

微信

2010 年底，马化腾召开腾讯主管以上级会议，作为一个项目主管你也列席会议，期间马总根据当前的局势做出了在 QQ 之外另外开发一款集合即时通讯与社交的产品：微信，需要大家竞争，哪个团队最先做出合格的产品，就能以微信的名义推向市场。与你们一同竞争的还有来自广州团队的张小龙。会上，马化腾提出了公司对于这款新产品的具体设想：

1. 产品的核心是用户(user)。每个用户有唯一的微信号(uid)，昵称(nickname)，所在位置经度 (longitude) 以及纬度 (latitude)。
2. 两个用户之间可以建立起朋友关系 (friend)。朋友关系记录哪两个人 (uid1,uid2) 是朋友，以及建立时间 (connect_time)，建立方式 (摇一摇，微信号，附近的人等) (type)，备注(note)等属性。两个用户只能建立一次朋友关系。一对好友关系有两条记录，所以 (uid1=1, uid2=2) 以及 (uid1=2, uid2=1) 表示 1 和 2 是好友，(uid1=1, uid2=2)这一条目中的 note 记录的是用户 1 对用户 2 的备注，反之记录的是用户 2 对用户 1 的备注。
3. 两个建立起朋友关系的用户之间可以发送信息(message)。每条信息是一个独立的实体，由唯一的消息号码 (mid) 表示，此外还有消息内容(content)，发送次数 (shared_count) 等属性。
4. 我们还需要记录发送信息 (send) 这一行为，记录谁 (uid_sender) 给谁 (uid_receiver) 发送了哪条信息 (mid)，此外还要记录发送时间(sent_time)，发送状态 (已发送，待发送) (status)等属性。**注意，因为存在转发等行为，同一条信息可以被发送多次，每一次的发送状态不一样。每发送一次就要把该信息的发送次数加一。一个人也可以给同一个人发送同一条信息多次，比如刷屏。。。**
5. 用户可以建立群组(groups)，每个群组有唯一的群号(gid)，群名称(gname)，建群时间(create_time)，群公告(bulletin)，每个群有一个唯一的群主(uid_owner)
6. 每个用户都可以加入某个群组(join)，加入群组之后，需要记录哪个用户(uid)加入哪个群(gid)以及加入该群的时间 (join_time)，这个用户可以修改针对这个群的一些特定设置。比如群备注(note)，自己在这个群里面的昵称(nickname)。一个用户只可以加入一个群组一次。
7. 在群里面的用户可以向群发送信息 (group_send)。也需要记录谁 (uid) 给哪个群 (gid) 发送了哪条信息 (mid)，此外还要记录发送时间(sent_time)属性。注意同一条信息可以发给人，也可以发给群，发送给群的消息不存在待发送的情况，因此不用记录发送状态。一个人也可以在同一个群里发送同一条信息多次。
8. 微信还支持朋友圈功能(moment)。每条朋友圈有唯一的朋友圈 id (moid)，以及相应的朋友圈内容。
9. 每个朋友圈只能由一个用户发布，每个用户可以发布多个朋友圈。每次发表需要记录发表时间(post_time)，发表方式 (公开，仅好友可见，仅自己可见) (type)等属性。

有了朋友圈，群，好友这些概念，我们需要通过 SQL 来实现各种功能：

1. 获取与用户 1 互为好友的所有用户。返回 uid，昵称，备注 (note) 以及添加方式。将该查询保存为一个视图 user1_friend(uid,nickname,note)，方便后面的查询直接使用。(3 分)
2. 获取与 1 互为好友的用户的“公开”或是“仅好友可见”的朋友圈内容。返回该用户昵称，以及朋友圈内容，发布时间，按发布时间倒叙排列 (2 分)

3. 获取发送给用户 1 所在的所有群组的消息内容, 请返回群名称, 消息内容以及发送时间, 按群名称以及发送时间降序排列 (2 分)
4. 获取所有发给 1 的状态为"待发送"的所有消息, 按发送人以及发送时间降序排列。通过 sql 返回发送人名称, 信息内容以及发送时间 (2 分)
5. 用户 1 搜索与用户 6 之间的聊天记录, 通过 sql 返回他们之间互相发送的包含"晚安"关键词的所有"已发送"信息。返回信息内容, 发送人昵称。(3 分)
6. 上述两个查询时间较长, 请通过一个索引加速。请尝试在不同的列或者列的组合上建立索引, 然后比较查询速度的改变, 找出你认为索引效率最高的索引方式, 并且回答为什么。(3 分)
7. 在加入一个群的时候如果一个人与所有其他群成员之间没有好友关系我们会收到提示, 请利用 sql 实现这个功能, 返回用户 1 与群 1 中所有群员之间互为好友关系的人的数量。(2 分)
8. 请实现添加附近的人的功能, 对于用户 1, 通过经纬度计算与其他人的距离, 请返回所有距离用户 1 小于 10 并且当前与 1 没有好友关系的人。请返回用户昵称及与用户 1 的距离。注意: A 与 B 之间的距离可以直接用经纬度计算欧几里得距离, 不需要投影什么的。计算欧几里得距离的时候需要用到开方操作, 但是我们用的 Sqlite DBBrowser 在编译 sqlite 的时候没有定义需要包含数学函数, 因此不支持 sqrt()开方操作。所以所有的距离都不用开方。也就是返回距离的平方大于 100 的就行了。(3 分)
9. 用户 0 删除与用户 1 之间的好友关系需要一下几步: (5 分)
 - a) 将用户 0 与 1 之间互相发送过的信息的发送次数减一。(2 分)
 - b) 删除 shared_count 等于 0 的所有信息。(1 分)
 - c) 删除用户 0 与用户 1 之间的信息发送记录。注意这里删除的是发送信息记录, 而不是信息本身。(1 分)
 - d) 删除用户 0 与用户 1 之间的好友记录 (1 分)