Normalization	is a process of organizing data to minimize data redundancy. Which in turn ensures data consistency.	**Advan. Of RDBMS**	● Controlling Redundancy. ● Integrity can be enforced. ● Inconsistency can be avoided. ● Data can be shared. ● Standard can be enforced	
1NF	Identity each record uniquely using the primary key.	**JPA**	Mapping between database tables and java objects called ORM. JPA provides an ORM facility for managing relational tables in Java applications.	
2NF	move redundant data to separate table. creates a relationship between these tables using foreign keys.	**EntityManager**	provides the operations from & to the database, e.g. find objects, persists them, remove objects from the db, etc.	
3NF	doesn't contain attributes which are partially dependent upon primary key.	**Annotations**	**1. Entity**: Used with model classes to specify that they are entity beans. **2**. **Table**: Used with entity beans to define the corresponding table name in database. **3. Access**: define the access type, either field or prop. **4. Id**: Used to define the primary key in the entity bean. **5. EmbeddedId**: define composite pk in the entity bean. **6**. **Column**: Used to define the column name in db table. **7**. **GeneratedValue**: strategy to be used for generation of pk. Used in conjunction with GenerationType enum. **8**. **OneToOne**: define the mapping b/w two entities., one-to-one, OneToMany, ManyToOne and ManyToMany **9**. **Cascade**: define the cascading b/w two entities, used with mappings. It works in conjunction with CascadeType **10.PrimaryKeyJoinColumn**: define the property for foreign key. Used with GenericGenerator and Parameter	
Adv Of Hibernate	1. Removes boilerplate code that comes with JDBC. 2. supports inheritance, associations and collections. 3. implicitly provides transaction management 4. HQL is more object oriented. 5. supports caching that is better for performance. 6. provide option where we can create db tables too 7. Supports JPA annotations.			
Important Interfaces of Hibernate	1. SessionFactory (org.hibernate.SessionFactory) 2. Session (org.hibernate.Session) 3. Transaction (org.hibernate.Transaction)	**Hibernate Config file**	Hibernate configuration file contains database specific configurations and used to initialize SessionFactory. database credentials or JNDI resource or dialect info.	
Session Factory	SessionFactory is an immutable thread-safe cache of compiled mappings for a single database. SessionFactory instance is used to get the Session objects for database operations.	**Hibernate mappings file**	Hibernate mapping file is used to define the entity bean fields and database table column mappings.	
Session	Session is a single-threaded, short-lived object representing a conversation between the application and the persistent store.	**Get vs load**	1. load() is better because it support lazy loading. 2. load() throws exception when data is not found 3. use get() to make sure data exists in the db.	
Transaction	Transaction is a single-threaded, short-lived object used by the application to specify atomic units of work.	**Transient**: an obj never persisted/associated with any session.	**Persistent**: an obj associated with uniq sess.	**Detached**: an obj prev persistent but not assoc. with any sess.
Named Query	Hibernate provides Named Query that we can define at a central location and use them anywhere in the code. @NamedQuery and @NamedNativeQuery.	**Save**: save with no transaction. Results in data inconsistency.	**Persist**: Save with transaction.	**SaveorUpdate**: just like the name depends on the data.
Hiber. Design patterns	Domain model pattern, Data mapper, Proxy pattern, Factory pattern	**Optimistic Locking** is is a strategy where you read a record, take note of a version number (other methods to do this involve dates, timestamps or checksums/hashes) and check that the version hasn't changed before you write the record back.		
Hiber Collec's	Bag, Set, List, Array, Map	**Pessimistic Locking** is when you lock the record for your exclusive use until you have finished with it.		

Note: Information gathered in this document has been collected from various sources on the Internet.