抢购、秒杀是如今很常见的一个应用场景，主要需要解决的问题有两个：

1 高并发对数据库产生的压力

2 竞争状态下如何解决库存的正确减少（"超卖"问题）

先重点解决问题2：

常规写法：

|  |
| --- |
| 1. <?php 2. $conn=mysql\_connect("localhost","big","123456"); 3. **if**(!$conn){ 4. echo "connect failed"; 5. exit; 6. } 7. mysql\_select\_db("big",$conn); 8. mysql\_query("set names utf8"); 10. $price=10; 11. $user\_id=1; 12. $goods\_id=1; 13. $sku\_id=11; 14. $number=1; 16. //生成唯一订单 17. **function** build\_order\_no(){ 18. **return** date('ymd').substr(implode(NULL, array\_map('ord', str\_split(substr(uniqid(), 7, 13), 1))), 0, 8); 19. } 20. //记录日志 21. **function** insertLog($event,$type=0){ 22. **global** $conn; 23. $sql="insert into ih\_log(event,type) 24. values('$event','$type')"; 25. mysql\_query($sql,$conn); 26. } 28. //模拟下单操作 29. //库存是否大于0 30. $sql="select number from ih\_store where goods\_id='$goods\_id' and sku\_id='$sku\_id'";//解锁 此时ih\_store数据中goods\_id='$goods\_id' and sku\_id='$sku\_id' 的数据被锁住(注3)，其它事务必须等待此次事务 提交后才能执行 31. $rs=mysql\_query($sql,$conn); 32. $row=mysql\_fetch\_assoc($rs); 33. **if**($row['number']>0){//高并发下会导致超卖 34. $order\_sn=build\_order\_no(); 35. //生成订单 36. $sql="insert into ih\_order(order\_sn,user\_id,goods\_id,sku\_id,price) 37. values('$order\_sn','$user\_id','$goods\_id','$sku\_id','$price')"; 38. $order\_rs=mysql\_query($sql,$conn); 40. //库存减少 41. $sql="update ih\_store set number=number-{$number} where sku\_id='$sku\_id'"; 42. $store\_rs=mysql\_query($sql,$conn); 43. **if**(mysql\_affected\_rows()){ 44. insertLog('库存减少成功'); 45. }**else**{ 46. insertLog('库存减少失败'); 47. } 48. }**else**{ 49. insertLog('库存不够'); 50. } 51. ?> |

优化方案1：将库存字段number字段设为unsigned，当库存为0时，因为字段不能为负数，将会返回false

|  |
| --- |
| 1. //库存减少 2. $sql="update ih\_store set number=number-{$number} where sku\_id='$sku\_id' and number>0"; 3. $store\_rs=mysql\_query($sql,$conn); 4. **if**(mysql\_affected\_rows()){ 5. insertLog('库存减少成功'); 6. } |

优化方案2：Mysql的事务，锁住操作的行

|  |
| --- |
| 1. <?php 2. $conn=mysql\_connect("localhost","big","123456"); 3. **if**(!$conn){ 4. echo "connect failed"; 5. exit; 6. } 7. mysql\_select\_db("big",$conn); 8. mysql\_query("set names utf8"); 10. $price=10; 11. $user\_id=1; 12. $goods\_id=1; 13. $sku\_id=11; 14. $number=1; 16. //生成唯一订单号 17. **function** build\_order\_no(){ 18. **return** date('ymd').substr(implode(NULL, array\_map('ord', str\_split(substr(uniqid(), 7, 13), 1))), 0, 8); 19. } 20. //记录日志 21. **function** insertLog($event,$type=0){ 22. **global** $conn; 23. $sql="insert into ih\_log(event,type) 24. values('$event','$type')"; 25. mysql\_query($sql,$conn); 26. } 28. //模拟下单操作 29. //库存是否大于0 30. mysql\_query("BEGIN");   //开始事务 31. $sql="select number from ih\_store where goods\_id='$goods\_id' and sku\_id='$sku\_id' FOR UPDATE";//此时这条记录被锁住,其它事务必须等待此次事务提交后才能执行 32. $rs=mysql\_query($sql,$conn); 33. $row=mysql\_fetch\_assoc($rs); 34. **if**($row['number']>0){ 35. //生成订单 36. $order\_sn=build\_order\_no(); 37. $sql="insert into ih\_order(order\_sn,user\_id,goods\_id,sku\_id,price) 38. values('$order\_sn','$user\_id','$goods\_id','$sku\_id','$price')"; 39. $order\_rs=mysql\_query($sql,$conn); 41. //库存减少 42. $sql="update ih\_store set number=number-{$number} where sku\_id='$sku\_id'"; 43. $store\_rs=mysql\_query($sql,$conn); 44. **if**(mysql\_affected\_rows()){ 45. insertLog('库存减少成功'); 46. mysql\_query("COMMIT");//事务提交即解锁 47. }**else**{ 48. insertLog('库存减少失败'); 49. } 50. }**else**{ 51. insertLog('库存不够'); 52. mysql\_query("ROLLBACK"); 53. } 54. ?> |

优化方案3：