Database Application Development Tools for Oracle8™

An Oracle Technical White Paper

June 1997

INTRODUCTION

The introduction of the Oracle8 Universal Data Server promises to redefine the nature of application development into the twenty first century. Significant developments in scalablity/throughput, and the introduction of object concepts to mainstream open databases through the evolution of relational technology, have resulted in an unbeatable environment for developers to produce powerful and performant applications.

Applications to support many thousands of concurrent users have traditionally been the domain of transaction processing systems, proprietary data stores and large amounts of custom-built third generation code. While extremely successful in achieving the goal of supporting large numbers of users and high performance, the low level nature of these applications makes them difficult to maintain or change and often proprietary in nature. Conversely, open systems databases and fourth generation tools have often been unsuitable for deployment at this level as, though easy to use, they did not scale to the required user communities.

This paper looks at the some of the features of Oracle8 and the Universal Data Server and how Designer/2000™ and Developer/2000™ support true enterprise-wide applications while retaining the high productivity that comes from a visual 4GL development environment.

DESIGNER/2000 AND ORACLE8

Designer/2000 is a business and application modeling tool with the unique ability to generate complete applications from those models. Business analysts and developers use a visual modeling interface to represent and define business objects, functionality, business rules and requirements in a simple, declarative way. Designer/2000 defines systems independently of implementation, so applications can be generated in multiple environments and configurations from a single model. Developers can reuse application definitions by simply dragging and dropping them into new models.

Client and Server Modeling and Generation

From its models, Designer/2000 generates and reverse engineers application components and server components.

The application components (client and application server) are delivered through:

- Developer/2000 client-server applications
- Developer/2000 Web applications
- Oracle Web Application Server™ applications
- and others

For the server Designer/2000 also models, generates and reverse engineers database schemas, Oracle8, Oracle7™, Oracle Rdb™ and others. In Designer/2000 2.0 support for Oracle8 includes all the Oracle8 scalability features such as:

- partitioned tables
- BLOBs (Binary Large Objects), CLOBs (Character Large Objects) etc.
- index organized tables

- deferred constraint checks
- and the Oracle8 object features including:
- user defined types
- type tables
- references
- embedded types
- collections
- object views.

These concepts are represented using an extension of the Unified Modeling Language (UML), the Object Management Group's (OMG) emerging open standard for object modeling.

Data Server Generation and Design Recovery

Database design andvisualization tools have proved to be extremely helpful to relational database designers and will prove equally so to designers of object-relational databases. Previous versions of Designer/2000 have successfully delivered this capability for Oracle7 designers. Now Designer/2000's database schema modeler has been extended to represent and manage Oracle8's scalability and object features without compromising its established ease of use.

Once a database schema design is complete, Designer/2000s server generator capability automatically translates that design from a graphical representation into the appropriate SQL DDL to implement it. This approach takes the effort out of manually building a database, and guarantees bug-free SQL.

Because it is equally important to be able to visualize existing database structures. Designer/2000 loads existing database schema definitions into its repository and automatically creates visual models of them. It does this not only for Oracle 8 schemas but also for Oracle 7, RDB and third party databases using native drivers or ODBC. Such recovered relational designs are manipulated within the design tool and then generated as object-relational schemas in Oracle8. The ability to recover designs from existing relational databases adds considerable impetus to re-implementation projects seeking to exploit to Oracle8.

Designer/2000 uses UML to represent its type models. However, in its base form, UML does not represent the specific implementation mechanisms that Oracle8 offers. Oracle 8 provides a rich set of options to implement a database design, including major new database concepts such as:

TYPES	- user defined datatypes beyond the simple scalars.
OBJECT TABLES	- tables that contain instances of complex datatypes.
VARRAYS	- multi-valued columns.
NESTED TABLES	- tables embedded within other tables.
OBJECT VIEWS	- represent a relational structure as though it were a type
REFS	- the ability to directly reference one object from another

as well as all the standard relational features provided in Oracle 7.

The availability of these implementation choices means that the database designer can use their skill to refine the type model to exploit the power and flexibility built into Oracle8.

The designer may decide to use foreign keys and object views for some or all of the schema implementation, or to use only object features. The designer must decide which collection mechanisms will be used to implement aggregation and composition associations. For example, the association between a *Customer* and their *Accounts* would probably be modeled as composition and might be implemented as a VARRAY of REFs since there will be a fairly low upper limit in the number of accounts a *Customer* will hold. The association from *Bank* to its *Branches* might also be formally modeled as composition, but there are likely to be a large number of *Branches* for one *Bank*, so it might be implemented as a nested table of REFs.

Designer/2000's facilities for designers to map base UML models onto explicit Oracle8 features means that optimal database implementations can be achieved without compromising real-world object models.

Application Modeling and Generation

Over a number of generations of the product, Designer/2000 has built up its capability to generate more and more powerful applications directly from its visual modeling tools. In parallel with the growth in generator technology, the modeling tools themselves have matured, with the result that using Release 2 of Designer/2000 developers can define **all** of the logic and structure of their designs and have the full application created automatically.

Developer/2000 has always been the primary application generation target for Designer/2000 and by using the generator's capacity to exploit Developer/2000, developers have access to all the powerful run-time facilities of Developer/2000.

The next section discusses how Developer/2000 exploits Oracle8 allowing users of Designer/2000 and Developer/2000 to define their needs at the business level and have them transformed into effective and efficient object relational implementations.

DEVELOPER/2000 AND ORACLE8

Scalability

Developer/2000 has, from inception, been designed as a tool to allow for large numbers of concurrent users in an OLTP environment. Through tight integration with the Oracle RDBMS it is able to take advantage of many performance enhancements within the kernel in a seamless manner.

Many of the bottle-necks faced by applications result from the time/resources required to access the data itself and the efficiencies of the client-server environment, both in the need to return data across the network and the method in which it is done. The scalability of an application is dependent on a number of factors.

Scalability can be defined here as;

- the size of the database and the effect that large amounts of data have on query time
- the number of concurrent users accessing the database
- transaction throughput

Oracle8 introduces a number of new technologies which are aimed at both the OLTP and VLDB environments many of which Developer/2000 is able to utilize. These features and Developer/2000's support for them are described below.

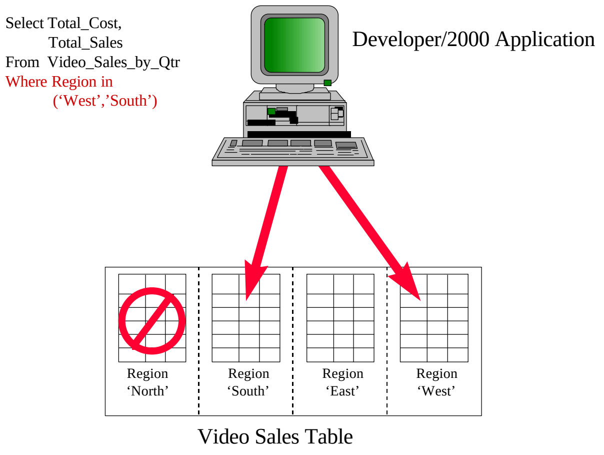
VLDB

Application scalability, in terms of database size, requires that as the database grows (Oracle8 supports up to 512 Petabytes of data) the performance of the application should also scale with the increased load.

Partitioned Tables :

As a given table grows in size the time taken to scan the records increases. By partitioning the table into manageable pieces that contain specific ranges of data a Developer/2000 application may:

- have increased access to table data as a partition which is unavailable (eg due to a disk crash) will not effect queries or DML on other partition in the table.
- access only that part of the table which contains data of interest (automatically skips unneeded partitions based on query criteria).



```
Select Total_Cost,
       Total_Sales
From  Video_Sales_by_Qtr
Where Region in
         ('West','South')
```

Figure1: Partitioned Tables

Parallel Query/DML

Oracle7 introduced the ability for a Developer/2000 application to process queries in parallel (across the available CPUs). The degree of parallelism is defined the using *hints* in the SQL issued by the application.

Example

```
SELECT /*+ parallel(VIDEO_SALES_BY_QTR,4) */  region, category, cost
FROM  VIDEO_SALES_BY_QTR
WHERE        Region = 'West'
```

The previous statement would query the VIDEO_SALES_BY_QTR table utilizing four CPUs.

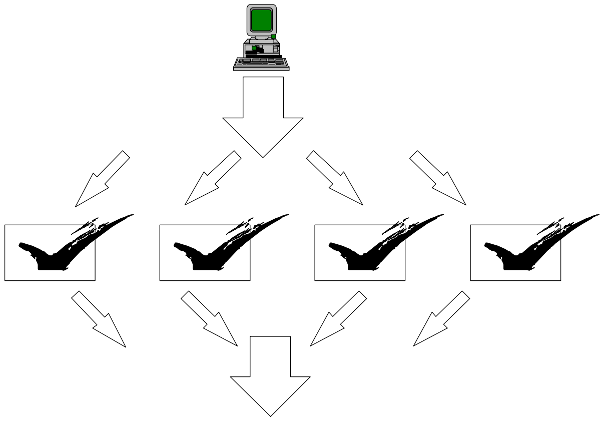


Figure2: Parallel Query/DML

With the introduction of Oracle8, Developer/2000 applications may now implement the same parallel processing techniques on transactional operations as well as queries. Applications which require batch update/insert or delete of large numbers of records will be able to process them in parallel rather than sequentially. This further complements developers ability to pass several DML transactions to the server as array sets (see below).

OLTP

As the number of concurrent users on the system increases, the connection load on the server increase as does the amount of data carried across the network (with its limited band width). In order to allow for more users, an application tool must minimize the number of round trips required to the server as well as optimize the transfer of data when it is required. With this in mind, Oracle8 introduces several functions which, along with Developer/2000, allow applications to decrease the number of calls to the server as well as take advantage of Oracle8's advanced queuing facilities and X/A compliance.

Same language on client & server

One of the greatest benefits Developer/2000 brings to database application development is support for the easy partitioning of procedural logic between the client and server tiers. The use of PL/SQL in both the client and server allows for procedures to be located where they are going to be most performant, Interface logic in the client, data logic in the server.

Advanced queuing of transactions

As shown by transaction processing systems, one of best methods of increasing the effective number of transactions is to buffer the transaction and to defer its execution. In this manner the end user is not dependent on the time it takes for an individual transaction to complete before moving to the next.

Applications developed with Developer/2000 are able to implement a similar transactional model by use of Oracle8’s Advanced Queuing support. Commands are queued through the use of PL/SQL stored procedures which may be called via the standard trigger mechanism, for example transactional triggers. As the development language of the client is also PL/SQL the calling of these procedures is transparent to the application developer.

Example: Transactional trigger ON-INSERT contains the following

```
DBMS_AQ.ENQUEUE(q_name=>’Application Queue Name’,
                priority=> nn (priority level of transaction),
                user_data=> (transactional data),
                new_msgid=>(message sequence number).
```

A database server side process (or application server) will then be able to process the transaction in the queue via use of the DBMS_AQ.DEQUEUE procedure.

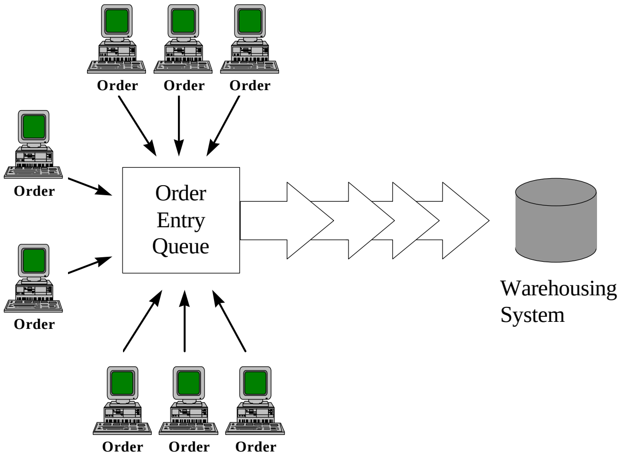


Figure 3: Advanced queuing and deferred transactions

Transaction processing monitor support

Where the same messaging/deferred execution is required across a distributed, heterogeneous database environment, Developer/2000 supports access to TP monitors through the use of PL/SQL packages relevant to each vendor’s product. Oracle8 is an XA compliant resource manager and applications may access data within it via this interface in environments where TP monitors are the preferred solution.

Efficient client-server

Where it processing data across the network is unavoidable (querying rows to the client or performing DML on those records) the method used by the tool to transfer that data may have a serious effect on the scalablity of the application.

Given the finite band width of the network infrastructure as more users make calls across the network, the higher the probability of collisions, network saturation and performance degradation.

Most application development tools available on the market today process records one at a time, causing multiple round trips across the network and hence increasing the overall network load. Developer/2000 is designed to minimize network traffic by performing both fetches and DML operations in arrays of records (rather than one at a time) hence decreasing the number of required trips and maximizing the benefits of Oracle8's parallel query and DML functionality.

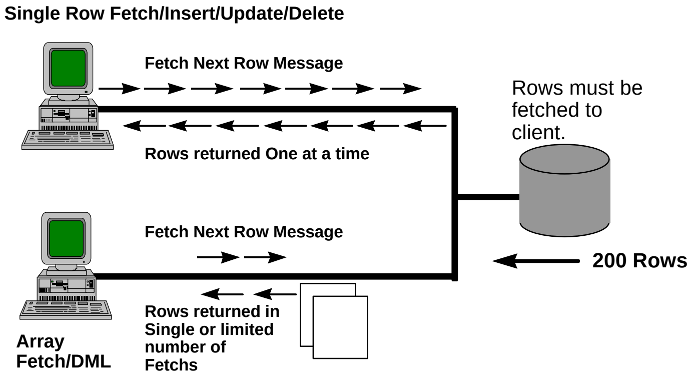

Single Row Fetch/Insert/Update/Delete
Fetch Next Row Message
Rows returned One at a time
Rows must be fetched to client.
Fetch Next Row Message
200 Rows
Rows returned in Single or limited number of Fetchs
Array Fetch/DML

Figure:4 Process Multiple Rows From/to Client Application

Oracle8's companion product Net8 introduces several new features to maximize the scalablity of client-server applications. Through the use of 'connection pooling' and multiplexed connections an application will be able to share multiple client connections through a single transport connection (saving significant machine/network resources). An upcoming release of Developer/2000 will include support for Net8.

NCA Support

Oracle's specification for a Network Computing Architecture (NCA) provides for a cross-platform, standards-based environment for developing and deploying network centric applications. Oracle8 contains an open architecture to support the NCA. Developer/2000 further extends the Oracle8/client-server application model to that of the NCA by the introduction of its various components as NCA supported cartridges. Developer/2000 Forms, Reports and Graphics applications are now accessible over the Web via application server tiers and client side presentation technologies such as JAVA, HTML and Adobe's portable document format.

The ability to partition an application in this manner (between thin client, application server and database server) allows Developer/2000 applications to further extend the scalablity of Oracle8-based applications by distributing processing over multiple tiers and by utilizing the resources/benefits of each tier in turn. By having individual application users run through a series of multi-tiered servers, along with database connection mutiplexing that comes with Net8, Developer/2000 and Oracle8 are well placed for application deployment to extremely large communities.

Universal Data Server Support

The vision of Oracle's Universal Data Server is to allow for the creation of any database application, against any type of data and with any number of online users. Oracle8 takes the vision of a truly universal data server forward with its support for data stores of virtually any size and larger and larger user communities. As significant amounts of corporate information is held externally to a traditional database, a truly universal application must be able to take advantage of emerging rich data types beyond that which is modeled in the database.

Freeform text

Over 90% of a corporation's knowledge is held external to the databases in an unstructured format (e.g. word processing documents), yet this vital information still needs to be searched in a expedient, secure method and the result displayed. Through the use of the CONTEXT OPTION, the Oracle8 Universal Data Server is able to store and search documents based on both a word and thematic basis. The interface to this functionality is via a PL/SQL library, allowing Developer/2000 applications to directly access this method of information storage and to display the resulting documents on the client (or Web).

Stream Data (audio/video)

Humans assimilate information significantly more easily when it is presented in a multimedia format such as audio/video. Because of this, more and more organizations are looking to incorporate these media types into their applications, enabling their users to get a greater benefit from the organization's information (such as, online help systems, data kiosks and medical systems). Support for stream based data with the Oracle8 Universal Data Server is via the VIDEO SERVER OPTION. Applications built with Developer/2000 may utilize stream based data from the Video server in two ways depending on where the application is being run. On a MS-Windows based PC, Developer/2000 supports the use of the client-side component of the video server, while in a NCA/web based application the browser itself becomes the video client.

Spatial Data

Applications developed with Developer/2000 can support the use of spatial data through its support for third party display mechanisms such as Graphical Information Systems (GIS) packages.

Oracle8 Object Support

With the requirements of business systems changing rapidly, many organizations are looking to model their real life processes using an object-based metaphor. Developer/2000 aids this technique by supporting the definition of objects within a class library. Object stored within this library (which may be stored within the Oracle8 database), may be sub-classed to allow for the easy reuse of both visual objects and methods within the development environment. Furthermore, the introduction of Oracle8 takes this modeling technique beyond the development environment, by the evolution of relational data to include many of the concepts found in object based systems. (such as object types and methods associated with them).

With this in mind, Developer/2000 will phase in support for Oracle8 object types, at the same time furthering its support for VLDB and large numbers of users in an OLTP environment. The first phase will introduce support for object data types, LOBs, object views, nested tables and methods, while subsequent releases will further extend object support to include Varrays and object Refs (relations). To ease the adoption of object-based data, the wizard based development metaphor employed within Developer/2000 is being enhanced to accommodate visualization of this type of data, allowing for the easy interpretation and manipulation of object based data via a familiar mechanism.

CONCLUSION

The introduction of Oracle8 promises to push open databases and the applications built on them to a new level of scalablity both in the amount of data stored but also in the number of users accessing the data. Designer/2000 provides the most complete support for modeling and generating both the relational and object constructs in Oracle8. Developer/2000 is unique in its ability to scale applications to the required user loads, and as more Oracle8 technology is adopted by application developers further enhancements to the product will ensure that this scalablity continues as applications continue to grow and expand in their scope. Together, Designer/2000 and Developer/2000 provide the most comprehensive application development environment for building Oracle8 applications.

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.415.506.7000
Fax +1.415.506.7200
http://www.oracle.com/

Database Application Development Tools for Oracle8

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document