Hibernate supports 3 types of Inheritance

1. Table per class
2. Table per sub-class
3. Table per concrete-class

In hibernate inheritance hierarchy if we save derived class object then base class object also will be stored into the database automatically.

* Consider 3 POJO classes, Employee class as a base class and SalEmployee, HourEmployee as derived classes.
* For all POJO classes we have only one mapping XML file which is nothing but Base class mapping file.
* Both derived classes data has to map in base class mapping file only.

**Table per class :**

* In Table per class hierarchy, both base class and derived class data will be stored into a single table which is base class related table.
* For all 3 POJO classes we have single mapping file which is for base (Employee) class.
* To map derived classes details into base class mapping file , we need to use <subclass> tag.
* Here if we save either HourEmployee or SalEmployee then automatically Employee object also will be stored into database. Both derived class object (HourEmployee/SalEmployee) and base class object(Employee) will be stored into Employee related table only.
* In Table per class hiierarchy, we need to use one special column in the database table which is known as discriminator.
* Discriminator column will help us to identify which derived class object is stored along with base class object.

**Table per sub-class :**

* In Table per class, both derived class object and base class object are stored in base class related table only, which means for all POJO classes we have single table.
* But in Table per sub-class, per each class we have a separate table(including subclass also).   
  **X number of classes = X number of tables**.
* In Table per sub-class also for all POJO classes we have one mapping file which is base classes mapping file. Derived class details we have to map in base class mapping file only.
* To map derived class details inside base class mapping file, we use <joined-subclass> tag.
* In Table per sub-class, discriminator column not required because each class data is stored in separate table. But we need to take one <key> tag, inside <joined-subclass> tag.

**Note :**POJO class, Mapping and Configuration files are same as above application.

**Table per concrete-class :**

* In Table per concrete class, we have tables only for derived classes.   
  **X number of derived classes = X number of tables**
* Here Base class data also will be saved into derived class related table.
* One mapping file for all POJO classes. Here also we need to take mapping file only for base class.
* To map derived class details we need to take <union-subclass> tag inside base class related mapping file.

**Note :**POJO class, Mapping and Configuration files are same as above application.