

1、行级锁

我们开启一个事务：

该事务先执行一条查询sql语句，该SQL末尾加上for update后，所查询出来的每一行记录数据在该事务结束前没法被另一个事务中的修改操作修改，当这个事务结束后，其他事务才能修改之前被锁住的数据，需要事务排队

```
mysql> select * from emp where job = 'MANAGER' for update;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL      | COMM | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7566  | JONES | MANAGER  | 7839 | 1981-04-02 | 2975.00  | NULL | 20     |
| 7698  | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850.00  | NULL | 30     |
| 7782  | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450.00  | NULL | 10     |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (1.73 sec)
```

--注意行级锁，锁住的是查询出来的数据中那一行的所有字段数据，并不是锁住查询出来的某些字段数据
像上面锁住该数据需要等待当前事务结束后才允许被其他事务修改的锁叫做悲观锁

--悲观锁：

对锁住的数据，事务间需要等待，一个事务结束，另一个事务才能修改数据

--乐观锁：

对于锁住的每一行数据，提供一个版本号，例如两个事务都要对某一行被锁数据进行修改，大家都拿到一个该行数据的一个一样的初始版本号，

当某个事务修改完成后，发现该行数据的版本号与之前拿到的一样，直接提交结束事务，此时该行数据版本号改变了；

当另一个事务较慢修改了准备提交，用它之前拿到的版本号配对时发现该行数据版本号改变了，就知道被修改过了，该事务执行回滚结束

乐观锁支持并发，事务也不需要排队，但需要一个版本号。