109 案例实战：千万级用户场景下的运营系统SQL调优（1）

今天开始我们正式进入MySQL的SQL性能优化的案例实战部分，我们一共将会讲解4个SQL优化案例，每个案例都会放在一周内通过三次文章来讲解，每个案例都会分为业务场景引入、SQL性能问题分析、SQL性能调优三个部分。

今天我们就开始讲解咱们的第一个案例，也就是**千万级用户场景下的运营系统的复杂SQL调优实战案例**

先说下这个案例的背景，简单来说，这是一个互联网公司的系统，这个互联网公司的用户量是比较大的，有百万级日活用户的一个量级。

在这个互联网公司里，有一个系统是专门通过各种条件筛选出大量的用户，接着对那些用户去推送一些消息的，有的时候可能是一些促销活动的消息，有的时候可能是让你办会员卡的消息，有的时候可能是告诉你有一个特价商品的消息。

总而言之，其实通过一些条件筛选出大量的用户，接着针对这些用户做一些推送，是互联网公司的运营系统里常见的一种功能，在这个过程中，比较坑爹，也比较耗时的，其实是筛选用户的这个过程。

因为这种互联网公司，我们已经说过了，用户是日活百万级的，注册用户是千万级的，而且如果还没有进行分库分表的话，那么这个数据库里的用户表可能就一张，单表里是上千万的用户数据，大概是这么一个情况。

现在我们来对运营系统筛选用户的SQL做一个简化，写出来给大家看个热闹，这个SQL经过简化看起来可能是这样的：

SELECT id, name FROM users WHERE id IN (SELECT user_id FROM users_extent_info WHERE latest_login_time < xxxxx)

上面的SQL语句是啥意思？给大家解释一下，它的意思就是说一般存储用户数据的表会分为两张表，一个表用来存储用户的核心数据，比如id、name、昵称、手机号之类的信息，也就是上面SQL语句里的users表

另外一个表可能会存储用户的一些拓展信息，比如说家庭住址、兴趣爱好、最近一次登录时间之类的，就是上面的users_extent_info表

所以上面的SQL语句的意思就很明显了，有一个子查询，里面针对用户的拓展信息表，也就是users_extent_info查询了一下最近一次登录时间小于某个时间点的用户，这里其实可以是查询最近才登陆过的用户，也可以查询的是很长时间没登录过的用户 ，然后给他们发送一些push，无论哪种场景，这个SQL都是适用的。

然后在外层的查询里，直接就是用了id IN字句去查询 id 在子查询结果范围里的users表的所有数据，此时这个SQL往往一下子会查出来很多数据，可能几千、几万、几十万，都有可能，所以其实一般运行这类SQL之前，都会先跑一个count聚合函数，看看有多少条，比如下面这儿样。

SELECT COUNT(id) FROM users WHERE id IN (SELECT user_id FROM users_extent_info WHERE latest_login_time < xxxxx)

然后内存里做一个小批量多批次读取数据的操作，比如判断如果在1000条以内，那么就一下子读取出来，如果超过1000条，可以通过LIMIT语句，每次就从这个结果集里查1000条数据，查1000条就做一次批量PUSH，再查下一波1000条。

这就是这个案例的一个完整的业务背景和讲解，那么当时产生的问题是什么呢?

很简单，就是在千万级数据量的大表场景下，上面的SQL直接轻松跑出来耗时几十秒的速度，所以说，这个SQL不优化是绝对不行了!

下次我们就来针对这个SQL，分析一下他的执行计划以及他的性能之所以差的问题所在。

End