

Java Persistence

What's New

What's (Still) The Same

Craig L. Russell
JAX India 2009

Agenda

Objects vs. Relational
Alternative Approaches
Object-Relational Mapping (ORM)
Some Cool ORM Features
Java Persistence API (JPA)
Java Data Objects (JDO)
Where Do We Go From Here?
Q & A

Presenter's "Baggage"

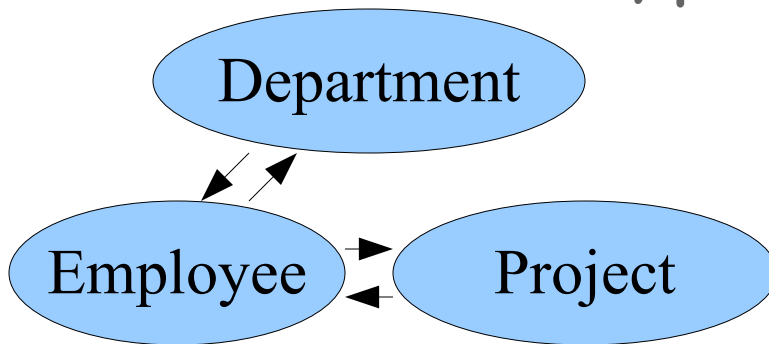
Java™ Data Objects Specification Lead

Apache OpenJPA PMC Chair

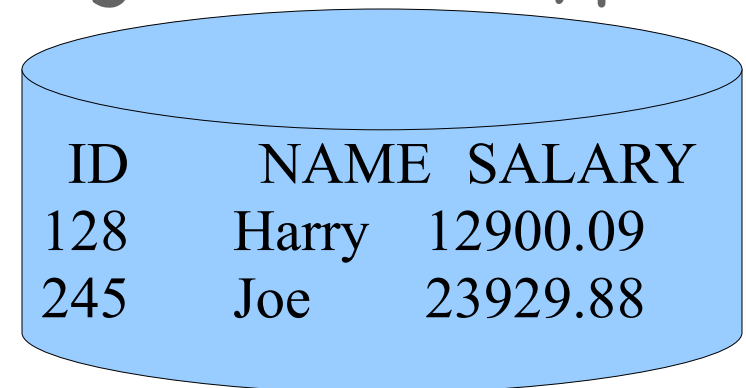
Java EE Container Managed Persistence Architect

Objects vs. Relational

- Classes
- Fields
- Methods
- References
- Multi-valued types



- Tables
- Columns
- Stored Procedures
- Foreign Keys
- Single-valued types



ID	NAME	SALARY
128	Harry	12900.09
245	Joe	23929.88

Alternative Approaches

- Direct API
 - Write SQL, Embed in Statement, Get ResultSet
- SQL Helper Classes
 - Write SQL, Put Results into User's Classes
- Data Access Object
 - Write SQL, Translate Results to Domain Objects
- Object-Relational Mapping
 - Write Mapping, Write Queries in Domain Model
 - Automatic Change Detection

Object-Relational Mapping

- Class \leftrightarrow Table
- Property \leftrightarrow Column
- Identity \leftrightarrow Primary Key
- Bidirectional-Relationship \leftrightarrow Foreign Key
- Inheritance \leftrightarrow Foreign Key == Primary Key
- HashMap \leftrightarrow Join Table with 3 columns
- Embedded \leftrightarrow Multiple Columns
- Class \leftrightarrow Multiple Tables with same Primary Key

Some Cool ORM Features

- Auto-increment fields/properties
 - Especially useful for primary keys
- Orphan deletion
 - Implements containment (composite) pattern
- Generate source from database schema
 - When you already have a database
- Surrogate (artificial) primary keys
 - Enforces "best practice" automatically

Some (More) Cool ORM Features

- Detached objects
 - Domain objects are DTOs (data transfer objects)
- Life cycle callbacks
 - Aspect-oriented without the pain
- Optimistic locking
 - Use version columns for consistency
- Uniquing
 - Guarantee one object per unique row

ORM "Potential Benefits" †

- Productivity
 - Writing Mapping is faster than writing SQL
- Scalability
 - Fewer team members need to be SQL experts
- Performance
 - ORM can write better SQL than you can
- Quality
 - Complex SQL needs extensive testing

† Legal made me say this

Popular ORM Solutions

- Java Persistence API Implementations
 - DataNucleus
 - OpenJPA (used in BEA, IBM app servers)
 - EclipseLink
 - Hibernate
- Java Data Objects Implementations
 - DataNucleus
 - Kodo (used in BEA app servers)

JPA 2.0 Features

- QueryBuilder API
 - Collections of Basic Types (e.g. String) †
 - Nested/Relationships Embedded Objects †
 - Delete Orphans †
 - Object Cache Interface †
 - Expanded Map Mapping (e.g. Join Table) †
 - Derived Identity (e.g. Master Detail) †
- † Included in JDO 2.2

JDO 2.1, 2.2, 2.3 Features

- JDOQL Subqueries
- Dynamic Fetch Groups
- Read-Only Transactions
- Transaction Isolation Level
- Persistence Manager Proxy
- `getServerTimeZone()`
- `getServerDate()`
- Enhancer Invocation API

Persistence Solutions †

- ORM Java 2,620,000 Google hits
- Java JPA 1,490,000 Google hits
- Java iBATIS 1,100,000 Google hits
- Hadoop 1,310,000 Google hits
- EclipseLink 523,000 Google hits
- Java Hibernate 361,000 Google hits
- Java JDO 112,000 Google hits

† take with a grain of salt

Where do we go (from here)?

- Object to Relational Impedance Mismatch
- Web Framework Object Persistence
- Relational vs. Object Database
- ORM Standards
- Cloud Computing, Massive Scale Data

Questions?

Answers!

Craig L. Russell
OpenJPA PMC Chair
ASF