CHAPTER 8

CONCLUSIONS AND FUTURE SCOPE

8.1 CONCLUSIONS

During the sessions of the thesis, we studied the basics of MLS databases which included the motivations behind the development of MLS database, their architectures, models etc. In order to store an MLS relation at physical layer it needs to be decomposed into a number of single level relations. When a query with multiple attributes is fired these decomposed relations are needed to be joined back to reconstruct the original relation. For this, different decomposition and recovery algorithms are given in the literature.

These algorithms of decomposition and recovery have been analyzed. On the basis of the number of tuples in a relation, we formulated the number of operations required for all these three algorithms i.e. SeaView Decomposition & Recovery, Decomposition & Recovery using Polyinstantiation and Novel Decomposition and Recovery. According to the formula, it can be concluded that novel model is the most efficient since it requires minimum number of operations and is equal to the number of tuples in the relation.

Further, to make the query processing more secure, a concept of secure dictionary with the facility of coding and decoding is introduced and analyzed to see the overheads. It has been observed that by adding a few more operations, security can become sounded. In this part of research, various security constraints that can be applied during processing of a query are studied. To prevent unauthorized access of important security-related information, the concept of security catalogue that is in the coded form and replicated at each site, was introduced. It consists of the clearance level and the classification level of user and transactions. The proposed architecture was analyzed on the basis of extra operations required and it was observed that with very little overheads, we can provide extra security. Further, a secure database environment is created by using Oracle11g Release 2. Different set of queries for selection, deletion, updation, alteration and for dropping the schema are constructed.

Firstly, the set of all the queries were executed in an insecure database environment and then the same queries were executed on the multilevel secure database with two levels for security and then three levels for security. It was observed that as the number of security levels are increased, the execution time also increased. The significant difference is
observed in the execution time of the queries for dropping the schema. The experiments were run using Oracle11g Release2 installed on a machine with i3, 1.8GHz processor.

A multilevel secure database protects the classified information from unauthorized users based on the classification of the data and clearance of the users. Therefore the process of deadlock detection and resolution becomes more complicated. The algorithm for deadlock detection and resolution in distributed databases, given by Alom et.al. was studied and analyzed. It was observed that the local deadlocks are dependent on global deadlocks. On the basis of this observation, a new algorithm is developed for the detection and recovery of deadlocks. By taking illustrative examples, it is observed that the number of transactions aborted in the proposed algorithm is less than the number of transactions aborted in the algorithm given by Alom et.al. Further, the proposed algorithm is extended for multilevel secure databases in a distributed environment. The algorithm was based on the property that in MLS databases, if two two transactions executing at different clearance levels, resulting in a deadlock, then to break the deadlock the transaction with by high clearance level will be aborted. This algorithm was analyzed and it was concluded that the number of transactions aborted are directly proportional to the number of transactions and number of sites rather than the number of security levels.

Further, the impact of security constraints of MLS database on the conventional concurrency control techniques have been investigated. The two major problems were studied based on which the traditional concurrency control techniques cannot be applied to MLS Databases. These two problems were Covert Channels and Inference.

The algorithms given by Kim et. al. [70] and Kaur et. al. [63] were studied and analyzed with the help of illustrative examples. It was observed that these algorithms were suffering from Starvation, Retrieval Anomaly and Covert Channel problems. To eliminate these problems a new algorithm based on Multi-version concurrency control technique has been proposed. In this approach, the concept of version lock, LDeg function and virtual levels is introduced to make the technique starvation and covert channel free. The algorithm was analyzed and it was concluded that the proposed algorithm is free from starvation and covert channels and is suitable in the environments where users with different clearance levels issue the queries.
8.2 FUTURE SCOPE

In this research, the techniques for decomposition and recovery of multilevel secure databases have been analyzed. We have formulated formula for the number of operations required by different techniques based on the number of tuples in the relation. It can be concluded that the number of operations required for decomposition and recovery are minimum in case of novel decomposition. In future, the number of security levels can be increased to show their effect on the existing decomposition techniques. Further, new techniques can be developed to reduce the number of operations.

The proposed algorithm for concurrency control in MLS databases can be implemented in a database to analyze its performance in real time. Based on the observations, it can further be modified. Also a tradeoff is present in the algorithm i.e. if covert channels are prevented to be established, then there is a chance of starvation and if starvation is to be prevented completely, then channels may be established. This trade off can also be further studied in order to improve the algorithm.

Although the protocol offers the good solution in multilevel secure databases, but it suffers from Performance degradation due to following:

1. Overheads due to Blocking time, version selection time, level modifications.
2. This degradation is due to transaction blocking, restarts, or both.