CS307 Database Project Report

Name: 傅伟堡(Weibao Fu)

SID:11812202

1. Title

How do DBMS makes it easier for us to manage data?

2. Introduction

We are living in an era of data explosion, with so much data that we need to pay more attention to the management of data. Also, many scientists are constantly looking for ways to manage data more efficiently. Currently, we usually use database management system to manage our data, which makes it very convenient and fast for us to manage data. However, it is not enough just to be able to use database management system, we also need to know how DBMS is convenient for us. So in this project, I will from the data loading, storage, query speed, insert speed, user privileges management, index and other aspects to explain how do DBMS makes it easier for us to manage data.

3. Experimental Design

3.1 Experimental data and envrionments

3.1.1 Data

Introduction

Douban movie is a Chinese website which allows users to share their comments and views. Users can also give them marks. This dataset collects more than 2 million comments of 28 movies in Douban Movie website. We use this dataset during our project.

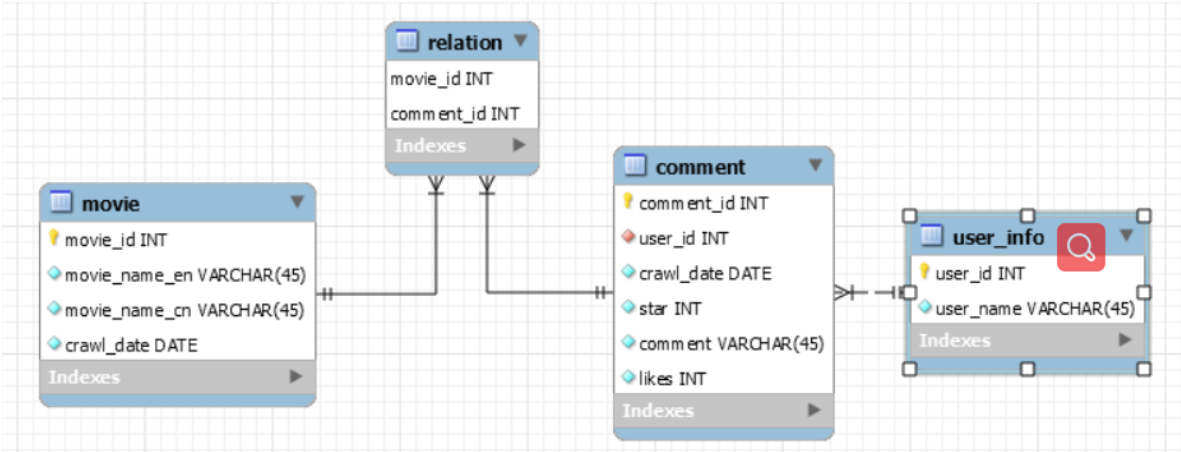
Description

In the original dataset, it contains a lot of columns. The content and description of each column is as follow:

Column Name	Description of Column
ID	The ID of the comment
Movie_Name_EN	The English name of the movie
Movie_Name_CN	The Chinese name of the movie
Crawl_Date	The date that the data are crawled
Number	The number of the comment
Username	The username of the account
Date	The date that the comment posted
Star	The star that user give to the movie (0-5)
Comment	The content of the comment
Like	The count of 'like' on the comment

Data Organization

Through my analysis, I can conclude that the column have the following relationship:
<Comment Table> Movie, User, Date, Star, Like are related to the comment.
<User Table> Comment is related to the user.
<Movie Table> Movie_Name_EN, Movie_Name_CN, Crawl_Date, Comment are realated to the movie.
Here I give ER graph to show my data organization:



We use the following clause to create each table.

```
create table movie(
    movie_id serial primary key,
    movie_name_EN varchar(100) not null ,
    movie_name_CN varchar(50) not null ,
    crawl_date date,
    unique(movie_name_EN,movie_name_CN,crawl_date)
);

create table user_info(
```

```
    user_id serial primary key ,
    user_name varchar(50) not null ,
    unique (user_name)
);

create table comment(
    comment_id serial primary key,
    user_id int not null
        constraint user_id references user_info,
    crawl_date date,
    star int ,
    comment varchar(500) not null ,
    likes int,
    check(star in (1,2,3,4,5))
);

create table relation
(
    movie_id int not null
        constraint movie_id
            references movie,
    comment_id int
        constraint comment_id
            references comment,
    unique (movie_id,comment_id),
    primary key (movie_id,comment_id)
);
```

Data Processing

In the process of my project, I think data processing is a very troublesome task. I need to transfer the original dataset to those satisfy my expectation. Since the dataset contains more than 2 million sample, it is impossible to do the data processing manually. After considering, I decide to use another programming language (Python) to help me to finish my work. Here I will show a part of my code to explain how I do this process.

1. To create the dataset of table (comment, user_info, movie)

```
# to create the dataset of table<comment>
import pandas

data = pandas.read_csv("F:/Important Files/Data Base/DMSC.csv")
user_dataFrame = pandas.DataFrame(data.drop_duplicates(subset=['Username'], keep="first", ignore_index=True))
user_dataFrame['Username'].to_csv("F:/Important Files/Data Base/User_info.csv", index=True, sep=',',
                                  encoding='utf_8_sig')
movie_dataFrame = pandas.DataFrame(data.drop_duplicates(subset=["Movie_Name_EN"], keep="first", ignore_index=True))
movie_dataFrame[["Movie_Name_EN", "Movie_Name_CN", "Crawl_Date"]].to_csv("F:/Important Files/Data Base/Movie_info.csv",
                                                                         index=True, sep=',', encoding='utf_8_sig')
comment_dataFrame = pandas.DataFrame(data)
comment_dataFrame[["Date", "Star", "Comment", "Like"]].to_csv("F:/Important Files/Data Base/Comment_info3.csv",
                                                              index=True, sep=',', encoding='utf_8_sig')
```

2. To create the dataset of table <relation>.

```
# create the dataset of table<relation>
import pandas

data_filename = "F:/Important Files/Data Base/DMSC.csv"
movie_filename = "F:/Important Files/Data Base/Movie_info.csv"
user_filename = "F:/Important Files/Data Base/User_info.csv"
comment_filename = "F:/Important Files/Data Base/Comment_info2.csv"
data = pandas.read_csv(data_filename)
m = pandas.read_csv(movie_filename)
u = pandas.read_csv(user_filename)
c = pandas.read_csv(comment_filename)
movie_dict = dict(zip(m['Movie_Name_CN'], m.iloc[:, 0]))
user_dict = dict(zip(u['Username'], u.iloc[:, 0]))
data['MovieId'] = data['Movie_Name_CN'].map(movie_dict)
data['CommentId'] = data.index
dataFrame = pandas.DataFrame(data)
dataFrame[['MovieId', 'CommentId']].to_csv("F:/Important Files/Data Base/Relation3.csv", index=False, sep=',',
                                           encoding='utf_8_sig')
```

In this way, I finish the data processing successfully.

3.1.2 DBMS

As recommended in this lecture, I use PostgreSQL during this project.

3.1.3 Programming language

In this project, I use Java to show the difference of DBMS and File. Besides, I use Python to do some neccessary data processing.

3.2 Experiments

3.2.1 Store the data into DBMS and File (Large data sets)

First of all, we need to store the data into our DBMS and File separately. We guess that there may have some difference between them, so we do an experiment to test whether there have any difference between DBMS and File. It is worthy to tell that I will use a large dataset to do the experiment.

3.2.2 Load the data from DBMS and File

After we stored the data successfully, we need to get the infomation of the dataset and to use it in our program. Such that the current thing we need to do is to load the data from DBMS and file. Also, I will load the data from DBMS and File separately to show the difference in loading data between DBMS and File.

3.2.3 Comparison between DBMS and File of Insertion

If we have loaded data, let we do some interesting things. That's to insert something into our database. We can compare the speed between the DBMS and File to get more information. We should do many times and get the average time to reduce the error and to increase the correctness of the experiment.

3.2.4 Comparison between DBMS and File of Query

Also, we can query something in our DBMS and File separately. Then, we can compare the speed of DBMS and File. By doing this, we can get the information of speed of Insertion and querying. Next, we can guess the data structure of DBMS. We should do many times and get the average time to reduce the error and to increase the correctness of the experiment

3.2.5 User privileges management

We will use this part to show that how DBMS convient to manage user privileges. It is a very strong tools to manage the data.

3.2.6 Rich query set

Actually, DBMS provides us a very strong function. We can easily use it to create a query clause to retrieve the data from database. Also, it provides us a lot of common aggregate functions, filters and orders.

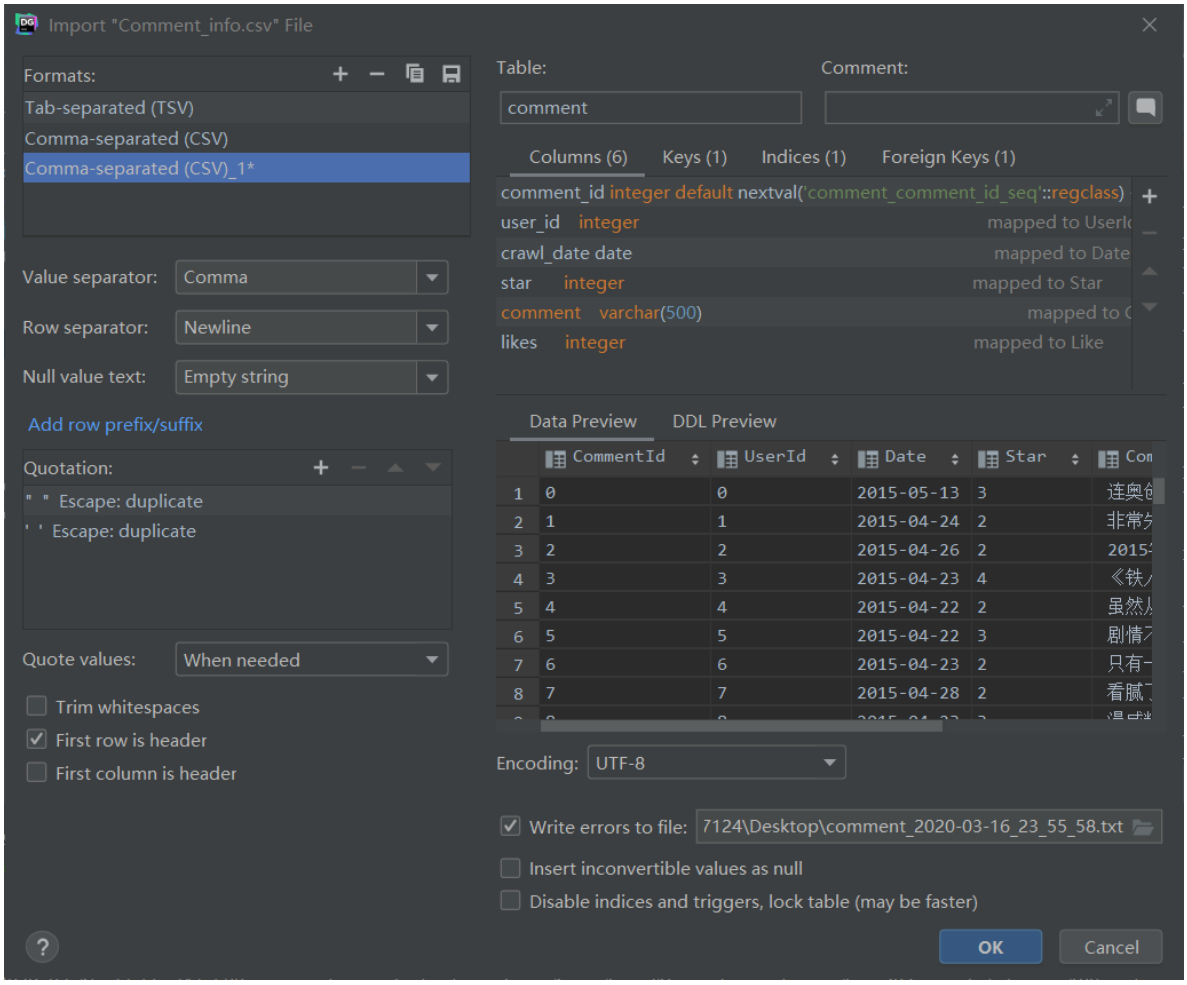
3.2.7 Database index and File IO

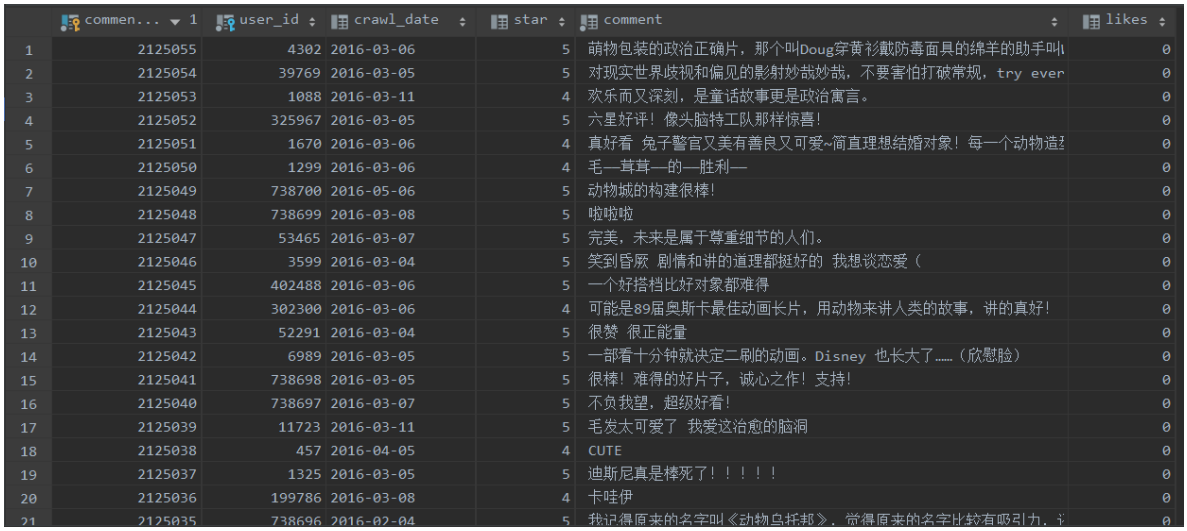
Actually, DBMS can query data faster than File due to the structure in DBMS and File is different. So I will use this part to analyze the difference between them and tell the principle of Database index and File IO.

3.3 Experiments Result

3.3.1 Store the data into DBMS and File (Large data sets)

If we store the data into DBMS, we can store it very without difficulties. To my convience, I use Datagrip to help me do this. The screenshot show that we store 2 million data successfully.





	commen... 1	user_id	crawl_date	star	comment	likes
1	2125055	4302	2016-03-06	5	萌物包装的政治正确片，那个叫Doug穿黄衫戴防毒面具的绵羊的助手叫	0
2	2125054	39769	2016-03-05	5	对现实世界歧视和偏见的影射妙哉妙哉，不要害怕打破常规，try ever	0
3	2125053	1088	2016-03-11	4	欢乐而又深刻，是童话故事更是政治寓言。	0
4	2125052	325967	2016-03-05	5	六星好评！像头脑特工队那样惊喜！	0
5	2125051	1670	2016-03-06	4	真好看 兔子警官又美有善良又可爱~简直理想结婚对象！每一个动物造	0
6	2125050	1299	2016-03-06	4	毛—茸茸—的—胜利—	0
7	2125049	738700	2016-05-06	5	动物城的构建很棒！	0
8	2125048	738699	2016-03-08	5	啦啦啦	0
9	2125047	53465	2016-03-07	5	完美，未来是属于尊重细节的人们。	0
10	2125046	3599	2016-03-04	5	笑到昏厥 剧情和讲的道理都挺好的 我想谈恋爱（	0
11	2125045	402488	2016-03-06	5	一个好搭档比好对象都难得	0
12	2125044	302300	2016-03-06	4	可能是89届奥斯卡最佳动画长片，用动物来讲人类的故事，讲的真好！	0
13	2125043	52291	2016-03-04	5	很赞 很正能量	0
14	2125042	6989	2016-03-05	5	一部看十分钟就决定二刷的动画。Disney 也长大了……（欣慰脸）	0
15	2125041	738698	2016-03-05	5	很棒！难得的好片子，诚心之作！支持！	0
16	2125040	738697	2016-03-07	5	不负我望，超级好看！	0
17	2125039	11723	2016-03-11	5	毛发太可爱了 我爱这治愈的脑洞	0
18	2125038	457	2016-04-05	4	CUTE	0
19	2125037	1325	2016-03-05	5	迪斯尼真是棒死了！！！！！	0
20	2125036	199786	2016-03-08	4	卡哇伊	0
21	2125035	738696	2016-02-04	5	我记得原来的名字叫《动物乌托邦》，觉得原来的名字比较有吸引力，i	0

But if we want to store the data into file, we will meet some difficulties. Since the dataset is more than 2 million, we cannot store all the data. I use a simple method to check how many data I store.

```java
public String check() {
    int row=0;
    StringBuilder sb = new StringBuilder();
    try (BufferedReader bufferedReader = new BufferedReader(new FileReader( fileName: "E:\\Program Files (x86)\\Java\\V
        bufferedReader.readLine();
        while ((bufferedReader.readLine()) != null) {
            row++;
        }
        sb.append("The dataset have "+row+" rows\n");
    } catch (IOException e) {
        e.printStackTrace();
    }
    return sb.toString();
}
```

From the result, we can notice that only a part of data can be store in file.

```
The dataset have 1046706 rows

Process finished with exit code 0
```

From the experiment result, we can conclude that DBMS can conveniently and easily to store the data into our database without losing data. However, we will get a incomplete data if we use File, that is to say, we cannot store the data into our database totally. In this case, DBMS is better than File obviously.

3.3.2 Load the data from DBMS and File

We have shown that it is very easy to store the data into our DBMS. Also, we can use the data directly. I will show how I get the $< comment >$ information.

```
select *
from comment;
```

Output | db_project1.public.comment

1-500 of 501+ | Tx: Auto | DDL

	comment_id	user_id	crawl_date	star	comment	likes
1	1019681	13767	2016-11-19	2	结尾跳出了姐妹撕逼的圈子，超越了一般青春片的格局，可是片子里的道理很虚幻，很难打	0
2	1019682	59512	2016-09-25	4	还不错的片子	0
3	1019683	470443	2016-10-06	4	当你努力成为我的时候，我却走在了你的人生路上。我不是安生也没有七月。	0
4	1019684	282920	2016-09-15	3	好坏参半 剧情反转 镜头很美 不过情节老套 有点矫情 台词略多	0
5	1019685	13305	2016-12-07	4	整部片子节奏掌控得不错，冬雨妹子演技大爆发，结尾比较巧妙	0
6	1019686	409711	2016-11-08	4	感情处理的很细腻也比较真实，周冬雨和马思纯都超水平发挥，但整体还是逃不出浓浓的安	0
7	1019687	120413	2016-09-20	4	身后的女生哭的稀里哗啦的。	0
8	1019688	268402	2016-11-02	3	国内的青春片就是恋爱 闺蜜胡斯 没点别的了	0
9	1019689	44799	2016-09-21	3	好看，喜欢女生之间的感情	0
10	1019690	13515	2016-11-29	4	就喜欢女主角丑丑的电影，被我低估了的作品，导演还是很有想法，至少努力想展现生活和	0

Obviously, we use only a simple instruction. But if we want to get the information from the file, we will meet a huge difficulties.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
    at Project1.FileManipulation.allMovie(FileManipulation.java:38)
    at Project1.Client.main(Client.java:9)

Process finished with exit code 1
```

That's because when we the column `comment` can also exists `\n`. Since we use `bufferedReader.readLine()`, such that it make some mistakes when we load the data. My solution is to delete those samples. I know it is incorrert to delete samples when we do data analysis, but I still delete them because I think it is the most convient way to continue my experiment and deleting the sample have no effects in this experiment. I will show you how I select those data and delete them.

First, I use the following code to output the "problem" comment_id.

```
            try{
//                  if (!continentNames.contains(movieInfo[1])) {
                      sb.append(movieInfo[1]+"\t"+movieInfo[2]+"\t"+movieInfo[3]+"\n");
//                      continentNames.add(movieInfo[1]);
                    temp = movieInfo[0];
//                  }
            }catch (Exception e){
                System.out.print("\""+temp+"\",");
                continue;
            }
```

The output:

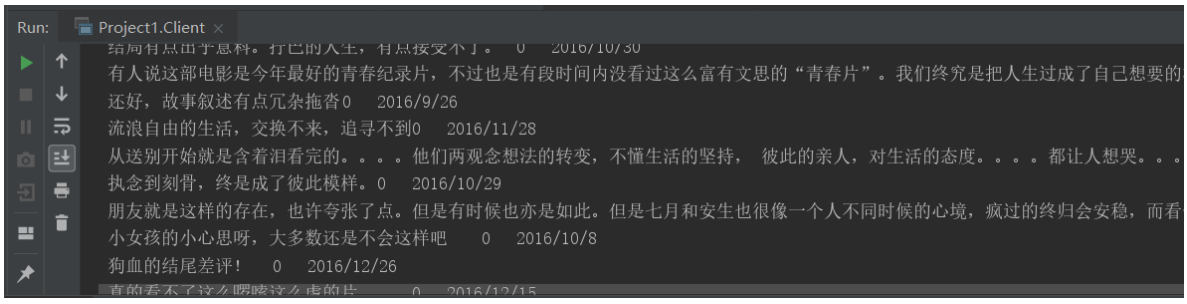
```
"575230","575285","575366","575366","575366","575422","575422","575422","575422","575437","575499","575515",
```

After doing this, I use Python again. And I use the following code to delete those samples directly.

```
import pandas

data_filename = "F:\DMSC2.csv"
data = pandas.read_csv(data_filename)
list = ["871530", "872218", "872218", "872806", "873092", "873529", "873749", "874100", "874549", "875142", "8762
data = data[-data["ID"].isin(list)]
dataFrame = pandas.DataFrame(data)
dataFrame.to_csv("F:\DMSC2.csv", index=False, sep=',', encoding='utf_8_sig')
```

By doing such a complex operation, we finish load the data from file successfully.



By doing this experiment, we can see the DBMS is so convenient to help us processing the data. On the other side, we will meet a huge difficulty when we use File because we should notice that the data set may include `\n`. Such that, we may need to do some data preprocessing. In this case, it is undoubtedly that DBMS is better than File.

3.3.3 Comparison between DBMS and File of Insertion

Now let we use the following code to insert a data into our DBMS.

```java
@Override
public int addOneMovie(String str) {
    getConnection();
    int result = 0;
    String sql = "insert into movie (movie_name_en,movie_name_cn,crawl_date) " +
            "values (?,?,?)";
    String movieInfo[] = str.split( regex: ";");
    try {
        PreparedStatement preparedStatement = con.prepareStatement(sql);
        preparedStatement.setString( parameterIndex: 1, movieInfo[0]);
        preparedStatement.setString( parameterIndex: 2, movieInfo[1]);
        java.util.Date date = new SimpleDateFormat( pattern: "yyyy-MM-dd").parse(movieInfo[2]);
        preparedStatement.setDate( parameterIndex: 3, new java.sql.Date(date.getTime()));
        System.out.println(preparedStatement.toString());
        long time1 = System.currentTimeMillis();
        result = preparedStatement.executeUpdate();
        long time2 = System.currentTimeMillis();
        System.out.println("**********************\nThe instruction TimeCost is:" + (time2 - time1) + " ms\n**********
    } catch (SQLException | ParseException e) {
        e.printStackTrace();
    } finally {
        closeConnection();
    }
    return result;
}
```

And we can get the following result:

```
**********************
The instruction TimeCost is:10 ms
**********************
```

If we want to using the following code to insert a data into our file:

```java
@Override
public int addOneMovie(String str) {
    try (FileWriter writer = new FileWriter( fileName: "E:\\Program Files (x86)\\Java\\Workspace\\DataBase\\src\\F
        String movieInfo[] = str.split( regex: ";");
        long time1 = System.currentTimeMillis();
        writer.write( str: movieInfo[0]+","+movieInfo[1]+","+movieInfo[2]+"\n");
        long time2 = System.currentTimeMillis();
        System.out.println("**********************\nThe instruction TimeCost is:"+(time2-time1)+" ms\n*******
    } catch (IOException e) {
        e.printStackTrace();
        return 0;
    }
    return 1;
}
```

Then, we can get the result:

```
**********************
The instruction TimeCost is:156 ms
**********************
```

After do 10 times, we get the average time of DBMS is 12.8ms and the average time of File is 134.6ms. From the result, we can clearly to see that the DBMS is faster than File.

3.3.4 Comparison between DBMS and File of Query

If we want to query the movie information by movie_id, we can also use DBMS and File.

If we use DBMS to query:

```
@Override
public String findCommentByMovie(int id) {
    getConnection();
    StringBuilder sb = new StringBuilder();
    String sql = "select movie_name_EN,movie_name_CN,comment\n" +
            "from movie m\n" +
            "    join relation r on m.movie_id = r.movie_id\n" +
            "    join comment c on r.comment_id = c.comment_id\n" +
            "where m.movie_id = " + id + ";";
    try {
        Statement statement = con.createStatement();
        sb.append("Movie_Name_EN\tMovie_Name_CN\tComment\n");
        long time1 = System.currentTimeMillis();
        resultSet = statement.executeQuery(sql);
        long time2 = System.currentTimeMillis();
        while (resultSet.next()) {
            sb.append(resultSet.getString( columnLabel: "movie_name_en") + "\t" + resultSet.getString( columnLabel: "movie_name_cn
        }
        sb.append("**********\nThe instruction TimeCost is:" + (time2 - time1) + " ms\n**********
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        closeConnection();
    }
    return sb.toString();
}
```

The result is as follow:

```
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  正正    2017-01-25  5    现在有很多的探险队，莫非也是在偷偷的盗墓吧。  0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  好高    2017-01-25  5    九层妖塔和九层妖楼是一个东东么？  0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  收旗卷  2017-01-25  5    胡八一为毛就选择了这条冒险之旅呢？0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  道合    2017-01-25  5    到目前为止，从九层妖塔的海报里推断出了胡八一杨萍、胡八一曹伟伟、
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  失桑    2017-01-25  5    请问这部电影适合15岁的小孩子看吗？十一期间我表弟要来我家玩，我想
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  倾巢而  2017-01-25  5    九层妖塔的海报都好气势！  0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  忘战    2017-01-25  5    姚晨胜在气场。 0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  断梗飞  2017-01-25  5    我是来看岳小凤的….什么时候出他的新剧照？ 0
Chronicles of the Ghostly Tribe九层妖塔    2017-01-25  放浪无  2017-01-25  5    怪兽一点也不萌怎么办？ 0
**********
The instruction TimeCost is:1652 ms
**********
```

If we use File to query:

```java
@Override
public String findMovieById(int id) {
    String movie;
    switch (id){
        case 0:
            movie = "复仇者联盟2";
            break;
        case 1:
            movie = "大鱼海棠";
            break;
//          case ...    for testing
        default: movie = "九层妖塔";
    }
    String line;
    StringBuilder sb = new StringBuilder();

    try (BufferedReader bufferedReader = new BufferedReader(new FileReader( fileName: "E:\\Program Files (x86)\\Java\\Workspace\\DataBase\\src
        bufferedReader.readLine();
        long time1 = System.currentTimeMillis();
        while ((line = bufferedReader.readLine()) != null) {
            String movieInfo[] = line.split( regex: ",");
            if (movie.equals(movieInfo[2])) {
                sb.append(movieInfo[1]+"\t"+movieInfo[2]+"\t"+movieInfo[3]+"\t");
                sb.append(movieInfo[5]+"\t"+movieInfo[6]+"\t"+movieInfo[7]+"\t");
                sb.append(movieInfo[8]+"\t"+movieInfo[9]+"\n");
            }
        }
        long time2 = System.currentTimeMillis();
        sb.append("**********\nThe instruction TimeCost is:"+(time2-time1)+" ms\n**********");
```

The result is as follow:

```
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   虐老兽  2015/9/3    5    秀才，秀才你在哪，你的郭女神回来了，再不来你的女神就要被怪
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   腰缠万  2015/9/3    4    求花式打怪招数~    0
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   推己    2015/9/2    4    姚晨最性感了  0
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   xiaoyide0711    2015/9/2    4    陆川？我是姚晨的粉丝，好像没看过她主演的电影 期待.
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   wubainian0229   2015/9/2    4    陆川的电影比较手法细腻，场面非常壮观，加上故事情节
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   xiecuiyi1126    2015/9/2    4    大黑牛没有头发依然很帅，尤其是亮剑那块儿。    0
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   露露    2015/9/2    5    陆川，他是一位叙事大师，像黑泽明或马丁-斯科塞斯，他善于在
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   怅然    2015/9/2    3    唐嫣其他片感觉都是傻白甜，但是在在鬼吹灯中看上去还是蛮有质
Chronicles of the Ghostly Tribe九层妖塔    2017/1/25   sunoom  2015/8/31   1    远观就是来悼原著的，如果质量好再来改评价  0
**********
The instruction TimeCost is:3595 ms
**********
```

After repeating the experiment 10 times, we get the average time of DBMS is 1823.5ms and the average time of File is 3681.2ms. We notice that though we have less data in our file, it costed more 2 times than the DBMS.

3.3.5 User privileges management

- **Create users and Drop users easily.**

We try to create a user called $user1$

```
db_project1=# create user user1;
CREATE ROLE
db_project1=# \password
输入新的密码:
再次输入:
```

Then we check the usertable, we can get that $user1$ is created successfully.

```
db_project1=# \du
                                   角色列表
 角色名称 |                                   属性                                   | 成员属于
----------+--------------------------------------------------------------------------+----------
 checker  | 超级用户                                                                 | {}
 postgres | 超级用户，建立角色，建立 DB，复制，绕过RLS                               | {}
 user1    |                                                                          | {}
```

We try to drop $user1$

```
db_project1=# drop user user1;
DROP ROLE
```

By checking the usertable, we can ensure $user1$ is deleted as expectation.

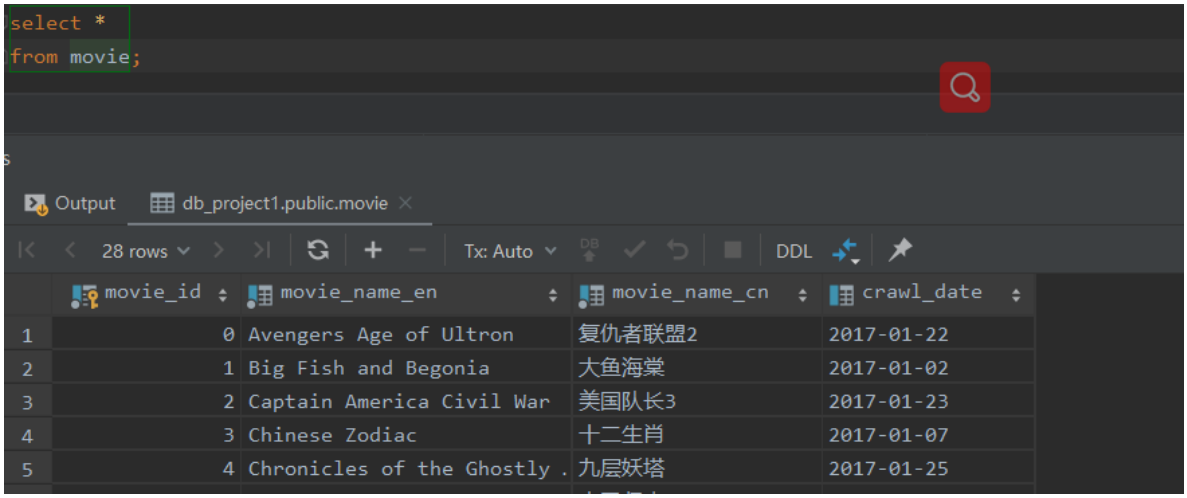
```
db_project1=# \du
                                   角色列表
 角色名称 |                         属性                         | 成员属于
----------+------------------------------------------------------+----------
 checker  | 超级用户                                             | {}
 postgres | 超级用户，建立角色，建立 DB，复制，绕过RLS | {}
```

- **Make some user can only $select$ one table.**

If we give $user1$ the privileges of selecting the movie table as follow:

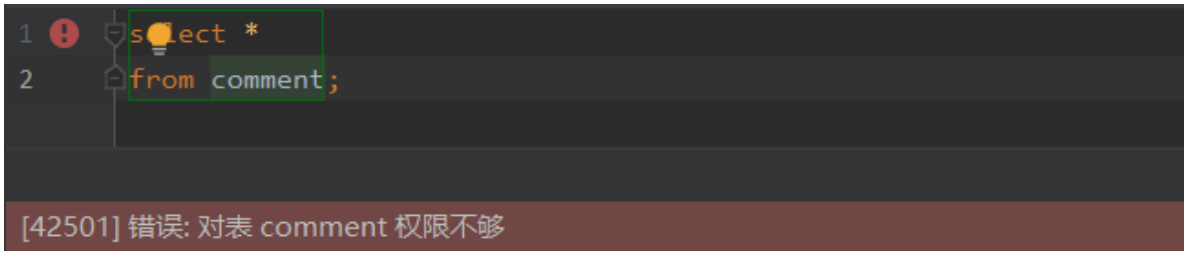
```
db_project1=# grant select on movie to user1;
GRANT
```

And then if we excute the $select$ instructions as follow, and then we can get the result.



If we try to delete the data from $movie$ table, we will faill.

If we try to select the data from other table, we will also fail.

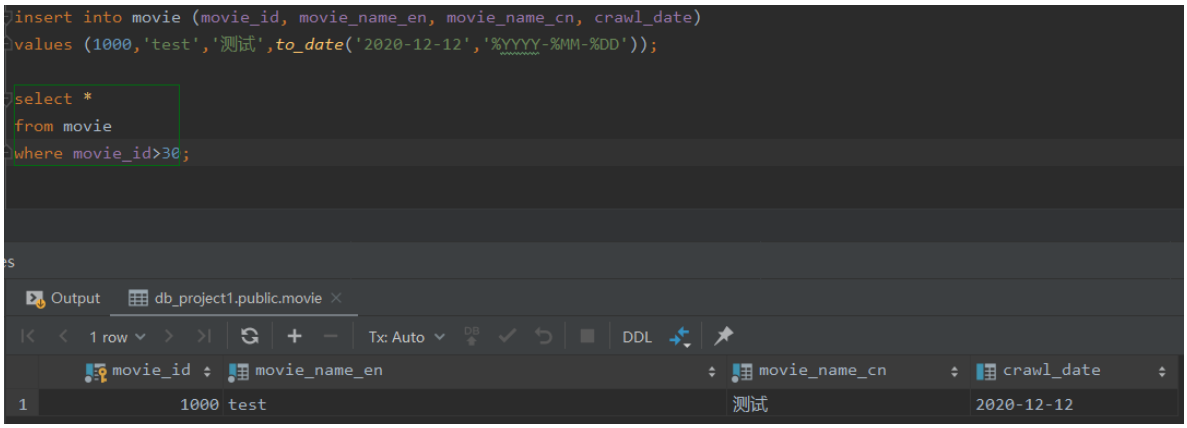


- **Make some user can $insert$ and $select$ one table.**

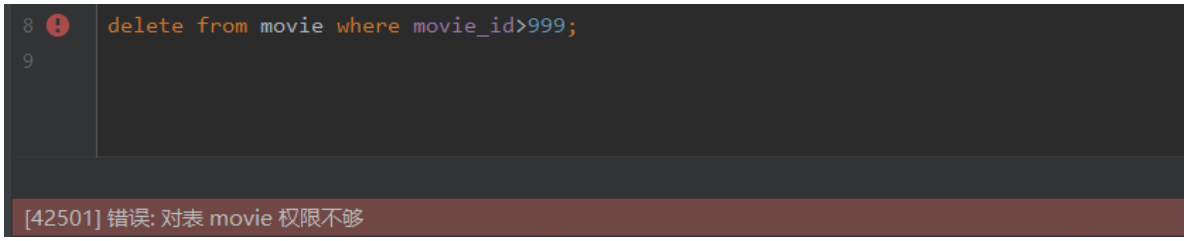
If we give $user1$ the privileges of inserting and selecting the movie table as follow:

```
db_project1=# grant select on movie to user1;
GRANT
db_project1=# grant insert on movie to user1;
GRANT
```

And then we do the $select$ and $insert$, the result is as follow:

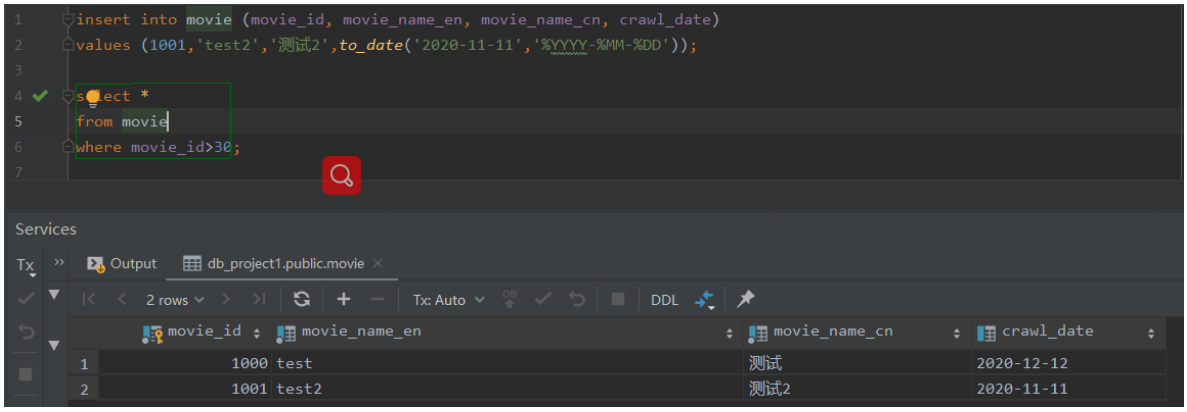


But if we try to still have no previlege to delete the data of movie and do other action on other table.



- **Make some user can do all action in one table.**

We can also have some users which can do any operation our database.
The operation: insert and select the data in movie is as follow:



The operation: delete the data in movie is as follow:

```
delete from movie
where movie_id>30;

select *
from movie
where movie_id>30;
```

Services

Output | db_project1.public.movie

0 rows | Tx: Auto | DDL

movie_id	movie_name_en	movie_name_cn	crawl_date

The operation in other table is as follow:

```
select *
from comment;
```

Services

Output | db_project1.public.comment

1-500 of 501+ | Tx: Auto | DDL

	comment_id	user_id	crawl_date	star	comment	likes
1	1415848	171539	2016-12-28	1	豆瓣加油哦	1
2	1416423	198178	2016-12-20	4	还行吧～	0
3	1412599	33517	2016-12-18	3	评价一面倒的烂　我觉得还行啊...说剧本弱智的难道妇联不弱智吗..	3

3.3.6 Rich query set

DBMS provides a lot of useful function to help us retrieve the data.

- Use $where$ to filter the data

```
select comment,star
from comment
where star=5;
```

Services

Output | Result 9

1-500 of 501+

	comment	star
1	比原著舒服，自然。	5
2	27岁，闺蜜	5
3	你留下来吧。你跟我走吧。你留下来吧。	5
4	安生。	5
5	原来七月与安生的故事不是在讲爱情是在	5
6	安妮宝贝伴随我整个青春期，小说终于改	5
7	在吐槽30分钟前狗血设定的我绝想不到	5
8	从客家围楼到汪洋大海，从火树银花到谁	5
9	椿和楸都让人觉得好心疼😔 很纯真的感	5

- Use aggregate functions

```
select movie_name_CN,count(*)
from movie
join relation r on movie.movie_id = r.movie_id
join comment c on r.comment_id = c.comment_id
group by r.movie_id,movie_name_CN;
```

Output | Result 15

28 rows

	movie_name_cn	count
1	复仇者联盟2	52809
2	大鱼海棠	79642
3	美国队长3	59620
4	十二生肖	44003
5	九层妖塔	41692
6	大圣归来	127675
7	栀子花开	29026
8	夏洛特烦恼	103375

- Use *having* to filter the data

```
select movie_name_CN,count(*)
from movie
join relation r on movie.movie_id = r.movie_id
join comment c on r.comment_id = c.comment_id
group by r.movie_id,movie_name_CN
having r.movie_id=3;
```

Output | Result 17

1 row

	movie_name_cn	count
1	十二生肖	44003

- To order the data

```
select movie_name_CN,count(*)
from movie
join relation r on movie.movie_id = r.movie_id
join comment c on r.comment_id = c.comment_id
group by r.movie_id,movie_name_CN
order by movie_name_CN ASC ;
```

Output | Result 18

28 rows

	movie_name_cn	count
1	爱乐之城	91582
2	变形金刚4	56787
3	大圣归来	127675
4	大鱼海棠	79642
5	疯狂动物城	132581
6	釜山行	98930
7	复仇者联盟	76172
8	复仇者联盟2	52809
9	钢铁侠1	23179
10	何以笙箫默	25191
11	后会无期	114673
12	九层妖塔	41692
13	湄公河行动	33918

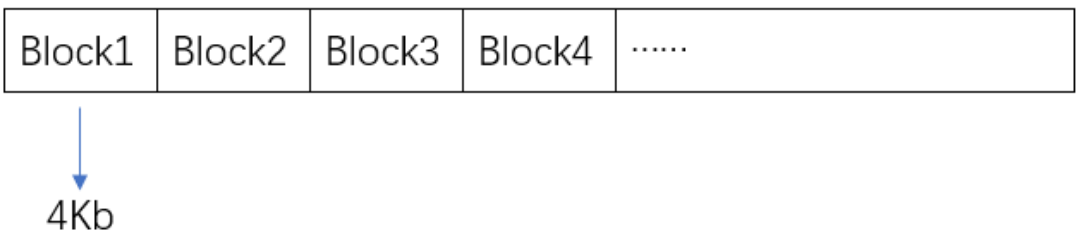
3.3.7 Database index and File IO

Firstly, we will talk about File IO. According to Wikipedia[1], the definition of IO is:

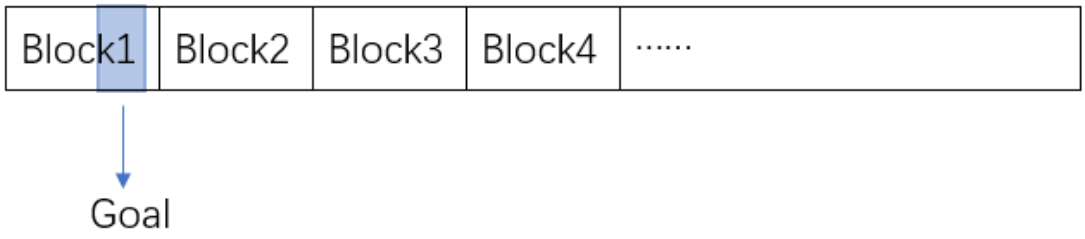
> In computing, **input/output** or **I/O** (or, informally, **io** or **IO**) is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system. Inputs are the signals or data received by the system and outputs are the signals or data sent from it. The term can also be used as part of an action; to "perform I/O" is to perform an input or output operation.

Actually, each IO have its costs and the cost is related to the consumption per read from disk and memory. From a hardware perspective, we recognize the limitations of memory itself. Such that, we will mainly improve the speed in software perspective. In fact, different operation systems give a helpful support to IO.

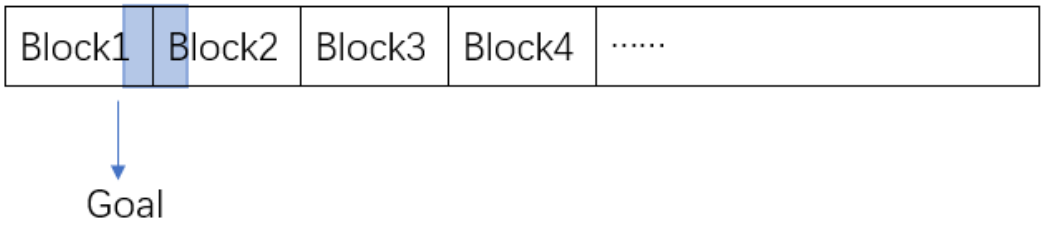
In our disk, the data is managed by blocks, and each block is 4Kb. We can think of the disk as the following sequential storage structure.



When the operation system want to get the data from disk, it will use certain block id to drive the disk to read the data and search the block data. It is improtant to note that these operations start with a block, and the smalledt unit of data per read is also 4Kb. Such that if the information we want to get is less than 4Kb or it doesn't lie in the start of the block, the disk will still read the whole data of the block.



Even more, if the data we want to get cross two blocks, the disk will read both of the two blocks.



In order to avoid the above situation, we align the target data in blocks to reduce the number of block reads during disk I/O. We know the cost of I/O is very consuming, so we will think about a way to manage the data in our database to reduce the disk I/O as possible.

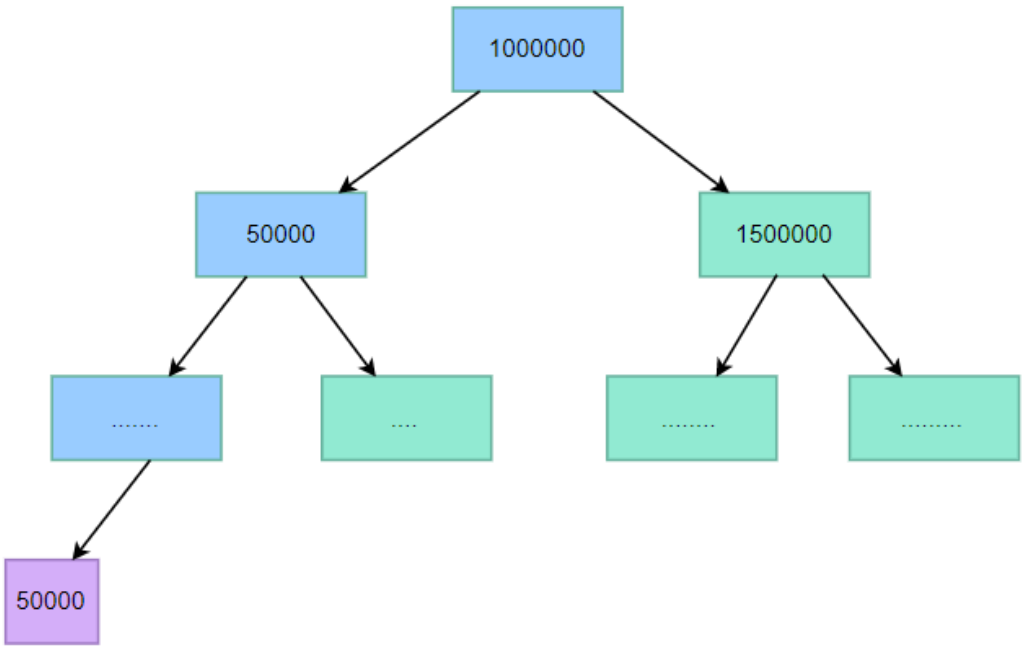
Consider we need to find the username with user_id = 2 in my user_info table. Since we can store about 1 million data in my file, we will use about 60000-80000 blocks. In this case, the pressure on the computer to process the data will rise, and as the data increases, the complexity will increase. Then the Database Index I discussed next will solve the retrieval problem of a large amount of data.

Actually, the Database Index is not something speacial. Here I list an example to show how index works. If I want to use user_id to retrieval data, we can use a data structure to locate the id.

user_id	Record the block_id
1	N
10000	N+k
20000	N+2k
.....	.....
1000000	N+100k

That is to say, we map the user_id to the block_id. For example, if we want to find user_id=50000, then from the map relation, we can predict the location is in [N+5k,N+6k]. In this way, we can reduce the operation of IO.

Another data structure is binary search tree. Since we have learned this structure before, I will not give another explanation here. I just give a figure to show this work if I want to find the user which user_id is 50000.



Actually, the data structure in DBMS is not binary search tree. It is a more complex tree structure -- B+ tree. It is more suitable to seach and delete something in the data. But I will not give more details here.

In Postgresql, we can use the following clause to create an index:

```
create index index1
on user_info(user_id);
```

In fact, DBMS will automatically set the primary key as the index. That is also one of the reasons we must set at least one primary key in a table. What's more, this is also why DBMS is faster than file. Its reduce the time by logN. However, the operation of insert will also cost logN.

4. Conclusions

Through this project, we feel the convenience of DBMS from various aspects. At the same time, we also understand the DBMS data structure, master its working principle. After this project, we can have a deeper understanding of the database and should become more proficient in the future use. However, the wheel of scientific development is forward, and the DBMS is constantly updating and iterating, so the study of database management system must not end here, there is more to be discovered.

5. Reference and Citation

5.1 Citation

[1] : I/O definition. Retrieved from: https://en.wikipedia.org/wiki/Input/output

5.2 Reference

[1] : Advantage of database manage system over file system. Retrieved from:

https://www.csestack.org/advantages-of-database-management-system-over-file-system/

[2] : Advantages of Database Management System. Retrieved from:

https://www.tutorialspoint.com/Advantages-of-Database-Management-System

[3] : Characteristics and benefits of a database. Retrieved from:

https://opentextbc.ca/dbdesign01/chapter/chapter-3-characteristics-and-benefits-of-a-database/

[4] : 深入浅出数据库索引原理. Retrieved from: https://zhuanlan.zhihu.com/p/23624390

[5] : 菜鸟教程|PostgreSQL 索引. Retrieved from:

https://www.runoob.com/postgresql/postgresql-index.html

5.3 Tools

[1] : 在线树状图绘制软件. Retrieved from:

https://online.visual-paradigm.com/cn/diagrams/features/dendrogram-software/