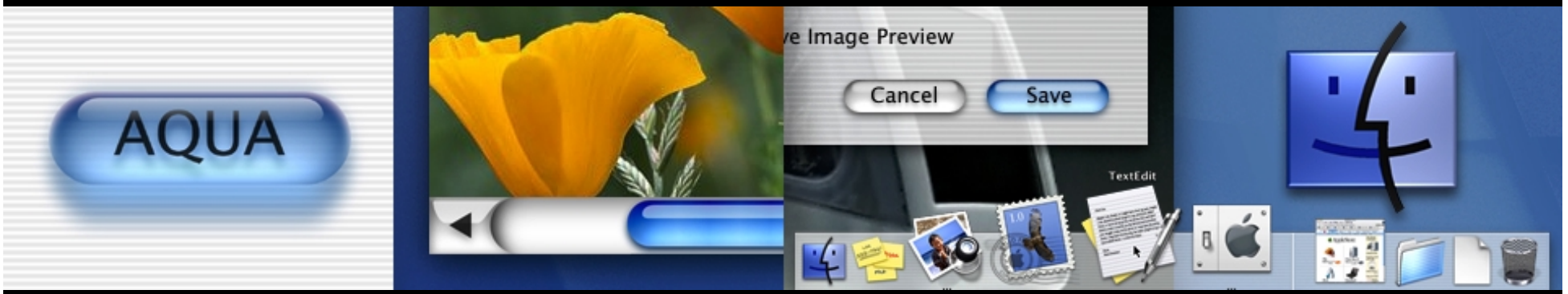
Session 403

Persistent Object Modeling with EOModeler



Mike Gobbi
Senior Developer Trainer, Apple iServices

Introduction

- WebObjects is a powerful technology for building dynamic web sites
- Most dynamic web sites are backed by relational databases
- EOF allows you to access persistent data from within your object-oriented application

What You'll Learn

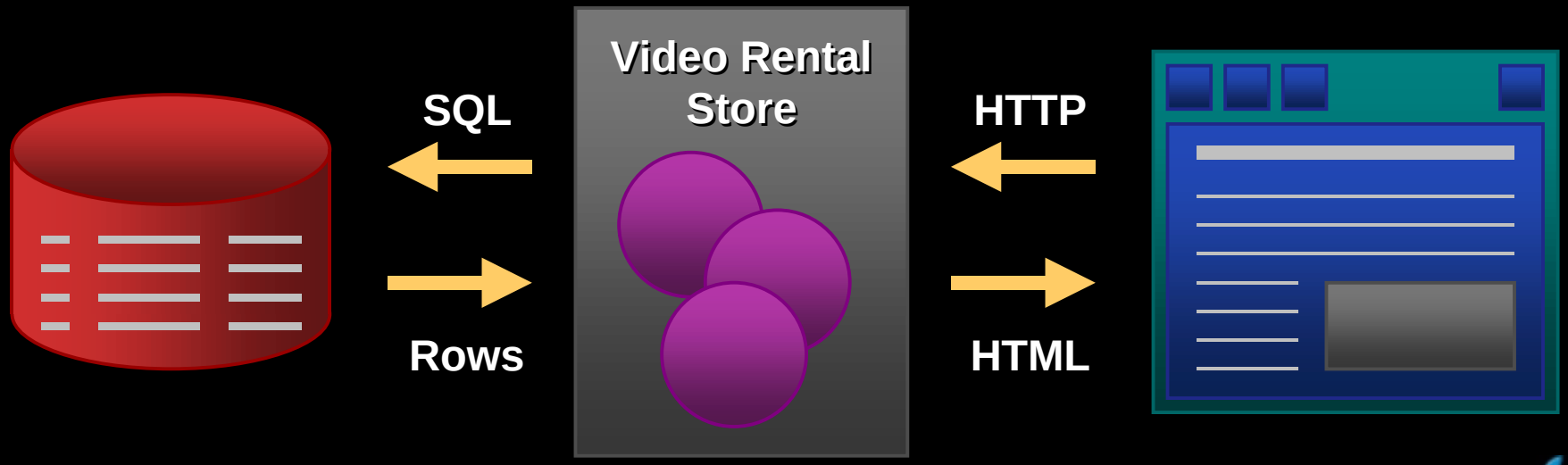
- Basic EOF and WOF concepts
- How to use EOModeler to
  - Work with the database schema
  - Define validation rules and fetch specifications for use at runtime

Part 1: Basic Concepts

- Structure of a Web application
- WOF/EOF architecture
- Entity-Relationship modeling concepts

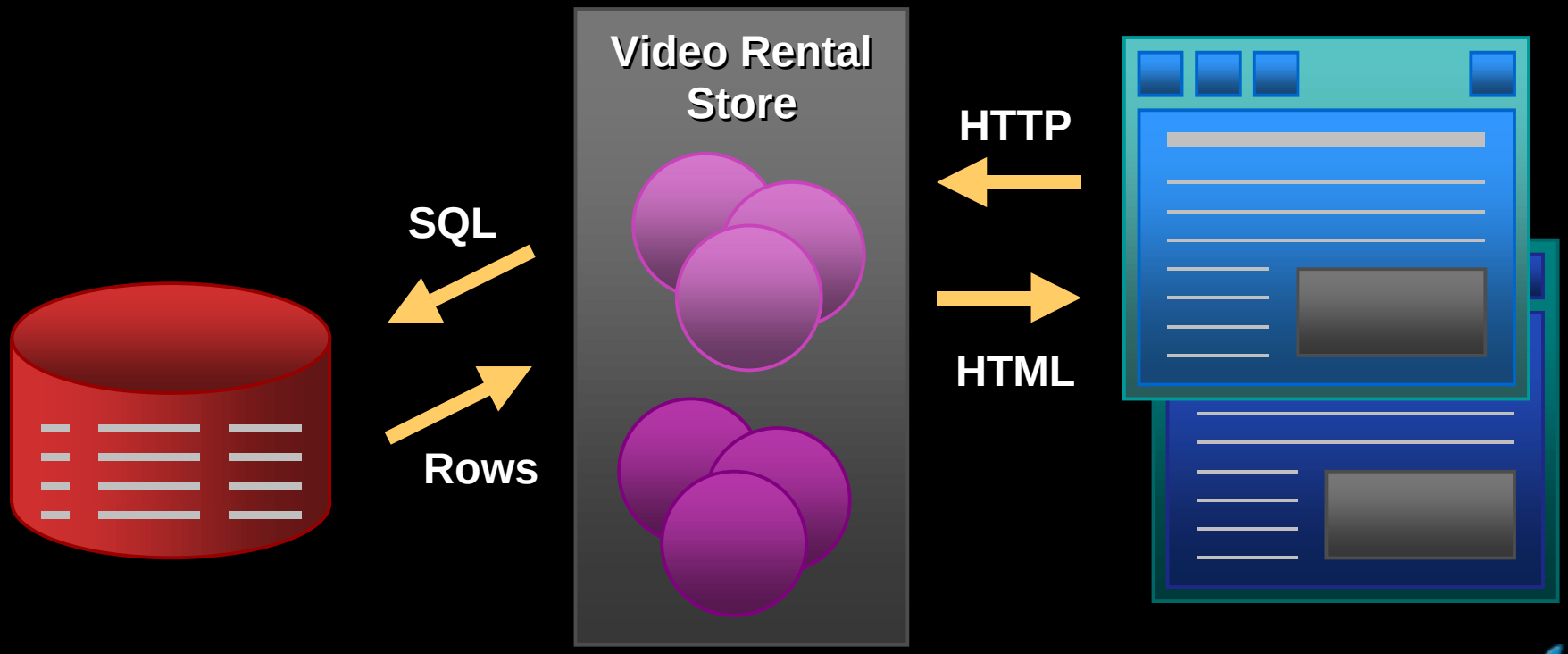
Web Applications

- Service web requests using information stored in relational databases



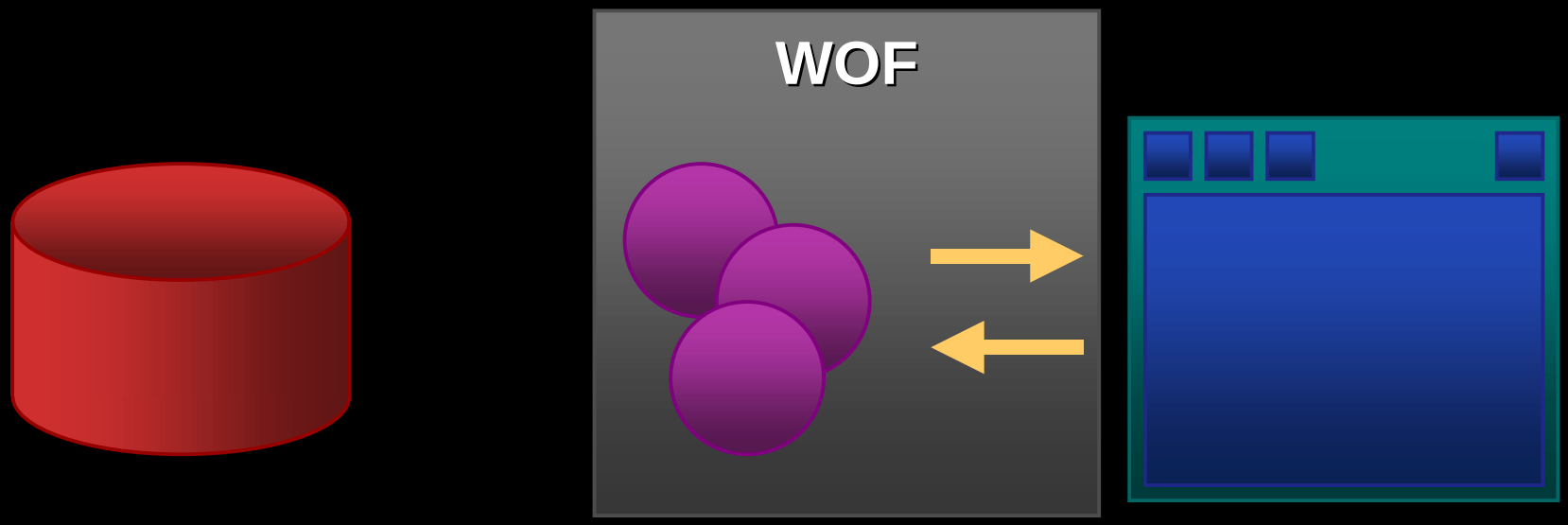
Web Applications

- Maintain a unique editing context for each user



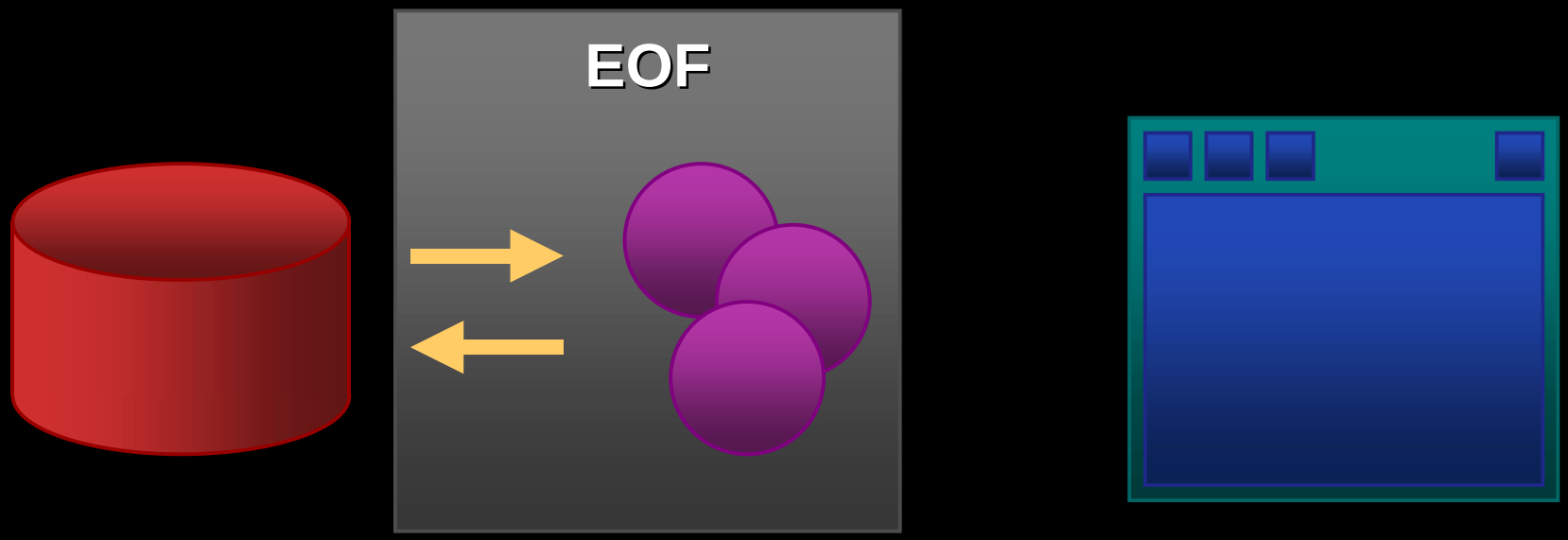
WebObjects Framework

- Creates HTML pages from your business objects

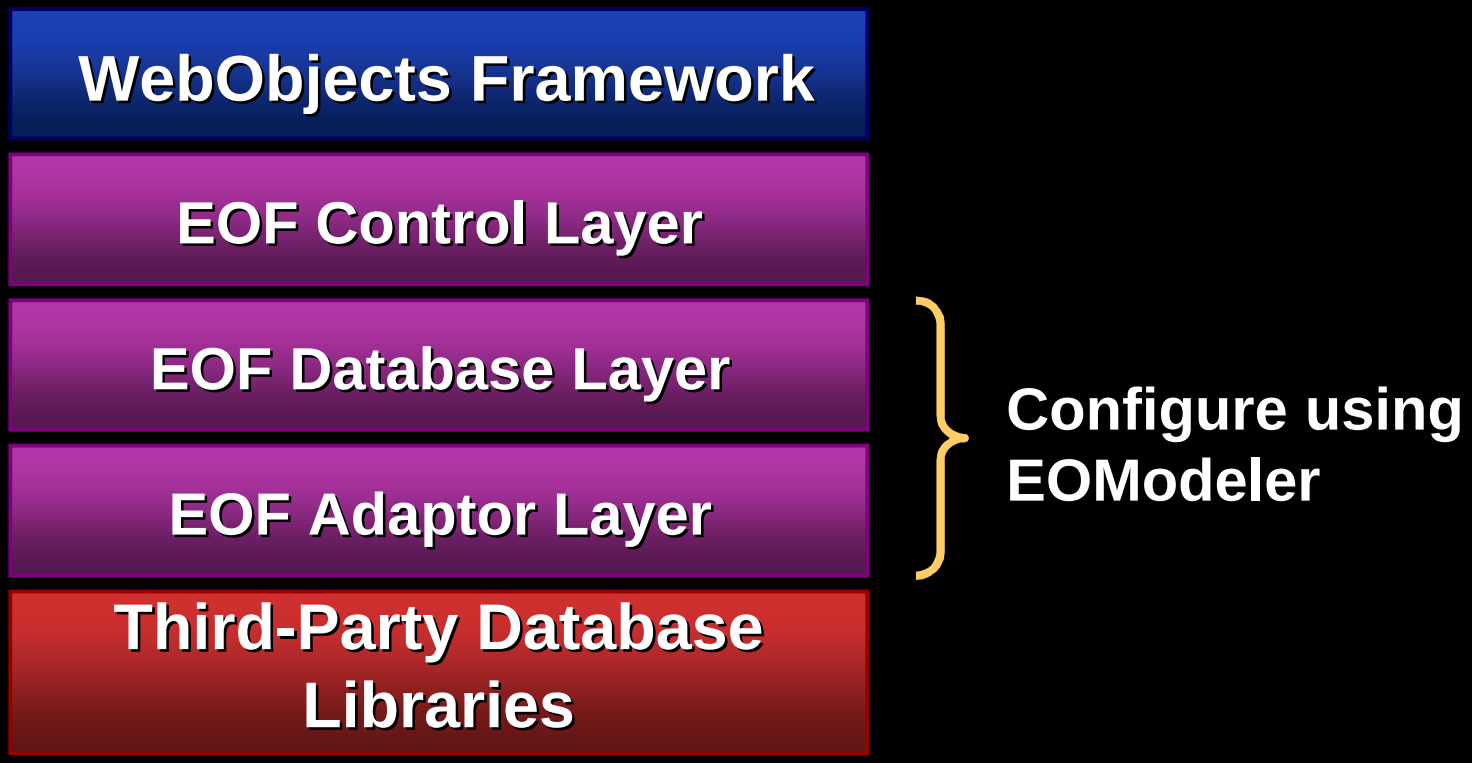


Enterprise Objects Framework

- Creates business objects from database records

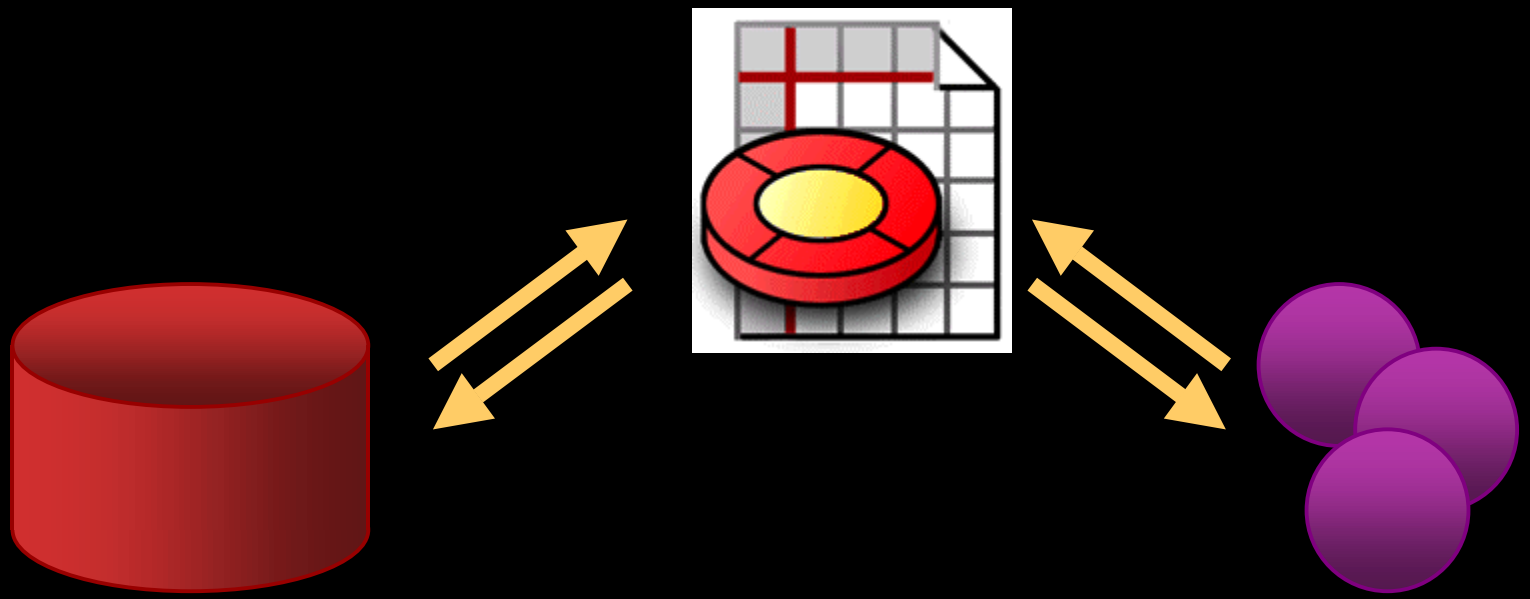


WebObjects Application Architecture



Model File

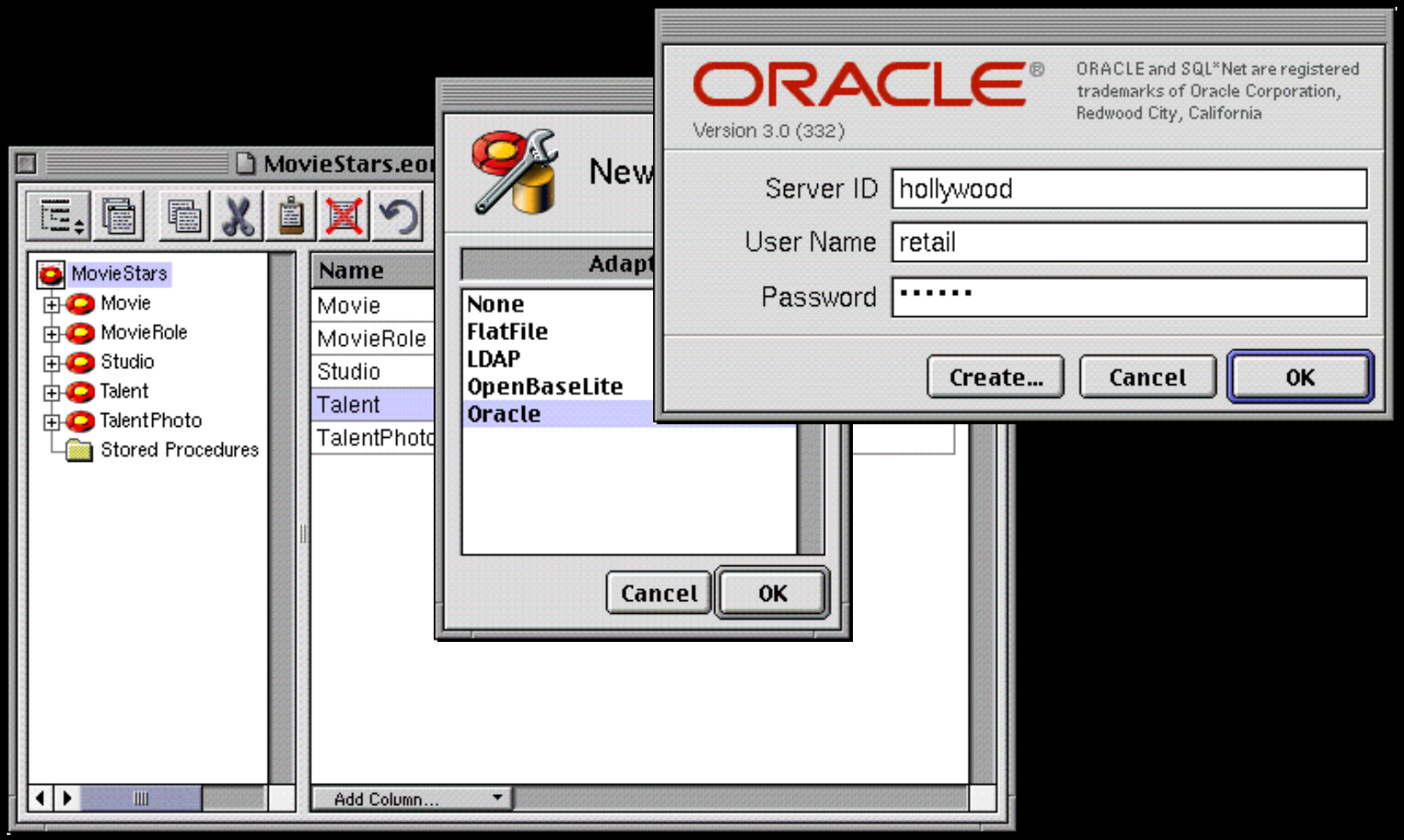
- Describes the mapping between database tables and object-oriented classes



EOModeler

- A tool for editing model files:
  - Configure database adaptor
  - Work with schema:
    - Entities
    - Attributes
    - Relationships
  - Identify stored procedures
  - Create fetch specifications

Configuring the Database Adaptor



Entities: Table and Class

STUDIO

NAME	STD_ID
Warner Brothers	112
20th Century Fox	113

MOVIE

TITLE	CATEGORY	MOV_ID	STD_ID
Casablanca	Drama	345	112
Alien	Horror	456	113
Star Wars	Action	567	113

Movie
title "Casablanca"
category "Drama"
studio

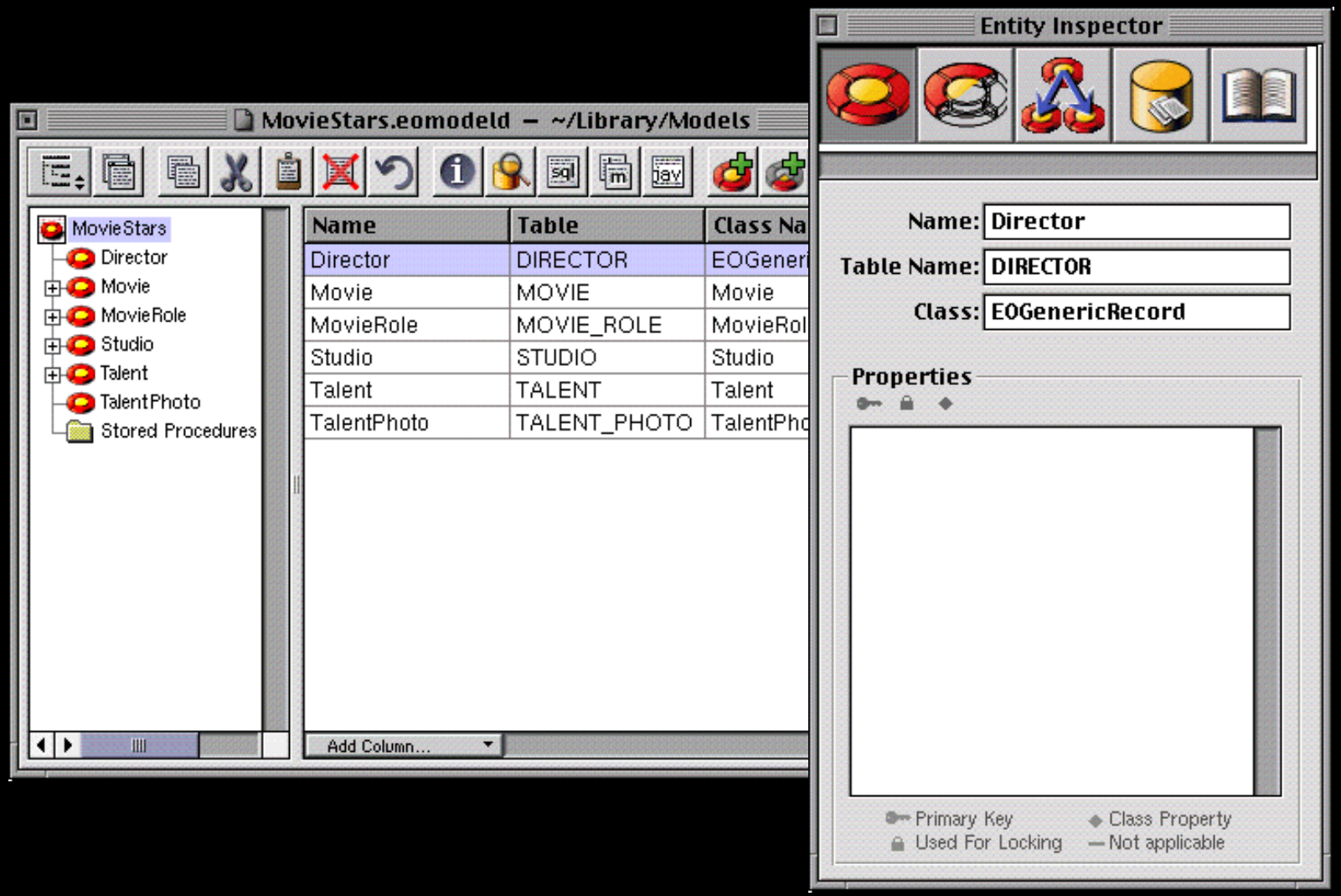
Movie
title "Alien"
category "Horror"
studio

Movie
title "Star Wars"
category "Action"
studio

Studio
name "Warner Brothers"
movies

Studio
name "20th Century Fox"
movies

Creating an Entity



Attributes: Column and Instance Variable

STUDIO

NAME	STD_ID
Warner Brothers	112
20th Century Fox	113

MOVIE

TITLE	CATEGORY	MOV_ID	STD_ID
Casablanca	Drama	345	112
Alien	Horror	456	113
Star Wars	Action	567	113

Movie
title "Casablanca"
category "Drama"
studio

Movie
title "Alien"
category "Horror"
studio

Movie
title "Star Wars"
category "Action"
studio

Studio
name "Warner Brothers"
movies

Studio
name "20th Century Fox"
movies

Simple Attributes

- Represent simple data types
  - Strings
  - Integers
  - Floating-point numbers
  - Dates
- Are automatically converted to objects by EOF

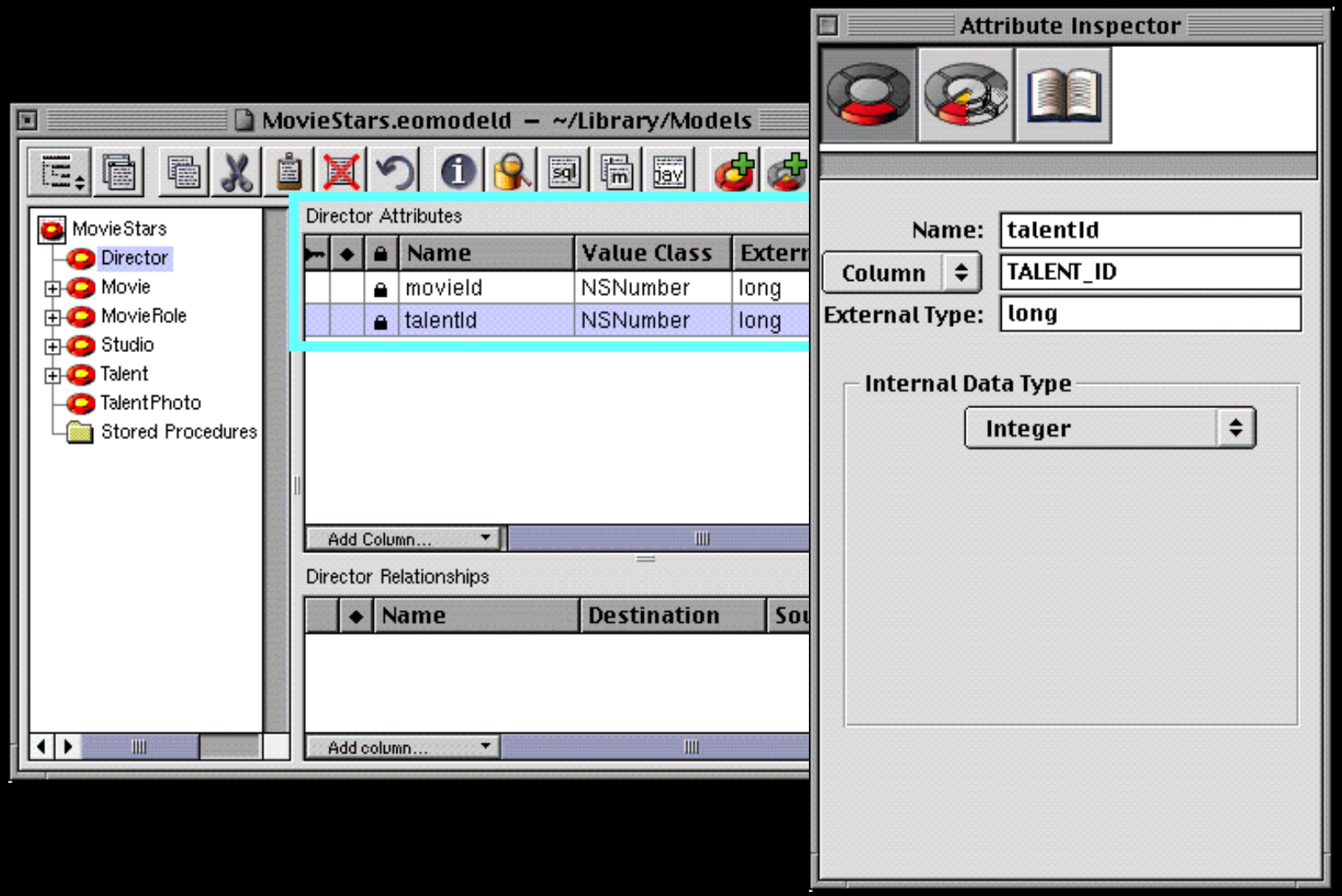
Blobs

- Represent arbitrary data types:
  - Images
  - Sounds
  - QuickTime movies
- Are read from the database as NSData objects
- You specify class and factory methods

Primary and Foreign Keys

- Are columns in the database
- Generally do *not* appear as instance variables in the application
- Are used to define relationships

Editing Attributes



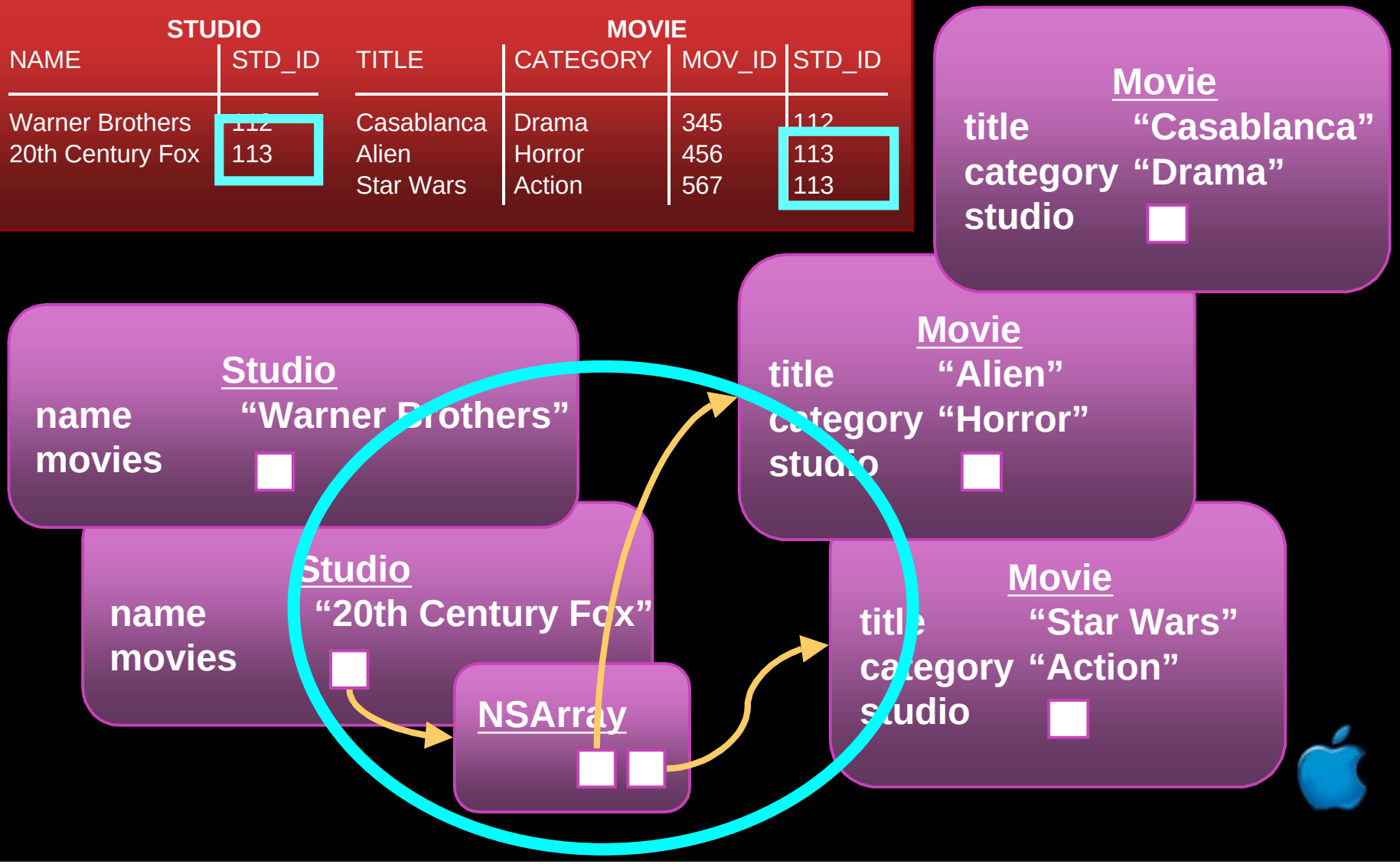
Relationships: Join and Object Reference

STUDIO

NAME	STD_ID
Warner Brothers	112
20th Century Fox	113

MOVIE

TITLE	CATEGORY	MOV_ID	STD_ID
Casablanca	Drama	345	112
Alien	Horror	456	113
Star Wars	Action	567	113



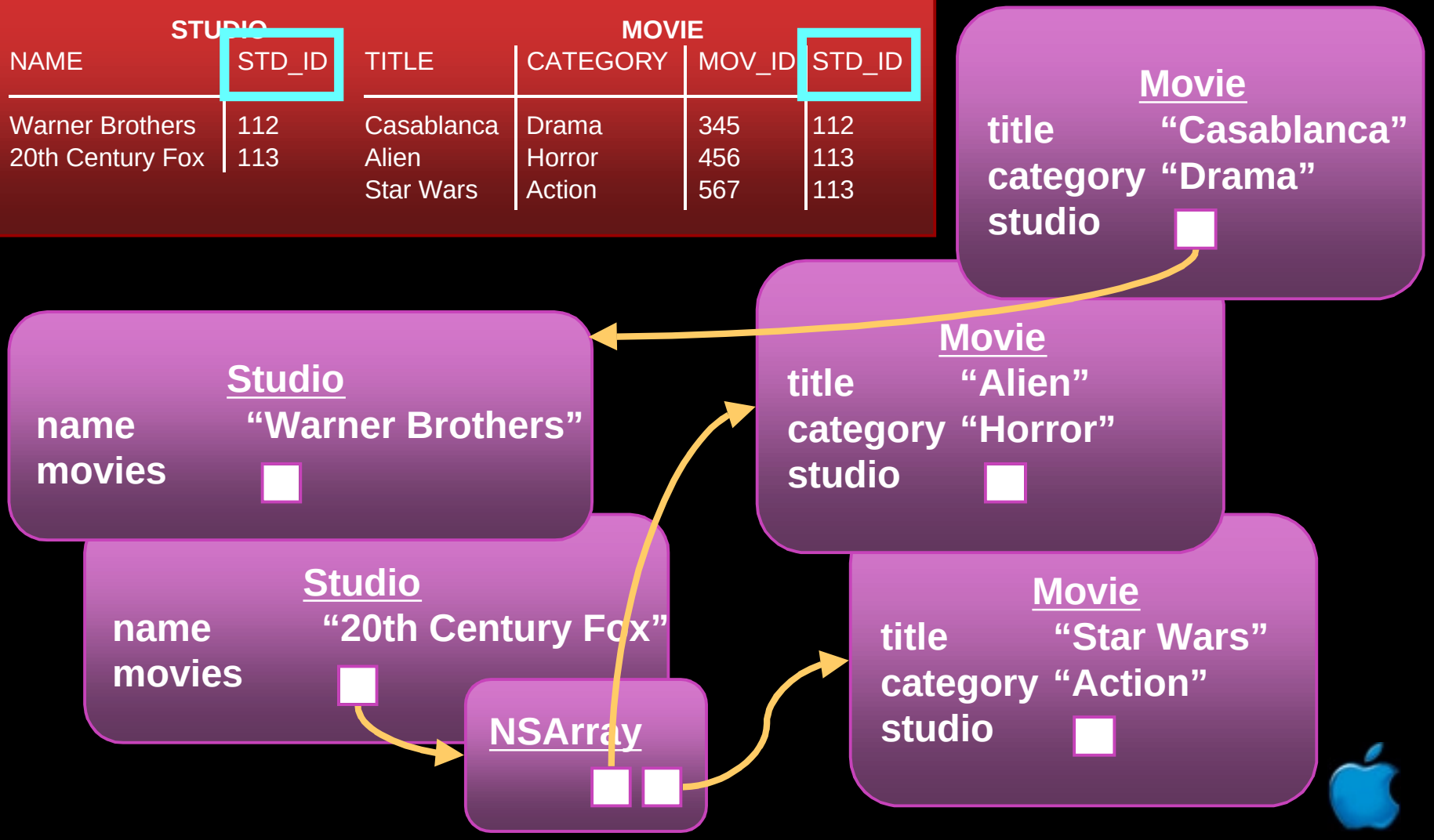
To-One vs. To-Many Relationships

STUDIO

NAME	STD_ID
Warner Brothers	112
20th Century Fox	113

MOVIE

TITLE	CATEGORY	MOV_ID	STD_ID
Casablanca	Drama	345	112
Alien	Horror	456	113
Star Wars	Action	567	113



Two-Way vs. One-Way Relationships

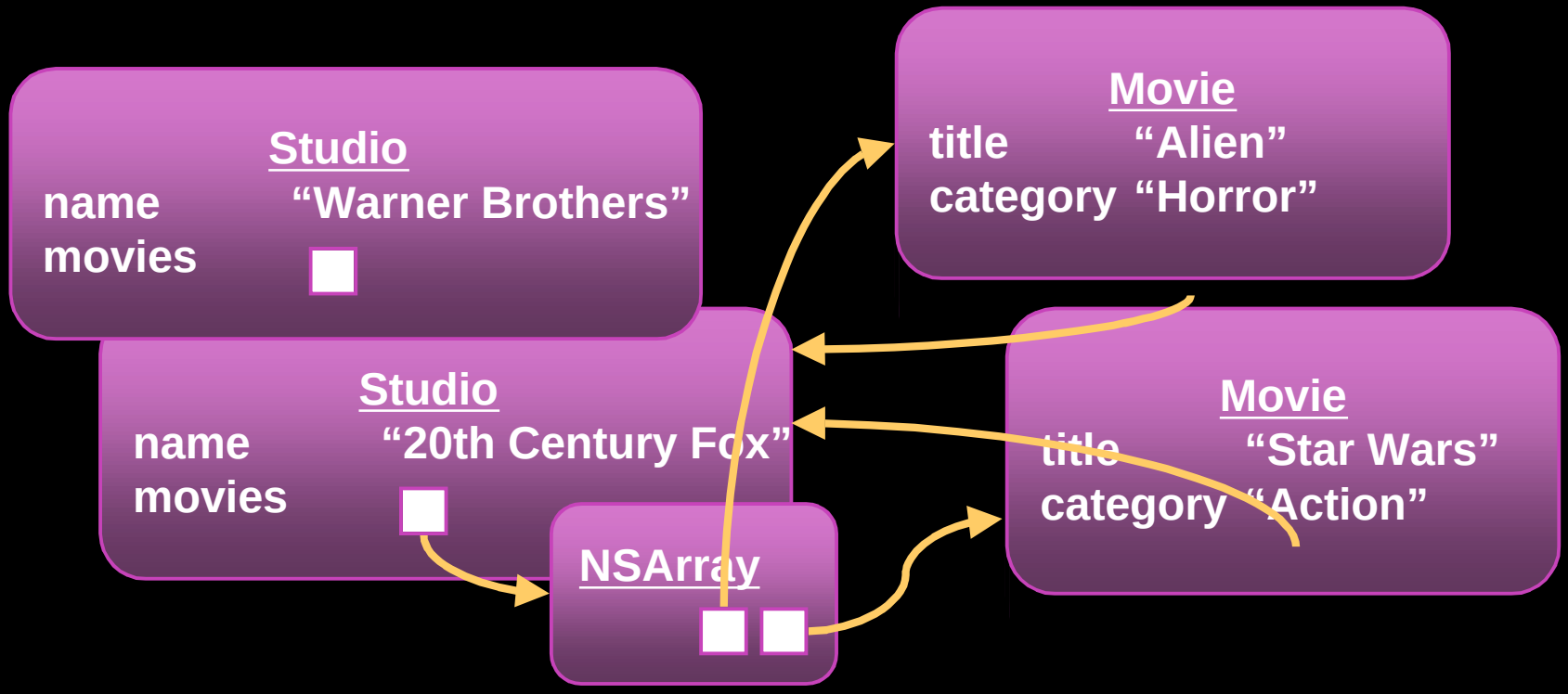
STUDIO

NAME	STD_ID
Warner Brothers	112
20th Century Fox	113

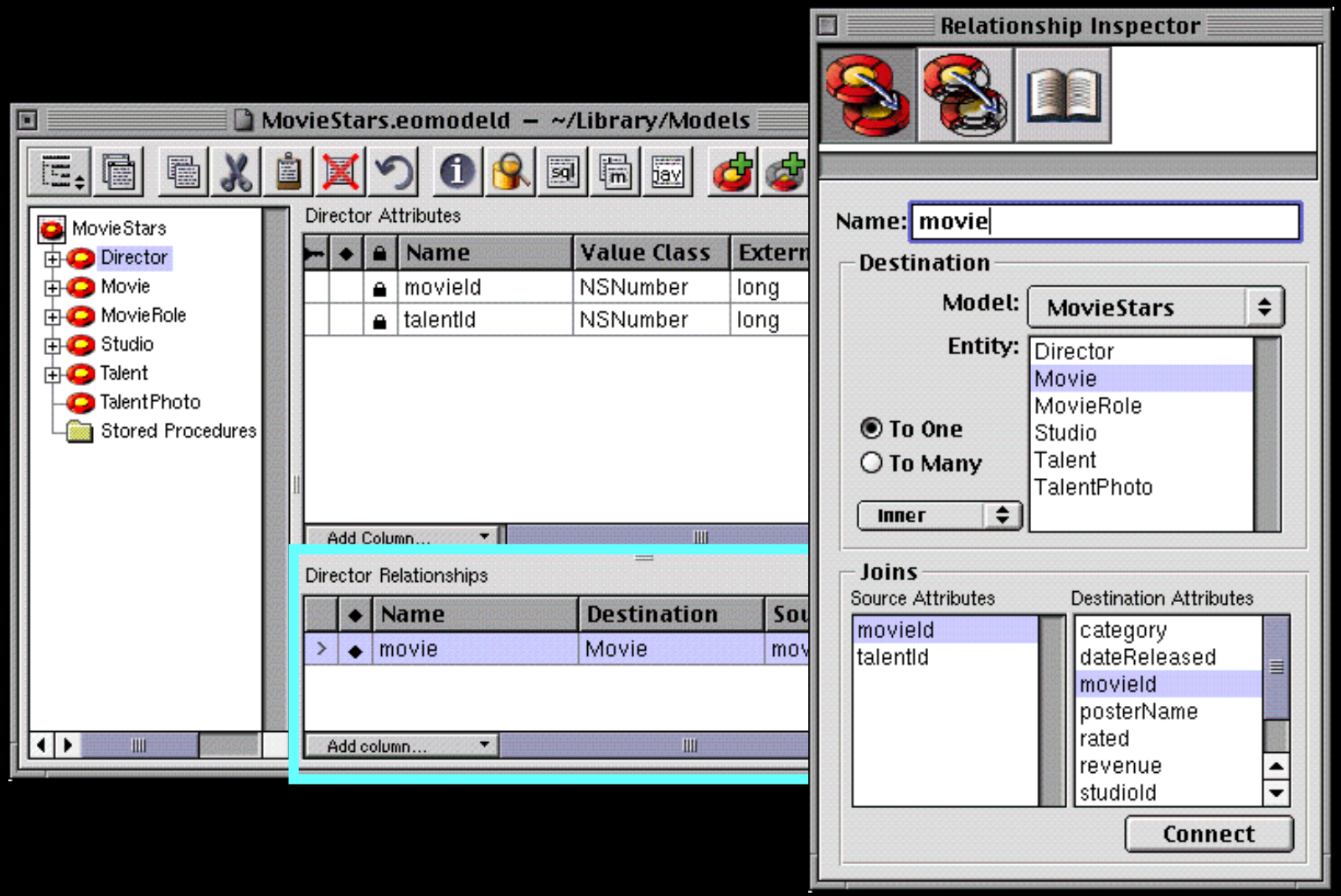
MOVIE

TITLE	CATEGORY	MOV_ID	STD_ID
Casablanca	Drama	345	112
Alien	Horror	456	113
Star Wars	Action	567	113

Movie
title "Casablanca"
category "Drama"



Editing Relationships



Many-to-Many Relationships

- Join table is not in application

MOVIE

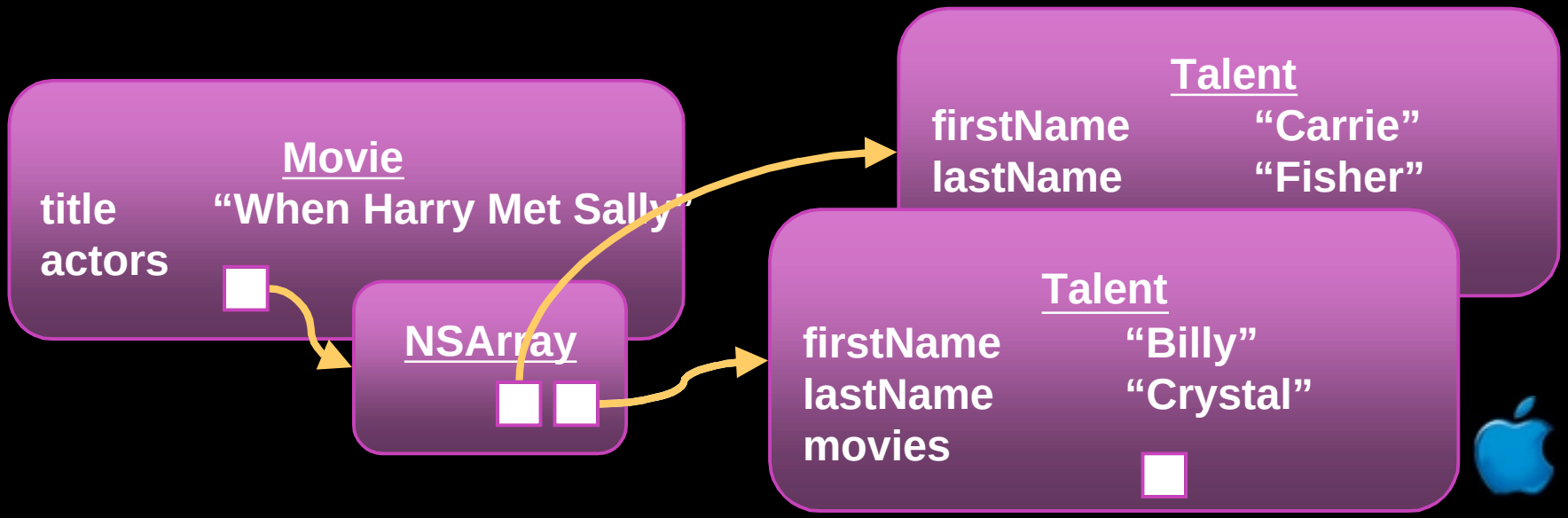
TITLE	MOV_ID
Star Wars	567
When Harry Met Sally	789

ACTED_IN

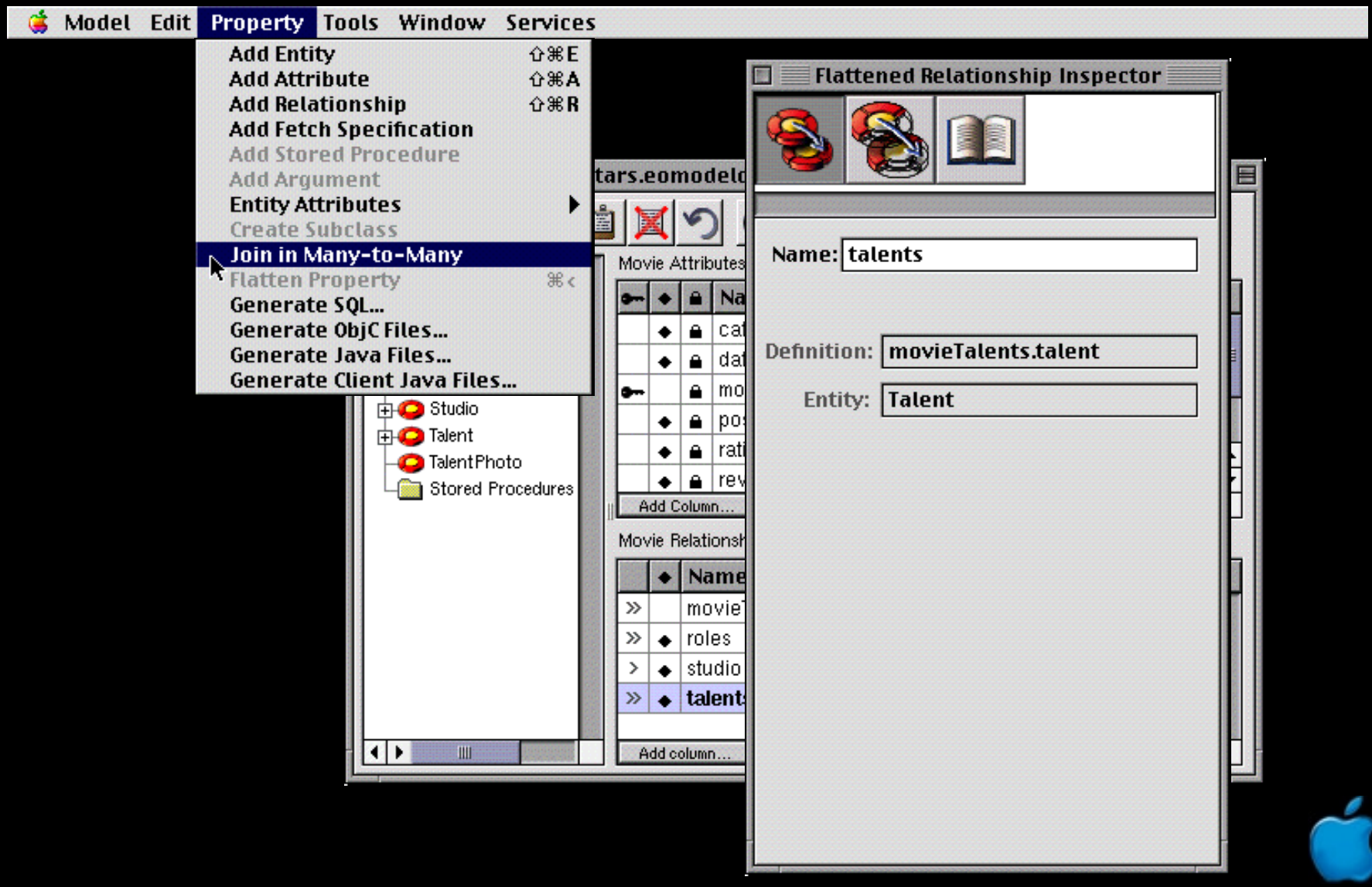
MOV_ID	TLT_ID
567	33
789	33
789	44

TALENT

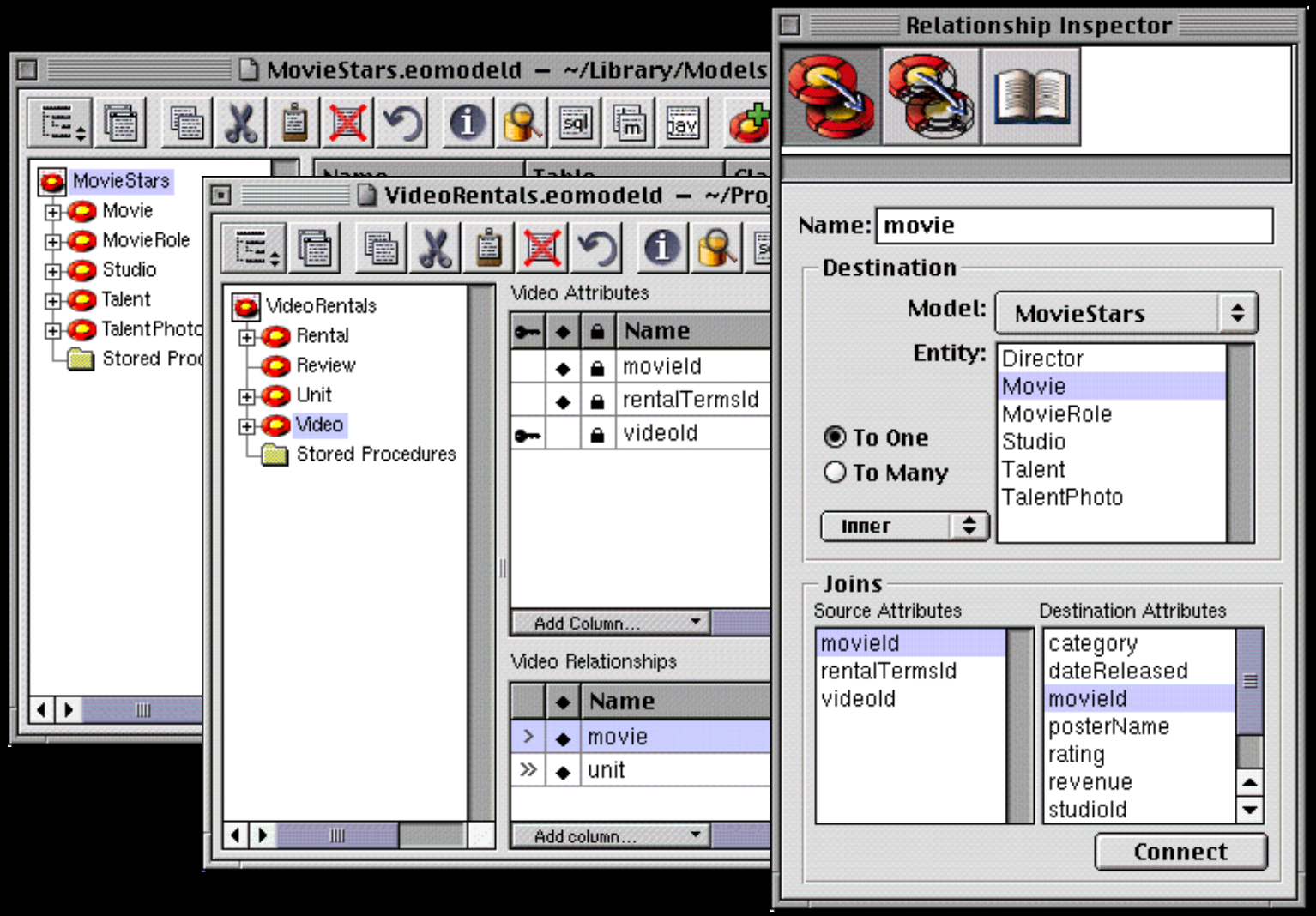
FIRST	LAST	TLT_ID
Carrie	Fisher	33
Billy	Crystal	44



Many-to-Many in EOModeler



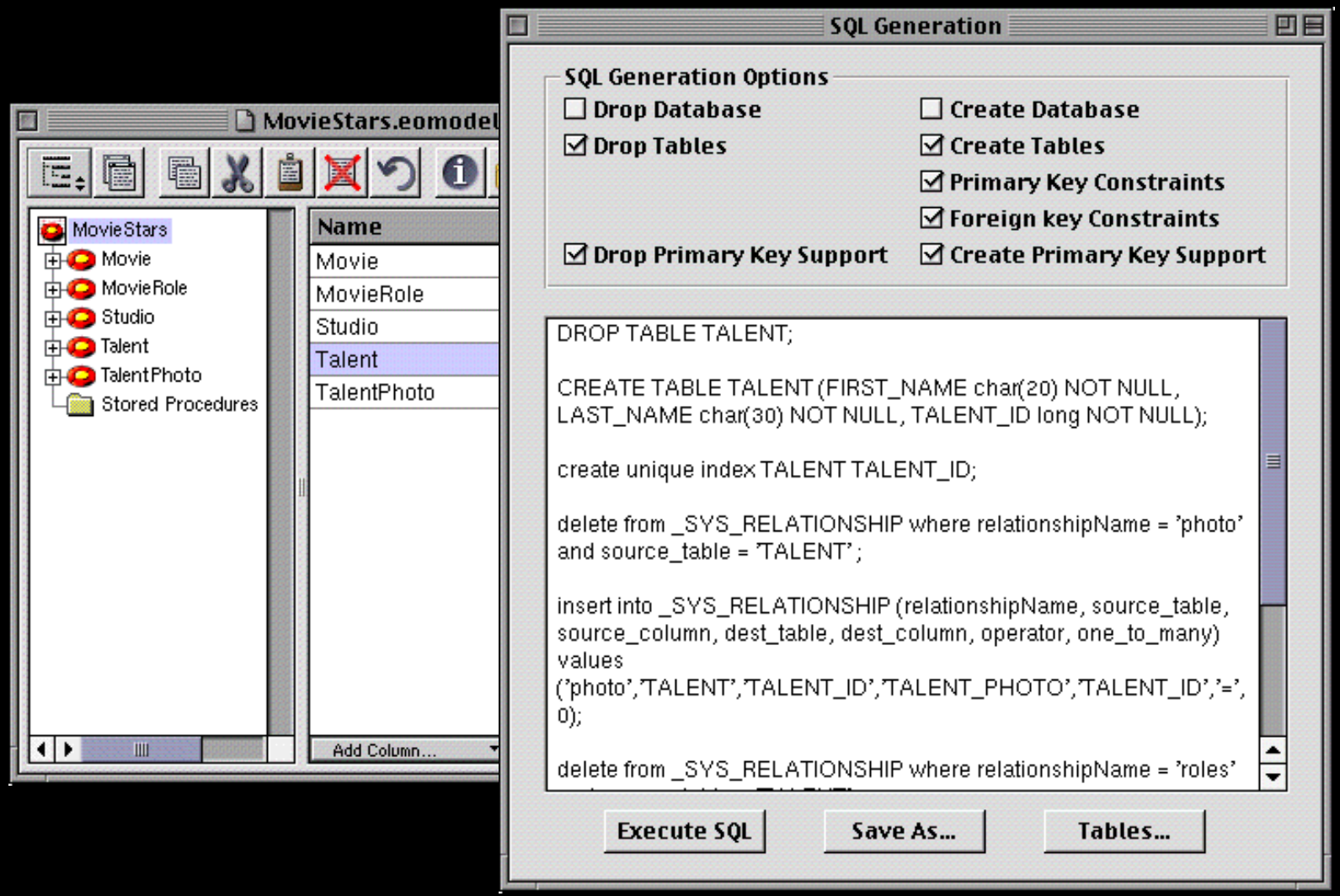
Relationships Between Different Databases



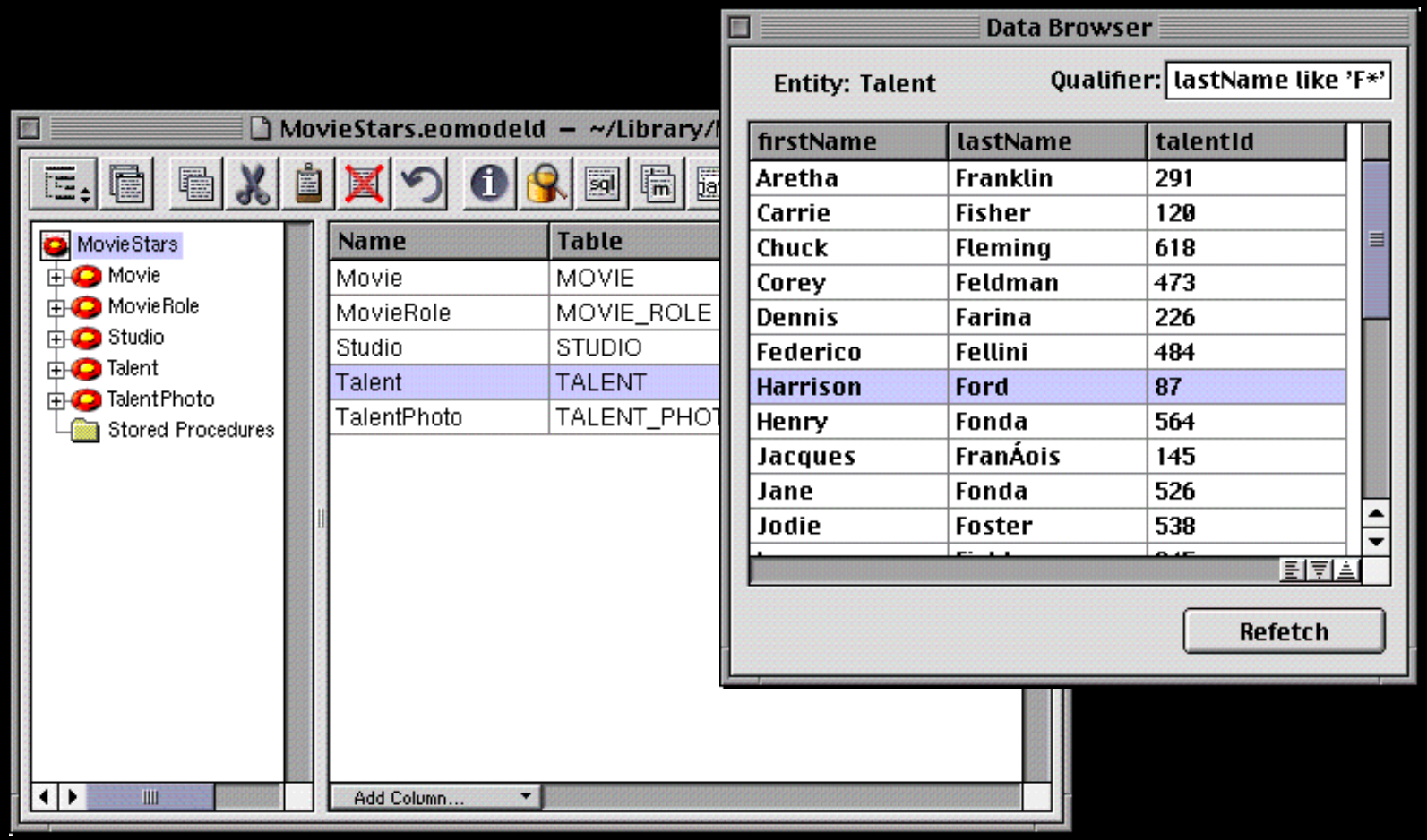
Part 2: Working With the Entities

- Generating SQL
- Browsing data
- Importing an existing database
- Generating Java classes

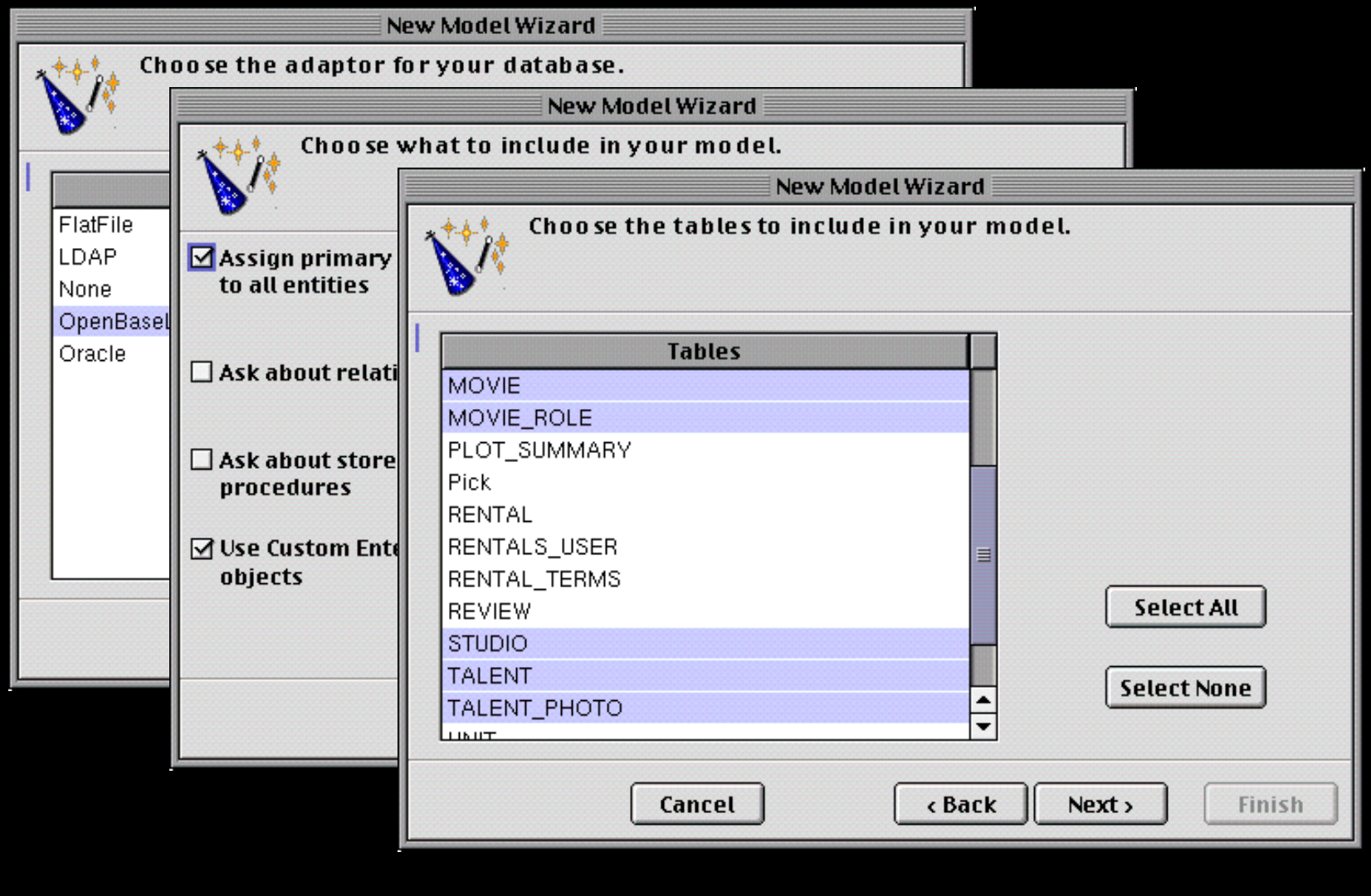
Generating SQL



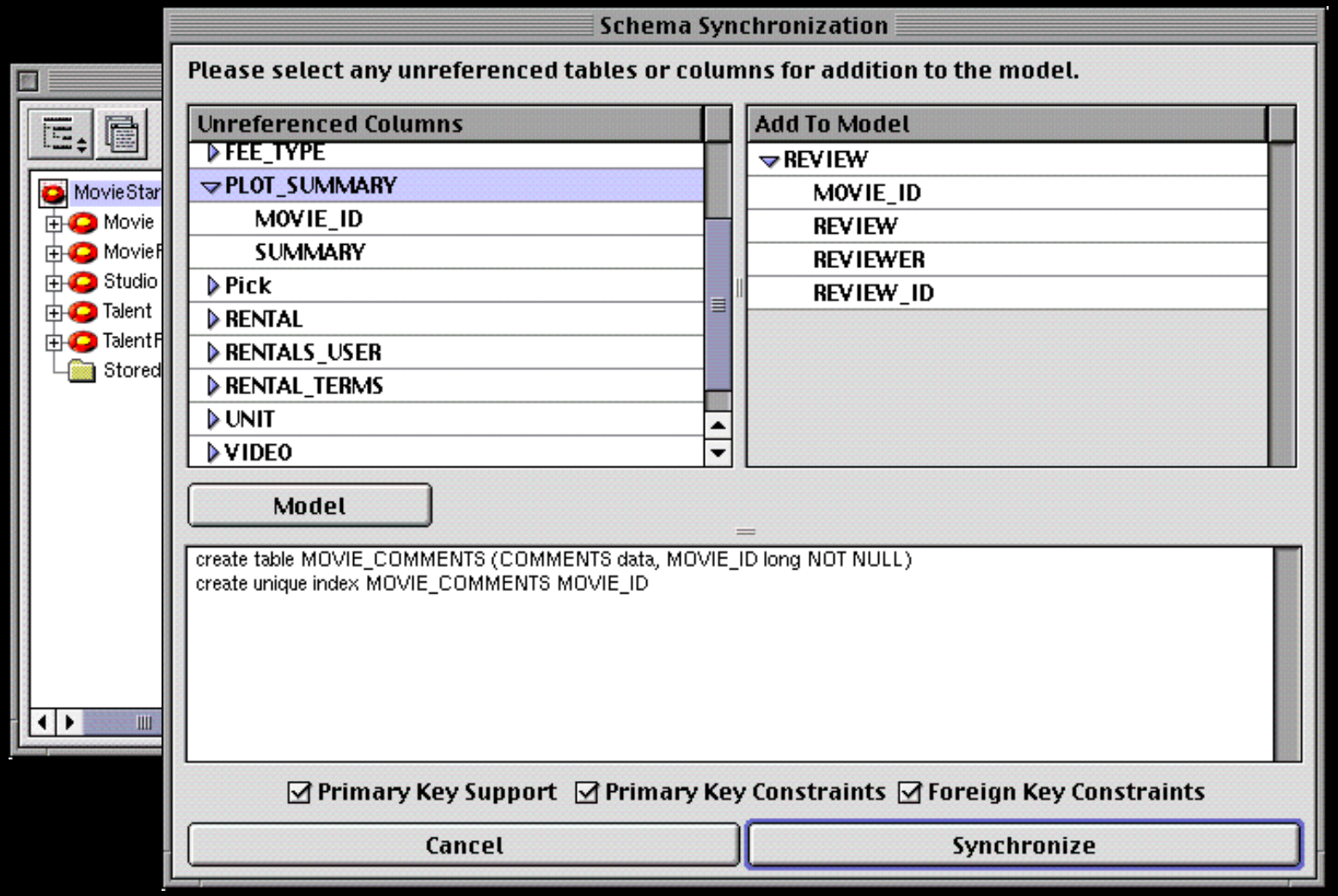
Browsing Data



Reverse Engineering



Synchronizing the Schema



Generating Java Classes

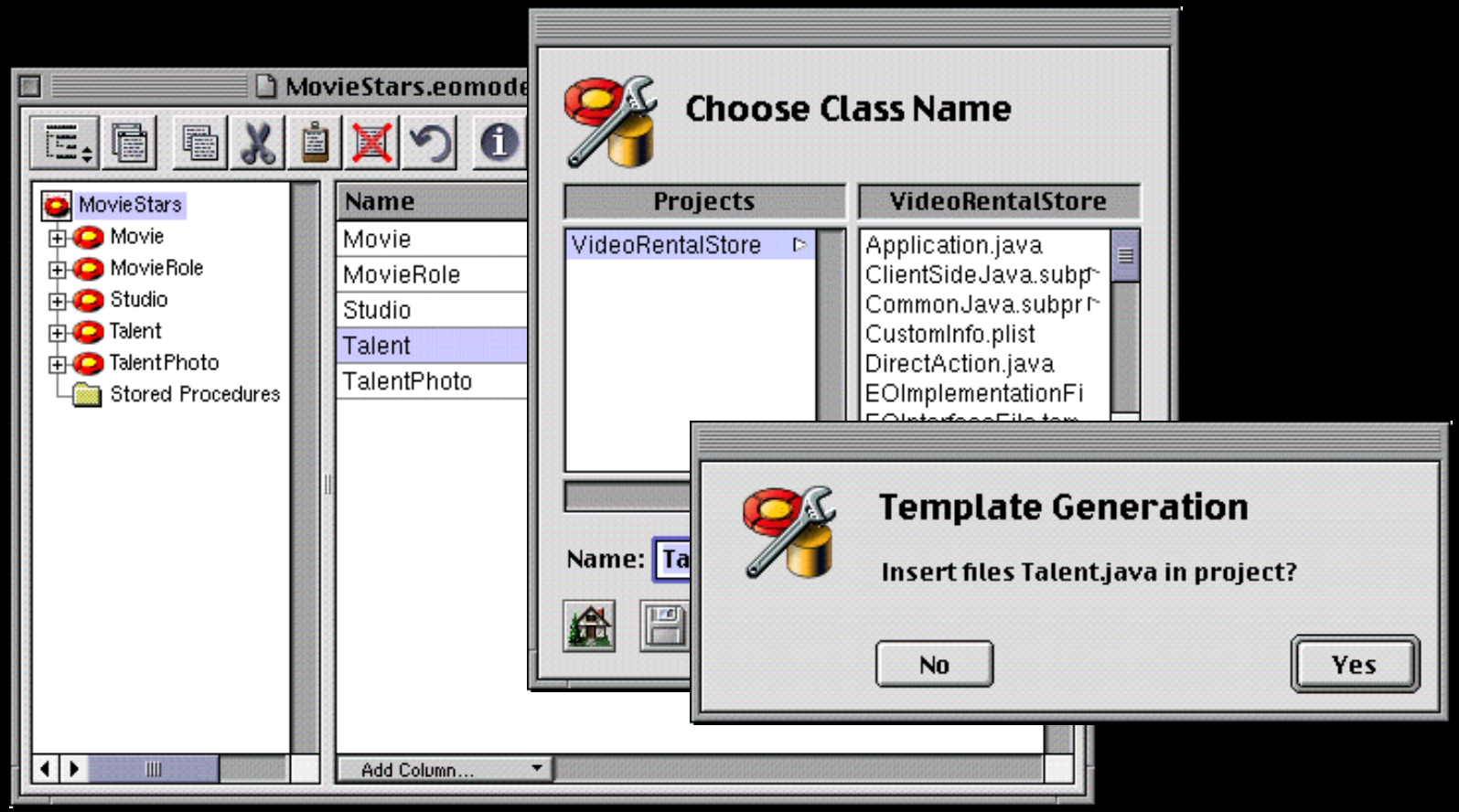
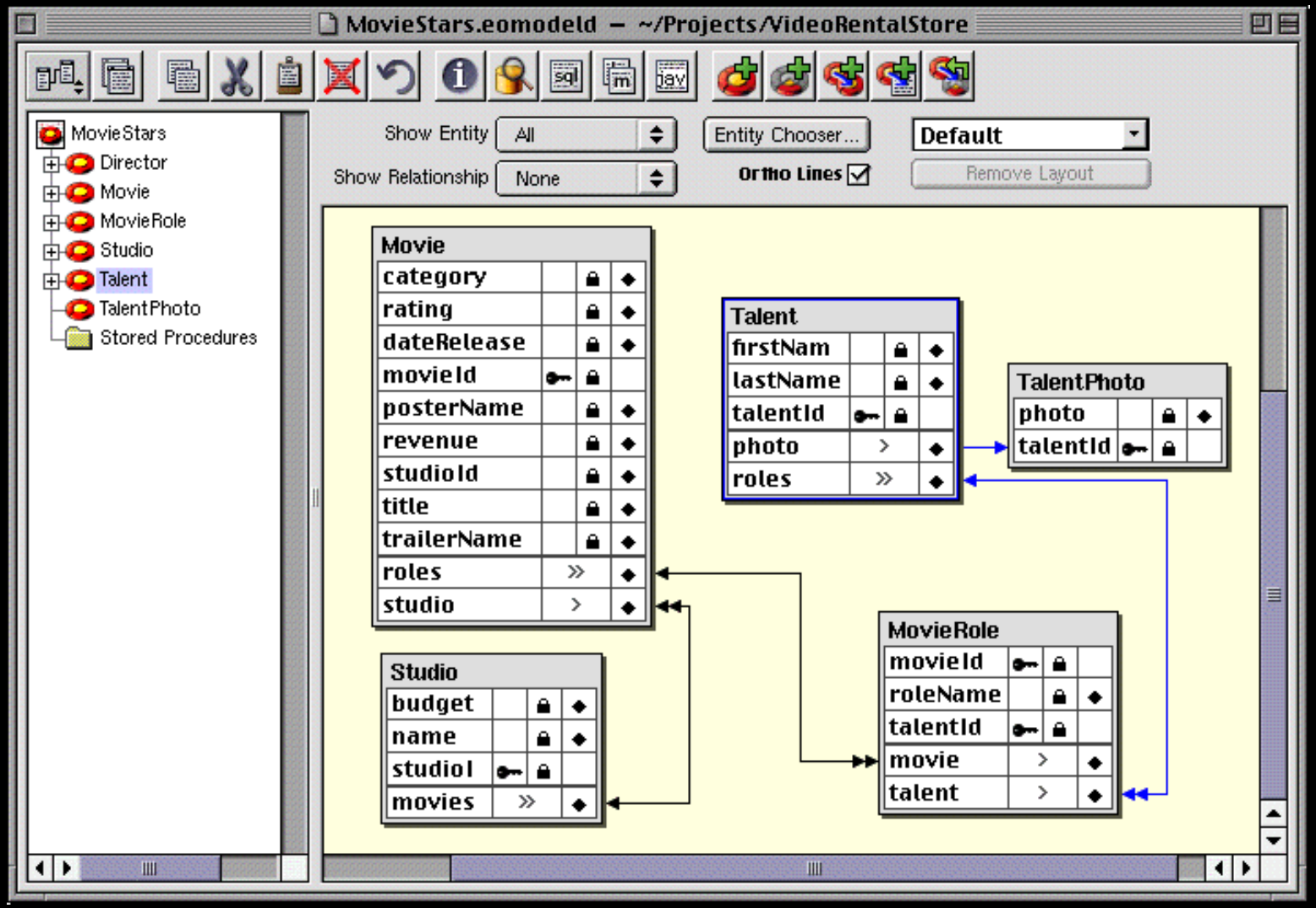


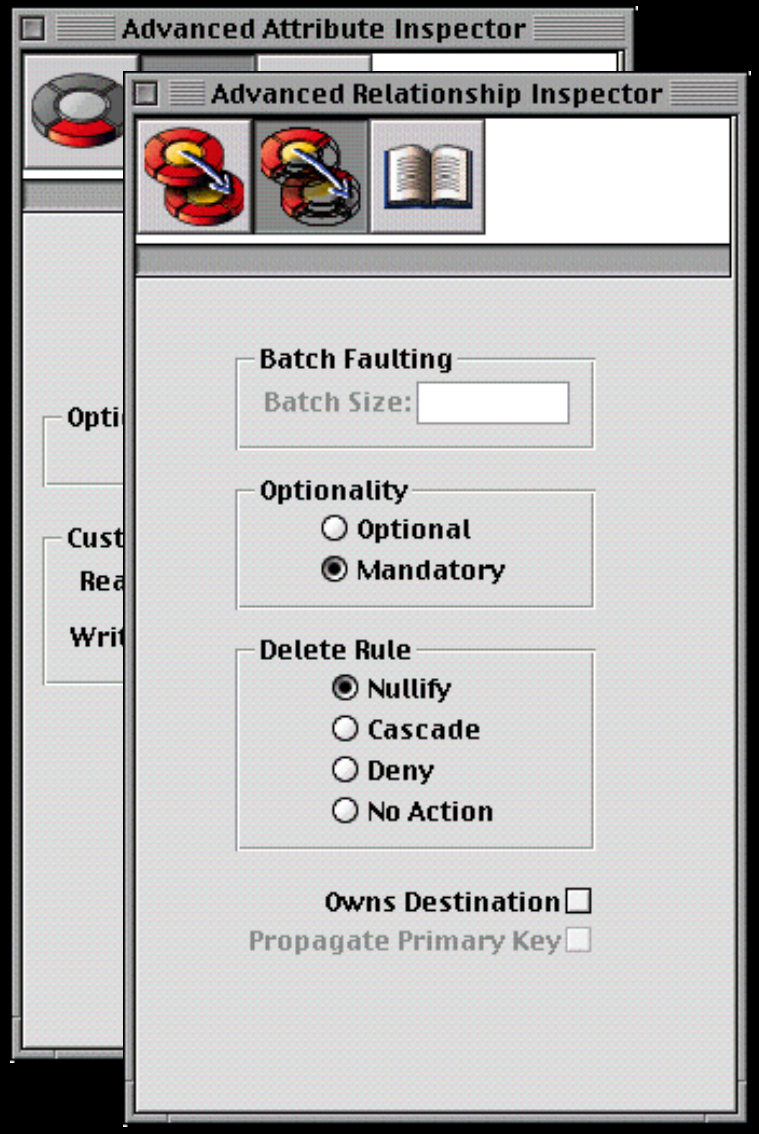
Diagram View



Part 3: Beyond the Schema

- Validation
- Locking and conflict detection
- Shared entities
- Stored procedures
- Fetch specifications

Validation in EOModeler



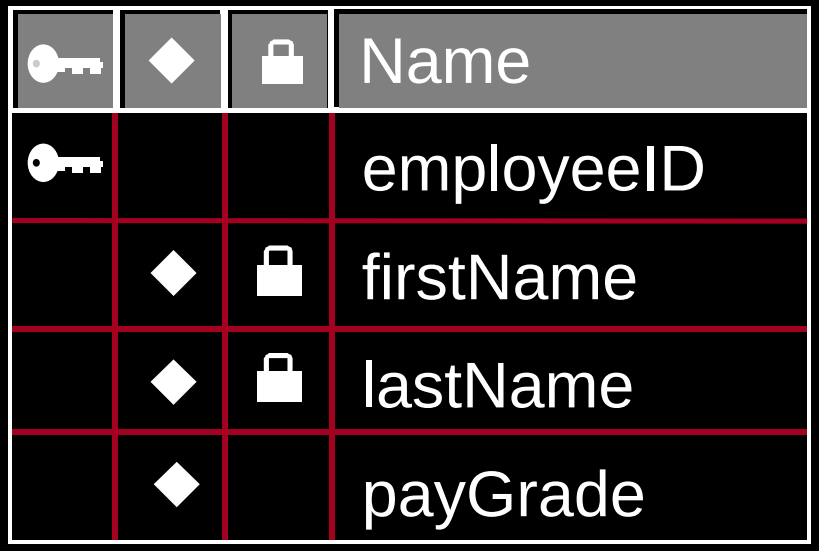
- Prohibit NULL values
- Limit string lengths
- Make relationships mandatory
- Specify whether deletions should cascade

Validation in Code

- Specify more complex validation rules:

```
public void validateForSave()
throws EOValidation.Exception
{
    if (age < 16 && hasDriversLicence)
        throw new EOValidation.Exception
            ("Too young to drive");
}
```

Locking and Conflict Detection

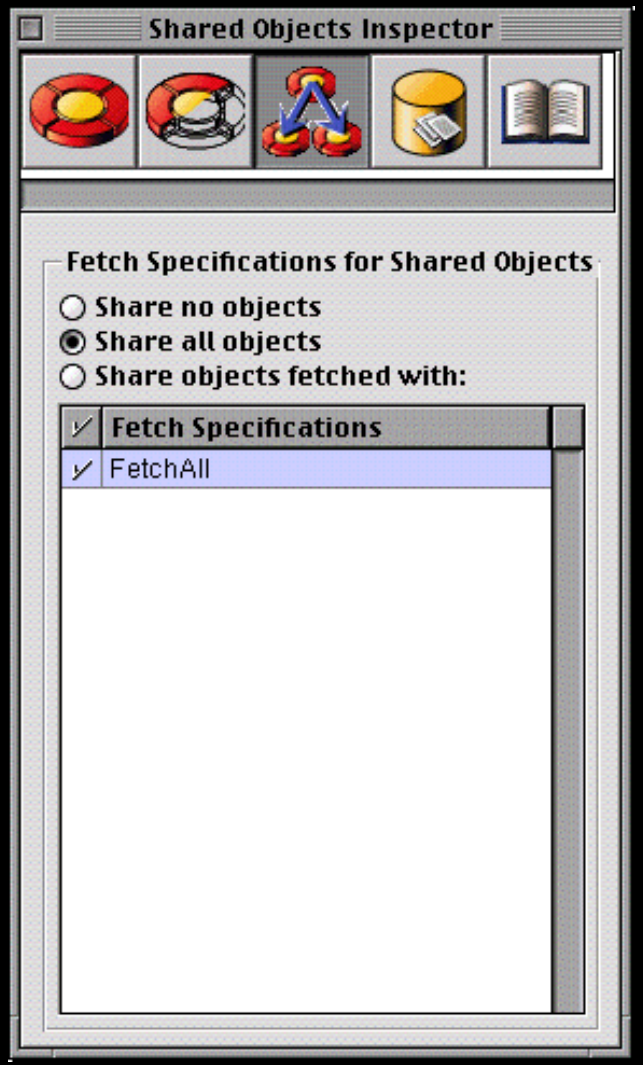


- Pessimistic locking blocks access to the database row
- Optimistic locking allows shared access

Catching Exceptions at Runtime

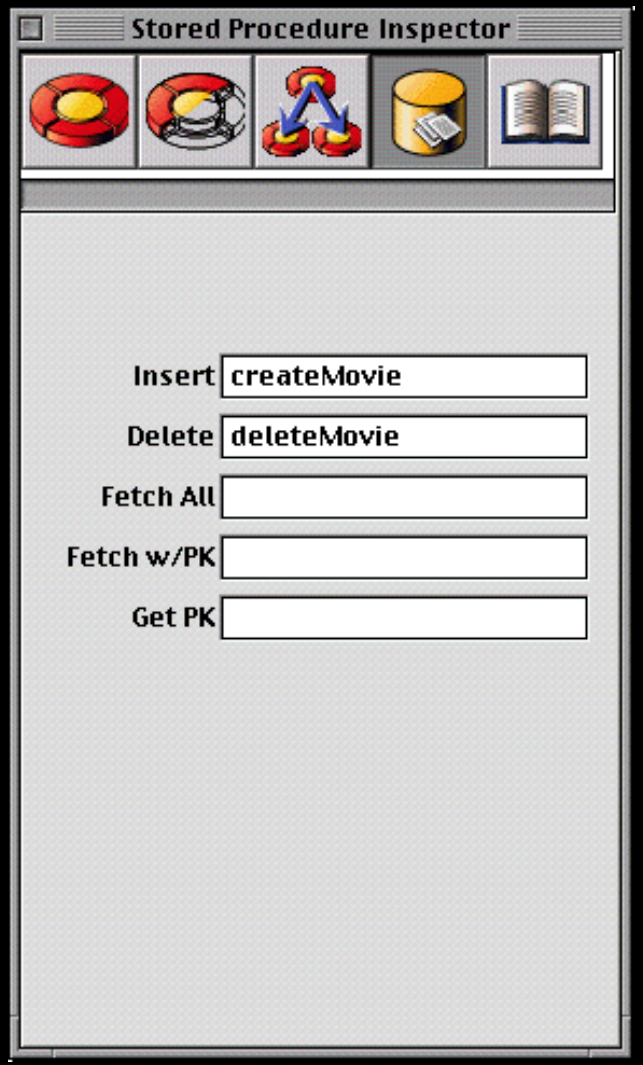
```
public WOComponent saveChanges()
{
    EOEditingContext ec =
        session().defaultEditingContext();
    try
        {
        ec.saveChanges();
        }
    catch (EOValidation.Exception e)
        {
        return createErrorPage(e);
        }
    return null;
}
```

Shared Objects



- Usually each user has their own copy of the objects
- This can be very memory-intensive
- Read-only entities can be safely shared

Stored Procedures

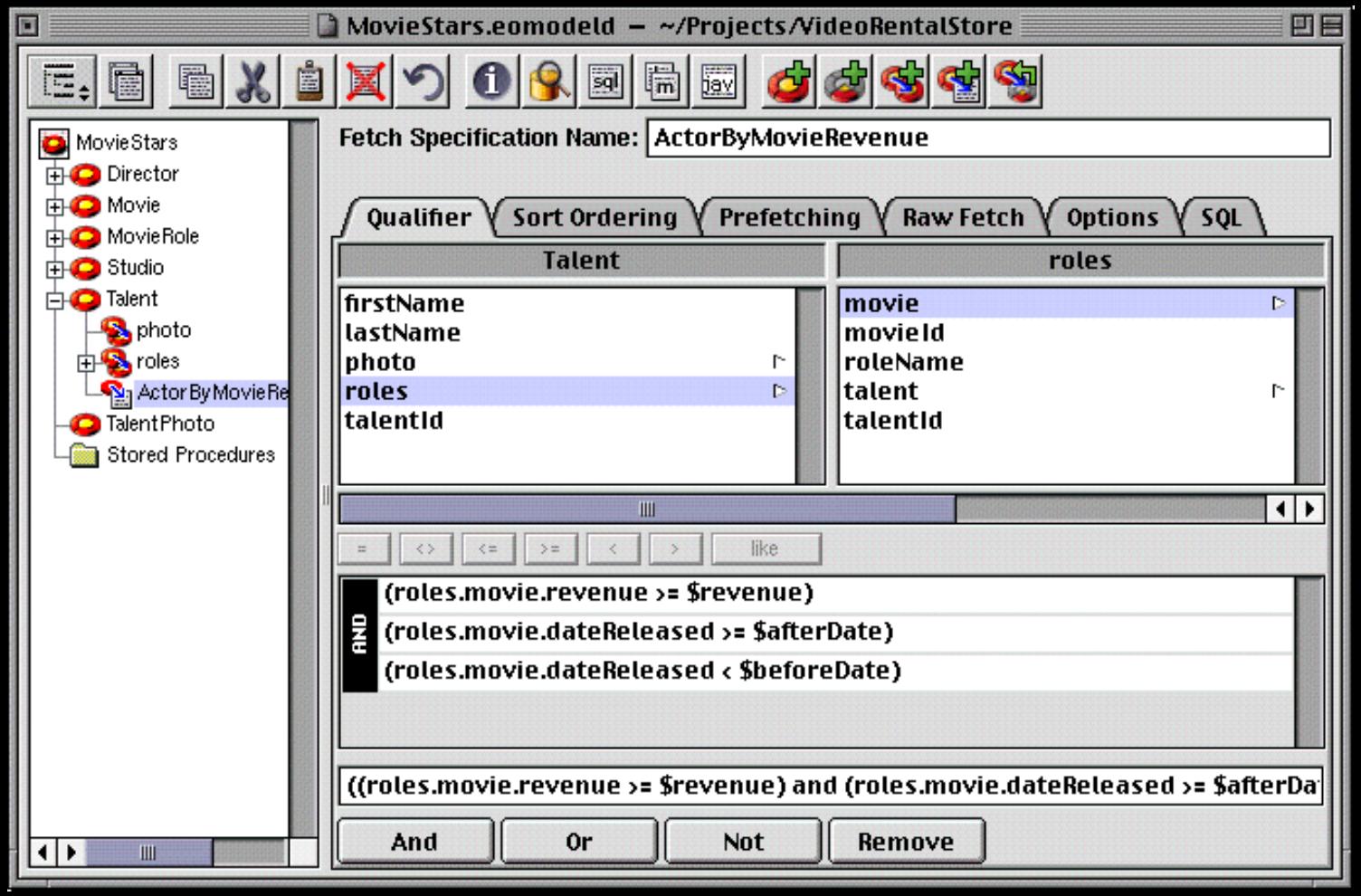


- Can be used to perform common operations:
  - Insert and delete
  - Fetch
  - Generate Primary Key
- You can also call arbitrary stored procedures

Fetch Specifications

- Build complex queries once in EOModeler
- Use variable placeholders in query
- Provide actual query values at runtime

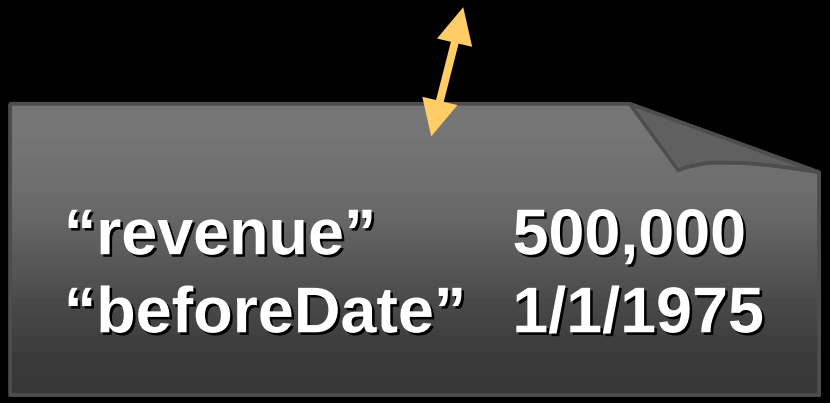
Fetch Specifications in EOModeler



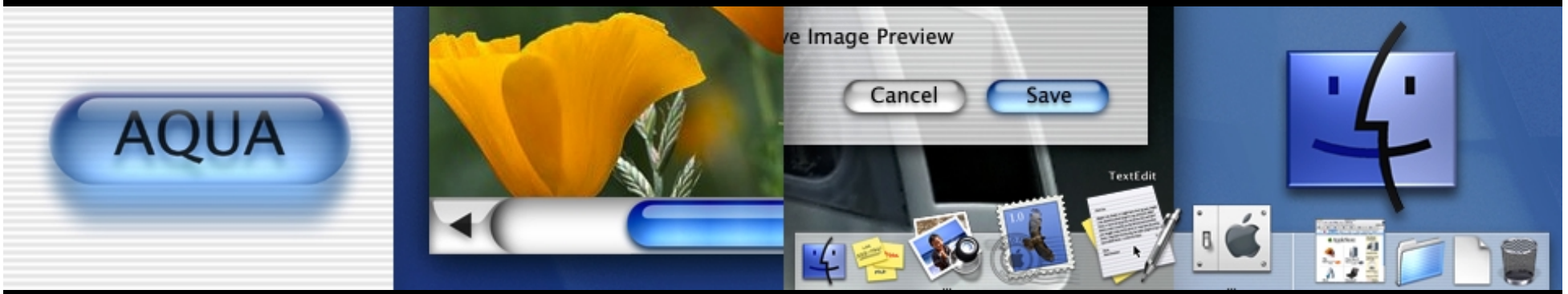
Using Fetch Specifications in Code

```
public NSArray performQuery()
{
    EOEditingContext ec = session().
        defaultEditingContext();
    NSArray results;

    results = EOUtilities.
         objectsWithFetchSpecificationAndBindings
        (ec, "Customers",
        "ActorByMovieRevenue", dictionary);
    return results;
}
```



DEMO



Mike Gobbi
Senior Developer Trainer, Apple iServices

Roadmap

407 WebObjects: EJB The wonderful fruit...	Room J2 **Wed., 2:00 p.m.**
410 WebObjects: Optimization Tuning your applications	Room J2 **Thur., 9:00 a.m.**
412 EOF Caching & Synchronization Client-Server done right	Room J2 **Thur., 2:00 p.m.**
415 Advanced EOF Everything you ever wanted to know...	Room J2 **Fri., 9:00 a.m.**
417 Building Large Applications Welcome to the real world	Room J2 **Fri., 2:00 p.m.**

For More Information

http://www.apple.com/webobjects

Visit the WebObjects lab downstairs!
Everyday from 11:00 a.m.–2:00 p.m.

Try out your WebObjects 4.5 Evaluation CD!

WebObjects Community BOF
Wed., 6:30 p.m.–8:00 p.m.

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
wofeedback@group.apple.com

Ernest Prabhakar
Product Line Manager, WebObjects
wofeedback@group.apple.com

Steve Miner
WebObjects Engineer

WWDC

Worldwide Developers Conference 2000

Think different.