**Hibernate XML Config implementation walk through**

**SessionFactory :** SessionFactory is a factory for Session objects in Hibernate. It is a thread-safe, immutable object. It is created only once during application startup using Hibernate configuration files. It is responsible for establishing a connection with the database and managing Hibernate configurations.

**End-to-End Hibernate Operation Flow using SessionFactory**

1. Load configuration → Create SessionFactory
2. Open a Session from SessionFactory
3. Begin a transaction
4. Perform CRUD operations
5. Commit the transaction
6. Close the Session
7. Optionally close SessionFactory at the end of the application

**Session:** A Session is a lightweight, non-thread-safe object used to interact with the database. It is a wrapper around JDBC connection, created by the SessionFactory. Each session represents a single unit of work. It provides methods like save() ,persist() ,get() ,load(),update(),merge() to perform CRUD operations.  
**Lifecycle of a Hibernate Session**

1. SessionFactory creates a Session
2. Begin a Transaction
3. Perform DB operations (CRUD)
4. Commit or Rollback the Transaction
5. Close the Session

**Transaction :** A transaction in Hibernate represents a single unit of work like insert, update, or delete . It ensures ACID properties which are Atomicity,Consistency,Isolation,Durability.  
**Transaction Lifecycle**

1. Start the session
2. Begin transaction: session.beginTransaction()
3. Perform operations (save/update/delete)
4. Commit the transaction: tx.commit()  
   Rollback on error: tx.rollback()
5. Close the session

**beginTransaction():** It is a method of the Session interface that starts a new database transaction. It marks the beginning of a unit of work. Internally, it opens a JDBC transaction. All the following DB operations are queued. These are not committed immediately. Once you call tx.commit(), changes are flushed and committed to the DB.It belongs to Session and returns the Transaction object.  
  
**commit() :** It is used to finalize a transaction and persist all changes made in the session to the database.We need to use that after all database operations are successful and you want to save them permanently.If we not use the commit() no changes can be seen in the data base.

**rollback() :** Used to cancel all operations done in the current transaction. Ensures the data base goes back to the state before the transaction began.It is used prevent partial or corrupt data entries.

**session.save() :** It inserts a new row into the database table.It requires an object identity.  
  
**session.createQuery().list() :** Executes an HQL Hibernate Query Language query and returns the result as a list. HQL uses class names and field names instead of table and column names.  
  
**session.get() :** It should be like session.get(Class, Serializable id).It fetches a single record from the database using the primary key.It returns the actual object or null if not found.  
  
**session.delete() :** It is used to delete a persistent object like a row in the database using Hibernate.Checks and marks the object for deletion.It removes the object from the persistence context.