ODBC and JDBC O/R Mapping

David Rabinowitz

Objects And Databases

- Most of the applications today are written using object oriented techniques and languages.
- Persistence is the key of saving data
- Many of today's applications use databases
  - Dynamic websites
  - Management applications (ERP, CRM, etc.)
  - Many many more…
- How do we easily combine the two?
  - Or: How do we easily persist objects?

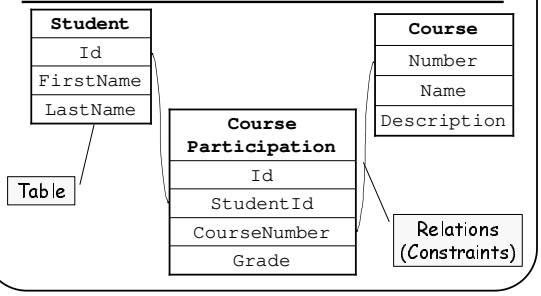
A Brief Introduction To Databases (1)

- We want to represent pieces of data
  - Same meta data
  - Student – first name, last name, …
  - Course – number, name, …
- We want to relate between data
  - Which courses does a student take?

A Brief Introduction To Databases (2)

- We define our data as *entities*
- We define their connections as *relationships*
- And then define the *Entity-Relationship Diagram*
- More on that in the databases course
  - Very recommended!

Example – The Relations



Example – The Tables (1)

Students

Id	FirstName	LastName
111111111	Bart	Simpson
222222222	Lisa	Simpson
333333333	Milhouse	Van Houten

Courses

Number	Name	Description
67615	ood	Object Oriented Design
67506	db	Databases
67808	os	Operating Systems

Example - The Tables (2)

Id	StudentId	CourseNumber	Grade
2537	111111111	67506	95
2538	111111111	67615	90
2539	222222222	67615	97
2560	333333333	67808	88

- How do we use all this data ?

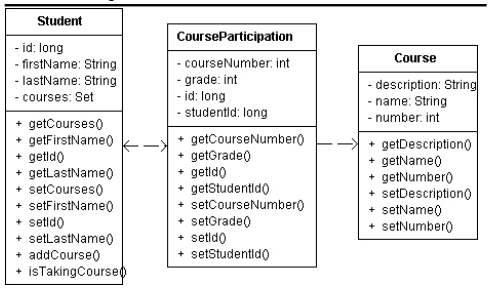
CRUD Operations (I)

- Create
  - **INSERT INTO Students (Id, FirstName, LastName) VALUES (55, "Maggie", "Simpson")**
- Read (or Retrieve)
  - **SELECT FirstName,LastName FROM Studnets WHERE FirstName="Maggie"**

CRUD Operations (II)

- Update
  - **UPDATE Students SET FirstName="Mag" WHERE Id=55**
- Delete
  - **DELETE FROM Students WHERE FirstName="Mag"**

The Objects



Database Connection

- SQL is an embedded language
  - Inserted as part of the code
- How to connect to the database?
  - Special libraries
  - Special network protocol
  - General interface libraries - ODBC, JDBC, etc.

ODBC

- Open Database Connectivity
- Access any data from any application
  - No matter what the database is!
  - Also, the data can be in Excel spreadsheets or text files.
- The basic idea - introduce a general database API to the application
  - Each database has different implementation

ODBC API

- Driver
  - The basic hook to the system
- Connection
  - Holds the connection to the database
- RecordSet
  - Wraps a result of database query
- Command
  - Used to send commands (insert, delete) to the database

Example

```
Set conn = new ADODB.Connection
conn.Open "<database-name>"
Set rs = new ADODB.RecordSet
rs.Open "SELECT * From Students", _
  conn
For Each f in rs.Fields
  Response.Write "<th>" & f.Name & _
     "</th>"
Next
```


Example (II)

```
While Not rs.EOF
  Response.Write "<tr>"
  For Each f in rs.Fields
     Response.Write "<td>"
     Response.Write rs(f.Name)
     Response.Write "</td>"
  Next
  Response.Write "</tr>"
  rs.MoveNext
Wend
```


JDBC

- Java Database Connectivity
- From the first version of Java
- Was modeled on ODBC
- Driver, Connection and ResultSet
- Lack in ODBC is Statement
  - PreparedStatment, CallableStatment
- Extensive use of the factory pattern

CRUD Problems

- Has to be written and maintained for each object
- Tedious and error prone work
- Most of the code is similar
  - Only table name and fields change

O/R Mapping Tools

- Specifications
  - EJB CMP (Container Managed Persistence)
  - JDO
- Commercial Products
  - TopLink
- Open Source Tools
  - Hibernate
  - OJB
- Many many more …

HIBERNATE

- An open source project
  - Created by Gavin King
  - Now belongs to the JBoss group
  - http://www.hibernate.org/

How does hibernate works?

```
// 1. Build a Product
Course c = new Course(67615, "ood",
                      "Object Oriented Course");
// 2. Fire up Hibernate
Configuration cfg = new Configuration();
cfg.addClass(Course.class);
SessionFactory sf = cfg.buildSessionFactory();
// 3. Open Session
Session session = sf.openSession();
// 4. Save Product and close Session
Transaction tx = sess.beginTransaction();
session.save(c);
tx.commit();
session.close();
```


Behind The Curtains

- How does Hibernates knows how to relate the class **Product** to the table?
- XML definition file for each class
  - Defines its mapping to the table in the database
  - Defines its relations to other classes

Course.hbm.xml

```
<hibernate-mapping>
    <class name="tir11.Course" table="Courses">
        <id name="id" type="int" unsaved-value="-1" >
            <column name="id" sql-type="integer"/>
            <generator class="my.generator"/>
        </id>
        <property name="name">
            <column name="name"
                    sql-type="char(255)" not-null="true"/>
        </property>
        <property name="description">
            <column name=" description"
                    sql-type="char(255)" not-null="true"/>
        </property>
    </class>
</hibernate-mapping>
```


HQL - Hibernate Query Language

- We are working on objects, not tables
  - SQL cannot be used
- HQL is similar to SQL, but works on objects

HQL Example

```
Query q =
  session.createQuery("select
  student from student in class
  tir11.Student where
  student.name=:name");
q.setString("name", "maggie");
List l = q.list();
Student maggie = l.get(0);
```

Updating & Deleting

```
//update
maggie.setName("mag");
session.saveOrUpdate(maggie);

//delete
Session.delete(bart);
Tx.commit();
```


Relations

- How to declare the relation between **Student** and **CourseParticipation**?
- Student.hbm.xml

```
<hibernate-mapping>
  <class name="tir11.Student" table="Students">
    <set name="courses" inverse="true"
         table="CourseParticipations">
      <key column="id" />
      <one-to-many class=
                  "tir11.CourseParticipation"/>
    </set>
  </class>
</hibernate-mapping>
```


And on the other side

```
<hibernate-mapping>
  <class name="tir11.CourseParticipation"
         table="CourseParticipations">
    <property ...>
    <many-to-one name="student"
                 class="tir11.Student"
                 column="StudentId" />
    <many-to-one name="course"
                 class="tir11.Course"
                 column="CourseNumber"/>
  </class>
</hibernate-mapping>
```


Example

```
Course ood = ...;
Course db = ...;
Student bart = new Student(1111111,
  "Bart", "Simpson");
bart.addCourse(ood);
bart.addCourse(db);
tx.begin();
session.save(bart);
tx.commit()
Session.close();
```


How can that happen?

- Reflection
- Bytecode manipulation

Objects & Databases

- Two widely used technologies
- … that not always cooperate easily
- Direct CRUD operations are expensive to write (test!) and maintain
- O/R mapping tools can ease the development