Entity Beans

B.Ramamurthy

Entity Bean

- Persistent
- Corresponds to a relation/table in a relational database
- Each instance of the bean corresponds to a row in the relational table
- Database entry possible created by SQL:

```
CREATE TABLE savingsaccount
(id VARCHAR(3) CONSTRAINT pk_savingsaccount
   PRIMARY KEY, firstname VARCHAR(24), lastname
   VARCHAR(24), balanace NUMERIC(10,2));
```

Savings Account Code

- Entity bean class: SavingsAccountBean
- Home interface: SavingsAccountHome
- Remote interface: SavingsAccount
- Utility class InSufficientBalanceException
- Client: SavingAccountClient

Entity Bean Class

- Implements:
- Entity bean interface
- Zero or more ejbCreate and ejbPostCreate methods
- Finder methods
- Business methods
- Home methods

EJBCreate Method

- Inserts the entity into the database
- Initializes the instance variables
- Returns primary key

- Lets look at the code.
- Access control is public, return type is primary key, arguments are legal Java RMI types, method modifier cannot be final or static.

EJBremove

- Client deletes the entity bean by invoking this method. Deletes the entity from the state of the database.
- Equivalent to deleting the row from the relational table.

EJBLoad and EJBStore

- EJB container uses these methods to synchronize the instance variables of an entity bean with the corresponding values stored in the database,
- ejbLoad method refreshes the instance variables, and the ejbStore method writes the variables to the database.

Finder methods

- Finder methods allow clients to locate entity beans.
- The SavingAccountClient program locates entity bean with three finder methods:

```
SavingsAccount jones =
    home.findByPrimaryKey("836");
Collection c = home.findByLastName("Smith");
Collection c = home.findInRange(20.0, 99.0);
```

- For every finder method available to client, the entity bean must implement a corresponding method that begins with a prefix ejbFind. Lets check the code.

Business Methods

- The business methods contain the business logic that you want to encapsulate within the entity bean.
- Usually business methods do not access the database, allowing you to separate the business logic from the database access code.
- Lets examine the methods.

Home methods

- A home method contains the business logic that applies to all entity beans of a particular class.
- In contrast logic of a business that applies to a single entity bean, an instance with unique identity.
- Consequently home method must not access the bean's persistence state or in other words instance variables.
- Typically a home method locates a collection of bean instances and invokes business methods as it iterates through the collection.
- Charge for low balance is an example.

SQL statements in SavingsAccountBean : DB calls

Method	SQL Statement
ejbCreate	INSERT
ejbFindPrimaryKey	SELECT
ejbFindByLastName	SELECT
ejbFindInRange	SELECT
ejbLoad	SELECT
ejbRemove	DELETE
ejbStore	UPDATE

Finally,

- Examine the home interface, remote interface code.
- Open, deploy and run the example.
- You may have to set up the database, in this case cloudscape.
- Later I will show you hot to access our Oracle db with the help of the j2eeadmin tool.