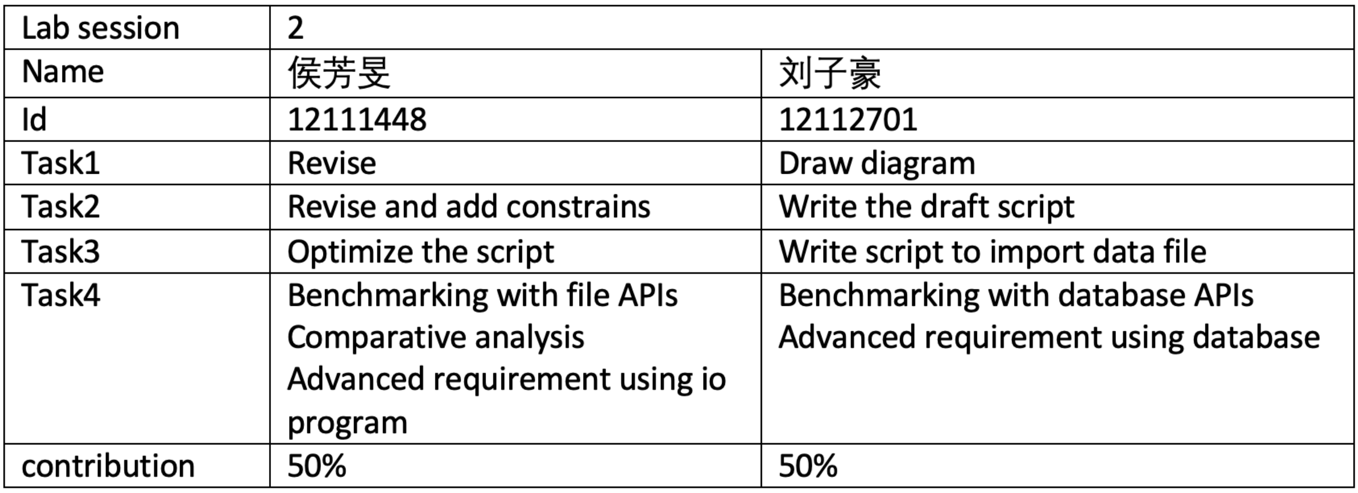
Fall 2022 CS307 Project Part I

> 12111448 侯芳旻
>
> 12112701 刘子豪

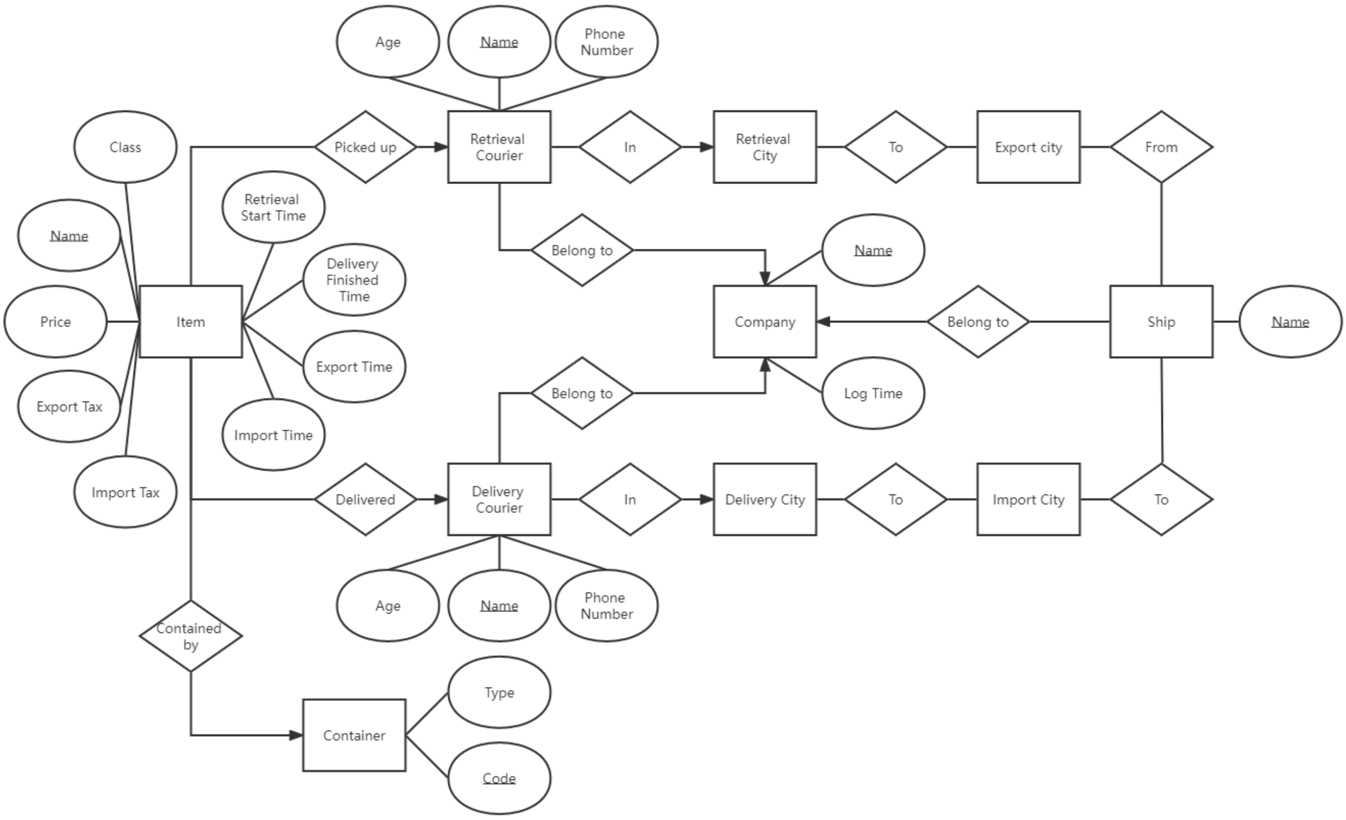
Basic Information of Group and Workloads



Lab session	2	
Name	侯芳旻	刘子豪
Id	12111448	12112701
Task1	Revise	Draw diagram
Task2	Revise and add constrains	Write the draft script
Task3	Optimize the script	Write script to import data file
Task4	Benchmarking with file APIs Comparative analysis Advanced requirement using io program	Benchmarking with database APIs Advanced requirement using database
contribution	50%	50%

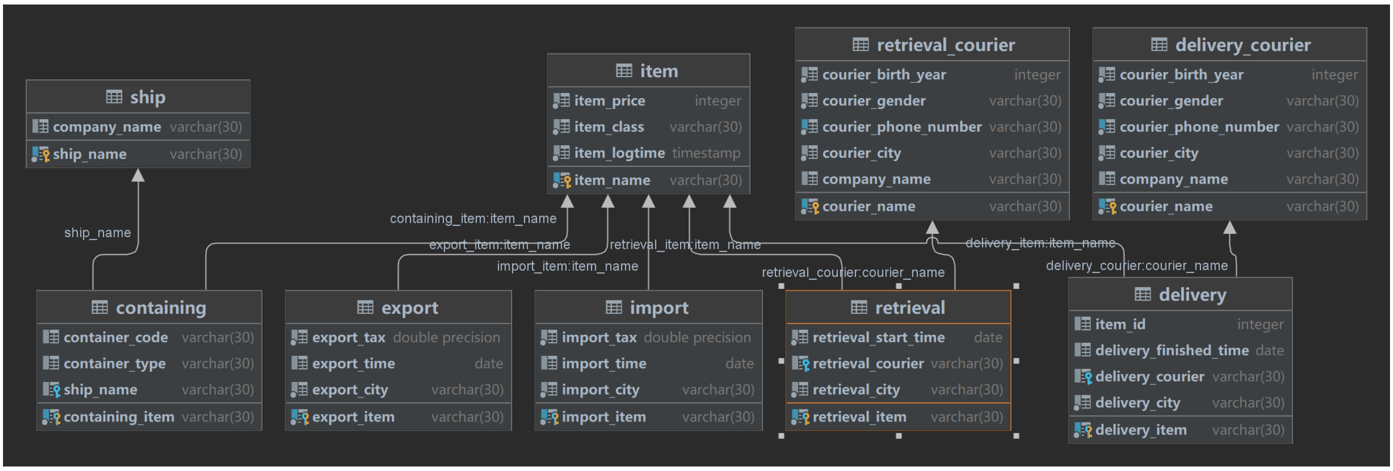
Task 1: E-R Diagram

- The diagram is drawn by https://www.processon.com
- Diagram



Task 2: Database Design

- Diagram



ship

company_name	varchar(30)
ship_name	varchar(30)

item

item_price	integer
item_class	varchar(30)
item_logtime	timestamp
item_name	varchar(30)

retrieval_courier

courier_birth_year	integer
courier_gender	varchar(30)
courier_phone_number	varchar(30)
courier_city	varchar(30)
company_name	varchar(30)
courier_name	varchar(30)

delivery_courier

courier_birth_year	integer
courier_gender	varchar(30)
courier_phone_number	varchar(30)
courier_city	varchar(30)
company_name	varchar(30)
courier_name	varchar(30)

containing

container_code	varchar(30)
container_type	varchar(30)
ship_name	varchar(30)
containing_item	varchar(30)

export

export_tax	double precision
export_time	date
export_city	varchar(30)
export_item	varchar(30)

import

import_tax	double precision
import_time	date
import_city	varchar(30)
import_item	varchar(30)

retrieval

retrieval_start_time	date
retrieval_courier	varchar(30)
retrieval_city	varchar(30)
retrieval_item	varchar(30)

delivery

item_id	integer
delivery_finished_time	date
delivery_courier	varchar(30)
delivery_city	varchar(30)
delivery_item	varchar(30)

- Description: We designed 9 tables to store all the information in the given csv file. The name of every table clearly described the content of the table. Every column is related to a kind of information in the given csv file. It can meet three normal forms. And the design contains no circular links;
- Item:
- The table contains all information straightly connected to item
- Item is specifed by its name
- ship
- As the ship name is unique, it is the primary key.
- ship is owned by a company

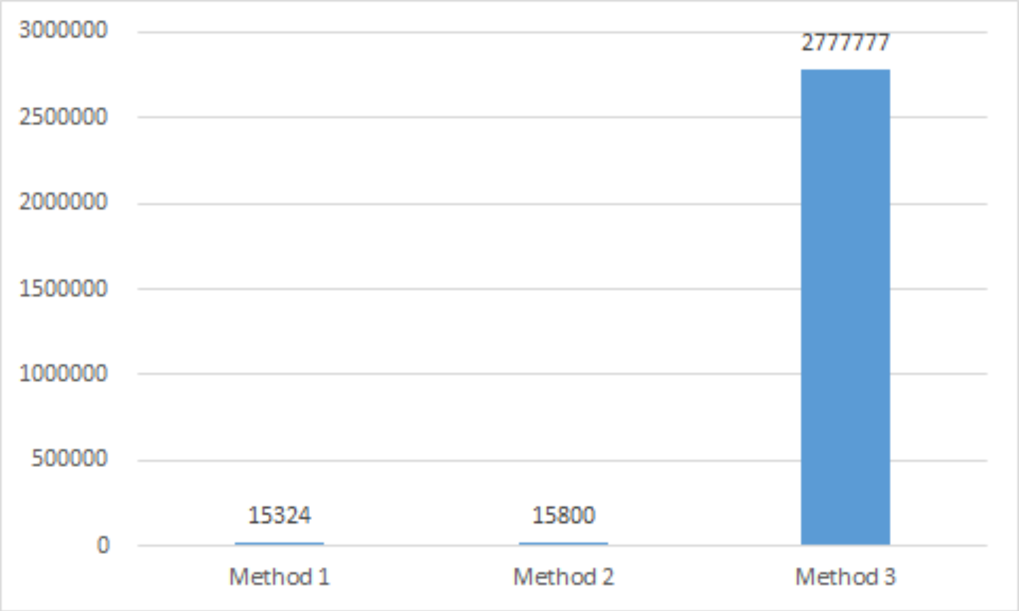
- Export/import
    - contains information about item that is generated while export/import
    - As these information is not owned by item, it is separated into another table
- Retrieval_courier/Delivery_courier
    - These 2 tables are about the

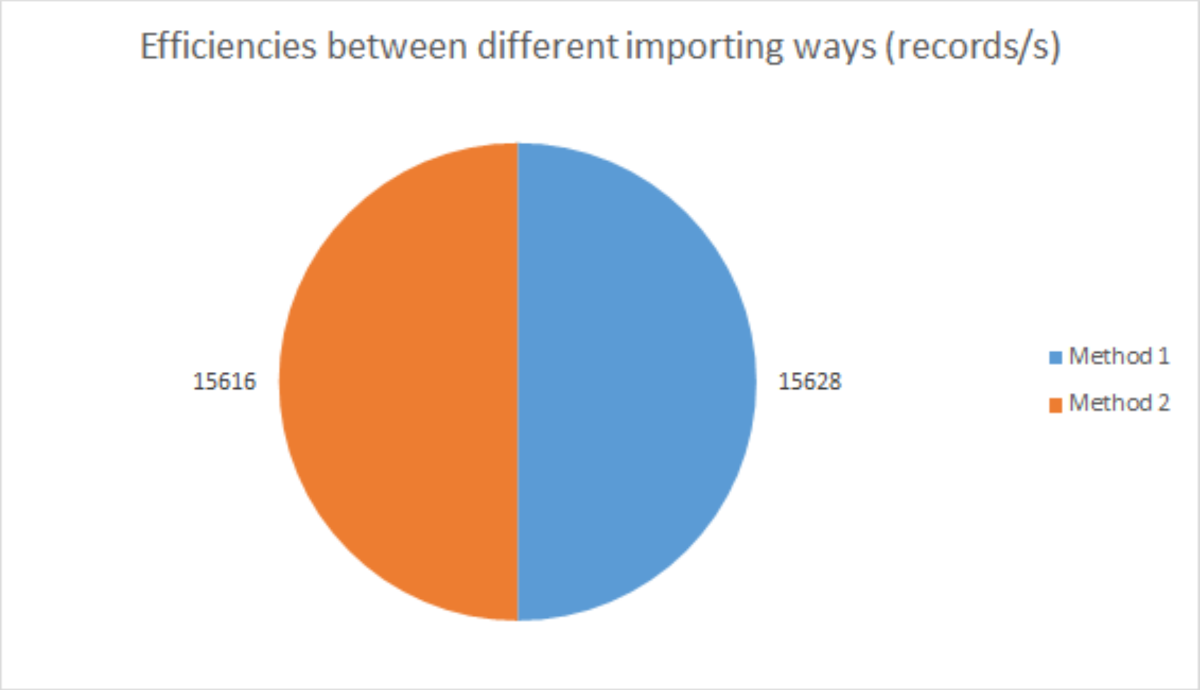
Task 3: Data Import

Description

- Steps:
    1. Using postgresql-42.2.5.jar to connect Java to the database
    2. Load the prepare statement
    3. Read the given csv file
    4. fill prepare statement using data getting from csv file
    5. execute the statement
    - Prerequisites:
        1. put the csv file into the project
        2. create the table
        3. input String[] args
    - Cautions:
        1. input required jar
        2. the sequence of importing data should meets the demand of the design the keys
        3. convert String to required data types

More than one way



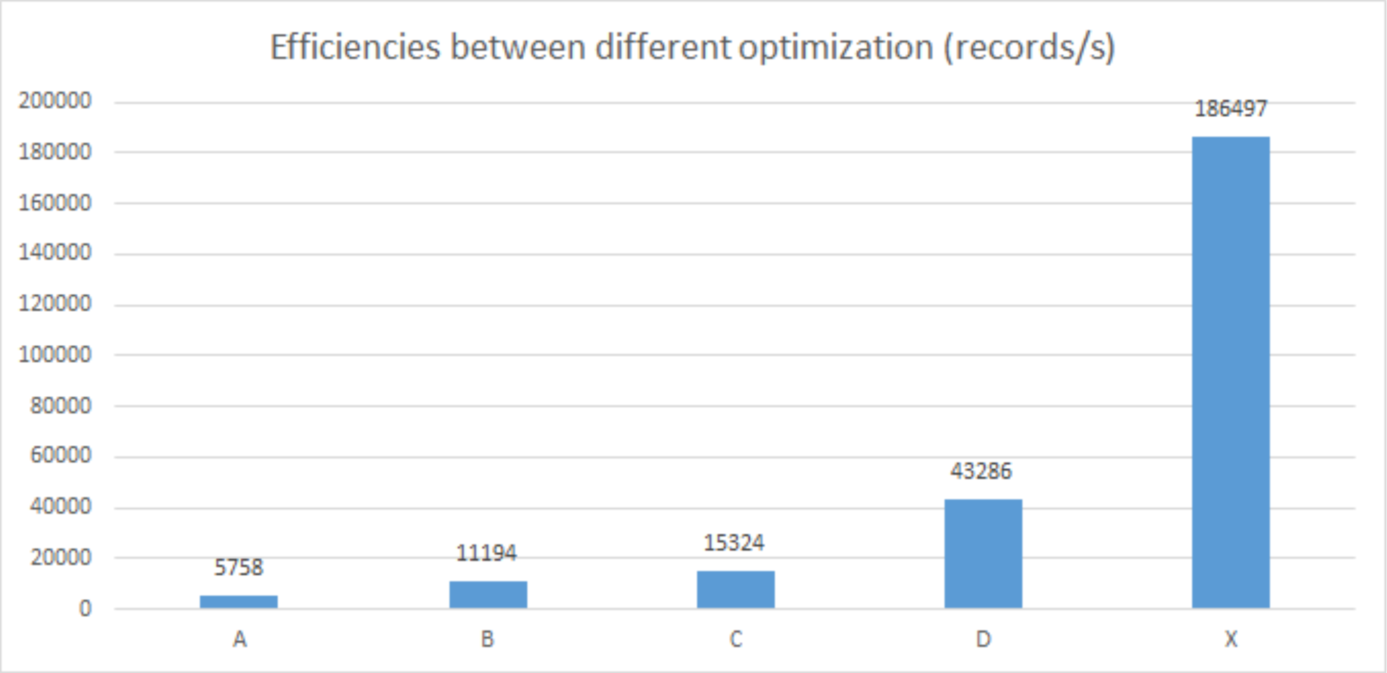


1. method 1: using BufferReader

2. method 2: using javacsv.jar

3. method 3: using the import method in the dataGrip

Optimize



1. A: worst
2. B: Close the auto commit and commit only after finishing all the import.
3. C: Close the auto commit and commit only after finishing all the import. And while importing every line of data, fill the prepared statement rather than creating a new statement every time.
4. D: Close the auto commit and commit only after finishing all the import. And while importing every line of data, fill the prepared statement rather than creating a new statement every time. And using addBatch method to add a set of parameters into the prepared statement and execute it until the batch size is full.
5. x: Improve the prepared statement by writing parameters straightly in the sql statements rather than fill it in every line's read

```
 private static void loadData(long cnt,String item_name, int item_price, String
item_class, String item_logtime)
            throws SQLException {
        PreparedStatement stmt;
        if (con != null) {
            stmt = con.prepareStatement("insert into
item(item_name,item_price,item_class,item_logtime) values('" + item_name + "','" +
item_price + "','"+item_class+"','"+item_logtime+"')");
            stmt.addBatch();
            if (cnt % BATCH_SIZE == 0) {
                stmt.executeBatch();
                stmt.clearBatch();
            }
        }
    }
```

Task 4: Compare DBMS with File I/O

- To simplify the process of comparison, we create a new table adding a column of id.

```
create table if not exists Delivery_Test
(
    Item_id                  int,
    Delivery_Item            varchar(30) primary key,
    Delivery_Finished_Time varchar(30),
    -- Can be EMPTY if the delivery is not finished.
    Delivery_Courier         varchar(30),
    Delivery_City            varchar(30) not null
);
```

###

test environment

- 12111448: MacBook Pro Apple M1 Pro 16 GB
- 12112701: 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz 2.69 GHz 16 GB

software

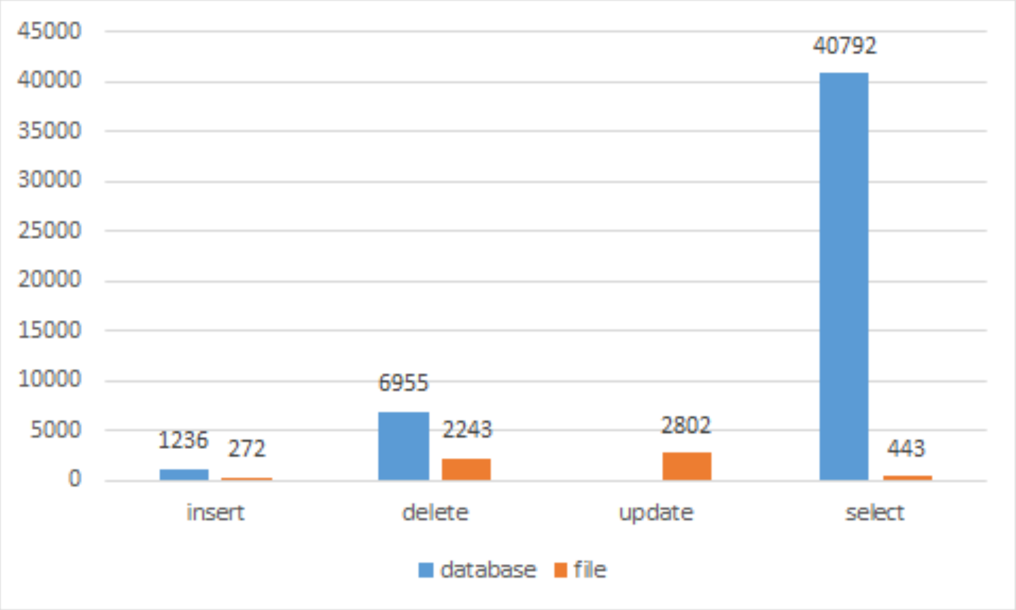
- DBMS:

  postgresql
- data file:

  csv file

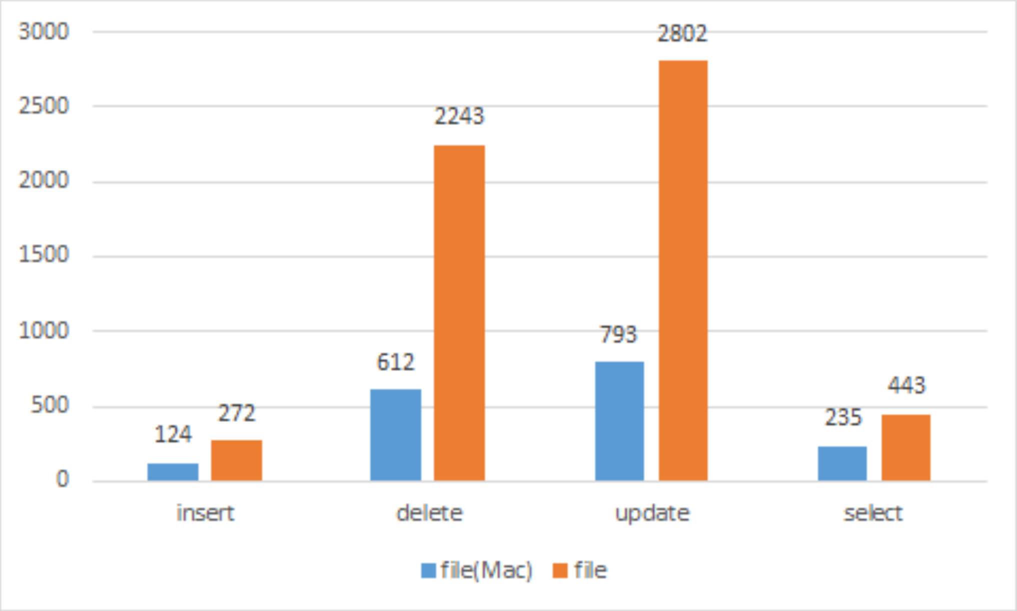
comparative

- we used the computer of one group number's computer run all the codes



advanced

- Can you identify the retrieval and delivery couriers who collect/send the greatest number of items for each company in each city through your database? How about your file I/O program? If so, which will perform well? If not, why?
  - Yes.
  - database performs better.
- If a company have different type of items (for example, half of the “Item Class” columns’ distinct values) that will be exported, which cities it shall choose respectively to reach the minimal overall export cost? Can your database deal with such problem? How about your file I/O program? If so, which will perform well?
  - yes.
  - database performs better.
- operating system / cpu plays a important role in the speed.


insert
delete
update
select
file(Mac)
file