

Algorithmes de fouille de données massives

Daniel Aloise <daniel.aloise@polymtl.ca>

Les définitions sont reprises partiellement du cours de Laurent Audibert et les images sont reprises de la publication de Kartikeya Sharma.

9 mai 2022

Big Data - SQL

- Le paradigme **MapReduce** s'étend rapidement.
- Aujourd'hui, il n'est pas rare de créer des programmes MapReduce à partir de systèmes de haut niveau, souvent une implémentation de **SQL**.
- Dans ce cours, nous allons voir comment une partie des **requêtes SQL** les plus communes peuvent être implémentées avec MapReduce.

Rappel de la terminologie SQL

- Un **attribut** est un identificateur (un nom) décrivant une information stockée dans une base. Exemples d'attribut : l'âge d'une personne
- Le **domaine** d'un attribut est l'ensemble, fini ou infini, de ses valeurs possibles. Par exemple, l'attribut numéro de sécurité sociale a pour domaine l'ensemble des combinaisons de quinze chiffres

Rappel de la terminologie SQL

- Une **relation** est un sous-ensemble du produit cartésien de n domaines d'attributs. Une relation est représentée sous la forme d'un tableau à deux dimensions dans lequel les n attributs correspondent aux titres des n colonnes
- Une **occurrence**, ou n -uplets (*tuples* en anglais), est un élément de l'ensemble figuré par une relation. Autrement dit, une occurrence est une ligne du tableau qui représente la relation
- Un **schéma de relation** précise le nom de la relation ainsi que la liste des attributs avec leurs domaines.

Exemple de relation - Personne

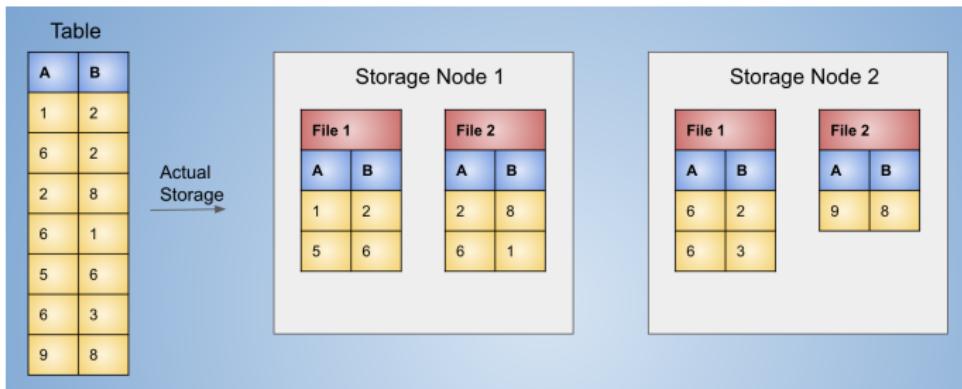
- Exemple de relation de schéma Personne (N° sécu : Entier, Nom : Chaîne, Prénom : Chaîne)

N° Sécu	Nom	Prénom
354338532195874	Durand	Caroline
345353545435811	Dubois	Jacques
173354684513546	Dupont	Lisa
973564213535435	Dubois	Rose-Marie

Exemple de relation - Web

- Une relation de schéma Lien(url1 : Chaîne, url2 : Chaîne) décrivant la structure du web contient $\approx 10^9$ occurrences
- Bien que larges, cette relation peut être stockée comme un fichier dans un **service de fichiers répartis** (*distributed file system* en anglais).
- En pratique, la relation est partitionnée en plein de petits fichiers qui sont stockés sur plusieurs noeuds

Exemple de relation stockée avec un service de fichiers répartis

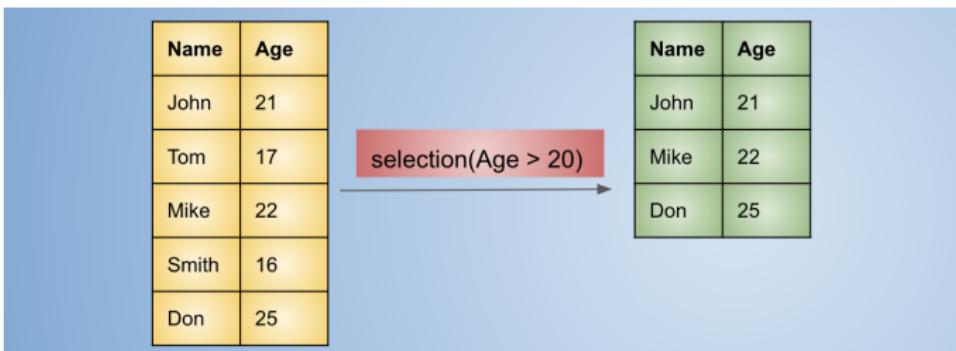


Algèbre relationnelle

- Plusieurs opérations sur les relations, communément appelées **algèbre relationnelle**, sont souvent utilisées pour implémenter les requêtes SQL.
- Nous allons étudier quelques-unes des opérations les plus courantes.

Algèbre relationnelle - Sélection

Sélection : (clause WHERE en SQL) génère une relation regroupant exclusivement toutes les occurrences de la relation qui satisfont la condition.



Algèbre relationnelle - Sélection

- La **sélection** peut être implémentée en MapReduce avec les fonctions suivantes :
 - **Map** : pour chaque ligne r dans la relation, retourne (r, r) si et seulement si la condition est satisfaite. Autrement, la fonction ne retourne rien.
 - **Reduce** : retourne simplement la clé reçue en entrée. De manière informelle, la fonction « ne fait rien ».

Algèbre relationnelle - Sélection

- Considérons un exemple dans lequel nous sélectionnons toutes les lignes dont la valeur de B est inférieure ou égale à 3.

The diagram illustrates two Map Workers, Worker 1 and Worker 2, processing data from two files each. The data is presented in tables with columns A and B.

Map Worker 1:

File 1		File 2	
A	B	A	B
1	2	2	8
2	3	4	4
5	6	6	1

Map Worker 2:

File 1		File 2	
A	B	A	B
6	2	9	8
6	3	3	3
7	6	0	1

Algèbre relationnelle - Sélection

- La sortie des Map workers :

Map Worker 1		Map Worker 2	
Key	Value	Key	Value
(1, 2)	(1, 2)	(6, 2)	(6, 2)
(2, 3)	(2, 3)	(6, 3)	(6, 3)
(6, 1)	(6, 1)	(3, 3)	(3, 3)
		(0, 1)	(0, 1)

Algèbre relationnelle - Sélection

- Les Reduce workers :

Reduce Worker 1		Reduce Worker 2	
RW 1		RW 2	
Key	Value	Key	Value
(1,2)	(1, 2)	(6,2)	(6,2)
(2, 3)	(2, 3)	(6, 3)	(6, 3)

RW 1		RW 2	
Key	Value	Key	Value
(1,2)	(1, 2)	(6,1)	(6, 1)
(2, 3)	(2, 3)	(3,3)	(3, 3)
		(0,1)	(0, 1)

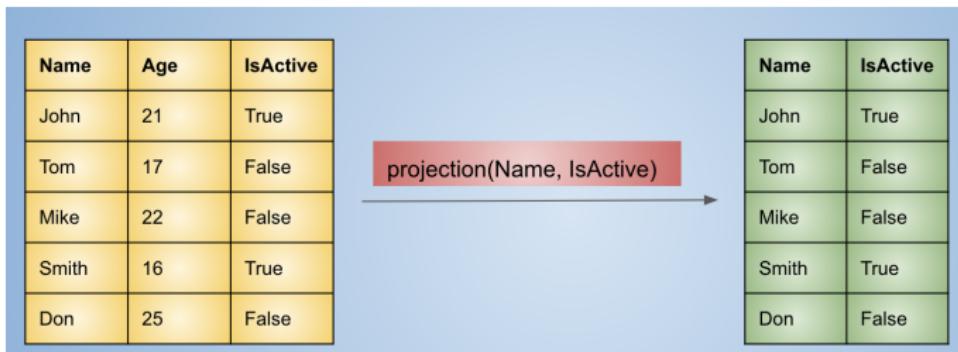
Algèbre relationnelle - Sélection

- La sortie des Reduce workers :

Reduce Worker 1		Reduce Worker 2	
File 1		File 1	
A	B	A	B
1	2	6	1
2	3	3	3
6	2	0	1
6	3		

Algèbre relationnelle - Projection

Projection : (**SELECT en SQL**) supprime les attributs d'une relation qui ne sont pas sélectionnés et à élimine les occurrences en double apparaissant dans la nouvelle relation.



Algèbre relationnelle - Projection

- La **projection** peut être implémentée en MapReduce avec les fonctions suivantes :
 - **Map** : pour chaque ligne r dans la relation, retourne (r', r') tel que r' ne contienne que les attributs sélectionnés. Après la suppression des attributs, il est possible que la sortie contienne des occurrences en double.
 - **Reduce** : reçoit en entrée $(r', [r', r', r', r', \dots])$ et retourne une seule paire (r', r') , ce qui a pour effet de supprimer les doublons.

Algèbre relationnelle - Projection

- Considérons un exemple dans lequel nous sélectionnons les attributs A et B.

The diagram illustrates two Map Workers processing data from two files. Each worker has two output slots labeled "File 1" and "File 2". Each slot outputs two columns labeled "A" and "B".

Map Worker 1		
File 1		
A	B	C
1	2	3
2	2	2
1	2	1

File 2		
A	B	C
4	2	1
6	8	4
3	2	2

Map Worker 2		
File 1		
A	B	C
1	2	5
2	3	2
1	3	1

File 2		
A	B	C
3	2	1
6	8	9
3	4	2

Algèbre relationnelle - Projection

- La sortie des Map workers :

Map Worker 1		Map Worker 2	
Key	Value	Key	Value
(1, 2)	[(1, 2), (1, 2)]	(1, 2)	[(1, 2)]
(2, 2)	[(2, 2)]	(2, 3)	[(2, 3)]
(4, 2)	[(4, 2)]	(1, 3)	[(1, 3)]
(6, 8)	[(6, 8)]	(3, 2)	[(3, 2)]
(3, 2)	[(3, 2)]	(6, 8)	[(6, 8)]
		(3, 4)	[(3, 4)]

Algèbre relationnelle - Projection

- Les Reduce workers :

Reduce Worker 1		Reduce Worker 2	
RW 1		RW 1	
Key	Value	Key	Value
(1,2)	[(1, 2), (1, 2)]	(1,2)	[(1, 2)]
(2, 2)	[(2, 2)]	(2, 3)	[(2, 3)]
(4, 2)	[(4, 2)]	(1, 3)	[(1, 3)]

RW 2		RW 2	
Key	Value	Key	Value
(6, 8)	[(6, 8)]	(3, 2)	[(3, 2)]
(3, 2)	[(3, 2)]	(6, 8)	[(6, 8)]
		(3, 4)	[(3, 4)]

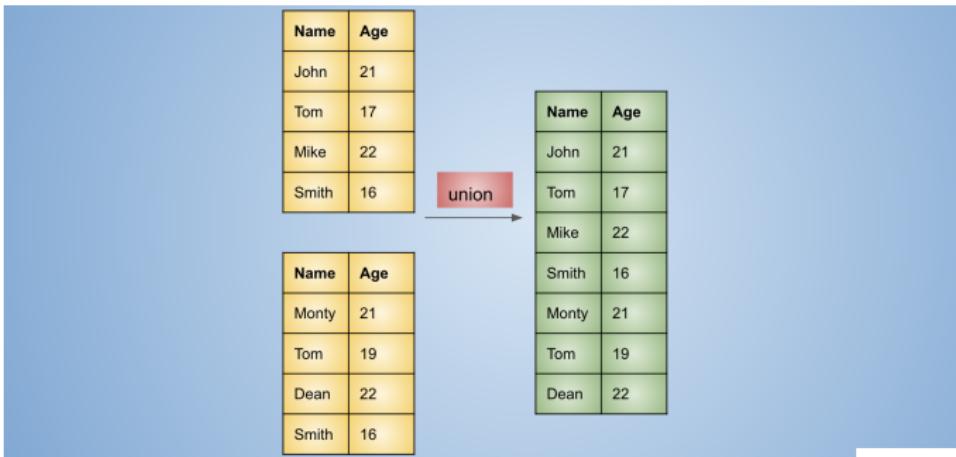
Algèbre relationnelle - Projection

- La sortie des Reduce workers :

Reduce Worker 1		Reduce Worker 2	
File 1		File 1	
A	B	A	B
1	2	6	8
2	2	3	2
4	2	3	4
2	3		
1	3		

Algèbre relationnelle - Union

Union : (**UNION en SQL**) L'union est une opération portant sur deux relations ayant le même schéma et construisant une troisième relation constituée des occurrences appartenant à chacune des deux relations sans doublon.



Algèbre relationnelle - Union

- Les opérations de sélection et de projection sont appliquées à une seule relation alors que l'union est appliquée à deux relations ou plus. Nous supposons donc que les relations ont le même schéma.
- L'**union** peut être implémentée en MapReduce avec les fonctions suivantes :
 - **Map** : pour chaque ligne r , retourne (r, r)
 - **Reduce** : reçoit en entrée $(r', [r', r', r', r', \dots])$ et retourne une seule paire (r', r') , ce qui a pour effet de supprimer les doublons.

Algèbre relationnelle - Union

- Dans l'exemple ci-dessous, les occurrences jaunes et vertes appartiennent à deux relations différentes.

The diagram illustrates the results of a join operation across two map workers. Each worker's output is shown in a separate box:

- Map Worker 1:**
 - Table 1:** Contains three rows with columns A and B. The first row has A=1, B=2. The second row has A=2, B=3. The third row has A=5, B=6.
 - Table 2:** Contains four rows with columns A and B. The first row has A=2, B=3. The second row has A=4, B=4. The third row has A=6, B=1. The fourth row has A=6, B=1.
- Map Worker 2:**
 - Table 1:** Contains three rows with columns A and B. The first row has A=6, B=1. The second row has A=6, B=3. The third row has A=7, B=6.
 - Table 2:** Contains four rows with columns A and B. The first row has A=9, B=8. The second row has A=3, B=3. The third row has A=0, B=1.

Algèbre relationnelle - Union

- La sortie des Map workers :

Map Worker 1		Map Worker 2	
Key	Value	Key	Value
(1, 2)	[(1, 2)]	(6, 1)	[(6, 1)]
(2, 3)	[(2, 3), (2, 3)]	(6, 3)	[(6, 3)]
(5, 6)	[(5, 6)]	(7, 6)	[(7, 6)]
(4, 4)	[(4, 4)]	(9, 8)	[(9, 8)]
(6, 1)	[(6, 1)]	(3, 3)	[(3, 3)]
		(0, 1)	[(0, 1)]

Algèbre relationnelle - Union

- Les Reduce workers :

Reduce Worker 1		Reduce Worker 2	
Key	Value	Key	Value
(1,2)	[(1, 2)]	(4, 4)	[(4, 4)]
(2, 3)	[(2, 3), (2, 3)]	(6, 1)	[(6, 1), (6, 1)]
(5, 6)	[(5, 6)]	(7, 6)	[(7, 6)]
(6, 3)	[(6, 3)]	(9, 8)	[(9, 8)]
(3, 3)	[(3, 3)]		
(0, 1)	[(0, 1)]		

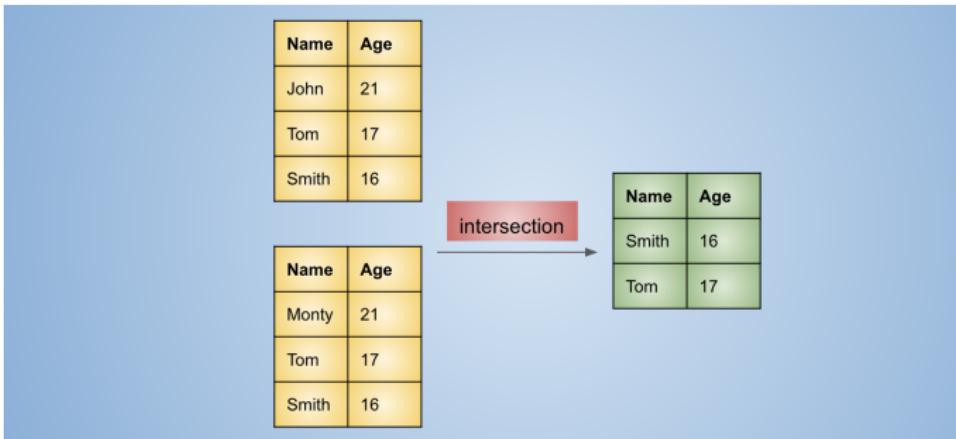
Algèbre relationnelle - Union

- La sortie des Reduce workers :

Reduce Worker 1		Reduce Worker 2	
A	B	A	B
1	2	4	4
2	3	6	1
5	6	7	6
6	3	9	8
3	3		
0	1		

Algèbre relationnelle - Intersection

Intersection : (**INTERSECT** en SQL) L'intersection est une opération portant sur deux relations ayant le même schéma et construisant une troisième relation dont les occurrences sont constituées de ceux appartenant aux deux relations.



Algèbre relationnelle - Intersection

- L'**intersection** peut être implémentée en MapReduce avec les fonctions suivantes :
 - **Map** : pour chaque ligne r , retourne (r, r)
 - **Reduce** : chaque clé peut être associée à une ou deux valeurs (puisque nous supposons qu'il n'y a pas de doublons dans les relations). Dans le cas où il y a deux valeurs, la fonction retourne (r, r) , autrement rien n'est retourné.

Algèbre relationnelle - Intersection

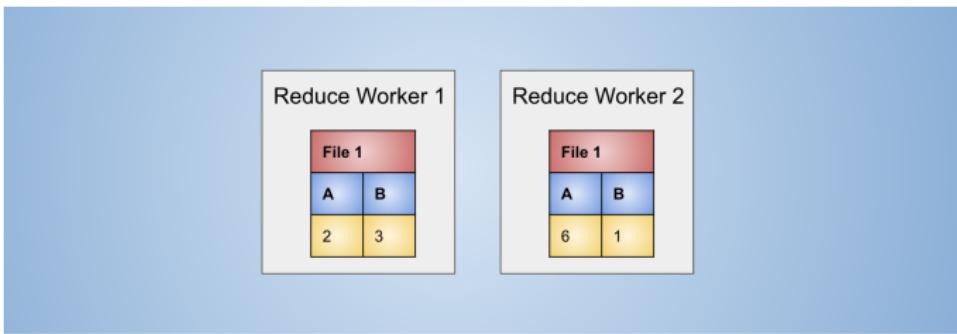
- Considérons le même exemple que pour l'union.
- Puisque la fonction Map est la même pour l'union et l'intersection, regardons directement la fonction Reduce :

Reduce Worker 1	
Key	Value
(1,2)	[(1, 2)]
(2, 3)	[(2, 3), (2, 3)]
(5, 6)	[(5, 6)]
(6, 3)	[(6, 3)]
(3, 3)	[(3, 3)]
(0, 1)	[(0, 1)]

Reduce Worker 2	
Key	Value
(4, 4)	[(4, 4)]
(6, 1)	[(6, 1), (6, 1)]
(7, 6)	[(7, 6)]
(9, 8)	[(9, 8)]

Algèbre relationnelle - Intersection

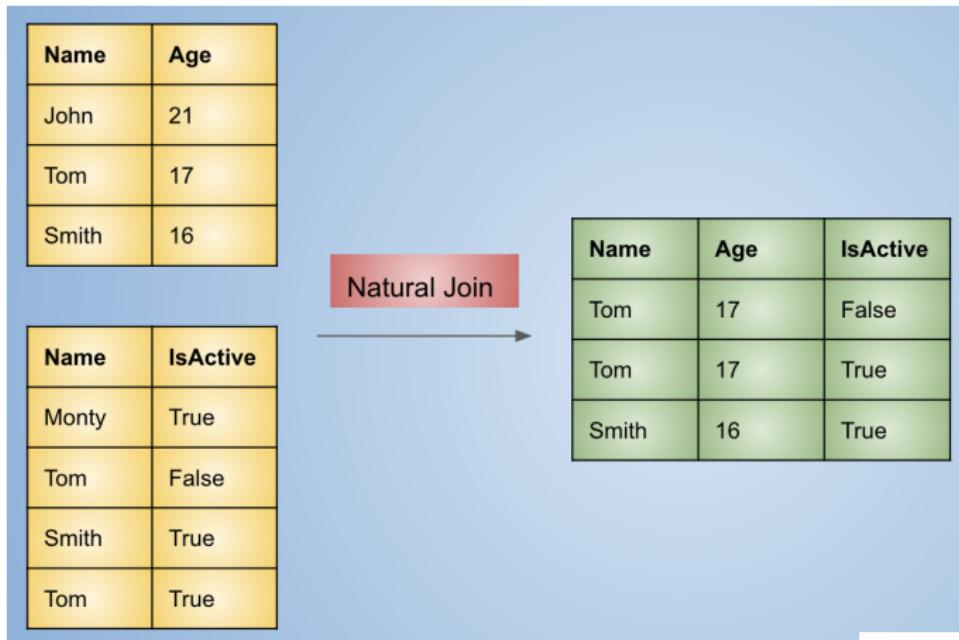
- La sortie des Reduce workers :



Algèbre relationnelle - Jointure naturelle

Jointure naturelle : (**INNER JOIN en SQL**) La jointure est une opération portant sur deux relations qui construit une troisième relation regroupant toutes les possibilités de combinaison des occurrences des relations qui satisfont un test d'égalité entre les attributs qui portent le même nom dans les relations. Dans la relation construite, ces attributs ne sont pas dupliqués, mais fusionnés en une seule colonne par couple d'attributs.

Algèbre relationnelle - Jointure naturelle



Algèbre relationnelle - Jointure naturelle

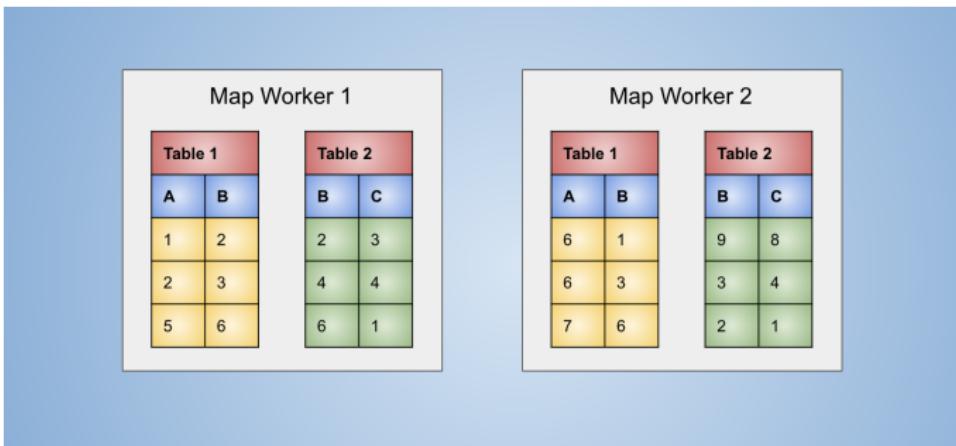
- Pour faire la **jointure naturelle** avec MapReduce, nous avons besoin de connaître la relation d'origine de chaque valeur.
- Si deux valeurs avec la même clé proviennent de relations différentes, alors il faut créer une occurrence contenant ces deux valeurs.
- La **jointure** peut faire exploser le nombre d'occurrences puisqu'il est possible de créer chaque combinaison possible entre les occurrences des deux relations.

Algèbre relationnelle - Jointure naturelle

- **Map** : Pour les deux relations `Table1(A, B)` et `Table2(B, C)`, la fonction `Map` retourne `(b,[T1, a])` ou `(b,[T2, c])` selon la relation d'origine.
- **Reduce** : Pour chaque clé `b`, la fonction construit toutes les paires possibles contenant une valeur de chaque relation. La fonction retourne toutes les combinaisons sous la forme `(b, [a, c])` représentant une occurrence `(a, b, c)` dans la nouvelle relation.

Algèbre relationnelle - Jointure naturelle

- Considérons la **jointure naturelle** des relations **Table 1** et **Table 2**, où **B** est l'attribut commun.



Algèbre relationnelle - Jointure naturelle

- La sortie des Map workers :

Map Worker 1	
Key	Value
2	$[(T1, 1), (T2, 3)]$
3	$[(T1, 2)]$
6	$[(T1, 5), (T2, 1)]$
4	$[(T1, 4)]$

Map Worker 2	
Key	Value
1	$[(T1, 6)]$
3	$[(T1, 6), (T2, 4)]$
6	$[(T1, 7)]$
9	$[(T2, 8)]$
2	$[(T2, 1)]$

Algèbre relationnelle - Jointure naturelle

- Les Reduce workers :

Reduce Worker 1	
Key	Value
2	$\{(\text{T1}, 1), (\text{T2}, 3), (\text{T2}, 1)\}$
3	$\{(\text{T1}, 2), (\text{T1}, 6), (\text{T2}, 4)\}$
1	$\{(\text{T1}, 6)\}$

Reduce Worker 2	
Key	Value
6	$\{(\text{T1}, 5), (\text{T1}, 7), (\text{T2}, 1)\}$
4	$\{(\text{T1}, 4)\}$
9	$\{(\text{T2}, 8)\}$

Algèbre relationnelle - Jointure naturelle

- La sortie des Reduce workers :

Reduce Worker 1		
B	A	C
2	1	3
2	1	1
3	2	4
3	6	4

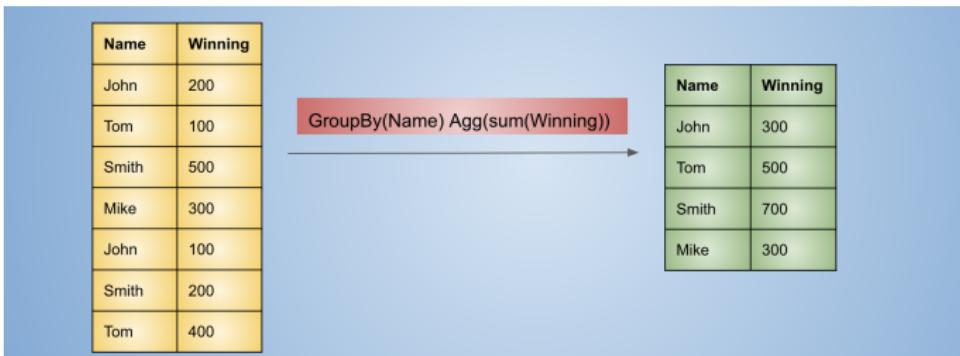
Reduce Worker 2		
B	A	C
6	5	1
6	7	1

- Si une clé ne contient que des valeurs de T1 ou T2, alors la fonction Reduce ne retourne rien.

Algèbre relationnelle - Regrouper et agréger

Regrouper et agréger : (GROUP BY en SQL) regroupe les occurrences selon un ensemble d'attributs et effectue une opération d'agrégation (`sum`, `count`, `max`, `min`, etc.) pour chaque groupe sur un autre attribut.

Algèbre relationnelle - Regrouper et agréger



Algèbre relationnelle - Regrouper et agréger

- Supposons que nous regroupons les occurrences selon l'attribut **A** et que nous agrégeons selon l'attribut **B**.
- **Map** : Pour chaque ligne **(a,b,c)** dans la relation, retourne **(a,b)**.
- **Reduce** : Chaque clé **a** représente un groupe. La fonction applique l'opération d'agrégation (**sum**, **count**, **max**, **min**, etc.) sur les valeurs et retourne le résultat.

Algèbre relationnelle - Regrouper et agréger

- Groupons les occurrences selon les attributs (A, B) et effectuons la somme de l'attribut C.

Map Worker 1

File 1				
A	B	C	D	
1	2	3	1	
2	2	3	2	
1	2	1	3	

File 2				
A	B	C	D	
4	2	1	3	
6	8	4	4	
3	2	2	4	

Map Worker 2

File 1				
A	B	C	D	
1	2	5	2	
2	3	2	4	
1	3	1	3	

File 2				
A	B	C	D	
3	2	1	3	
2	3	9	2	
3	4	2	1	

Algèbre relationnelle - Regrouper et agréger

- La sortie des Map workers :

Map Worker 1	
Key	Value
(1, 2)	[3, 1]
(2, 2)	[3]
(4, 2)	[1]
(6, 8)	[4]
(3, 2)	[2]

Map Worker 2	
Key	Value
(1, 2)	[5]
(2, 3)	[2, 9]
(1, 3)	[1]
(3, 2)	[1]
(3, 4)	[2]

Algèbre relationnelle - Regrouper et agréger

- Les Reduce workers :

Reduce Worker 1	
Key	Value
(1,2)	[3, 1, 5]
(2, 2)	[3]
(4, 2)	[1]
(2, 3)	[(2, 9)]

Reduce Worker 2	
Key	Value
(6, 8)	[4]
(3, 2)	[1, 2]
(3, 4)	[2]
(1, 3)	[1]

Algèbre relationnelle - Regrouper et agréger

- La sortie des Reduce workers :

Reduce Worker 1

A	B	Sum
1	2	9
2	2	3
4	2	1
2	3	11

Reduce Worker 2

A	B	Sum
6	8	4
3	2	3
3	4	2
1	3	1