Comparing Enterprise Development Productivity:

SQL Server 2005
versus
Oracle Database 10*g*

3 LEAF

3 Leaf, LLC
January, 2006

Table of Contents

1.0 Executive Summary

Microsoft SQL Server provides features such as failover clustering, replication, database snapshots, database mirroring, and snapshot isolation, helping ensure that your critical databases maintain high availability. The platform provides outstanding programmability, enabling developers to build robust custom solutions efficiently, relative to competing products.

A key proof point about the relative programmability of SQL Server 2005 versus Oracle Database 10*g* is how efficiently each platform allows developers to achieve the application functionality required to meet business needs. This productivity study compared two developers working independently to provide the same added features and functionality to a baseline application; one of the developers worked with SQL Server 2005, and the other used Oracle Database 10*g*. By comparing the time required by each developer to perform equivalent development tasks, the study quantified one instance of improved development efficiency in SQL Server 2005, relative to Oracle Database 10*g*.

Both developers in this productivity study worked in isolation from one another, each working from a baseline application developed in the environment under study (i.e., on the SQL Server platform for the SQL Server part of the study and on the Oracle platform for the Oracle part of the study). Those applications were as functionally identical as possible; each was a dynamic, database-driven web application that delivered news content, including premium content using various subscription levels, with a web-based front end and a database-driven back-end.

A web service enabled syndication of content to other web sites by users at a premium subscription level, and it also provided advertisers with the capability to associate specific ads with individual articles by keyword. The application also supported middle-tier caching, to decrease the load on the database server. The productivity study was divided into two phases, with each phase adding a new feature set to the baseline application:

- **Phase 1, Custom and Ad-Hoc Reporting,** provided reporting functionality to gather metrics associated with the site's revenue-generating activities.
- **Phase 2, Offline Smart Client Application for Ad Sales,** provided a tool to support sales while salespeople are online or offline.

The study was designed to keep the applications in alignment in terms of architecture, UI elements, and database structure throughout these two phases, in order to make the productivity study as fair a comparison as possible between the platforms under study. Communication between the developers was intentionally severely limited and moderated by a project manager to further isolate the effects of either development effort on the other.

The results of the study were that the SQL Server 2005 developer benefited dramatically from the programmability of the SQL Server 2005 platform. He reported substantial benefits from the maturity of the platform, which provided smooth implementations of core functionality and robust support for associated tasks. Rapid completion of the development tasks required by this study was particularly afforded by excellent support for ad-hoc and custom supporting, data synchronization, and replication. Those efficiencies caused the SQL Server 2005 development in the study to be completed far more quickly than the development effort based on Oracle Database 10*g*, as shown in the table below:

Phase	SQL Server 2005	Oracle Database 10*g*	Key SQL Server 2005 Feature
Phase 1: Custom and Ad-Hoc Reporting	16.5 hours	30 hours	Report Builder
Phase 2: Offline Smart Client Application	15 hours	96 hours	Data Synchronization & Replication
Total Development Time:	**31.5 hours**	**126 hours**	

The SQL Server 2005 developer was able to complete the tasks called for in both phases of the study in a total development time of 31.5 hours, or just under four developer-days. The Oracle Database 10*g* developer reported total development time of 126 hours, or just under 16 developer-days, approximately four times the amount of time required by the SQL Server 2005 developer.

2.0 Study Goals and Methodology

2.1 Overview

The goal of the study was to provide an objective measure of developer resources required to obtain common functionality using SQL Server 2005 versus Oracle Database 10*g*. Developers were chosen for each platform with similar backgrounds in database and web-application development on their chosen platform, and each of them worked in isolation from the other to add the same specific functionality to a common baseline application. Further detail about the developers in the study is contained in the appendix to this document.

In order to provide a means of comparison between the parallel development efforts, the study was divided into two phases, as described below. Each developer kept a running log of the hours they invested in the development of their solution, as well as development diaries to be used in the preparation of this study. Those logs and diaries were not visible to the other developer during the course of the project. Further, in order to maintain a level playing field and independence between the developers, only limited communication was allowed between them, as moderated by a project manager. Those discussions were limited to answering questions that were applicable to development in both environments. Developers were free, however, to conduct research using other resources such as web sites, newsgroups, etc., just as they would for a real-world project.

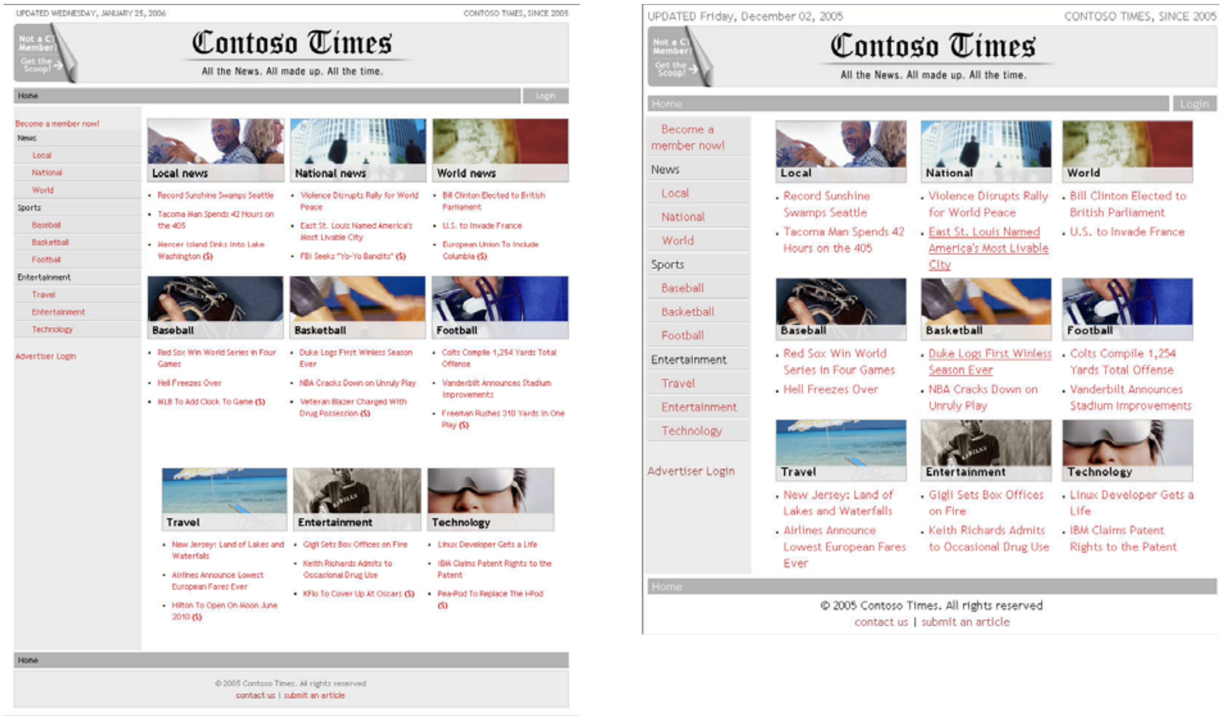
2.2 Development Firm: 3 Leaf

The development firm that carried out this productivity study, 3 Leaf, was chosen for its extensive experience working with both mature and emerging software technologies and products from Microsoft and others. Since 2000, 3 Leaf has specialized in creating reference, proof of concept, and sample applications, targeted training content, and demonstrations and keynote presentations for Microsoft, Intel and others, in addition to more traditional development efforts. Its developers have been early adopters of each generation of SQL Server, ASP, and ASP.NET, as well as competing technologies, helping the new aspects of these technologies to become adopted into the mainstream. This productivity study continues with that focus, working with SQL Server and the Oracle database simultaneously to root the development in solid, long-standing expertise with both platforms. 3 Leaf development can be contacted by visiting www.3leaf.com.

2.3 The Baseline Contoso Times Web Application

The SQL Server 2005 developer and the Oracle Database 10*g* developer each created versions of the baseline application from a common specification, using their own development environment. Those development environments are described in the appendix of this document. The developers each used their baseline application as the basis for their work in the rest of this productivity study. Thus, the SQL Server 2005 developer worked from a baseline application created in SQL Server 2005 and ASP.NET, and the Oracle Database 10*g* developer worked from a baseline application created in Oracle Database 10*g* and Java.

The Contoso Times application was a dynamic, data-driven web site that delivered newspaper-style article content to end users. A quality-control engineer and project manager worked with each developer in isolation from the other to ensure that the baseline applications, created using different platforms, were as functionally identical as possible. That same methodology was used throughout the productivity study, to ensure similar functionality of the applications at each phase of the study.



UPDATED WEDNESDAY, JANUARY 25, 2006
CONTOSO TIMES, SINCE 2005

Not a CT Member? Get the Scoop!

Contoso Times

All the News. All made up. All the time.

Home | Login

Become a member now!
News
- Local
- National
- World

Sports
- Baseball
- Basketball
- Football

Entertainment
- Travel
- Entertainment
- Technology

Advertiser Login

Local news
- Record Sunshine Swamps Seattle
- Tacoma Man Spends 42 Hours on the 405
- Mercer Island Sinks Into Lake Washington ($)

National news
- Violence Disrupts Rally for World Peace
- East St. Louis Named America's Most Livable City
- FBI Seeks "Yo-Yo Bandits" ($)

World news
- Bill Clinton Elected to British Parliament
- U.S. to Invade France
- European Union To Include Columbia ($)

Baseball
- Red Sox Win World Series in Four Games
- Hell Freezes Over
- MLB To Add Clock To Game ($)

Basketball
- Duke Logs First Winless Season Ever
- NBA Cracks Down on Unruly Play
- Veteran Blazer Charged With Drug Possession ($)

Football
- Colts Compile 1,254 Yards Total Offense
- Vanderbilt Announces Stadium Improvements
- Freeman Rushes 310 Yards In One Play ($)

Travel
- New Jersey: Land of Lakes and Waterfalls
- Airlines Announce Lowest European Fares Ever
- Hilton To Open On Moon June 2010 ($)

Entertainment
- Gigli Sets Box Offices on Fire
- Keith Richards Admits to Occasional Drug Use
- KFlo To Cover Up At Oscars ($)

Technology
- Linux Developer Gets a Life
- IBM Claims Patent Rights to the Patent
- Pea-Pod To Replace The I-Pod ($)

Home

© 2005 Contoso Times. All rights reserved
contact us | submit an article

UPDATED Friday, December 02, 2005
CONTOSO TIMES, SINCE 2005

Not a CT Member? Get the Scoop!

Contoso Times

All the News. All made up. All the time.

Home | Login

Become a member now!
News
- Local
- National
- World

Sports
- Baseball
- Basketball
- Football

Entertainment
- Travel
- Entertainment
- Technology

Advertiser Login

Local
- Record Sunshine Swamps Seattle
- Tacoma Man Spends 42 Hours on the 405

National
- Violence Disrupts Rally for World Peace
- East St. Louis Named America's Most Livable City

World
- Bill Clinton Elected to British Parliament
- U.S. to Invade France

Baseball
- Red Sox Win World Series in Four Games
- Hell Freezes Over

Basketball
- Duke Logs First Winless Season Ever
- NBA Cracks Down on Unruly Play

Football
- Colts Compile 1,254 Yards Total Offense
- Vanderbilt Announces Stadium Improvements

Travel
- New Jersey: Land of Lakes and Waterfalls
- Airlines Announce Lowest European Fares Ever

Entertainment
- Gigli Sets Box Offices on Fire
- Keith Richards Admits to Occasional Drug Use

Technology
- Linux Developer Gets a Life
- IBM Claims Patent Rights to the Patent

Home

© 2005 Contoso Times. All rights reserved
contact us | submit an article

Figure 1. The baseline applications in SQL Server 2005 (left) and Oracle Database 10*g* (right) were regulated by a quality-control engineer and project manager to be as functionally identical as possible.

2.3.1 Baseline Application Front-End Functionality

The web site front-end was based on an ASP.NET or Java web application. News content was divided into sections that corresponded roughly to traditional newspaper sections, with each article categorized as one of two types:

- **Public content available to all users of the site.** This content is available free of charge, without requiring users to authenticate.
- **Premium content available only to registered, authenticated users for a fee.** A free portion of each premium article (approximately three lines long) is viewable without charge. The pay-for-content model supports three payment modalities:
    - **On Demand:** This option allows users to users to pay to view or download content that has been designated with a 'premium' flag on a per-article basis.
    - **Subscription:** This option allows users to purchase the rights to view and download all content on the site.
    - **Premium Subscription:** This option allows users to purchase rights not only to view and download all content on the site, but also to syndicate it onto their own web sites or other publications, using a web service built for that purpose.

The baseline application also provides the ability for advertisers to match their ads with related articles. On a regular basis, the entire editorial contents of the site is indexed and matched to advertisements, so that specific advertisements can be displayed alongside the articles that provide the best match in terms of subject matter.

This targeted advertising mechanism allows the web site to provide various rate structures to advertisers, with an eye toward increasing advertising revenue. For example, advertisers could pay more to have their ads appear more often, or they could cut advertising costs by having their ads appear less often but only alongside the most relevant stories. This flexibility adds business value for both the advertisers and the site owners.

2.3.2 Baseline Application Back-End Infrastructure

A SQL Server 2005 or Oracle Database 10*g* database consisting of a single user-defined table stored all site content. For each article record, it contained an **ItemID** to uniquely identify the article, a **CategoryID** to place the article in one of the nine content sections, and the text of the title, subtitle, and body of the article. It also included various stored procedures to retrieve, delete, and update article content.

The Contoso Times used forms authentication to identify users. Users who wished to log in could do so through a login form. User names, passwords, and other information regarding registered users was stored securely in a back-end membership database. Once a user had logged in, the Web site used role-based security to differentiate between types of users. Ordinary users could access the site's public and premium content, but they could not access the site's administrative pages. Users who were administrators enjoyed access to the administrative portion of the site, where articles could be viewed, edited, and deleted.

An administration area enabled articles to be designated as premium content, as well as to create the free teaser portion of paid articles. The capability to flag articles was available only to selected user roles within the Contoso Times organization. A form was also provided that captured and encrypted user data (using a wrapper class and the Advanced Encryption Standard) such as a username/password pair, contact information, and credit card information. For the subscription model, stored user credentials allowed or denied access to premium content.

To facilitate the premium subscription model, the application exposed article content using a web service interface that could be called by the subscriber in order to publish the content on their site. The developers were both able to generate the web service using built-in functionality in Visual Studio and the JDeveloper IDE, respectively. The web service also automatically added appropriate information to each article to attribute the content to Contoso Times, as well as providing a link back to the Contoso Times site. Finally, it incorporated authentication mechanisms so that only authorized users could access it.

Separate web-based interfaces allowed both new articles and new advertisements to be associated with the system. The advertisement interface supplied rates for having ads appear at various frequencies, collected the advertising content itself, and allowed advertisers to enter up to five keywords. When new advertisements were entered into the system, a database trigger launched asynchronous indexing of every article in the system to match the advertisement keywords with target articles in a database mapping table.

A web interface collected advertiser billing and contact information, which was stored in an encrypted state. Reporting capability enabled advertisers to generate various reports that show click-through statistics on each of their ads, including demographic information about the users who had clicked on their ads.

The application also implemented middle-tier caching to improve performance and scalability of the overall site. By reducing the number of queries against the content database, this implementation reduced the average load on the database server per page-view, resulting in decreased latency from the user's point of view. Because middle-tier caching allowed the site owners to support a larger number of users with the same amount of server infrastructure, this strategy enabled them to obtain better return on their hardware investment.

The middle-tier caching functionality generated notifications when either articles or advertisements changed on the site. In the event that an article or advertisement had been added, changed, or deleted, the site cache logic needed to query the database to obtain current information; otherwise, it delivered the appropriate cached data. The SQL Server/ASP.NET solution used the **SqlCacheDependency** namespace to provide caching functionality, because of its simplicity and reliability. The Oracle Database/Java solution used the Oracle Application Development Framework (rather than custom in-memory caching using the **CachedHashtable()** class or EJB Entity Beans), also because of simplicity and reliability, as well as good support from the JDeveloper IDE for this modality.

2.4 High-Level Overview of Development Phases in the Study

This study consisted of the phased rollout of new functionality into the baseline application, much as new features are often added to in-house or commercial applications. In order to provide specific milestones for comparison between development in SQL Server 2005 and Oracle Database 10*g*, the study was divided into two distinct phases, summarized below.

2.4.1 Phase 1 Functionality: Custom and Ad-Hoc Reporting

In order to gauge the performance of the site, as well as the success of the previous changes to the environment, the Contoso Times management needed improved ability to gather metrics associated with the site's revenue-generating activities. The first phase of this study addressed these requirements by providing robust reporting mechanisms related to the site. To support this phase, the developers were required to implement a custom reporting solution based on a web-based interface and an ad-hoc mechanism that allowed employees to create reports on an as-needed basis.

2.4.2 Phase 2 Functionality: Offline Smart Client Application for Ad Sales

In the second phase of this productivity study, the developers were called upon to improve the efficiency of the advertising-sales process. Their goal was to achieve that improvement by providing a smart client application that works both online and offline, for use by both tele-sales representatives and field sales personnel. The application must enable sales representatives to generate reports on existing advertisements, as well as to take orders for new advertising sales.

3.0 Phase 1 Development: Custom and Ad-Hoc Reporting

This phase called for the addition of reporting mechanisms to gather metrics related to the success of various aspects of the site's operation, such as subscription information, advertising impact, and paid article consumption. It also allows members of the Contoso Times staff to generate ad hoc reports on subscription and advertising data. Such specialized reports can be valuable, for example, in support of sales activities. By empowering staff members to create reports based on their individual needs, when they need them, those staff members are able to perform their jobs more efficiently, and the costs associated with designing custom queries for one-time reports are avoided.

Ad hoc reports are also available to advertisers, providing post-sales support at very low cost to the Contoso Times organization by enabling advertisers to independently quantify the success of their advertising investment.

3.1 Technology Requirements Underlying Phase 1 Development

Custom Reports are available only to members of the Contoso Times staff, while ad hoc reports are available to advertisers as well. Users can save ad hoc reports for future use.

The site implements custom reports using a web-based interface that is available to all members of the Contoso Times organization, although certain reports are available only to certain members of the company. The reports themselves are static and printable, incorporating no 'drill-down' or dynamic sorting functionality. Reporting users are based on Contoso Times user accounts with temporary assignment of higher access privileges for the purpose of running the report.

3.2 Summary of Phase 1 SQL Server 2005 Development

3.2.1 Phase 1 SQL Server 2005 Development Time Requirements

The SQL Server 2005 developer required a total of 16.5 hours to implement the functionality called for in phase 1 of the study. He reported the breakdown of those hours as follows:

Task	Time
Research and preparation	4.25 hours
Build reports	10.25 hours
Miscellaneous	2 hours
Total SQL Server 2005 Development, Phase 1:	**16.5 hours**

3.2.2 Phase 1 SQL Server 2005 Platform-Impact Statement

The SQL Server 2005 developer reported that he found the built-in Report Viewer control and the new Report Builder in Reporting Services quite helpful in creating local reports and ad hoc reports, respectively. Report Builder is a ClickOnce Windows Forms application launched from the Report Server that enabled the SQL Server 2005 developer to build the required reports very quickly.

In a real-world deployment, moreover, SQL Server 2005's new Report Builder would provide a user-friendly business-query model that would enable general users to generate additional ad hoc reports. That capability translates into ongoing, recurring business advantages, since users are not dependent upon an IT department or other technical assistance in order to generate reports. That capability would potentially generate productivity increases and potential support-cost savings, beyond the scope of what is reported in this study.

3.3 Summary of Phase 1 Oracle Database 10*g* Development

3.3.1 Phase 1 Oracle Database 10*g* Development Time Requirements

The Oracle Database 10*g* developer required a total of 30 hours to implement the functionality called for in phase 1 of the study. He reported the breakdown of those hours as follows:

Task	Time
Research and preparation	8 hours
Build reports	14 hours
Miscellaneous	8 hours
Total Oracle Database 10*g* Development, Phase 1:	**30 hours**

This outcome represents approximately 1.8 times the development time required by the SQL 2005 developer for this phase of the study.

3.3.2 Phase 1 Oracle Database 10*g* Platform-Impact Statement

The Oracle Database 10*g* developer reported that the Report Wizard and Reports Builder tool provided by the platform provide a good interface and are fairly simple to use. The tools provide the ability to deploy reports easily as JSP pages, as well as in a printable format. The efficiency with which he was able to perform this development, however, was somewhat less than that experienced by the SQL Server 2005 developer for equivalent tasks.

The templates and query builder provided with Oracle Reports Builder would enable a technical user to create reports against simple datasets, according to the Oracle Database 10*g* developer. This functionality targets a smaller user base than the general users supported by the equivalent user base in SQL Server 2005, however. That characteristic could be expected to provide a lower level of functionality to general users over the long term, relative to the SQL Server 2005 platform, corresponding to a greater need for IT staff to provide reporting support to the user base in a real-world deployment.

3.4 Phase 1 Competitive Outcome

The outcome of this phase was that the SQL Server 2005 developer was able to implement the solution in slightly more than two developer-days, whereas it took the Oracle Database 10*g* developer just under four days.

The difference in development time for this phase was generated primarily by the efficiency of the reporting tools in SQL Server 2005, which enabled the developer to perform similar tasks to the Oracle 10*g* developer in a smaller amount of time. This ease of use may be related to the relative maturity of SQL Server 2005 as a programmable platform, as compared to Oracle Database 10*g*. The SQL Server 2005 developer reported significant advantages of this current version, relative to previous versions of SQL Server. The Oracle Database 10*g* developer, on the other hand, reported that many of the Oracle tools had not yet reached full maturity, which hampered his programming efforts, within the scope of this study.

4.0 Phase 2 Development: Offline Smart Client Application for Ad Sales

This phase of the site development targeted improving the efficiency with which sales personnel can sell advertising by providing a smart client application that works both online and offline. The application incorporates two general spheres of functionality:

- **Reporting, Charting, Graphs, and Research:** This aspect of the application primarily targets pre-sales support by enabling sales representatives to collect and report on advertising and viewing data.
- **Order Fulfillment:** This portion of the application lets sales staff actually take orders and enter the relevant information to close the sale. It supports both the advertising information itself, as well as the contact and payment information of the purchaser.

4.1 Technology Requirements Underlying Phase 2 Development

The reporting, charting, graphs, and research portion of the application takes advantage of the reporting functionality already implemented in Phase 1 of the study, but it also incorporates the local caching and synchronization functionality to allow this reporting to take place while the user is offline. The SQL Server solution uses reporting functionality built into Visual Studio 2005 to create ad hoc reports from a local data set, which is stored using SQL Server Express. The Oracle solution uses reporting tools built into the Oracle Developer Suite to create reports from a local data set saved using Personal Oracle Lite.

The order-fulfillment part of the application provides a smart-client interface that encrypts personal data including contact and credit card information using CLR classes in the case of the SQL Server application and using Oracle Transparent Data Encryption (TDE) in the Oracle application. Merge replication uploads the data to the central server, using partitioned articles to limit the data returned to the representative.

Both applications use a simplified deployment model to enable sales representatives to easily receive updated versions of the application via web download from a virtual directory on the Contoso Times web site. This flexibility enables sales representatives to move about freely and perform their duties in an efficient manner with or without connectivity to the network, while also letting site administrators govern their access to data. The SQL Server application uses ClickOnce technology as the basis of this deployment model, while the Oracle application downloads a small, platform-specific updater built into Oracle Mobile Server that is responsible for synchronizing new versions of the application.

4.2 Summary of Phase 2 SQL Server 2005 Development

4.2.1 Phase 2 SQL Server 2005 Development Time Requirements

The SQL Server 2005 developer required a total of 15 hours to implement the functionality called for in phase 2 of the study. He reported the breakdown of those hours as follows:

Task	Time
Create interface and advertiser/advertisement data management	8 hours
Implement database access and replication	2 hours
Implement reporting	4.5 hours
Miscellaneous	.5 hours
Total SQL Server 2005 Development, Phase 2:	**15 hours**

4.2.2 Phase 2 SQL Server 2005 Platform-Impact Statement

During the development of the smart client application, the SQL Server 2005 developer reported substantial efficiency benefits in setting up replication. He set up both the publication and subscription engines to ensure that they would replicate the data using built-in wizards. He used the 'transactional publication with updatable subscription' functionality. This type of replication enables changes from the publisher to stream to the subscriber after the initial snapshot, and changes from the subscriber are applied at the publisher.

At one point, he had to make changes to the subscription, which required him to drop the subscription and re-create it. He was able to simply step through a wizard and change the update process from occurring on a 'simultaneously commit changes' basis to occurring on a 'queue changes and commit when possible' basis. The SQL Server 2005 developer stated that this made implementing the offline/online capability of the application very easy.

4.3 Summary of Phase 2 Oracle Database 10*g* Development

4.3.1 Phase 2 Oracle Database 10*g* Development Time Requirements

The Oracle Database 10*g* developer required a total of 96 hours to implement the functionality called for in phase 2 of the study. He reported the breakdown of those hours as follows:

Task	Time
Create interface and advertiser/advertisement data management	64 hours
Implement database access and replication	20 hours
Implement reporting	12 hours
Miscellaneous	0 hours
Total Oracle Database 10*g* Development, Phase 2:	**96 hours**

This outcome represents approximately 6.4 times the development time required by the SQL 2005 developer for this phase of the study.

4.3.2 Phase 2 Oracle Database 10*g* Platform-Impact Statement

The Oracle 10*g* developer reported that Swing support behaved smoothly, and support for AWT (Alternative Window Toolkit) functionality provided for straightforward GUI development. On the other hand, implementation of that user interface and related data-management functionality proved to be far more time consuming than the corresponding efforts by the SQL Server 2005 developer. Substantial issues were reported by the Oracle Database10*g* developer in writing client code to access the local client-side database, as well as in synchronizing data between the server and mobile client.

Data and file synchronization is automated by Oracle Database Workbench, but the developer encountered numerous error messages in implementing this functionality. For example, after defining the mobile project, Mobile Database Workbench would not allow him to reopen the project. Mobile Database Workbench also threw numerous NullPointerException errors, and the packaging wizard failed repeatedly. The developer reported that the documentation associated with these errors was sparse, which contributed to making the resolutions of the issues take longer than might be expected.

4.4 Phase 2 Competitive Outcome

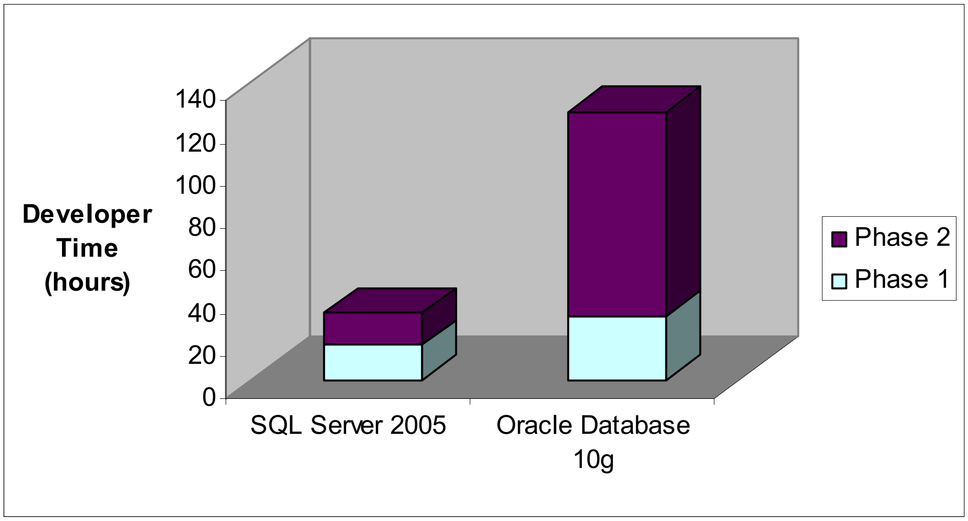
The outcome of this phase was that the SQL Server 2005 developer was able to implement the solution in just under two developer-days, whereas it took the Oracle Database 10*g* developer the equivalent of 12 days.

The most significant contributor to the more rapid development in this phase by the SQL Server 2005 developer was the programmability of the SQL Server 2005 platform in terms of creating the user interface and associated data-management functionality. The 8 hours required by the SQL Server 2005 developer, as compared to 64 hours by the Oracle Database 10*g* developer, was far and away the largest contributor to the favorable competitive outcome by SQL Server 2005 in this phase of the productivity study.

The smooth data and file synchronization enjoyed by the SQL Server 2005 enabled him to finish the development tasks associated with creating the mobile smart client application quickly and efficiently. The mature toolset and associated functionality afforded by the SQL Server 2005 platform enhanced overall programmability and delivered favorable results.

5.0 Overall Outcome

For the total development time of both phases of the study, the SQL Server 2005 developer reported 31.5 hours, or just under four developer-days. The Oracle Database 10*g* developer reported total development time of 126 hours, or 15.75 developer-days, four times the amount of time required by the SQL Server 2005 developer. The comparison of the hours required by each developer is summarized in the chart below:



This difference in the resources required by the two developers to create the solutions under SQL Server 2005 and Oracle Database 10*g* reflects the excellent value and programmability of the SQL Server 2005 platform.

6.0 Key SQL Server 2005 Usability Benefits Reported by Developers

The developers involved in this study reported significant advantages to the SQL Server 2005 environment in day-to-day usability, relative to the Oracle Database 10*g* environment. The study found that the combination of SQL Server 2005 and Visual Studio 2005 provides significant efficiency advantages over the Oracle database and toolset being tested[1]. Following are a few of the key usability differences between the two platforms, as they pertain to this study.

6.1 Reporting Advantages

As mentioned in the Phase 1 discussion above, the SQL Server 2005 developer benefited from the built-in Report Viewer control and the new Report Builder in Reporting Services to create local reports and ad hoc reports, respectively. The Oracle Database 10*g* developer obtained similar functionality from the Reports Builder tool in the Oracle Developer Suite, which supports both pre-defined and ad-hoc reports, but it required him approximately 137% as much time to create the reports, compared to the SQL Server 2005 developer. Moreover, the focus of SQL Server 2005 on enabling ad-hoc reporting by general, non-technical users could be expected to deliver ongoing advantages, relative to the focus of the Oracle platform on providing that functionality to technical users.

6.2 Authentication Advantages

By leveraging the **aspnetdb** database and the login controls, the SQL Server 2005 developer easily created and maintained users, as well as limiting their access. He reported that the automation associated with this functionality was robust and simple to use. The Oracle Database 10*g* developer identified multiple unexpected behaviors and debugging issues associated with logging out and back in again, particularly with regard to Struts Forms and Beans in Phase 1 of the study.

6.3 Replication Advantages

As mentioned in the Phase 2 discussion above, the SQL Server 2005 developer benefited substantially in terms of efficiency when he set up replication. He used built-in wizards to set up both the publication and subscription engines to ensure that they would replicate the data. He indicated that this automated functionality substantially decreased the time required to implement the offline/online capability of the application. The Oracle Database 10*g* developer, on the other hand, had substantial difficulty with the replication functionality associated with the Oracle tool stack. He encountered numerous sparsely documented errors that slowed development, particularly in Phase 2 of this study, which was a major contributor to the slower Oracle Database 10*g* development in the study overall*,* relative to SQL Server 2005.

6.4 Interface and File-Management Advantages

The SQL Server 2005 developer responded especially favorably to the addition of SQL Server Management Studio in the new version if this platform. In particular, he benefited from being able to test queries and resolve debug issues without switching back and forth between multiple environments. Visual Studio 2005's inclusion of a local web server also provided significant advantages. This functionality simplified development by removing the need to configure Internet Information Services. Moreover, it enabled him to put web applications in any directory he wanted, moving them at will with a simple cut and paste operation. The Oracle Database 10*g* developer, on the other hand, reported that the debugging functionality of the tool set he employed added substantial overhead to the development effort.

7.0 Summary and Conclusion

The findings of this study demonstrate the real efficiency benefits that are available to developers in their day-to-day activities using SQL Server 2005, relative to Oracle Database 10*g*. The comparative 31.5 hours required by the SQL Server 2005 developer to perform the same tasks that took 126 hours for the Oracle Database 10*g* developer are emblematic of the advanced value and functionality afforded by SQL Server 2005.

The maturity of the SQL Server 2005 platform, as well as excellent support for ad-hoc and custom reporting, data synchronization, and replication, were instrumental to the efficiency gains realized by the

[1] For details about the tools and technologies used by each developer, see the appendix of this document.

SQL Server 2005 developer in this study. Increased programmability, efficiency, and value are available for business-critical implementation by enterprise developers today.

Learn more about the benefits of SQL Server 2005 at http://www.microsoft.com/sql/default.mspx.

Appendix: The Development Environments

The developers were chosen specifically for their extensive and similar levels of expertise in relation to the tasks required in this productivity study. Both have seven to eight years of experience programming with various generations of the database platforms they worked with in this study. Likewise, both have worked for six to seven years in web development, including ASP and ASP.NET by the SQL Server developer and JSP by the Oracle developer. Overall, they have equivalent skill sets in terms of the core technologies and tools they used in the study.

Both developers used similar environments, consisting of tools provided by Microsoft and Oracle, respectively. Thus, for instance, the Oracle developer did not make use of the full scope of tools from the Oracle/Java community at large; instead, to the highest degree possible, he used just the core Oracle toolset.

The development process focused as much as possible on the use of tools provided by the database environments themselves, as opposed to external tools such as those provided by the IDEs. For example, in Phase 2, the developers were compelled to use merge replication with SQL Server Express and Personal Oracle Lite, respectively, as opposed to building out custom synchronization mechanisms for the smart client application.

The following table compares key tools and technologies used in the study:

Functionality	SQL Server Environment	Oracle Environment
Database	SQL Server 2005 CTP June (moving to CTP Sept)	Oracle Database 10*g*
Integrated Development Environment	Visual Studio 2005 Beta 2 (moving to RC1 as the study progressed)	JDeveloper 10*g*, augmented with BEA DBVisualizer (a SQL client tool)
Application server	Windows Server 2003	Oracle Application Server 10*g*
Web technology	ASP.NET	Java Server Pages Struts
Report generation	Report Viewer (local reports) Report Builder (ad-hoc reports)	Report Builder
Local database for smart client	SQL Server Express	Personal Oracle Lite
Smart client application deployment/synchronization	ClickOnce technology	Oracle Mobile Server
Encryption mechanism	Built-in encryption functionality	Oracle Transparent Data Encryption

This productivity study was conducted by a 3 Leaf, a firm unaffiliated with Microsoft. Since 2000, 3 Leaf has specialized in creating reference, proof of concept, and sample applications, targeted training content, and demonstrations and keynote presentations for Microsoft, Intel and others, in addition to more traditional development efforts. Its developers are early adopters of a wide range of technologies.

3 LEAF

For more information about 3 Leaf, please visit www.3leaf.com.

To contact 3 Leaf directly, please email info@3leaf.com.