Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据

Hibernate缓存分类：

Hibernate一级缓存：Session的缓存”

由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。第一级缓存是必需的，不允许而且事实上也无法卸除。在第一级缓存中，持久化类的每个实例都具有唯一的OID。

Hibernate二级缓存Hibernate：SessionFactory的缓存”

由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别

什么样的数据适合存放到第二级缓存中？  
1 很少被修改的数据  
2 不是很重要的数据，允许出现偶尔并发的数据  
3 不会被并发访问的数据  
4 常量数据  
不适合存放到第二级缓存的数据？  
1 经常被修改的数据  
2 .绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发  
3 与其他应用共享的数据。

如何使用：

一般先去查询OID。查不到再去查看二级缓存，再查不到再去查数据库。把结果按照ID放入到缓存,删除、更新、增加数据的时候，同时更新缓存。Hibernate管理缓存实例无论何时，当你给save()、update()或saveOrUpdate()方法传递一个对象时，或使用load()、 get()、list()、iterate() 或scroll()方法获得一个对象时, 该对象都将被加入到Session的内部缓存中。 当随后flush()方法被调用时，对象的状态会和数据库取得同步。 如果你不希望此同步操作发生，或者你正处理大量对象、需要对有效管理内存时，你可以调用evict() 方法，从一级缓存中去掉这些对象及其集合

.在Hibernate中实现数据检索的5种方式？

答：① 对象导航（关联级别的数据检索）

② HQL语句

③ SQL语句

④ QBC语句

⑤ 通过OID加载（get( ) / load( )）

事务并发访问的问题？

答：①脏读：一个事务读到了另一个事务未提交的数据；

②不可重复读：在同一个事务中，多次查询的结果不一致（由update引起的）

③虚读/幻读：在同一个事务中，多次查询的结果不一致（由insert引起的）

使用Hibernate的基本流程：配置Configuration对象、产生SessionFactory、创建session对象，开启事务，完成CRUD操作，提交事务，关闭session。