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：

```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：

	uid2	nickname	note	type
1	0	sbycfmzog	sarnqjvsb	附近的人
2	6	oilolc	txcwlozt	摇一摇
3	7	lruma	gtipe	附近的人
4	22	bfrif	rilrr	微信号
5	23	bqysxk	jstxcul	微信号
6	29	wowryu	ztitlu	摇一摇
7	47	mqwvfbndb	fzcpm	附近的人

```
执行完成，无错误。
结果：203 行返回，耗时 45ms
在行 1:
SELECT
FRIEND.UID2,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
```

第二题：

SQL：

```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：

	nickname	content	post_time
1	ahjlreh	csjmgsrvtqventguoorotyymqwsijzpjqbqt…	999
2	nndupcc	hnnzhjocmxkxomlpixwzvlavxhhhxwlqdkclt	998
3	dplshmv	rbkagugiervuapjqwclagtmfvffqqbblrmrv…	996
4	uxziiv	iulrumdtgbhrlazpfqmhtlqce	996
5	gbuntfp	xacpxbfqepdtigvbrgttosjnlnwcsqsupdjx…	994
6	qbydpagxl	dyyenccwpmifmvynvwjikihjvfcngqvfpfay…	993
7	fkhdduk	dpkkilbsugsyaamsbvwitxnqdxoskxnnlocc…	993

```
执行完成，无错误。
结果：888 行返回，耗时 15ms
在行 1:
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;
```

第三题：

SQL：

```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：

	gname	content	sent_time
1	htbdloshcqogi	tyzsb	999
2	htbdloshcqogi	paqwgiv	999
3	htbdloshcqogi	ygeisq	999
4	htbdloshcqogi	xedbklvftozjnggzh	999
5	htbdloshcqogi	xsj	999
6	htbdloshcqogi	niwde	999
7	htbdloshcqogi	marmrqhqxmharcaifv	999

```
执行完成，无错误。
结果：8740 行返回，耗时 172ms
在行 1:
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;
```

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：

	nickname	content	sent_time
1	afllhtbix	qjxfeq	997
2	afllhtbix	wvn	995
3	afllhtbix	cnudydswqxdaz	991
4	afllhtbix	b	991
5	afllhtbix	wu	990
6	afllhtbix	rgsemgejdxq	987
7	afllhtbix	launb	986

```
执行完成，无错误。
结果：8752 行返回，耗时 512ms
在行 1:
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;
```

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：

	content	nickname
1	hghou晚安e	sonqoa
2	aygqqcoppocrfctfi晚安wop	sonqoa
3	omeshu晚安mclatzuqlscqxnlmc	oilolc
4	dobsxjapfbj晚安kqmtnleqtpqxdzcf	oilolc
5	u晚安hwwtlcnkslyoxs	oilolc
6	bmcmbfs晚安wkgregf	oilolc

执行完成，无错误。
结果：6 行返回，耗时 565ms
在行 1:

```
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"
```

第六题：

SQL：

```
CREATE INDEX IDX_SEND_RECEIVER_STATUS ON SEND (UID_RECEIVER, STATUS);
```

结果：

执行完成，无错误。
结果：8752 行返回，耗时 27ms
在行 1：

```
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;
```

```
执行完成，无错误。
结果：6 行返回，耗时 11ms
在行 1:
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";
```

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：

```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：

	cnt
1	11

第八题：

SQL：

```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：

	nickname	distance
1	sonqoa	0
2	qfjavzte	72

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：

```
执行完成，无错误。
结果：查询执行成功。耗时 4ms，120 行数据受影响
在行 1:
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
    );
```

b)

SQL：

```
DELETE FROM MESSAGE

WHERE

   SHARED_COUNT = 0;
```

结果：

```
执行完成，无错误。
结果：查询执行成功。耗时 823ms，32 行数据受影响
在行 1:
DELETE FROM MESSAGE
WHERE
     SHARED_COUNT = 0;
```

c）

SQL：

```
DELETE FROM SEND
WHERE
   (uid_sender = 0
   AND uid_receiver = 1)
   OR (uid_sender = 1
   AND uid_receiver = 0);
```

结果：

```
执行完成，无错误。
结果：查询执行成功。耗时 4ms，127 行数据受影响
在行 1:
DELETE FROM SEND
WHERE
     (uid_sender = 0
     AND uid_receiver = 1)
     OR (uid_sender = 1
     AND uid_receiver = 0);
```

d)

SQL：

```
DELETE FROM FRIEND
WHERE
    (UID1 = 0
    AND UID2 = 1)
    OR (UID1 = 1
    AND UID2 = 0);
```

结果：

```
执行完成，无错误。
结果：查询执行成功。耗时 0ms，2 行数据受影响
在行 1:
DELETE FROM FRIEND
WHERE
     (UID1 = 0
     AND UID2 = 1)
     OR (UID1 = 1
     AND UID2 = 0);
```

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