事务：

1. ACID

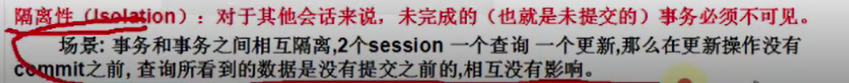
原子性

一致性

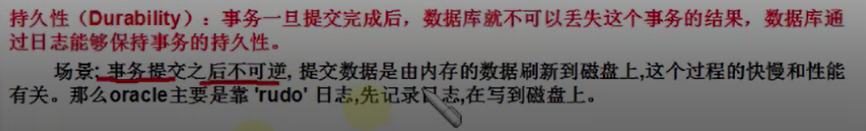
持久性

隔离性

1. 9：00 用户A执行一条select语句，那他看到到就是9:00那一刻数据库中所有的数据，9:01分的他都看不到，如果他select语句结果很大，花了十分钟才给他结果，结果也是9点那一刻的数据
2. 事务的隔离性：



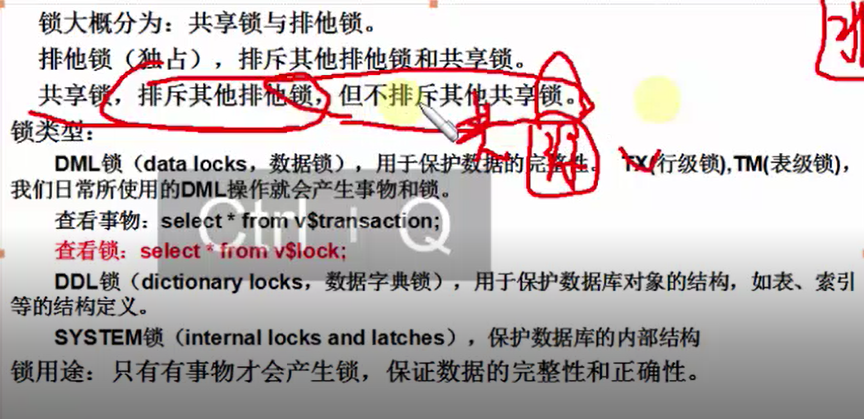
1. 持久性。事务已经提交。数据库就不能丢失这个事务的结果，数据库通过日志能够保证事务的持久性





查看oracle数据库事务select \* from v$transaction，数据库语句一提交，事务就结束了

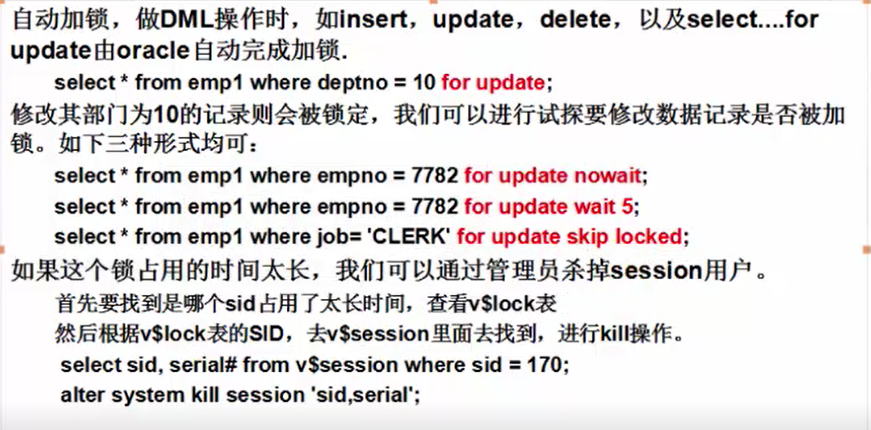
1. 锁的概念：



1. 自动加锁模式

Select \* from emp where … for update





1. Oracle自动解决死锁问题

Update a set id =1000 where id =100

Update a set id = 2000 where id =200

Update a set id = 3000 where id = 200

Update a set id = 4000 where id = 100

看到检测到死锁，然后这条记录被回滚

1. 索引

空间换性能 B树索引

一般sql优化的几个方案

1 索引

2 分区

3 物化视图

4 并行查询

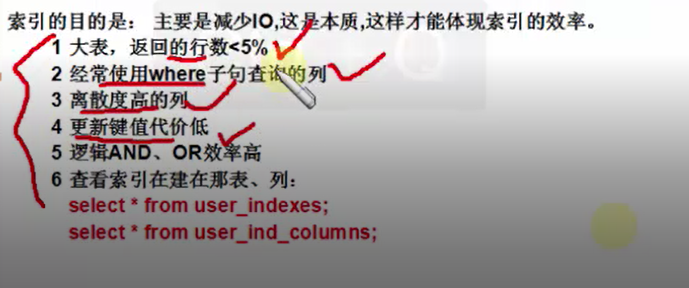
索引可以优化查询但是不能优化DML操作，插入和更新效率和性能上会降低，软件设计就是两者之间的权衡，空间换时间；

如果SQL语句仅访问被索引的列，数据库只需从索引读取数据，而不用读取表

如果该语句同时还要访问除索引的列之外的列，那么，数据库会使用rowid来查找表中

的行，通常，为检索表数据，数据库以交替的方式先读取索引块，然后读取相应的表块

建索引的目的：



查询索引的语句：select \* from user\_indexes

常用的索引：唯一索引（唯一不重复）一般索引（可以重复）和组合索引（两个或者更多字段做索引）

一般索引

Create index tttttt on em1(empno)

唯一索引

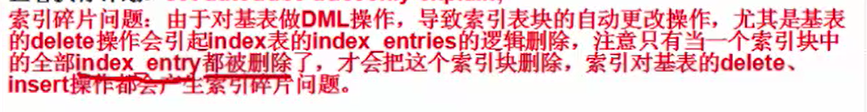
Create unique index tttttt on em1(empno)

组合索引

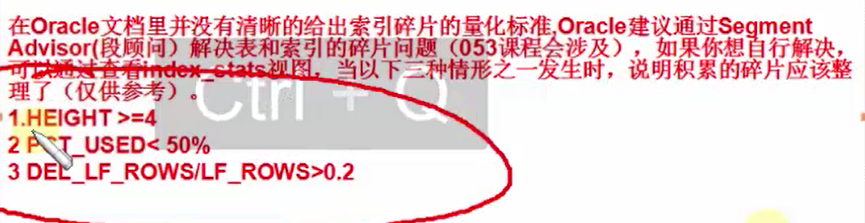
Create unique index tttttt on em1(empno,deptno)

索引碎片的问题：

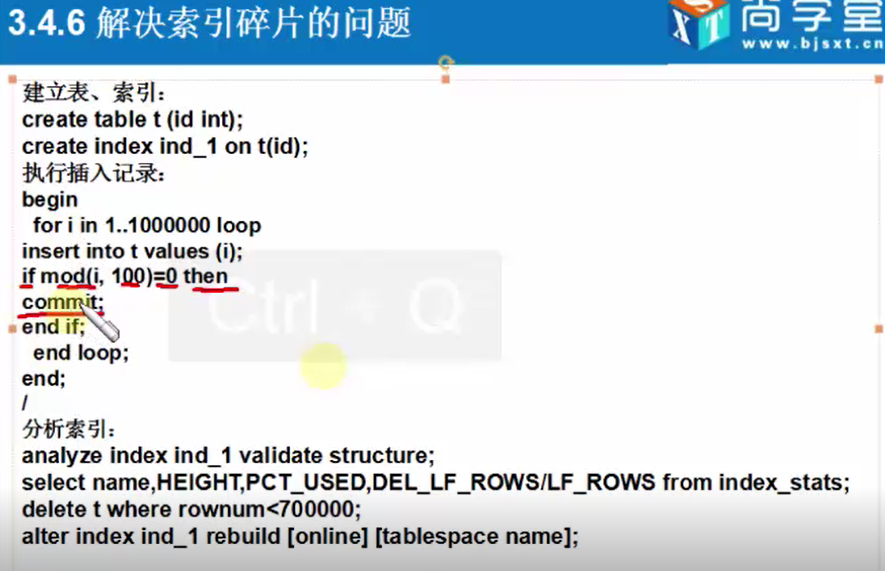
1. 更新，插入，删除，长时间（2-3年）导致索引变慢 2s ->4,5s耗时



1. 索引碎片整理



1. 索引



索引分析：

analyze index INT\_1 validate structure

select \* FROM INDEX\_STATS

alter index int\_1 rebuild online