Preface

Why This Book?

Object orientation has now invaded traditional relational database-management systems. Oracle™ without exception has included object-oriented features in its system. SQL is now richer due to these additional features. However, the object-oriented elements in Oracle™ will not be fully utilized without a proper database design. For example, a database application designed using a traditional database modeling, such as entity-relationship (E/R) modeling, will not be able to make use of most object-oriented features in Oracle™. This is simply due to the absence of object-oriented elements in the design. Even with a proper object-oriented design, without careful transformation from design to implementation, many of the object-oriented features will be lost.

Object-Oriented Oracle™ addresses this need by not only explaining the new object-oriented features in Oracle™, but most importantly how these features can be fully utilized in database applications. We put a heavy emphasize on how an object-oriented conceptual model is implemented in Oracle™. This includes the static aspect of an object-oriented conceptual model, including the inheritance, association, and aggregation relationships, as well as the dynamic aspect covering generic object-oriented methods and user-defined queries.

Just as we enjoyed writing this book, we hope that you will enjoy reading it, and most importantly gain valuable lessons from it. We trust that this book will give you a comprehensive insight into object-oriented Oracle™.

Distinguishing Features

Object-Oriented Oracle™ presents the right mix between theoretical and practical lessons on object-oriented features of Oracle™.

In the theoretical part, it describes the foundation of object-oriented concepts and how they are used in the implementation. The importance of these concepts is invaluable because without this understanding, the new object-oriented features offered by Oracle™ will not be fully utilized. Therefore, these theoretical elements serve as the foundation of object orientation in Oracle™.

In the practical part, the book contains two case studies (Chapters VII and VIII) that thoroughly explain the development of a database application using the object-oriented technology of Oracle™. The case studies start with the description of an application, followed by the appropriate object-oriented designs. The designs are then transformed for implementation in Oracle™.

Each chapter also contains extensive examples and code. These examples and code will give readers a better understanding of how object-oriented elements are used in Oracle™.

At the end of each chapter, a set of problems, together with their solutions, are given. These will be suitable exercises for the classroom. The solutions will be useful for both students and their teachers.

Topical Coverage

Object-Oriented Oracle™ contains eight chapters.

Chapter I starts with object-relational approaches that cover the object-oriented conceptual model. There have been many approaches in amalgamating the object-oriented model with database systems, from which the new era of object-relational databases is born.

Chapter II explains object-oriented features in Oracle™. These include the use of *type* and *object* in conjunction with table creation, *varray*, and *nested table*. These features, together with the *ref* relationships, index cluster, and the *under* clause for subtyping, change the whole concept of database modeling.

Chapter III describes how these object-oriented features should be properly used in Oracle™. This includes how the object-oriented conceptual model described in Chapter I is implemented using the features presented in Chapter

II. This chapter particularly focuses on the static aspect of the object-oriented conceptual model, including the inheritance, association, and aggregation relationships.

Chapter IV justifies how the dynamic aspect of the object-oriented conceptual model (encapsulation and object-oriented methods) is implemented using the new features of Oracle™, namely member procedures and functions.

Chapter V describes generic methods in Oracle™. This covers generic methods found in the object-oriented conceptual model, including the inheritance, association, and aggregation relationships. The generic methods comprise typical database operations (e.g., update, delete, and insert) applied to the member attributes of a class. The use of generic methods is a direct implementation of object-oriented encapsulation features.

Chapter VI focuses on user-defined queries. New SQL features, covering referencing and dereferencing using *ref*, super- and subclass accesses using *treat*, nesting techniques using *the* and *table*, are explained. The chapter also discusses the *varray* and nested-table collection types, object references *deref*, the *is dangling* clause, and object attributes.

Chapter VII introduces a university case study that contains a database to maintain the running of courses in a university. This case study shows the entire database-application development life-cycle process from the object-oriented design to transformation for implementation in Oracle™.

Finally, Chapter VIII presents another case study based on a retailer-chain company. In addition to using the object-oriented conceptual model for the database design, implementation is carried out using Oracle™ Form Developer. The aim is to show how a window-based database application can be developed using the object-oriented technology in Oracle™.

Intended Audience

Object-Oriented Oracle™ is intended for the following audiences.

- **Database Practitioners**

  Object orientation in Oracle™ has now opened a wide opportunity in exploring new ways for building database applications. This book shows how object-oriented features can be adapted for database-application development. It describes not only the practical aspects of database-application development, but also the theoretical foundations that lead to the use of the object-oriented technology in database applications using Oracle™. The two case studies included in this book show the two flavours of database applications using the object-oriented technology as their foundation whereby the first application is a text-based application, and the second is window-based using Oracle™ Form Developer.

- **College Students and Teachers**

  This book is suitable as a textbook for database courses at any level: an introductory database course whereby this book can be used as a supplement to the standard database-management textbook, or an advanced database course concentrating on object-oriented database development. Students who are learning the standard material of SQL are now able to learn, at the same time, the new object-oriented features of SQL. Furthermore, students are now able to relate how a database design, in this case using an object-oriented method, can smoothly be implemented in Oracle™, thus making the entire database-application-development life cycle transparent.

- **General IT Readers**

  General IT readers who are keen on the new technology of Oracle™ will find this book useful and informative. Object orientation has been an interesting topic in general due to the popularity of object-oriented programming languages, like C++ and Java. The object-oriented concepts, which underpin these programming languages, have been widely understood. However, their applications to database systems have not been broadly explored. This book demonstrates how object-oriented features could be used easily in Oracle™, and most of all, how they could be used appropriately and efficiently.

- **IT Researchers**

  Object orientation in relational database systems has been an active research area in the last decade. Many researchers have proposed methods for transforming object-oriented design to relational database implementation. Other groups of researchers have been concentrating on object-relational databases. Due to the increasing trend whereby most database-management-system vendors are positioning themselves in the object-oriented tracks, there are plenty of research opportunities in this important area. This book will give researchers the basic foundation for amalgamating two different elements: object-oriented and relational database systems.

Feedback and Comments

Although we have fully tested all code included in this book, should there be any problems or confusion about the code, please do not hesitate to contact us.

We would appreciate if you could also share any other comments or feedback with us so that we can incorporate them in a future edition. Comments and feedback may be sent directly to the publisher at

Object-Oriented Oracle™
Idea Group Inc.
701 East Chocolate Avenue, Suite 200
Hershey, PA 17033-1240, USA