

Object-Oriented Analysis and Design

Appendix: ObjectStore Mechanism

Identify Design Mechanisms Slides

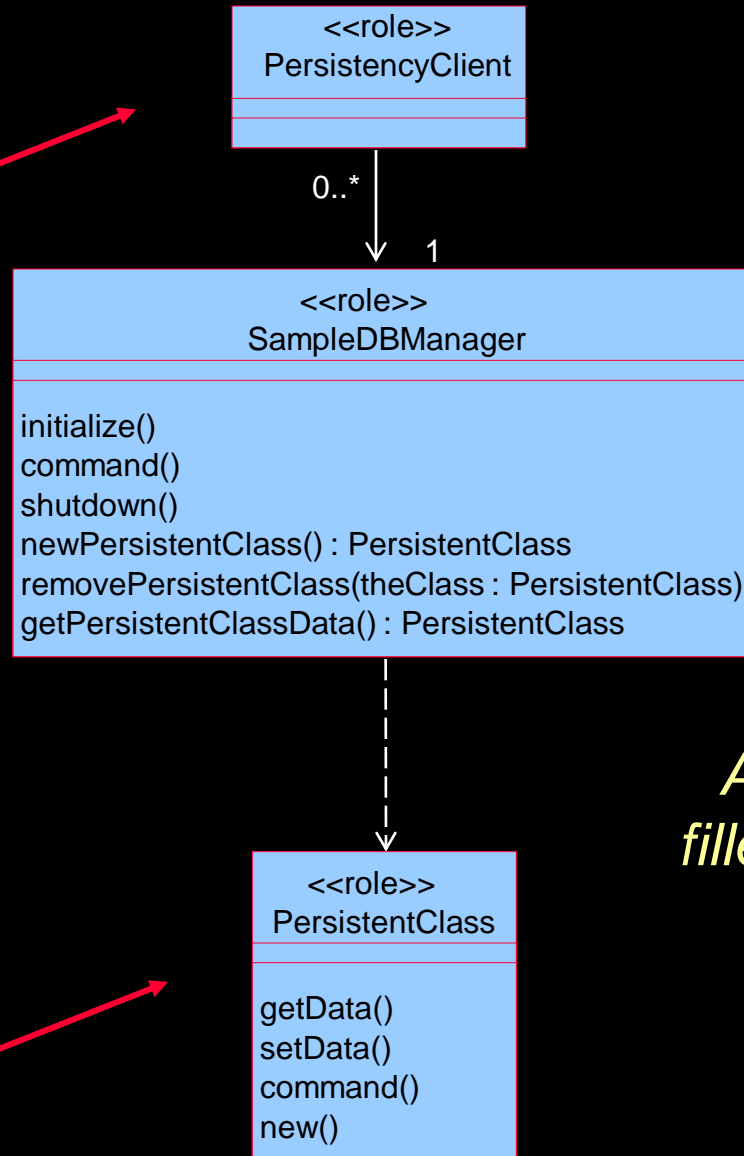
The following slides can be inserted during the Identify Design Mechanisms module

Example: Persistency: OODBMS: ObjectStore

*Example of a
persistent class client*

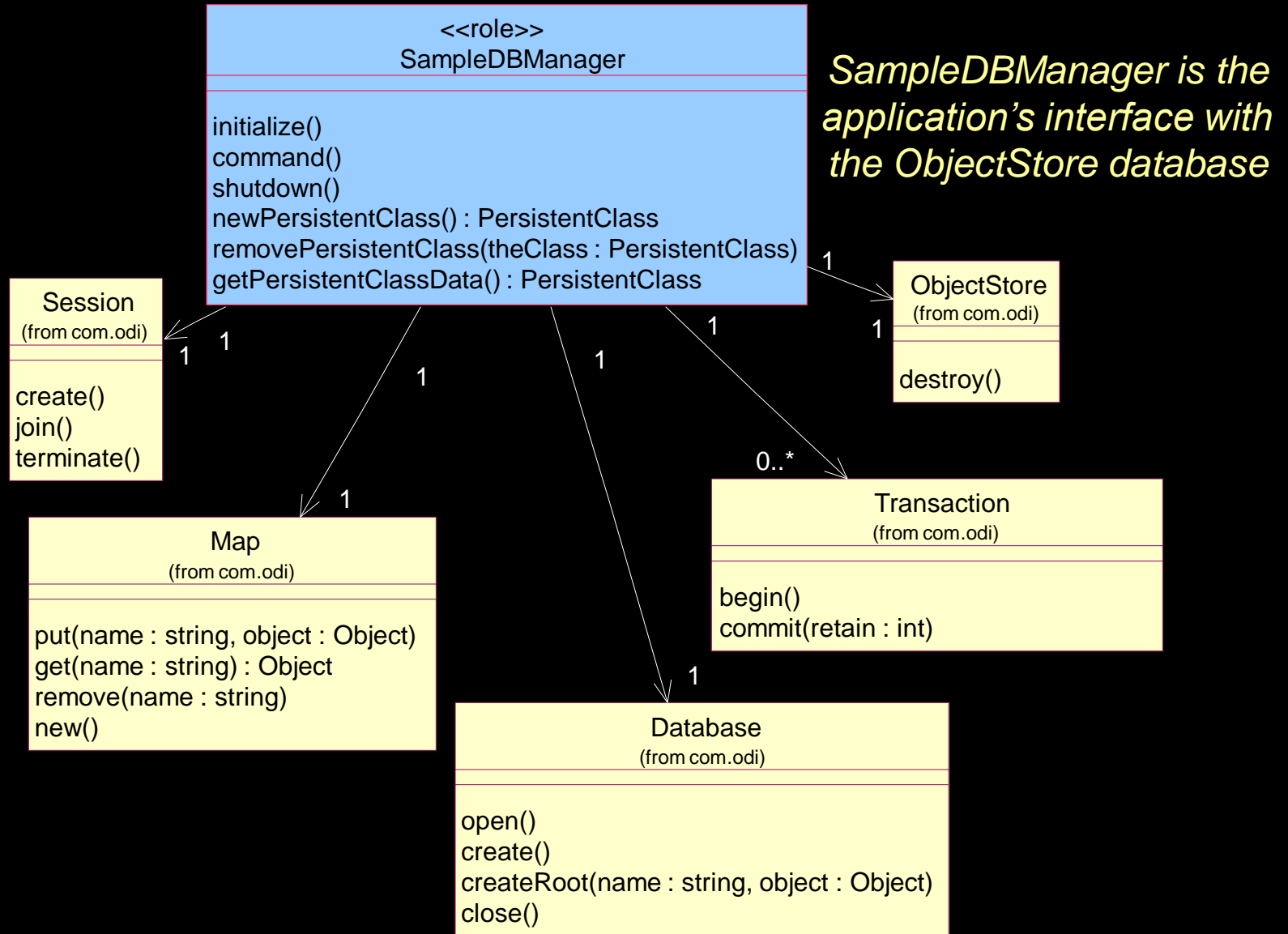
*Create to support
persistency*

*Example of a
persistent “root” class*

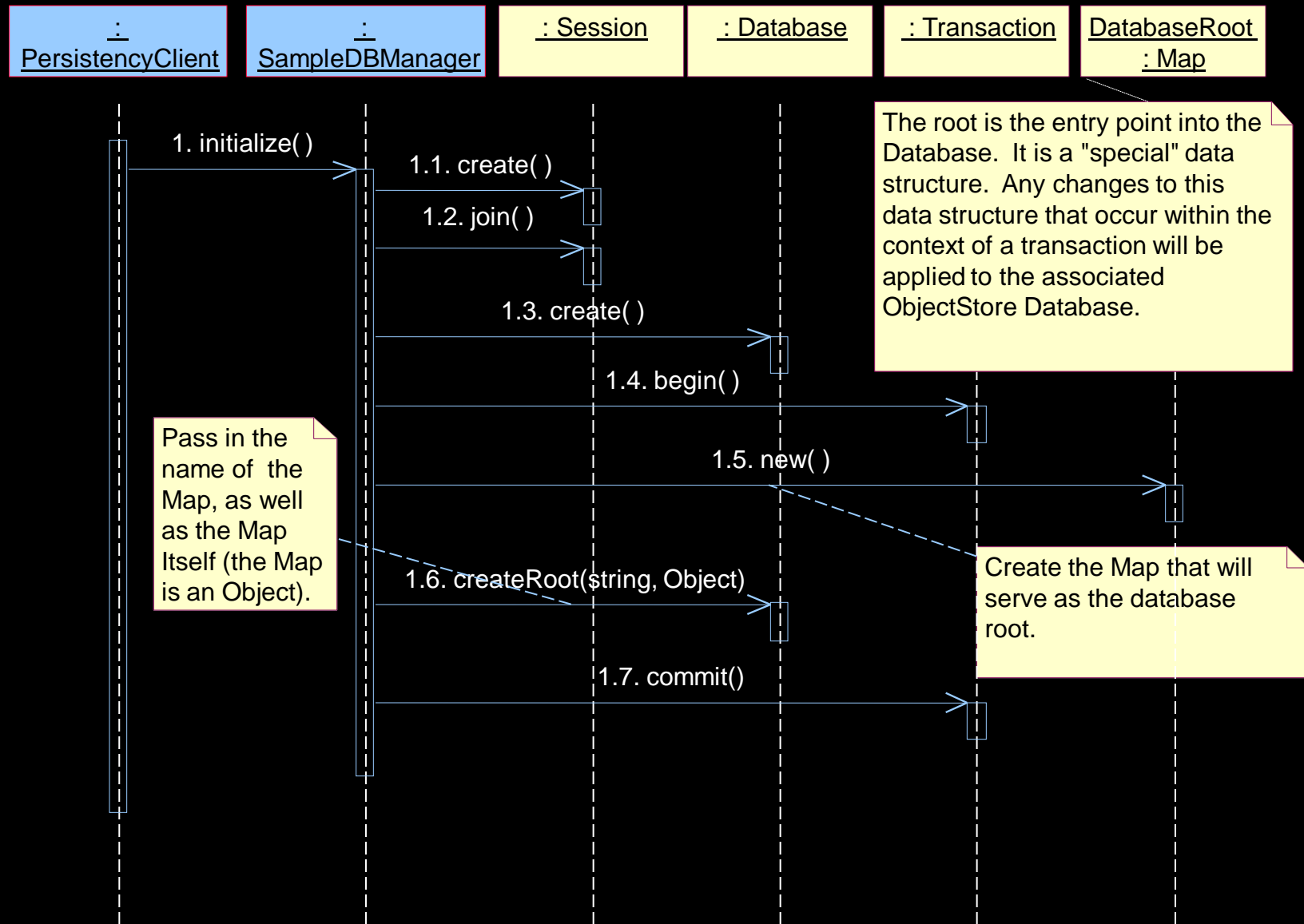


*All are roles to be
filled by the designer
applying the
mechanism*

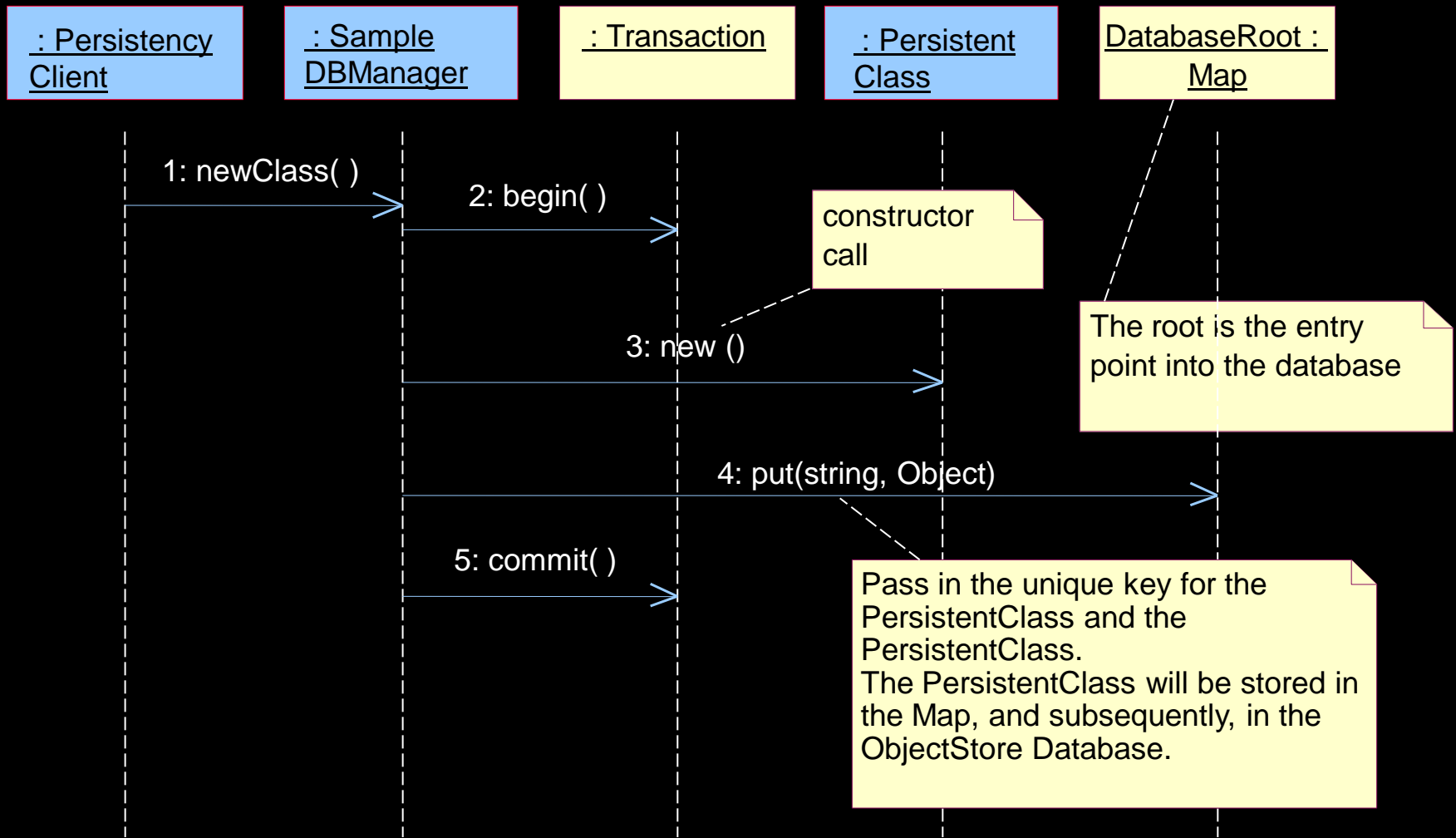
Example: Persistency: OODBMS: ObjectStore: SampleDBManager



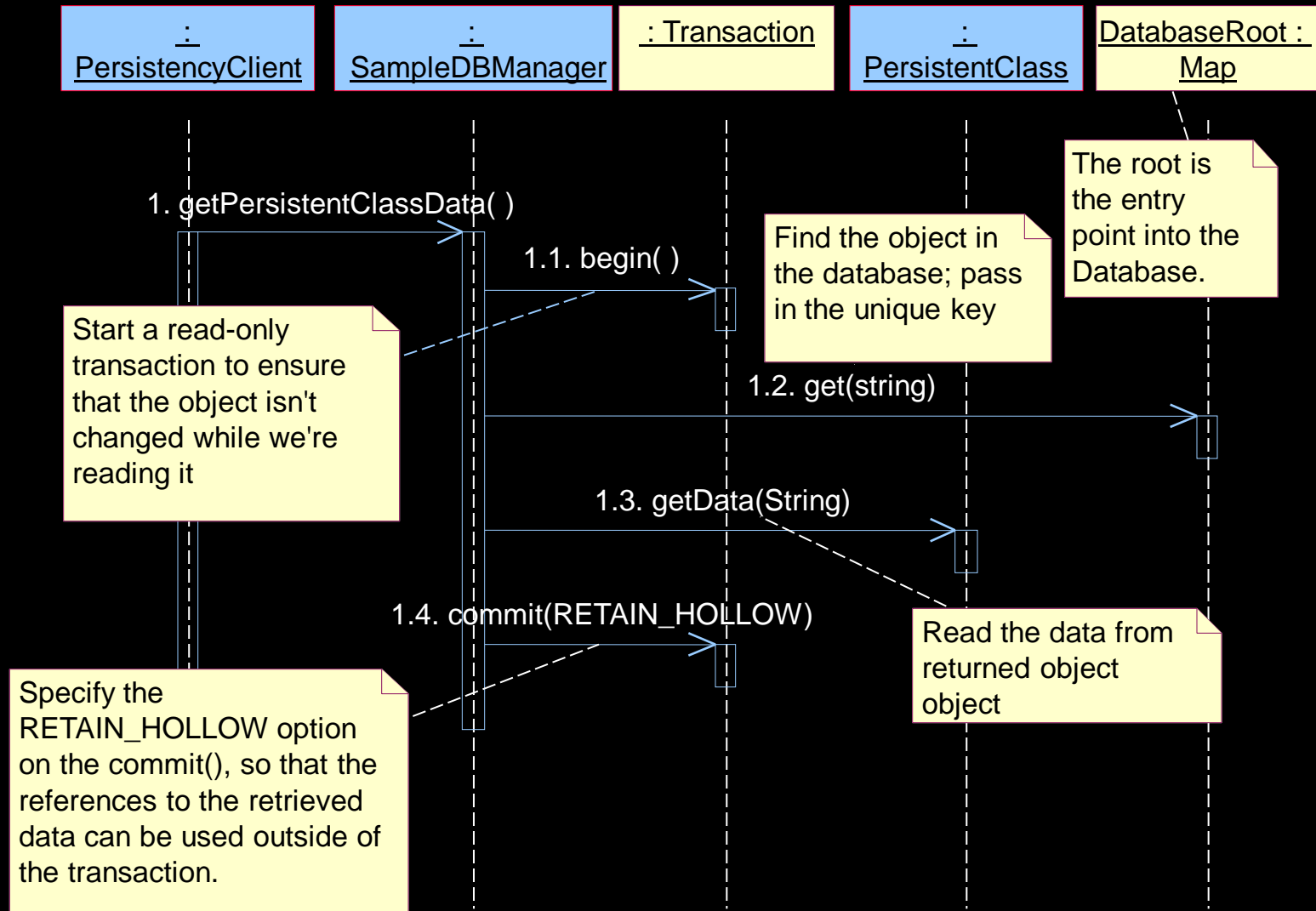
Example: Persistency: OODBMS: ObjectStore: Initialize



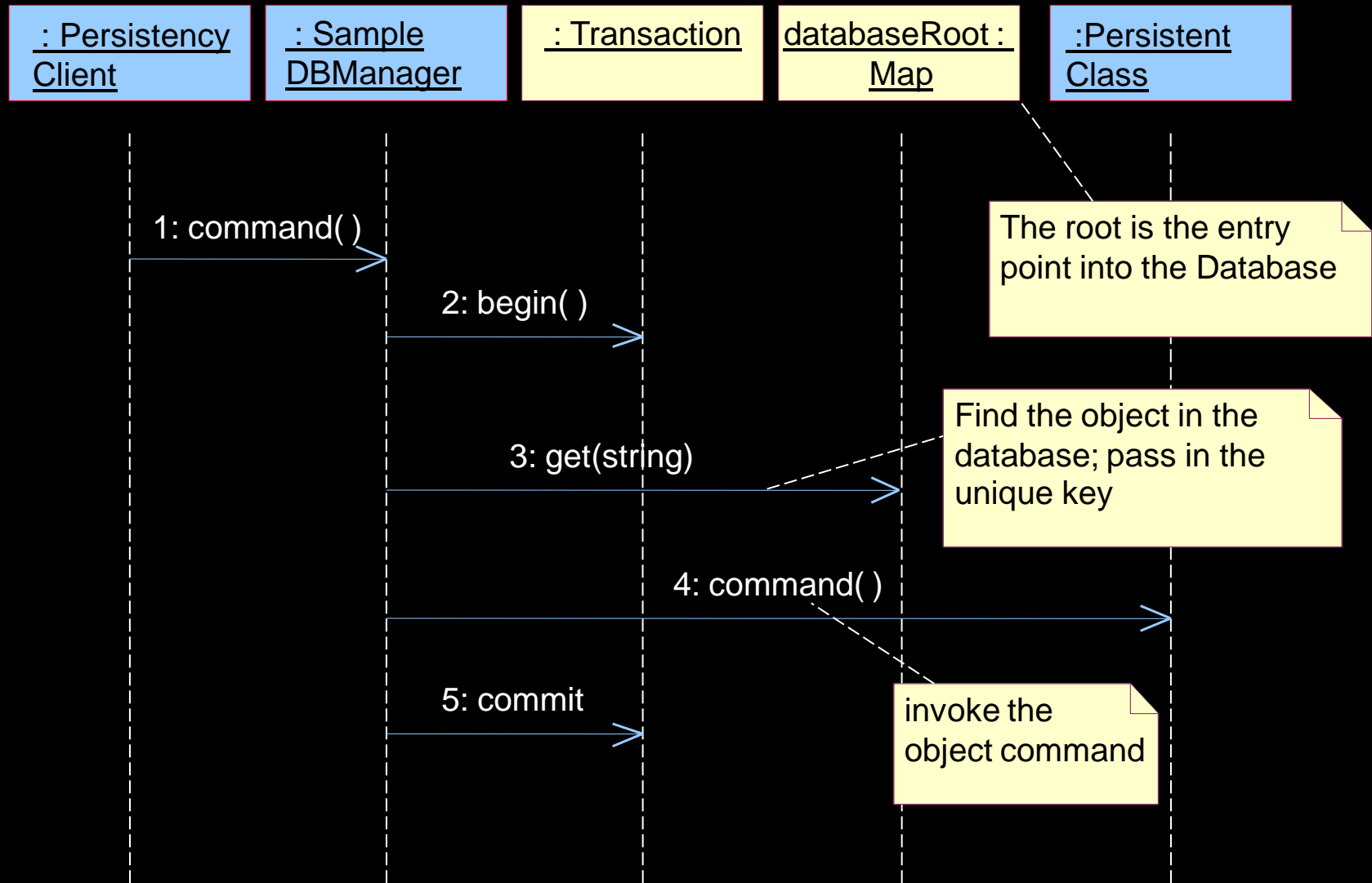
Example: Persistency: OODBMS: ObjectStore: Create



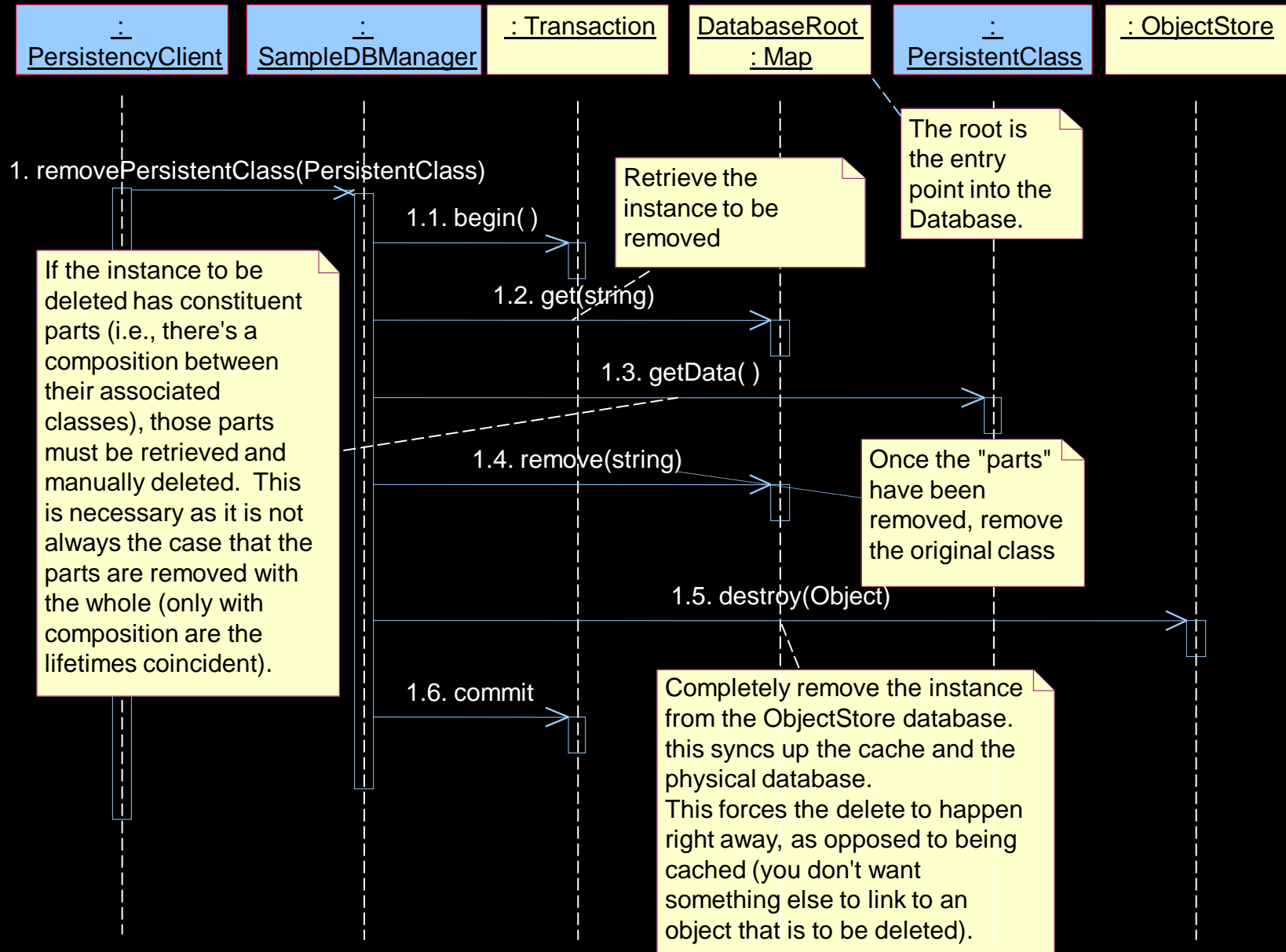
Example: Persistency: OODBMS: ObjectStore: Read



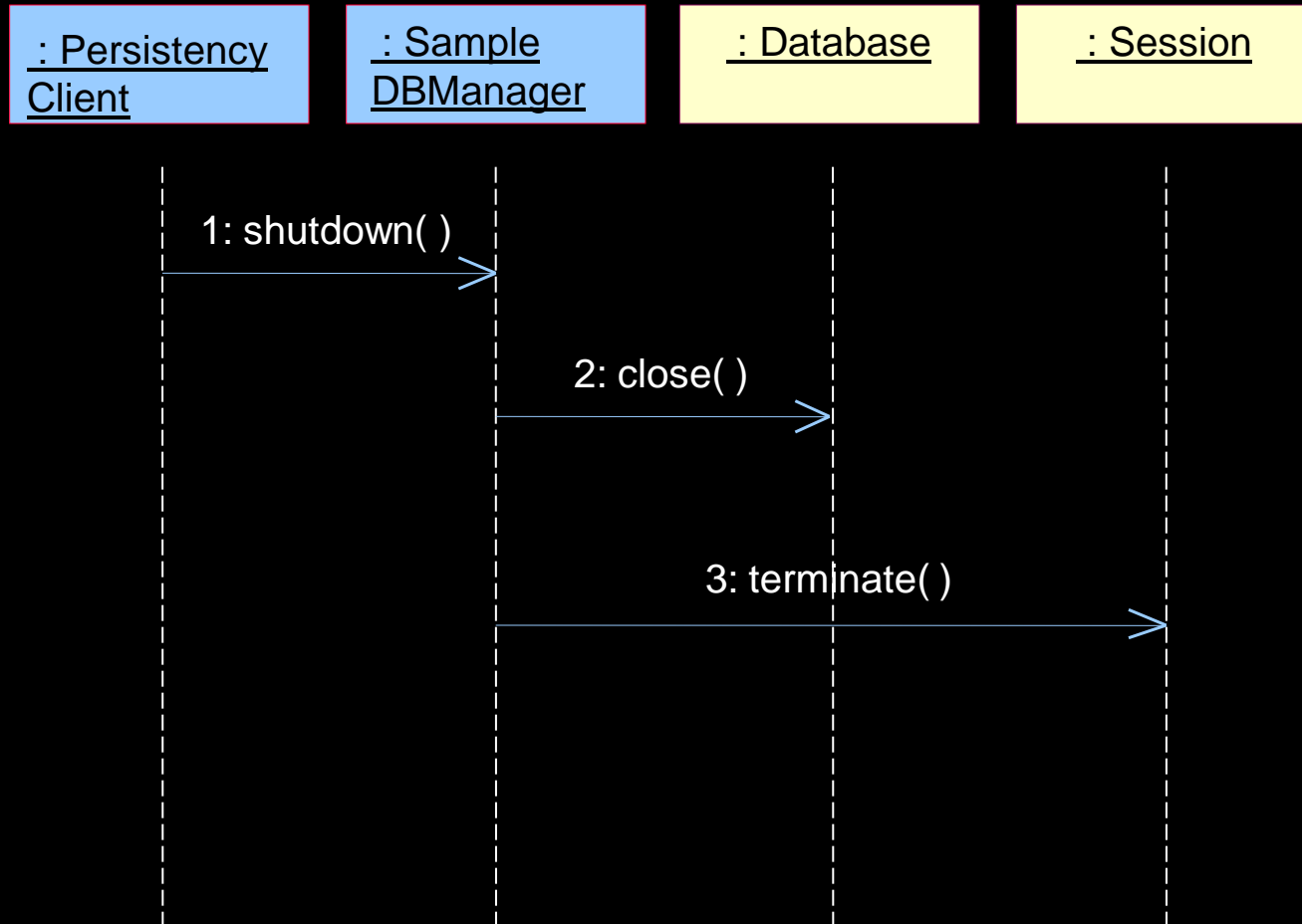
Example: Persistency: OODBMS: ObjectStore: Update



Example: Persistency: OODBMS: ObjectStore: Delete



Example: Persistency: OODBMS: ObjectStore: Shutdown



Incorporating ObjectStore: Steps

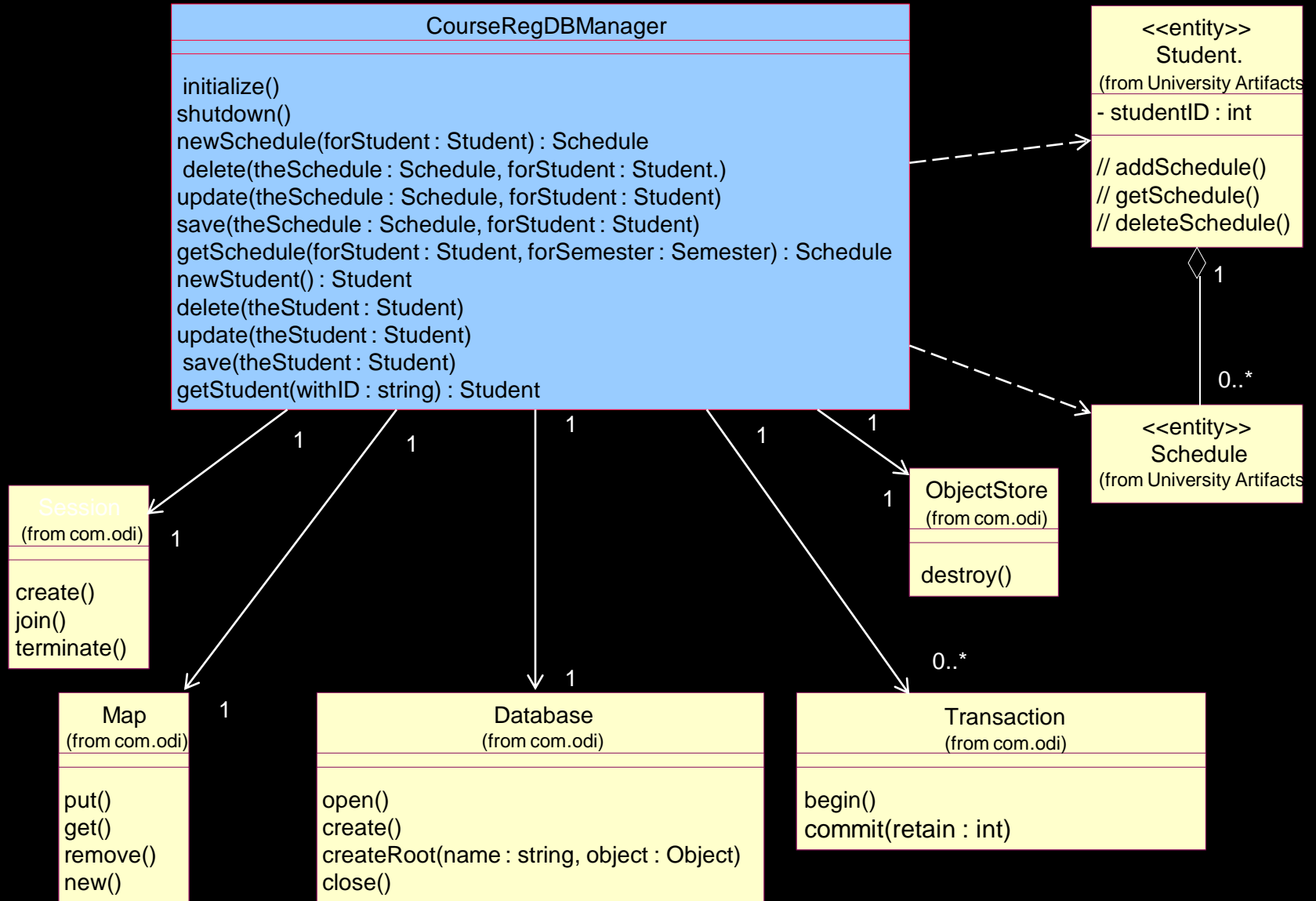
- ◆ Provide access to the class libraries needed to implement ObjectStore access
 - *Dependency on com.odi*
- ◆ Select the database root class(es)
 - *Student class*
- ◆ Select the container class(es) that will serve as the database root(s) (contains the selected root class(es))
 - *Map (from com.odi)*
 - *Key will be Student ID*

(continued)

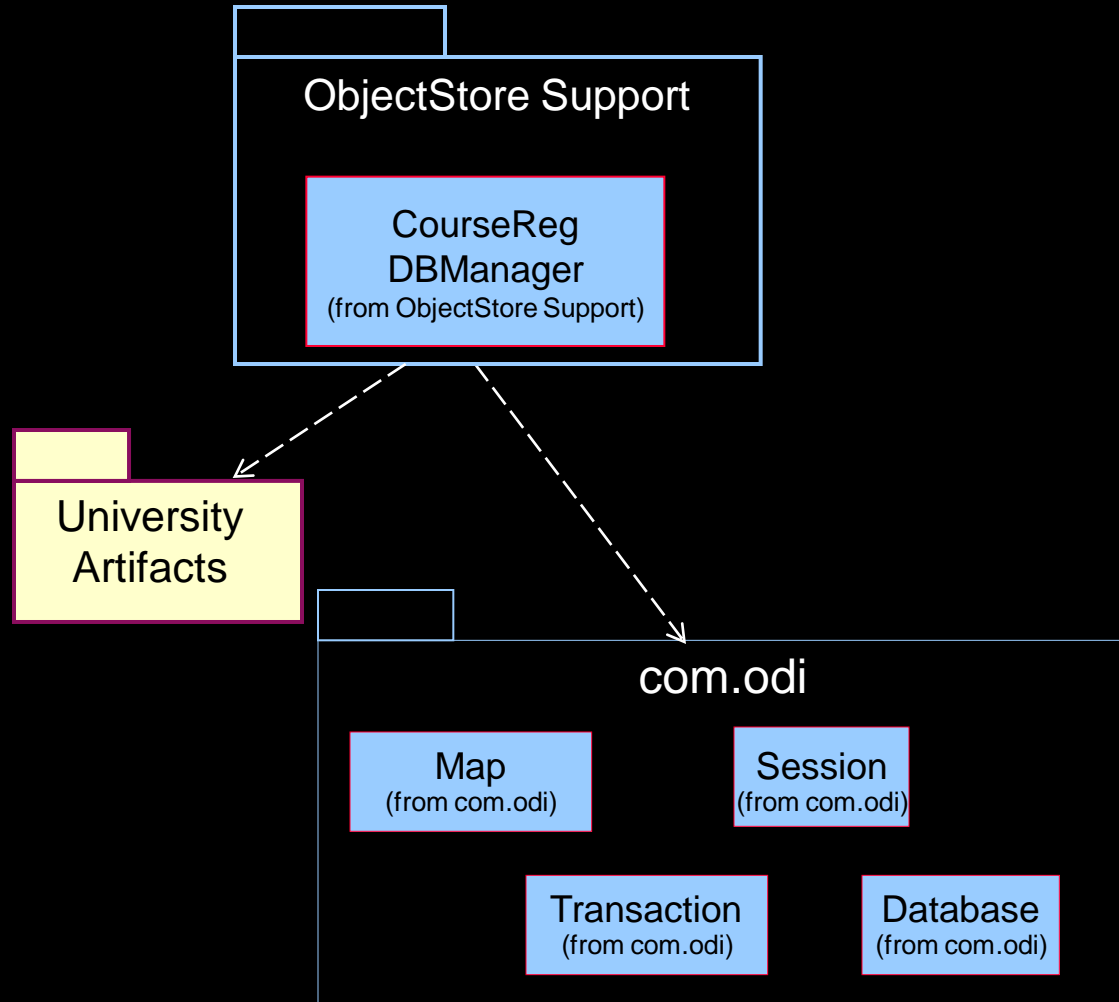
Incorporating ObjectStore: Steps (cont.)

- ◆ Create a DBManager (one per ObjectStore database instance)
 - *Single Course Registration Database => CourseRegDBManager*
 - *Will “live in” ObjectStore Support package*
 - ◆ Add operations to DBManager to access entities in the OODBMS
 - *Create operations for Student and Schedule*
 - ◆ Create/Update interaction diagrams that describe:
 - Database initialization and shutdown
 - Persistent class access: Create, Read, Update, Delete
 - ◆ Implement persistent classes
 - Add “import com.odi.*” statement
 - *Implementer to include this statement*
- Deferred*
- Out of scope*

Example: Incorporating ObjectStore



Example: Incorporating ObjectStore (cont.)



Use-Case Design Slides

The following slides can be
inserted during the Use-Case
Design module

Incorporating ObjectStore: Steps

- ◆ Provide access to the class libraries needed to implement ObjectStore access
 - ✓ ▀ *Dependency on com.odi*
- ◆ Select the database root class(es)
 - ✓ ▀ *Student class*
- ◆ Select the container class(es) that will serve as the database root(s) (contains the selected root class(es))
 - ✓ ▀ *Map (from com.odi)*
 - ✓ ▀ *Key will be Student ID*

(continued)

✓ - **Done**

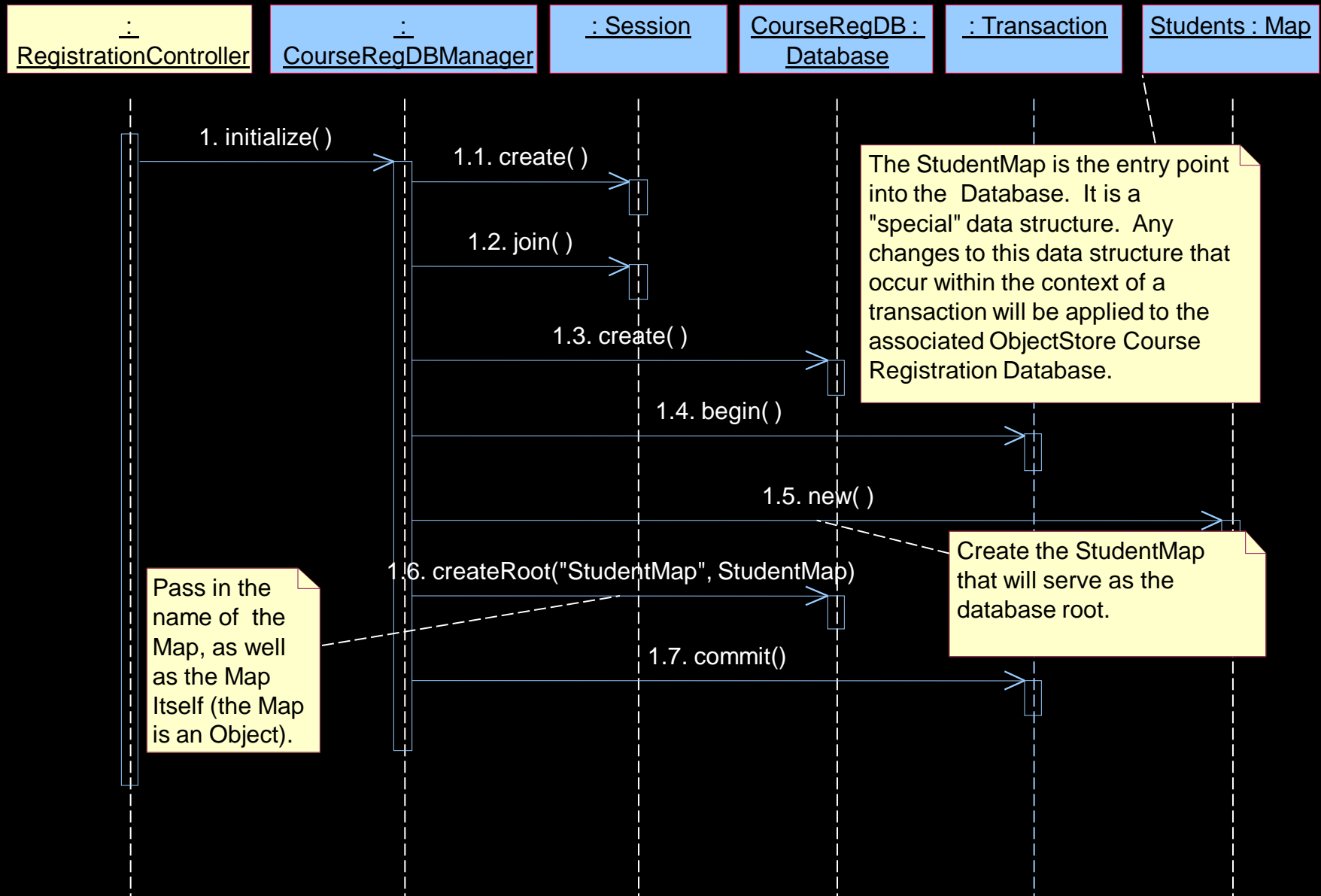
Incorporating ObjectStore: Steps

- ◆ Create a DBManager (one per ObjectStore database instance)
 - ✓ ▫ *Single Course Registration Database => CourseRegDBManager*
 - ✓ ▫ *Will “live in” ObjectStore Support package*
- ◆ Add operations to DBManager to access entities in the OODBMS
 - ✓ ▫ *Create operations for Student and Schedule*
- ◆ Create/Update interaction diagrams that describe:
 - Database initialization and shutdown
 - Persistent class access: Create, Read, Update, Delete
- ◆ Implement persistent classes
 - Add “import com.odi.*” statement
 - *Student to include this statement*

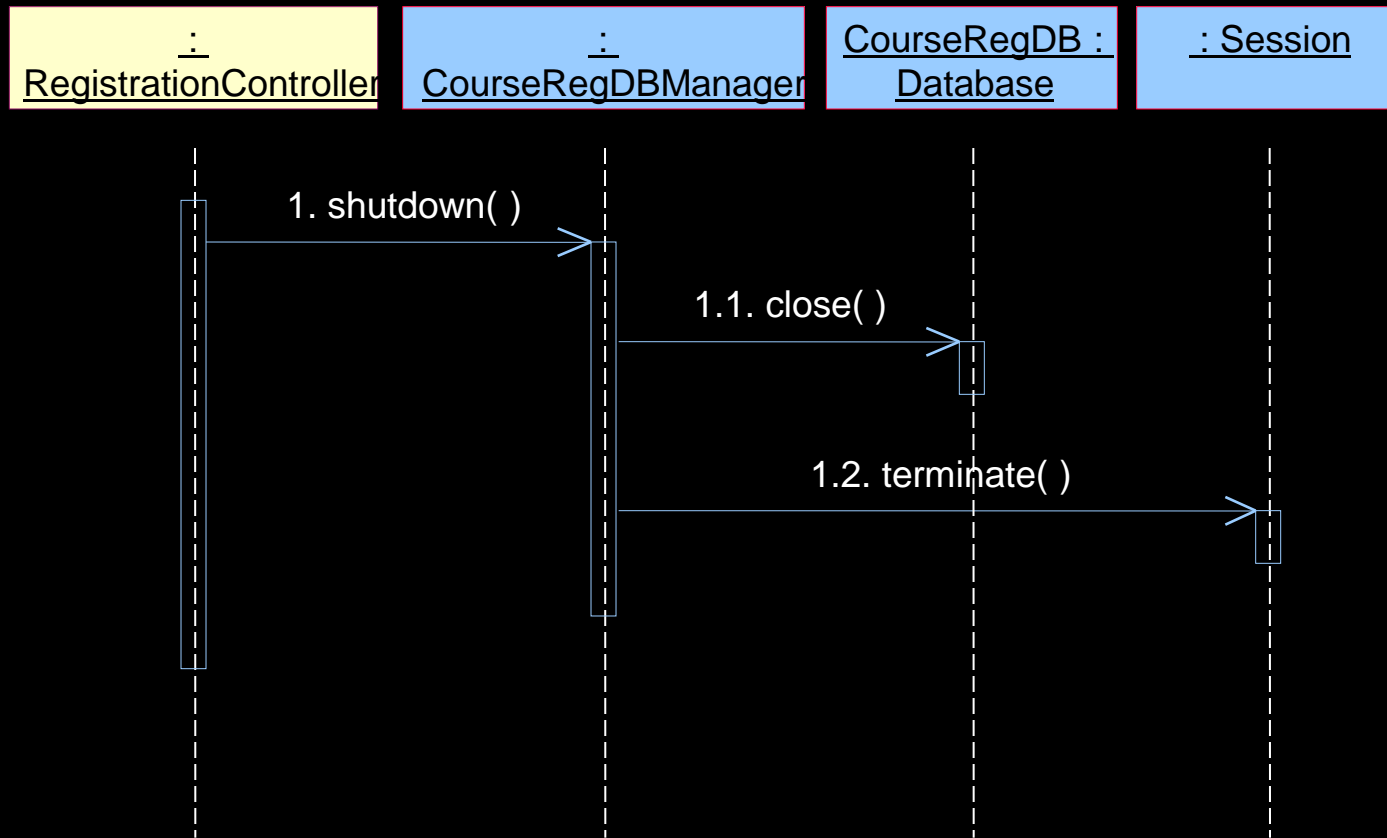
Out of scope

✓ - **Done**

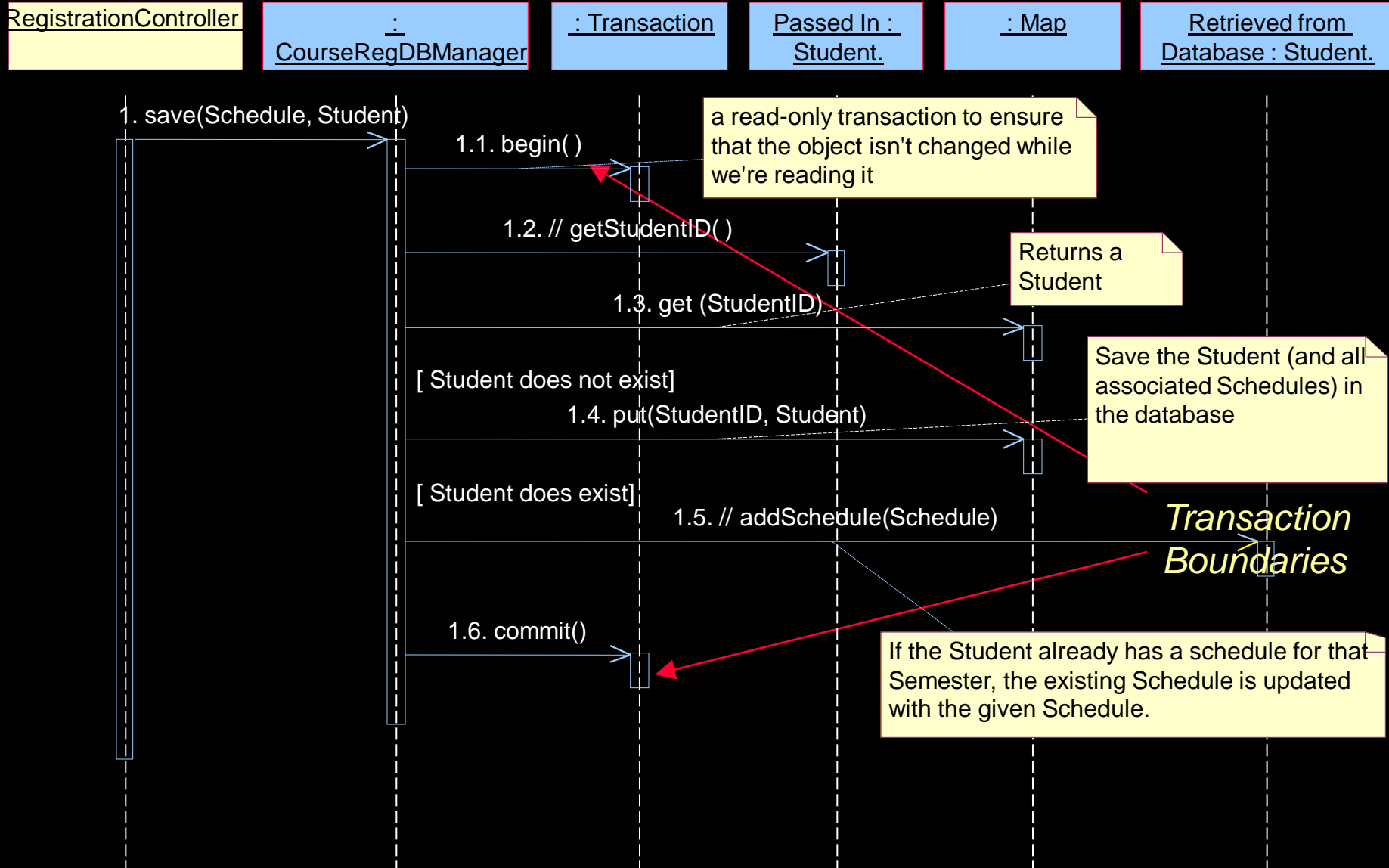
Example: Incorporating OODBMS Persistency: Initialize



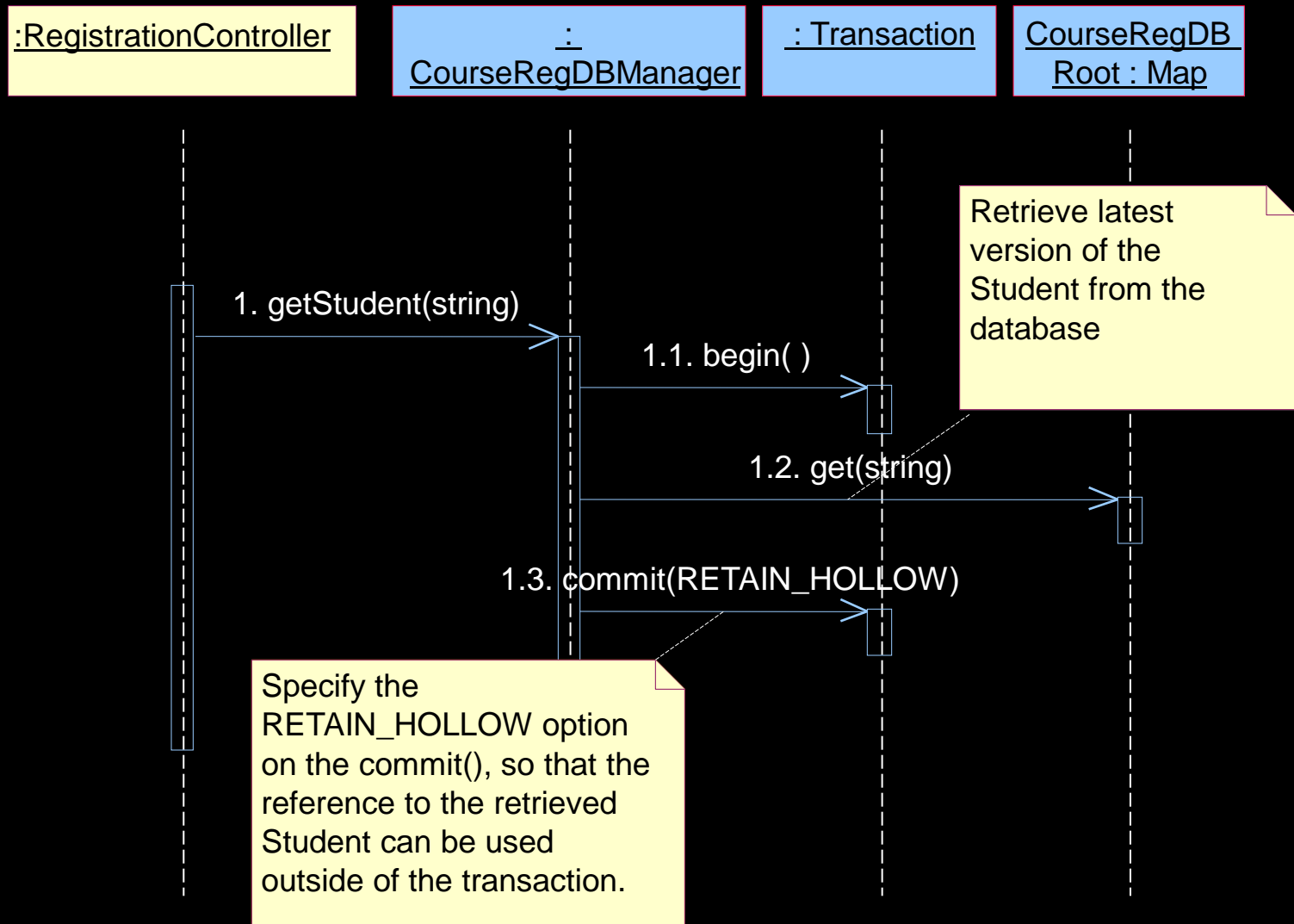
Example: Incorporating OODBMS Persistency: Shutdown



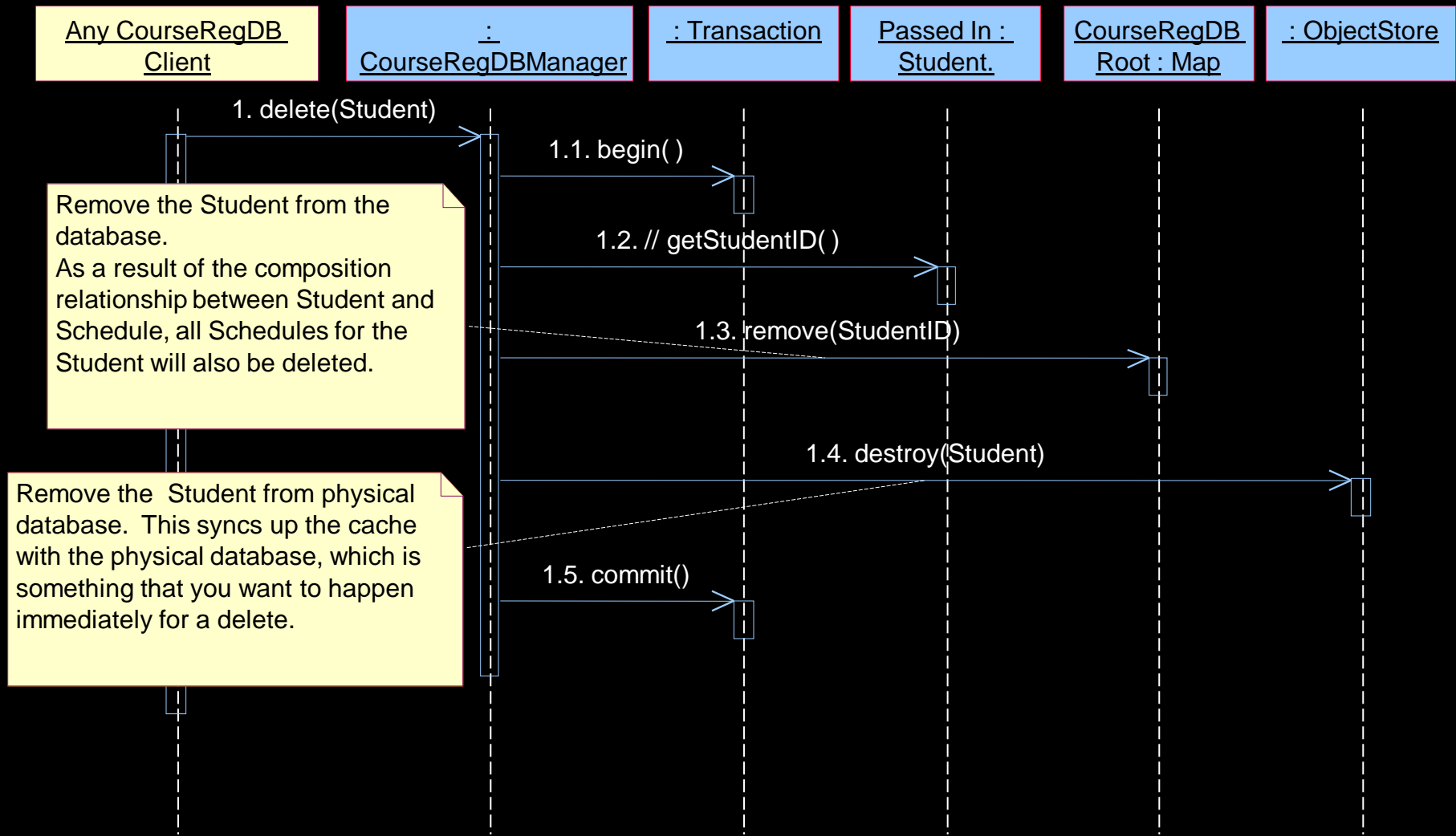
Example: Incorporating OODBMS Persistency: Create/Update



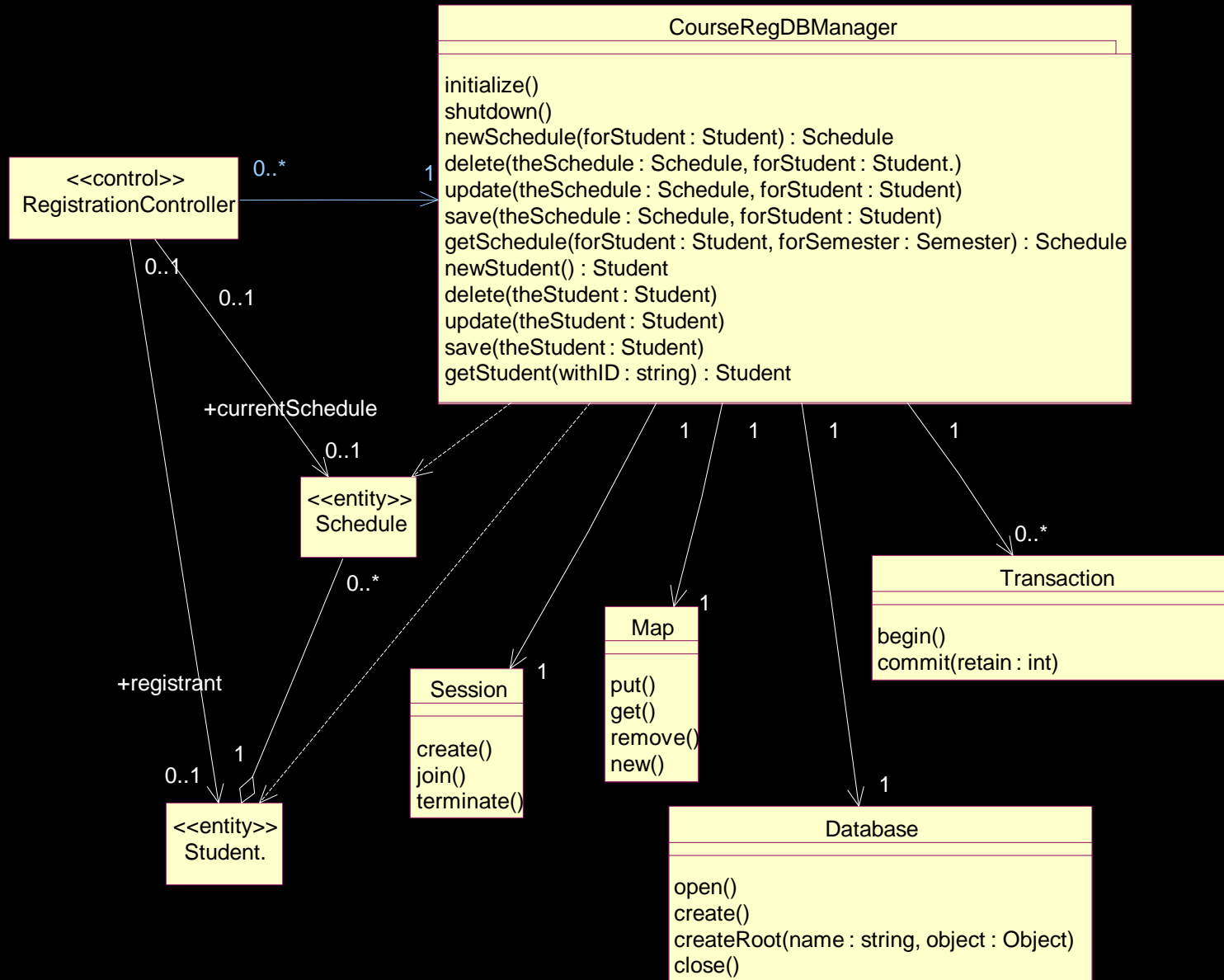
Example: Incorporating OODBMS Persistency: Read



Example: Incorporating OODBMS Persistency: Delete



Example: Incorporating OODBMS Persistency: VOPC



Example: Incorporating ObjectStore (cont.)

