CMP 420 FINAL PROJECT

Robert Guzman 05/23

Final Project

I am currently in the process of developing a comprehensive database specifically tailored for the computer science department at Lehman College. The main objective of this database is to enable efficient tracking of students who are seeking assistance within the school's various labs. To achieve this, my database incorporates several essential components, including an Entity-Relationship (EER) diagram, Normalization techniques, a relation model, and the corresponding normalization procedures. By employing these advanced methodologies, the database will seamlessly organize and manage pertinent student information, ensuring a streamlined system for tracking and providing support to students throughout their academic journey at Lehman College.

EER diagram

I opted to utilize an Enhanced Entity-Relationship (EER) diagram for this project, primarily due to the distinct categorization of the employee entity into two separate categories: Tutor and Coordinator. To create the EER diagram, I employed draw.io as a tool, which facilitated the creation of the necessary entities essential to this project. These entities include Employee, Coordinator, Tutor, Student, Book, and Courses, each accompanied by their respective attributes and a primary key, namely e_emplid. Additionally, I introduced the Employee entity with two sub-entities, ensuring a clear distinction between the Coordinator and Tutor entities while acknowledging their shared information and attributes within the broader Employee category.

Relational Model

The diagrams I have created exhibit relational connections among them. For instance, considering the scenario where a dean can possess multiple laptops and books that can be loaned to students in the computer science lab, I carefully worked through the details and devised appropriate relationships, including many-to-many (m to n) and vice versa (n to m) relationships. This meticulous approach ensured that all the criteria were met.

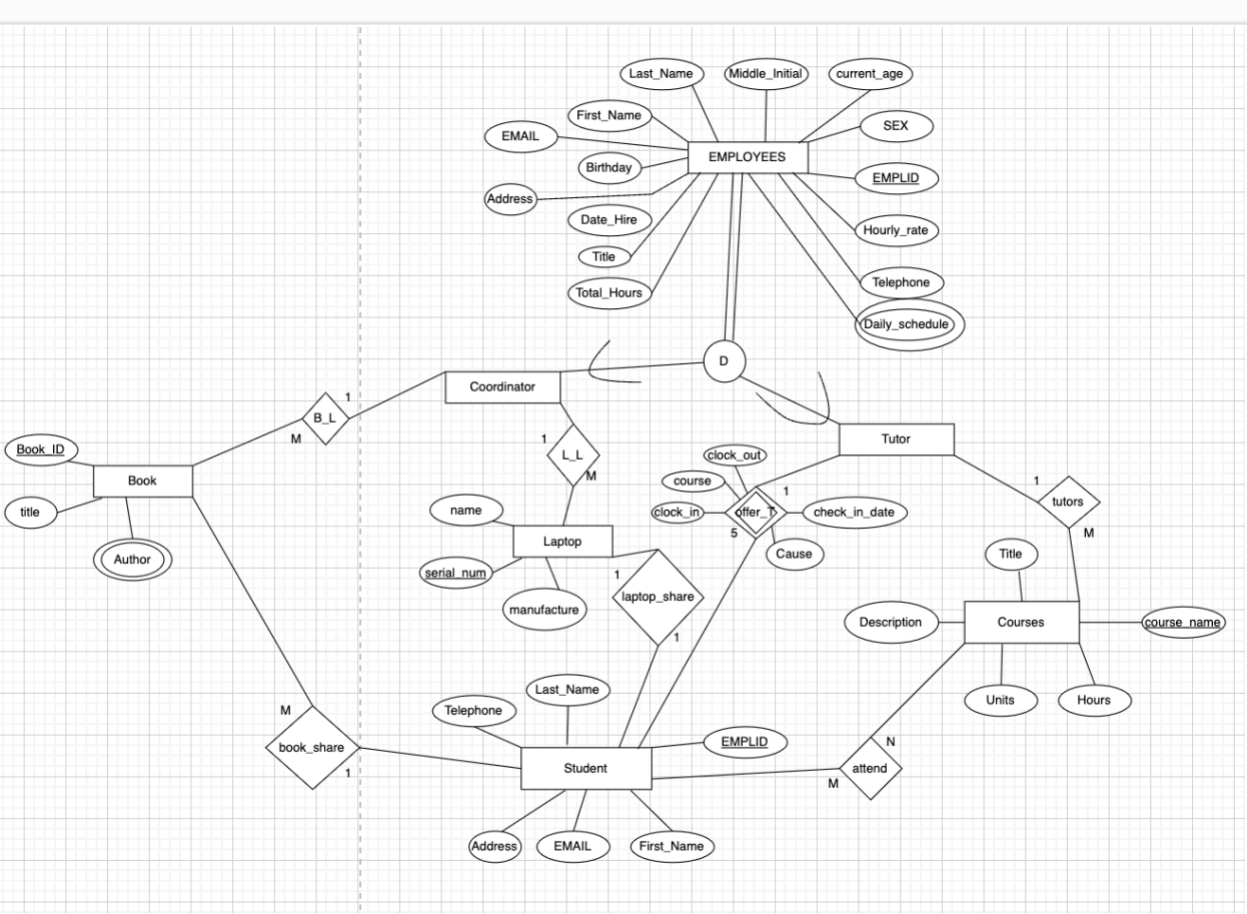
Now, the real fun begins as I transform the diagram into a database schema, essentially representing the "tables" involved. During this process, I meticulously break down the entities and establish connections between the attributes, considering their dependencies. However, I introduced a unique aspect by incorporating two additional tables, namely book_author and employee_schedule. These tables have their respective unique keys, enabling me to precisely identify their origins and establish the relationships they have with one another. Furthermore, both tables incorporate foreign keys that effectively establish the interconnections between them.

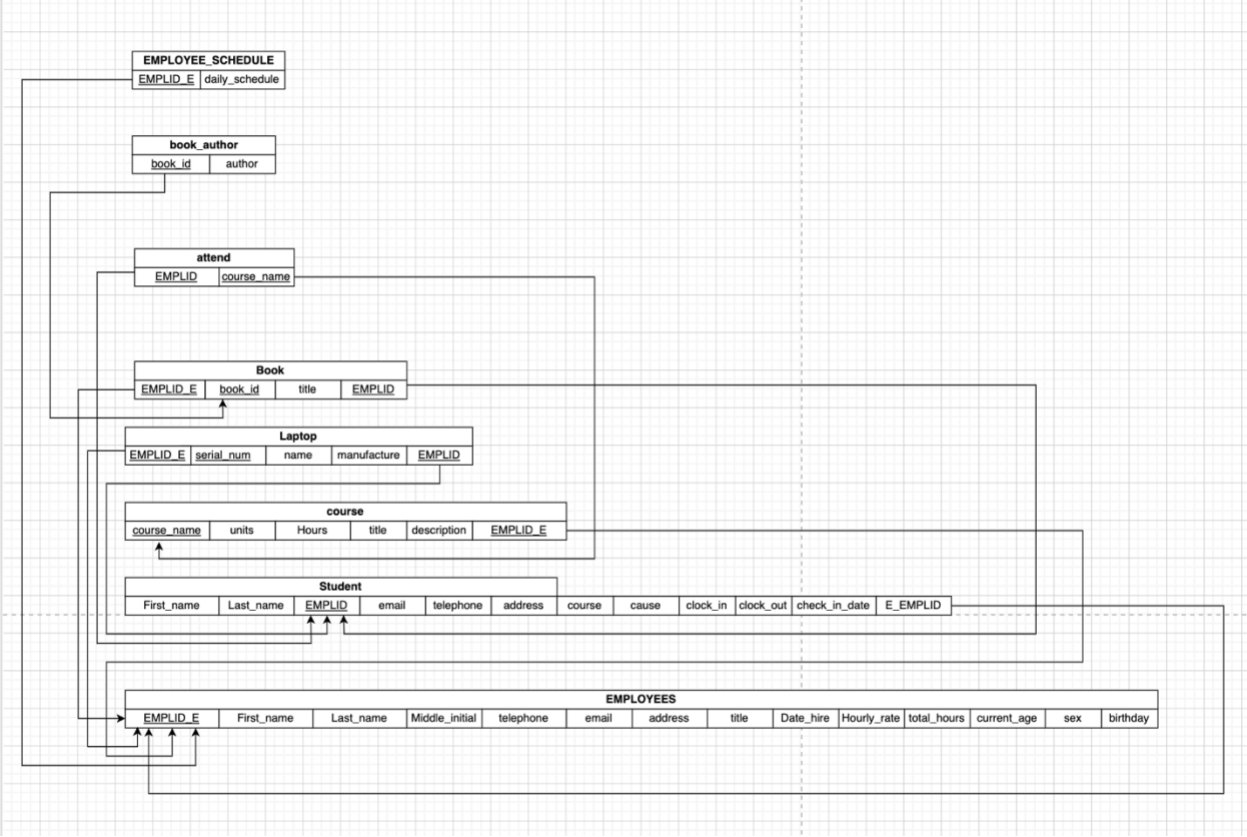
Queries

One of the most challenging aspects of this project has been handling the famous queries. However, I tackled this task with determination and perseverance. To begin, I created the database using the "create database if not exist;" statement, ensuring its existence. To streamline the process of inputting data into SQL without encountering any bugs or issues.

To retrieve the data and view it in the console (or the output in SQL terminology), I employed the "Select * from course;" statement. This query allowed me to fetch all the data stored in the "course" table and display it conveniently. Similarly, I used the "select first_name, last_name" statement to extract specific columns from the "employee" table, repeating this process for all relevant tables in the database.

By employing these query techniques, I successfully retrieved and displayed the necessary data from the SQL database, overcoming the challenges associated with querying effectively.






```
DROP TABLE IF EXISTS `employee_info`;

CREATE TABLE `employee_info` (
  `work_info`  serial NOT NULL,
  `title` TEXT default NULL,
  `date_hire` varchar(255),
  `hourly_rate` float(24) default NULL,
  `total_hours` mediumint default NULL,
  `emplid_e` int  UNIQUE NOT NULL,
  PRIMARY KEY (`work_info`)
) AUTO_INCREMENT=1;

INSERT INTO `employee_info` (`title`,`date_hire`,`hourly_rate`,`total_hours`,`emplid_e`)
VALUES
  ("computer","Mar 2, 2022",58.36,4,36523701),
  ("Tutor","Dec 26, 2021",8.94,26,59587473),
  ("Tutor","Mar 18, 2022",53.69,7,77500899),
  ("Tutor","Nov 14 2022",51.72,33,28349302),
  ("Tutor","Feb 19, 2022",64.53,23,20757820);
```

```
 Select* from employee_info;
 DROP TABLE IF EXISTS `student_needs`;

CREATE TABLE `student_needs` (
 `learner` varchar(3) NOT NULL,
 `cause` TEXT default NULL,
 `clock_in` TEXT default NULL,
 `clock_out` TEXT default NULL,
 `check_in_date` varchar(255),
 PRIMARY KEY (`learner`)
) AUTO_INCREMENT=1;

INSERT INTO `student_needs` (`learner`,`cause`,`clock_in`,`clock_out`,`check_in_date`)
VALUES
 ("#21","Not understanding the class ","12PM","4PM","Aug 26, 2022"),
 ("#45","homework help ","12PM","3PM","Jan 15, 2023"),
 ("#68","exam review ","12PM","4PM","Dec 22, 2022"),
 ("#78","project criteria ","10AM","5PM","Oct 26, 2022"),
 ("#65","quiz preparation ","11AM","5PM","Jul 24, 2023");

 Select* from student_needs;
```

Conclusion

In conclusion, this project has been a highly valuable endeavor, despite the challenges I faced in working with queries, which have always been a bit difficult for me to grasp. Nevertheless, I am determined to continue expanding this project, as it serves as an impressive showcase for future technical interviews and provides a solid foundation of SQL knowledge for my upcoming internship.

One noteworthy aspect is that I undertook this project as an individual endeavor, which greatly contributed to the enhancement of my SQL skills. Working independently allowed me to delve deep into the intricacies of SQL, honing my abilities and further developing my understanding of the subject matter.

Moving forward, I am excited about the prospects of this project, both in terms of personal growth and professional opportunities. It has provided me with a tangible demonstration of my SQL expertise and will undoubtedly serve as a valuable reference point as I pursue future endeavors in the field.

Who made the highest dollar per hour in the CS lab?

```
Select MAX(hourly_rate) from employee_info;

Select hourly_rate from employee_info
order by hourly_rate desc;
```