

实验3 XDSQL数据库优化

实验环境

云主机 2C4G

操作系统 windows server 2016

实验目的

在DBMS原型数据库开发中，select语句功能实现是较为复杂的，涉及到单表查询、多表查询、条件查询和嵌套查询等功能的实现。在以往大三下学期课程编译原理的作业——实现一个简易的DBMS数据库系统中，大多数学生只实现了单表查询和条件查询，多表查询只支持两张表的连接查询。为了提高学生动手能力，加深学生对笛卡尔积和多表查询的理解，本次实验在XDSQL数据库源代码的基础上，提供给学生一款只支持两张表连接查询的数据库系统XDSQL_for_select和其源代码，由学生将其改为支持多表查询的数据库系统。

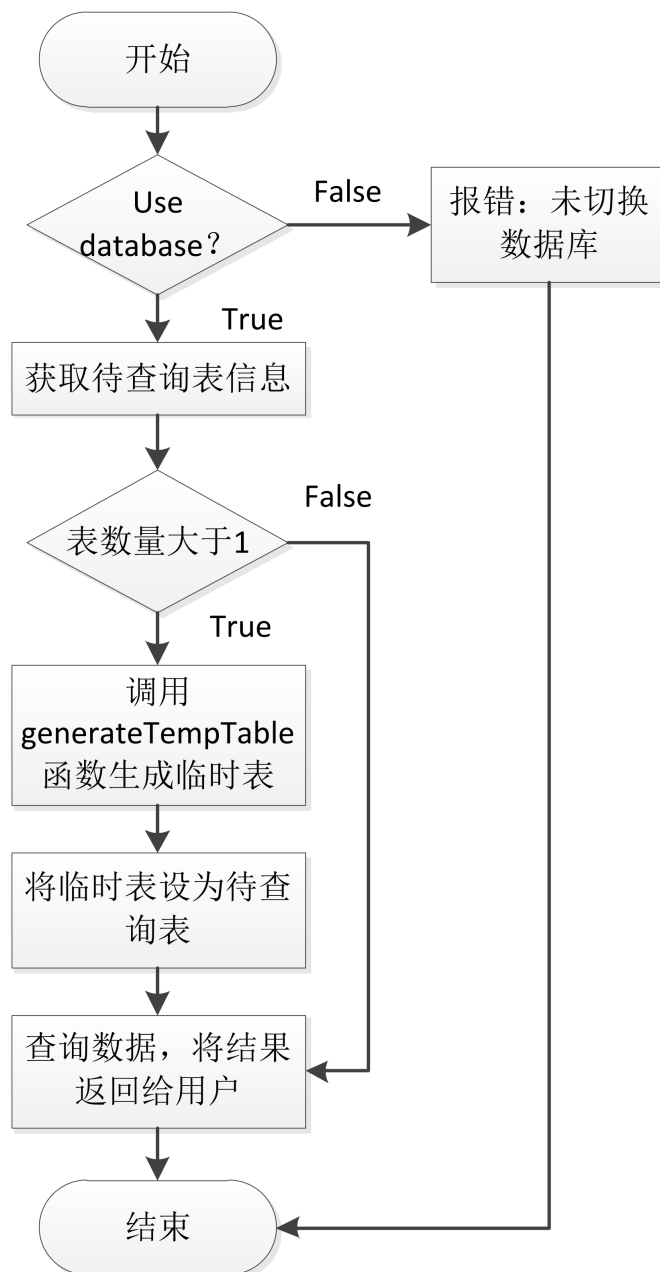
实验内容

XDSQL_for_select数据库系统是对XDSQL数据库修改得到的，其只支持以下SQL语句：

```
use database [数据库名称];
show databases;
show tables;
select * from [表名];
select * from [表1名],[表2名];
```

该数据库不支持创建数据库、删除数据库、创建表、删除表、插入、删除、更新等操作，目前只支持单表查询和双表查询，不支持条件查询和嵌套查询。

在XDSQL_for_select源代码中，负责查询的代码位于XDSQL_for_select/QueryProcessor.cpp中select函数内，其功能流程图如下：



当执行多表查询时，程序首先对待查询的多张表进行笛卡尔积操作，将结果保存到临时表中，随后基于临时表进行查询操作，最后删除临时表。生成临时表的函数名为generateTempTable，其中调用tablejoin函数用于多张表的笛卡尔积操作，并将结果存入临时表。目前tablejoin只支持两张表笛卡尔积操作。本次实验的目的便是修改tablejoin函数，使其支持多张表笛卡尔积操作。

涉及到的结构体说明：

tableNode结构体：具体定义见MetadataManager.h文件，记录数据表的元数据信息，包括表名、表内列数、一行数据的大小（单位：字节）和指向下一个tableNode结构体的指针。

storageNode结构体：具体定义见StorageEngine.h文件，可以将其视为指向数据表内数据行的指针。

涉及到的函数说明：

```
void tablejoin(tableNode* tablenode, char* row, int index, storageNode* sp);
```

负责对多张表进行笛卡尔积操作，并将结果通过sp写入临时表中。其传入参数如下：

tablenode：数据表节点，记录待查询数据表基本信息。

row：字节数组，作为临时数据行，负责笛卡尔积过程中数据的暂时存储。

index：字节数组指针，负责记录下一张表数据行数据在row字节数组中存储的首地址。

sp：临时数据表的数据行指针。

```
void open(storageNode* sp, char* tablename, int rowlength);
```

打开tablename数据表，并用sp指向该数据表内的数据行。其中tablename为数据表名称，rowlength为数据表内一行数据所占字节数。

```
void rnd_init(storageNode* sp);
```

初始化sp指针，使其指向数据表第一行数据。

```
bool isEOF(storageNode* sp);
```

判断数据表是否读取完毕，完毕返回true，否则返回false。

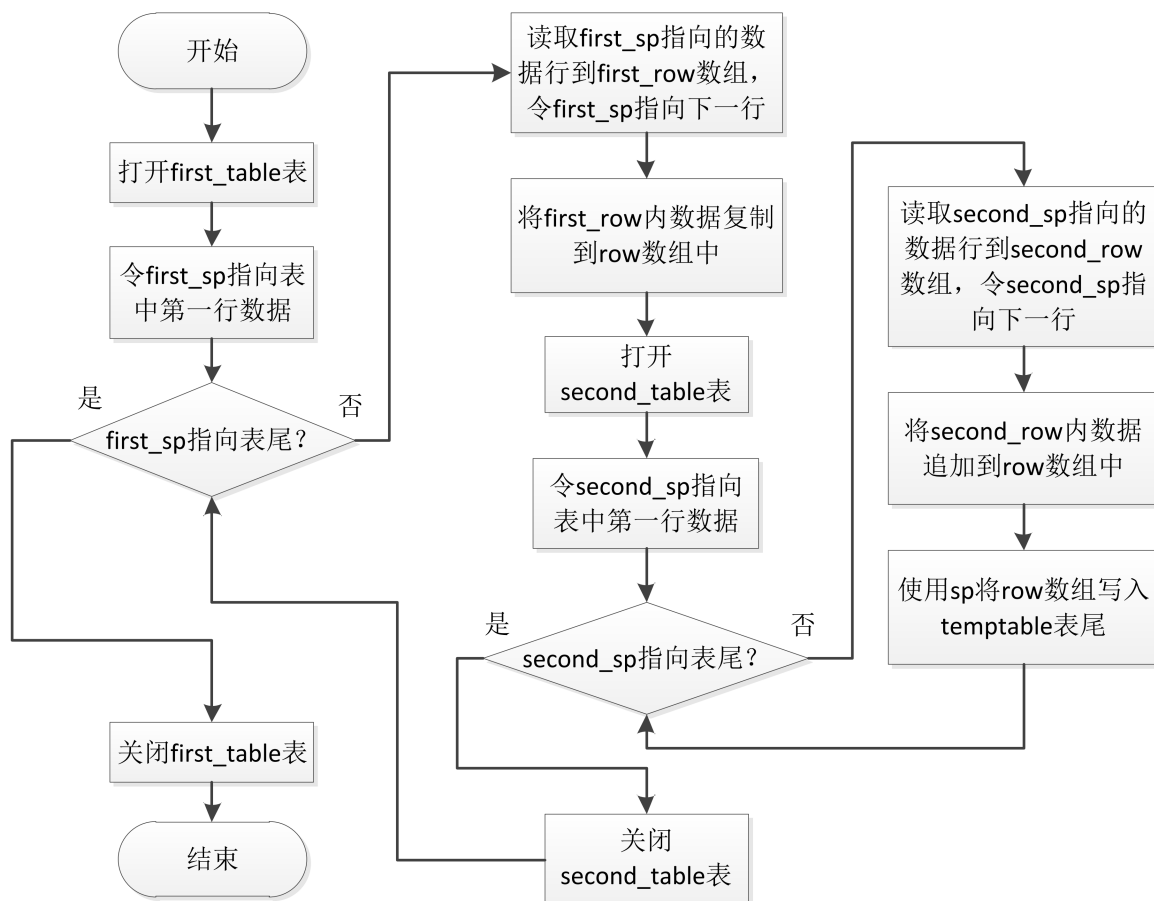
```
char* rnd_next(storageNode* sp);
```

读取sp指向的数据行数据，随后令sp指向下一行数据。返回值如果为NULL，说明该行数据已被删除，不对数据进行任何操作。

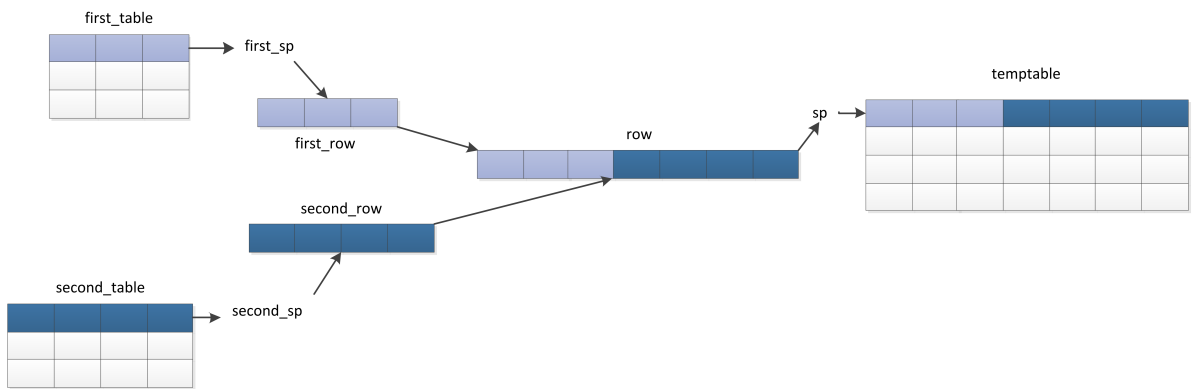
```
void write_row(storageNode* sp, char* row);
```

向sp指向的数据表表尾写入一行数据。row字节数组记录待写入的数据行数据。

tablejoin函数功能如下图所示：



其中first_table和second_table为待查询的表，temptable为临时表，是待查询表笛卡尔积后的结果表。下图展示了临时表第一行数据生成过程：



现需要同学们对tablejoin函数进行修改，使该函数具有多张表笛卡尔积功能，使newSQLdemo数据库支持多表查询。

实验步骤

本次实验在windows server 2016操作系统下进行，编译工具为VS2019，具体步骤如下：

1. 申请云主机，点击创建资源，获得ECS公网IP，登录名和密码

体验手册

云产品资源

实验报告

体验云账号，创建资源后生成

展开

ECS服务器

ECS公网地址:

47.101.195.21

ECS登录名:

administrator

登录密码:

Er7Vk6Gt6A

ECS实例ID:

i-uf6dr78if7esan0b1r7i

IP白名单:

0.0.0.0/0

地域:

华东 2 (上海)

2. 打开开始菜单，搜索远程桌面连接并运行，依次填写连接信息



3. 点击连接，登录到ECS服务器。第一次连接可能会出现连接失败，等待一两分钟后再次连接即可登录
4. 将Git-2.31.1-64-bit.exe复制到云主机（直接复制即可）
5. 安装git，安装步骤直接Next即可
6. 运行cmd，输入以下指令，将XDSQL_for_select数据库源代码下载到云主机

```
cd c:\
mkdir code
cd code
git config --global --unset http.proxy
git clone https://github.com/flytigerman/XDSQL_for_select.git
```

执行结果如图所示

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>cd c:\

c:\>mkdir code

c:\>cd code

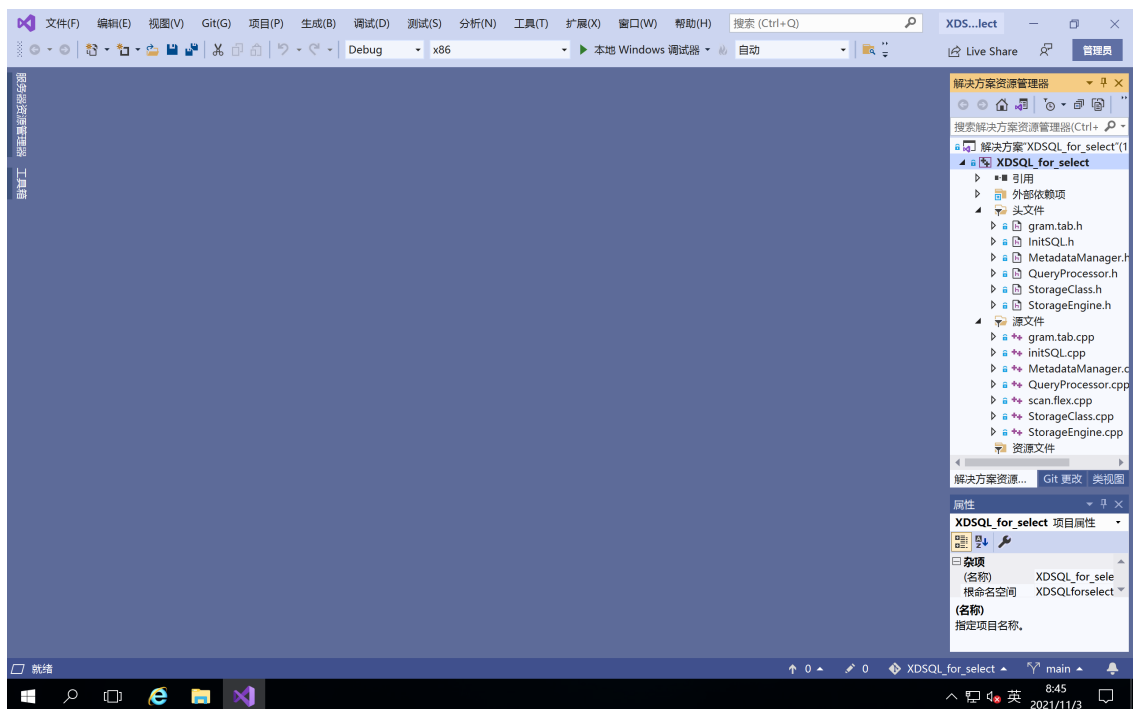
c:\code>git config --global --unset http.proxy

c:\code>git clone https://github.com/flytiger/XDSQL_for_select
Cloning into 'XDSQL_for_select'...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (102/102), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 102 (delta 23), reused 89 (delta 11), pack-reused 0
Receiving objects: 97% (99/102), 40.69 MiB | 7.75 MiB/s, done.
Resolving deltas: 100% (23/23), done.

c:\code>
```

7. 将VS2019复制到云主机，安装。安装时只安装使用C++的桌面开发即可，时间大约20分钟。

8. 打开VS2019，点击右上角的文件->打开->项目，打开XDSQL_for_select项目（路径为C:\code\XDSQL_for_select），执行结果如图所示



9. 打开QueryProcessor.cpp文件，对tablejoin函数进行修改，使其支持多表连接

10. 修改完毕后，将select函数中待查询表数量大于2的判断条件注释或删除，如下所示

```
if (tablelist->num > 1) {
    /*if (tablelist->num > 2) {
        printf("error: the number of tables is not more than 2");
        return;
    }*/ //删掉即可
    tablenode = generateTempTable(tablelist);
}
```

11. 点击运行，依次输入以下指令进行测试：

```
use database class_db;

select * from tb1,tb2,tb3,tb4;
```

显示以下结果，说明测试成功

```
D:\DBMS开发\Test_ZHG\SQLDemo\SQLDemo(Debug)\SQLDemo.exe
init SQL ok
SQL>>use database class_db;
use class_db
SQL>>select* from tb1,tb2,tb3,tb4;
tb1.col1      tb1.col2      tb2.col1      tb2.col2      tb3.col1      tb3.col2      tb3.col3      tb4.col1tb4.col2      tb4.col3
1      mike      mike      math      1      10      10      20      30
1      mike      mike      math      1      10      100      200      300
1      mike      mike      math      physics 2      3      10      20      30
1      mike      mike      math      physics 2      3      100      200      300
1      mike      lee      physics  math      1      10      10      20      30
1      mike      lee      physics  math      1      10      100      200      300
1      mike      lee      physics  physics 2      3      10      20      30
1      mike      lee      physics  physics 2      3      100      200      300
1      mike      andi     math      math      1      10      10      20      30
1      mike      andi     math      math      1      10      100      200      300
1      mike      andi     math      physics 2      3      10      20      30
2      andi     mike      math      physics 2      3      100      200      300
2      andi     mike      math      math      1      10      10      20      30
2      andi     mike      math      math      1      10      100      200      300
2      andi     mike      math      physics 2      3      10      20      30
2      andi     mike      math      physics 2      3      100      200      300
2      andi     lee      physics  math      1      10      10      20      30
2      andi     lee      physics  math      1      10      100      200      300
2      andi     lee      physics  physics 2      3      10      20      30
2      andi     lee      physics  physics 2      3      100      200      300
2      andi     andi     math      math      1      10      10      20      30
2      andi     andi     math      math      1      10      100      200      300
2      andi     andi     math      physics 2      3      10      20      30
2      andi     andi     math      physics 2      3      100      200      300
3      lee      mike      math      math      1      10      10      20      30
3      lee      mike      math      math      1      10      100      200      300
3      lee      mike      math      physics 2      3      10      20      30
3      lee      mike      math      physics 2      3      100      200      300
3      lee      lee      physics  math      1      10      10      20      30
3      lee      lee      physics  math      1      10      100      200      300
3      lee      lee      physics  physics 2      3      10      20      30
3      lee      lee      physics  physics 2      3      100      200      300
3      lee      andi     math      math      1      10      10      20      30
3      lee      andi     math      math      1      10      100      200      300
3      lee      andi     math      physics 2      3      10      20      30
3      lee      andi     math      physics 2      3      100      200      300
Total row: 36
SQL>>
```

备注：可以尝试采用递归实现。

补充说明

XDSQL_for_select数据库中目前存在一个数据库class_db，其中包含四张数据表，分别是tb1、tb2、tb3、tb4，每张表内有两到三条数据行。初始时数据库双表查询结果如下：

```

init SQL ok
SQL>>use database class_db;
use class_db
SQL>>select* from tb1,tb2;
tb1.col1      tb1.col2      tb2.col1      tb2.col2
1      mike      mike      math
1      mike      lee      physics
1      mike      andi      math
2      andi      mike      math
2      andi      lee      physics
2      andi      andi      math
3      lee      mike      math
3      lee      lee      physics
3      lee      andi      math
Total row: 9
SQL>>select* from tb1,tb3;
tb1.col1      tb1.col2      tb3.col1      tb3.col2      tb3.col3
1      mike      math      1      10
1      mike      physics      2      3
2      andi      math      1      10
2      andi      physics      2      3
3      lee      math      1      10
3      lee      physics      2      3
Total row: 6
SQL>>select* from tb2,tb3;
tb2.col1      tb2.col2      tb3.col1      tb3.col2      tb3.col3
mike      math      math      1      10
mike      math      physics      2      3
lee      physics      math      1      10
lee      physics      physics      2      3
andi      math      math      1      10
andi      math      physics      2      3
Total row: 6
SQL>>select* from tb4,tb3;
tb4.col1      tb4.col2      tb4.col3      tb3.col1      tb3.col2      tb3.col3
10      20      30      math      1      10
10      20      30      physics      2      3
100      200      300      math      1      10
100      200      300      physics      2      3
Total row: 4
SQL>>_

```

多表查询结果显示：

```

SQL>>select* from tb1,tb2,tb3;
error: the number of tables is not more than 2
SQL>>select* from tb1,tb2,tb4;
error: the number of tables is not more than 2
SQL>>select* from tb1,tb2,tb3,tb4;
error: the number of tables is not more than 2
SQL>>

```