Memorandum

To: Sarah Seay

From: Connie Murphy

Date: 2/20/2004

Re: Role of Object-Oriented Programming as it Relates to SQL-Compliant Databases

Introduction

There are many ways to access information in a database. SQL can be used as a command line language where each instruction is manually sent individually. With SQL you can also create tables, add data, delete data, splice data together, trigger actions based on changes to the database, and store your queries within your program or database. Another way to use the SQL command is to have a stored procedure that allows multiple SQL commands to be enacted in sequential order. Since allowing an application to have command line access through SQL to a database is not necessarily a safe or secure method of data management, there is another choice that provides stored procedures to be embedded in programming applications such as Java, Visual Basic or C++, which allows focus on the data processing, rather than the data procurement. Many front end applications provide simple, easy-to-use interfaces to the data.

The Microsoft Application Programming Interface (API) allows programmers to use SQL from inside Visual Basic, C, or C++. Source code can include embedded SQL statements, as long as the developer follows the rules for its inclusion:

1. placement of the SQL statements
2. location of comments to explain the code
3. rules for declaring variables

Visual Basic

With e-commerce as a focal point of most company policies today, there is a need for a programming language interface to the production database. The programming capabilities of Microsoft Visual Basic for Applications in Microsoft Access 2002 applications allow the developer to write custom functions and event-driven procedures. Easy GUI interfaces allow any user to obtain the needed information.

Java

The development of Java in internet applications enables a Java program to access a database using embedded Structured Query Language (SQL) statements. The role of SQL in Java programming is mainly used for object-relational mapping or integration. The combination of SQL and Java is extremely powerful and when XML is added to it, there is not only a manageable multiplatform interface, but now you have added document exchange as an option. Java allows developers to write applets and applications for the database. In late January 2002 Microsoft released a statement announcing that a Java driver would be built for SQL server 2000 at no additional cost, which made it that much more desirable as an interface.

C, C++

C and C++ can actually be used to build the database. There are two ways to access database servers from C++ development tools:

1. Use native database libraries that come with the tools.
2. Use third-party database libraries that integrate with your particular tool.

Borland C++, for instance, provides an add-on called Visual Database Tools (VDBT), which grants database access capabilities directly from Borland C++ applications. VDBT is built on top of the Borland Database Engine (BDE).

Microsoft's Visual C++ allows two ways to set up data access: via mechanisms in the MFC, and via the Component Gallery and wizard subsystems. The ODBC API can be used to access the data held in databases that provide ODBC drivers. Visual C++ supports data access objects (DAOs) through a new class in MFC. DAOs allow the manipulation of databases and record sets as well as to display and alter data inside the forms. As with Borland C++, DAO classes exist inside the OLE COM interface, which is linked directly to Microsoft's Jet database engine. Visual C++'s ClassWizard works with the database classes to link to the target database

Native database access subsystems are not the only way to access database servers from a C++ application. There are several third-party database access libraries available on the internet as well as retail.

C++ Development environments allow developers to extend the power of the C++ tool to meet the exact requirements of the application. This includes native GUI controls, database access, external messaging APIs, the Internet, and application-related classes. However, C++ development tools are only as good as the libraries they have available.

Complex distributed client/server applications that include distributed objects, TP monitors, and heterogeneous databases benefit most from C++ development. In these complex systems, the flexibility of C++ ties everything together, both at the back end and at the desktop. As a result, the need for C++ development tools is more important than ever.

References

http://www.microsoft.com/traincert/syllabi/2657AFinal.asp

Linthicum, David S. DBMS. (1996, August). Retrieved October 6, 2003 from: http://www.dbmsmag.com/9608d15.html

Morgan, B., Perkins, J., Plew, R., Stephens, R. (1998). Teach Yourself SQL in 21 Days, Second Edition. Retrieved October 6, 2003 from: http://members.tripod.com/er4ebus/sql/ch02.htm.

Price, Jason (2001). Setting Up Your Environment to Develop SQLJ Programs. Retrieved on October 6, 2003 from: http://www.onjava.com/pub/a/onjava/2001/12/05/learning_sqlj.html