**SessionFactory level cache-**

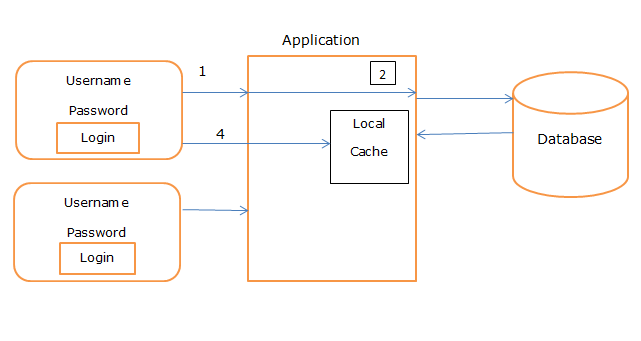
It is called as second level cache.

It will apply for all the users.

Program-2 Using SessionFactory

**Why?**

Problem with session level cache



Here admin1 and admin2 are the trying to access the same data, if we are using session, it will create the two session objects for some data or common data but if we use sessionfactory cache, data will fetch from database only one(data store under the sessionfactory) and data will available to any user.

**When?**

It is useful if you have multiple session objects from sessionfactory.

Program- using SessionFactory

**Student.java**

**package** com.session;

**import** javax.persistence.Cacheable;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

@Entity

**public** **class** Student {

@Id

**private** **int** id;

**private** String name;

**private** String city;

**private** String mobile;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL57Dialect</property>

<property name=*"hbm2ddl.auto"*>update</property>

<property name=*"show\_sql"*>true</property>

<property name=*"cache.use\_Second\_level\_cache"*>true</property>

<property name=*"hibernate.cache.provider\_class"*>

org.hibernate.cache.EhCacheProvider

</property>

<property name=*"net.sf.ehcache.configuration-ResourceName"*>ehcache.xml</property>

<mapping class=*"com.hibernate.Student"*></mapping>

</session-factory>

</hibernate-configuration>

Ehcache.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<ehcache xmlns:xsi=*"https://www.w3.org/2001/XMLSchema-instance"*

xsi:noNamespaceSchemaLocation=*"ehcache.xsd"* updateCheck=*"true"*

monitoring=*"autodetect"* dynamicConfig=*"true"*>

<diskStore path=*"java.io.tmpdir/ehcache"* />

<defaultCache maxEntriesLocalHeap=*"10000"* eternal=*"false"*

timeToIdleSeconds=*"120"* timeToLiveSeconds=*"120"*

diskSpoolBufferSizeMB=*"30"* maxEntriesLocalDisk=*"10000000"*

diskExpiryThreadIntervalSeconds=*"120"* memoryStoreEvictionPolicy=*"LRU"*

statistics=*"true"*>

<persistence strategy=*"localTempSwap"* />

</defaultCache>

<cache name=*"employee"* maxEntriesLocalHeap=*"10000"*

eternal=*"false"* timeToIdleSeconds=*"5"* timeToLiveSeconds=*"10"*>

<persistence strategy=*"localTempSwap"* />

</cache>

<cache name=*"org.hibernate.cache.internal.StandardQueryCache"*

maxEntriesLocalHeap=*"5"* eternal=*"false"* timeToLiveSeconds=*"120"*>

<persistence strategy=*"localTempSwap"* />

</cache>

<cache name=*"org.hibernate.cache.spi.UpdateTimestampsCache"*

maxEntriesLocalHeap=*"5000"* eternal=*"true"*>

<persistence strategy=*"localTempSwap"* />

</cache>

</ehcache>

**package** com.session;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration configuration = **new** Configuration();

configuration.configure("hibernate.cfg.xml");

SessionFactory sessionFactory = configuration.~~buildSessionFactory~~();

Session session=sessionFactory.openSession();

Student s1 = (Student) session.load(Student.**class**, 1);

System.***out***.println("First student data is>>");

System.***out***.println("Name>>" + s1.getName());

System.***out***.println("City>>" + s1.getCity());

System.***out***.println("Mobile>>" + s1.getMobile());

System.***out***.println("second level cache>>");

Session session1=sessionFactory.openSession();

Student s3 = (Student) session1.load(Student.**class**, 2);

System.***out***.println("Student data is>>");

System.***out***.println("Name>>" + s3.getName());

System.***out***.println("City>>" + s3.getCity());

System.***out***.println("Mobile>>" + s3.getMobile());

}

}

**Pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>session</groupId>

<artifactId>SessionDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.1.4.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.21</version>

</dependency>

<dependency>

<groupId>net.sf.ehcache</groupId>

<artifactId>ehcache-core</artifactId>

<version>2.6.9</version>

</dependency>

<!-- Hibernate EHCache API -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-ehcache</artifactId>

<version>5.2.16.Final</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>1.7.5</version>

</dependency>

</dependencies>

</project>