1. 这个月的主要工作任务是语音外呼，由于是面向客户的支撑平台，所以对数据并发的处理需要慎重思考。
2. 语音外呼支撑平台中并发的问题：
3. 自动获取一个导入的回访电话数据，然后更改获取到的这一条数据状态，其他人不能再获取这一条数据，存在的问题是，如果两个人同时查询到了一条数据，第二个的修改就会造成第一个人获取失效。为了避免这种情况的发生，我们使用了select for update对获取到的行进行了加锁。下面简单介绍一下select for update的使用方法。
5. 先说明一下会用到的名词概念：
6. statement: 一个SQL语句。
8. session: 一个由ORACLE用户产生的连接，一个用户能产生多个SESSION ，但相互之间是独立的。
10. transaction:所有的改动都能划分到transaction里，一个transaction包含一个或多个SQL。当一个SESSION建立的时候就是个TRANSACTION开始的时刻，此后transaction的开始和结束由DCL控制，也就是每个COMMIT/ROLLBACK都标示着一个transaction的结束。
12. consistency：是对于statement级别而不是transaction级别来说的。sql statement 得到的数据都是以sql statement开始的IMAGE。
14. 用法介绍：
15. update, insert ,delete, select ... for update会LOCK相应的ROW 。
16. 只有一个TRANSACTION可以LOCK相应的行，也就是说如果一个ROW已经LOCKED了，那就不能被其他TRANSACTION所LOCK了。
18. LOCK由statement产生但却由TRANSACTION结尾（commit，rollback），也就是说一个SQL完成后LOCK还会存在，只有在COMMIT/ROLLBACK后才会师释放。
20. 简单举例：
21. transaction A下面
22. select \* from  connector a where a.MOBILE='13937134399' for updata
24. 如上，这个将会对查询出来的行加上一个行锁，如果在
25. transaction B下面对这一行数据进行增删改都将会等待，普通的查询可以，使用 select for update 查询也需要等待。
26. 需要在transaction A下面执行需要的操作之后commit或者rollback之后，在其他transaction 下面才可以对此表此行进行操作。
28. 使用这个行锁的情况一般是对并发的情况要求比较高的时候，需要锁住某行进行一些更新语句之后进行释放，再让其他transaction 去操作。在这次外呼系统中很好的利用了这点解决了并发的问题。
30. for update 后面还可以跟着[OF cols] [NOWAIT]
31. of 的使用主要是针对多表关联的时候，如果不使用of，对两个表涉及到的行都将锁住，使用of可以指定锁定哪个表，
32. 例如：select a.MOBILE,b.NAME from  connector a,student b
33. where a.STU\_ID=b.ID and a.MOBILE='13937134399' for updata of a.MOBILE
34. 这样的话student表中对应的行是不加锁的，对connector一个表中行加锁
35. 不使用两个表都加锁。
37. [NOWAIT]的使用是当锁冲突的时候提示的情况：
38. 当有LOCK冲突时会提示错误并结束STATEMENT而不是在那里等待.返回错误是"ORA-00054: resource busy and acquire with NOWAIT specified" ，如果不使用就会一直等待，直到锁释放之后执行。
39. 在页面上调试的时候由于异常处理不好，把数据锁住了没有提交，也没有rollback，遇到这样的情况的时候可以通过以下方式解决：
41. -----查看被锁对象的序列号、sid
42. SELECT o.owner,o.object\_name,o.object\_type,s.sid,s.serial#
43. FROM v$locked\_object l,dba\_objects o,v$session s
44. WHERE l.object\_id=o.object\_id
45. AND l.session\_id=s.sid
46. ORDER BY o.object\_id,xidusn DESC
47. /
48. ------利用sid 和序列号删除
49. alter system kill session '243,10265';
50. 243是sid 10265是序列号
52. 所以在使用锁的时候一定要做好页面的异常控制，不然很容易出问题。
54. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
55. 数据库中锁类型的介绍：
56. 有两种基本的锁类型，排它锁（Exclusive Locks，即X锁）和共享锁（Share Locks，即S锁）。当数据对象被加上排它锁时，其他的事务不能对它读取和修改。加了共享锁的数据对象可以被其他事务读取，但不能修改。数据库利用这两种基本的锁类型来对数据库的事务进行并发控制。
57. 其中DML锁（data locks，数据锁），用于保护数据的完整性
58. DML锁的目的在于保证并发情况下的数据完整性，DML锁主要包括TM锁和TX锁，其中TM锁称为表级锁，TX锁称为事务锁或行级锁。
60. 在此分享的简单的select  for update应该属于DML锁中的TX锁，属于行级锁.