International Journal of Advanced Computer Research
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
Volume-5 Issue-20 September-2015

Dynamic fragmentation and query translation based security framework for distributed databases

Arunabha Sengupta[*]
Solution Architect at Tata Consultancy Services Limited, Kolkata, WB, India

Received: 30-June-2015; Revised: 05-August-2015; Accepted: 12-August-2015

Abstract

The existing security models for distributed databases suffer from several drawbacks viz. tight coupling with the choice of database; lack of dynamism, granularity and flexibility; non scalability and vulnerability to intrusion attacks. There is a lack of an integrated flexible and interoperable security framework that can dynamically control access to table, row, column and field level data entity. The objective of this proposed framework is to address the issue of security in distributed query processing using the dynamic fragmentation and query translation methodologies based on a parameterized security model which could be tailored based on the business requirements to take care of relational level, record level, column level as well as the atomic data element level security and access requirements. This solution has been implemented and tested for DML operations on distributed relational databases and the execution results are found to be very promising in terms of restricting access to data elements with higher security clearance; blocking queries that return data at/below user's level but its evaluation requires accessing columns/rows with higher security clearance; and blocking aggregate queries used for inferring classified information.

Keywords

Bell–LaPadula model, Database Security, Discretionary Access Control, Distributed Database Management System, Dynamic Fragmentation, Mandatory Access Control, Object Level Security, Operational Level Security, Role Based Access Control, Site Level Security, Subject Level Security.

[*]Author for correspondence

1. Introduction

Security is a paramount aspect which must be taken into account from the very first steps of the design process of distributed databases, especially in security critical environments. It is therefore important that the security considerations and requirements viz. availability, integrity and confidentiality are taken into account as main design objectives when working with a distributed database system. This is necessary, especially for applications where large amounts of highly sensitive data need to be persisted and processed. For these applications it is especially important that the distributed database management systems (DDBMS) should operate in a secure manner. The DDBMS should allow users to access the database consisting of data at a variety of sensitivity level without compromising on security i.e. data availability to be governed by user's authorized security level.

2. Distributed DBMS

2.1. Distributed Database Model

A distributed database appears to a user as a single database but is, in fact, a set of databases stored on multiple computers that may be geographically spread apart. The data on several computers can be simultaneously accessed and modified using a network. Each database server in the distributed database is controlled by its local DBMS, and each co-operates to maintain the consistency of the global database. A database user accesses the distributed database through –

- Local applications - applications which do not require data from other sites.
- Global applications - applications which do require data from other remote sites.

The following diagram (Figure 1) illustrates the model of a typical distributed DBMS. Essentially, there are two distinct layers – Global and Local. The below proposed dynamic security framework takes care of the security requirements at both the layers based on a Role Based Access Control (RBAC) coupled with Operation, Site, Subject and Object Security level categorization.

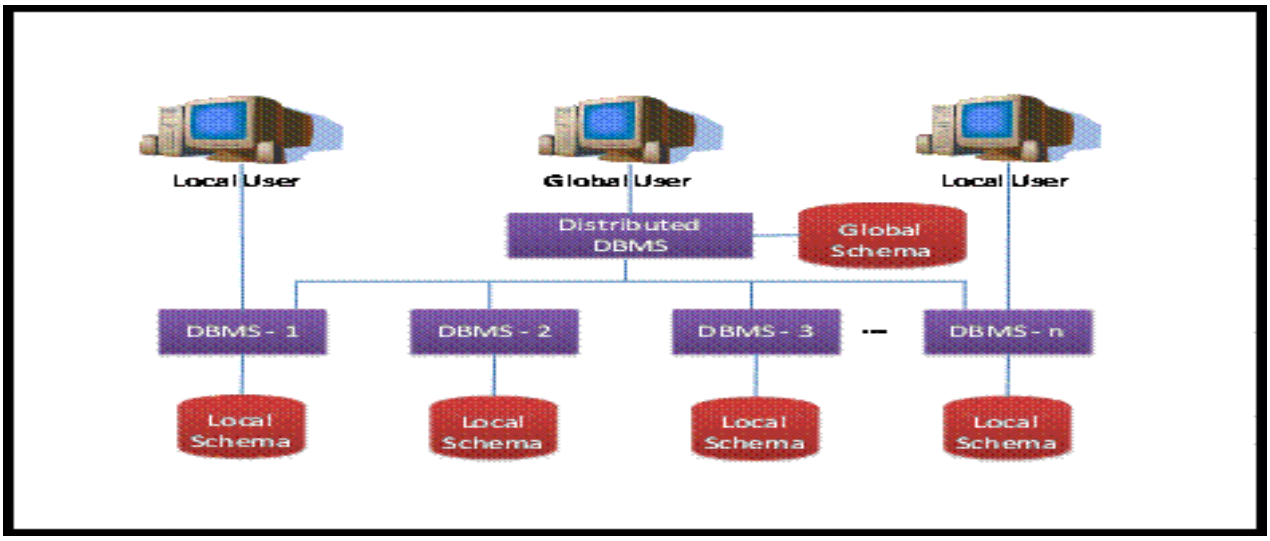


Figure 1: Distributed Database Topology

2.2. Distributed DBMS Architecture

A typical architecture diagram of a Distributed DBMS using a Global Conceptual Schema is shown in the below figure (Figure 2).

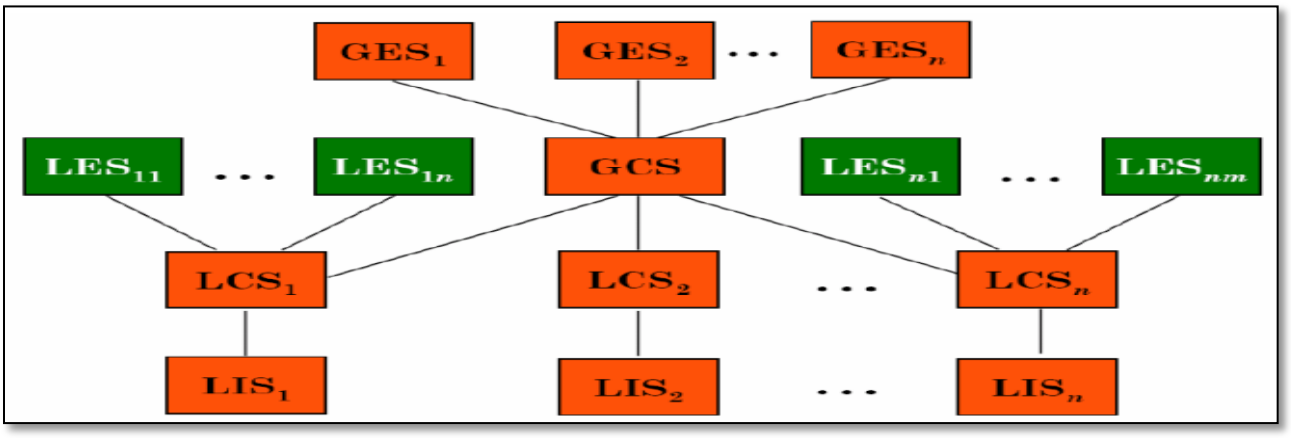


Figure 2: Distributed Database Architecture

Following are the high level abstractions of a distributed DBMS –

- Local internal schema (LIS) – It describes the local physical data organization.
- Local conceptual schema (LCS) - It describes the logical data organization at each site.
- Global conceptual schema (GCS) - It describes the global logical view of the data (Union of the LCSs).
- External schema (ES) - It describes the user/application view on the data.

There are two sets of users accessing the distributed DBMS – local users and global users. Each local user

can only access its own local DBMS using the corresponding LES and not allowed to access the other local DBMSs. This is ensured by the use of a Security Role Translation Table described in later sections. However a global user can access all the underlying local DBMSs remotely using the Global External Schema (GES) based on the role level authorization, access privileges and site security permissions of the local DBMSs.

The below proposed model for Dynamic Fragmentation and Query Translation caters to an integrated flexible and scalable security framework that is interoperable and allows to configure Operational, Site, Subject and Object level security definitions by which user access to data can be dynamically controlled to schema, table, row, column and individual data entity levels. It is based on Bell - La Padula model and can be applied on top of the RBAC mechanism.

3. Related Works

3.1. Survey of Existing Security Models

Various researchers have examined the underlying features of distributed database architecture in terms of security. Gupta et al. [5] in their paper "Concurrency Control and Security issues of Distributed Databases Transaction" dated Aug, 2012 have reviewed the coverage of concurrency control and security in distributed systems. The paper describes the four main security components of a distributed database viz. security authentication, authorization, access control and encryption and the components of a DDBMS, which include

- Distributed query processor (DQP) handles distributed queries
- Distributed transaction manager (DTM) processing Distributed transactions.
- Distributed metadata manager (DMM) for managing distributed metadata.
- Distributed integrity manager (DIM) for enforcing integrity constraints.
- Distributed security manager (DSM) for enforcing security constraints.

Kose [8] has reviewed the topic of security in distributed databases and proposed solutions to some of the commonplace security concerns. The paper points out that authentication, identification and enforcing appropriate access controls as the major issues in security. An extensive investigation on the inference problem for distributed database systems based on processing security constraints in a multilevel secure distributed database system has been reviewed. Miklau [10] in his 2005 paper describes some novel techniques to manage security in distributed databases. One such technique is the Active Protection mechanism which is a three step process –

- A subject s_i, after authenticating, posts a request for access to the database, in the form of their query q_i.
- For each such request, the owner determines whether the query can be answered, given the subject's access rights.
- If admissible, the server executes the query and returns the results to the subject.

Another mechanism described is "View Materialization and Publishing" in which authorized views are computed and transmitted to subjects in anticipation of actual requests. Coy [3] has discussed database security issues in general and how the database model affects database system security in particular. The security strengths and weaknesses of the object-oriented as well as relational database models have been evaluated and special problems found in the distributed environment are discussed. Remarkable work has been done by Batra and Aggarwal [1] in their 2012 paper in the field of distributed database security based on Access Control List. Resource is available only if owner assigns the access rights to the user. Their scheme uses three different security levels. On each level, user is verified whether he can access the resource at that particular security level or not.

- Security Level 1: Access is granted to the authorized user and as per privileges given, user can access the specific resources. User can also allow other users to access data by granting those privileges at a predefined security level.
- Security Level 2: Users cannot execute any DDL statement without permission of DBA or the privileged user who is authorized to further grant the access permission. It prevents the unauthorized modification in table structures.
- Security Level 3: Users cannot execute any DCL statement without permission for any specific transaction. It prevents the unauthorized control over a specific transaction.

The concepts of MAC and DAC for Distributed Database are instrumental in achieving Distributed Database Security as suggested by Lunt and Fernandez [9] along with the novel concepts of separation of duty and constraint based RBAC Model put forward by Crampton [4] in his 2003 paper. The latter have proposed a simple set-based specification scheme for authorization constraints in role-based access control systems and also suggested an enforcement model for a restricted subset of this scheme. Another method of privacy protection in distributed database is through fragmentation of records – vertical and horizontal as proposed by Narmada et al. [11].

In line with the their research of 2011, this paper proposes a dynamic fragmentation approach of records to take care of the intrusion attacks based on data aggregation results. In 2012, a technique of Vertical Fragmentation termed as "Vertical Splitting Algorithm" was suggested by Bhaskar and Sharma [2] which uses attribute affinity matrix and Bond Energy algorithm. Attribute affinity matrix is an m x m matrix for the m-attribute problem whose (i,j) element equals the "between attributes" affinity which is the total number of accesses of transactions referencing both attributes i and j. The designing of distribution involves making decisions on the fragmentation and placement of data across the sites of a computer network. The first phase of the distribution design in a top-down approach is the fragmentation phase, which is the process of clustering into fragments the data accessed simultaneously by applications. The fragmentation phase is then followed by the allocation phase, which handles the physical storage of the generated fragments among the nodes. The vertical fragmentation is executed in following manner –

- When a query uses attribute from a relation its value is true
- Information about databases and query are notified before fragmentation process.
- Query consists of attributes.

The use of attribute means accessing to the value of an existing attribute without any side-effect. The fragmentation discussed is composed of attribute fragmentation. Emphasis has been given to handle inference attacks especially Indirect Attacks like Statistical Inference through the proposed security model using mitigation control steps like Query Control [6] and Query Modification and Restriction [12] techniques. In his 2009 paper, Hylkema [6] has attempted a thorough coverage of the advancements in methods of inference attack detection and prevention. Query controls parse either the incoming query, the results from the execution of the query or both. If the query does not conform to a set of standards or it is deemed that an inference could be made from a combination of records requested, the query may be denied. Likewise, if the results of a query are such that an inference may be made, the query may be denied. Complex statistical methods are used to determine the likelihood that an inference is made. The two fundamental ways to pre-process are to either check the query before it is executed or to do so after query execution. In Query Modification technique, a query submitted by a user is modified to include further restrictions as determined by the user's authorization. Query Restriction ensures that all data used in the process of evaluating the query is dominated by the level of the user, and therefore prevents inferences. To this end, the system can either modify the user query such that the query involves only the authorized data or simply abort the query.

The steps proposed for a secure distributed database design by Khair et al. [7] has been extended further in this paper through the dynamic security framework which shifts the responsibility of security implementation from design time to execution time. The aim of their research has been to describe a step-by-step methodology for the design of a secure distributed medical database system. The methodology is based on the combination of mandatory and discretionary security approaches and uses hierarchies of user roles, data sets and sites in order to decide the secure distribution of the application. In this methodology, they have used an extension of the combination of the "best-fit" and the "all beneficial sites" methods for the allocation and replication of data, respectively. In the "best fit" approach, a measure is associated with each possible allocation and the site with the best measure is selected. In the "all beneficial sites" approach, the set of all sites where the benefit of allocating one copy of the fragment is higher than the cost is determined and a copy of the fragment to each element of this set is allocated.

Last but not the least, the concepts of Secure Database Query processing as proposed by Thuraisingham [13] and [14] has been referred while

implementation of the dynamic fragmentation algorithm. The former paper [13] focuses on secure query processing in a DDBMS. And implementation of secure query processing algorithms in a DDBMS as well as an analysis of the performance of the algorithms is described. The paper assumes that the security level of a relation is the security level of the user who creates it. A relation R classified at level L can have tuples at levels which dominate L. In other words, there could be a version of relation R at a level L^* which dominates L. Therefore, corresponding to an unclassified relation, there could be versions of that relation at Secret and Top Secret levels. In addition, each version at a security level could be fragmented at the same level. Furthermore, a fragment could also be replicated. When a user enters a tuple, the tuple is stored in a fragment of the relation at the security level of the user. If the user's security level is dominated by the security level of the relation, then the tuple is not entered. The tuples could be poly-instantiated.

However, within a security level, the primary key constraint cannot be violated. Each trusted DBMS is capable of creating a view of a relation at the security level of the querying subject. The distributed processing components merge the various views created by the individual trusted DBMSs in order to obtain a single global view at the security level of the querying subject. This paper views the privacy problem as a form of inference problem. The later paper [14] provides an overview of the privacy problem and then introduces the notion of privacy constraints. It describes architecture for a privacy-enhanced database management system and discusses algorithms for privacy constraint processing. Privacy constraints are rules, which assign privacy levels to the data. The paper then describes an integrated approach to constraint processing. That is, some constraints are handled during query processing, some during database updates, and some during database design. It then describes the design of two prototypes, which process constraints during query and update operations and the design of a database design tool. Finally it shows the query processor, update processor, and the database design tool could be combined so that an integrated solution to constraint processing can be achieved.

Several examples and scenarios have been taken up towards the end of this paper that demonstrates the effectiveness of the proposed security framework.

The existing security models for distributed databases suffer from following drawbacks –

- Lack of dynamism: Most of the existing distributed database security frameworks are pre-built based on the design time security considerations and hence fail to address the dynamic security requirements which varies from one user query to another.
- Tightly coupled with the choice of database: Majority of the security mechanisms are flawed since they are dependent on the choice of database – Relational or Object Oriented as well as tightly coupled with the underlying database software (configuration and version).
- Lack of granularity: Lack of the right granularity definition for data access as per the business specific requirements i.e. restricting user access to relation level, record level, column level, data element level is another factor which calls for fresh insight into the Distributed Database Security Framework formulation.
- Lack of flexibility: The existing security models lack flexibility in terms of configuring and tailoring the security mechanisms based on the business requirements.
- Non scalability: The sheer absence of non-scalability in the existing security frameworks is a major deterrent that limits their usage in enterprise database based applications.
- Suffers from intrusion attacks: Most of the current solutions are not full proof against the data intrusion attacks which warranted the evolution of a double layered dynamic query translation based security framework.

4. Dynamic Security Framework

4.1. Proposed Framework

The security requirements are taken care of based on categorization of the Subjects, Objects, Sites and the permitted user actions or operations. The proposed methodology introduces a double layered flexible and dynamic security framework as mentioned below –

- It is assumed that each of the users will have a global and a local role. The users will access the distributed DBMS through the Global External Schema (GES).

- Three distinct categories of global roles are assumed – Super User/DBA, DDL and DML as detailed in below section. The Global Schema will maintain a User Master which will store the mapping of the users and their corresponding global roles.
- A query along with the user's global DBMS login credentials (user id, password and global DBMS role) will be received in a Global External Schema. The GES will carry out authentication of user credentials as mentioned in the above section and role based authorization based on the associated global user role.
- Each LES will maintain a local Security Role Translation Table (SRTT) as described in the Site Level Security sub-section, which will store the mapping of the Global and Local Roles. This SRTT information will be replicated at each of the local sites.
- Each local schema will maintain its local user security matrix and table level security matrix described in the Subject Level Security and Object Level Security sub-sections.
- If the query involves only local relations –
  - The query will be passed to the Local Conceptual Schema (LCS). Based on the user role authorization / security clearance and the data/content based security level, the queries will be translated as detailed in the subsequent sections. This step is the dynamic fragmentation of record set based on the user/subject and data/object based security levels.
  - The translated/modified query will be passed to the Local Internal Schema (LIS) for execution and resulting record set will be passed back to the Global schema so that user can only view the data he/she is allowed to view. This will satisfy the confidentiality and availability requirements of the distributed database.
- If the query involves remote/non-local relations –
  - The GES will split the query into sub-queries based on the local DBMS involved. For each of the local DBMS sites involved, the GES will first verify the user authorization to carry out the database transactions/operations based on the associated global role and accordingly distribute the sub-queries to the corresponding LESs" along with the user's global credentials (user id, password and global role) by looking up the Site Metadata.
  - When the sub-query is received at a LES, the corresponding local role will be evaluated from the SRTT.
  - The LES will pass the sub-query to its Local Conceptual Schema (LCS) for the dynamic fragmentation step. Here the sub-query will be translated based on the user's local role authorization / security clearance and the data/content based security level.
  - The translated/modified sub-query will be passed to the Local Internal Schema (LIS) for execution and resulting record set passed back to the LES via the LCS. Each of the LES involved in the query execution will pass back their resultant record set to the GES for consolidation, which will be available to the user for viewing. This will satisfy confidentiality and availability requirements of the distributed DBMS.

The below section illustrates the implementation details of the dynamic fragmentation step, i.e. how a query will be translated based on the site security level, user's role authorization / security clearance and the data/content specific parameterized security matrix.

4.2. Security Layers

The proposed methodology comprises of two independent security layers – Global and Local. Both these security layers can be configured and managed independently of each other which makes the security framework flexible and loosely coupled and allows tailoring of the local security policies specific to each of the DBMS involved. The Global Security Layer is taken care by the Operational Level Security and the Local Security Layer is implemented using the Site Level, Subject Level and Object Level Security mechanisms. These security mechanisms provide flexibility since these different security levels can be configured independently of each other and combination of some or all of these mechanisms can be adopted based on business specific requirements.

4.2.1. Operation Level Security

The Operational Level Security mechanism determines the different types of permissible operations and actions on the distributed databases based on the Global User Roles definition. A typical distributed database comprises of the different levels or hierarchy as depicted in the Figure 3.

Each of the levels has different sensitivities and security requirements which can be managed by defining corresponding global roles. The levels have been classified in three broad categories as evident from the different colour codes and assign security labels to allow role based access in the following manner –

- The lower layer with green colour coding signifies the Record and Data Items. Users having access to this layer will be granted permissions for executing only DML statements over the database entities. Such a global role can be termed as DML user (G1).
- The middle layer with yellow colour coding signifies the Database entities like tables/views etc. Users having access to this layer will be granted permissions for executing DDL statements along with DML statements over the database entities. Such a global role can be termed as DDL user (G2).
- The topmost layer with orange colour coding signifies the root level of the distributed database. Users having access to this layer will be granted the highest level of authority over the entire distributed DBMS. Such a global role can be termed as the DBA or the Super user (G3).

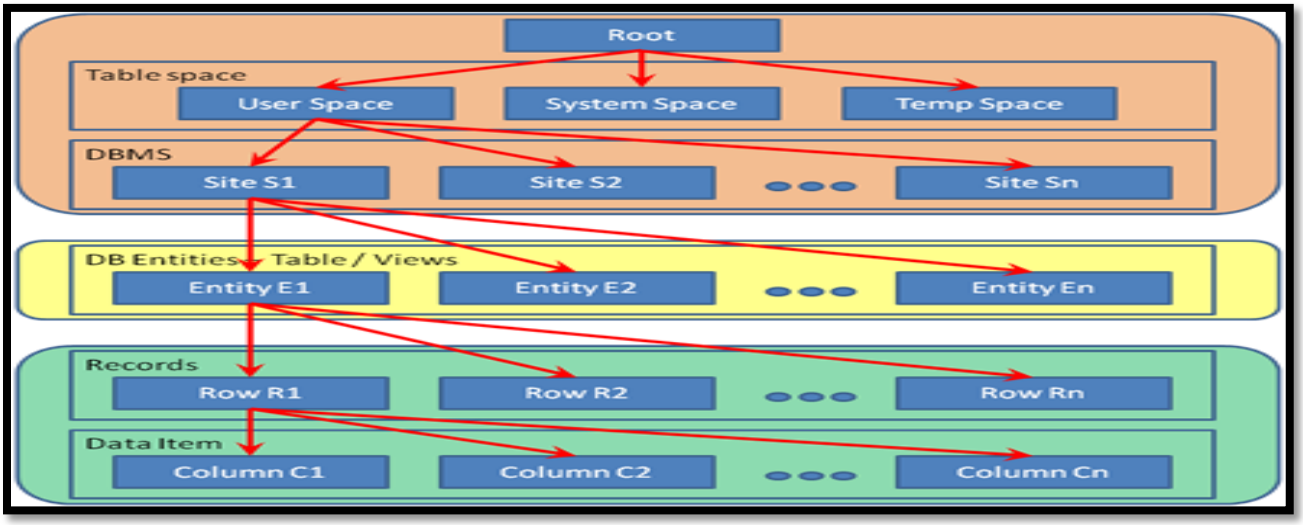


Figure 3: Security Hierarchy in Distributed DBMS

These three global roles are site agnostic and will determine the Operational Level security i.e. what are the permissible set of operations allowed by each of these global roles. Additionally the Global schema will maintain a User-Role mapping matrix and a Site Metadata which contains the list of underlying local DBMSs and site information.

4.2.2. Site Level Security

The Site Level Security caters to the mapping of the global user roles and local site specific user roles for the database users across the different database sites to implement and tailor the site based security requirements.

Table 1: Security Role Translation Table

User	Global User Role	Local User Role
U001	G1	L1
U002	G1	L1
U003	G1	L2
U004	G1	L2
U005	G1	L3
U006	G2	L3
U007	G3	L3

The Local Security mechanism will be implemented using the Security Role Translation Table (SRTT) maintained at the local schema. The SRTT table shown above stores the mapping of the Local User Roles (L1, L2, L3 etc.) and the Global User Roles (G1, G2, G3 etc.) and can be used to tailor the local site specific security requirements. The local user roles will determine the access privilege and security clearance of the users associated with the role and can be defined differently in each local schema, i.e. L2 role may signify Low Security level for a site S1 whereas L2 may signify Medium Security level in site S2. The SRTT for a Local Schema can be configured independently of the Global Schema. Records with no global role defined will determine a pure local role and users belonging to those roles will not be able to query for remote/non-local relations. Such records are marked with a global role as Global Access Denied (GAD).

4.2.3. Subject Level Security

The Subject Level Security mechanism defines the user level security definition which is based on the Role Based Access Control (RBAC) principles. The User Security table shown below stores the User Security definition used for the proposed implementation.

Table 2: USER_SECURITY Relation

Userrole	Seclevel
L1 (Normal)	1
L2 (Sophisticated)	2
L3 (Super)	3

SECLEVEL High (H): 3, Medium (M): 2, Low (L): 1
Translated Security Privilege:

- High Security Level implies privilege of Low, Medium and High levels (L, M, H)
- Medium Security Level implies privilege of Low and Medium levels (L, M)
- Low Security Level implies privilege of Low level only (L)

4.2.4. Object Level Security

The Object Level Security mechanism addresses the categorization and labelling of the data elements to control access to the data elements as per the granularity requirement i.e. relation/table level, record/row level, column level or individual cell level (through the combination of row and column level security mechanisms).

The below mentioned EMPLOYEE_DETAILS relation belonging to a local DBMS has been used to demonstrate the Object Level Security. The Row Level and the Column Level Security values are defined as part of the Object Level Security definition.

Table 3: EMPLOYEE_DETAILS Relation

EMP ID	EMP NAME	CITY	SALARY	CREATED BY
E001	X1	Kolkata	35000	U001
E002	X2	Delhi	40000	U005
E003	X3	Mumbai	50000	U004
E004	X4	Chennai	35000	U004
E005	X5	Bangalore	45000	U003
E006	X6	Indore	20000	U002
E007	X7	Pune	30000	U001
E008	X8	Hyderabad	40000	U005
E009	X9	Jaipur	25000	U003

Row Level Security - The Row Level Security values are based on the access level/privileges/roles of the user creating the record which will determine the record level access, i.e.

- User with Security Level H can access rows created by users with privilege of Low, Medium and High levels (L, M, H).
- User with Security Level M can access rows created by users with privilege of Low and Medium levels (L, M).
- User with Security Level L can access rows created by users with privilege of Low Level (L)

Column Level Security - This determines the column level access. In this implementation, it is assumed that there exist the following data constraints –
Simple Constraints (Design time constraints):

- C3 has Medium Security Level
- C4 has High Security Level
- Rest of the columns have default Security Level, i.e. Not Applicable

Table 4: TAB_COL_SECURITY Relation

Tabcolid	Tablename	Column Name	Seclevel
T001	EMPLOYEE_DETAILS	City	2
T002	EMPLOYEE_DETAILS	Salary	3

Content Based Constraints:

- Based on access privileges of the user creating the record, the default security level will be set to Low, Medium or High levels (L/ M/ H)
- Content Based constraint will be applicable on top of Simple Constraints

This final data/object security is evaluated based on the translated logic from the above two tables (Table 3 and Table 4 shown above). The translated security matrix is defined in the following table (Table 5).

Table 5: EMPLOYEE_SECURITY Relation

	SECLEVEL OF COLUMNS				
EMP ID	**EMP ID**	**EMP NAME**	**CITY**	**SALARY**	**CREATED BY**
E001	1	1	2	3	1
E002	3	3	2	3	3
E003	2	2	2	3	2
E004	2	2	2	3	2
E005	2	2	2	3	2
E006	1	1	2	3	1
E007	1	1	2	3	1
E008	3	3	2	3	3
E009	2	2	2	3	2

4.3. Site Administration

Site Security Administration has to be done as mentioned below –

- The local site level administrator will configure the local security definitions for that site which includes identification of the set of local roles, Subject and Object level security matrix and then create a SRTT matrix to specify the mappings of the local roles with the global roles.
- When a local site is removed from the DBMS, it is just dropped from the user space and is no longer part of the distributed database table spaces. Additionally, the site entry is removed from the Site Metadata maintained in Global schema.
- Changes in role based security level for a particular site can be done by modifying the local security definitions for that site by the local administrator.

4.4. Role Administration

Role Administration typically involves the following- Global Roles Management: The Global roles are assumed to be fixed and limited to DBA/Super user (G1), DDL user (G2) and DML user (G3) as per the proposed model. Local Roles Management: The local roles viz. L1, L2 and L3 used in this model are maintained in the Local Schema by a Local DBMS administrator.

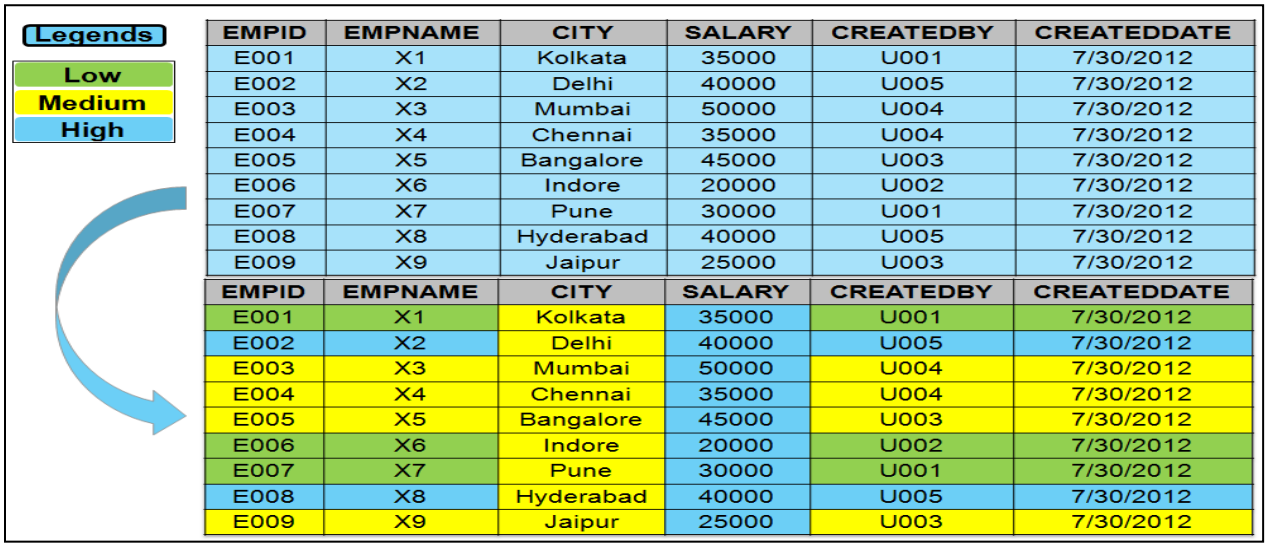
These roles indicate the security clearance/privileges for a category of users/applications for accessing only the local DBMS entities. Local Role Administration can be done as mentioned below -

- In case a new local role is introduced, the Local Administrator for the corresponding site has to define a new entry in the local SRTT and specify the corresponding Global Role mapping. This matrix needs to be replicated across all the sites. Additionally, the Subject and Object level security matrix have to be configured for the new local role in the Local schema. No changes are needed to be done in any of the other local DBMS or the Global Schema.
- When a local role is removed, the entry for that local role should be deleted from the SRTT from all the sites.

5. Implementation Details

5.1. High Level Strategy

Based on the parameterized security definitions at Operational, Site, Subject and Object level, the Dynamic Fragmentation and runtime Query Translation would be done. The original table/ relation used for the implementation and the corresponding data security mapping at row and column levels have been depicted through different colour legends in the below diagram (Figure 4).



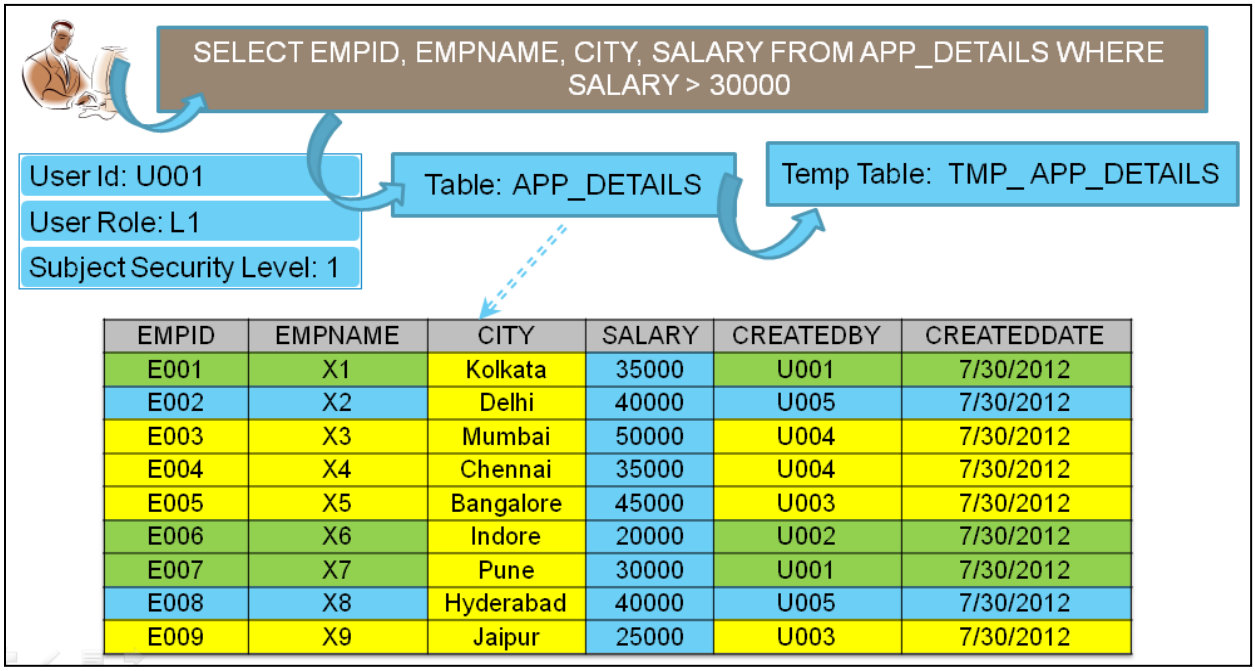
EMPID	EMPNAME	CITY	SALARY	CREATEDBY	CREATEDDATE
E001	X1	Kolkata	35000	U001	7/30/2012
E002	X2	Delhi	40000	U005	7/30/2012
E003	X3	Mumbai	50000	U004	7/30/2012
E004	X4	Chennai	35000	U004	7/30/2012
E005	X5	Bangalore	45000	U003	7/30/2012
E006	X6	Indore	20000	U002	7/30/2012
E007	X7	Pune	30000	U001	7/30/2012
E008	X8	Hyderabad	40000	U005	7/30/2012
E009	X9	Jaipur	25000	U003	7/30/2012

EMPID	EMPNAME	CITY	SALARY	CREATEDBY	CREATEDDATE
E001	X1	Kolkata	35000	U001	7/30/2012
E002	X2	Delhi	40000	U005	7/30/2012
E003	X3	Mumbai	50000	U004	7/30/2012
E004	X4	Chennai	35000	U004	7/30/2012
E005	X5	Bangalore	45000	U003	7/30/2012
E006	X6	Indore	20000	U002	7/30/2012
E007	X7	Pune	30000	U001	7/30/2012
E008	X8	Hyderabad	40000	U005	7/30/2012
E009	X9	Jaipur	25000	U003	7/30/2012

Figure 4: Dynamic Fragmentation of the Relation

The high level steps for dynamic query translation based on the Subject and Object level security classifications have been elaborated below –

- Step 1: Identify the user's local role and associated Subject Security Level.
- Step 2: Identify all the involved relations that are part of the user query i.e. list of tables in the FROM clause of the query as shown in Figure 5. Figure 6 is the snapshot of the involved table and procedure used for the actual implementation.



EMPID	EMPNAME	CITY	SALARY	CREATEDBY	CREATEDDATE
E001	X1	Kolkata	35000	U001	7/30/2012
E002	X2	Delhi	40000	U005	7/30/2012
E003	X3	Mumbai	50000	U004	7/30/2012
E004	X4	Chennai	35000	U004	7/30/2012
E005	X5	Bangalore	45000	U003	7/30/2012
E006	X6	Indore	20000	U002	7/30/2012
E007	X7	Pune	30000	U001	7/30/2012
E008	X8	Hyderabad	40000	U005	7/30/2012
E009	X9	Jaipur	25000	U003	7/30/2012

Figure 5: Identification of user details and involved relations from the input query

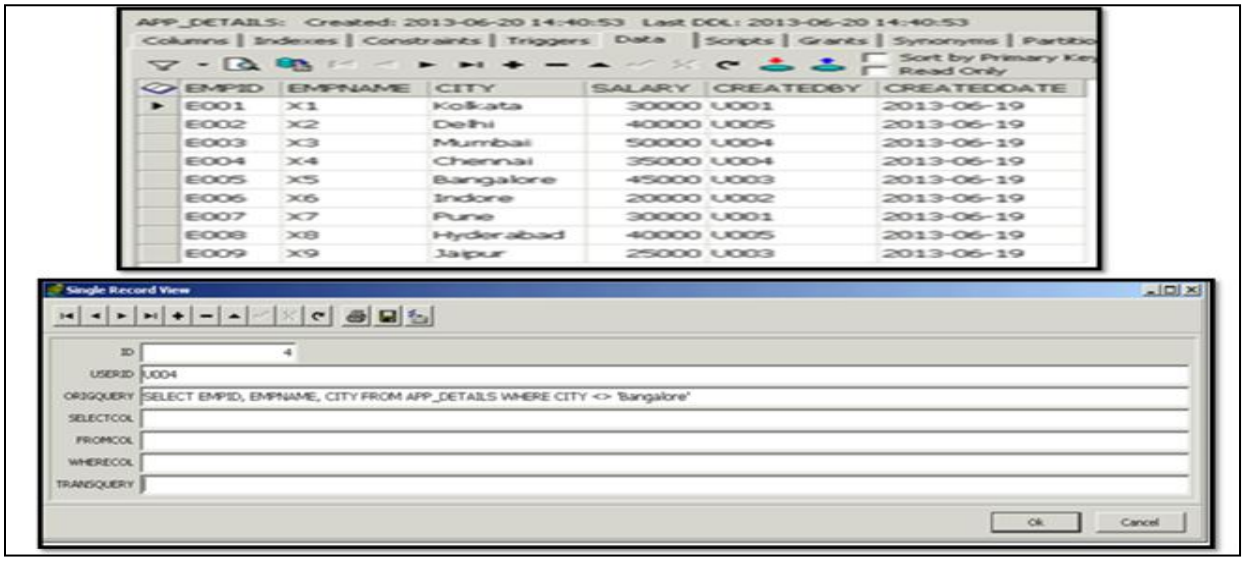
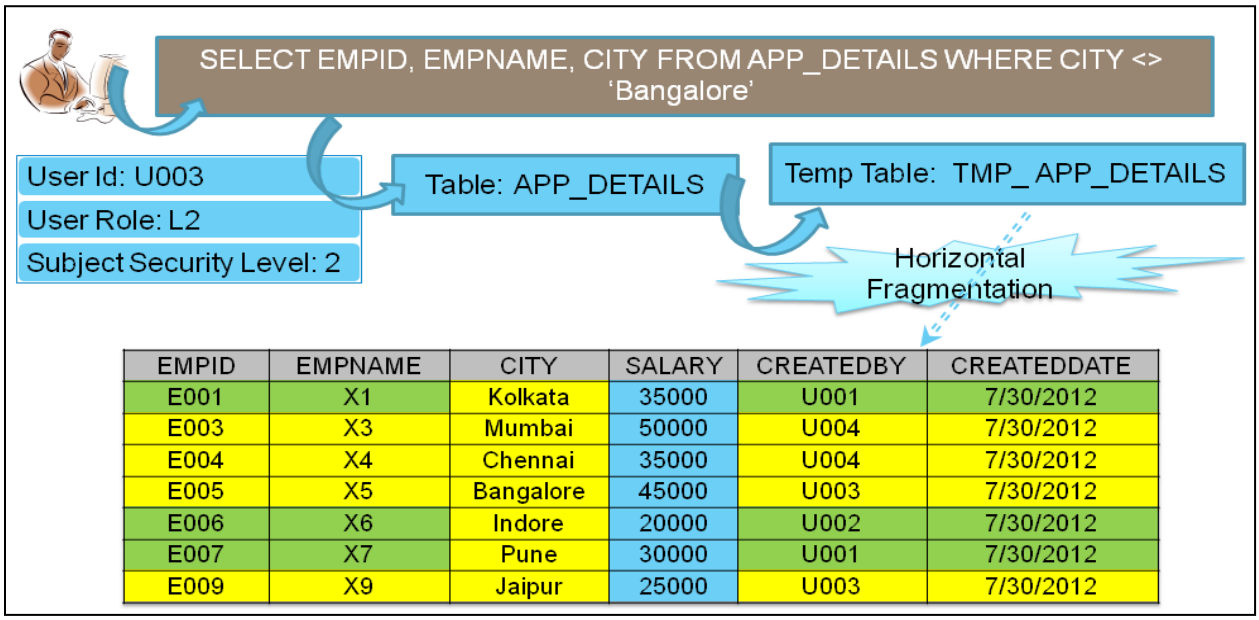


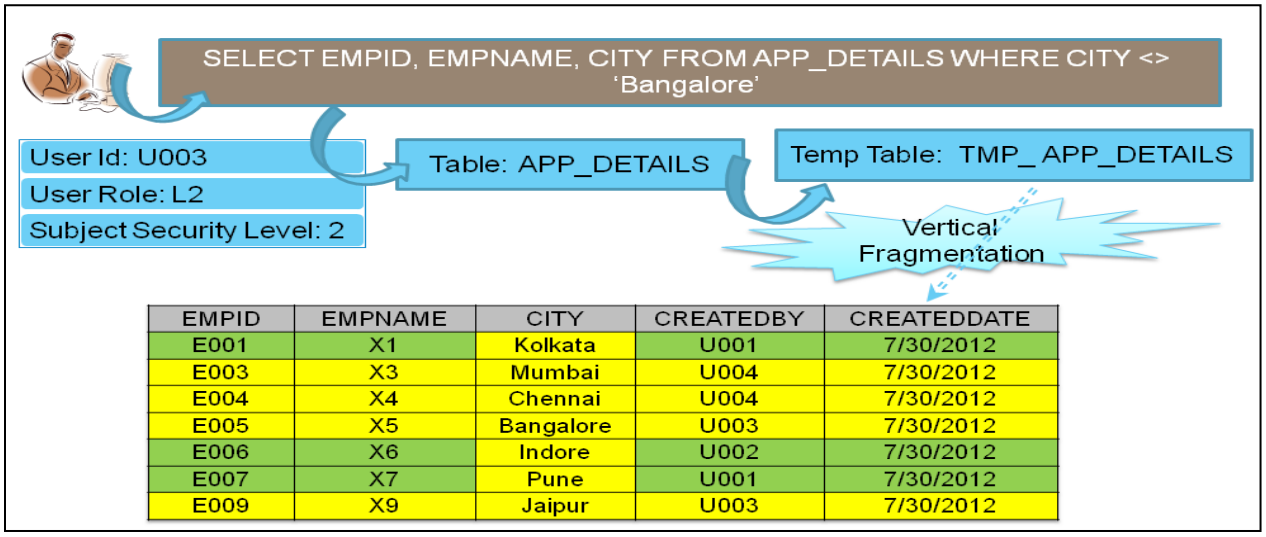
Figure 6: The Database Table and Procedure used for implementation

- Step3: For each of the relations identified in Step2, create corresponding temporary tables taking into consideration the Row level and Column level security values. This can be achieved using derived dynamic fragmentation of the table into temporary table i.e. the vertical fragmentation of the relation will ensure that only columns corresponding to user's security level are included in the temporary table and the horizontal fragmentation of the relation will restrict the record-set in the temporary table as per user's security clearance. The below figures (Figure 7 and Figure 8) demonstrate the derived dynamic fragmentation strategy.



EMPID	EMPNAME	CITY	SALARY	CREATEDBY	CREATEDDATE
E001	X1	Kolkata	35000	U001	7/30/2012
E003	X3	Mumbai	50000	U004	7/30/2012
E004	X4	Chennai	35000	U004	7/30/2012
E005	X5	Bangalore	45000	U003	7/30/2012
E006	X6	Indore	20000	U002	7/30/2012
E007	X7	Pune	30000	U001	7/30/2012
E009	X9	Jaipur	25000	U003	7/30/2012

Figure 7: The Horizontal Fragmentation step



EMPID	EMPNAME	CITY	CREATEDBY	CREATEDDATE
E001	X1	Kolkata	U001	7/30/2012
E003	X3	Mumbai	U004	7/30/2012
E004	X4	Chennai	U004	7/30/2012
E005	X5	Bangalore	U003	7/30/2012
E006	X6	Indore	U002	7/30/2012
E007	X7	Pune	U001	7/30/2012
E009	X9	Jaipur	U003	7/30/2012

Figure 8: The Vertical Fragmentation step

- Step4: Translate the user query by replacing the involved tables in the user input query with the corresponding temporary tables. The translated query will run against the temporary table generated using derived dynamic fragmentation as shown in the below figure (Figure 9).

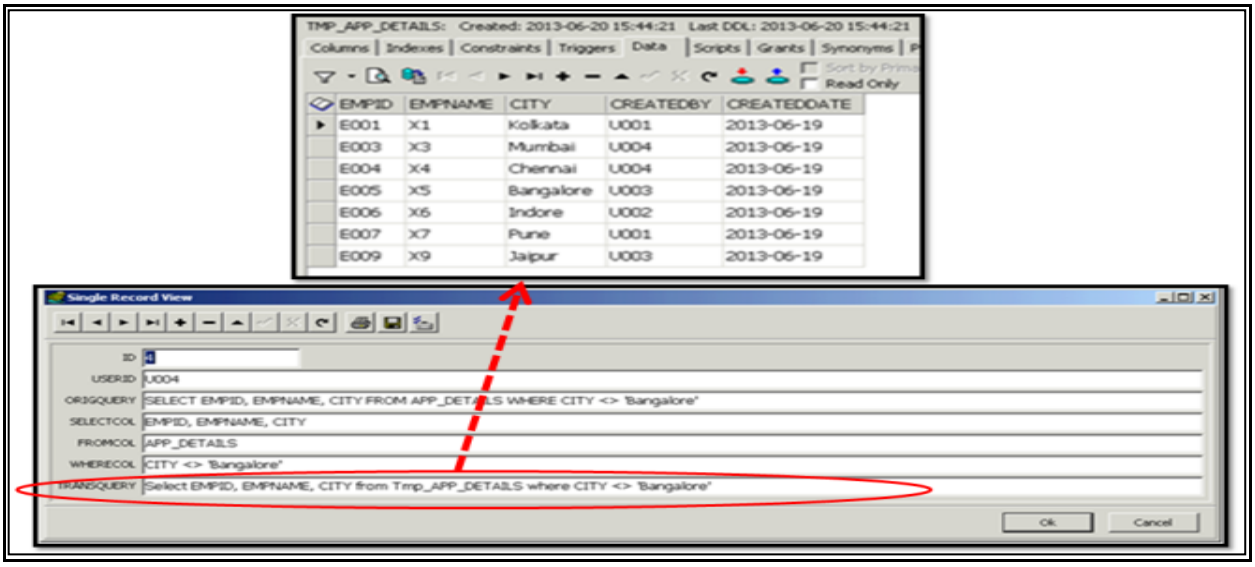


Figure 9: Dynamic Fragmentation of the relation and translated query output from procedure

- Step5: Validate the translated query to check whether the SELECT and WHERE clauses involves any reference to non-existent columns.
- Step6: If query validation is successful, execute the translated query; else display message to user that the query is invalid. The below figures (Figure 10 and Figure 11) show the execution result for the translated query.

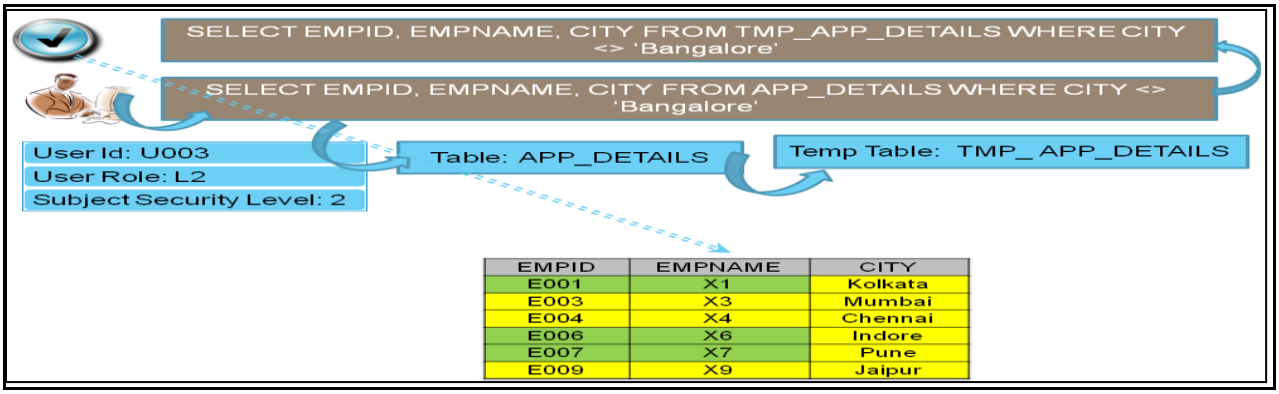


Figure 10: Query Translation and Validation step

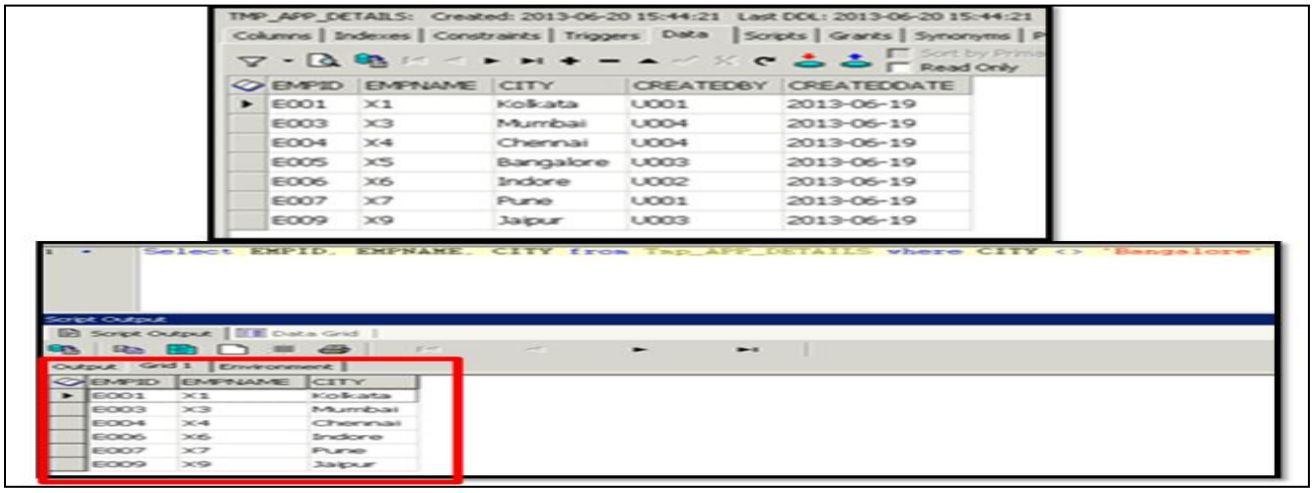


Figure 11: Runtime Query Translation and execution results

5.2. Security Violation Scenarios

Listed below are examples of some of the commonly observed security violations which can be managed through the proposed security framework -

- Querying for columns with higher security clearance i.e. SELECT clause comprises of columns which the user is not authorized to view as per the Subject Level security clearance.
- Querying for records created by a different user role with higher security clearance.
- Using aggregate queries to infer about secure information.
- Usage of queries that return data at/below user's security level but its evaluation requires accessing columns and rows having higher security clearance.

6. Conclusion and Future Work

This paper starts with the concept of the distributed database security and the issues involved in secure distributed query processing. Thereafter a survey of the existing security mechanisms is presented followed by a detailed description of the proposed dynamic security framework and its various security levels. Finally, the security framework was implemented using a sample RDBMS and the results were discussed subsequently. The objective of this proposed methodology was to address the issue of secure distributed query processing using a dynamic and robust framework which could be tailored based on the business requirement to take care of relational level, record level, column level as well as atomic

data element level security and access requirements. Following are the highlights of the proposed solution:

- Dynamism: The proposed solution offers a dynamic security mechanism since it is based on the principle of user query translation.
- Robustness: It offers a robust and highly secured solution mechanism which is effective in preventing majority of the database intrusion attacks.
- Granularity of security levels – The proposed methodology comprises of two independent security layers – Global and Local. The Global Security Layer is taken care by the Operational Level Security and the Local Security Layer is implemented using the Site Level, Subject Level and Object Level Security mechanisms. These security mechanisms provide flexibility since these different security levels can be configured independently of each other and combination of some or all of these mechanisms can be adopted based on the business specific requirements.
- Flexibility – The global and local security layers can be configured and managed independently of each other which makes the security framework flexible and loosely coupled and allow tailoring of the local security policies specific to each of the local DBMS involved.
- Scalability – The solution framework can be scaled for RDBMS based enterprise applications.

In this paper, only the database query operations have been explored to demonstrate the security mechanism. This mechanism can be used for other DML operations as well. The dynamic security framework for distributed databases can be extended in cloud based architectures, grid computing, social media etc. Currently, the RDBMS is the better choice for a distributed application. This is due to the relative maturity of the relational model and the existence of universally accepted standards.

This solution has been devised for distributed relational databases but the solution offers potential for use in object oriented database environments as well. In addition to the security aspect discussed in this paper, there are also opportunities for research in several other areas viz. subject authorization strategies for heterogeneous distributed systems, inference prevention strategies for both centralized and distributed database systems, and distributed object-oriented database security standards.

Acknowledgment

I am extremely grateful to Dr. Parama Bhaumik, Assistant Professor, Department of Information Technology, Jadavpur University for providing me the needed guidance to complete this work and for helping me with her invaluable suggestions and necessary information in this field.

References

[1] Batra N., Aggarwal H., "Autonomous Multilevel Policy Based Security Configuration in Distributed Database", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012, pp. 170-176.

[2] Bhaskar R., Sharma R., "An Analysis of Vertical Splitting Algorithm", International Journal of Computer Applications, August 2012, pp. 30-36.

[3] Coy S.P., "Security Implications of the Choice of Distributed Database Management System Model: Relational vs. Object-Oriented", NISSC 96, 2008, pp. 428-437.

[4] Crampton J., "Specifying and Enforcing Constraints in Role-Based Access Control", Proc. Eighth ACM Symp. Access Control Models and Technologies (SACMAT '03), June 2003, pp. 43-50.

[5] Gupta V. K., Sheetlani J., Gupta D., Datta S. B., "Concurrency Control and Security issues of Distributed Databases Transaction", Research Journal of Engineering Services, August 2012, pp. 1-4.

[6] Hylkema M., "A Survey of Database Inference Attack Prevention Methods", Technical report, Boston University, December 2009, pp. 1-3.

[7] Khair M., Mavridis I., Pangalos G., "Design of Secure Distributed Medical database Systems", Proceedings of 9th International Conference, DEXA‟98 (Database and Expert Systems Applications), Vienna, Austria, August 1998, pp. 402-500.

[8] Kose I., "Distributed Database Security", Data and Network Security, Spring 2002, GYTE, pp. 1-5.

[9] Lunt T. F., Fernandez E. B., "Database security", ACM SIGMOD Record, v.19 n.4, December 1990, pp. 90-97.

[10] Miklau G., "Confidentiality and Integrity in Distributed Data Exchange", PhD thesis, University of Washington, 2005, pp. 32-75.

[11] Narmada V., Swamy B.N., Kumar D.L.S., "An enhanced security algorithm for distributed databases in privacy preserving data bases". International Journal of Advanced Engineering Sciences and Technologies (IJAEST), 2011, pp. 219–225.
[12] Sandhu R.S., Jajodia S., "Data and database security and controls", Handbook of information security management, Auerbach Publishers, 1993, pp. 481-499.
[13] Thuraisingham B., Kamon A., "Secure query processing in distributed database management systems: Design and performance studies". In Proceedings, 6th Annual Computer Security Applications Conference, Tucson, Arizona, December 1990, pp. 88-102.
[14] Thuraisingham B., "Privacy constraint processing in a privacy-enhanced database management system", Data & Knowledge Engineering, 2005, pp. 159-188.

Arunabha Sengupta received his B.Tech. degree in Computer Science and Technology from Kalyani Govt. Engineering College, Kalyani University, WB, India, in 2004. He was awarded the M.E. degree in Software Engineering (Information Technology) from Jadavpur University, Kolkata, WB, India, in 2013. He is presently working as Associate Consultant at Tata Consultancy Services Limited in the role of Solution Architect. He has vast experience in all the SDLC phases involving multiple technology stacks with exposure in Banking and Financial domain, Insurance, Loan Origination, Credit Card businesses across global geographies. His interest areas include Database Modeling and Architecture Design, DWH/BI and Analytics, Middleware Integration, J2EE and SOA technologies.
Email: arunabha23@gmail.com