

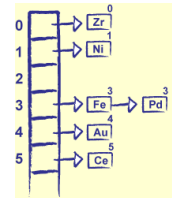
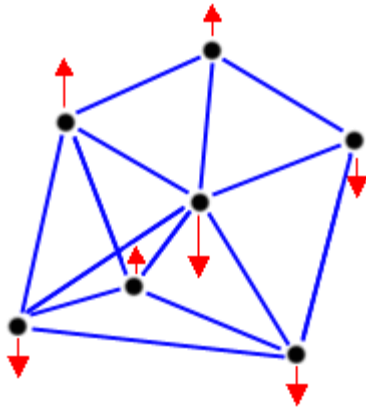
井

#



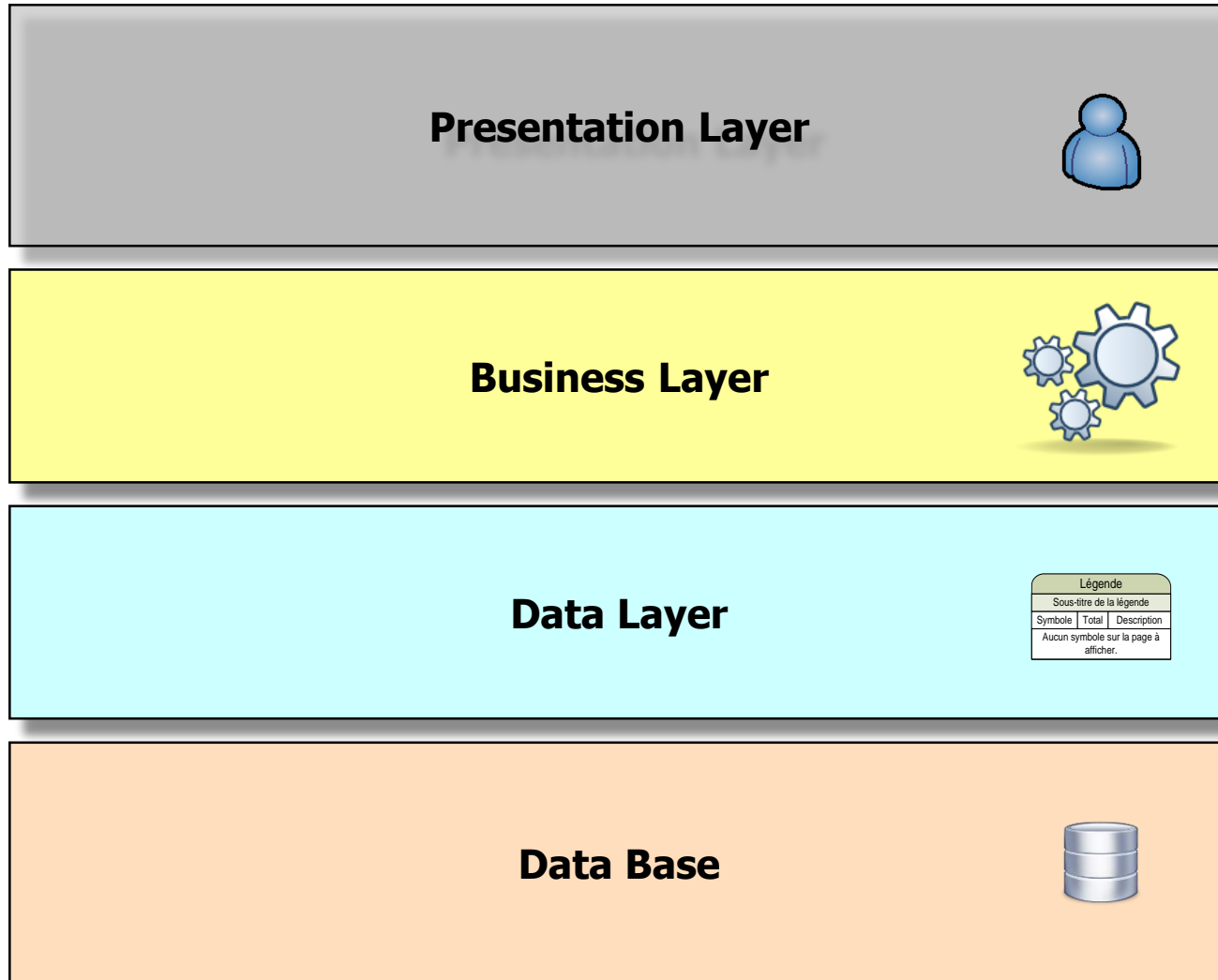
Conception Avancée de Bases de Données

Hash Selection



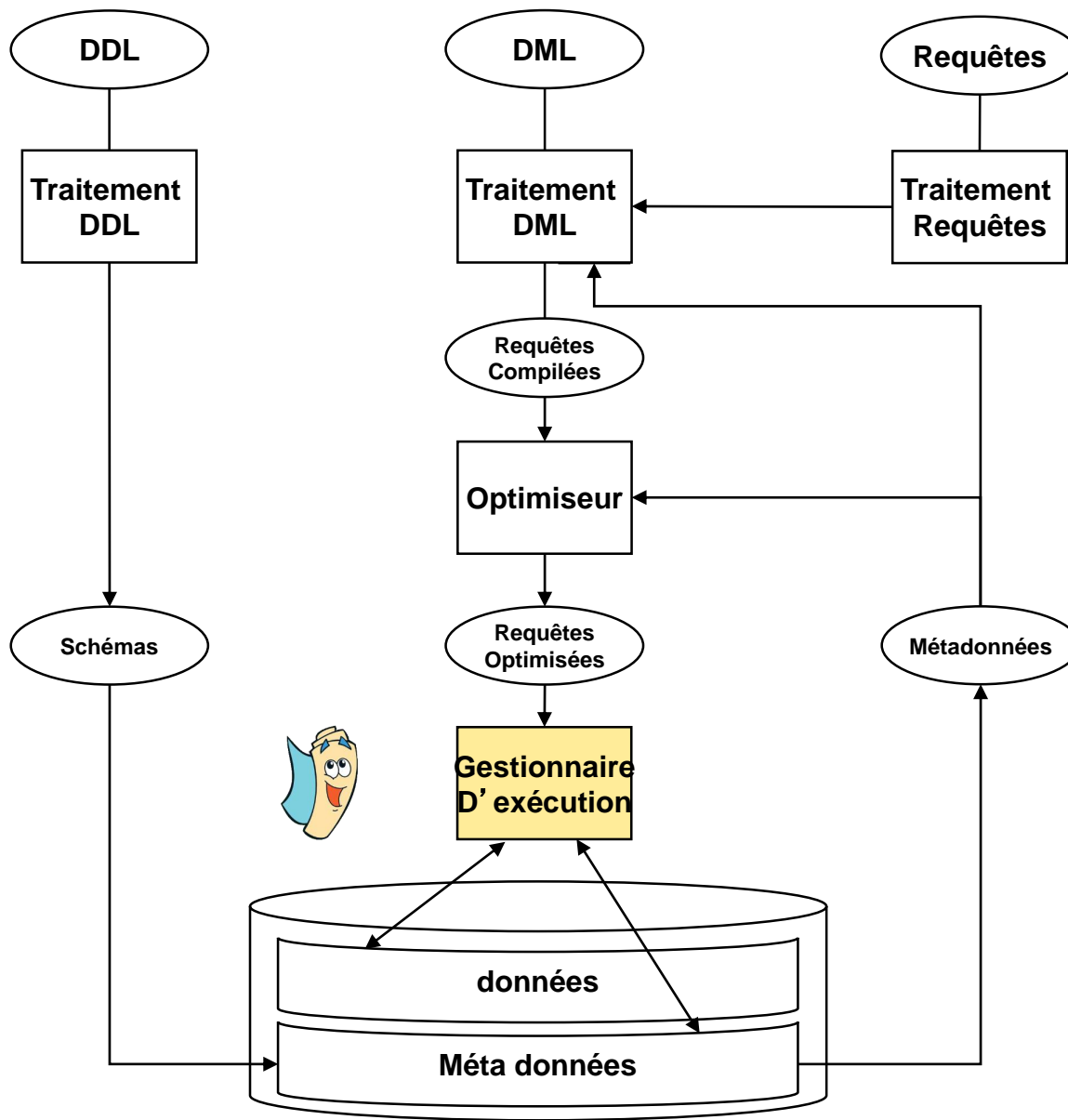
Traduction en cours

Layered Architecture



Big Picture

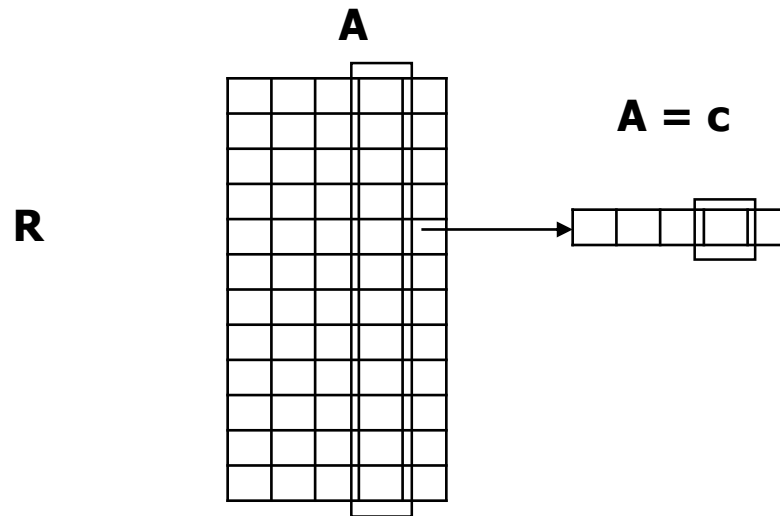
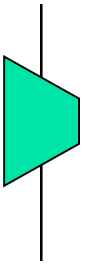




D'après C.J DATE

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

Sélection d'un Tuples : Attribut A = 'c'

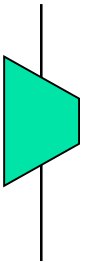
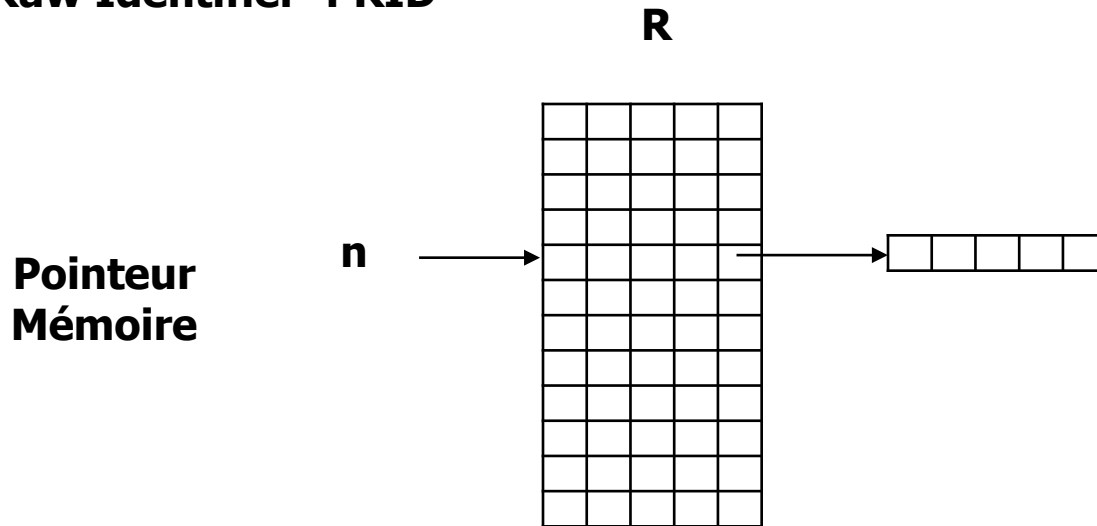


Accès par adresse : RID

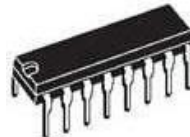


Hypothèse : On connaît l'adresse physique de la ligne

Raw Identifier : RID



**Tableau
Cases Mémoires**



Recherche dans un tableau



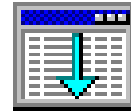
- Intuitivement, la recherche d'une chaîne de caractère dans un tableau nécessite la lecture de chacune des cases du tableau et la comparaison avec la chaîne de référence.

Position du problème



- Pour rechercher dans une table un Tuple ayant un attribut A tel que $A = c$

- Solution 1 : Seq Scan



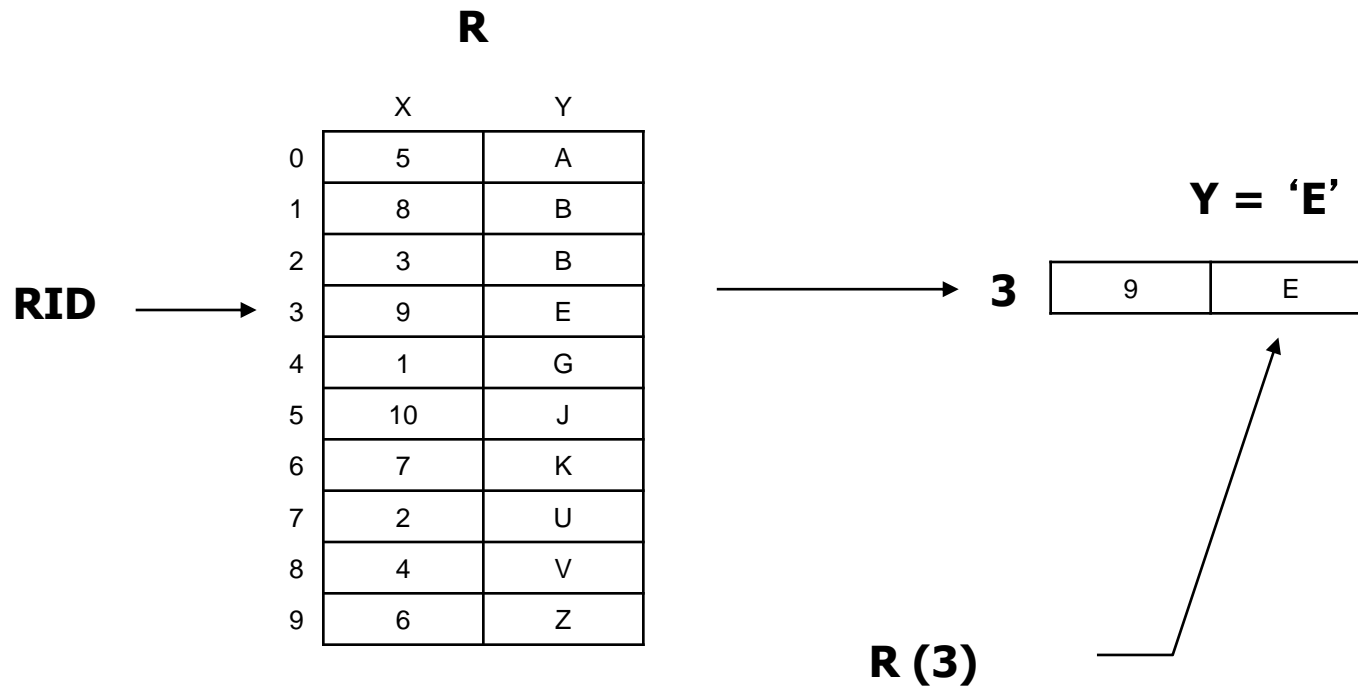
- Solution 2 : Hash



Tuple physical access : RID



Raw Identifier (RID)

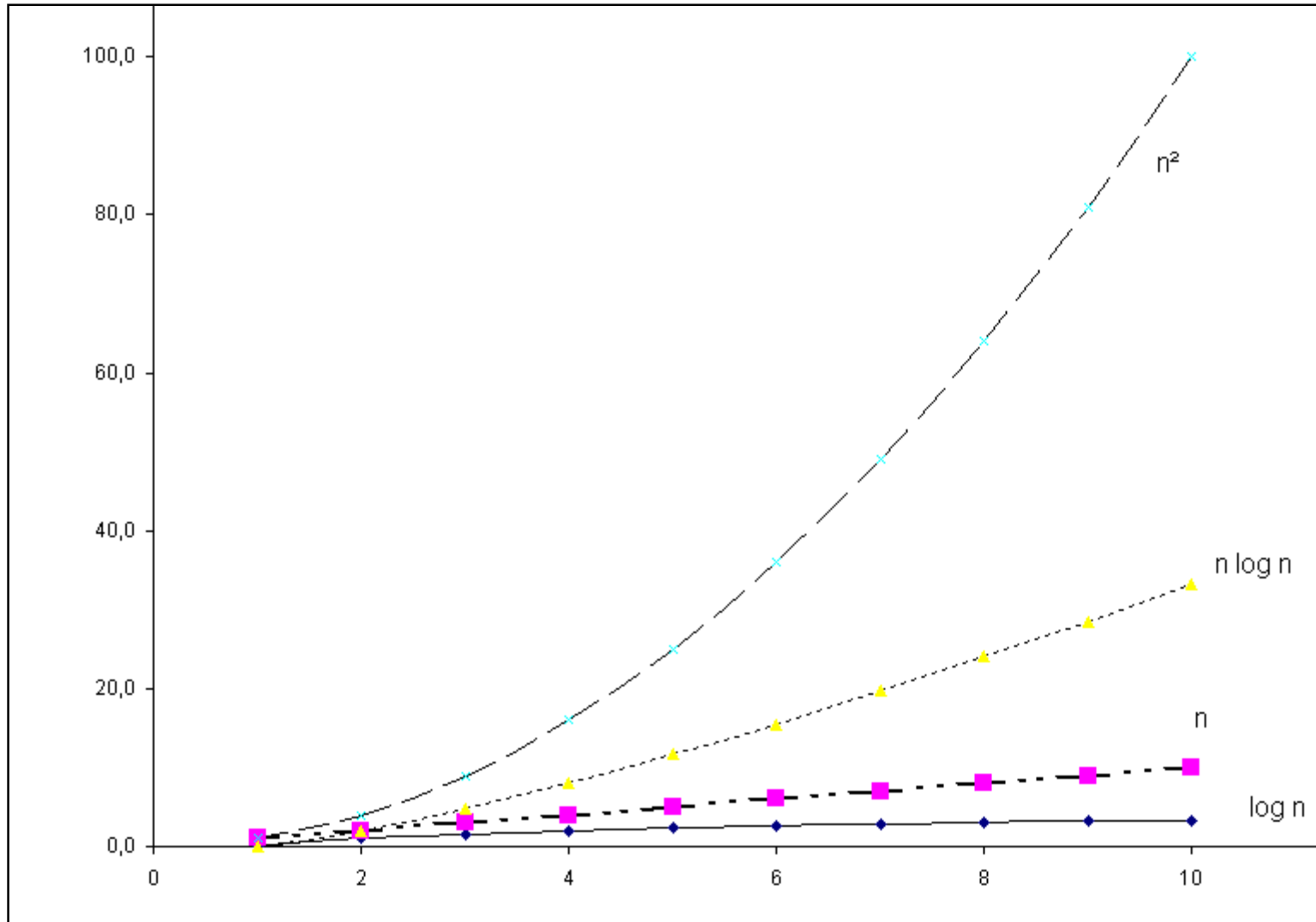


Solution 2

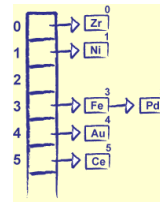


- Pour rechercher dans une table un Tuples ayant un attribut A tel que $A = c$
 - La fonction de hachage renvoie par calcul le RID du Tuples.
- Le hachage permet d'éviter les comparaisons.

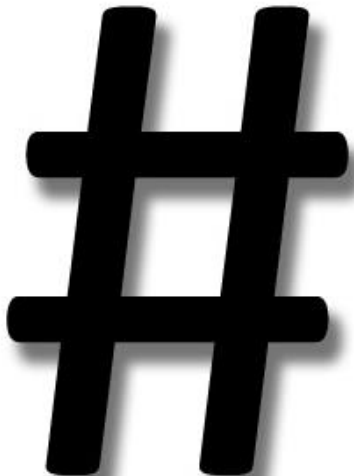
Complexity



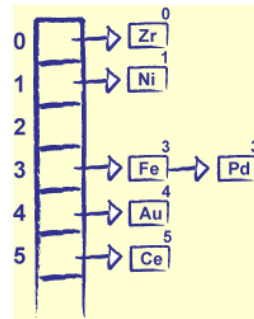
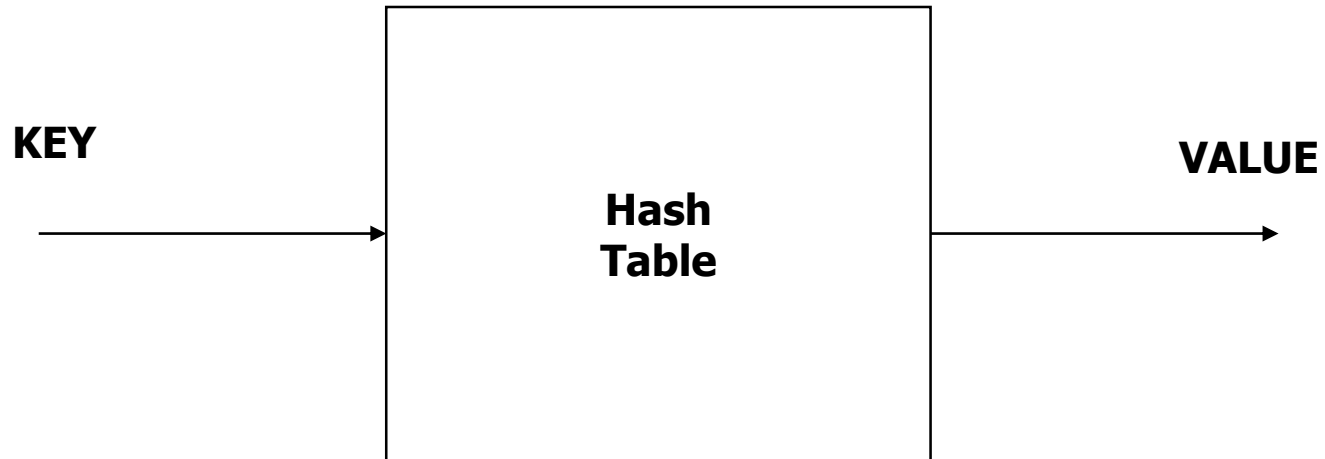
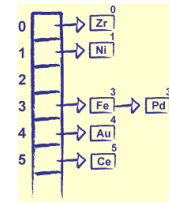
Principe du hachage (associatif)



- Une table de hachage est une structure de données qui permet une association clé éléments.
- Le temps moyen pour chercher un élément est en $O(1)$.
- Le temps pour le cas le pire est en $O(n)$.



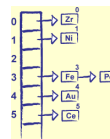
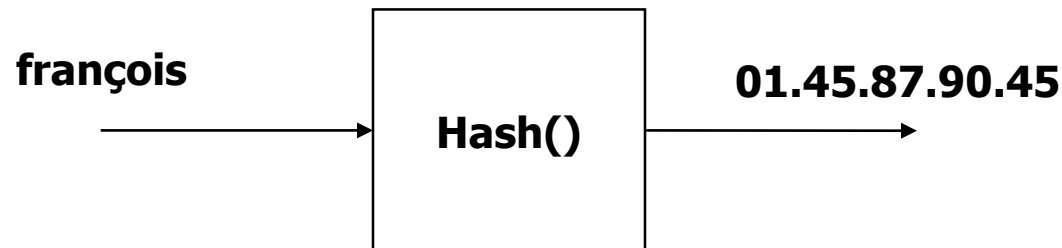
Key Value Pair



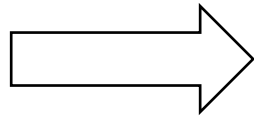
Exemple : Liste de numéros de téléphone



Key	VALUE
pierre	01.42.78.96.12
paul	03.67.90.67.00
françois	01.45.87.90.45
mohamed	04.88.8945.29
khaled	06.98.56.22.48
laila	01.23.45.67.89
alex	34.56.15.27.78.
claire	34.56.73.45.67
philippe	02.34.26.48.26.

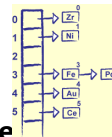
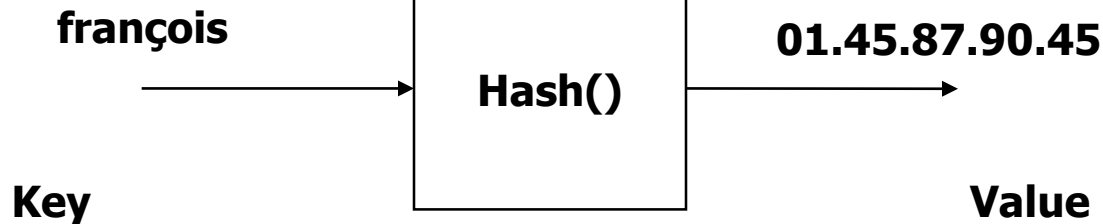


Exemple : Liste de numéros de téléphone



	Key	VALUE
0	pierre	01.42.78.96.12
1	paul	03.67.90.67.00
2	françois	01.45.87.90.45
3	mohamed	04.88.8945.29
4	khaled	06.98.56.22.48
5	laila	01.23.45.67.89
6	alex	34.56.15.27.78.
7	claire	34.56.73.45.67
8	philippe	02.34.26.48.26.

**Texte transformé en
indice de tableau**



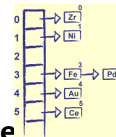
Exemple : Liste de numéros de téléphone

#



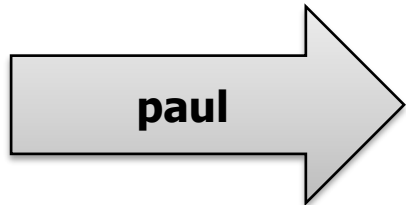
	Key	Key RID	VALUE
0	pierre	0	01.42.78.96.12
1	paul	1	03.67.90.67.00
2	françois	2	01.45.87.90.45
3	mohamed	3	04.88.8945.29
4	khaled	4	06.98.56.22.48
5	laila	5	01.23.45.67.89
6	alex	6	34.56.15.27.78.
7	claire	7	34.56.73.45.67
8	philippe	8	02.34.26.48.26.

Texte transformé en
indice de tableau



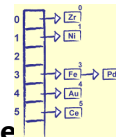
Exemple : Liste de numéros de téléphone

#



	Key	Key RID	VALUE
0	pierre	0	01.42.78.96.12
1	paul	1	03.67.90.67.00
2	françois	2	01.45.87.90.45
3	mohamed	3	04.88.8945.29
4	khaled	4	06.98.56.22.48
5	laila	5	01.23.45.67.89
6	alex	6	34.56.15.27.78.
7	claire	7	34.56.73.45.67
8	philippe	8	02.34.26.48.26.

Texte transformé en
indice de tableau



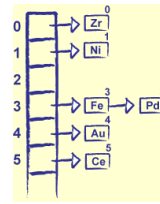
Octothorpe



- The symbol # is most commonly known as a number sign, hash, or pound sign. Other names include octothorpe and hashtag.
- Not to be confused with the Chinese character 井, the sharp sign (#),



Tableau associatif



- Les Tuples sont retrouvés dans une table par une approche associative.
- Ce qui est le principe du modèle relationnel

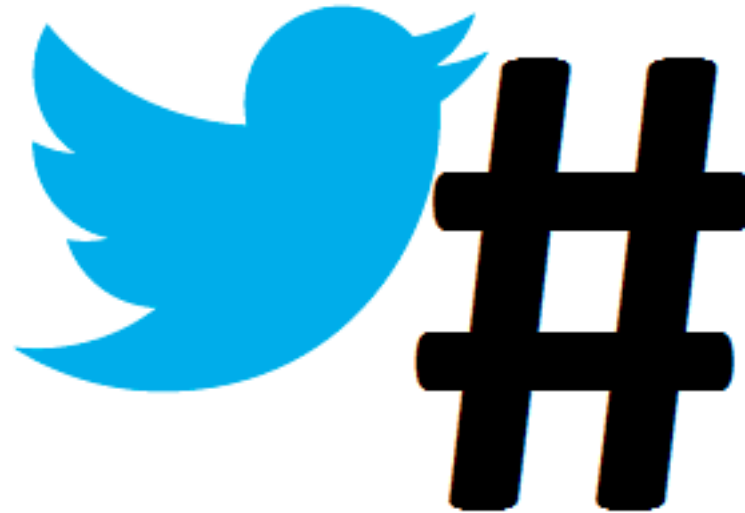
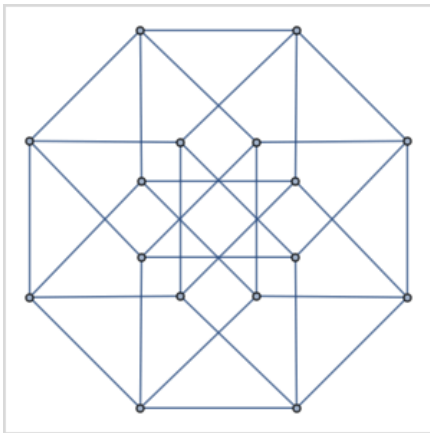
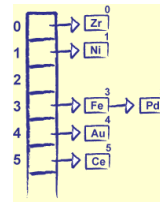
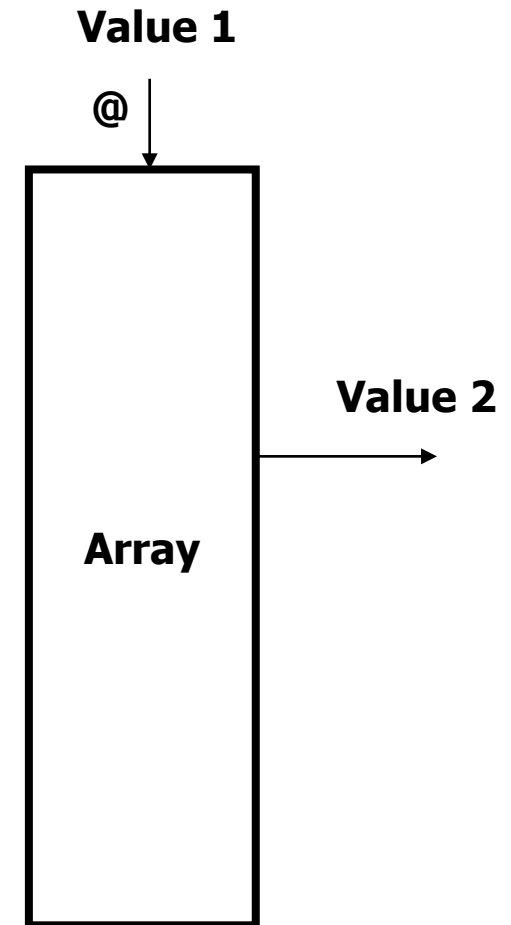


Tableau associatif



- Les Tuples sont retrouvés dans une table par une approche associative.
- Ce qui est le principe du modèle relationnel
- Adresse = Value



Principe de la mémoire associative



- Gestion de la mémoire virtuelle
- Mémoire cache
- Anté Mémoire

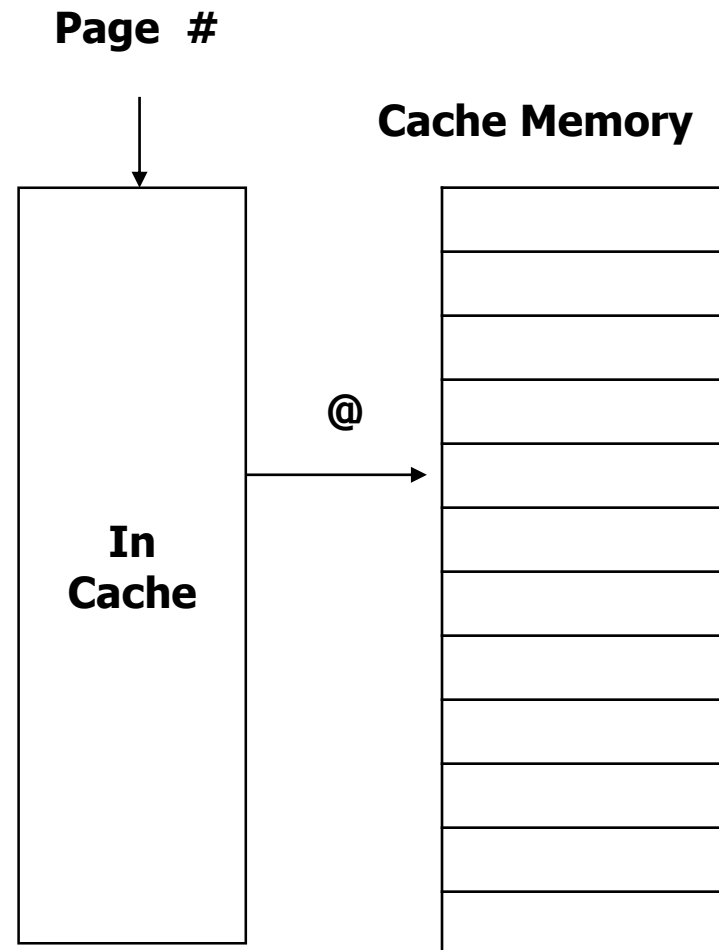
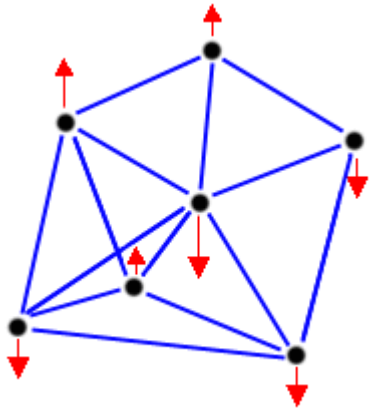
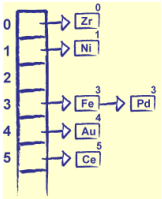
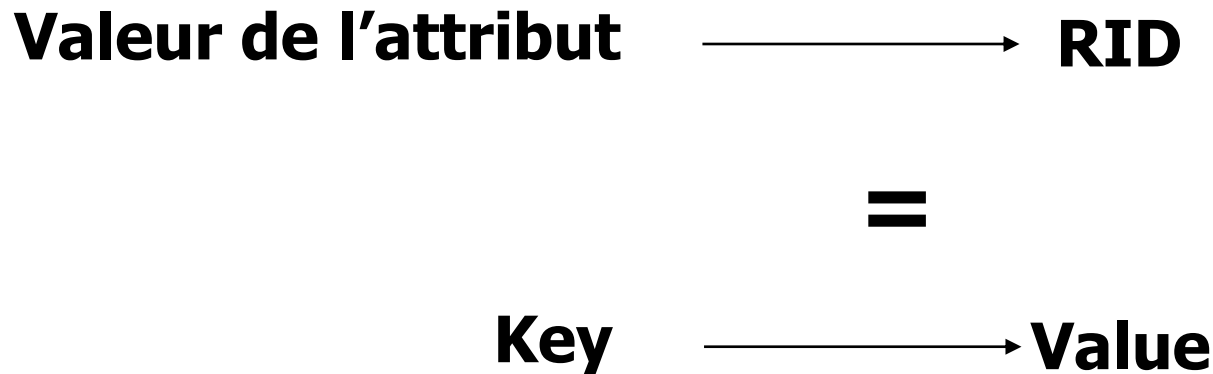


Table de hachage



- Une table de hachage est une structure de données qui permet une association « clef » « éléments ».



Class HashMap JSE 1.4



- Object **get**(Object key)
 - Returns the value to which the specified key is mapped in this identity hash map, or null if the map contains no mapping for this key.
- Object **put**(Object key, Object value)
 - Associates the specified value with the specified key in this map.
- Object **remove**(Object key)
 - Removes the mapping for this key from this map if present

Java Class Object



- Class Object is the root of the class hierarchy. Every class has Object as a superclass.
- All objects, including arrays, implement the methods of this class.
- Every class you write inherits the instance methods of Object.
- You may need to override them with code that is specific to your class.
- The `java.lang.Object.hashCode()` method returns a hash code value for the object.

JavaDoc Snippet



hashCode

```
public int hashCode()
```

Returns a hash code for this string. The hash code for a `String` object is computed as

$$s[0]*31^{(n-1)} + s[1]*31^{(n-2)} + \dots + s[n-1]$$

using `int` arithmetic, where `s[i]` is the *i*th character of the string, `n` is the length of the string, and `^` indicates exponentiation. (The hash value of the empty string is zero.)

Overrides:

`hashCode` in class `Object`

Returns:

a hash code value for this object.

Hachage d'une String



x : chaîne de caractères

$$h(x) = \sum_{i=0}^{l-1} x[i] B^{l-1-i} \bmod N$$

$$h(x) = (x[1]x B^{l-1} + x[2]x B^{l-2} + \dots + x[l]) \bmod N$$

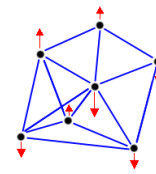
B puissance de 2.

$B = 128, B = 256$

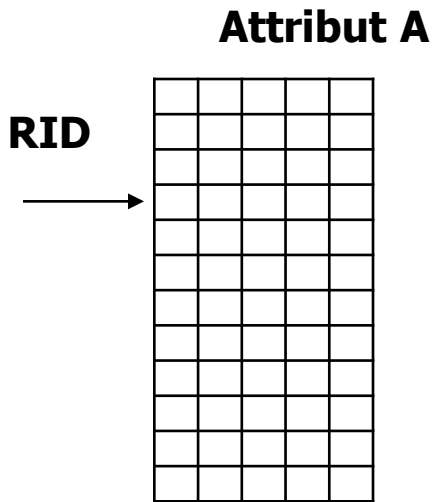
N nombre premier

Valeur de l'Attribut = Key

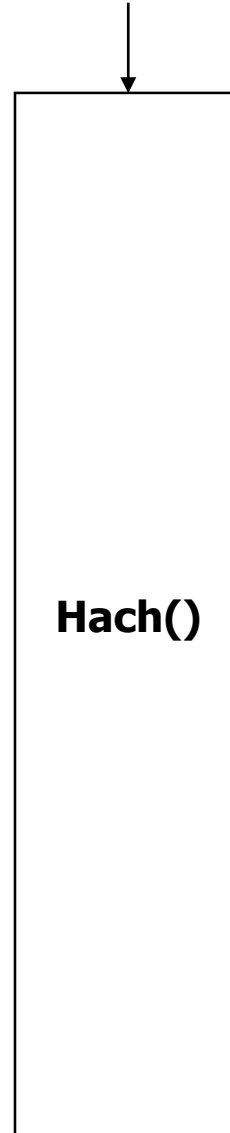
Tableau associatif



Relation R



« c »

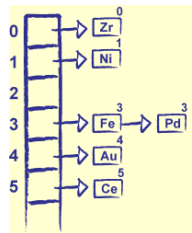


hashtable

Relation

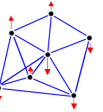
RID = Value

Nombre de lecture = 1



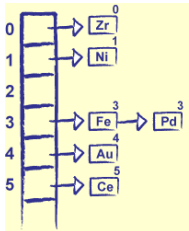
Valeur de l'Attribut = Key

Tableau associatif



Relation R

Attribut A	
RID	
1	A
2	B
3	G
4	J
5	U
6	K
7	E
8	Z
9	V
10	B



« E »



Relation R

A
B
G
J
U
K
E
Z
V
B

Hach()

7

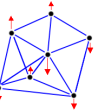
RID =
Value

hashtable

Nombre de lecture = 1

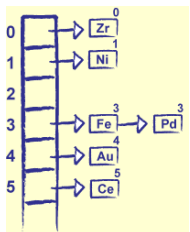
Valeur de l'Attribut = Key

Tableau associatif

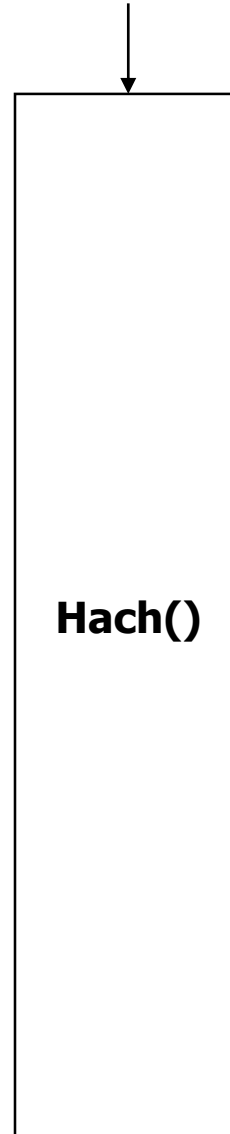


Relation R

RID	Attribut A
1	A
2	B
3	G
4	J
5	U
6	K
7	E
8	Z
9	V
10	B



« U »



hashtable

Relation R

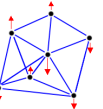
A
B
G
J
U
K
E
Z
V
B

5

RID =
Value

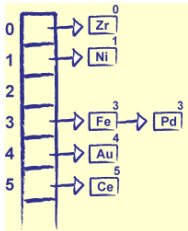
Nombre de lecture = 1

Tableau associatif

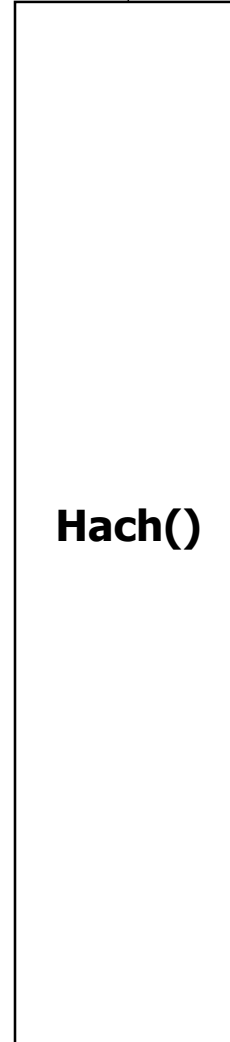


Relation R

RID	Attribut A
1	A
2	B
3	G
4	J
5	U
6	K
7	E
8	Z
9	V
10	B



« B »



hashtable

Relation R

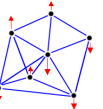
A
B
G
J
U
K
E
Z
V
B

2

10

Nombre de lecture = 1

Tableau associatif

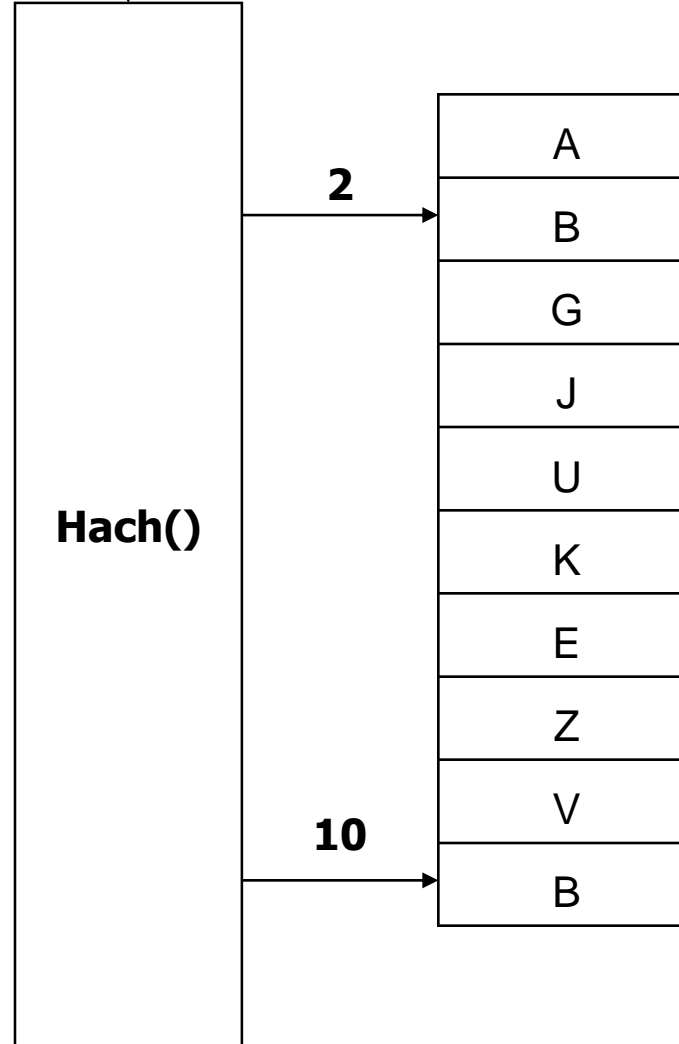


Relation R

Attribut A	
RID	
1	A
2	B
3	G
4	J
5	U
6	K
7	E
8	Z
9	V
10	B

« B »

Relation R



hashtable

Traitement des doublons

