# JAVA Hibernate

Name: Pallavi Pachika

//\*\*All the programs are attached along with this document.

1. **CRUD Operations:**

A CRUD operation deals with creating, retrieving, updating and deleting records from the table in a database.

**Program: CRUD**

1. **HQL:**

Hibernate Query Language is similar to SQL. Here we talk about classes and objects instead of tables and columns.

**Program: HQL**

1. **Calling Stored Procedure:**

1. Native SQL – createSQLQuery   
  
You can use createSQLQuery() to call a store procedure directly.   
  
Query query = session.createSQLQuery(   
"CALL GetStocks(:stockCode)")   
.addEntity(Stock.class)   
.setParameter("stockCode", "7277");   
  
List result = query.list();   
for(int i=0; i<result.size(); i++){   
Stock stock = (Stock)result.get(i);   
System.out.println(stock.getStockCode());   
}   
  
  
2. NamedNativeQuery in annotation   
  
Declare your store procedure inside the @NamedNativeQueries annotation.   
  
//Stock.java   
...   
@NamedNativeQueries({   
@NamedNativeQuery(   
name = "callStockStoreProcedure",   
query = "CALL GetStocks(:stockCode)",   
resultClass = Stock.class   
)   
})   
@Entity   
@Table(name = "stock")   
public class Stock implements java.io.Serializable {   
...   
  
  
  
-Call it with getNamedQuery().   
  
  
Query query = session.getNamedQuery("callStockStoreProcedure")   
.setParameter("stockCode", "7277");   
List result = query.list();   
for(int i=0; i<result.size(); i++){   
Stock stock = (Stock)result.get(i);   
System.out.println(stock.getStockCode());   
}

1. **Annotations / Configuration Based:**

In hibernate we deal with many annotations like @Table, @Entity, @Basic, @Transient, @Lob, @GeneratedValue, @ID, @ElementCollection, @JoinTable, @ColelctionId, @GenreicGenerator etc.

**Program: Annotations, Configurations**

1. **Collections:**

We can map collection elements of Persistent class in Hibernate. You need to declare the type of collection in Persistent class from one of the following types:

* java.util.List
* java.util.Set
* java.util.SortedSet
* java.util.Map
* java.util.SortedMap
* java.util.Collection
* or write the implementation of org.hibernate.usertype.UserCollectionType

@ElementCollection is used as annotation in hibernate

**Program: Collections**

1. **Caching:**

First Level Caching- Default in Hibernate- Session

Second Level caching: 1. Across sessions in an application

2. Across Applications

3. Across Clusters

**Program: Caching**

1. **Lazy loading and eager Loading:**

Lazy loading: Get it only when you need it. Here we don’t initialize entire object. Only initialize first level member variables.

Eager Loading: Intilaize everything at first. Takes lot of time.

**Program: Lazy\_Eager\_Loading**

1. **Transient, Persistent and detached objects:**

Transient: An object is transient if it is instantiated using new operator and it is not associated with hibernate.

Persistent: It has representation in database. Hibernate will detect any changes made to an object in persistent state.

Detached: It is no longer tracked by hibernate.

**Program: Transient\_Persistent**

1. **What is JPA**

Java Persistence API is a collection of classes, methods to persistently store vast amount of data into database.

1. **Session.update, merger,lock ,save:**

**update method:** it is used to update the already persistent object in the datastore.

**Merge method:** it will act like update but here if a persistent object with the same identifier is already in the session it will update the detached object values in the persistent object and save it.

**save method:** it is used to insert the newly created object in the datastore.

**Lock:** Locking refers to actions taken to prevent data in a relational database from changing between the time it is read and the time that it is used.