

## Tutorial Proposal

**Tutorial Title:** "Objection Sustained!" An Introduction to Object Database Management Systems

**Tutorial Organizers:** George J. Pothering and Roxann H. Stalvey  
Computer Science Department  
College of Charleston  
Charleston, South Carolina 29424  
USA  
stalveyr@cofc.edu, pother@cs.cofc.edu  
1-843-953-8156

### Introduction

A quarter century ago the database management system world was involved in a tug-of-war as systems based on the network and hierarchical models were being challenged by the relational model. A similar struggle is taking place today as the complexity of today's data (images, sounds, etc.) have pushed the limits of the relational model. The established vendors of the relational database management systems (Oracle, Sybase, Informix) have offered the object-relational database management system (ORDBMS) to attempt to meet these new demands. The current pervasiveness of object-oriented programming and the desire to share persistent object data among diverse systems via the Web, however, may ultimately result in the torch being passed to a class of database management systems specifically designed for these tasks – object database management systems (ODBMSs).

We propose to conduct a half-day tutorial focusing on object database management systems (ODBMSs). An ODBMS can be loosely defined as a database to store persistent objects. Typically, the objects to be stored in an ODBMS are created using an object-oriented language supported by the ODBMS (typically Java or C++). They are created in exactly the same way as non-persistent objects but inherit additional characteristics that make them persistence-capable. Likewise, objects retrieved from an object database are manipulated or used in a program as any other object would be. The key to working with an ODBMS, therefore, is to have a familiarity with the special classes that are provided by the ODBMS for the object-oriented languages it supports in order: to create or access a database, to specify the classes of objects it stores and any relationships among these objects, and to store and retrieve objects in the database. The purpose of our tutorial is to develop this familiarity and to give the tutorial participants an opportunity to work with an ODBMS during the workshop.

### Content and Activities

The tutorial will be conducted as a three-hour, integrated lecture and laboratory, featuring *PowerPoint* slides and program demonstrations. A distinguishing feature of the tutorial will be the establishment of a local wireless network within the lecture room so that those participants who bring laptops equipped with wireless cards will be able to access an ODBMS via programs running from their own systems. The particular ODBMS chosen (*Objectivity*) supports programs written in Java and C++. By allowing the participants to work on their own systems and with a familiar development environment for their Java or C++ programs, we hope to substantially reduce the time required to get programs up and running.

Among the topics to be covered are:

1. A comparison of ODBMSs with relational DBMSs and object-relational DBMSs.
2. A review of data modeling concepts, especially one-to-one, one-to-many, and many-to-many relationships.
3. The ODMG standard 3.0 on specifications for persistence of object-oriented programming language objects in databases. In particular we will focus on its specifications for an object definition language (ODL) and for an object query language (OQL). The object definition language is used to define the object types corresponding to a given data model, while the object query language (based on SQL) provides a nonprocedural language for querying and updating objects stored in an object database.

ODMG Standard 3 also specifies an object interchange format (OIF) to facilitate transfers between an ODBMS and a file or set of files, but we shall only address this if time permits.

4. Language bindings between C++ or Java and an ODMG-compliant object database. Unlike relational databases, object databases require an object-oriented host language to create the objects that are to be made persistent and to invoke OQL queries to retrieve and manipulate persistent objects.
5. An introduction to *Objectivity* and to *Objectivity's* support for the ODMG 3 standard.
6. Database structure in *Objectivity* – federated databases and databases. How to create federated databases and databases in *Objectivity*.
7. Storing objects in, and retrieving objects from an *Objectivity* database.
8. The machinery of concurrency in *Objectivity* – sessions, transactions and locks.
9. Object Query Language.
10. A comparison of object databases and object-relational databases
11. ODBMSs in the undergraduate computer science curriculum, especially the early CS curriculum.

### ***Advance Program Description***

An introduction to object database management systems. Topics will complement those covered in a typical first course in databases based on the relational model and will include data modeling concepts, language bindings for C++ and Java, and transactions, but as they apply to object databases. Users with laptops equipped with a Wi-Fi certified card can work with an object DBMS via a wireless network to be established just for this tutorial.

### ***Background of the Presenters***

George Pothering is a Professor of Computer Science at the College of Charleston. RoxAnn H. Stalvey is an Instructor of Computer Science at the College of Charleston