

Relational Algebra Operations and MapReduce

Relational Algebra Operators

- The relational algebra and the SQL language have many useful operators
 - Selection
 - Projection
 - Union, intersection, and difference
 - Join (see Join design patterns)
 - Aggregations and Group by (see the Summarization design patterns)

Relational Algebra Operators

- The MapReduce paradigm can be used to implement relational operators
 - However, the MapReduce implementation is efficient only when a full scan of the input table(s) is needed
 - i.e., when queries are not selective and process all data
 - Selective queries, which return few tuples/records of the input tables, are usually not efficient when implemented by using a MapReduce approach

Relational Algebra Operators

- Most preprocessing activities involve relational operators
 - E.g., ETL processes in the data warehousing application context

Relations/Tables

- Relations/Tables (also the big ones) can be stored in the HDFS distributed file system
 - They are broken in blocks and spread across the servers of the Hadoop cluster

Relations/Tables

- Note
 - In relational algebra, relations/tables do not contain duplicate records by definition
 - This constraint must be satisfied by both the input and the output relations/tables

Selection

- $\sigma_C (R)$
 - Applies predicate (condition) C to each record of table R
 - Produces a relation containing only the records that satisfy predicate C
- The selection operator can be implemented by using the filtering pattern

Selection

Courses

<u>CCode</u>	CName	Semester	ProfID
M2170	Computer science	1	D102
M4880	Digital systems	2	D104
F1401	Electronics	1	D104
F0410	Databases	2	D102

- Find the courses held in the second semester
- $\sigma_{\text{Semester}=2}(\text{Courses})$

Selection

Courses

<u>CCode</u>	CName	Semester	ProfID
M2170	Computer science	1	D102
<i>M4880</i>	<i>Digital systems</i>	<i>2</i>	<i>D104</i>
F1401	Electronics	1	D104
<i>F0410</i>	<i>Databases</i>	<i>2</i>	<i>D102</i>



Result

<u>CCode</u>	CName	Semester	ProfID
M4880	Digital systems	2	D104
F0410	Databases	2	D102

Selection

- Map-only job
- Each mapper
 - Analyzes one record at a time of its split
 - If the record satisfies C then it emits a (key,value) pair with key=record and value=null
 - Otherwise, it discards the record

Projection

- $\pi_S(R)$
 - For each record of table R , keeps only the attributes in S
 - Produces a relation with a schema equal to S (i.e., a relation containing only the attributes in S)
 - Removes duplicates, if any

Projection

Professors

<u>ProfId</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	Smith	Electronics

- Find the surnames of all professors
- $\pi_{\text{PSurname}}(\text{Professors})$

Projection

Professors

<u>ProfId</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	Smith	Electronics

Result

PSurname
Smith
Jones

- Duplicated values are removed

Projection

- Each mapper
 - Analyzes one record at a time of its split
 - For each record r in R
 - It selects the values of the attributes in S and constructs a new record r'
 - It emits a (key,value) pair with $\text{key}=r'$ and $\text{value}=\text{null}$
- Each reducer
 - Emits one (key, value) pair for each input (key, [list of values]) pair with $\text{key}=r'$ and $\text{value}=\text{null}$

Union

- $R \cup S$
 - R and S have the same schema
 - Produces a relation with the same schema of R and S
 - There is a record t in the output of the union operator for each record t appearing in R or S
 - Duplicated records are removed

Union

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics

- Find information relative to the professors of degree courses or master's degrees
- DegreeCourseProf \cup MasterCourseProf

Union

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics



Result

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics
D101	Red	Electronics

Union

- Mappers
 - For each input record t in R , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{null}$
 - For each input record t in S , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{null}$
- Reducers
 - Emit one (key, value) pair for each input (key, [list of values]) pair with $\text{key}=t$ and $\text{value}=\text{null}$
 - i.e., one single copy of each input record is emitted

Intersection

- $R \cap S$
 - R and S have the same schema
 - Produces a relation with the same schema of R and S
 - There is a record t in the output of the intersection operator if and only if t appears in both relations (R and S)

Intersection

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics

- Find information relative to professors teaching both degree courses and master's courses
- DegreeCourseProf \cap MasterCourseProf

Intersection

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics

Result

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering



Intersection

- Mappers
 - For each input record t in R , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{"R"}$
 - For each input record t in S , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{"S"}$

Intersection

- Reducers
 - Emit one (key, value) pair with key=t and value=null for each input (key, [list of values]) pair with [list of values] containing two values
 - It happens if and only if both R and S contain t

Difference

- $R - S$
 - R and S have the same schema
 - Produces a relation with the same schema of R and S
 - There is a record t in the output of the difference operator if and only if t appears in R but not in S

Difference

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics

- Find the professors teaching degree courses but not master's courses
- DegreeCourseProf - MasterCourseProf

Difference

DegreeCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D105	Jones	Computer engineering
D104	White	Electronics

MasterCourseProf

<u>ProfID</u>	PSurname	Department
D102	Smith	Computer engineering
D101	Red	Electronics



Result

<u>ProfID</u>	PSurname	Department
D105	Jones	Computer engineering
D104	White	Electronics

Difference

- Mappers
 - For each input record t in R , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{name of the relation (i.e., } R)$
 - For each input record t in S , emit one (key, value) pair with $\text{key}=t$ and $\text{value}=\text{name of the relation (i.e., } S)$
- Two mapper classes are needed
 - One for each relation

Difference

- Reducers
 - Emit one (key, value) pair with key=t and value=null for each input (key, [list of values]) pair with [list of values] containing only the value R
 - It happens if and only if t appears in R but not in S

Join

- The join operators can be implemented by using the Join pattern
 - By using the reduce side or the map side pattern depending on the size of the input relations/tables

Aggregations and Group by

- Aggregations and Group by are implemented by using the Summarization pattern