Hibernate:

Hibernate cleaner and simpler code compared to JDBC based solution including Spring JDBC template.

Caching strategy:

* none
* read-only
* read-write
* nonstrict read-write
* transactional

|  |
| --- |
| The persistent Life Cycle |

|  |  |
| --- | --- |
| Transient to Persistent | Persistent to Detached |
| save  saveOrUpddate  merge  persist | evict  close  clear |
| Detached to Persistent | Persistent to Removed |
| update  saveOrUpddate  merge | delete  remove |

To save modifications made to a detached object, either re-attach or merge it.

A detached instance will continue to remain detached even after the merge operation.

Transaction isolation issues:

|  |  |  |
| --- | --- | --- |
| Issues | Overcoming isolation level | Implementation |
| Lost update | Read uncommited | Writing transaction blocks other transactions to write but allows reads |
| Dirty read | Read commited | Reading transaction allows all other transactions to read and write  Writing transactions block all other reading and writing transactions |
| Unrepeatable read | Repeatable read | Reading transaction blocks other writing transactions but allows reading transactions  Writing transactions block all other reading and writing transactions |
| Phantom read | Serializable | Emulates serial execution. |