Course Information

COMP3278A 2023-24

Introduction to Database Management Systems

Dr. Ping Luo
Email : pluo@cs.hku.hk

Department of Computer Science, The University of Hong Kong

Schedule

Date	Venue	Lecture	Date	Venue	Tutorial
			Sep 4	MWT1 &Zoom	Introduction
Sep 7	MWT1	ER Modelling	Sep 11	MWT1 &Zoom	Course Project
Sep 14	MWT1	ER Design & Assignment 1 (15%)	Sep 18	MWT1 &Zoom	Course Project
Sep 21	MWT1	SQL I	Sep 25	MWT1 &Zoom	ER Modelling & Design
Sep 28	MWT1	SQL II	**Oct 2**	**Public Holiday**	
Oct 5	MWT1	SQL Challenge (10% bonus)	Oct 9	MWT1 &Zoom	SQL Exercise
Oct 12	MWT1	Relational Algebra I & Assignment 2 (15%)	**Oct 16**	**Reading Week**	
Oct 19	**Reading Week**		**Oct 23**	**Public Holiday**	
Oct 26	MWT1	Relational Algebra II	Oct 30	MWT1 &Zoom	Relational Algebra
Nov 2	MWT1	Functional Dependence	Nov 6	MWT1 &Zoom	Relational Algebra
Nov 9	MWT1	Database Normalization	Nov 13	MWT1 &Zoom	Functional Dependence
Nov 16	MWT1	Indexing	Nov 20	MWT1 &Zoom	Database Normalization
Nov 23	MWT1	Project Presentation	Nov 27	MWT1 &Zoom	Project Presentation
Nov 30	MWT1	Q&A for Final Exam			

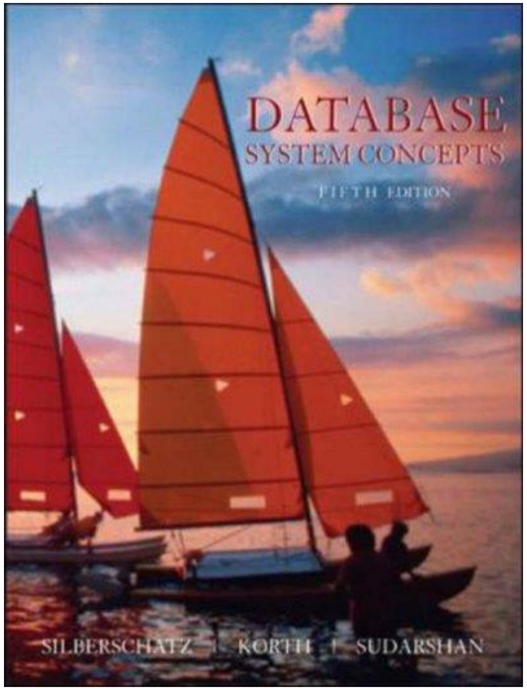
The teaching plan maybe rescheduled according to the actual progress of the course.

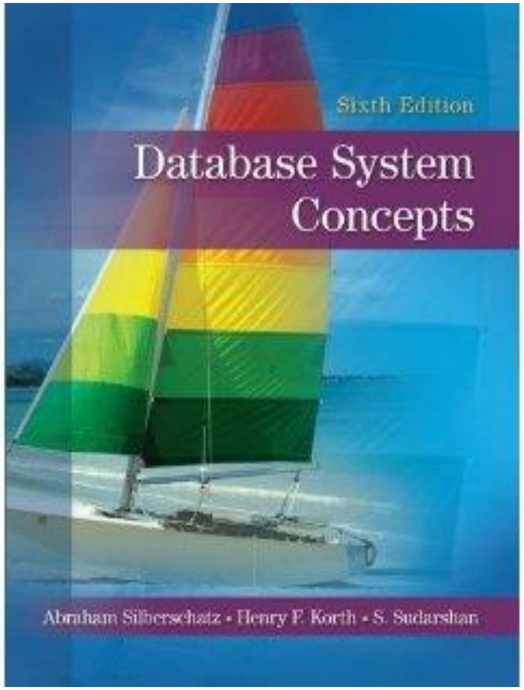
Relational database

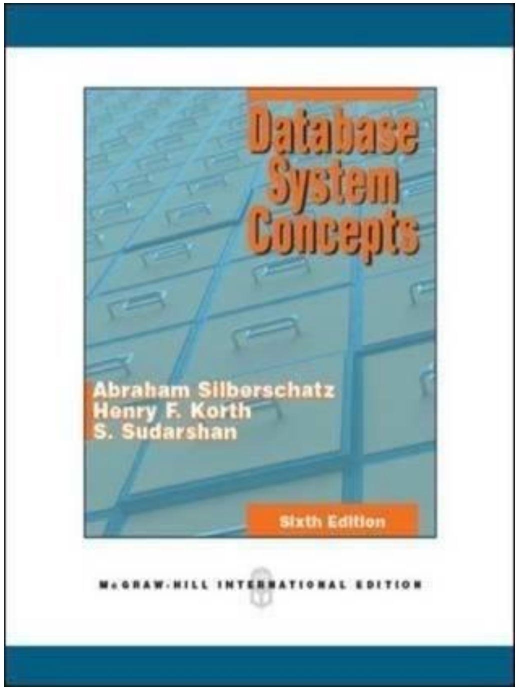
- MySQL database provided by HKUCS department.
  - **CS account:** https://intranet.cs.hku.hk/csintranet/newstudent.jsp
  - **MySQL account:** https://intranet.cs.hku.hk/common/mysqlacct/
  - **Interface:** https://i.cs.hku.hk/phpmyadmin/
- You can install MySQL on your own computer, by following the documentations of MySQL or many instructional videos on Youtube.

Reference books

- A.Silberschatz, H. Korth, and S. Sudarshan. "***Database System Concepts***", *5th/6th Edition, McGraw Hill.*







What will you learn?

- **Outcome 1. Information Modeling**

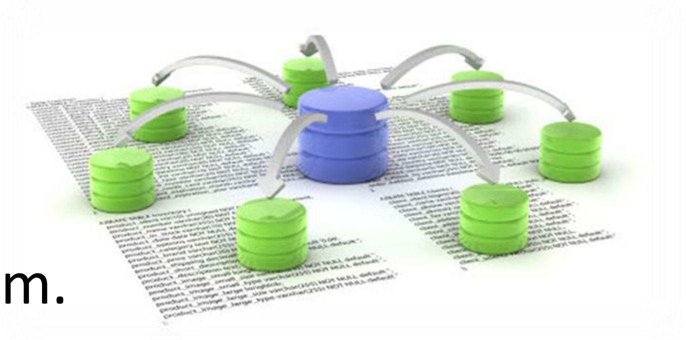
  - Able to understand the modeling of real life information in a database system.
  - Able to learn how to capture user requirements and design a database by using the **entity-relationship (E-R) model**.

- **Outcome 2. Query Languages**

  - Able to understand and use the languages designed to access data from a database.
  - We study **SQL** and **relational algebra** for accessing data in a convenient and systematic way.

What will you learn?

- **Outcome 3. System Design**
  - Able to understand the design of an efficient and reliable database system.
  - These includes various topics, e.g., **normalization**, **storage and indexing schemes**.



- **Outcome 4. Application Development**
  - Able to implement a practical application on a real database, MySQL.
  - Able to finish an assignment that includes the use of a real open source database system (i.e. MySQL).

Assessments

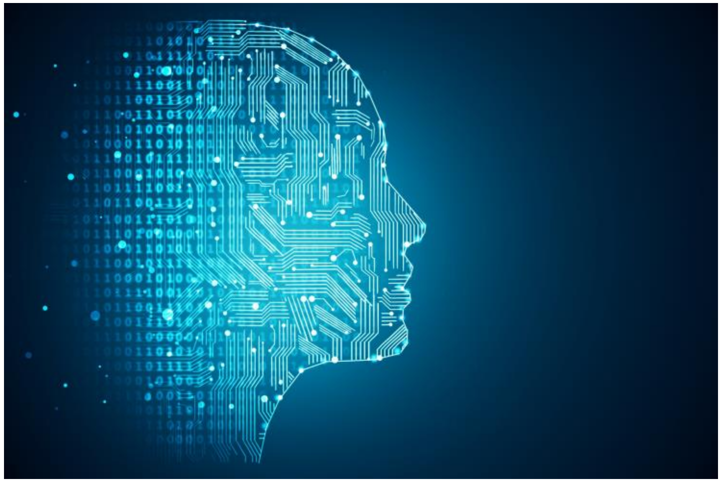
- **2 Written Assignments (30%)**
- **1 Group Project (20%)**
  - Maximum **5** students as a group;
    Find your groupmates (Send group information to TA Chengyue Wu)
  - Implementation (Coding, DB design): 10%
  - Presentation (5 min each group, Nov. 23, 27): 10%
- **SQL Challenge** (each group submit one answer; 10% bonus)
- **Final Examination (50%)**

You have more chance to get high grade: 30 assignments + 20 project + 50 exam + 10 bonus for SQL Challenge.

Your final grade will be capped at 100.

About us…

- **Dr. Ping Luo (Instructor)**
  - You may call me “**Ping**” ☺
  - **Email:** pluo@cs.hku.hk
  - **Room:** Chow Yei Ching Building, Rm 326
  - **Research:** AI, Deep Learning, and Computer Vision
- **Zhixuan Liang (TA)** 👋
  - **Email**: liangzx@connect.hku.hk **(For making consultation appointments)**
- **Chengyue Wu (TA)** 👋
  - **Email**: hillwu@connect.hku.hk **(For making consultation appointments)**
- **Mengkang Hu (TA)** 👋
  - **Email**: **(For making consultation appointments)**



Policy

- On-time assignments
  - 5 bonus points. The maximum score of the best assignment is 100+5. Do linear interpolation between 0 and 5 for the others. For example, we rank the scores of A1 in decreasing order, where the first place will have 5 bonus pts and the last will have 0. This is to encourage you to submit high-quality assignments on time.
- Late assignments
  - 15% mark deduction per day, and we do not accept submissions after 2 days.

Policy

- **Collaboration in study groups discussing the assignments is encouraged. However, you should write your own assignments.**
- **Plagiarism detection software will be used in every assignment.**
  - 0 marks for assignment/ project.
  - Serious cases will be reported to university, possibly resulting in disqualification.

Reminder

- This course does not require prior knowledge in database.
- However, knowledge in algorithms, data structures, and discrete mathematics, as well as programming skills are assumed.
- **Dedication** and **time** are needed!
- Materials for Tutorials, Lectures, and Assignments will be posted on Moodle!

Group Project (20%)

Project Descriptions. You are invited to develop an **Intelligent Course Management System (ICMS)** with a facial login component. It may have the following functions.

- When a student login with his/her face, his/her information such as name, login time, and welcome message will be presented in the graphics user interface (GUI).
- If the student has class within one hour, the corresponding course information, classroom address, teacher's message, links of Zoom, tutorial/lecture notes, other course materials and so on and so forth will be presented in the GUI. The student could click the links to redirect to Zoom or other materials. The GUI should also allow the student to send the above information to his/her email address by email.
- If the student does not have class at the moment, the GUI could present a personal class timetable for the student.
- The system should record the latest behaviour of the student, such as when he/she logins the system, how long the student stays in the system, etc.

Requirements.

- **Group**: 1-5 students as a group.
- **GUI**: Each group may design GUI based on the understanding of the above user requirement (You could make your own design choice, because in real project, clients typically don't know what they really want).
- **Database**: your database should have at least five tables. How to design the tables is your design choice.

Course Project

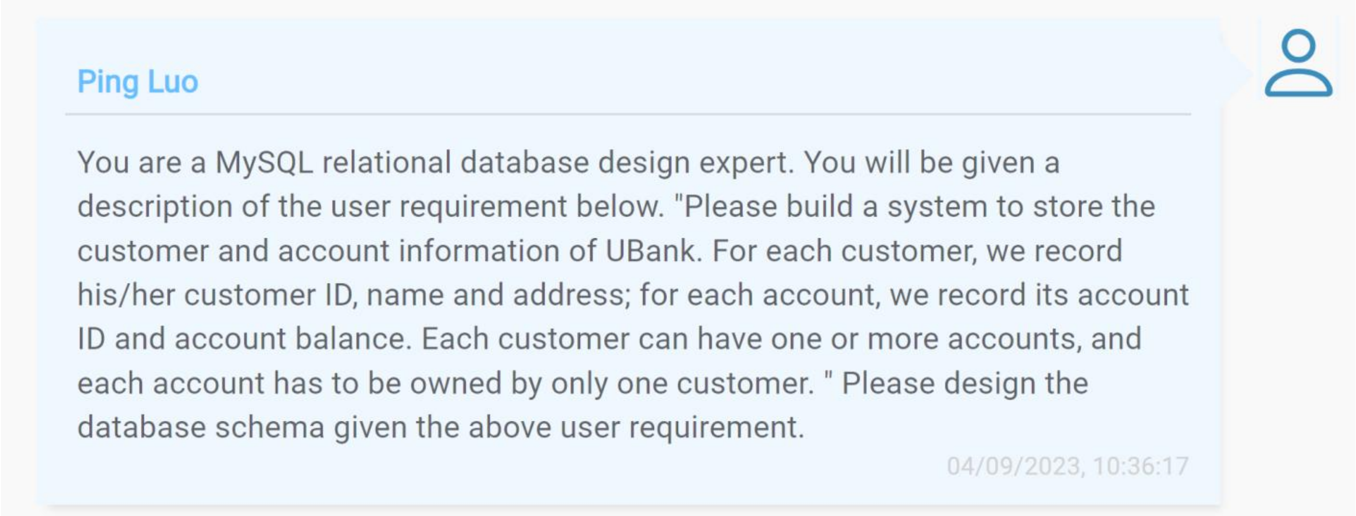
Development Tools.

- **Face Login:** Python + OpenCV (full codes provided.)
- **GUI**: Python GUI or Qt. (sample codes provided.)
- **Database**: Python + MySQL (sample codes provided.)
- **Other**: You can use any other Python packages if you see fit.

Marks (course project 20% of the final mark).

- 10% for software development. (4% GUI + 6% database)
- Other 10% for 5-minute presentation, including but not limited to development plan, milestones, contribution of each group member, video recording of demo, software design, database design (ER Diagram, tables), difficulties you encountered and how to solve them, etc.
- Live demo is allowed, but please make sure your program works well and stably in order to save time in presentation.
- Creative GUI design, creative software functions or creative DB design will have bonus points.

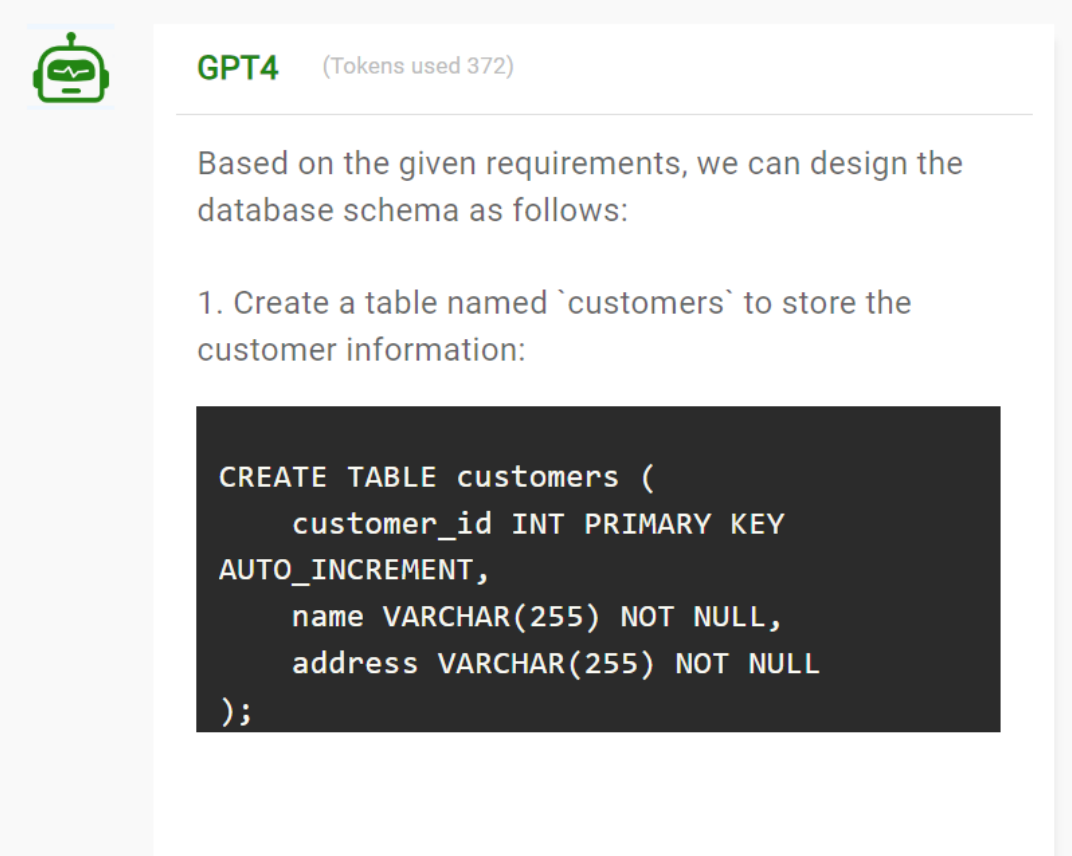
- You are welcome to involve ChatGPT in your design loop.
- You have to clearly discuss what and how ChatGPT help your projects (e.g., which part you used ChatGPT, the percentage (%) of contribution of ChatGPT in the project) in the 5-min presentation. Submit all prompts and returns from ChatGPT as supplementary materials.
- Excellent use of LLMs may have bonus points.



Ping Luo

You are a MySQL relational database design expert. You will be given a description of the user requirement below. "Please build a system to store the customer and account information of UBank. For each customer, we record his/her customer ID, name and address; for each account, we record its account ID and account balance. Each customer can have one or more accounts, and each account has to be owned by only one customer. " Please design the database schema given the above user requirement.

04/09/2023, 10:36:17



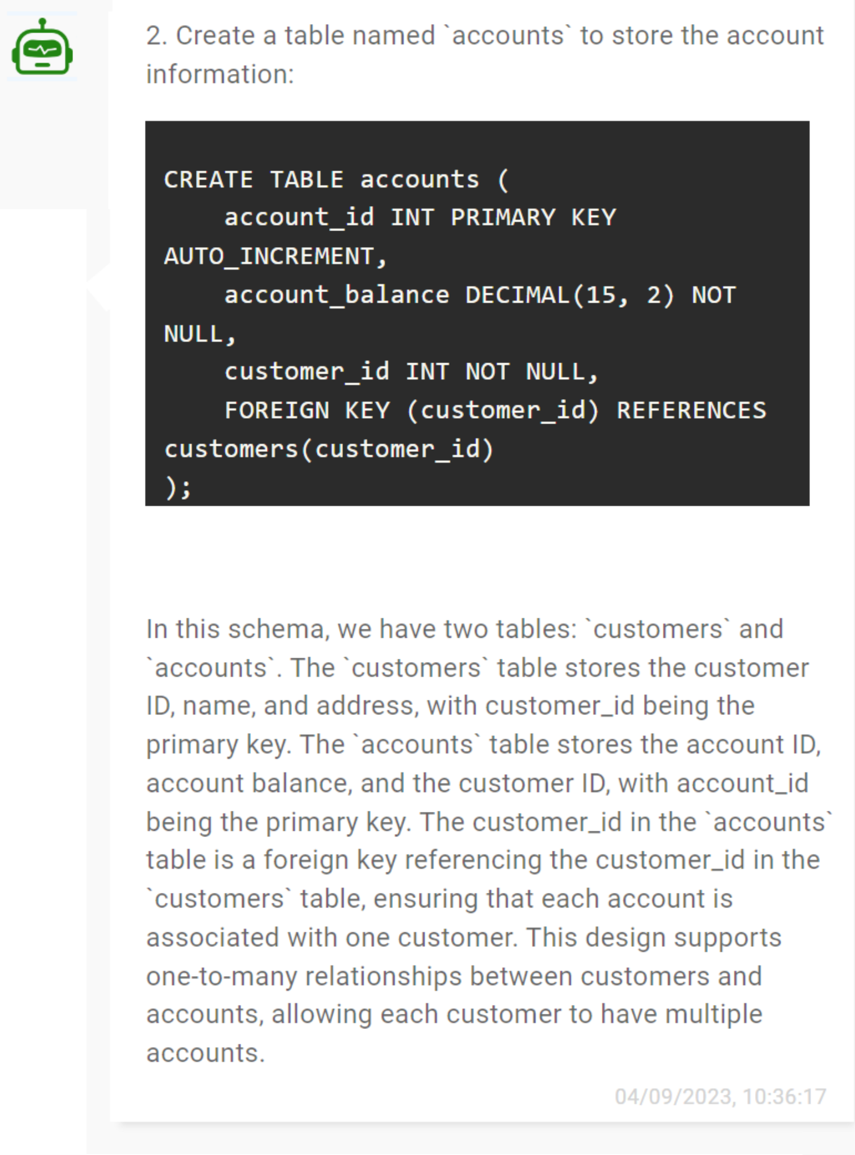
GPT4 (Tokens used 372)

Based on the given requirements, we can design the database schema as follows:

1. Create a table named `customers` to store the customer information:

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY
AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL
);
```

- It is a reasonable design as a starting point.
- **However, the correct answer should be based on the course materials and cannot be disputed!**



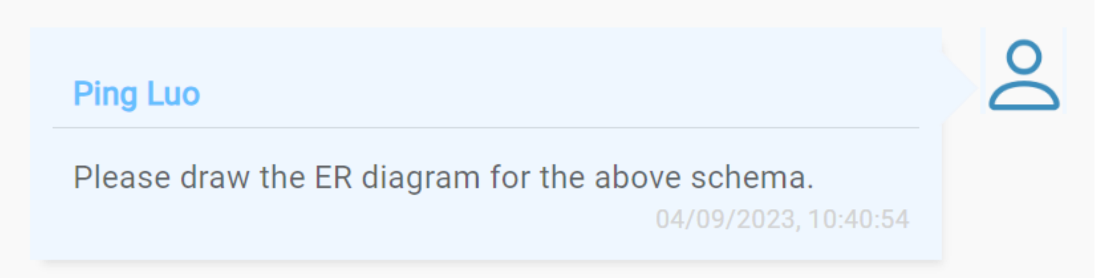
2. Create a table named `accounts` to store the account information:

```
CREATE TABLE accounts (
    account_id INT PRIMARY KEY
AUTO_INCREMENT,
    account_balance DECIMAL(15, 2) NOT
NULL,
    customer_id INT NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES
customers(customer_id)
);
```

In this schema, we have two tables: `customers` and `accounts`. The `customers` table stores the customer ID, name, and address, with customer_id being the primary key. The `accounts` table stores the account ID, account balance, and the customer ID, with account_id being the primary key. The customer_id in the `accounts` table is a foreign key referencing the customer_id in the `customers` table, ensuring that each account is associated with one customer. This design supports one-to-many relationships between customers and accounts, allowing each customer to have multiple accounts.

04/09/2023, 10:36:17

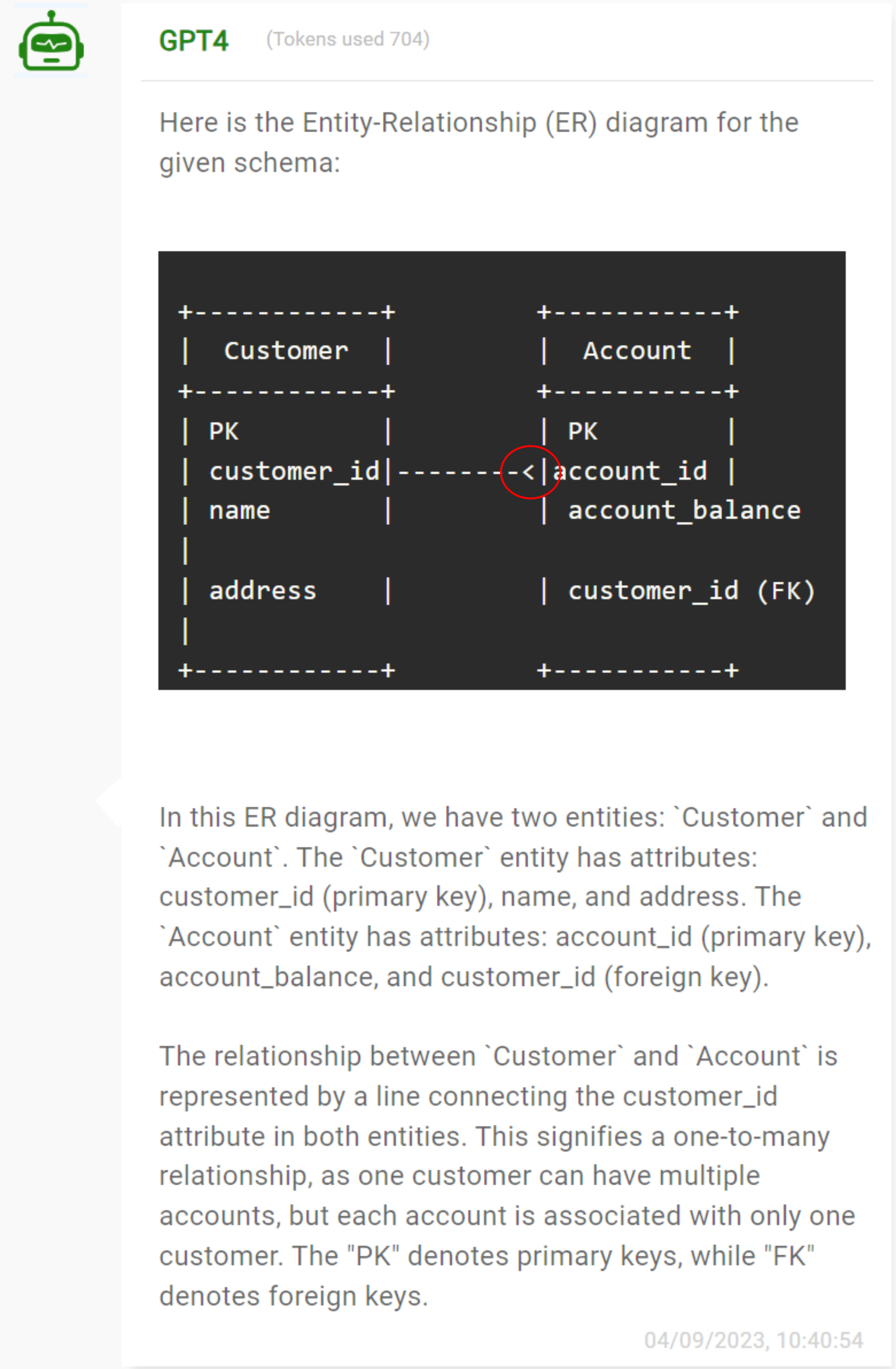
- It is a reasonable design as a starting point.
- **However, the correct answer should be based on the course materials and cannot be disputed!**



Ping Luo

Please draw the ER diagram for the above schema.

04/09/2023, 10:40:54



GPT4 (Tokens used 704)

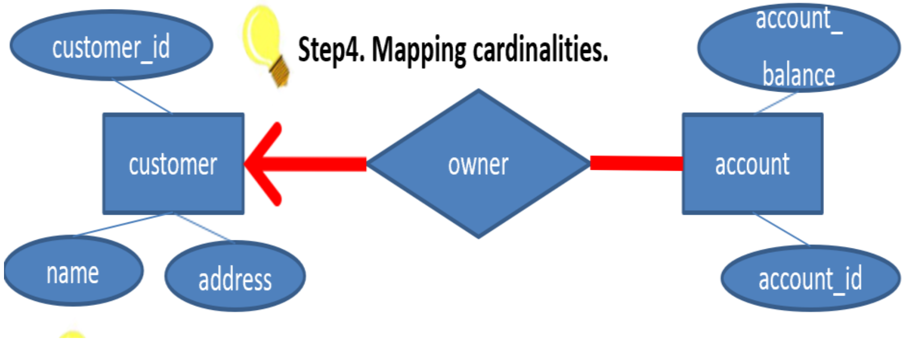
Here is the Entity-Relationship (ER) diagram for the given schema:

```
+------------+          +-----------+
|  Customer  |          |  Account  |
+------------+          +-----------+
| PK         |          | PK        |
| customer_id|-------<|account_id |
| name       |          | account_balance
|
| address    |          | customer_id (FK)
|
+------------+          +-----------+
```

In this ER diagram, we have two entities: `Customer` and `Account`. The `Customer` entity has attributes: customer_id (primary key), name, and address. The `Account` entity has attributes: account_id (primary key), account_balance, and customer_id (foreign key).

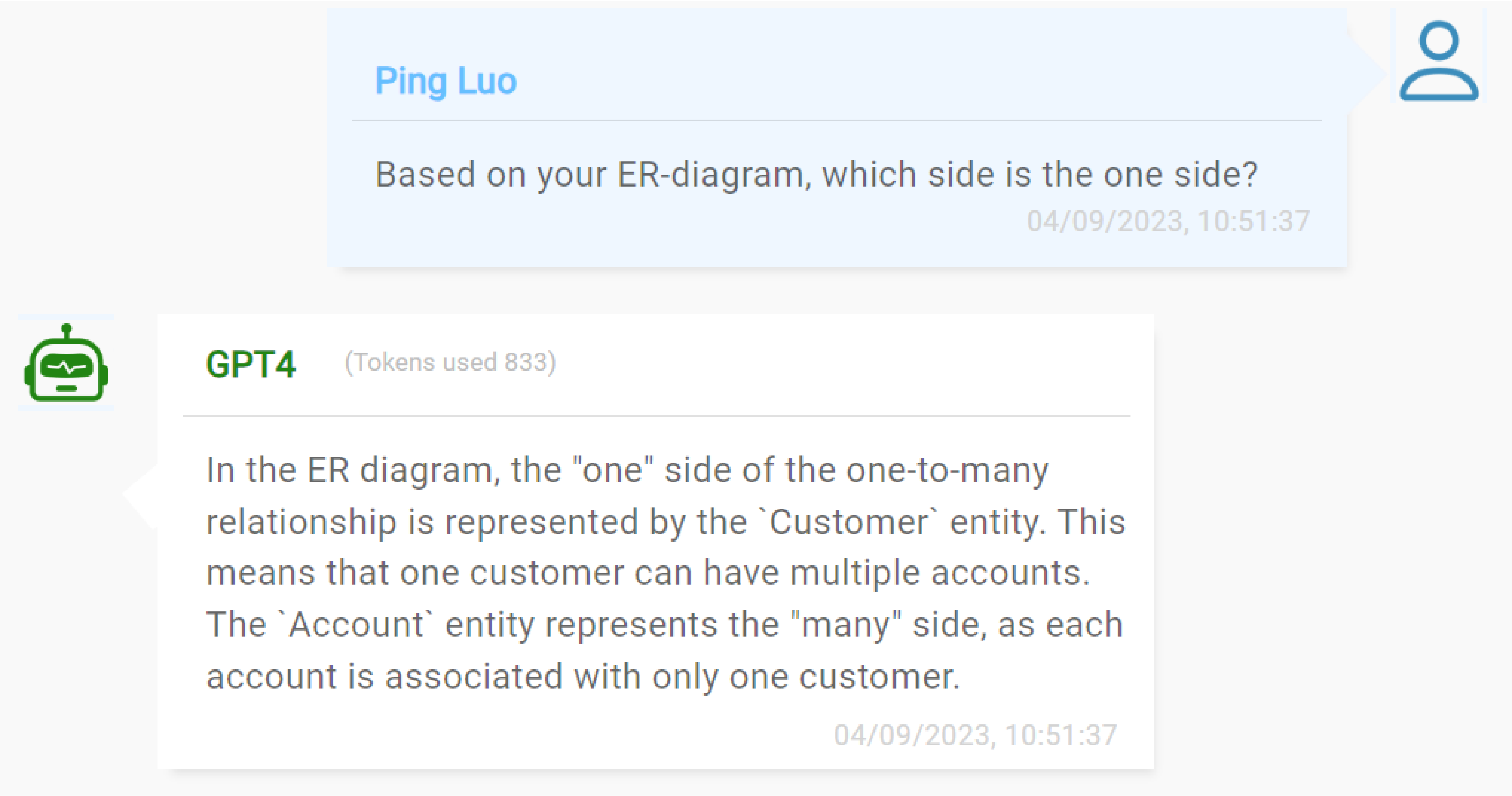
The relationship between `Customer` and `Account` is represented by a line connecting the customer_id attribute in both entities. This signifies a one-to-many relationship, as one customer can have multiple accounts, but each account is associated with only one customer. The "PK" denotes primary keys, while "FK" denotes foreign keys.

04/09/2023, 10:40:54



Step4. Mapping cardinalities.

ChatGPT is different from the Lecture! Follow the lecture!



Ping Luo

Based on your ER-diagram, which side is the one side?

04/09/2023, 10:51:37

GPT4 (Tokens used 833)

In the ER diagram, the "one" side of the one-to-many relationship is represented by the `Customer` entity. This means that one customer can have multiple accounts. The `Account` entity represents the "many" side, as each account is associated with only one customer.

04/09/2023, 10:51:37