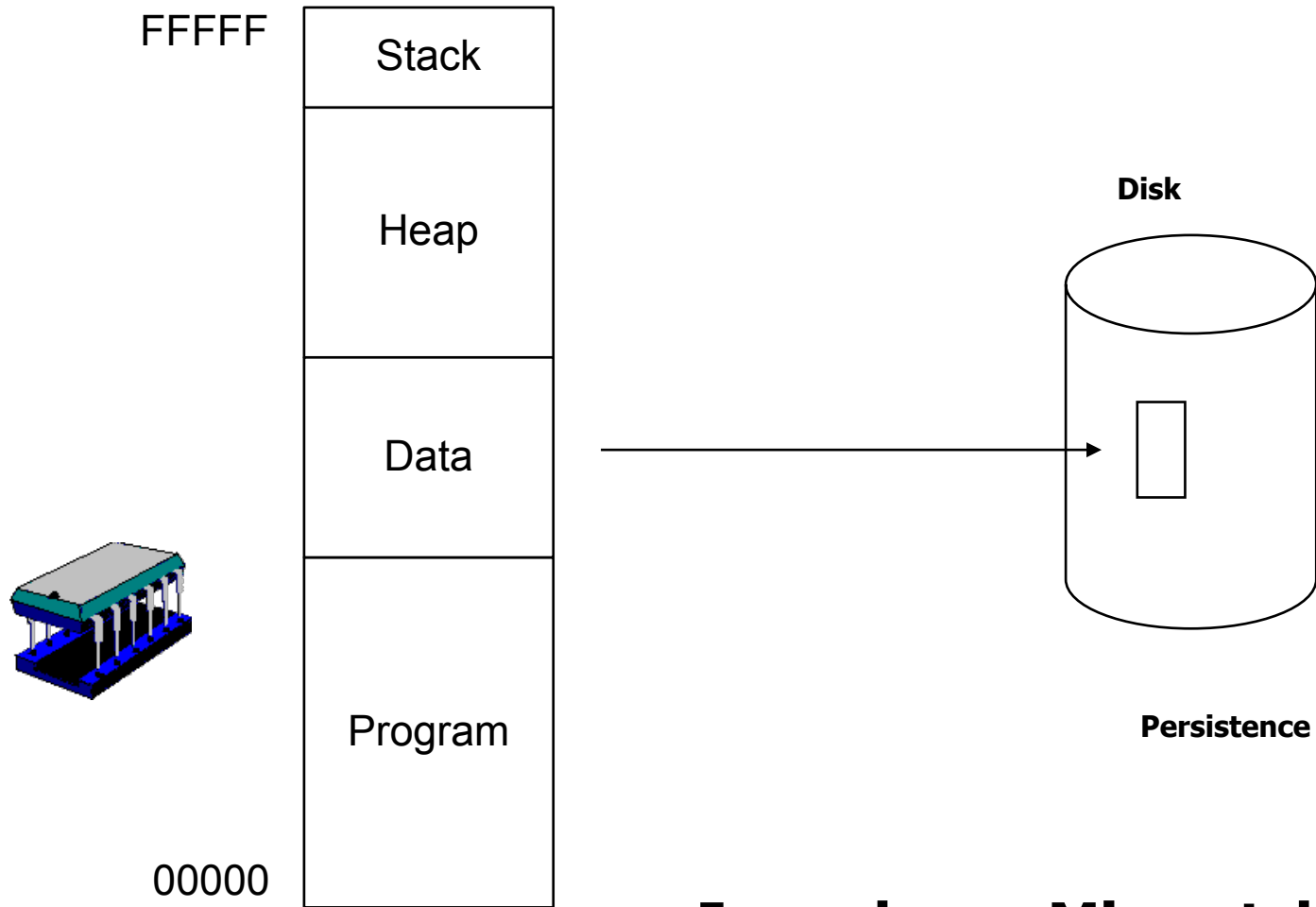
- Opération sur les données.
- Opération la plus utilisée : "select"
- Recherche d'éléments sur un critère de choix.
- Techniques utilisées dans toutes les applications de traitement de données.
- Opération réalisées à base d'index.
- Select = jointure, une ou deux tables.



Data in program



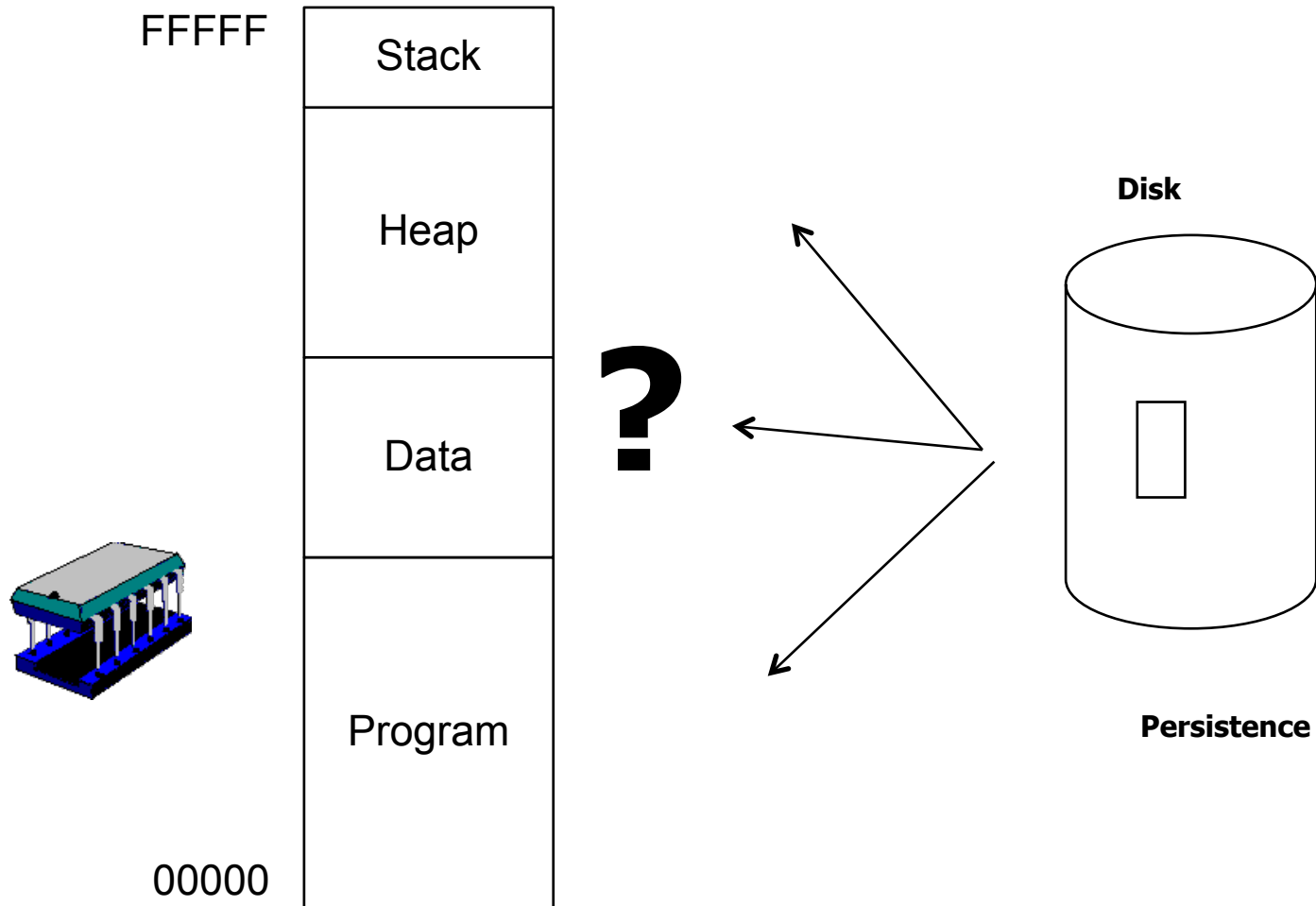
Data in program and on disk



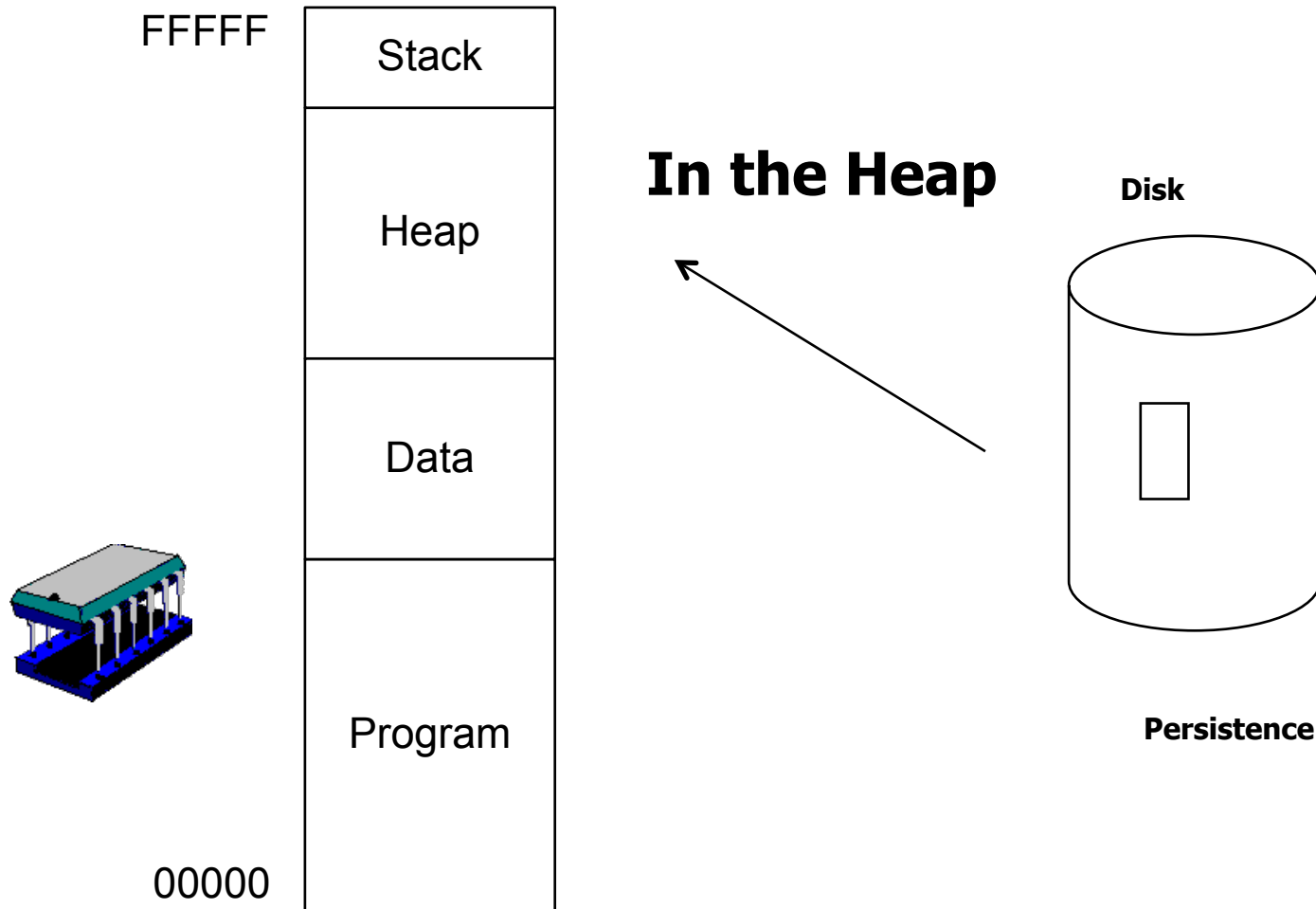
Impedance Mismatch



In which Region are data base data loaded ?



In which Region are data base data loaded ?



Relations sur disque



Relation

II
N
M
Y
K
OO
E
Z
U
NN
X
V
D
I
GG
S
P
JJ
PP
Q
CC
W
NN
DD
EE
R
T
HH
O
A
BB
C
H
QQ
F
J
AA
FF
B
MM
G
KK
LL
L

**Segments
De relation**

Block 2

X
V
D
I
GG
S
P
JJ
PP
Q

Block 3

CC
W
NN
DD
EE
R
T
HH
O
A

Block 1

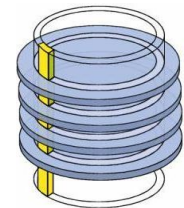
II
N
M
Y
K
OO
E
Z
U
NN

Block 4

BB
C
H
QQ
F
J
AA
FF
B
MM

Block 5

G
KK
LL
L



Analyse et compréhension du problème



- 1) cas minimum : jointure en mémoire
 - Les relations tiennent en mémoire
 - Pas de segmentation
 - Pas de gestion des partitions sur disque
- 2) cas complexe : jointure sur disque
 - Les relations ne tiennent pas en mémoire
 - Les relations sont stockées dans des blocks disques
 - Gestion des partitions sur disque
 - Gestion des buffers mémoire

Jointures Physiques



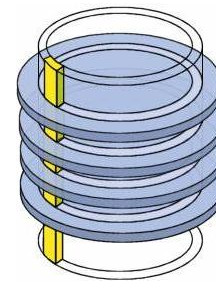
■ Jointure en mémoire

- Nested Loop
- Merge join
- Hash join



■ Jointure sur disque

- Nested Loop
- Sort Merge Join
- Hash join

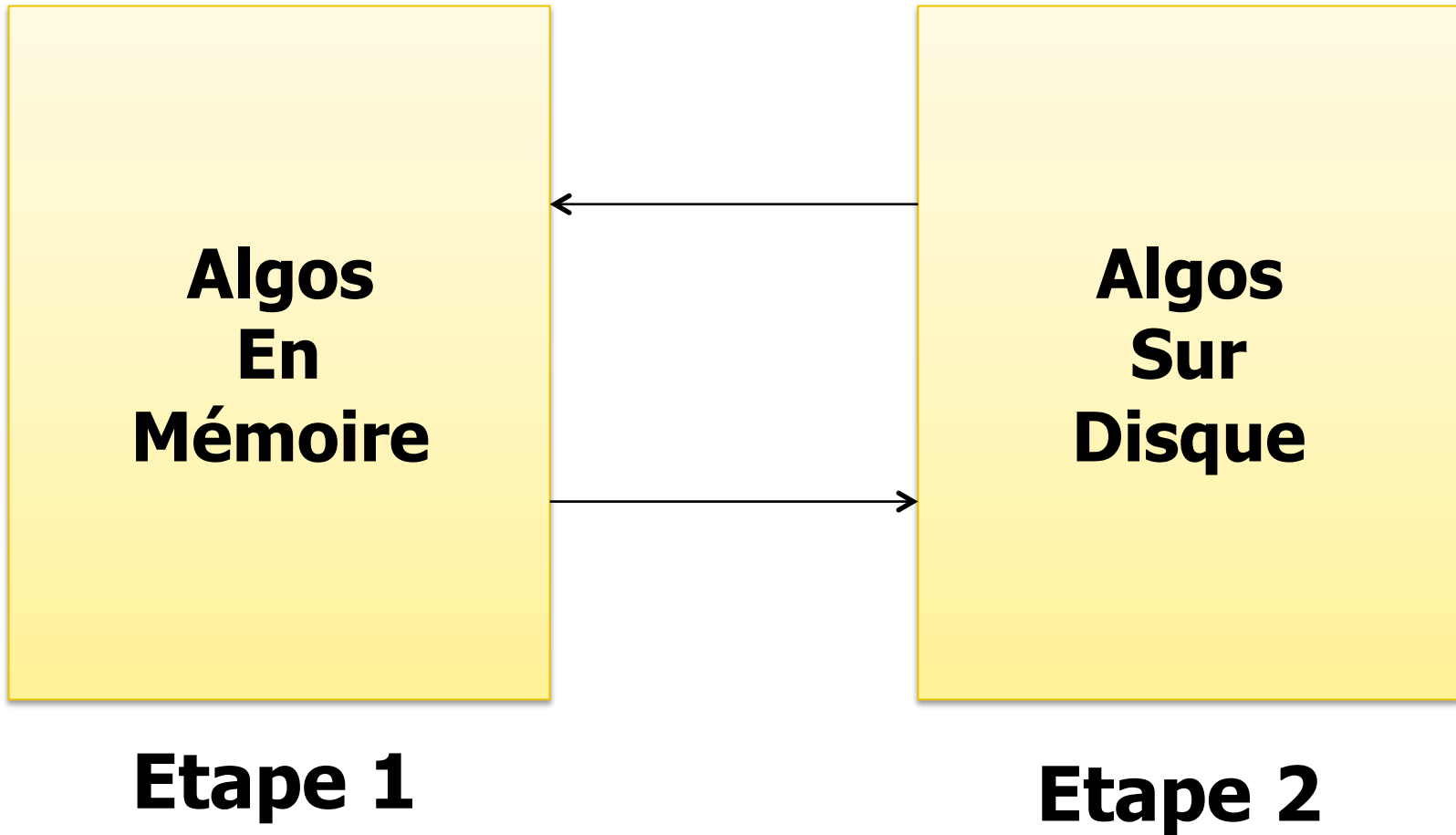


Principes de conception



Jointures en mémoire

Jointures sur disque



Principes de conception



- Objectifs : Les algos sur disque utilisent les algo en mémoire.
- Étape 1 :
 - Simplification des algos en mémoire
 - Ils seront adaptés plus tard pour être utilisés par les algos sur disque.
- Etape 2 :
 - Les algos sur disque rajoutent la gestion des blocs disque et pages mémoire (buffer).

Modèle de contrainte mémoire et disque



- Les tables sont rangées sur disque dans des fichiers constitués de blocks disque.
- Un block disque contient 10 éléments
 - C'est une simplification, on pourrait tout aussi bien prendre une taille de 20,...
- Les blocks disques sont modélisés sous la forme de fichiers texte de 10 éléments.

Principes de simplification des algos en mémoire

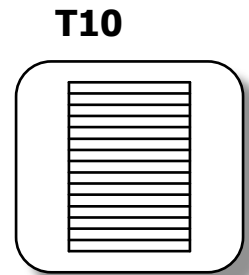


- Chaque relation tient dans un seul fichier lu en une seule fois.
- Une seule colonne par relation (index)
- Pas de doublons, sauf cas particulier.
- Le résultat de la jointure tient dans un seul fichier écrit en une fois.
- Taille des relations fixée à 10 lignes pour simplifier les tests.
- Seule contrainte : utilisation de tableaux de taille 10.

Modèle de contrainte mémoire et disque



- Nous disposons de 3 pages mémoires de 10 éléments.
- Modélisées pour les besoins du TP sous la forme de tableaux de 10 éléments.



Heap Size : 30 bytes

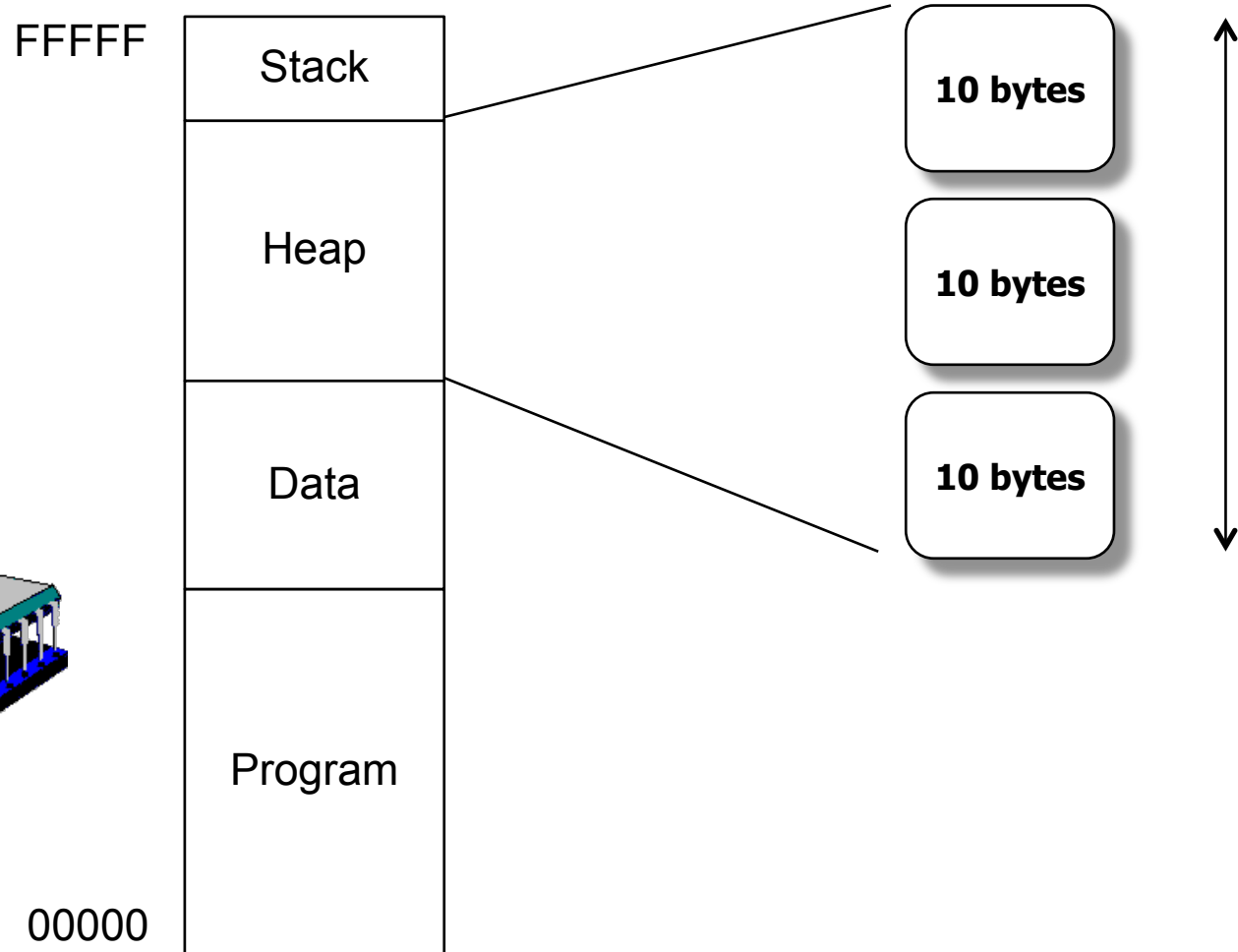
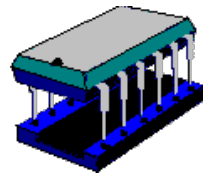


10 bytes

10 bytes

10 bytes

Data in program



Principes de simplification des algos en mémoire en Java



- Pas de collections Java
- Que des tableaux de char
 - `char[] r =`
 - `char[] s =`
- Pas de String
 - `String[] r =`
 - `String[] s =`
- Pas de Char, rien qui commence avec une majuscule.

Principes de simplification des algos en mémoire en C



- Pas de lib spécifiques non contrôlées
- Que des tableaux de char
 - `char r [10];`
 - `char s [10];`
- Ce sont des tables pas des string

A
Z
G
J
U
K
E
B
V
D

A	Z	G	J	U	K	E	B	V	D	null
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-------------

Principes de simplification des algos en mémoire

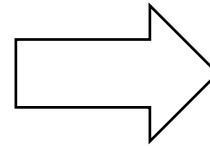


- Tables => Mono Column => Index
- Index :
 - Files
 - Database
 - Big Data

Principes de simplification des algos en mémoire



A	B	C
a	g	v
v	e	g
b	y	e
y	r	h
i	d	j
k	s	l
l	f	m



A
a
v
b
y
i
k
l

Une table

On s'implifie

Tables => Mono Column => Index



Relation R

Attribut A

A diagram of a table with 15 rows and 5 columns. The third column from the left is highlighted in yellow. An arrow points from the label 'Attribut A' to this highlighted column.

**Index on
Attribut A**

A
Z
G
J
U
K
E
B
V
D

Tableaux de char (pas Char)



R

char

A
Z
G
J
U
K
E
B
V
D

S

char

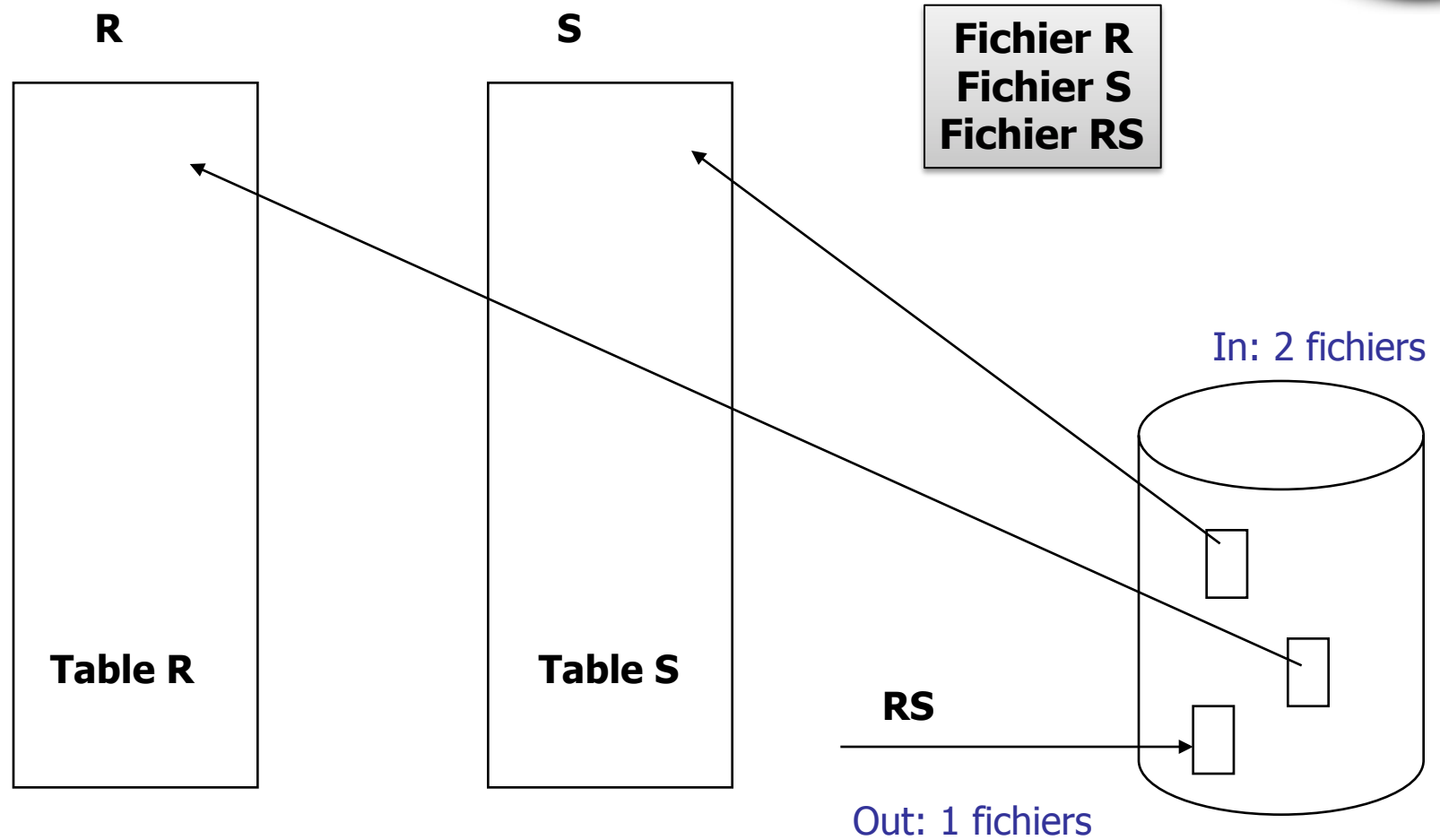
B
U
Z
K
X
V
N
L
M
E

Table R

Table S

$RS = \text{join}(R, S)$

Une table égale un fichier sur disque



Memory Nested Loop Join

