Web Databases

A web database is a database that can be queried and/or updated through the World Wide Web (WWW). As web technologies are evolving, the WWW turned out to be the preferred medium for many applications, e.g., e-commerce and digital libraries. These applications use information that is stored in huge databases and can only be retrieved by issuing direct queries to the back-end databases. Database-driven web sites have their own interfaces and access forms that create HTML pages on-the-fly. Web database technologies define the way that these forms can connect to and retrieve data from database servers.

The number of database-driven web sites is growing exponentially. The dynamically created web pages by these sites are hard to be reached by traditional search engines. Traditional search engines are used to crawl and index static HTML pages. However, traditional search engines cannot send queries to web databases. The hidden information inside the web database sources is called the "deep web" in contrast to the "surface web" that is easily accessed by traditional search engines.

Database Connectivity:

Querying the database via direct SQL is one of the most common ways to access data from a database. The need for a standard way to query different databases arises as the number of existing database servers is very huge and they differ in their query interfaces. *Open DataBase Connectivity (ODBC)* was developed by Microsoft as a standard access method to different databases. The ODBC interface defines:

- A standard way to connect and log on to a DBMS.
- A standardized representation for data types.
- Libraries of ODBC API function calls that allow an application to connect to a DBMS, execute SQL statements, and retrieve results.

Using ODBC, writing a program to read data from a database does not need to target a specific DBMS. All what is needed is to use the database vendor's supplied *ODBC driver* to link the code to the required database.

ODBC Levels of Conformance:

To provide a way for applications and drivers to implement portions of the ODBC API specific to their needs, the ODBC standard defines conformance levels for drivers in two areas; the *ODBC API* and the *ODBC SQL* grammar. There are three *ODBC API* conformance levels: Core, Level 1 and Level 2. Each level of conformance supports a different set of ODBC functions. After the success of ODBC, Microsoft introduced **OLE DB,** an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data not only data stored in DBMS.

Java DataBase Connectivity (JDBC):

Later on, the "Write Once, Run Anywhere" Java Platform calls for a new java standard for database connectivity. **Java DataBase Connectivity (JDBC)** is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides the same functionalities as ODBC but may be invoked from inside java programs. **JDBC-ODBC bridges** are drivers that provide JDBC access via ODBC drivers, which is widely used by developers to connect to databases in a non-Java environment.

Database-to-Web Connectivity:

Several Database-to-Web connectivity technologies took place. The various technologies differ in who sends queries to the database. Mainly, a web database environment contains a web browser (the client), a web server (understands HTTP) and the DBMS (understands SQL). Database-to-web connectivity asks for a middleman that can understand both HTTP and SQL and can do the transformations between them back and forth. Database-to-Web connectivity technologies are either two-tier or three-tier technologies. Figure 1 gives a simple view of the system architecture in both technologies.

Two-Tier technologies:

In two-tier architecture, only two tiers are involved in the database-to-web integration. The client tier (web browser and web server) and the server tier (DBMS). The following are three different two-tier database-to-web technologies:

Common Gateway Interface (CGI):

CGI is a program that runs on the server tier and is responsible for all the transformations between HTTP and SQL. A link to the CGI program exists in the web database access form. When the HTML form is referenced by a client, the web server extracts the query parameters from it and forwards them to the CGI program on the server tier. The CGI program reads the parameters, formats them in the appropriate way, and sends a query to the database. Then, the result of the query is sent back to the client tier as HTML pages formatted by the CGI program. CGI programs can be written in many languages, e.g., Perl and JavaScript.

Java Applets: Java applets are java programs that can be dynamically loaded by the browser and executed in the client site. Applets use JDBC to connect directly to the DBMS via sockets (no web server is needed).

Server Side Includes (SSI): SSI is a code that is written inside an HTML page and is processed by the web server. When the HTML page is invoked, the web server executes the scripts, that in turn use ODBC or JDBC to read data from the DBMS, and then the web server formats the data into HTML pages and sends it back to the browser. Example SSI scripting languages are ASP, ASP.NET, JSP, PHP and ColdFusion.

Three-Tier Technologies:

In three-tier architecture a Middleware tier (*the Application Server*) is added between the client tier and the server tier. The middleware is responsible to transfer data between the web server and the DBMS. The application server handles all the application operations and connections for the clients. Application servers offer other functions such as transaction management and load balancing. Sometimes the application server and web server are merged into the *Web Application Server*. Examples of application servers are IBM Websphere, Oracle 9i, and Sun ONE Application servers.
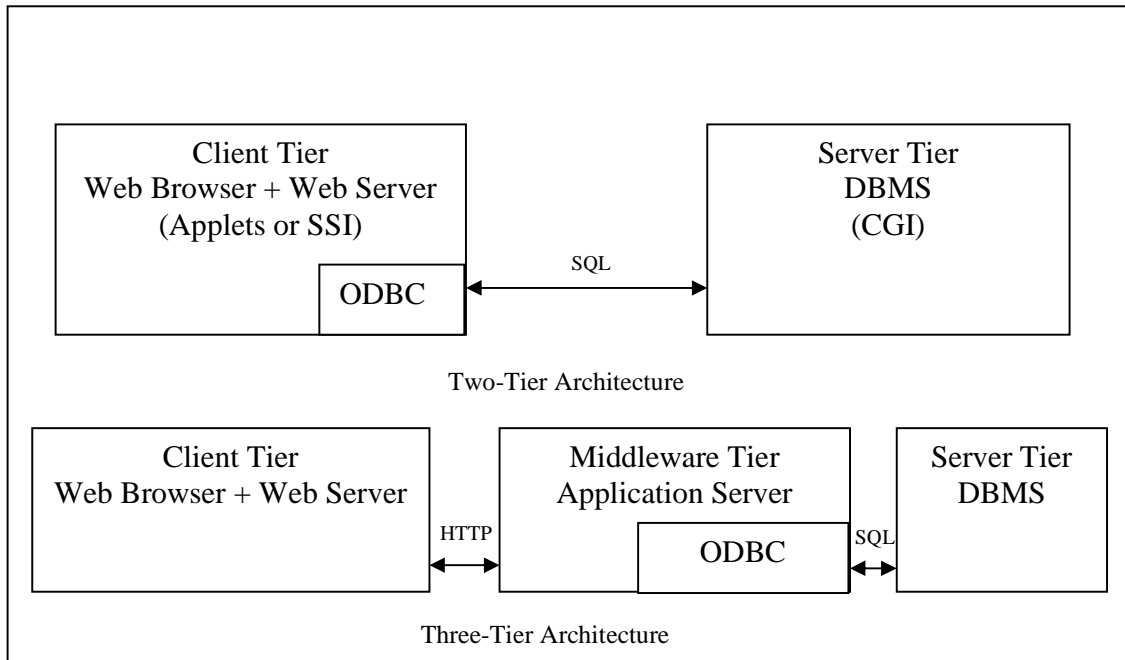
Figure 1: Two-Tier web database technologies vs. Three-tier Web database technologies