IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11119

Exploration of Relational Database, Query Language, Relational Algebra and its operations

Barkha Gupta

Programme Assistant, Department of Computer, Agriculture University, Jodhpur, Rajasthan, India

Abstract: DBMS acronym of Database Management System comprises of a collection of inter related data. It is a set of program or software that help to access the stored data in a very easy and efficient manner. Database are generally used to design and to manage large amount of data in a very systematic manner. The management of stored information includes both the definition of structures for the storage of information and provision mechanism for manipulation of this information. This information is stored permanently in various files. Various application software is written to fetch or extract the information from these files as and when required. DBMS removes the data redundancy and inconsistency among the data. DBMS also provides different levels of security to the data. It also retrieves the data form the database as per the criteria specified by the user.

Keywords: RDBMS, DBMS, Relational Algebra, Relational Calculus, Procedural Language, No-Procedural Language, Query Language

I. INTRODUCTION

RDBMS acronym of Relational Database Management System. It is the body of related information stored in two-dimensional manner. It means the data are stored in tabular form, consisting rows and columns. There are other forms of database management system which are also available that are Network Database Management System, Hierarchical Database Management System and Object-Oriented Database Management System. Network Database Management System consist a collection of records connected with each other through links. Each record is a collection of attributes. Hierarchical Database Management System consists a collection of records that are connected to each other through a link. It is tree like structure which follows top-down approach. The link in this is associated with two records. It forms a parent child relationship between the records. Object-Oriented Database Management System consists of classes and object associated with it. It is used for description of data and their inter-relationships in accordance with the concept.

II. RELATIONAL DATABASE

Relational database is a database system that is collection of programs which enables user to create, manipulate, update, delete and modify the data in the relational structure. Relational database store data in tabular format and it merely looks like table. Relation is the other name of table which is collection of rows and columns. Row here represent one or complete set of information. Row is horizontal in nature. Row is also known as tuple in the relational database terminology. Column contains one type of information. Column in the relational database terminology is also known as field. Column is vertical in nature. Relational database system is one of the most commonly used database system in the various organization for storing the data as well as accessing the data via query using Structured Query Language (SQL).

Product_Code	Product_Name	Product_Description	Manufacturing_Date	Product_Price
101	Shampoo	For oily hair	11/02/2021	Rs. 500/-
102	Cream	Moisturizer Cream	05/05/2020	Rs. 1056/-
103	Oil	Hair oil	08/07/2021	Rs. 899/-

Column/Field

Row/Tuple

Here, in the above table Product_Code, Product_Name, Product_Description, Manufacturing_Date and Product_Price represents the columns or fields.

DOI: 10.17148/IJARCCE.2022.11119

III. QUERY LANGUAGE

Query implies questioning or requesting information as per needed. A query language is any computer-based programming language that requests and retrieves data from information systems or database by sending queries. It works on user entered structured and formal programming command-based queries to find the data as per user needs from host databases. Query language is initially created for creating, accessing, modifying and deleting the data into the database. Query language required user to write an English like command generally known as syntax to fetch or access the data from the database. For example: -

SELECT * FROM EMPLOYEE;

This command written in SQL (Structured Query Language) is a syntax which is used to access all the data from the database named employee.

Query language is further classified into two types: -

(i) Procedural Language

(ii) Non-Procedural Language

IV. PROCEDURAL LANGUAGE

Procedural language is a language which is written in proper sequence. In procedural language user has to define “what is” his requirements which what has to be done and user has also defined “How” that task will going to be performed. Procedural language is a command driven language. Generally, the size of the program is large here as user has to define both “what is to be done” and how that will be done”. Procedural language is a step-by-step procedure that is written to solve a particular or specific task which is executed in sequential manner. The overall efficiency of this language is high. Procedural language works with the state of machine. Procedural language returns restricted datatype and certain allowed values only. This language is not suitable for those applications where time is the major and critical constraint. The iterative loops and recursive calls are used while working in procedural languages.

Examples of procedural languages include Relational Algebra, BASIC, ALGOL, C, C++, Java, COBOL, FORTRAN and Pascal.

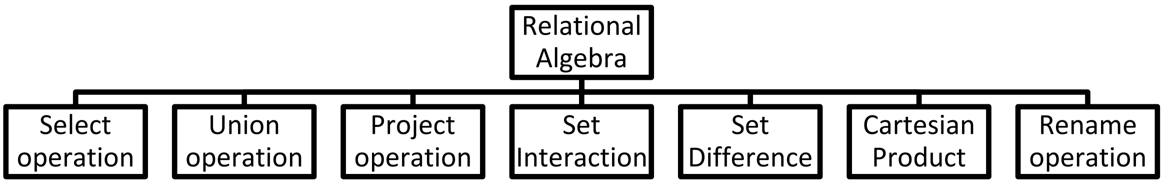
V. NON-PROCEDURAL LANGUAGE

Non-Procedural language is a language in which proper sequence is not mandate. In this language user has only to define “What to do” and neither is no requirement to mention “how that will be done”. It is also known as applicative or functional language. Its programming structure and semantics are simpler than procedural language. Generally, the size of the program is not that large as in procedural language as here user has to define only “what is to be done” instead of how that will be done”. It works with the help of mathematical functions. It has ability to return and datatype and sort of value available in database. It is also known as function-driven language. The overall efficiency of non-procedural language is low in comparison to procedural language. It is also well-suited with the applications where time is a major and critical constraint.

Examples of non-procedural languages include Relational Calculus, LISP, SQL, PROLOG.

VI. RELATIONAL ALGEBRA

Relational Algebra is a procedural language which specifies step by step procedure to generate the output. Relational Algebra takes relation as an input and operates as per the instructions or program and generate a relation in form of output. It uses operators to perform query and generate the result. The operators can be unary operator and binary operator. Relational Algebra is performed recursively to generate the result. The fundamental operations performed in Relational Algebra are as follows: -



DOI: 10.17148/IJARCCE.2022.11119

(i) Select Operation-

Select operation is used to fetch the desired or required information from the database which satisfy the given predicate. Select operation is denotes by σ symbol.

Notation – σ p(r)

σ implies selection prediction

p implies propositional logic which used connectors like AND, OR, NOT.

r implies relation/table

Example: - Table name Product

Product_Code	Product_Name	Product_Description	Manufacturing_Date	Product_Price
101	Shampoo	For oily hair	11/02/2021	Rs. 500/-
102	Cream	Moisturizer Cream	05/05/2020	Rs. 1056/-
103	Oil	Hair oil	08/07/2021	Rs. 899/-
104	Cream	Fuit Cream	12/08/2021	Rs. 211/-
105	Shampoo	For dry hair	07/12/2021	Rs. 800/-

Select Operation-

σ Product_Name ="Cream" (Product)

This implies it will display all the Product_Name having cream in its field from the product database.

So, the output will be

Product_Code	Product_Name	Product_Description	Manufacturing_Date	Product_Price
102	Cream	Moisturizer Cream	05/05/2020	Rs. 1056/-
104	Cream	Fuit Cream	12/08/2021	Rs. 211/-

(ii) Union Operation-

Union operation is a binary operation which is used to combine the data of two tables and it eliminates the duplicate tuples. Union operation is denoted by ∪ symbol. The mandatory condition to union two relation is that both the relation must have same number of attributes.

Notation − r U s

R implies relation 1

S implies relation 2

Example: - Table name Product

Product_Code	Product_Name
101	Shampoo
102	Cream
103	Oil
104	Cream
105	Shampoo

Table name Grocery

Product_Code	Product_Name
201	Cabbage
202	Potato
203	Orange
104	Cream
105	Shampoo

Union Operation- ∏ Product_Code (Product) ∪ ∏ Product_Code (Grocery)

This implies it will display all the Product_Code from the product database and Grocery database and automatically elimination the duplicate data.

So, the output will be

Product_Code
101
102
103
104

105
201
202
203

(iii) Project Operation-

Project operation is used to displays the set of attributes which satisfy the given query. Project operation is denoted by ∏ symbol.

Notation $\prod_{A1}$, A_2, A_n (r)

A_1, A_2, A_n are the attributes

r is the relation

Example: - Table name Product

Product_Code	Product_Name	Product_Description	Manufacturing_Date	Product_Price
101	Shampoo	For oily hair	11/02/2021	Rs. 500/-
102	Cream	Moisturizer Cream	05/05/2020	Rs. 1056/-
103	Oil	Hair oil	08/07/2021	Rs. 899/-
104	Cream	Fuit Cream	12/08/2021	Rs. 211/-
105	Shampoo	For dry hair	07/12/2021	Rs. 800/-

Project operation - ∏ Product_Name, Product_Description (Product)

This implies it will display all the Product_Name and Product_Description from the product database and remove other attributes from the database Product.

So, the output will be

Product_Name	Product_Description
Shampoo	For oily hair
Cream	Moisturizer Cream
Oil	Hair oil
Cream	Fuit Cream
Shampoo	For dry hair

(iv) Set Interaction-

Set Interaction is a relation operation which is binary in nature that means it applied on two relations. It will return all the tuples that are available in both relation or rows that are common in both relations. Set Interaction is denoted by ∩ symbol.

Notation: R ∩ S

R implies relation named 1

S implies relation named 2

Example Table name Product

Product_Code	Product_Name
101	Shampoo
102	Cream
103	Oil

Table name Grocery

Product_Code	Product_Name
201	Cabbage
202	Potato
203	Orange
104	Cream
105	Shampoo

Set Interaction:- ∏ Product_Name (Product) ∩ ∏ Product_Name (Grocery)

DOI: 10.17148/IJARCCE.2022.11119

This implies it will display all the Product_Name from both the relation product and Grocery having common values.
So, the output will be

Product_Name
Cream
Shampoo

(v) Set Difference-

Set Difference is a relation operation which is binary in nature that means it applied on two relations. It will return all the tuples that are available in first relation and not available in second relations. Set Interaction is denoted by - symbol.
Notation: R - S
R implies relation named 1
S implies relation named 2
Example Table name Product

Product_Code	Product_Name
101	Shampoo
102	Cream
103	Oil

Table name Grocery

Product_Code	Product_Name
201	Cabbage
202	Potato
203	Orange
104	Cream
105	Shampoo

Set Difference ∏ Product_Name (Product) - ∏ Product_Name (Grocery)
This implies it will display all the Product_Name that is available in product databse and not available in grocery database.
So, the output will be

Product_Name
Oil

(vi) Cartesian Product-

Cartesian product is a binary operation which is applied on two relations. It is used to combine each tuple in one table with each tuple in the other table. It is denoted by X symbol.
Notation: R X S
R implies relation named 1
S implies relation named 2
Example Table name Product

Product_Code	Product_Name
101	Shampoo
102	Cream
103	Oil

Table name Grocery

Item_Code	Item_Name
201	Cabbage
202	Potato
203	Orange

Cartesian: - Product X Grocery
This implies it will display all the tuples with cross multiplying with both the relation Product and Grocery.
So, the output will be

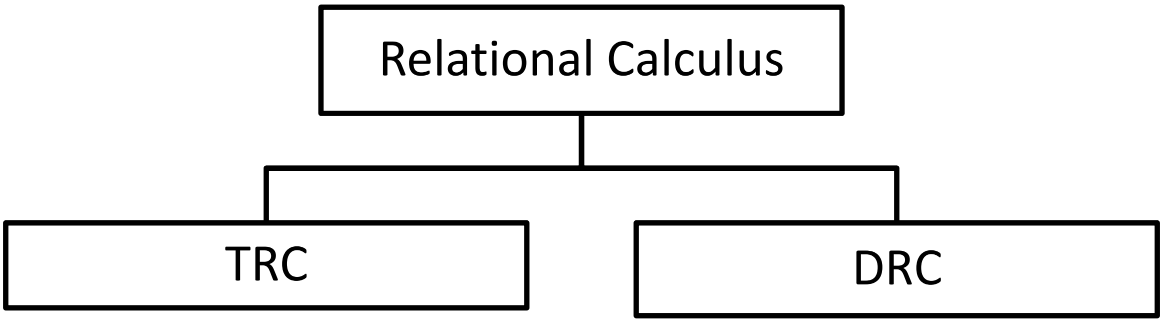
Product_Code	Product_Name	Item_Code	Item_Name
101	Shampoo	201	Cabbage
101	Shampoo	202	Potato
101	Shampoo	203	Orange
102	Cream	201	Cabbage
102	Cream	202	Potato
102	Cream	203	Orange
103	Oil	201	Cabbage
103	Oil	202	Potato
103	Oil	203	Orange

(vii) Rename Operation-

Rename operation is used to change the name of the output relation. Rename' operation is denoted with small Greek letter rho ρ.
Notation – ρ x (E)
x is the new name of the relation
E is the result set
Example: ρ(New_Product, Product)
This implies the relation’s name has been changed from Product to New_Product.

VII. RELATIONAL CALCULUS

Relational calculus is a non-procedural language in which user only define what is required by him and not specifies how to attain that information.



(i) TRC (Tuple Relational Calculus) –

The tuple relational calculus is specified to select or extract the tuples/rows in a relation. In TRC, filtering variable uses the tuples of a relation. The result of the relation can have one or more tuples. It provides only the description of the query but it does not provide the methods or procedure to solve it. Thus, it explains what to do but not how to do. It also uses Existential (∃) and Universal Quantifiers (∀) to bind the variable.

In Tuple Calculus, a query is expressed as
{T | P (T)} or {T | Condition (T)}
T implies resulting tuple
P(T) is the condition to fetch
Example {t| t ∈ Product ∧ t[Product_Price]>=850}
Output will be

Product_Code	Product_Name	Product_Description	Manufacturing_Date	Product_Price
102	Cream	Moisturizer Cream	05/05/2020	Rs. 1056/-
103	Oil	Hair oil	08/07/2021	Rs. 899/-

(ii) DRC (Domain Relational Calculus)-

The second form of relation is known as Domain relational calculus. In domain relational calculus, filtering variable uses the domain of attributes. Domain relational calculus uses the same operators as used by tuple calculus. It uses logical connectives ∧ (and), ∨ (or) and ┐ (not). It also uses Existential (∃) and Universal Quantifiers (∀) to bind the variable.

Notation: {a1, a2, a3, ..., an | P (a1, a2, a3, ... ,an)}

Where a1, a2, a3 are attributes.

P stands for formula built by inner attributes

Example

{< Product_Name, Product_Description, Product_Price > | ∈ Product ∧ Product_Name = 'Cream'}

Product_Name	**Product_Description**	**Product_Price**
Cream	Moisturizer Cream	Rs. 1056/-
Cream	Fuit Cream	Rs. 211/-

VIII. CONCLUSION

Database is a far more efficient mechanism to store and organize data than spreadsheets. It allows for a centralized facility that can easily be modified and quickly shared among multiple users.

Relational databases store data in tables. Tables can grow large and have a multitude of columns and records. Relational database management systems (RDBMSs) use SQL (and variants of SQL) to manage the data in these large tables. The RDBMS you use is your choice and depends on the complexity of your application. The database once created is permanent in nature and can be used several times in different manner and different point of time. Database can be shared among different set of users with various level of security given by the administrator. It usage is vast and dynamic in nature and keeps on changing over time.

IX. ACKNOWLEDGEMENT

Author is grateful to Agriculture University, Jodhpur for encouraging writing in my field.

X. REFERENCES

[1] P. S. Gill, “Database Management System”, published by I. K. International Publishing House Private Limited, 2010

[2] ITL Education Solutions Limited, “Introduction to Database Systems”, published by Pearson, 2010

[3] Ramez Elmasri, Shamkant B. Navathe “Database Systems-Models, Languages, Design and Application Programming”, Pearson Education India, Sixth Edition.

[4] Gerard Blokdyk, “RDBMS Relational Database Management System: A User Guide”, Published by Createspace Independent Publishing Platform, 2017.

[5] Ramon A. Mata-Toledo, Pauline K. Cushman, “Schaum’s Outline of Fundamentals of Relational Databases”, Published by McGraw Hill Education, 2000.

[6] Dr. Madhavi Vaidya, “RDMBS in Depth - Mastering SQL and PL/SQL Concepts, Database Design, ACID Transactions, and Practice Real Implementation of RDBM”, Published by BPB PUBN, 2021.

[7] C. J. Date, “Database Design and Relational Theory: Normal Forms and All That Jazz”, Published by Apress, Second edition, 2019.

[8] José Oliveira, Michael Winter, Wolfram Kahl, “Relational and Algebraic Methods in Computer Science”, Published by Springer International Publishing, 2015.