
UNIT 2 EMERGING DATABASE MODELS, TECHNOLOGIES AND APPLICATIONS - II

Structure	Page Nos.
2.0 Introduction	21
2.1 Objectives	21
2.2 Mobile and Personal Database	22
2.2.1 The Basic Framework for Mobile Computing	
2.2.2 Characteristics of Mobile Databases	
2.2.3 Wireless Networks and Databases	
2.3 Web Databases	24
2.4 Accessing Databases on different DBMSs	28
2.4.1 Open Database Connectivity (ODBC)	
2.4.2 Java Database Connectivity (JDBC)	
2.5 Digital Libraries	32
2.6 Data Grid	33
2.7 Summary	35
2.8 Solutions/Answers	36

2.0 INTRODUCTION

Database applications have advanced along with the advancement of technology from age old Relational Database Management Systems. Database applications have moved to Mobile applications, Web database applications, Digital libraries and so on. The basic issues to all advanced applications are in making up-to-date data available to the user, in the desired format related anywhere, anytime. Such technology requires advanced communication technologies; advanced database distribution models and advanced hardware. In this unit, we will introduce the concepts of mobile and personal databases, web databases and the issues concerned with such databases. In addition, we will also discuss the concepts and issues related to Digital Libraries, Data Grids and Wireless Communication and its relationship with Databases.

Mobile and personal databases revolve around a mobile computing environment and focuses on the issues that are related to a mobile user. Web databases on the other hand, are very specific to web applications. We will examine some of the basic issues of web databases and also discuss the creation of simple web database development. We will examine some of the concepts associated with ODBC and JDBC – the two standards for connecting to different databases. Digital Libraries are a common source of information and are available through the databases environment. Such libraries are equipped to handle information dissemination, searching and reusability. Data grids allow distributed storage of data anywhere as one major application unit. Thus, all these technologies have a major role to play in the present information society. This unit discusses the issues and concerns of technology with respect to these databases system.

2.1 OBJECTIVES

After going through this unit, you should be able to:

- define the requirements of a mobile database systems;
- identify and create simple web database;
- use JDBC and ODBC in databases;
- explain the concept of digital libraries, and
- define the concept of a data grid.



2.2 MOBILE AND PERSONAL DATABASE

In recent years, wireless technology has become very useful for the communication of information. Many new applications of wireless technology are already emerging such as electronic valets which allow electronic money anywhere, anytime, mobile cells, mobile reporting, etc. The availability of portable computing devices that supports wireless base a communication helps database users access relevant information from anywhere, at anytime. Let us discuss mobile database systems and the issue related to it in this section.

2.2.1 The Basic Framework for Mobile Computing

Mobile Computing is primarily a distributed architecture. This basic architecture is shown in *Figure 1*.

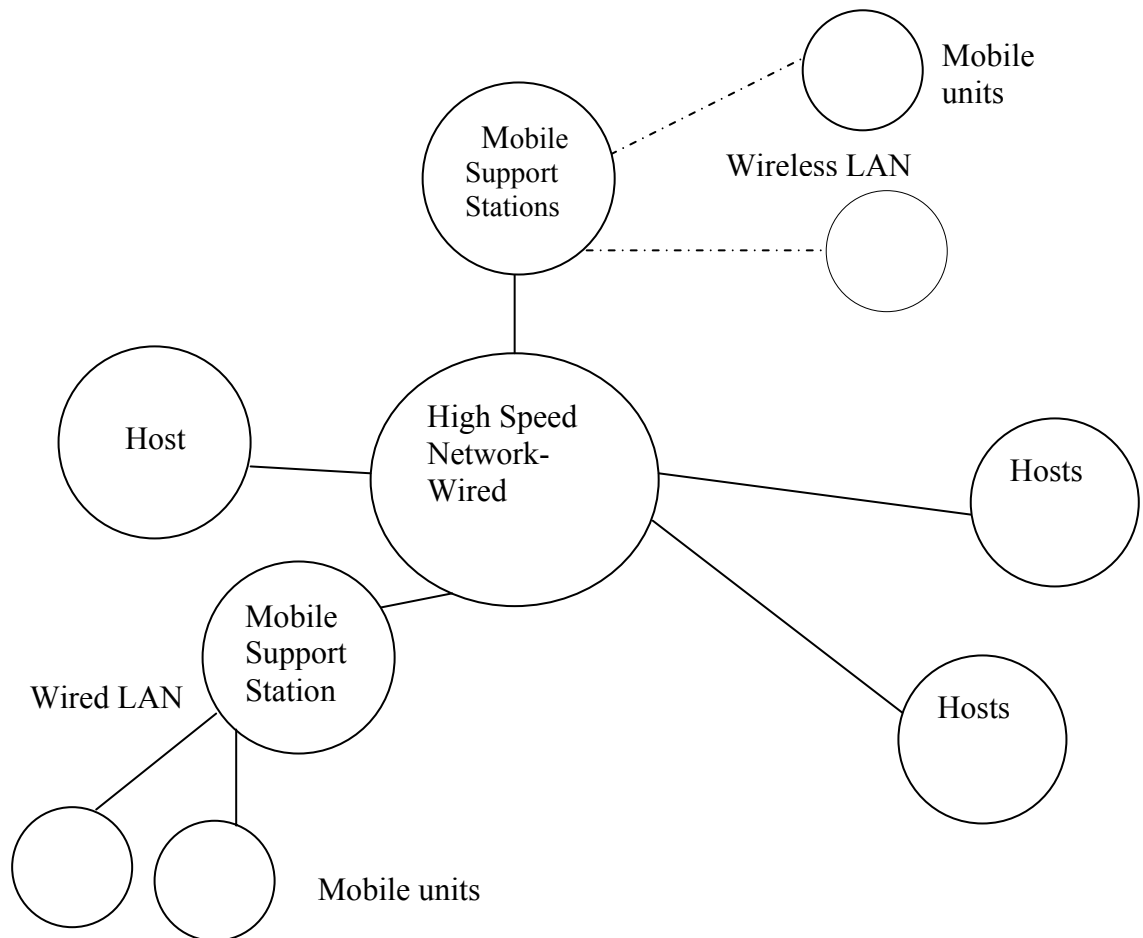


Figure 1: A typical view for Mobile Computing

A mobile computing environment consists of:

- host Computers which are fixed,
- mobile support stations,
- mobile client units, and
- networks.

Please note that the basic network may be a wired network but there are possibilities of Wireless LANs as well.



2.2.2 Characteristics of Mobile Databases

The mobile environment has the following characteristics:

- 1) *Communication Latency*: Communication latency results due to wireless transmission between the sources and the receiver. But why does this latency occur? It is primarily due to the following reasons:
 - a) due to data conversion/coding into the wireless formats,
 - b) tracking and filtering of data on the receiver, and
 - c) the transmission time.
- 2) *Intermittent wireless connectivity*: Mobile stations are not always connected to the base stations. Sometimes they may be disconnected from the network.
- 3) *Limited battery life*: The size of the battery and its life is limited. Information communication is a major consumer of the life of the battery.
- 4) *Changing location of the client*: The wireless client is expected to move from a present mobile support station to an other mobile station where the device has been moved. Thus, in general, the topology of such networks will keep on changing and the place where the data is requested also changes. This would require implementation of dynamic routing protocols.

Because of the above characteristics the mobile database systems may have the following features:

Very often mobile databases are designed to work offline by caching replicas of the most recent state of the database that may be broadcast by the mobile support station. The advantages of this scheme are:

- it allows uninterrupted work, and
- reduces power consumption as data communication is being controlled.

However, the disadvantage is the inconsistency of data due to the communication gap between the client and the server.

However, some of the challenges for mobile computing are:

- (i) *Scalability*: As the number of stations increase, latency increases. Thus, the time for servicing the client also increases. This results in increase in latency, thus more problems are created for data consistency.

The solution: Do data broadcasting from many mobile stations thus, making the most recent information available to all, thus eliminating the enough latency time.

- (ii) *Data Mobile problem*: Client locations keeps on changing in such networks thus, keeping track of the location of the client is important for, the data server and data should be made available to the client from the server which is minimum latency way from the client.

2.2.3 Wireless Networks and Databases

A mobile support station covers a large geographic area and will service all the mobile hosts that are available in that wireless geographical area – sometimes referred to as the cell. A mobile host traverses many mobile zones or cells, requiring its information to be passed from one cell to another, not necessarily adjacent, as there is an overlapping of cells. However, with the availability of wireless LANs now, mobile hosts that are in some LAN areas may be connected through the wireless LAN rather than a wide area cellular network. Thus, reducing the cost of communication as well as overheads of the cell movement of the host.



With wireless LAN technology it is now possible that some of the local mobile hosts may communicate directly with each other without the mobile support station. However, please note that such a communication should be based on some standard protocol/technology. Fortunately, the Blue tooth standard is available. This standard allows wireless connectivity on short ranges (10-50 metres) and has nearly 1 megabytes per second speed, thus allowing easy use of PDA mobile phones and intelligent devices. Also there are wireless LAN standards such as 801.11 and 802.16. In addition, wireless technology has improved and packet based cellular networking has moved to the third generation or is even more advanced allowing high-speed digital data transfer and applications. Thus, a combination of all these technologies viz., blue tooth, wireless LANs and 3G cellular networks allows low cost infrastructure for wireless communication of different kinds of mobile hosts.

This has opened up a great potential area for mobile applications where large, real-time, low cost database applications can be created for areas such as, just in time accounting, teaching, monitoring of resources and goods etc. The major advantage here would be that the communication of real time data through these networks would now be possible at low costs.

Major drawback of mobile databases – is the limitation of power and available size of the display unit has found newer technologies like use of flash, power-saving disks, low-powered and power saving displays. However, the mobile devices since are normally smaller in size requires creation of presentation standards. One such protocol in the areas of wireless networking is the Wireless Access Protocol (WAP).

Thus, mobile wireless networks have opened up the potential for new mobile database applications. These applications may be integrated into very strong web-based, intelligent, real time applications.

2.3 WEB DATABASES

The term web database can be used in at least two different way:

Definition 1: A web database may be defined as the organised listing of web pages for a particular topic. Since the number of web pages may be large for a topic, a web database would require strong indexing and even stronger spider or robot based search techniques.

We are all familiar with the concept of web searching. Web searching is usually text based and gathers hundreds or thousands of web pages together as a result of a search. But how can we sort through these pages solve this problem? How do the web database could help in this regard. However, this is not the basic issue for the discussion in this section. We would like to concentrate on the second definition of a web database.

Definition 2: A web database is a database that can be accessed through the web.

This definition actually defines a web application with the database as the backend. Let us discuss more about web database application systems.

Features of a Web Database Application

A Web database application should have the following features:

- it should have proper security,
- it should allow multiple users at a time,
- it should allow concurrent online transactions,
- it should produce a response in finite time,



- its response time should be low,
- it should support client – server (2-tier or 3 tier) architecture through the website.
- on the three-tier architecture an additional application server tier operates between the client and server.

Creating a Web Database Application

How do we create a web database? Well, we would like to demonstrate this with the help of a very simple example. To implement the example, we use Active Server Pages (ASP), which is one of the old popular technologies with the backend as Ms-Access. ASP is a very easy technology to use. Although, Microsoft Access does not support websites with a lot of traffic, it is quite suitable for our example. A web database makes it very convenient to build a website.

Let us show the process of creating a web database. We would need to follow the following steps to create a student database and to make it accessible as a web database.

- The first step is to create a database using Ms-Access with the following configuration:

Student-id	Text (10)	Primary Key
Name	Text (25)	
Phone	Text (12)	

Now, you would need to enter some meaningful data into the database and save it with the name students.mdb.

- Put your database online by using ftp to transfer students.mdb to the web server on which you are allowed access. Do not put the file in the same directory in which your web site files are stored, otherwise, the entire databases may be downloaded by an unauthorised person. In a commercial set up it may be better to keep the data on the Database server. This database then can be connected through a Data Source Name (DSN) to the website. Let us now build the required interface from ASP to the Database. A simple but old method may be with connecting ASP using ActiveX Data Object (ADO) library. This library provides ASP with the necessary functionality for interacting with the database server.
- The first and most basic thing we need to do is to retrieve the contents of the database for display. You can retrieve database records using ADO Recordset, one of the objects of ADO.

```
Dim recordsettest
Set recordsettest = Server.CreateObject("ADODB.Recordset")
```

The commands given above create a variable (recordsettest) to store new Recordset object using the Server object's CreateObject method.

- Now fill this Recordset with records from the database with its Open method. Open takes two parameters:
 - the table name that contains the records to be fetched and
 - the connection string for the database.

Now, the name of the table is straight forward, as we would obviously, create a table with a name. However, the connection string is slightly more complex. Since the ADO library is capable of connecting to many database servers and other data sources, the string must tell Recordset not only where to find the database (the path and file name) but also how to read the database, by giving the name of its database provider.



A database provider is a software that has a very important role. It allows ADO to communicate with the given type of database in a standard way. ADO the provider for MS-Access, SQL Server, Oracle, ODBC database servers etc. Assuming that we are using the provider Jet.OLEDB to connect to an Access database, the connection string would be:

```
Provider = Microsoft.Jet.OLEDB.version; Data Source = ~\student\student.mdb
```

A connection string for MS SQL may be like:

```
Provider = SQLOLEDB; Data Source = servername; Initial Catalog = database name;  
User Id = username; Password = password.
```

So a sample code could be:

```
Dim recordsettest  
Dim db_conn 'The database connection string  
recordsettest= Server.CreateObject ("ADODB.Recordset")  
db_conn = "Provider=Microsoft.Jet.OLEDB.version; Data Source=  
~\student\student.mdb  
recordsettest.Open "student", db_conn
```

However, since many ASP pages on the site will require such a string, it is common to place the connection string in an application variable in the global workspace.

```
<SCRIPT LANGUAGE= "VBScript" RUNAT= "Server">  
Sub Application_OnStart ( )  
Dim db_conn  
db_conn= "Provider" = Microsoft.Jet.OLEDB.version;  
Data Source= ~\student\student.mdb Application ("db_conn") = db_conn  
End Sub  
</SCRIPT>
```

The code to retrieve the contents of the student table will now include only the code for Recordset.

```
Dim recordsettest  
Set recordsettest = Server.CreateObject("ADODB.Recordset")  
recordsettest.Open "Student", Application("db_conn")
```

Thus, we have established the connection and can get information from the MS-Access database. But another problem remains. How do we display the results?

- Now, we can access recordsets, like of the database table, the result sets with the data row as one database record. In our example, we have put the contents of the student table into the recordsetrest object. An opened Recordset keeps track of the current record. Initially the first record is the current record. A MoveNext method of the Recordset object moves the pointer to the next record in the set, if any. An EOF property of the Recordset will become true at the end of Recordset. Thus, to display student id and name, you may write the following code:

```
Do While Not recordsettest.EOF  
Response. Write "<li> "& recordsettest ("students-id") &"  
Response. Write "<p> "& recordsettest ("name") &" </p> </li>  
recordsettest('name)  
recordsettest.MoveNext  
Loop  
If recordsettest.BOF Then
```



Response. Write “<p>No students data in the database. </p>”

Please note that the BOF property checks if the file is empty.

- Once you have completed the task then you must be close the recordset as:
Recordsettest.Close
This command sets free the connection to the database. As these connections may not be very large in numbers, therefore, they need not be kept open longer than it is necessary.

Thus, the final code may look like:

```
<html>
<head>
<title> Student Data</title>
</head>
<body>
<ol>
<%
Dim recordsettest
Set recordsettest = Server.CreateObject("ADODB.Recordset")
recordsettest.Open "Student", Application("db_conn")
Do While Not recordsettest.EOF
    Response. Write "<li> "& recordsettest ("students-id") &"
    Response. Write "<p> "& recordsettest ("name") &" </p> </li>"
    recordsettest("name")
    recordsettest.MoveNext
Loop
If recordsettest.BOF Then
    Response. Write "<p>No students data in the database. </p>"
Recordsettest.Close
```

Save this file on the web server. Now, test this program by storing suitable data in the database. This application should produce the simple list of the student. However, you can create more complex queries to data using SQL. Let us try to explain that with the help of an example.

Example: Get the list of those students who have a phone number specified by you. More than one student may be using use this phone number as a contact number. Please note that in actual implementation you would need to create more complex queries.

Please also note that the only change we need to perform in the programming given above is, in the Open statement where we need to insert SQL command, such that on opening the recordset only the required data from the database is transferred to it. Thus, the code for this modified database access may be:

```
<html>
<head>
<title> Student Data</title>
</head>
<body>
<ol>
<%
Dim recordsettest
Set recordsettest = Server.CreateObject("ADODB.Recordset")
recordsettest.Open "SELECT student-id, name FROM Student
WHERE phone = " & Request ("phone") ",
Application("db_conn")
```



```
Do While Not recordsettest.EOF
    Response.Write "<li> "& recordsettest ("students-id") &"
    Response.Write "<p> "& recordsettest ("name") &" </p> </li>"
    recordsettest('name)
    recordsettest.MoveNext
Loop
If recordsettest.EOF Then
    Response.Write "<p>No students data in the database. </p>"
Recordsettest.Close
```

You can build on more complex queries. You may also refer to more advanced ASP versions and connections. A detailed discussion on these is beyond the scope of the unit.

☞ Check Your Progress 1

- 1) What are the different characteristics of mobile databases?

.....

.....

.....

- 2) What are the advantages of using wireless LAN?

.....

.....

.....

- 3) What are the steps required to create a web database?

.....

.....

.....

2.4 ACCESSING DATABASES ON DIFFERENT DBMSS

Database access from different interfaces may require some standards. The two standards that are quite useful are ODBC for accessing a database from any DBMS environment that support ODBC and JDBC that allows JAVA program access databases. Let us discuss them in more detail in this unit.

2.4.1 Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) is a standard API (Application Programming Interface) that allows access to as database that is a part of any DBMS that supports this standard. By using ODBC statements in your program you may access any database in MS-Access, SQL Server, Oracle, DB2, even an Excel file, etc. to work with ODBC driver for each of the database that is being accessed from the ODBC. The ODBC allows program to use standard SQL commands for accessing database. Thus, you need not master the typical interface of any specific DBMS.

For implementing ODBC in a system the following components will be required:

- the applications themselves,
- a core ODBC library, and
- the database drives for ODBC.



The library functions as the transitory/ interpreter between the application and the database drivers. Database drivers are written to hide DBMS specific details. Such layered configurations allow you to use standard types and features in your DBMS applications without, knowing details of specific DBMS that the applications may encounter. It also simplifies the implementations of database driver as they need to know only the core library. This makes ODBC modular. Thus, ODBC provides the standard data access using ODBC drivers that exist for a large variety of data sources including drivers for non-relational data such as spreadsheet, XML files.

ODBC also has certain disadvantages. These are:

- if a large number of client machines require different drivers, and DLLs are to be connected through ODBC then it has a complex and large administration overhead. Thus, large organisations are on the lookout for server side ODBC technology.
- the layered architecture of ODBC may introduce minor performance penalty.

The Call Level Interface (CLI) specifications of SQL are used by the ODBC as its base. The ODBC and its applications are becoming stronger. For example, Common Request Broker Architecture (CORBA) that is distributed object architecture, and the Persistent Object Service (POS) – an API, are the superset of both the Call-Level Interface and ODBC. If you need to write database application program in JAVA, then you need Java database Connectivity (JDBC) application program interface. From JDBC you may use a JDBC-ODBC “bridge” program to reach ODBC-accessible database.

The designers of ODBC wanted to make it independent of any programming language, Operating System or DBMS.

Connecting Database to ODBC

A database application needs to have the following types of statements for connecting to ODBC:

- linkage of core ODBC Library to client program,
- call results in communication to the request of the client library to the server, and
- results fetched at the client end.

But how do we actually write a code using ODBC? The following steps are needed to write a code using ODBC:

- establish the connection,
- run a SQL query that may be written by using embedded SQL, and
- release the connection.

But how do we establish the connection? Some of the ways of establishing a connection are:

An ODBC/DSN connection to MySQL using ASP:

```
Dim db_conn
set db_conn = server.createobject("adodb.connection")
db_conn.open("dsn=DSNmysql; uid = username; pwd = password; database = student")
```



What is a DSN in the connection above? It is a Data Source Name that has been given to a specific user connection. DSN are more secure as you have to be the user as defined in DSN otherwise, you will not be allowed to use the data.

ODBC setting for Oracle:

To set a connection you may need to edit the `odbc.ini` file in certain installations of Oracle. The basic objective here is to allow you to connection to the relational data source of Oracle. Let us assume that we are using Oracle Version 8 and trying to connect to an Oracle data source `oradata` using an ODBC driver. You need to write the following:

```
[ODBC data Sources]
oradata=Oracle8 Source Data
...
[oradata]
Driver = / . . . /odbc/drivename
Description=my oracle source
ServerName=OracleSID
```

The last line defines the name of an Oracle database defined in the environment file of Oracle.

Once the connection is established through the required ODBC driver to the database through, the user-id and password, you can write the appropriate queries using SQL. Finally, you need to close the connection using `close()` method.

2.4.2 Java Database Connectivity (JDBC)

Accessing a database in Java requires, Java Database Connectivity (JDBC). JDBC allows you to accesses a database in your application and applets using a set of JDBC drivers.

What is JDBC?

Java Database Connectivity (JDBC) provides a standard API that is used to access databases, regardless of the DBMS, through JAVA. There are many drivers for JDBC that support popular DBMSs. However, if no such driver exists for the DBMS that you have selected then, you can use a driver provided by Sun Microsystems to connect to any ODBC compliant database. This is called JDBC to ODBC Bridge. For such an application, you may need to create, an ODBC data source for the database before, you can access it from the Java application.

Connecting to a Database

In order to connect to a database, let us say an oracle database, the related JDBC driver has to be loaded by the Java Virtual Machine class loader successfully.

```
// Try loading the oracle database driver
try
{
    Class.forName("oracle.jdbc.driver.OracleDriver").newInstance( ) ;
}
catch (ClassNotFoundException ce) // driver not found
{
    System.err.println ("Drivernotfound");
    // Code to handle error
}
```



Now, you can connect to the database using the driver manager class that selects the appropriate driver for the database. In more complex applications, we may use different drivers to connect to multiple databases. We may identify our database using a URL, which helps in identifying the database. A JDBC URL starts with “jdbc:” that indicates the use of JDBC protocol.

A sample database URL may be

```
jdbc:oracle@db.ignou.ac.in:2000:student
```

To connect to the database, we need to connect with a username and password. Assuming it to be “username” and “password”, the connection string would be:

```
// Try creating a database connection
Connection db_conn =
    DriverManager.getConnection
        (jdbc:oracle@db.ignou.ac.in:2000:student, “username”, “password”)
```

Thus, you will now be connected. Now the next step is to execute a query.

You can create a query using the following lines:

```
Statement stmt = db_conn.createStatement()
try {
    stmt.executeQuery (
        // Write your SQL query using SQL and host language
    )
} catch (.....) {.....}
stmt.close ()
db_conn.close ()
```

Thus, the JDBC standard allows you to handle databases through JAVA as the host language. Please note that you can connect to any database that is ODBC compliant through JAVA either through the specialised driver or through the JDBC – ODBC bridge if no appropriate driver exists.

☞ Check Your Progress 2

1) Why is there a need for ODBC?

.....

.....

.....

2) What are the components required for implementing ODBC?

.....

.....

.....

3) Why do you need JDBC? What happens when a DBMS does not have a JDBC driver?

.....

.....

.....



2.5 DIGITAL LIBRARIES

Let us now, define a digital library.

“ A digital library is a library that allows almost the same functionality as that of a traditional library, however, having most of its information resources in digital form that are stored using multimedia repositories. The information to the digital library may be through the web or intranet.”

The following may be some of the objectives of the digital library:

- to allow online catalog we access,
- to search for articles and books,
- to permit search on multiple resources at a time,
- to refine the search to get better results,
- to save the search results for future use, and
- to access search results by just clicking on the result.

Please note that most of the objectives above can easily be fulfilled due to the fact that a digital library allows web access.

But what are the advantages of the digital library?

A digital library has the following advantages:

- very large storage space,
- low cost of maintenance,
- information is made available in rich media form,
- round the clock operation,
- multi-user access, and
- can become a part of the world knowledge base/knowledge society.

Now the next question is what is the cost involved in creating and maintaining Digital Libraries?

The following are the cost factors for the creation and maintenance of digital libraries:

- cost of conversion of material into digital form,
- cost of maintaining digital services including cost of online access, and
- cost of maintains archival information.

So far, we have seen the advantages of digital libraries, but do digital libraries have any disadvantages?

Following are some of the disadvantages of digital libraries:

- only information available public domain can be made available,
- a digital library does not produce the same environment as a physical library, and
- the person needs to be technology friendly in order to use the library.

At present many Universities offer digital library facility online.

Functionality of Digital library

A digital library supports the following functions:

- Searching for information: A digital library needs a very strong search facility. It should allow search on various indexes and keyword. It should have a distributed search mechanism to provide answers to individual users search.



- Content management: A digital library must have facility for the continuous updating of source information. Older a information in such cases may need to be archived.
- Licenses and rights management: Only appropriate/authorised users are given access to protected information. A library needs to protect the copyright.
- All the links to further information should be thoroughly checked
- The library should store the Meta data in a proper format. One such structure for storing the meta data is the Dublin core.
- Library information should be represented in standard data formats. One such format may be XML. The contents may be represented in XML, HTML, PDF, JPEG, GIF, TIFF etc.

Technology Required for the Digital Library

A digital library would require expensive technology. Some of these requirements may be:

- Very large, reliable storage technologies supporting tera or even peta bytes of information. You may use Storage Access Networks (SAN).
- The transfer rate of information from the storage point to the computer should be high in order to fulfil the request of many users.
- High Internet bandwidth to store many users at a time.
- A distributed array of powerful servers that may process large access requests at the same time.
- Very reliable library software. Such software are developed on top of RDBMS to support features of full text indexing, meta-data indexing etc.

2.6 DATA GRID

Let us first define the term data grid. The concept of a data grid is somewhat difficult to define. A data grid can be seen as the process of creating a virtual database across hardware of almost the entire data that exists in some form. Thus, the key concept is Virtual Database. The basic tenet behind a data grid is that an application need not know either the place or the DBMS where the data is stored; rather the application is only interested in getting the correct results. *Figure 2* shows a typical structure for a data grid.

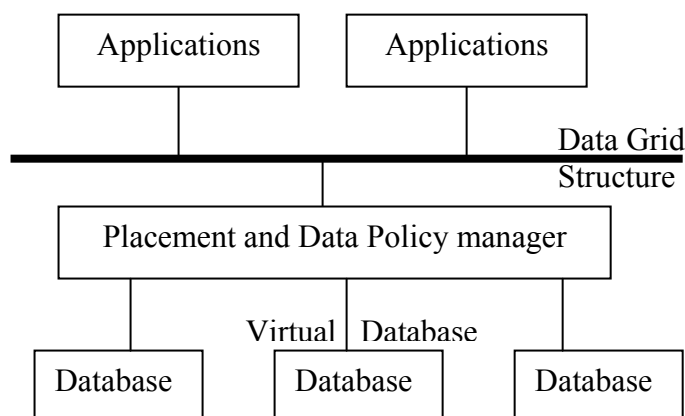


Figure 2: A Data Grid



A data grid should address the following issues:

- It should allow data security and domain specific description of data. Domain specific data helps in the identification of correct data in response to a query.
- It should have a standard way of representing information. [The possible candidates in this category may be XML].
- It should allow simple query language like SQL for accessing information.
- The data requirements should be fulfilled with a level of confidence, (that is there has to be a minimum quality of service in place).
- There needs to be a different role assigned to the data administrator. The data administrators should not be concerned with what the data represent. Data domain specification should be the sole responsibility of the data owner. Unfortunately, this does not happen in the present database implementations.
- The separation of the data administration and the data manager will help the database administrator to concentrate on data replication and query performance issues based on the DBMS statistics.

Thus, a data grid virtualises data. Many DBMSs today, have the ability to separate the roles of the database administrator and the data owner from the application access. This needs to be extended to the grid across a heterogeneous infrastructure.

What are the implications of using a data grid?

An application written in a data grid, references a virtual data source with a specified quality of service. Such an application need not be compiled again even if the data moves from one server to another or, there is a change in the infrastructure or due to changes in the data access methods. This happens as in the data grid, data sources are transparent to the application. Thus, it moves the concept of data independence further up the ladder.

A data grid permits a data provider with facilities that need not explicitly mention the location and structure of current and future applications. This information can be published in the data dictionary or Registry. A grid middleware then, is needed, to query the registry in order to locate this information for the applications. An administrator in a data grid is primarily concerned with the infrastructure and its optimum uses and providing the required qualities of service. Some of the things that the administrator is allowed to do based on the statistics stored in the data dictionary includes, - enabling replication, failure recovery, partitioning, changing the infrastructure etc.

Data Grid Applications

A data grid includes most of the relational database capabilities including schema integration, format conversion of data, distributed query support etc. In addition, a data grid should be scalable, reliable and should support efficient file access. These things are not easy to do. However, in this sub-section let us try to define some of the applications that a data grid may have.

A Medical Grid: Consider a situation in which a patient is admitted to a hospital where s/he has no record. By querying the data grid on a key field (may be a voter's id number) all the information about the previous medical history of the patient can be obtained using the data grid of hospitals. The data grid of hospitals may have their independent database systems, which is a part of the data grid. A hospital need not hand over all the confidential data as part of the data grid. Please note that the main



feature here is that the hospital is in complete control of its data. It can change, hide, and secure any part of its own database while, participating in the data grid federation. Thus, a virtual medical database that is partitioned across hospitals can be made. Please note that a query can be run across the whole set of hospitals and can retrieve consolidated results. A query may not need to retrieve data from all the hospitals participating in the grid to get significant information, for example, a query about the symptoms of a disease that has been answered by 70 percent of the hospitals can return meaningful results. However, on the other hand some queries would require all the data grid members to participate. For instance, a query to find the patient's complete medical record wherever its parts are stored, would require all the hospitals to answer.

Some of the major requirements of a data grid are:

- *Handling of failure of a data source:* A grid may have replicas and caches, but grid applications tend to access a data resource. But needs to be happens when this data source fails? A data grid flexible and it should have a middleware that automatically moves the operations to either another data resource with a similar data set or to multiple data resources each with a subset of the data.
- *Parallel access for local participant:* Some organisations may have very large data, thus, a query on that would take longer and such a query could be beyond the performance criteria set for the grid. Therefore, it may be a good idea to use a "virtual distributed" database to fetch data in parallels and keep processing it.
- *Global access through SQL:* A data grid would require dynamic selection of data sources. Therefore, it requires a complete SQL query transformation and optimisation capability.
- A data grid application may need to access data sources such as content management systems, Excel files, or databases not yet supported.

Check Your Progress 3

- 1) What are the basic functions of a digital library?

.....

.....

.....

- 2) What are the advantages of Data Grid?

.....

.....

.....

- 3) What are different types of hardware and software required for digital libraries?

.....

.....

.....

2.7 SUMMARY

This unit introduced you to several concepts related to emerging database applications. This unit also provides insight into some of the practical issues on connecting to databases from any DBMS or using JAVA, as well as analysed simple applications related to web database.



The unit introduces the requirements of mobile databases. Although most of the mobile applications in the past work on data broadcasting, however, this may change in the era when new wireless LAN technologies are available. This may give rise to new real time database applications. The ODBC and JDBC are two standards that can be used to connect to any database of any DBMS on any operating system and JAVA respectively. A web database may be the backend for the browser front end or the browser – application server tiers. However, it basically provides access to the data through the web. Digital libraries are a very useful application that has emerged in recent years. Digital libraries follow some of the meta-data and storage standards. The digital libraries also support distributed search. A Data grid is a virtual database created for the purpose of information sharing. The grid is loosely controlled.

2.8 SOLUTIONS/ANSWERS

Check Your Progress 1

- 1) The following are the characteristics of mobile databases:
 - Mobile database relies on the broadcast of data
 - Mobile stations may be working in standalone mode most of the time
 - The data on a mobile workstation may be inconsistent
 - Mobile database may be made scalable
 - Mobile database are closer to distributed database technology
 - A mobile unit may be changing its physical location. It is to be reached at all locations.
- 2) Wireless LANs may allow low cost communication between two mobile units that are located in the same LAN area. Thus, it may result in reduced cost of operation.
- 3) The following steps are required to create a web database:
 - Create a database and put it on a database server.
 - Create a connection string to connect to the server through a valid username and password
 - Open the connection to bring in the required data based on the suitable query interactions
 - Format and display the data at the client site.

Check Your Progress 2

- 1) ODBC allows using standard SQL commands to be used on any database on any DBMS. It gives application designer freedom from learning the features of individual DBMS, or OS etc. Thus, it simplifies the task of database programmers.
- 2) The following three components are required to implement ODBC:
 - The application
 - The core ODBC library
 - The database driver for ODBC
- 3) JDBC is an API that allows Java programmers to access any database through the set of this standard API. In case a JDBC driver is not available for a DBMS then the ODBC-JDBC bridge can be used to access data.



Check Your Progress 3

- 1) A digital library supports
 - Content management
 - Search
 - License management
 - Link management
 - Meta data storage
- 2) Data Grid is helpful in the sharing of large amount of information on a particular topic. Thus, allowing a worldwide repository of information, while, on the other hand, still giving full control of information to the creator.
- 3) Digital libraries require:
 - Very large secondary storage
 - Distributed array of powerful servers
 - Large bandwidth and data transfer rate
 - A reliable library software.