MAHARISHI UNIVERSITY OF MANAGEMENT

ENGAGE THE MANAGING INTELLIGENCE OF NATURAL LAW

CS544 EA

Applications

Concurrency: Pessimistic Locking

Pessimistic Locking

- For certain operations optimistic concurrency might not be enough
  - Stricter isolation might be required to prevent the unrepeatable reads problem
  - Hibernate can **request explicit database level locks** to provide increased isolation
  - These locks will be released on commit

SELECT FOR UPDATE
on DBs that support it

```
Customer cust = (Customer)em.find(Customer.class, 1);
em.lock(cust, LockModeType.UPGRADE);
```

```
Customer cust = em.find(Customer.class, 1, LockModeType.UPGRADE);
```

LockModeType

Lock Mode	Description
None	Nothing
OPTIMISTIC	Performs a version column check
OPTIMISTIC_FORCE_INCREMENT	Ensures an increment on the version column before the transactions commits
PESSIMISTIC_FORCE_INCREMENT	Forces and increment on the version column right away and tries to get a write lock
PESSIMISTIC_READ	Tries to get a read lock in the DB
PESSIMISTIC_WRITE	Tries to get a write lock in the DB
READ	Synonym of OPTIMISTIC
WRITE	Synonym of OPTIMISTIC_FORCE_INCREMENT

Throws PersistenceException on objects without version

Newer apps should use OPTIMISTIC versions Not READ / Write