data. E-Commerce web sites, community sites, and online web services rely on SQL databases to manage user data or process user purchases.

SQL has become popular among web developers due to its flexibility and simplicity. With some basic knowledge of HTML, PHP, and a database program such as Microsoft's SQL Server, a developer becomes capable of creating complex websites and applications while relying on online web services to provide a SQL backend in which user data is stored.

SQL Features & Benefits

SQL is both an easy-to-understand language and a comprehensive tool for managing data. Here are some of the major features of SQL and the market forces that have made it successful:

Vendor Independence: SQL is offered by all of the leading DBMS vendors, and no new database product over the last decade has been highly successful without SQL support.

Portability Across Computer Systems: SQL-based database products run on computer systems ranging from mainframes and midrange systems to personal computers, workstations, a wide range of specialized server computers, and even handheld devices.

High-Level, English-Like Structure: SQL statements look like simple English sentences, making SQL easy to learn and understand. This is in part because SQL statements describe the data to be retrieved, rather than specifying how to find the data.

Interactive, Ad Hoc Queries: SQL is an interactive query language that gives users ad hoc access to stored data. Using SQL interactively, a user can get answers even to complex questions in minutes or seconds, in sharp contrast to the days or weeks it would take for a programmer to write a custom report program. Because of SQL’s ad hoc query power, data is more accessible and can be used to help an organization make better, more informed decisions.

Programmatic Database Access: SQL is also a database language used by programmers to write applications that access a database. The same SQL statements are used for both interactive and programmatic access, so the database access parts of a program can be tested first with interactive SQL and then embedded into the program.

Complete Database Language: SQL was first developed as an ad hoc query language, but its powers now go far beyond data retrieval. SQL provides a complete, consistent language for creating a database, managing its security, updating its contents, retrieving data, and sharing data among many concurrent users.

Client/Server Architecture: SQL is a natural vehicle for implementing applications using a distributed, client/ server architecture. In this role, SQL serves as the link between “front-end” computer systems optimized for user interaction and “back-end” systems specialized for database management, allowing each system to do what it does best.

Internet Database Access: With the exploding popularity of the Internet and the World Wide Web, and their standards-based foundation, SQL found a new role in the late 1990s as an Internet data access standard. Early in the development of the Web, developers needed a way to retrieve and present database information on web pages and used SQL as a common language for database gateways. In the future, the role of SQL in the Internet will extend beyond web site architectures to include data management for collaborating applications and distributed objects in a web services architecture.

SQL Connection with MS Access through java code:

import java.lang.\*;

import java.sql.\*;

import java.util.\*;

public class Test1

{

 public Test1()

 {

 int arr[];

 int q=0;

 try {

 Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

 String dataSourceName = "dsn";

 String dbURL = "jdbc:odbc:" + dataSourceName;

 Connection con = DriverManager.getConnection(dbURL);

 Statement s = con.createStatement();

 Statement s1 = con.createStatement();

 s.execute("select Rollno from Table1");

 ResultSet rs = s.getResultSet();

 if (rs != null)

 {

 while(rs.next())

 {

 q++;

 }

 }

 arr = new int[q];

 s1.execute("select Rollno from Table1");

 ResultSet rs1 = s1.getResultSet();

for(int i=0;i<q;i++)

 {

 rs1.next();

 arr[i]=Integer.parseInt(rs1.getString(1));

 System.out.println("Data from field Rollno = " + arr[i] );

}

 int m,n,o=0;

 for(int i=0;i<q;i++)

 {

 m=arr[i];

 n=0;

 for(int r=0;r<q;r++)

 {

 if(m==arr[r])

 n++;

 }

 if(n>1)

 {

 System.out.println("Rollno "+arr[i]+ " is repeated.");

 o++;

 break;

 }

 }

 if(o==0)

 System.out.println("No repeated rollno.");

s.close();

 con.close();

 }

 catch (Exception err)

 {

 System.out.println("ERROR: " + err);

 }

}

 public static void main(String[] args)

 {

 Test1 t = new Test1();

 }

}