Understanding and Designing with EJB

B.Ramamurthy

Based on j2eetutorial documentation.

http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html

Review

- Request/Response Model
- Distributed Objects: stubs and skeleton providing location transparency
- Naming and Lookup: registry and binding
- Server-side technology: servlets
- Web applications: can be written entirely using Java Server Pages (static and dynamic content and data access can be provided); JSP is wrapper on servlet technology.
- Concept of initial context:The starting point for resolution of names for naming and directory operations.
- Data base access: using Java Data Base Connectivity

When to use EJB

- For large scale applications: where resources and data are distributed.
- When the application is run on servers at many locations.
- Where scalability is critical.
- Where transactions are required to ensure data integrity
- When a variety of clients need to handled.

Types of Enterprise Bean: Session

- **Session bean**: represents a single client inside the J2EE server. Session represents an interactive session. When a client terminates the session bean terminates/is no longer associated with the client.
- Stateful session bean: maintains a conversational state for the duration of a session. Ex: items reviewed in a session at some sites
- Stateless session bean: does not maintain a conversational state. Ex: computing a formula for a given value

Types of Enterprise Bean: Entity

- An entity bean represents a business object in a persistent storage mechanism. Ex: customers, orders, and products.
- Each entity bean typically has an underlying table in a relational database (business data), and each instance of the bean corresponds to a row in that table.
- Transactional and recoverable on a server crash.

Types of Enterprise Bean: Message-Driven

- A message driven bean is an enterprise bean that allows J2EE applications to process messages asynchronously.
- It acts as a JMS listener, which is similar to an event listener except that it receives messages instead of events.
- The messages can be sent by any J2EE component: an application client, another enterprise bean, or a web component, or a non-J2EE system using JMS.
- Retain no data or conversational state.

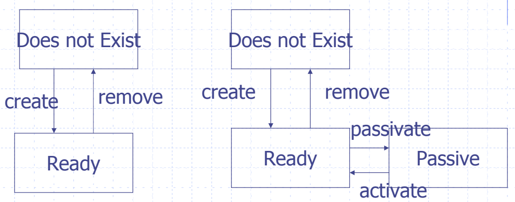
Contents of an Enterprise Bean

- Interfaces: The remote and home interface for remote access. Local and local home accesses for local access.
- Enterprise bean class: Implements the methods defined in the above interfaces.
- Deployment descriptor: An XML file that specifies information about the bean such as its type, transaction attributes, etc.
- Helper classes: non-bean classes needed by the enterprise bean class such as utility and exception classes.

The life cycles of enterprise beans

- An enterprise bean goes through various stages during its lifetime. Each type has different life cycle.

Session bean



Does not Exist

create

remove

Ready

Does not Exist

create

remove

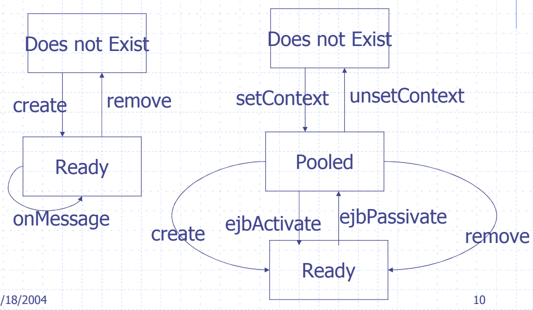
passivate

Ready

Passive

activate

Entity and Message-driven Bean Lifecycle



Does not Exist

create

remove

Ready

onMessage

Does not Exist

setContext

unsetContext

Pooled

create

ejbActivate

ejbPassivate

remove

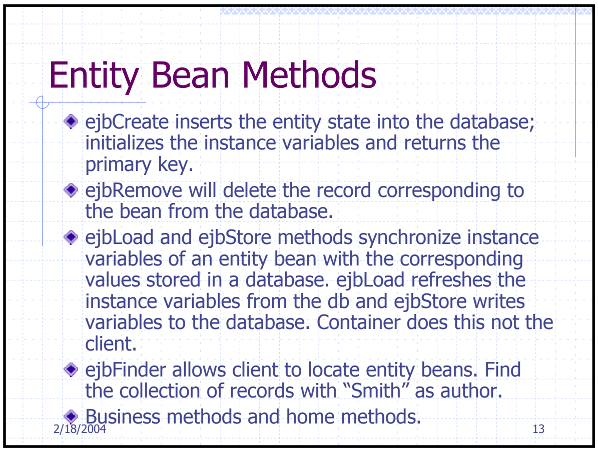
Ready

Entity Bean

- Data is at the heart of most business applications.
- In J2EE applications, entity beans represent the business objects that need persistence (need to be stored in a database.)
- You have choice of bean-managed persistence (BMP) and container-managed persistence (CMP).
- In BMP you write the code for database access calls. This may be additional responsibility but it gives control to the bean developer.

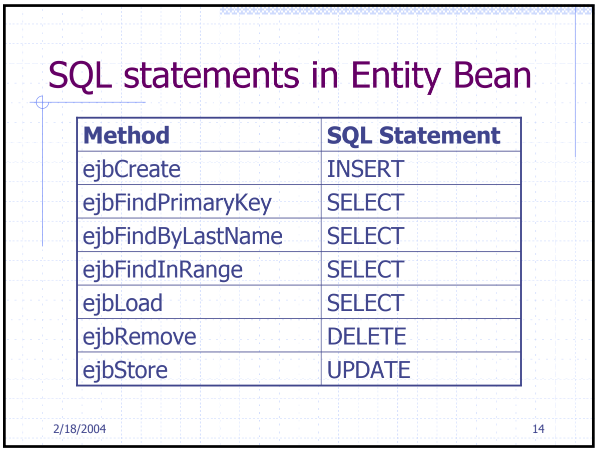
Entity Bean class

- Implements EntityBean interface
- Zero or more ejbCreate and ejbPostCreate methods
- Finder methods
- Business methods
- Home methods



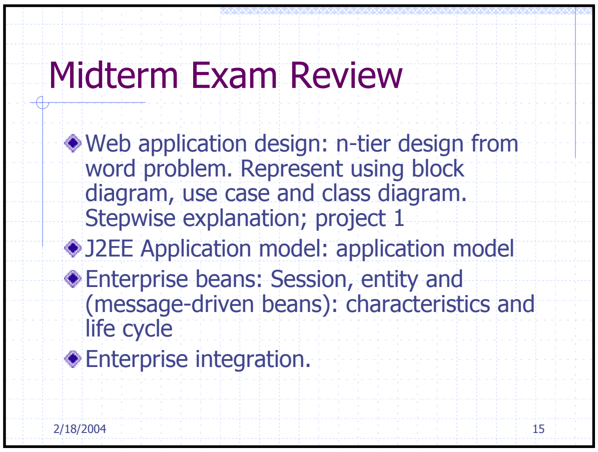
Entity Bean Methods

- ejbCreate inserts the entity state into the database; initializes the instance variables and returns the primary key.
- ejbRemove will delete the record corresponding to the bean from the database.
- ejbLoad and ejbStore methods synchronize instance variables of an entity bean with the corresponding values stored in a database. ejbLoad refreshes the instance variables from the db and ejbStore writes variables to the database. Container does this not the client.
- ejbFinder allows client to locate entity beans. Find the collection of records with "Smith" as author.
- Business methods and home methods.



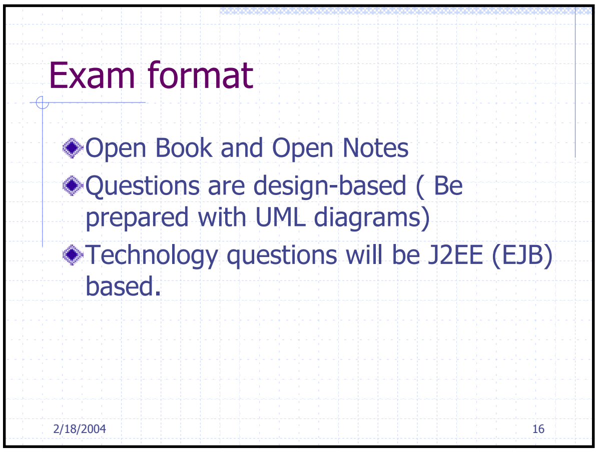
SQL statements in Entity Bean

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



Midterm Exam Review

- Web application design: n-tier design from word problem. Represent using block diagram, use case and class diagram. Stepwise explanation; project 1
- J2EE Application model: application model
- Enterprise beans: Session, entity and (message-driven beans): characteristics and life cycle
- Enterprise integration.



Exam format

- Open Book and Open Notes
- Questions are design-based (Be prepared with UML diagrams)
- Technology questions will be J2EE (EJB) based.