

README:

- Primary keys (PK) are in Bold and Underlined
- Strong and Weak relationships are represented using Solid and Dashed lines.
- Entities that are colored with Gray are bridges (to capture N:M relationships)
- Entities that are in Blue are regular entities

Assumption:

1. A student must be assigned to one schedule and a schedule can belong to one student that is why there is One to One relationship.
2. Every student will be assigned mandatorily to only one group that is related to one Project which is why there is One to One relationship between the Group_project and Student. The relationship between them is weak.
3. Every student may rate many instructors, courses, projects and every instructor, project, course may be evaluated by different students; therefore, I created 3 different bridges each bridge has a composed key to make it unique. In this case I will have one to many relationships between the Student_Poject_Rating_Bridge and a Student entity and One to Many relationship between the Student_Poject_Rating_Bridge and the Project entity. Same goes with (course, instructor rating entity)The bridge makes the relationship strong so there was a solid line.
4. Every Room can be used for many projects and a Project can be held in different rooms, that is why there is a bridge (Room_Project_Bridge) to capture the many to many relationships, so the Entity bridge will have One to Many relationships for the Room and Project entity. The same thing goes with Room and Course.
5. Every Room has zero or many tables and each table can only be existed in one room that is why there is Many to One relationship between Room and Table entity.
6. A Box entity can only belong to one table and a table can only has a single box , therefore there is a One to One relationship between the two entities.
7. Count attribute in the Part_Box_Bridge entity tells how many parts there are for a single box.
8. Same Parts can belong to different Box and a Box can have many parts, by using the bridge Part_Box_Bridge we have One to Many relationships between the two entities.
9. is_it_damage attributes in Damage_parts entity has a true or false values, if it is true Damage_Payment_Bridge calculates how much the damage should cost and assign the cost to a student who caused it to pay for it.
10. Damage_parts entity captures any damages happen to the parts which is used during the project by the students who improperly misused them. Because there could be Many students cause the damage to Many parts a bridge exists Damage_Payment_Bridge to capture this relationship. A One to Many relationship exist between the Damage_Payment_Bridge and the two entities.

11. Every order has many parts , therefore there is One to Many relationship between the Order and the Parts entities . In addition, very order can have many suppliers and each supplier can provide many orders therefore a bridge entity was created to capture this relationship (Order_Supplier_Bridge) where there is two one to many relationships associated with the bridge entity.
12. Every TextBook exists in one Library (one and only one) and the Library can contain the TextBook once; therefore there is a One to One relationship between the two entities.
13. Each Class MAY has multiple TextBook and each TextBook can be assigned (zero or many times to different Class) by many Instructors, therefore there is a bridge entity Textbook_Instructor_Class_Bridge exists to show the one to many relationships between the three entities.
14. Each Instructor can have multiple wages(he can get paid many times) based on the number of hours he taught or supervised that is why there is many to one relationship between Instructor and Payroll entities.
15. Every instructor teaches multiple classes, and a class can be taught by multiple instructors (M:N); therefore there is a Class_Instructor_Bridge that connect Class and Instructor entites and has a One to Many relationship between the two entities. Same thing goes with Project and Instructor entities.

Entities:

There are 27 entities:

1. **Student:** contains the attributes related to student's general data. The **student id** (Primary Key) of the table.
2. **Fee:** The purpose of this table is to keep tracking students payment of the flat fee which is happening in the beginning of the semester. The **Fees id** (Primary Key) of the table.
3. **Schedule:** This table help the student locate which room is used for the class and which room is used for the project at specific time and day. The **schedule id** (Primary Key), and student id (Primary Key & Foreign key) >> (composite key)
4. **Group Project:** This table help to know the name of the student group and which students are assign to which group and the which table they will be using. The **group Id** (Primary Key), project id (Primary Key & Foreign key) >> (composite key)
5. **Class:** contains class information such as its id and its name. The **class id** (Primary Key) of the table.

6. **Student lib bridge:** is a bridge entity to link the student and the library. The **student id** and **book id** are both (primary and foreign key) for the table. Has one to many relationship with both tables (student, library)
7. **Library :** used to store the book , its description and its author. The book id (Primary Key) of the table.
8. **Text Book:** contains general information about the textbook the professor will assign to the students. The text book id (Primary Key) of the table.
9. **Room:** contains room name and its id is the (Primary Key)
10. **Room project Bridge:** connect the room table and project table and has one to many relationship between the two tables. Its primary key and foreign key is composed of room id and project id.
11. **Project:** Tells the name and id of the project , project id (primary and foreign key) for the table.
12. **Room class bridge:** connect the room table and class table and has one to many relationship between the two tables. Its primary key and foreign key is composed of room id and class id.
13. **Text book instructor class bridge :** helps to know which textbook is assign to which class by which professor and each relationship between these tables are one to many relationships.
14. **Instructor:** contain general information about the instructor. Instructor id (primary key) for the table.
15. **Payroll:** store how much the professor will receive money by considering the number of hours he teaches and supervised.
16. **Rate:** Contains students rating for their professor approach of teaching and overall satisfaction, classes and project they have taken. rate id (primary key) and student id is (primary key and foreign key) for the table. Both keys work together as a composed key where they can specify each row accurately.
17. **Table:** Tells which group are sitting on it and in which room this table exist. Table id is (primary key)
18. **Box:** Is related to specific project and sit on specific table, so this table link the table and project table to know which box we need. Box id (primary key)
19. **Parts box bridge:** There are many to one relationship between parts box bridge and (parts , box)tables. The key is composed of parts and box keys working as primary and foreign key at the same time.
20. **Parts:** contains general information about the parts that will be used in a project. Parts id (primary key)
21. **Order:** This table has information about the request/ order of the parts. It store information such as when the parts will be delivered and when the order was requested and in which date. It has a composed key (order id PK , parts id FK, PK)

22. **Order supply bridge:** link the order , supply , bridge because many orders can be done from different supply and many suppliers provide same parts. The relationship between the bridge and other tables is one to many and the relationship is strong.
23. **Supplier:** Stores general information about the supplier such as the location , email address. The supplier id is the (primary key)
24. **Damage parts:** capture which parts where damage and write a description about damage. The damage id (primary key)
25. **Damage payment bridge:** link the students who cause the damage with the damaged parts and assign a cost where the students should pay for it. The relationship between is one to many in both ways. The (primary key) is composed of student id PK & FK and damage id working as PK& FK.
26. **Class Instructor Bridge:** connect the Class table and Instructor table and has one to many relationship between the two tables. Its primary key and foreign key is composed of class id and instructor id.
27. **Project Instructor Bridge:** connect the Project table and Instructor table and has one to many relationships between the two tables. Its primary key and foreign key is composed of project id and instructor id.