

## Project 2 Report

---

**Authors:** 陆荻芸 12011537 龚明道 12011204

**Lab Session:** 5

**Ratio of scores:**50% for each member.

## 1 Basic Part

### 1.1 API for manipulating original data

We did not put this part inside our front-end website. It is manipulated through java file.

#### 1.1.1 getEnterprise:

Get data from database. And we create enterprise class to store data.

#### 1.1.2 writeBackEnterprise:

Write the data back to database.

#### 1.1.3 addEnterprise:

Input: String name,String country,String city,String supply\_center,String industry  
Put this data to database.

#### 1.1.4 deleteEnterprise:

Input: String name  
Remove data from database.

#### 1.1.5 updateEnterprise:

Input: String name,String country,String city,String supply\_center,String industry  
Update data from database.

#### 1.1.6 selectEnterprise:

Input: boolean id,boolean name,boolean country, boolean city, boolean supply\_center,boolean industry, int idC,String nameC,String couC,String citC,String supC,String indC

Boolean type to confirm whether you query according to this condition.  
The following data represents the content of your query.  
Returns a result of String type.

## 1.2 APIs for data input

One button **impBasic** is used to import all original data and test data. The import step includes 5 detail parts of implementation. They are **API to import original data**, **API stockIn**, **API placeOrder**, **API updateOrder**, **API deleteOrder**.

All of them read input from given csv files and operate on legal data which can be used to modify database.

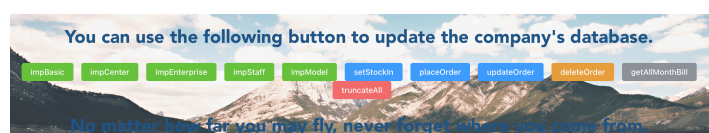


图 1: Buttons for these APIs

## 1.3 APIs for getting statistical information

All APIs in this part require no input parameters and return a result of statistical magnitude. In our implementation, you can use a button to get the answer of these counted information. The APIs are listed below.

### 1.3.1 getAllStaffCount:

Return numbers of staffs of all types.

### 1.3.2 getContractCount:

Return a total number of existing contracts.

### 1.3.3 getOrderCount:

Return a number of existing orders.

### 1.3.4 getNeverSoldProduct:

Return a number of the products that have never been sold.

### 1.3.5 getFavouriteProductModel

Return the models with highest sold quantity and its exact sales.

### 1.3.6 getAvgStockNumber:

Return the average quantity of remaining product models for each supply center and is rounded to one decimal place.

## 1.4 APIs for searching information

All APIs in this part require a input parameter and return multiple columns of information related to the search.

### 1.4.1 getProductNumber

**Input:** product number

**Output:** A row of information containing supply center the product belongs and its number, model name, purchase price and quantity.

### 1.4.2 getContractInfo

**Input:** contract number

**Output:** Contract information including its number, relating manager name, relating enterprise name and supply center. Also includes the informatino of all orders belong to this contract with the product's name, salesman's name, quantity, unit price, estimated delivery date and lodgement date of each specific order.

For all APIs with output, we are allowed to get the result with a button and displayed as a table. See an example below.

number	manager	enterprise	supply_center		
CSE0000101	Qian Yupeng	Chengdu AIG	Southwestern China		
product_model	salesman	quantity	unit_price	estimate_delivery_date	lodgement_date
PhotoBook0	Feng Runbin	420	300	2022-01-05	2022-01-07
InkjetPrinters54	Sean Jones	345	520	2022-01-06	2022-01-04
ImageAlarm41	Eleanor Patel	163	160	2022-01-05	2022-01-04
NotebookAccessoriesA7	Jo Roberts	255	570	2022-01-04	2022-01-05
CSE0000101	<div>search</div>				

图 2: getContractInfo example

## 2 Advanced Part

### 2.1 Query the order list based on multiple parameters

Our front page has an orderSelect window for users to query freely which allows multiple input parameters.



## 2.4 Update Order Type According to Time

A function and a trigger is created to judge if the inserted or updated order is already passed lodgement date. If it is, then straightly change its type to "Finished".

```
create or replace function time_check()
returns trigger
as
$$
BEGIN
    if (new.lodgement_date <= current_date)
    then
        new.contract_type = 'Finished';
        return new;
    else
        return null;
    end if;
end;
$$
language plpgsql;
create trigger checkTime
before insert or update
on orders
for each row
execute procedure time_check();
drop trigger checkTime on orders;
```

图 6: sql of corresponding function and trigger

## 2.5 GUI Design

Vue framework is used for front-end design.



图 7: GUI

## 2.6 Back-end Server

The springboot framework is used for back-end design.

Lombok is used to create entity for data storage, and directly transmit it to the front end for data display, simulating the table structure in the database.

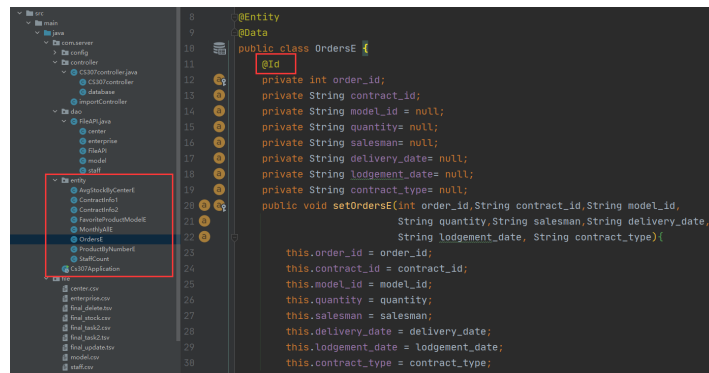


图 8: entity for data storage

Based on websocket communication, we have realized accessing our web pages in the LAN of campus network which allows multiple hosts to visit at the same time.

## 2.7 User Privileges

Our front-end provides a user permission switching interface. You can log in to the manager account or the staff account. When checking the order information, different data will be displayed due to different permissions. Staffs are only allowed to change the rows related to themselves by enabling row level security on tables.

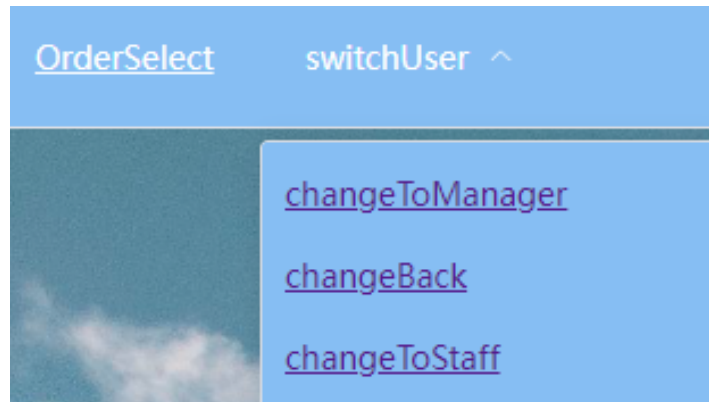


图 9: Switch User

## 2.8 Index

Datagrip has generated indexes on primary key and columns constrained with condition unique. In addition, we add more indexed according to our specific usage such as index on column lodgement date in table orders to speed the searching in the function mentioned above.