Spring 2022 CS307 Project Part1

> Main Contributors:
>
> Leader and Overall Design: ZHU Yueming
>
> Data Preparation and Documentation: WANG Lishuang, WANG Zhiyuan, ZHANG Chaozu, and WU Shangxuan
>
> Review: MA Yuxin
>
> Extended from the project of Spring 2021

General Requirement:

- It is a group project with **only 2 teammates** who are **in the same lab session**. Each group should finish the project independently and submit only one report written by the teammates.
    - The teammate you select for Project 1 will also be your teammate for Project 2. It is not allowed to change teammates once paired.
- You should submit the report before the deadline. All late submissions after the deadline will receive a score of zero.
- DO NOT copy ANY sentences and figures from the Internet and your classmates. Plagiarism is strictly prohibited in this course.
- The number of pages for your report should be between **8** and **20**. Reports **less than 8 pages** and **more than 20 pages** will receive a penalty in the scoring stage.

DBMS can help us manage data in a convenient manner and improve the efficiency of data retrieval. Your work of Project 1 is mainly divided into four parts below:

1. Design an E-R diagram based on the provided data file and data relationships.
2. Design a relational database using PostgreSQL according to the provided data file.
3. Import all data into the database.
4. Compare the performances of data retrieval and manipulations between databases and raw file I/Os in a programming language. All programming languages including C/C++, Java, and Python are allowed.

Background

This project is based on the requirements for a fictional international trading company called Southern Union manufacture of Science and Technology of China (SUSTC). In this project, you need to design a database for the Marketing Department of SUSTC. The database is used for storing the organizational structure of the Marketing Department and details of the sales contracts.

As a reference in your design process, we provide the structure of the Marketing Department as well as data samples that will be stored in the database. The data samples contain the information of employees and orders.

SUSTC is a top-tier international trading company based in China. The customers of SUSTC come from 7 main areas inside China and around the world including Europe, America, Asia, Eastern China, Northern China, Southern China, and Southwestern China, each of which is handled by a Supply Center. It should be noted that the Asia Supply Center manages businesses in all countries in Asia except for China (Note: for now, businesses in Hong Kong, Macau, and Taiwan Province are considered domestic trades in SUSTC). Each Supply Center has a manager who is responsible for the daily goings-on of the entire center. The names of the managers are listed below:

Europe	America	Asia	Eastern China	Northern China	Southern China	Southwestern China
Audrey Evans	Miriam Evans	Steven Edwards	Xu Zhuyu	Kong Yibo	Yang Penglong	Tao Yibo

In our dataset, there are 50000 sales records from 5000 sales contracts (one sales contract contains multiple sales records). Please refer to the following section for the format of records.

Data Description

The data includes **5,000** contracts, where a contract contains one or more orders. Each order represents a **single kind of** product that has been ordered, i.e., different products should be separated into multiple orders. There are **50,000** orders in total. Here is the explanation of the columns:

1. **contract number**: the unique identifier of each contract; each value likes CSEXXXXXXX, from CSE0000000 to CSE0004999.
2. **client enterprise**: the name of the client enterprise.
3. **supply center**: the corresponding Supply Center to the client enterprise.
4. **country**: the country where the client enterprise is located.
5. **city**: the city where the client enterprise is located.
6. **industry**: the industry where the client enterprise resides. Each industry belongs to only one supply center.
7. **product code**: the unique identifier of the product; each value is a mix of letters and numbers, e.g., L8N0649, C186H47, M40V792. Each product code has only one product name and more than one product model.
8. **product name**: the name of the product.
9. **product model**: the specific model of the product. Each project model has its own unit price.
10. **unit price**: the unit price of the product in this contract.
11. **quantity**: the quantity of products ordered in the contract.
12. **contract date**: the date of creating the contract.
13. **estimated delivery date**: the estimated date of the delivery of the product.
14. **lodgement date**: the actual date of the delivery of the product.
15. **director**: the person responsible for this contract.
16. **salesman**: the name of the salesman for this order.
17. **salesman number**: the number of the salesman.
18. **gender**: the gender of the salesman.

19. **age**: the age of the salesman.
20. **mobile phone**: the mobile phone number of the salesman.

Notes:

- the value of **city** will be **NULL** if the client enterprise is **not** in China;
- the value of **lodgement date** will be **NULL** if the date is later than 2022-3-2;

Requirement of the Project Report

Note: Some tasks consist of basic requirements and advanced requirements. You may not get full points if you only meet the basic ones.

Basic Information of Your Group

1. Names, student IDs, and the lab session of the group members
2. You are required to write down the contributions and the percentages of contributions for each group member. **Please clearly state which task(s)/part of the task(s) is/are done by which member in the group.**
    - **If you failed to link a task/part of a task to one of the group members, we will not count the score for the task** (since we don't know who accomplished this task; maybe it was done by an elf while you were sleeping at night?).

Task 1: E-R Diagram (15% in total)

Make an E-R Diagram of your database design with any diagram software. Hand-drawn results will not be accepted. Please follow the standard of E-R diagrams.

In the report, you are required to provide a snapshot of the E-R diagram (15% out of 15%). Also, please specify the name of the software/online service you use for drawing the diagram.

Task 2: Database Design (25% in total)

Design the tables and columns based on the background provided above. Generate the E-R diagram via the "Show Visualization" feature. Briefly describe the design of the tables and columns including (but not limited to) the meanings of tables and columns.

In the report, you are required to provide the following content (25% out of 25%):

1. Attach the snapshot of the E-R diagram generated by DataGrip.
2. Briefly describe the table designs and the meanings of each table and column.

In addition, please submit an SQL file as an attachment that contains the DDLs (`create table` statements) for all the tables you created. **Please make it into a separate file but not copy and paste the statements into the report.**

Notes for the database design:

1. All data items should base on the file **contract_info.csv**.
2. Your design needs to follow the requirements of the three normal forms
3. Use a primary key and foreign keys to indicate important attributes and relationships about your data
4. Every row in each table should be uniquely identified by its primary key. (You may use a simple or a composite primary key).
5. Every table should be involved in a foreign key. No isolated table is allowed. (每个表要有外键，或者有其他表的外键指向。)
6. Your design should contain no circular foreign-key links. (对于表之间的外键方向，不能有环。例如：A表有外键关联B表，B表有外健关联C表，C表有外健关联A表)
7. Each table should contain at least one mandatory (“Not Null”) column (including the primary key but not the id column).
8. Other than the system-generated self-increment ID column, there should be at least one column with the "unique" constraint. (除了主键自增的id之外，需要有其他unique约束的列)
9. You should use appropriate data types for different fields.
10. Your design should be easy to expand when requirements change.

Task 3: Data Import (25% in total)

In this task, you should write a script to import the content in `contract_info.csv` into the database you have designed before. After importing the data, you should also make sure all data is successfully imported.

In the report, you are required to accomplish the basic requirements (15% out of 25%):

1. The script you wrote to import the data file.
2. A description of how you use the script to import data. You should clearly state the steps, necessary prerequisites, and cautions in order to run the script and import data correctly.

You may also need to finish the following advanced requirements to get the remaining points (10% out of 25%):

1. Find more than one ways to import data, and provide a comparative analysis of the computational efficiencies between these ways.
2. Try to optimize your script. Describe how you optimized it and analyze how fast it is compared with your original script.

For the advanced points, please make sure to describe your test environment, procedures, and actual time costs. It is required to write a paragraph or two to analyze the experiment results. You may refer to the requirements for reporting experimental results in Task 4 for details.

Task 4: Compare DBMS with File I/O (35%)

In this task, you are required to compare the performance of data retrieval and manipulation between database APIs and file APIs in a programming language. Please conduct the comparative analysis according to the following steps:

1. Benchmarking with Database APIs: First, prepare a table with at least **20000** rows. You'd

better reuse any tables from our course or this project. Then, write a program in a general-purpose programming language (e.g., C/C++, Java, Python) that accesses the database via database APIs and contains a series of `INSERT`, `DELETE`, `UPDATE`, and `SELECT` statements. You may specify the number of statements in each type on your own. Finally, you need to record the running time of each statement and statements in the same type.
2. Benchmarking with file APIs: This step is designed to replicate all operations in the first step, but in a generic programming language using its standard file APIs. First, create a file that stores **exactly the same data as you have in the table in the DBMS**. Then, write a program to insert, delete, update, and find the data items as you did in the SQL statements and queries. Be sure that the file operations (and the number of operations) are identical to the SQL operations (and the number of the statements). Finally, record the running time of each file operation.
3. Comparative analysis: Compare the recorded running time of the same operation/statement from the DBMS and the file , respectively. You can conduct comparisons from multiple levels, such as comparing statements with corresponding operations (statement-level) or comparing the total time of all statements in a specific type with the corresponding operation type (type-level).

In the report, you are required to finish the following basic requirements (20% out of 35%):

1. A description of your test environment, including:
    - Hardware specification, including (but not limited to) the CPU model, size of memory, whether you are using a solid-state disk (SSD) or hard disk drive (HDD).
    - Software specification, including (but not limited to) the version of your DBMS and operating system, the programming language you choose, and the development environment (such as the version of the language, the specific version of the compilers and libraries, etc.).
    - While reporting the environment, you can think about this question: If someone else is going to replicate your experiment, what necessary information should be provided for him/her?
2. A specification of how you organize the test data in the DBMS and the data file, including the DDLs of the tables and the data format of the files.
3. A description of your test SQL script and the source code of your program. DO NOT copy and paste the entire script and the program in the report. Instead, please submit source codes as attachments.
4. A comparative study of the running time for the corresponding statements/operations. You are encouraged to use data visualization to present the results. Besides a list/figure of the running time, you are required to describe the major differences with respect to running performance, what you find interesting in the results, what insights you may show to other people in the experiments, etc.

Some notes on how to finish this task in a better way:

1. The number of statements/operations should not be small (e.g., less than 100 insertions or 10 select queries).
2. You can choose or design any format you want to store data in the file, such as plain text formats (CSV, JSON, XML, etc.) or a self-defined binary format.

3. Please **only stick to standard file APIs**, such as `java.io` or the file object in Python. The only exception is that if you choose to use JSON and XML, you may utilize standard or third-party JSON/XML libraries (e.g., Gson for Java, the `json` package in Python, etc.).
4. We acknowledge that there are numerous libraries that can facilitate the data manipulation works or even speed up the performance of insertions and selections significantly (e.g., Pandas in Python). You are encouraged to also compare the performance of these libraries with DBMS. However, you should conduct the analysis of DBMS vs. standard file APIs beforehand.
5. Some useful resources:
    - Advantage of database management system over file system
    - Advantages of Database Management System
    - Characteristics and benefits of a database

In addition to the basic requirements, you can also think about some of the following advanced tasks (but not limited to the following ones) to challenge yourself and get the remaining points. (15% out of 35%)

1. High concurrency and transaction management
2. User privileges management
3. Database index and file IO
4. Better data visualization to present your experimental results.
5. Comparisons of performance with different database software (e.g., MySQL, MariaDB, SQLite), file systems, programming languages, libraries, etc. under different operating systems.

How to Submit Your Report

Submit the report in PDF format with necessary attachments (such as SQL scripts and source code files) on the Sakai website before **23:30 on April 17th, 2022, Beijing Time (UTC+8)**. For attachments, please put them into separate directories based on the task, and compress them into a `.zip` archive.

Disclaimer

The names, characters, businesses, and events in the background of this project are purely fictional. The items in the files are randomly-generated fake data. Any resemblance to actual events, entities or persons is entirely coincidental and should not be interpreted as views or implications of the teaching group of CS307.