RESEARCH REVIEW JOURNALS
Volume-04
Issue-05
May-2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com[UGC Listed Journal]

The Relational Calculus and Relational Algebra

[1]Enam Ahmad & [2]Dr. Manish Kumar

[1]Research Scholar, Department of Computer Science, (Under P.G. Dept. of Mathematics) Magadh University, Bodh-Gaya (India)
[2]Associate Professor, P.G.Dept. of Mathematics, A.N. College, Patna, Bihar (India)

ARTICLE DETAILS

Article History
Published Online: 25 May 2019

Keywords
Formal language, relational model, relational algebra, relational Calculus, SQL, relational DBMSs, predicate calculus.

ABSTRACT

In this Article we discuss the two formal languages for the relational model: first relational algebra and the second relational Calculus. The relational algebra and relational calculus were developed before the SQL language. In fact, in some ways, SQL is based on concepts from both the relational algebra and the relational calculus, as we shall see. Because most relational DBMSs use SQL as their language, the basic set of operations for the relational model is the relational algebra. These operations permit a user to specify basic retrieval requests as relational algebra expressions. At the other hand a relational calculus expression creates a new relation. In relational calculus declaration, there is no order of operations to define how to retrieve the query result-only what information the result should hold. Relational calculus based on the branch of mathematical logic called predicate calculus.

1. Introduction

The relational calculus (in fact there are two calculus: the tuple relational calculus and the domain relational calculus) is non-procedural and refers to quasi-natural-language expressions used for declarative formulating SQL (Structural Query Language) queries and statements (the relational calculus was also used for Datalog [1,2]); on the other hand, the relational algebra is used for procedural transformations of SQL expressions. Both the calculus and the algebra are the core parts of the relational model. The relational algebra and the relational calculus are logically equivalent.

The tuple relational calculus (introduced by Edgar F. Codd [3,4,5]) is a notation for defining a result of a query through description of its properties. The representation of a query in relational calculus consists of two parts: a target list and a selection expression. Together with the relational algebra, the calculus constituted the basis for SQL (formerly also for the already forgotten query language). However, the relational model has never been implemented completely complying with its mathematical foundations. The relational model theory defines a domain (or a datatype), a tuple (an ordered multiset of attributes) being an ordered pair of a domain and a Value and a relation variable (relvar) standing for and ordered pair of a domain and a name (a relation header), and a relation defined as a set of tuples. In relational databases a relation is reflected in a table and a tuple corresponds to a row (a record). The tuple calculus involves atomic values (atoms), operators, formulae and queries. The domain relational calculus was proposed in [6] as a declarative approach to query languages. It uses the same operators as the tuple calculus (and quantifiers) and it is meant for expressing queries as formulae. Again, its detailed description is omitted as it can be found in [6].

The relational algebra [3, 7] is based on the mathematical logic and set theory and it is equivalent to the domain calculus. The relational algebra was applied as a basis for a set of query languages, e.g., ISBL, Tutorial D, Rel, and SQL.

The relational algebra is important for many reasons. (i), it provides foundation for relational model operations. (ii), it used as a foundation for applying and an optimizing queries in the query processing and query optimization modules. Relational algebra are basic parts of relation database management systems (RDBMSs), and (iii), some of its concepts are incorporated into the SQL standard query language for RDBMSs.

There is a set of primitive operations defined in the algebra, that is the selection(σ),projection (□ □ □ *and*□the Cartesian (X) or Cross join, the set union *(*□ □ □ the set difference(-), and *rename (*□ □. All other operators, including the set intersection, the division and the natural join are expressed in terms of the above primitive ones. In terms of the optimization, the relational algebra can express each query as a tree, where the internal nodes are operators, leaves are relations and subtrees are sub expressions. Such trees are transformed to their semantically equivalent forms, where the average sizes of the relations yielded by sub expressions in the tree are smaller than they were before the optimization (e.g., avoiding calculating cross products).

Whereas the relational calculus provides a higher-level declarative language for specifying relational queries. In calculus declaration, there is no sequence of operations to specify how to retrieve the query result-only what information the result should contain. This is the main distinguishing feature between both (relational algebra and calculus). The relational calculus is important because it is fixed in Arithmetical logic and the standard query language (SQL) for RDBMSs has some of its foundations in an alteration of relational calculus known as the tuple relational calculus.

First, we describe the relational calculus operations in Section 2.1, and then we discuss Relational algebra operation in Section 2.2. In Section 3 deals with the additional operations. Section 4 represents conclusion.

2.1 Relational calculus:

Relational calculus is the formal and non-procedural query language.It also known as *declarative language* [11]. It tells what to do but never explain how to do it.

Relational calculus exists in two forms-

(a) Tuple Relational Calculus (TRC):

Filtering variable ranges over tuples [8].

Notation- { T | Condition }

Returns all tuples T that satisfies a condition.

For Example- {T.name | Author (T) AND T.article='C++'}

Output- Returns tuples with 'name' from author who has written articles on 'C++'.

TRC can be quqntified.we can use Existential (∃) and Universal Quantifiers (∀).

For example-{ R | ∃T ∈ Authors(T.artcle='C++' AND R.name=T.name)}

Output- The above query will return the same result as the previous one.

(b) Domain Relational Calculus (DRC):

In DRC, the filtering variable uses the domain of attributes instead of entire tuple values (as done in TRC, above mentioned)[8,9].

Notation- { A1,A2,A3,……A_N | P (A1,A2,A3,……A_N}

WHERE *A1,A2* are represented attributes and *P* stands for formulae built by inner attributes.

For example- { < article, page, subject > | ∈ Totpoint A subject='C++'}

Output – It return Article, Page, and Subject from the relation Totpoint, where subject is C++.

2.2 **Relational Algebra :**

The relational algebra is often considered to be an integral part of the relational data model, which takes instances of relations as input and give instances of relations as output [8,9,10]. The order is specified in which the operations have to be performed. It uses operators to perform queries. An operator can be either unary or binary.

The fundamental or basic operations of relational algebra are as follows -:

- Select (σ)
- Project (∏)
- Union (∪)
- Set different (-)
- Cartesian product (X)
- Rename (ρ)

Select Operation (σ)

The SELECT (selection, or restriction) operation is used to choose a subset(in terms of set theory) of the tuples from a relation that satisfies a selection condition. The SQL selection is similar of the 'SELECT' query statement with a 'Where' clause. In other words, The SELECT operator is unary; i.e., it is applied to a single relation. In addition, the selection operation is applied to every tuple individually; hence, selection conditions cannot involve more than one tuple.

Notation − σp(r).

Where (σ) sigma stands for selection predicate, r stands for relation and p for prepositional logic formula which may use connectors like and, or, and not. This may also use relational operators like − <, >, =, ≠, ≥, ≤.

For example – σsubject = "C++"(Book)

Output – selects tuples from Book where subject is 'C++'.

σsubject = "C++" and cost = "400" or year > "2018" (Book)

output - Selects tuples from Book where subject is 'C++' and 'cost' is 400 or those Book published after 2018.

Project Operation (∏)

Project operation is represted by(∏) pi symbol.It projects column(s) that satisfy a given predicate.

Notation − ∏B1, B2, B_N (r)

Where B1, B2 , B_N are attribute names of relation r.

Duplicate rows are automatically removed, as relation is a set.

For example:- ∏subject, author (Book)

The above example shows columns named as subject and author from the relation or table Book.

Union operation (∪)

The union operation result is denoted by R ∪ S, Union is a relation that includes all tuples that are either in R or in S or in both R and S. Duplicate tuples are removed. The relational union operator is equal to the SQL UNION operator.

Union operation performs binary union connecting two given relations and is denoted as -

R ∪ S.

In above notation **R** and **S** are either database relations or relation which include all tuples that are either in R or in S or in Both R and S. It also eliminated duplicate tuples.

For a valid union operation, the following criteria's must be present-

- **R** and **S** must have the same No. of attributes.
- Attribute domains must be compatible.
- Duplicate tuples are automatically removed.

∏ author (Book) ∪ ∏ author (Article).

Output: – It Projects the names of the authors who have either written a Book or an article or both.

Set Difference (−)

Set Difference is used in SQL in the form of the EXCEPT or MINUS operator. It is denoted by R-S. It includes all tuples that are in R but not in S.

Notation:- R - S

search all the tuples that are present in R but not in S.

∏author (Book) -∏author (Article)

Output – It Provides the name of the authors who have written Book but not an Article.

Cartesian Product (X)

The cartesian product of two relations is a join that is not restricted by any criteria, resulting in every tuple of the first relation being matched with every tuple of the second relation. The cartesian is used in SQL as the 'Cross Join' operator, which combines information of two different relations into one.

Notation:- R X S

Where R and S are relations and their result will be denoted as-

RXS={q t | q ∈ r and t ∈ s}.

σauthor = 'Totpoint' (Books X Articles).

Output − Yields a relation, which shows all the books and articles written by Totpoint.

Rename Operation (ρ)

The rename operation used to rename the output relation or give another name to a relation. It is denoted by small Greek letter rho (ρ).

Notation − ρ x (E)

Where the output of expression E is saved with the name of x.

3. Additional operations are −

- Set intersection
- Assignment
- Natural join

The equal sign (=) is used as assignment operator in SQL. It assigns a value to a variable, whereas natural join is a join operation that creates an implicit join clause for us based on the common columns in the two tables being joined. Common columns are columns that have the same name in both the tables.

Natural join can be an Inner join, a Left Outer join, or Right Outer join in RDBMSs.

4. Conclusions

Relational algebra and calculus are formal languages for the relational model of data. A relation is said to be a set of rows in a table with labeled columns. Relational algebra and its associated operations are the origin for SQL. These relational operations define algebra. Relational algebra is not common as it is possible to write programs producing queries that cannot be specified using relational algebra. Relational algebra can be specified using SQL syntax. The other, “more mathematical” notation came first and is used in research and other venues, but not commercially [10,12]. Relational Algebra and Relational Calculus both have similar demonstrative power. The main difference between them is just that Relational Algebra specify how to retrieve data and Relational Calculus defines what data is to be retrieved.

References

1. Datalog and Logic-Based databases, http://cs.wwc.edu/`aabyan/415/Datalog.html
2. Datalog, http://en.wikipedia.org/wiki/Datalog
3. Codd E. F.: A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387
4. Codd E. F.: A database sublanguage founded on the relational calculus, In Proceedings of the ACM-SIGFIDET Workshop, Data Description, Access, and Control (San Diego, Calif., Nov. ll-12). ACM, New York, 1971, pp. 35-68.
5. Codd E. F.: Relational completeness of data base sublanguages, In Courant Computer Science Symposia No. 6: Data Base Systems. Prentice-Hall, New York, 1972, pp. 67-101
6. Lacroix M., Pirotte A.: Domain-Oriented Relational Languages, VLDB 1977, pp. 370-378
7. Codd E. F.: Relational Completeness of Data Base Sub-languages, In R. Rustin, editor, Data Base Systems. Prentice Hall, 1972
8. Tuple relational calculus. Wikipedia, n.d. http://en.wikipedia.org/wiki/Tuple_relational_calculus, accessed Jan 2011.
9. https://www.brainkart.com/article/The-Relational-Algebra-and-Relational-Calculus_11419
10. Augustsson, L. and Ågren, M., 2016, September. Experience report: types for a relational algebra library. In *ACM SIGPLAN Notices* (Vol. 51, No. 12, pp. 127-132). ACM.
11. Hellings, J., Gyssens, M., Wu, Y., Van Gucht, D., Bussche, J.V.D., Vansummeren, S. and Fletcher, G.H., 2018. Comparing downward fragments of the relational calculus with transitive closure on trees. *arXiv preprint arXiv:1803.01390*.
12. Jananthan, H., Zhou, Z., Gadepally, V., Hutchison, D., Kim, S. and Kepner, J., 2017, December. Polystore mathematics of relational algebra. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 3180-3189). IEEE.

