



# Tuple Relational Calculus (TRC) in DBMS

| India\_1998

[Read](#)

[Discuss](#)

[Courses](#)

[Video](#)

**Tuple Relational Calculus (TRC)** is a non-procedural query language used in relational database management systems (RDBMS) to retrieve data from tables. TRC is based on the concept of tuples, which are ordered sets of attribute values that represent a single row or record in a database table.

TRC is a declarative language, meaning that it specifies what data is required from the [database](#), rather than how to retrieve it. TRC queries are expressed as logical formulas that describe the desired tuples.

**Syntax:** The basic syntax of TRC is as follows:

$$\{ t \mid P(t) \}$$

where  $t$  is a **tuple variable** and  $P(t)$  is a **logical formula** that describes the conditions that the tuples in the result must satisfy. The **curly braces**  $\{ \}$  are used to indicate that the expression is a set of tuples.

Employee ID
Name
Salary
Department ID

To retrieve the names of all employees who earn more than \$50,000 per year, we can use the following TRC query:

$$\{ t \mid \text{Employees}(t) \wedge t.\text{Salary} > 50000 \}$$

In this query, the “Employees(t)” expression specifies that the tuple variable t represents a row in the “Employees” table. The “ $\wedge$ ” symbol is the logical AND operator, which is used to combine the condition “t.Salary > 50000” with the table selection.

The result of this query will be a set of tuples, where each tuple contains the Name attribute of an employee who earns more than \$50,000 per year.

TRC can also be used to perform more complex queries, such as joins and nested queries, by using additional logical operators and expressions.

While TRC is a powerful query language, it can be more difficult to write and understand than other SQL-based query languages, such as [Structured Query Language \(SQL\)](#). However, it is useful in certain applications, such as in the formal verification of database schemas and in academic research.

Tuple Relational Calculus is a **non-procedural query language**, unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do it.

## Tuple Relational Query

In Tuple Calculus, a query is expressed as

$$\{t \mid P(t)\}$$

where  $t$  = resulting tuples,

$P(t)$  = known as Predicate and these are the conditions that are used to fetch  $t$ . Thus, it generates a set of all tuples  $t$  such that Predicate  $P(t)$  is true for  $t$ .

Aptitude   Engineering Mathematics   Discrete Mathematics   Operating System   DBMS   Computer Networks

It also uses quantifiers:

$\exists t \in r (Q(t))$  = "there exists" a tuple in  $t$  in relation  $r$  such that predicate  $Q(t)$  is true.

$\forall t \in r (Q(t))$  =  $Q(t)$  is true "for all" tuples in relation  $r$ .

## Domain Relational Calculus (DRC)

[Domain Relational Calculus](#) is similar to Tuple Relational Calculus, where it makes a list of the attributes that are to be chosen from the relations as per the conditions.

$$\{ \langle a_1, a_2, a_3, \dots, a_n \rangle \mid P(a_1, a_2, a_3, \dots, a_n) \}$$

where  $a_1, a_2, \dots, a_n$  are the attributes of the relation and  $P$  is the condition.

## Tuple Relational Calculus Examples

Table Customer

Customer name	Street	City
Saurabh	A7	Patiala
Mehak	B6	Jalandhar

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Customer name	Street	City
Ria	A5	Patiala

### Table Branch

Branch name	Branch City
ABC	Patiala
DEF	Ludhiana
GHI	Jalandhar

### Table Account

Account number	Branch name	Balance
1111	ABC	50000
1112	DEF	10000
1113	GHI	9000
1114	ABC	7000

### Table Loan

Loan number	Branch name	Amount
L33	ABC	10000
L35	DEF	15000

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Loan number	Branch name	Amount
L49	GHI	9000
L98	DEF	65000

### Table Borrower

Customer name	Loan number
Saurabh	L33
Mehak	L49
Ria	L98

### Table Depositor

Customer name	Account number
Saurabh	1111
Mehak	1113
Suniti	1114

**Example 1:** Find the loan number, branch, and amount of loans greater than or equal to 10000 amount.

$$\{t \mid t \in \text{loan} \wedge t[\text{amount}] \geq 10000\}$$

Resulting relation:

Loan number	Branch name	Amount
L33	ABC	10000
L35	DEF	15000
L98	DEF	65000

In the above query,  $t[\text{amount}]$  is known as a tuple variable.

**Example 2:** Find the loan number for each loan of an amount greater or equal to 10000.

$$\{t \mid \exists s \in \text{loan}(t[\text{loan number}] = s[\text{loan number}] \wedge s[\text{amount}] \geq 10000)\}$$

Resulting relation:

Loan number
L33
L35
L98

**Example 3:** Find the names of all customers who have a loan and an account at the bank.

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}]) \wedge \exists u \in \text{depositor}(t[\text{customer-name}] = u[\text{customer-name}])\}$$

Resulting relation:

Customer name
---------------

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Customer name
Mehak

**Example 4:** Find the names of all customers having a loan at the “ABC” branch.

$$\{t \mid \exists s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}] \wedge \exists u \in \text{loan}(u[\text{branch-name}] = \text{“ABC”} \wedge u[\text{loan-number}] = s[\text{loan-number}]))\}$$

Resulting relation:

Customer name
Saurabh

Last Updated : 19 Jun, 2023

98

## Similar Reads



Difference between Tuple Relational Calculus (TRC) and Domain Relational Calculus (DRC)



Difference between Relational Algebra and Relational Calculus



Domain Relational Calculus in DBMS



Introduction of Relational Algebra in DBMS



Relational Model in DBMS



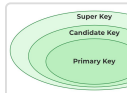
Introduction of Relational Model and Codd Rules in DBMS



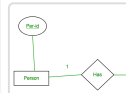
Difference between E-R Model and Relational Model in DBMS



Tuple in DBMS



Types of Keys in Relational Model (Candidate, Super, Primary, Alternate and Foreign)



Mapping from ER Model to Relational Model

Previous

Next

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

## Article Contributed By :

India\_1998

India\_1998

Follow

## Vote for difficulty

Current difficulty : [Easy](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [Nityananda Gohain](#), [VijaykumarSakhare](#), [simmytarika5](#), [5758rushistng](#), [adityaarxehn](#)

Article Tags : [dbms](#), [DBMS](#), [GATE CS](#)

Practice Tags : [DBMS](#), [DBMS](#)

Improve Article

Report Issue



A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).





## Company

About Us  
Legal  
Careers  
In Media  
Contact Us  
Advertise with us

## Explore

Job-A-Thon Hiring Challenge  
Hack-A-Thon  
GfG Weekly Contest  
Offline Classes (Delhi/NCR)  
DSA in JAVA/C++  
Master System Design  
Master CP

## Languages

Python  
Java  
C++  
PHP  
GoLang  
SQL  
R Language  
Android Tutorial

## DSA Concepts

Data Structures  
Arrays  
Strings  
Linked List  
Algorithms  
Searching  
Sorting  
Mathematical  
Dynamic Programming

## DSA Roadmaps

DSA for Beginners  
Basic DSA Coding Problems  
Complete Roadmap To Learn DSA  
DSA for FrontEnd Developers  
DSA with JavaScript  
Top 100 DSA Interview Problems  
All Cheat Sheets  
DSA Roadmap by Sandeep Jain

## Web Development

HTML  
CSS  
JavaScript  
Bootstrap  
ReactJS  
AngularJS  
NodeJS  
Express.js  
Lodash

[GATE CS Notes](#)[Operating Systems](#)[Computer Network](#)[Database Management System](#)[Software Engineering](#)[Digital Logic Design](#)[Engineering Maths](#)

## Data Science & ML

[Data Science With Python](#)[Data Science For Beginner](#)[Machine Learning Tutorial](#)[Maths For Machine Learning](#)[Pandas Tutorial](#)[NumPy Tutorial](#)[NLP Tutorial](#)[Deep Learning Tutorial](#)

## Competitive Programming

[Top DSA for CP](#)[Top 50 Tree Problems](#)[Top 50 Graph Problems](#)[Top 50 Array Problems](#)[Top 50 String Problems](#)[Top 50 DP Problems](#)[Top 15 Websites for CP](#)

## Interview Corner

[Company Wise Preparation](#)[Preparation for SDE](#)[Experienced Interviews](#)[Internship Interviews](#)[Competitive Programming](#)[Python Programming Examples](#)[Django Tutorial](#)[Python Projects](#)[Python Tkinter](#)[OpenCV Python Tutorial](#)[Python Interview Question](#)

## DevOps

[Git](#)[AWS](#)[Docker](#)[Kubernetes](#)[Azure](#)[GCP](#)

## System Design

[What is System Design](#)[Monolithic and Distributed SD](#)[Scalability in SD](#)[Databases in SD](#)[High Level Design or HLD](#)[Low Level Design or LLD](#)[Top SD Interview Questions](#)

## GfG School

[CBSE Notes for Class 8](#)[CBSE Notes for Class 9](#)[CBSE Notes for Class 10](#)[CBSE Notes for Class 11](#)[CBSE Notes for Class 12](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

## Commerce

Accountancy  
Business Studies  
Economics  
Management  
Income Tax  
Finance  
Statistics for Economics

## SSC/ BANKING

SSC CGL Syllabus  
SBI PO Syllabus  
SBI Clerk Syllabus  
IBPS PO Syllabus  
IBPS Clerk Syllabus  
Aptitude Questions  
SSC CGL Practice Papers

## UPSC

Polity Notes  
Geography Notes  
History Notes  
Science and Technology Notes  
Economics Notes  
Important Topics in Ethics  
UPSC Previous Year Papers

## Write & Earn

Write an Article  
Improve an Article  
Pick Topics to Write  
Write Interview Experience  
Internships

@geeksforgeeks , Some rights reserved