**#JDBC CONNECTION**

Java.sql.Connection conn=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/springhibernate?useSSL=false","root","admin");

**Hibernate**

* The **SessionFactory** is a **factory of session and client of ConnectionProvider**. It holds second level cache (optional) of data. The org.hibernate.SessionFactory interface provides factory **method to get the object of Session.**
* The **session** object provides an interface between the application and data stored in the database. It is a short-lived object and wraps the JDBC connection. It is factory of Transaction, Query and Criteria. It holds a first-level cache (mandatory) of data. The org.hibernate.Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.
* The **transaction** object **specifies the atomic unit of work**. It is **optional**. The org.hibernate.Transaction interface provides methods for transaction management.
* **Connection Provider**

It is a factory of JDBC connections. It abstracts the application from DriverManager or DataSource. It is optional.

**HIBERNATE ANNOTATION**

**@GenericGenerator in order to generate hibernate based pk**

**This collection should have and identifier(like as pk)**

**@CollectionId(columns =@Col)**

**##why and wen to use**

**@fetch=fetchType.eager/lazay**

**When there is object or collection for object in an entity class BY DEFAULT HINERNATE WILL NOT FETCH such object (by default its lazy type only 1st level of member var will initialize i.e parent and composition will not)**

**To fetch it we hav to either call those method explicity or uses @Eager annot**