Advanced Database Management System Laboratory

Student Name: Junaid Altaf Girkar

SAP ID: 60004190057

Batch: TE Comps A4

Experiment No: 01

Aim: Case Study on professional and commercial database.

Labwork:

(paste the screenshots of your selected database and write 4 to 5 line description about it)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 02

Aim: Simulate query optimization by applying an SQL query on your selected database.

Labwork:

(paste the screenshots of query (10 queries which we performed initially) on your selected database)

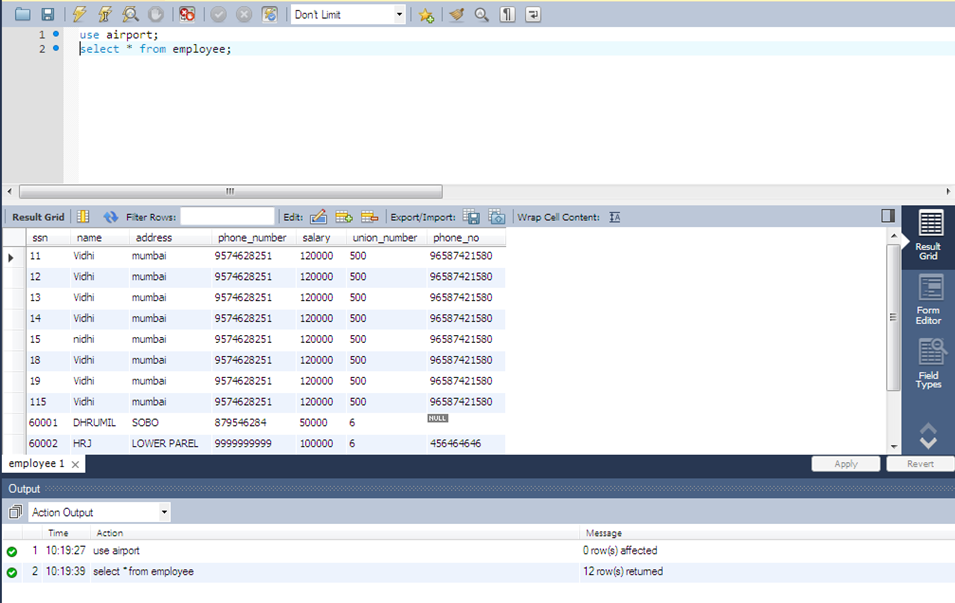
Theory: Distributed query optimization requires evaluation of a large number of query trees each of which produce the required results of a query. This is primarily due to the presence of large amount of replicated and fragmented data. Hence, the target is to find an optimal solution instead of the best solution.

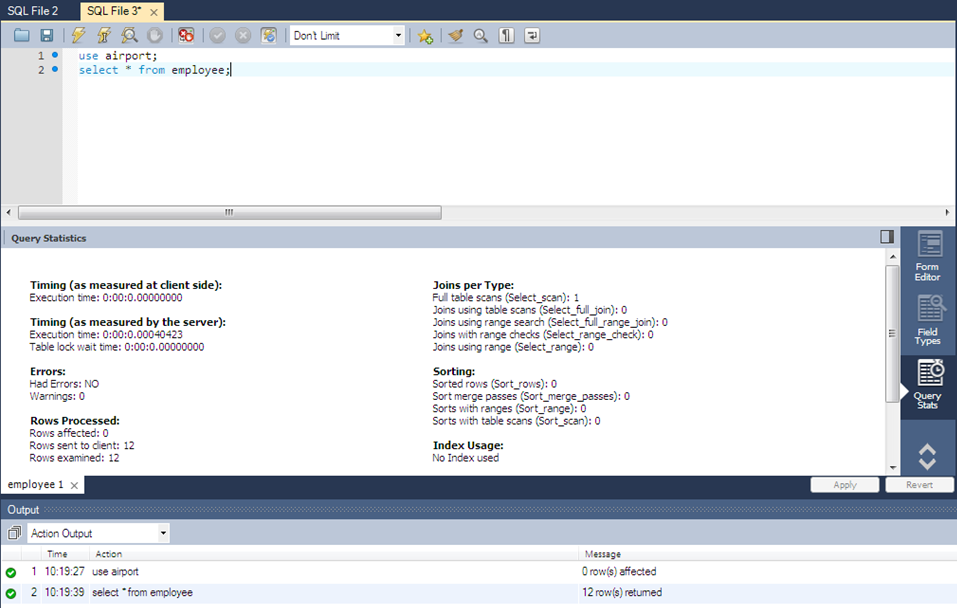
The main issues for distributed query optimization are −

* Optimal utilization of resources in the distributed system.
* Query trading.
* Reduction of solution space of the query.

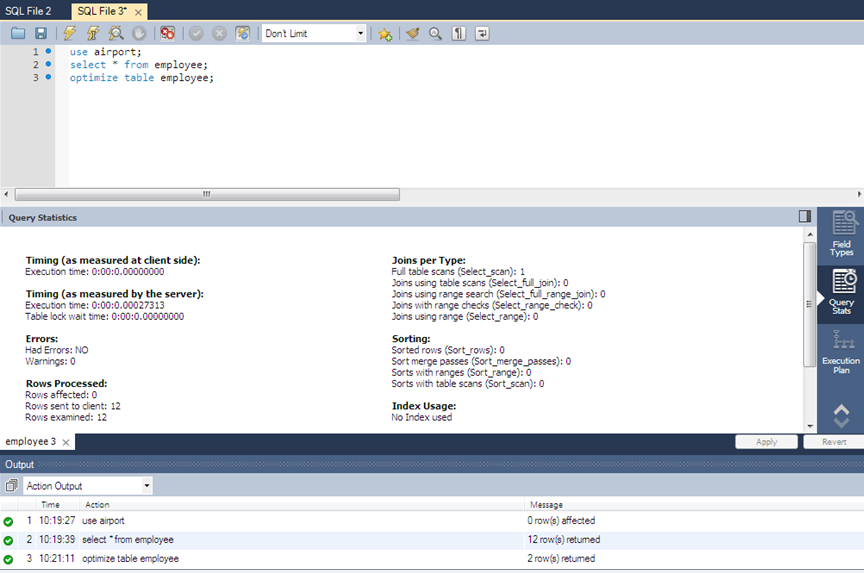
**TABLE LEVEL OPTIMIZATION**

**Normal Query : select \* from employee;**



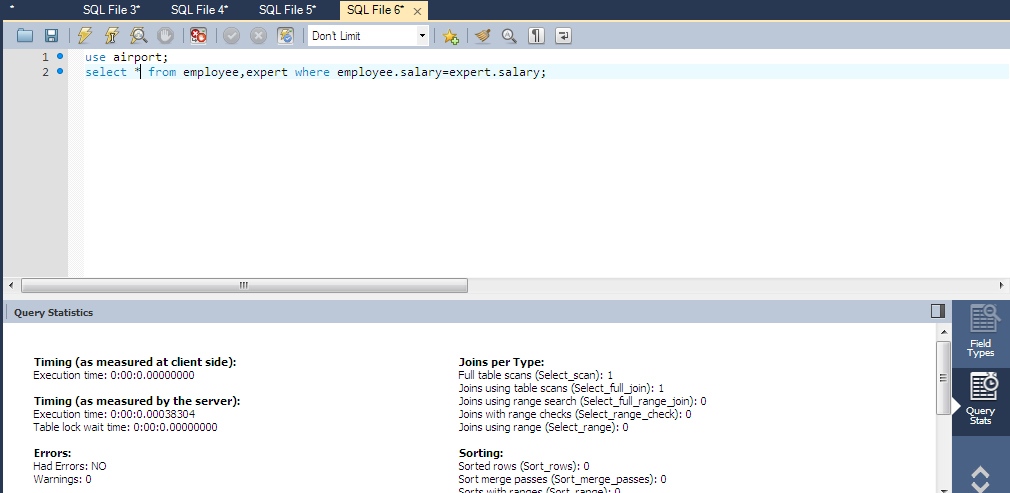
**Without Optimization**

**After Optimization**

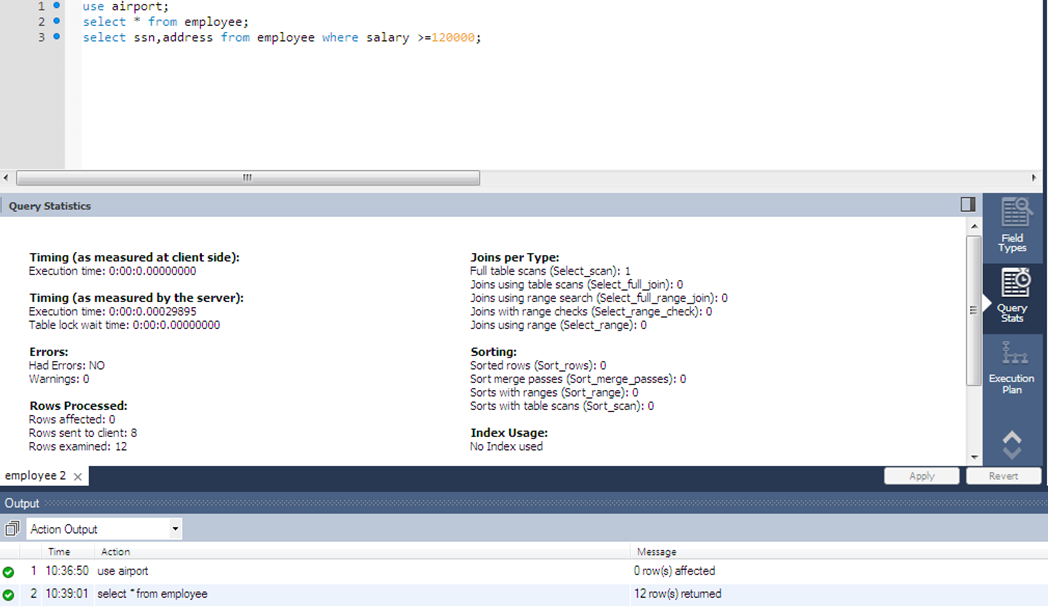


**INDEX LEVEL OPTIMIZATION**

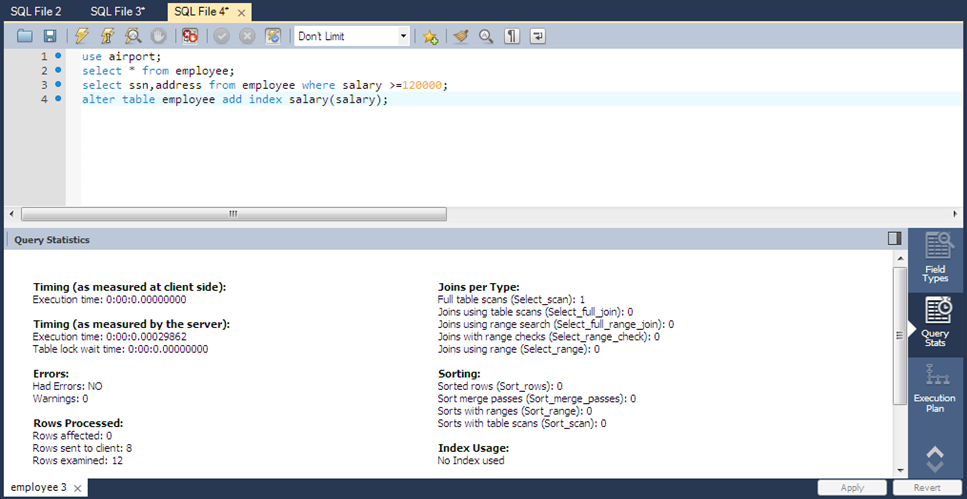
**Before adding index join query**

****

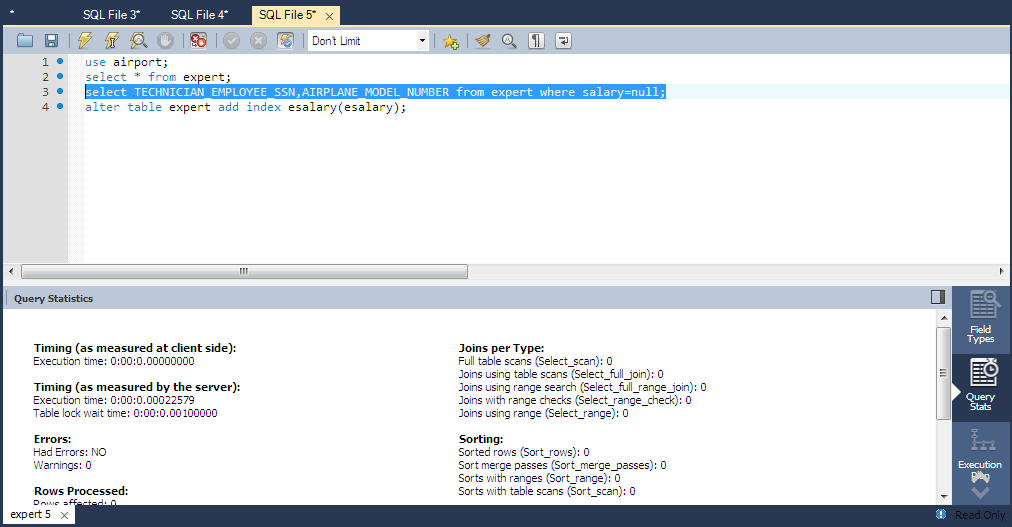
**Normal query on table employee**

****

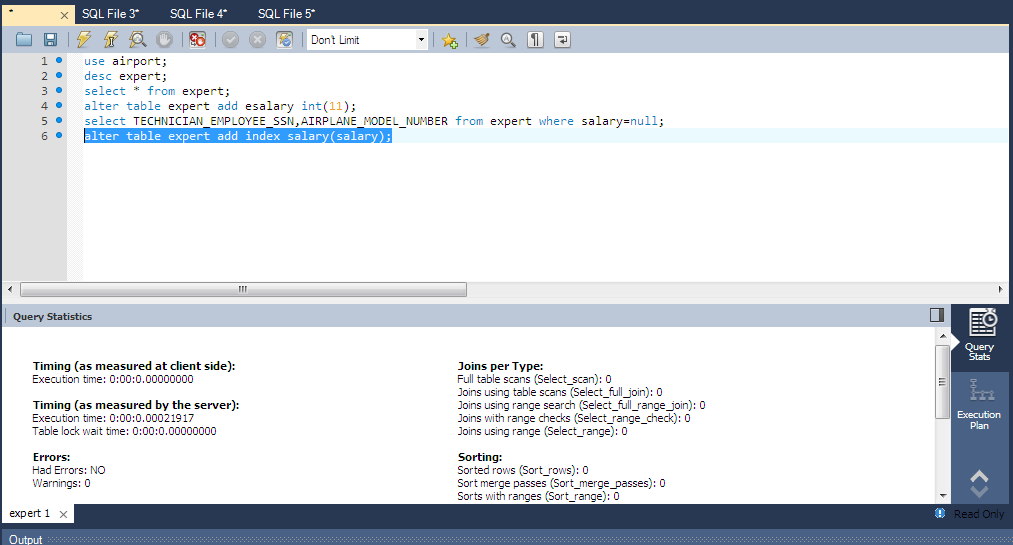
**After adding index**



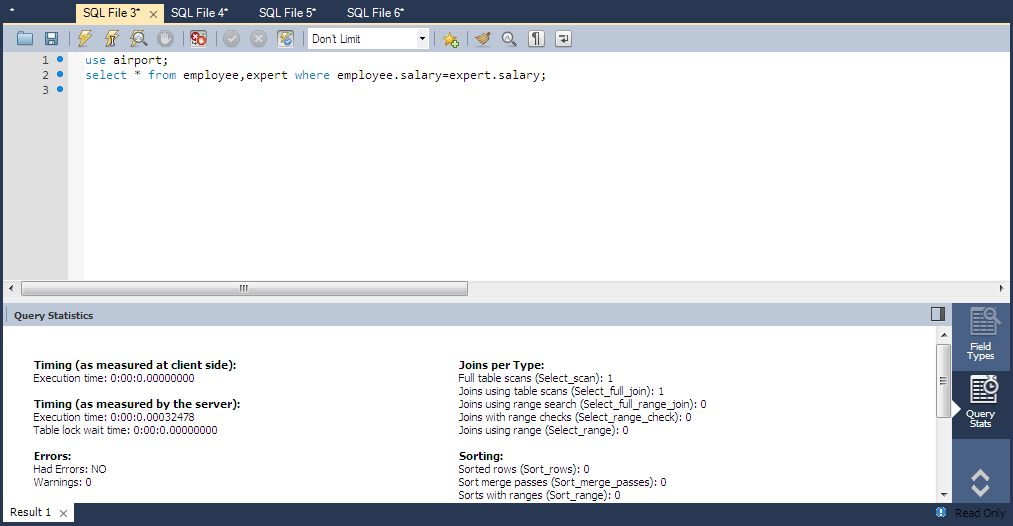
**Normal query on table expert**



**After adding index**



**After performing join**



(Paste the screenshots of query which you have written for preparing the Index and write 4 to 5 lines about Index)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

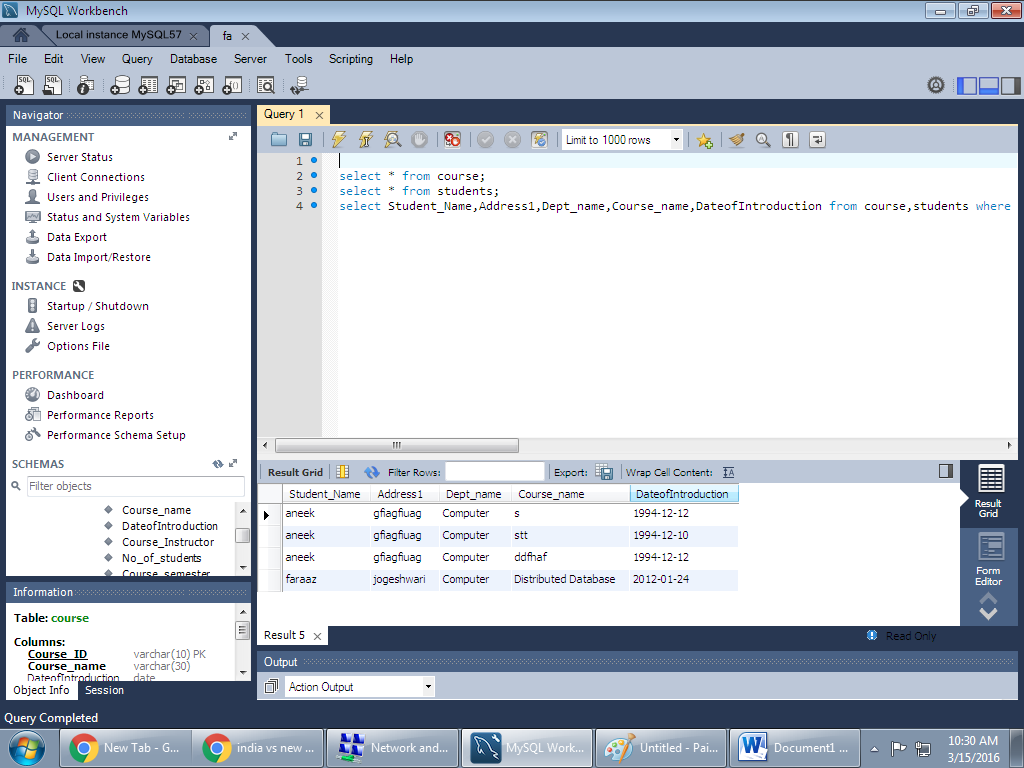
Experiment No: 03

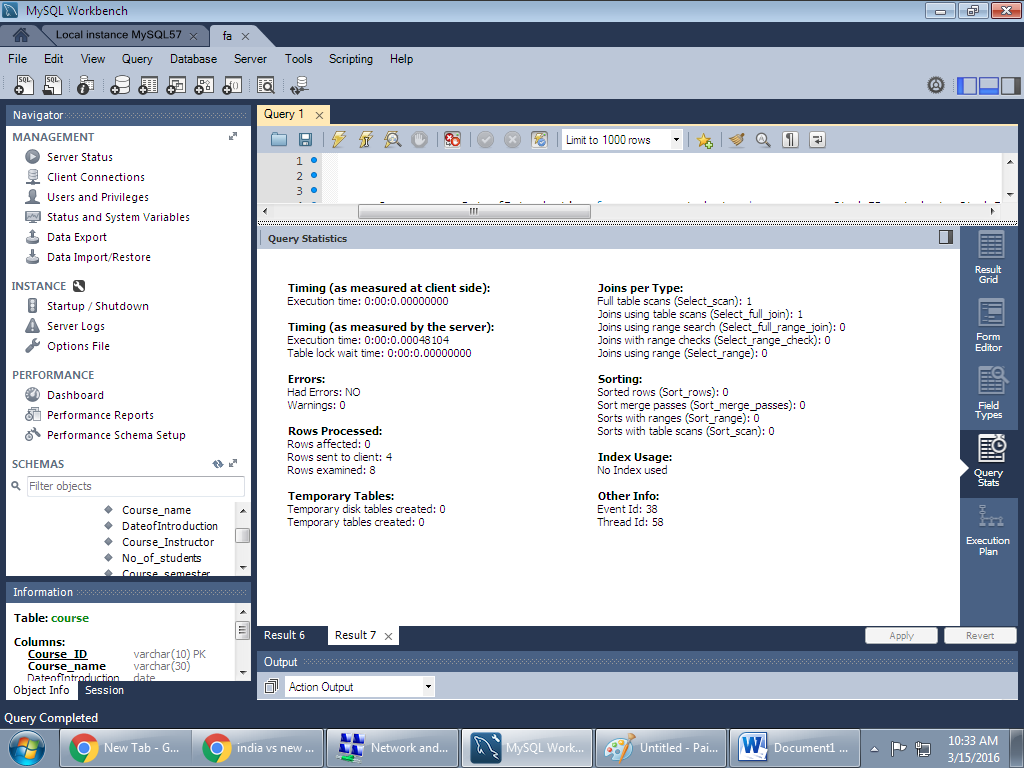
Aim: Implementation of query monitor.

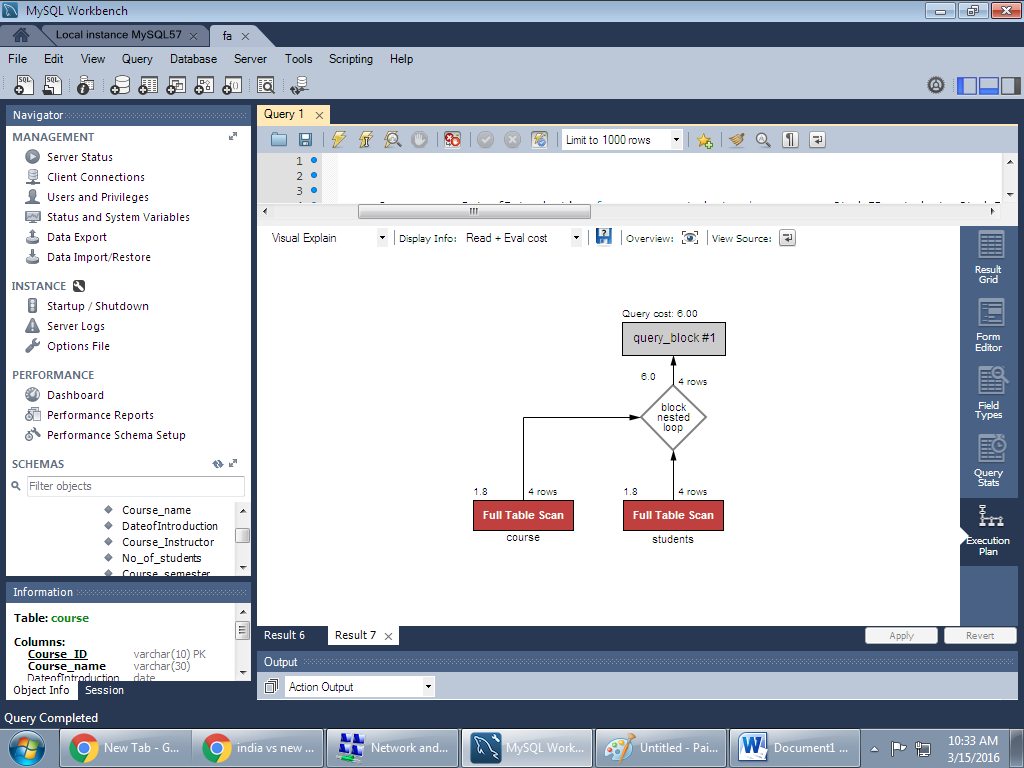
Labwork:

Join query on course and students tables

select Student\_Name,Address1,Dept\_name,Course\_name,DateofIntroduction from course,students where course.Stud\_ID= students.Stud\_ID;

1. 





(paste the screenshots of query execution plan from SQL server)

(write 4 to 5 lines about query execution plan)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 04

Aim: Perform Fragmentation

Labwork:

(paste the screenshots of all fragmentation queries you have written with output)

(Refer the oracle document which I had shared for this experiment and write about Fragmentation in 4 to 5 lines)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 05

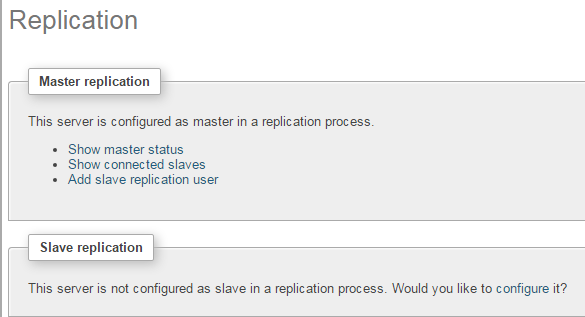
Aim: Implementation of Replication transparency in DDB.

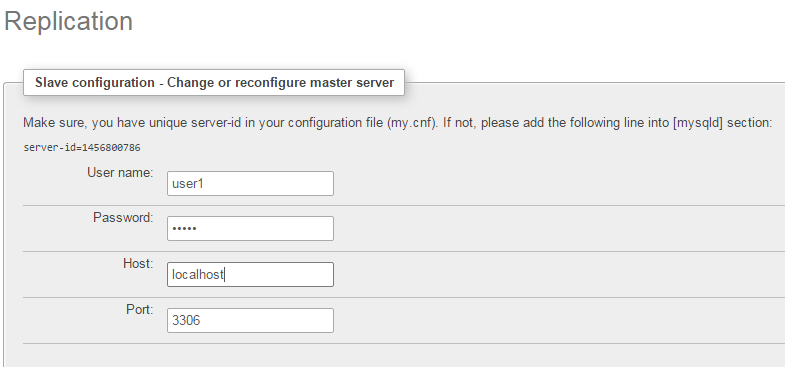
Labwork:

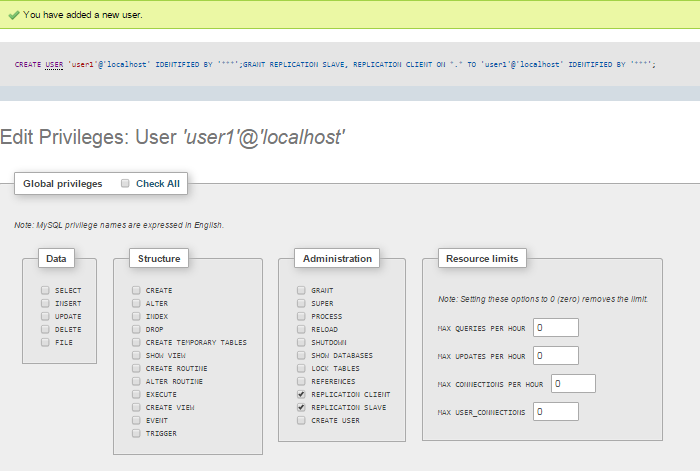
**Theory:** Replication is actually quite straightforward. At its core, it merely involves an administrator picking a server to act as the master, and then registering one or more slave servers to receive updates from the master. Each slave server is responsible for contacting the master server. This master server records all data manipulation statements in a binary log, which is then fed in a stream to any slave(s) that contact the master. The slave computers then play back these statements locally, thus updating their own data copies accordingly. In addition, a slave can, in turn, act as a master to other servers. This lets you construct sophisticated chains of replication servers.  
Obviously, there are many steps to follow to correctly configure and use replication, but the preceding discussion describes it accurately at a high level.  
  
**BENEFITS OF REPLICATION:**  
Replication is recommended if any of the following are met:  
1. high availability--the data stored on your MySQL server needs to be accessible 24 x 7  
2. Frequent backups--to protect against data loss, you often back up your databases.  
3. Mixed processing profiles--your MySQL database server must field requests from online transaction process (OLTP) and decision support system (DSS) users.  
4. Abundant, low-performance computers--your organization might not have the fastest computers, but they have lots of them.  
5. Widely dispersed users--your MySQL users are spread among multiple locations.  
6. Modular application code--your MySQL-based applications can be easily altered to read data from the slave servers while writing data to the master.

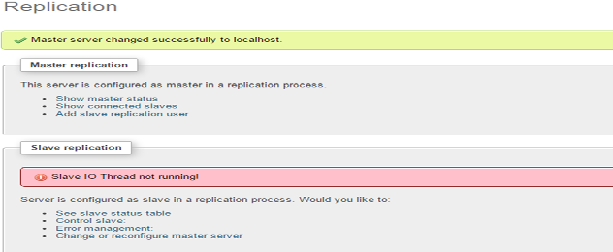
**Implementation:** Use following link for replication server configuration.

<https://www.youtube.com/watch?v=nfsmnx24gxU>

****







Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 06

Aim: Implementation of Two Phase/ Three Phase Commit Protocol.

Labwork:

2-PHASE COMMIT PROTOCOL:

In [transaction processing](http://en.wikipedia.org/wiki/Transaction_processing), [databases](http://en.wikipedia.org/wiki/Database), and [computer networking](http://en.wikipedia.org/wiki/Computer_networking), the two-phase commit protocol (2PC) is a type of [atomic commitment protocol](http://en.wikipedia.org/wiki/Atomic_commit) (ACP). It is a [distributed algorithm](http://en.wikipedia.org/wiki/Distributed_algorithm) that coordinates all the processes that participate in a [distributed atomic transaction](http://en.wikipedia.org/wiki/Distributed_transaction) on whether to [commit](http://en.wikipedia.org/wiki/Commit_(data_management)) or abort (roll back) the transaction (it is a specialized type of [consensus](http://en.wikipedia.org/wiki/Consensus_(computer_science)) protocol). The protocol achieves its goal even in many cases of temporary system failure (involving either process, network node, communication, etc. failures), and is thus widely utilized.

1. The commit-request phase (or voting phase), in which a coordinator process attempts to prepare all the transaction's participating processes (named participants,cohorts, or workers) to take the necessary steps for either committing or aborting the transaction and to vote, either "Yes": commit (if the transaction participant's local portion execution has ended properly), or "No": abort (if a problem has been detected with the local portion), and
2. The commit phase, in which, based on voting of the cohorts, the coordinator decides whether to commit (only if all have voted "Yes") or abort the transaction (otherwise), and notifies the result to all the cohorts. The cohorts then follow with the needed actions (commit or abort) with their local transactional resources (also called recoverable resources; e.g., database data) and their respective portions in the transaction's other output (if applicable).

3 PHASE COMMIT PROTOCOL:

In [computer networking](http://en.wikipedia.org/wiki/Computer_networking) and [databases](http://en.wikipedia.org/wiki/Database), the three-phase commit protocol (3PC) is a [distributed algorithm](http://en.wikipedia.org/wiki/Distributed_algorithm) which lets all nodes in a [distributed system](http://en.wikipedia.org/wiki/Distributed_system) agree to [commit](http://en.wikipedia.org/wiki/Commit)a [transaction](http://en.wikipedia.org/wiki/Database_transaction). Unlike the [two-phase commit protocol](http://en.wikipedia.org/wiki/Two-phase_commit_protocol) (2PC) however, 3PC is non-blocking. Specifically, 3PC places an upper bound on the amount of time required before a transaction either commits or [aborts](http://en.wikipedia.org/wiki/Abort_(computing))..

* 3PC prevents blocking in the absence of communications failures (unrealistic, but …). It can be made resilient to communications failures, but then it may block
* 3PC is much more complex than 2PC, but only marginally improves reliability — prevents some blocking situations.
* 3PC therefore is not used much in practice
* Main idea: becoming certain and deciding to commit are separate steps.
* 3PC ensures that if any operational process is uncertain, then no (failed or operational) process has committed.
* So, in the termination protocol, if the operational processes are all uncertain, they can decide to abort (avoids blocking).

2PC/3PC protocol code:

import java.io.\*;

import java.net.\*;

public class TPCClient1

{

public static void main(String a[])throws Exception

{

InetAddress lclhost;

lclhost=InetAddress.getLocalHost();

Client clnt=new Client(lclhost);

clnt.setSendPort(9001); //recport=8000

clnt.setRecPort(8000); //sendport=9001

clnt.recData();

clnt.sendData();

clnt.recData();

}

}

class Client

{

InetAddress lclhost;

int sendPort,recPort;

Client(InetAddress lclhost)

{

this.lclhost=lclhost;

}

public void setSendPort(int sendPort)

{

this.sendPort=sendPort;

}

public void setRecPort(int recPort)

{ this.recPort=recPort; }

public void sendData()throws Exception

{

BufferedReader br;

DatagramSocket ds;

DatagramPacket dp;

String data="";

System.out.println("Enter the Response 'VOTE\_COMMIT' || 'VOTE\_ABORT' ");

br=new BufferedReader(new InputStreamReader(System.in));

data = br.readLine();

System.out.println("Data is "+data);

ds=new DatagramSocket(sendPort);

dp=new DatagramPacket(data.getBytes(),data.length(),lclhost,sendPort-1000);

ds.send(dp);

ds.close();

}

public void recData()throws Exception

{

byte buf[]=new byte[256];

DatagramPacket dp;

DatagramSocket ds;

ds=new DatagramSocket(recPort);

dp=new DatagramPacket(buf,buf.length);

ds.receive(dp);

ds.close();

String msgStr=new String(dp.getData(),0,dp.getLength());

System.out.println("Client1 data " +msgStr);

};

Class server

public void setRecPort(int recPort)

{ this.recPort=recPort; }

public void sendData()throws Exception

{ BufferedReader br;

DatagramSocket ds;

DatagramPacket dp;

String data="";

System.out.println("Enter the Response 'VOTE\_COMMIT' || 'VOTE\_ABORT' ");

br=new BufferedReader(new InputStreamReader(System.in));

data = br.readLine(); System.out.println("Data is "+data);

ds=new DatagramSocket(sendPort);

dp=new DatagramPacket(data.getBytes(),data.length(),lclhost,sendPort-1000);

ds.send(dp);

ds.close(); }

public void recData()throws Exception

{ byte buf[]=new byte[256];

DatagramPacket dp;

DatagramSocket ds;

ds=new DatagramSocket(recPort);

dp=new DatagramPacket(buf,buf.length);

ds.receive(dp);

ds.close();

String msgStr=new String(dp.getData(),0,dp.getLength());

System.out.println("Client1 data " +msgStr); } };

OUTPUT:

(this code I had already shared at the time of experiment, paste the screenshots of output)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 07

Aim: Query Execution on XML database.

Labwork:

XML stands for EXtensible Markup Language.

XML was designed to describe data.

XML is a software- and hardware-independent tool for carrying information.

XSLT (eXtensible Stylesheet Language Transformations) is the recommended style sheet language for XML.

XSLT is far more sophisticated than CSS. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

XSLT uses XPath to find information in an XML document.

| The XSLT processor takes one or more XML source documents, plus one or more XSLT stylesheets, and processes them to produce an output document. In contrast to widely-implemented [imperative](http://en.wikipedia.org/wiki/Imperative_programming) programming languages like [C](http://en.wikipedia.org/wiki/C_(programming_language)), XSLT is declarative. This makes a given XSLT program more resilient to change to the input it is likely to receive, useful in a language used for information processing applications. The basic processing paradigm is pattern matching. Rather than listing an imperative sequence of actions to perform in a stateful environment, template rules only define how to handle a node matching a particular XPath-like pattern, if the processor should happen to encounter one, and the contents of the templates effectively comprise [functional](http://en.wikipedia.org/wiki/Functional_programming) [expressions](http://en.wikipedia.org/wiki/Expression_(programming)) that directly represent their evaluated form: the result tree, which is the basis of the processor's output.  The processor follows a fixed algorithm. First, assuming a stylesheet has already been read and prepared, the processor builds a source [tree](http://en.wikipedia.org/wiki/Tree_data_structure) from the input XML document. It then processes the source tree's root node, finds the best-matching template for that node in the stylesheet, and evaluates the template's contents. |
| --- |

HTML File Creation

A standard HTML file must have a file extension type of [ .html ]. You have a few options to create them, and it also depends on what operating system you use. Some people like to use plain text editors that have no robust markup features, while others will use a very robust text editing software tool.

XML documents use a self-describing and simple syntax:

<?xml version="1.0" encoding="UTF-8"?>  
<note>  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>

The first line is the XML declaration. It defines the XML version (1.0).

The next line describes the root element of the document (like saying: "this document is a note"):

<note>

The next 4 lines describe 4 child elements of the root (to, from, heading, and body):

<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>

And finally the last line defines the end of the root element:

</note>

.

XML Documents Form a Tree Structure

XML documents must contain a root element. This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

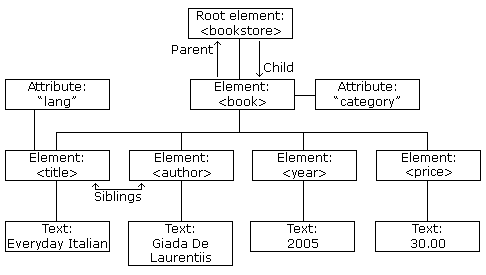
All elements can have sub elements (child elements):

<root>  
  <child>  
    <subchild>.....</subchild>  
  </child>  
</root>

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).

All elements can have text content and attributes (just like in HTML).

Example:



The image above represents one book in the XML below:

<bookstore>  
  <book category="COOKING">  
    <title lang="en">Everyday Italian</title>  
    <author>Giada De Laurentiis</author>  
    <year>2005</year>  
    <price>30.00</price>  
  </book>  
  <book category="CHILDREN">  
    <title lang="en">Harry Potter</title>  
    <author>J K. Rowling</author>  
    <year>2005</year>  
    <price>29.99</price>  
  </book>  
  <book category="WEB">  
    <title lang="en">Learning XML</title>  
    <author>Erik T. Ray</author>  
    <year>2003</year>  
    <price>39.95</price>  
  </book>  
</bookstore>

The root element in the example is <bookstore>. All <book> elements in the document are contained within <bookstore>.

The <book> element has 4 children: <title>,< author>, <year>, <price>.

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No: 08

Aim: Data Handling using JSON

Labwork:

(paste the screenshot of JSON object written by you and write about JSON in brief)

Conclusion: (write your observations and paste the screenshots if you have done anything extra)

Experiment No : 09

Aim: Database Security Issue.

Labwork:

Write about **ANY ONE** security issue in short

Conclusion: (write your observations and paste the screenshots if you have done anything extra)