Java第四阶段\_day01\_Sharding-jdbc答案

1. 简述题：简述什么是Sharding-JDBC。

Sharding-JDBC是当当网研发的开源分布式数据库中间件，Sharding-JDBC的核心功能为数据分片和读写分离，通过Sharding-JDBC，应用可以透明的使用jdbc访问已经分库 分表、读写分离的多个数据源，而不用关心数据源的数量以及数据如何分布。

1. 编程题：Sharding-JDBC使用完成水平分库。

# 定义多个数据源  
spring.shardingsphere.datasource.names = m1,m2  
​  
spring.shardingsphere.datasource.m1.type = com.alibaba.druid.pool.DruidDataSource spring.shardingsphere.datasource.m1.driver‐class‐name = com.mysql.jdbc.Driver spring.shardingsphere.datasource.m1.url = jdbc:mysql://localhost:3306/order\_db\_1?useUnicode=true spring.shardingsphere.datasource.m1.username = root spring.shardingsphere.datasource.m1.password = root  
​  
spring.shardingsphere.datasource.m2.type = com.alibaba.druid.pool.DruidDataSource spring.shardingsphere.datasource.m2.driver‐class‐name = com.mysql.jdbc.Driver spring.shardingsphere.datasource.m2.url = jdbc:mysql://localhost:3306/order\_db\_2?useUnicode=true spring.shardingsphere.datasource.m2.username = root spring.shardingsphere.datasource.m2.password = root  
...  
# 分库策略，以user\_id为分片键，分片策略为user\_id % 2 + 1，user\_id为偶数操作m1数据源，否则操作m2。spring.shardingsphere.sharding.tables.t\_order.database‐strategy.inline.sharding‐column = user\_id   
spring.shardingsphere.sharding.tables.t\_order.database‐strategy.inline.algorithm‐expression = m$‐>{user\_id % 2 + 1}

@Test  
public void testInsertOrder(){   
    for (int i = 0 ; i<10; i++){  
 orderDao.insertOrder(new BigDecimal((i+1)\*5),1L,"WAIT\_PAY");  
 }  
 for (int i = 0 ; i<10; i++){  
 orderDao.insertOrder(new BigDecimal((i+1)\*10),2L,"WAIT\_PAY");  
 }  
}

1. 面试题： 什么是读写分离？

面对日益增加的系统访问量，数据库的吞吐量面临着巨大瓶颈。 对于同一时刻有大量并发读操作和较少写操作类型的应用系统来说，将数据库拆分为主库和从库，主库负责处理事务性的增删改操作，从库负责处理查询操作，能 够有效的避免由数据更新导致的行锁，使得整个系统的查询性能得到极大的改善。