

ACID

- | | | |
|------------|-----------|--------------------|
| Atomic | IT-System | unit of work |
| Consistent | Developer | primary keys |
| Isolated | IT-System | no interference |
| Durable | IT-System | no lost on crashes |

Transaction Isolation

- | | |
|-------------------|--|
| Dirty read | read of uncommitted changes |
| Unrepeatable read | not the same, if read again (data changes) |
| Phantom read | new records, if queried again (new data) |

Isolation Levels

- | | |
|------------------|---|
| Read uncommitted | highest performance |
| Read committed | solves dirty read (default with optimistic locking) |
| Repeatable read | solves also unrepeatable read |
| Serializable | solves also phantom problem |

TransactionAttribute

- | | |
|--------------|--|
| NotSupported | transaction gets paused |
| Supports | part or not part |
| Required | existing or new one (default) |
| RequiresNew | always new one |
| Mandatory | transaction must exist, else TransactionRequiredException' |
| Never | transaction may not exist |
- Stateless Beans supports all attributes
 - Stateful Beans with SessionSynchronization interface only Required, RequiriesNew or Mandatory
 - Bean with Timer Service only Required, RequiriesNew or NotSupported
 - A WebService cannot use Mandatory
 - MDB can use only Required and NotSupported

Optimistic Locking

- @Version, on update version gets incremented
- On update, version value gets compared against value on start of the transaction, OptimisticLockException

LockModeType

- | | |
|-------|---|
| Read | transactions just reads data, row gets locked |
| Write | transaction also alter data (... FOR UPDATE) |

TransactionManagement (CMT, BMT)

- Container Container manages transaction, IllegalStateException on call of Connection open(), .commit() or Session.close() EJBContext setRollbackOnly()
- Bean developer defines begin(), commit(), UserTransaction from EJBContext, EJBException, if connection is not closed UserTransaction rollback()
- Can be applied on Session Beans and MDBs

Exceptions

- Rollback on RuntimeException, RemoteException
- @ApplicationException(rollback=true)
- Client receives SystemException wrapped in an EJBException and ApplicationExceptions directly
- Container will discard instance of the Bean on a SystemException

Security Goals

- Authentication (JAAS)
- Authorization (Security roles or programmatic)
- Confidentiality and integrity protection

Annotations

- @DeclareRoles (which roles needs the Bean)
- @RolesAllowed (who can access)
- @RunAs (per class), new calls are from RunAs, does not overwrite callers principal
- @PermitAll
- @DenyAll

Programmatic Security

- EJBContext getCallerPrincipal()
- EJBContext isCallerInRole(), exception if called on MDB

Deployment Descriptor

- ```

<assembly-descriptor>
 <container-transaction>
 <method>
 <ejb-name>EJBX</ejb-name>
 <method-name>*</method-name>
 </method>
 <trans-attribute>Required</trans-attribute>
 </container-transaction>

 <security-role>
 <role-name>ADMIN</role-name>
 </security-role>

 <method-permission>
 <method>
 <ejb-name>EJBX</ejb-name>
 <method-name>test</method-name>
 <!--<method-intf>Remote</method-intf>-->
 </method>
 </method-permission>

 <exclude-list>
 <method>
 <ejb-name>EJBX</ejb-name>
 <method-name>update</method-name>
 </method>
 </exclude-list>
</assembly-descriptor>

```
- ```

<session>
  <ejb-name>X</ejb-name>

  <!-- bean relies on following roles -->
  <security-role-ref>
    <role-name>employee</role-name>
  </security-role-ref>

  <security-identity>
    <run-as>
      employee
    </run-as>
  </security-identity>
</session>

```