计算机科学与技术学院 数据库系统 课程实验报告

|  |  |  |
| --- | --- | --- |
| 实验题目：综合实验4 | | 学号：202300130183 |
| 日期：2024/12/17 | 班级：23级人工智能 | 姓名：宋浩宇 |
| Email：2367651943@qq.com | | |
| 实验软件和硬件环境：  实验软件：  系统：Windows 11 家庭中文版23H2 22631.4317  编辑器：Visual Studio Code  数据库：SQL Server 2022  数据库管理工具：Microsoft SQL Server Management Studio 18  硬件环境：  CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz  内存：32.0 GB (31.6 GB 可用)  磁盘驱动器：NVMe WD\_BLACKSN850X2000GB | | |
| 实验步骤：  第一题：  SQL：  CREATE VIEW USER1\_FRIEND AS      SELECT          FRIEND.UID2   AS UID,          USER.NICKNAME,          FRIEND.NOTE,          FRIEND.TYPE      FROM          USER          JOIN FRIEND          ON USER.UID = FRIEND.UID2      WHERE          (USER.UID, 1) IN (              SELECT                  UID1,                  UID2              FROM                  FRIEND          )          AND (1, USER.UID) IN (              SELECT                  UID1,                  UID2              FROM                  FRIEND          )          AND FRIEND.UID1 = 1;  结果：  结果图1-1  结果图1-2  第二题：  SQL：  SELECT      USER1\_FRIEND.NICKNAME,      MOMENT.CONTENT,      MOMENT.POST\_TIME  FROM      USER1\_FRIEND      JOIN MOMENT      ON MOMENT.UID = USER1\_FRIEND.UID  WHERE      MOMENT.TYPE = "公开"      OR MOMENT.TYPE = "仅好友可见"  ORDER BY      MOMENT.POST\_TIME DESC;  结果：  结果图2-1  结果图2-2  第三题：  SQL：  SELECT      GROUPS.GNAME,      MESSAGE.CONTENT,      GROUP\_SEND.SENT\_TIME  FROM      GROUP\_SEND      JOIN MESSAGE      ON GROUP\_SEND.MID = MESSAGE.MID      JOIN GROUPS      ON GROUPS.GID = GROUP\_SEND.GID  WHERE      (1, GROUPS.GID) IN (          SELECT              UID,              GID          FROM              JOINGROUP          WHERE              UID = 1      )  ORDER BY      GROUPS.GNAME,      GROUP\_SEND.SENT\_TIME DESC;  结果：  结果图3-1  结果图3-2  第四题：  SQL：  SELECT      USER.NICKNAME,      MESSAGE.CONTENT,      SEND.SENT\_TIME  FROM      SEND      JOIN MESSAGE      ON SEND.MID = MESSAGE.MID      JOIN USER      ON SEND.UID\_SENDER = USER.UID  WHERE      SEND.UID\_RECEIVER = 1      AND STATUS = "待发送"  ORDER BY      USER.NICKNAME,      SEND.SENT\_TIME DESC;  结果：  结果图4-1  结果图4-2  第五题：  SQL：  SELECT      MESSAGE.CONTENT,      USER.NICKNAME  FROM      MESSAGE      JOIN SEND      ON SEND.MID = MESSAGE.MID      JOIN USER      ON SEND.UID\_SENDER = USER.UID  WHERE      MESSAGE.CONTENT LIKE "%晚安%"      AND( (SEND.UID\_SENDER = 1      AND SEND.UID\_RECEIVER = 6)      OR (SEND.UID\_SENDER = 6      AND SEND.UID\_RECEIVER = 1) )      AND SEND.STATUS = "已发送";  结果：  结果图5-1  结果图5-2  第六题：  SQL：  CREATE INDEX IDX\_SEND\_RECEIVER\_STATUS ON SEND (UID\_RECEIVER, STATUS);  结果：  结果图6-1结果图6-2  原因解释：我尝试了多种索引，原本的思路是找到这两个sql的共有的查询的部分，用那一部分的索引来同时加速两个查询，但实际上这么做对于两个sql的加速效果都不明显，但是使用这个索引，着重加速第二个查询，效果就非常显著了，因为本身第一个查询用的时间也并不长，把第二个查询最大限度加速，整体上省的时间就最多了。  第七题：  SQL：  SELECT      COUNT(\*) AS CNT  FROM      GROUPS      JOIN JOINGROUP      ON GROUPS.GID = JOINGROUP.GID  WHERE      GROUPS.GID = '1'      AND JOINGROUP.UID IN (          SELECT              UID          FROM              USER1\_FRIEND      );  结果：  结果图7  第八题：  SQL：  SELECT      USER.NICKNAME,      (USER.LONGITUDE - 309)\*(USER.LONGITUDE-309) + (USER.LATITUDE-470)\*(USER.LATITUDE-470) AS DISTANCE  FROM      USER  WHERE      (USER.LONGITUDE - 309)\*(USER.LONGITUDE-309) + (USER.LATITUDE-470)\*(USER.LATITUDE-470) < 100      AND USER.UID NOT IN (          SELECT              UID1          FROM              FRIEND          WHERE              UID2 = 1          UNION          SELECT              UID2          FROM              FRIEND          WHERE              UID1 = 1      );  结果：  结果图8  第九题：  a)  SQL：  UPDATE MESSAGE  SET      SHARED\_COUNT = SHARED\_COUNT - 1  WHERE      MID IN (          SELECT              MID          FROM              SEND          WHERE              uid\_sender=0              AND uid\_receiver = 1          UNION          SELECT              MID          FROM              SEND          WHERE              uid\_sender=1              AND uid\_receiver = 0      );  结果：  结果图9-a  b)  SQL：  DELETE FROM MESSAGE  WHERE      SHARED\_COUNT = 0;  结果：  结果图9-b  c)  SQL：  DELETE FROM SEND  WHERE      (uid\_sender = 0      AND uid\_receiver = 1)      OR (uid\_sender = 1      AND uid\_receiver = 0);  结果：  结果图9-c  d)  SQL：  DELETE FROM FRIEND  WHERE      (UID1 = 0      AND UID2 = 1)      OR (UID1 = 1      AND UID2 = 0);  结果：  结果图9-d  附加说明：由于表自带的外键约束，我们实际的删除过程是删除好友记录->删除消息发送记录->删除信息 | | |