**CPCS241-Database I-Spring2021-Project**

**[School Database]**

**DB Design**

**Group No:**

|  |  |
| --- | --- |
| **Student Name** | **Student Number** |
| **Abdualrhman Alsharif** | **2046431** |
| **Mohamed Al-Namri** | **2036222** |
| **Ahmed qahtan** | **2046143** |
| **Ibrahim basher** | **2045910** |

**Contents**

[PART I: Analysis 4](#_Toc2107209730)

[1 Problem Definition and Data Requirements 4](#_Toc867919534)

[1.1 Problem Description 4](#_Toc390743572)

[1.2 Data Requirements 4](#_Toc1928148712)

[1.3 Business Rules 6](#_Toc692705563)

[1.4 Intended Output of the system 7](#_Toc315793910)

[PART II: DB DEISGN 7](#_Toc1882660358)

[2 ER Diagram Design 8](#_Toc1146003723)

[2.1 ER diagram 8](#_Toc709492070)

[2.2 Design of Business Rules 9](#_Toc52533380)

[3 ER-to-logical schema mapping 10](#_Toc1453758600)

[3.1 Mapping of Regular Entity Types 10](#_Toc899888638)

[3.2 Mapping of Weak Entity Types 13](#_Toc1507547818)

[3.3 Mapping of binary 1-1 relationship types 13](#_Toc1151178639)

[3.4 Mapping of binary 1-N relationship types 15](#_Toc1084757445)

[3.5 Mapping of binary M-N relationship types 17](#_Toc771141507)

[3.6 Mapping of multivalued attributes 18](#_Toc1451198140)

[3.7 Mapping of n-ary relationship types 18](#_Toc1279662768)

[3.8 Schema Diagram 19](#_Toc1324537581)

[4 Normalization 20](#_Toc1128521423)

[4.1 First Normal Form 21](#_Toc569023110)

[4.2 Second Normal Form 21](#_Toc387004524)

[4.3 Third Normal Form 21](#_Toc379618060)

[5 Final DB Schema Diagram 21](#_Toc916088120)

[PART III: IMPLEMENTATION 22](#_Toc874329604)

[6 Table Creation Script 23](#_Toc1003451153)

[6.1 23](#_Toc1819390849)

[<Employee> TABLE 23](#_Toc87258476)

[6.2 <Emp\_Phone> TABLE 23](#_Toc422523339)

[6.3 <Teacher> TABLE 24](#_Toc1003097061)

[6.4 <IT\_Worker> TABLE 24](#_Toc2033575529)

[6.5 <Cleaning\_Staff> TABLE 24](#_Toc1545807843)

[6.6 <librarian> TABLE 25](#_Toc2142351813)

[6.7 <lib\_managers> TABLE 25](#_Toc1875819545)

[6.8 <Library> TABLE 26](#_Toc523690607)

[6.9 <Books> TABLE 26](#_Toc2083740294)

[6.10 <Members> TABLE 26](#_Toc1691795078)

[6.11 <Students> TABLE 27](#_Toc1641786411)

[6.12 <courses> TABLE 27](#_Toc1400483845)

[6.13 <Enrollment> TABLE 28](#_Toc595214693)

[6.14 <Legal\_Guardian> TABLE 29](#_Toc1545201342)

[6.15 <LG\_Info> TABLE 29](#_Toc1420874145)

[6.16 <LG\_Phones> TABLE 29](#_Toc1984117036)

[6.17 <Grades> TABLE 30](#_Toc1809600419)

[6.18 <Exams> TABLE 30](#_Toc1645549869)

[6.19 <Rooms> TABLE 31](#_Toc913109728)

[6.20 <Class> TABLE 31](#_Toc1469328849)

[7 Constraints Script 32](#_Toc208539535)

[8 Queries 33](#_Toc450299437)

[8.1 <Year\_Works library members> 33](#_Toc1348042899)

[8.2 <Covid-19 students> 34](#_Toc1091757418)

[8.3 <Non-Physics 1 teachers> 34](#_Toc18071557)

[8.4 <All failed students in 2021> 35](#_Toc1094093962)

[8.5 <Fundimental year students> 36](#_Toc1694964439)

[APPENDIX 37](#_Toc1906450633)

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

Dealing with a massive amount of data requires a lot of organization and to be able to do that you need help from a computer. Dumping all data in files and some excel sheets is inefficient and slow. Furthermore, most schools are having a lot of issues dealing with this massive data. However, schools need management software to keep all members' data such as students, all employees including teachers, IT-Workers, and even cleaning staff members. In addition to that, we need to manage the courses and which student took which course and when plus the grades he got. So, we have decided to solve this problem by creating a full database system that includes all the massive data that any school needs.

1.2 Data Requirements

* **Employees Entity:**
* *EmployeeID*: a unique ID for every employee.
* *Name*: a composite attribute that consists of the first name (Fname), middle name (Mname), and last name (Lname).
* *Sex*: the gender of the employee.
* *Salary*: the salary of the employee.
* *EMP\_Phone*: The phone number of the employee.
* *BDate*: the birth date of the employee.

Note: Employee is a superclass with 4 subclasses: Teacher, Cleaning staff,

IT-Worker and Librarian.

* **Teacher Entity:**
* *Specialization*: The Specialization of the teacher.
* **IT-Worker:**
* *Specialization*: The Specialization of the IT-Worker.
* **Students Entity:**
* *StudSSN*: a unique ID for every employee.
* *Name*: a composite attribute that consists of the first name (Fname), middle name (Mname), and last name (Lname).
* *BDate*: the birth date of the student.
* *STU\_Phone*: The phone number of the student.
* **Class entity:**
* *Class\_ID*: a unique id for every class.
* *Period*: the time that takes by each class.
* *Start\_time*: the start time of the class.
* **Courses Entity:**
* *Course\_ID*: a unique ID for every course.
* *Course\_Name*: the name of the course.
* *prerequisite*: the course required for registration in a new course.
* **Enrollment Entity:**
* *Quarter*: the name of the quarter.
* *Year*: the year of the enrollment.
* **Exams Entity:**
* *Exam\_ID*: a unique ID for every exam.
* *Exam\_Mark*: the mark of the exam.
* *Exam\_Date*: the date of the exam.
* *Exam\_Time*: the time of the exam.
* **Grades Entity:**
* *Grade\_ID*: a unique ID for every grade.
* *Attend*: the attendance grade.
* *Behaviour*: the behaviour grade.
* *Final\_Mark*: the final mark for each course, which is a derived attribute.
* **Legal Guardian Entity:**
* *Name*: the name of the Guardian.
* *Bdate*: The Birth date of the Guardian.
* *Relationship*: the relationship between Student and his guardian.
* *Sex*: the gender of the guardian
* *LG\_Phone*: Legal Guardian’s Phone number.

**Note**: the Legal Guardian Entity is a weak entity.

* **Rooms Entity:**
* *Room\_No*: a unique ID for every room.
* *Floor\_NO*: the number of the floor.
* *Capacity*: the capacity for each room.
* **Library Entity:**
* *Lib\_ID*:a unique ID for every library.
* *Lib\_Name*: the name of the library.
* **Members Entity:**
* *Mem\_ID*: a unique ID for every Library membersز
* *Mem\_Name*: name of the member.
* **Books Entity:**
* *Book\_ID*: a unique ID for every book.
* *Book\_Name*: the name of the book.
* *Author*: the author of the book

1.3 Business Rules

A school job is to hire employees on the Employees tables, and accepts students from Student table, and stores their data and manipulate it, the school offers courses for the student that are stored in the Courses table, the courses is taught by the employees a school has an Id, name and an address, every school has its own Employees, students -and library-, although they teach the same courses

**Employee**

The employees are the people who are going to run and manage the school, and the employees differ from teacher to supervisors to working staff, each and every one of them are important to get the school going. managers will be responsible for the teacher and student affairs, the teachers will be in charge of subjects to teach the student. The IT workers will be in charge of managing the data of students and employees, the cleaning staff must keep the school clean, and finally there are the librarians who will be in charge of the library.

**Employee/Teacher**

A teacher **[GIVES]** a class based on his specialty.

**Employee/Cleaning\_staff**

All cleaning staff are reponsible of all the school.

**Employee/IT\_Worker**

Each IT worker is responsible of technical things going in school based on his specialty such as database admin, data entry, data analysis.

**Employee/Librarian**

Each library has one library manager. Using **Emp\_ID** as FK

**Library**

The students can go to the library which is managed by the Librarian employee to have access to lots of books that can help them, whether in their courses or just for fun.

**Class**

A class is the session where teachers teach courses and it does have **Teacher\_ID** as a foreign key.

**Students**

First the students must provide their data and their parents info, after a student is accepted in the school, a **Stud\_ID** is automatically generated.

**Legal guardian**

The table which stores parents or people who are responsible of each student.

**Courses**

The school defines what courses are taught and who are the teachers that are going to teach the courses. The courses have exams that determine whether the students are going to pass the course or not. The course will be given a class with a specific time that does not contradict with other classes, so students won't miss them and every class will also be given a classroom to hold the lectures in.

**Enrollment**

It is the table where all semester data is stored in such as, enrollment year, the status of passing. It has **Stud\_ID**, **Course\_ID** as a foriegn key.

**Grades**

Includes all marks for **Stud\_SSN** such as, behavior, attendance, year works.

1.4 Intended Output of the system

**Output & Queries:**

* Display all employees with their jobs type.
* Display all students with their courses.
* Display weekly courses schedule.
* Display all students with their dependence.
* Display report for a specific Student.
* Search for a Student by ID.
* Search for available courses students can enrol in.
* Search for a teacher by ID.
* Display grades of specific students.
* etc…

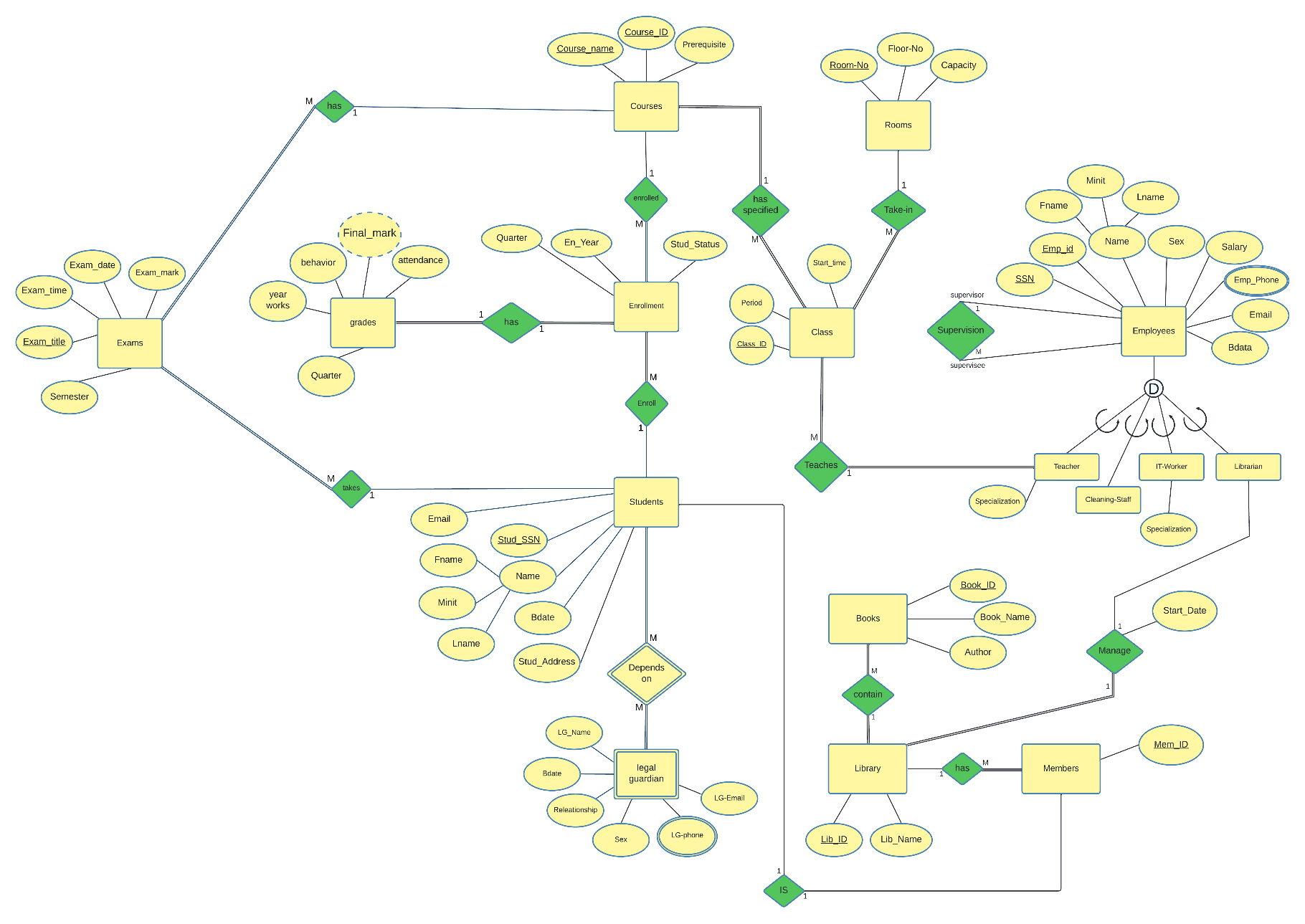
***Transactions:***

* Insert a new student.
* Update supervisor of employees.
* Assign a teacher to a specified course.
* delete a teacher.
* Assign room to specified course.

PART II: DB DEISGN

2 ER Diagram Design

2.1 ER diagram

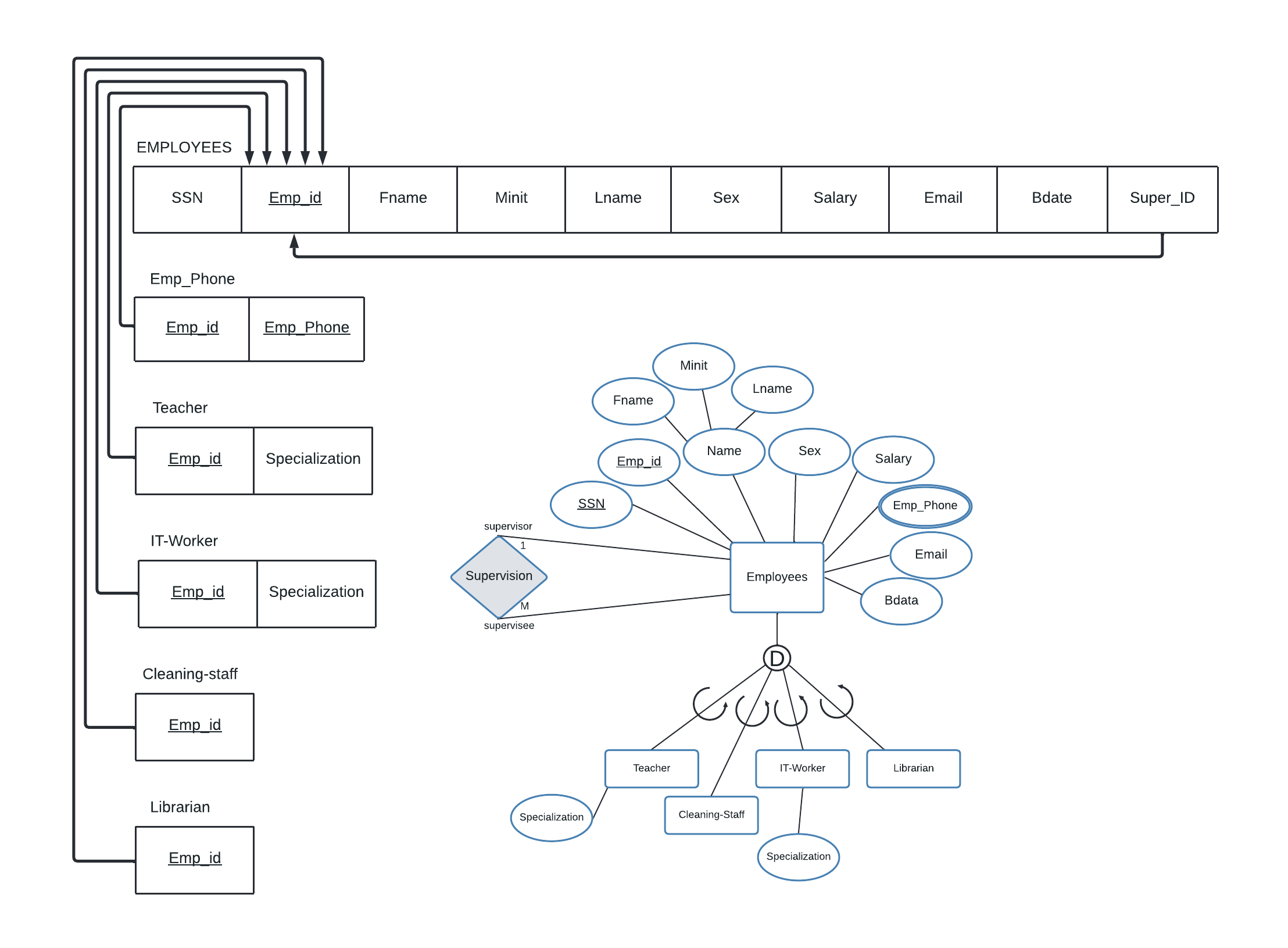


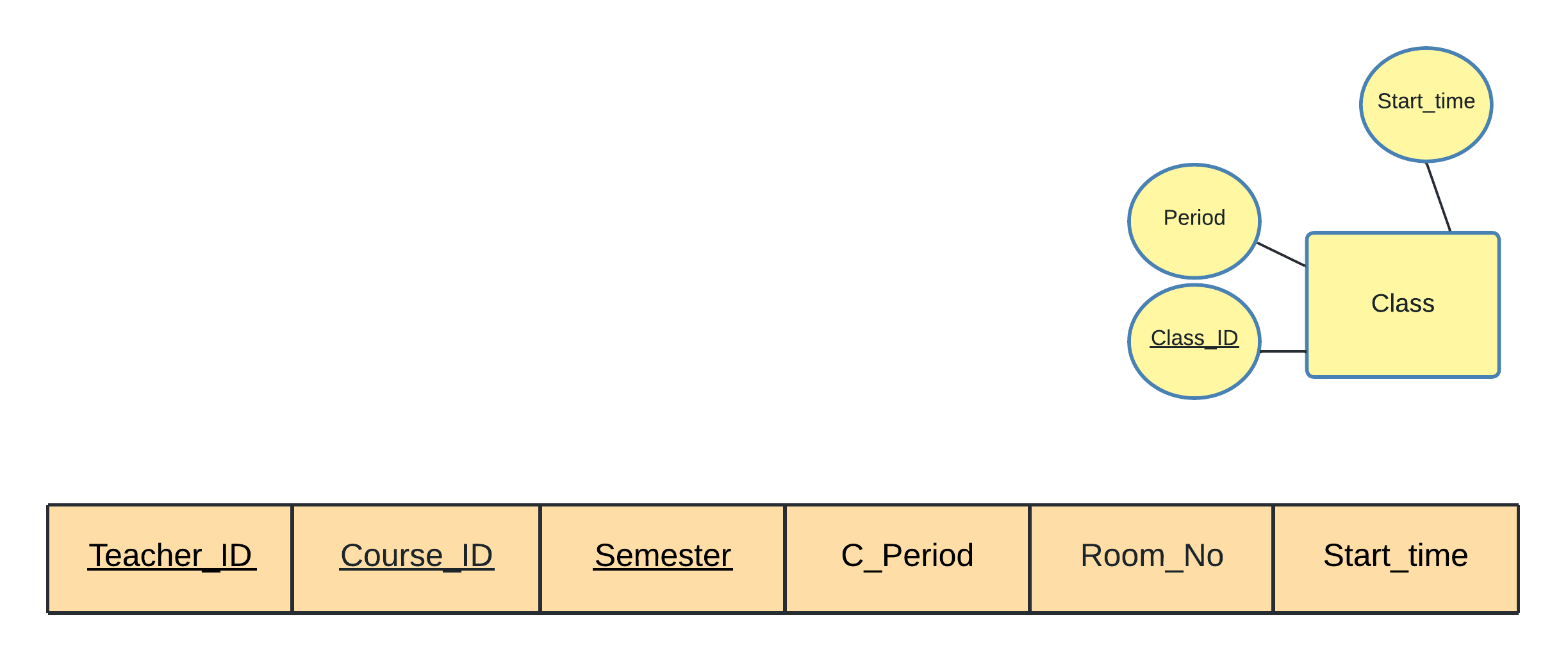
2.2 Design of Business Rules

|  |  |  |
| --- | --- | --- |
| Business Rule | Design Decisions | Justification (if any) |
| Each employee must have a specialization. | Superclass/subclass relationship with disjointness and completeness constraints. | Total since every employee has  a specialization. It’s disjointed since each employee has only one specialization. |
| Each teacher gives classes. | 1: N binary relationship between Teacher and Class. | A teacher gives many classes  while a class is given by one  teacher only. |
| No room can be taken  by multiple class at the  same period.  And multiple class can  be made for different  available rooms. | 1: N Binary relationship  between class  and ROOM. | Each room can be taken only  by one class for a certain period  And since some class can be  given in the same room. |
| Each course has many classes to be given in. | 1: N Binary relationship  between Course  and Class | A course can be given in multiple classes while 1 class can give one course only. |
| Each enrollment can contain multiple courses. | 1: N Binary relationship  between Course  and Enrollment | When enrolling a semester, you can enroll many courses but you can’t enroll two semesters. |
| Each room takes one class. | 1: 1 relationship  between room and Class | A class can be given in one room and 1 room can have one class only. |
| Each student can have many exams while one exam has one student. | 1: N Binary relationship  between students  and exams. | A student has more than one exam according to the number of courses in the same semester, while one exam can only be for one student. |
| One course has many exams.  The exam cannot include many courses. | 1: N Binary relationship  between courses  and Exams | One course has many exams, But  One exam can not include many courses. Only one course for each exam. |
| Each enrollment (semester) contains many course’s grades | 1: N Binary relationship  between Enrollment  and Grades | One enrollment includes many courses which has many grades and one grade is included in one enrollment |
| One student can enroll one time only | 1:1 relationship  between Student and Enrollment | One student can have one enrollment at a semester and one enrollment is being enrolled by one student |
| Each Student can have many legal guardian. | 1:M binary relationship  between Students and  Legal guardian. | One student does have many or 0 legal guardians while many legal guardians only does have one dependant |

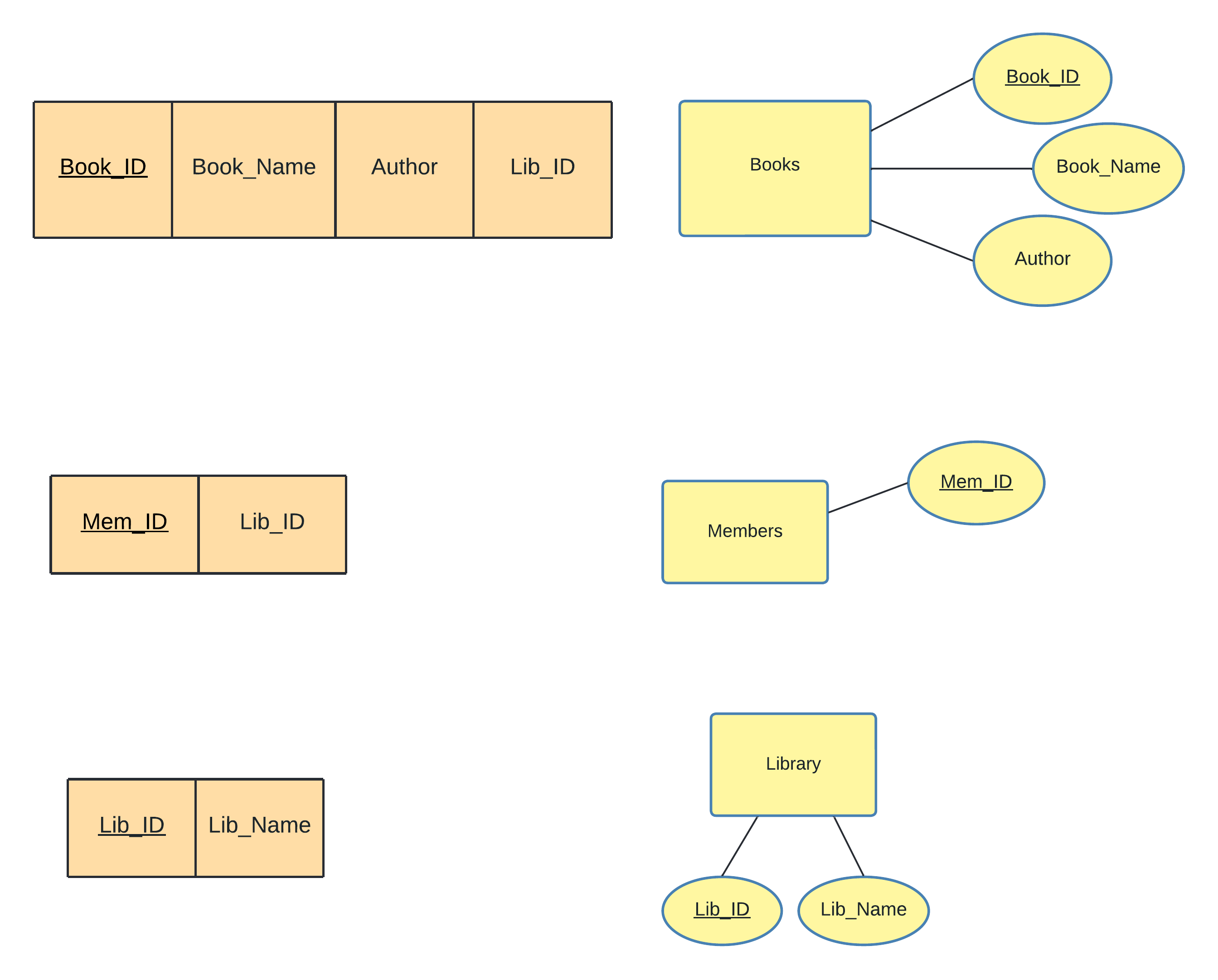
3 ER-to-logical schema mapping

3.1 Mapping of Regular Entity Types





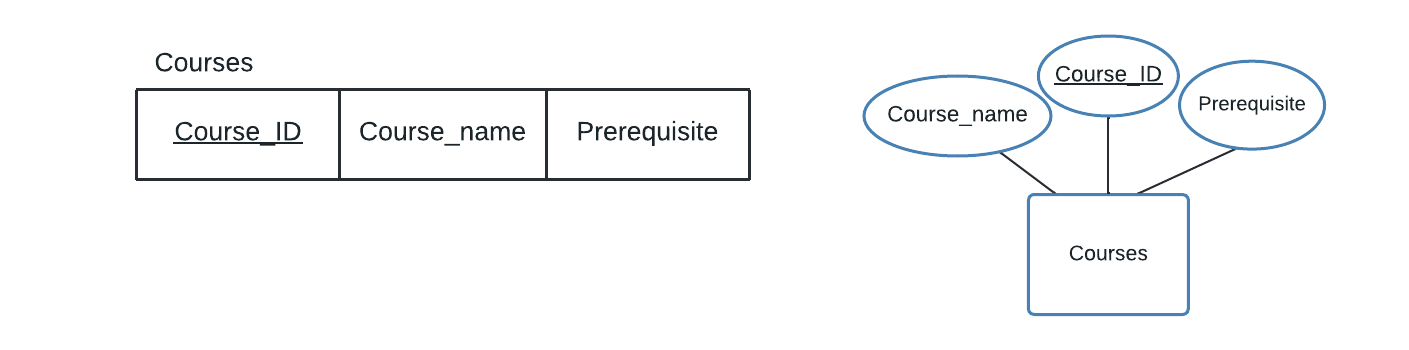
Class

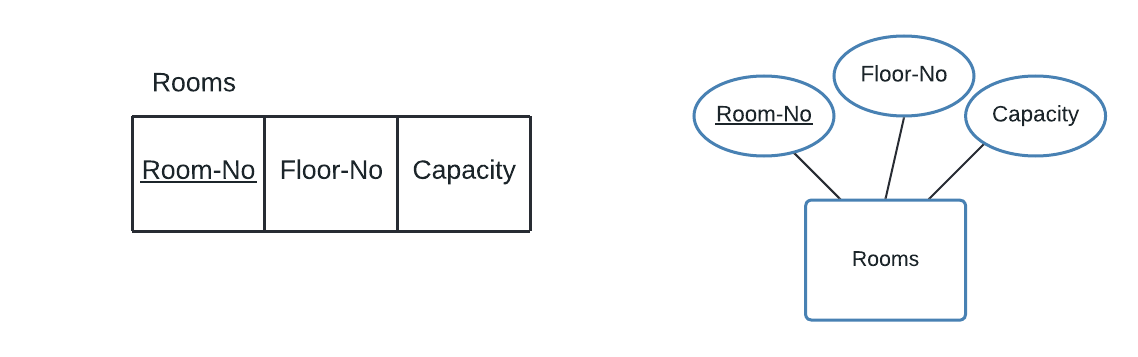


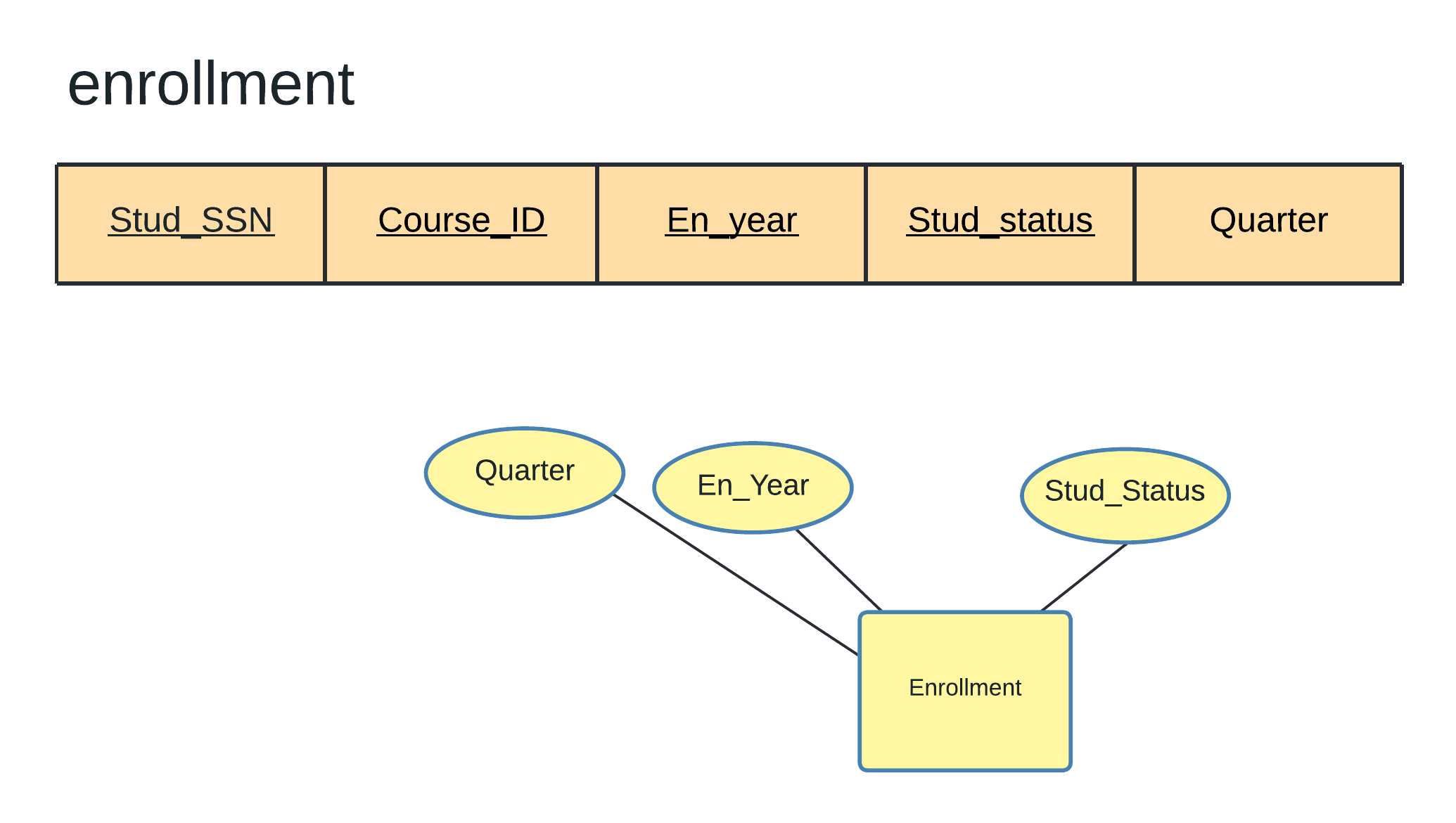
Books

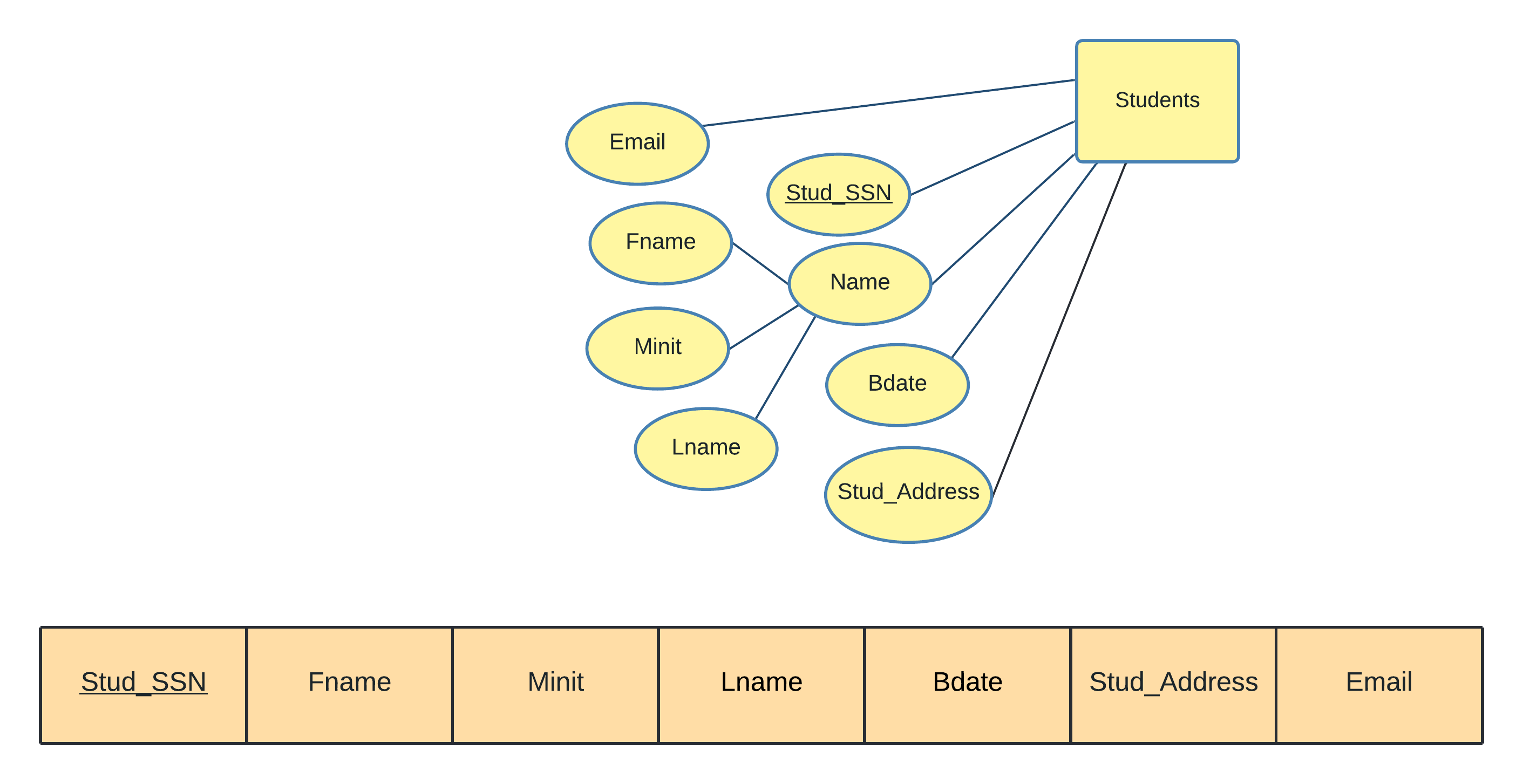
Members

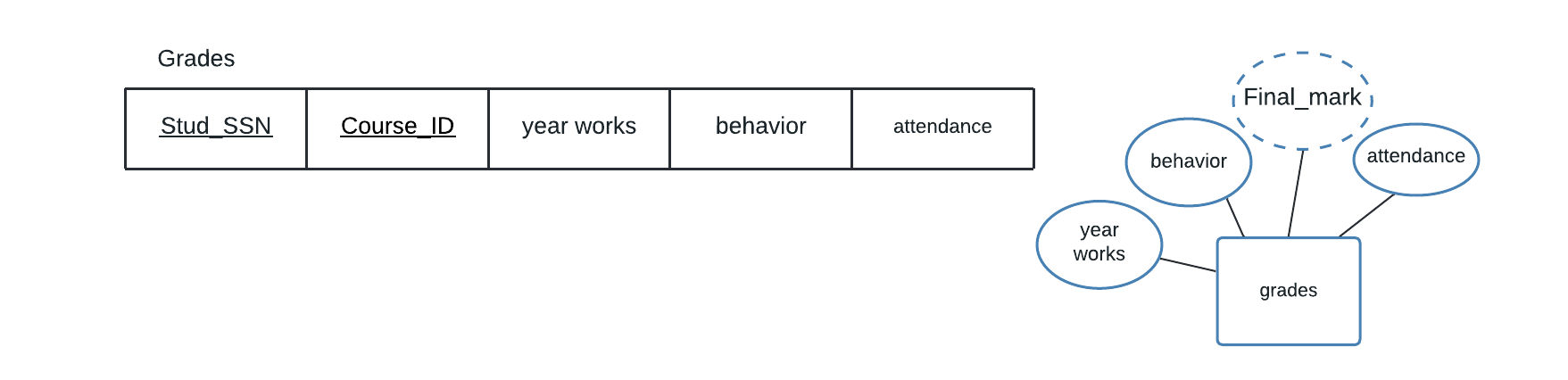
Library

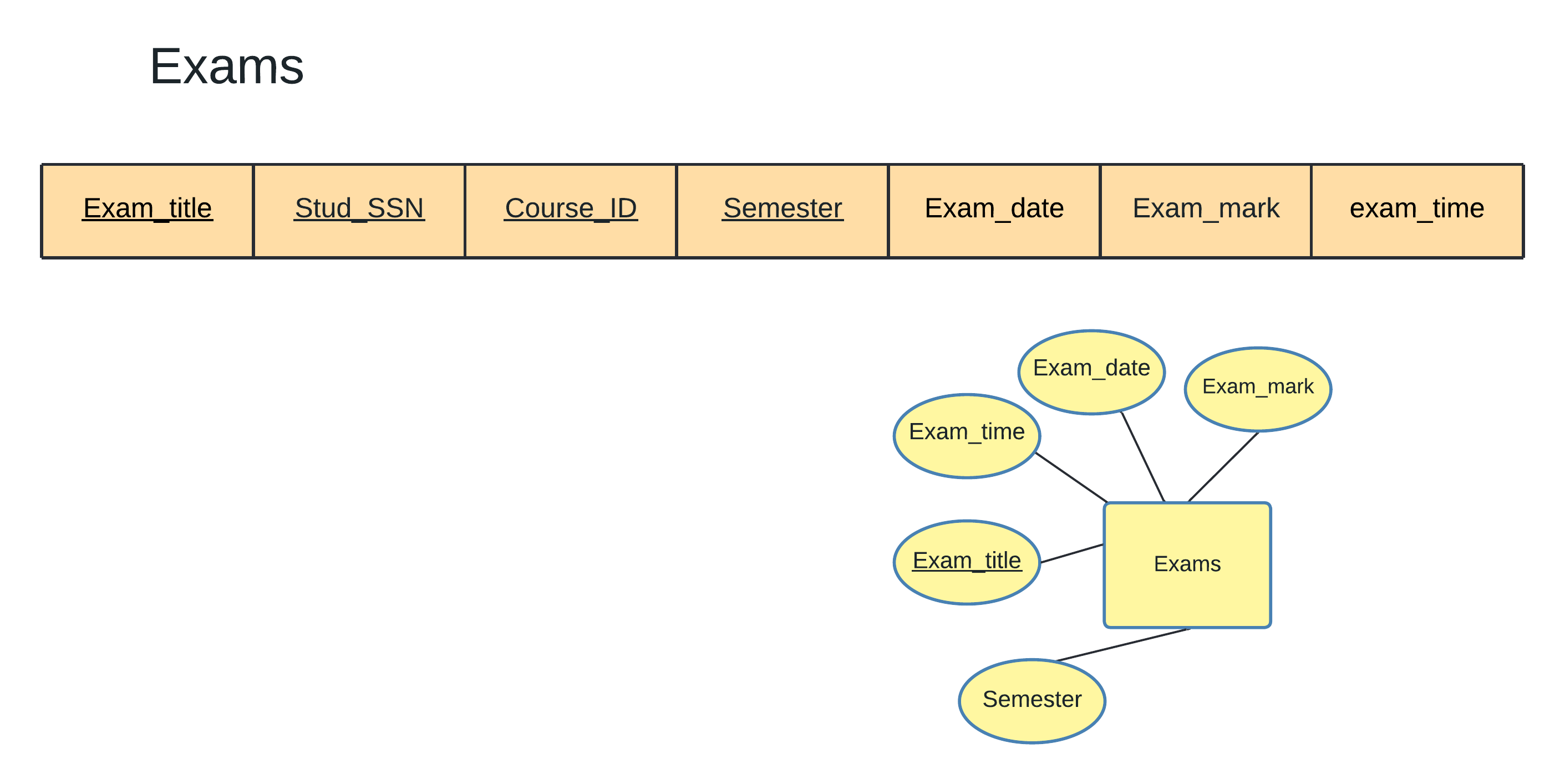




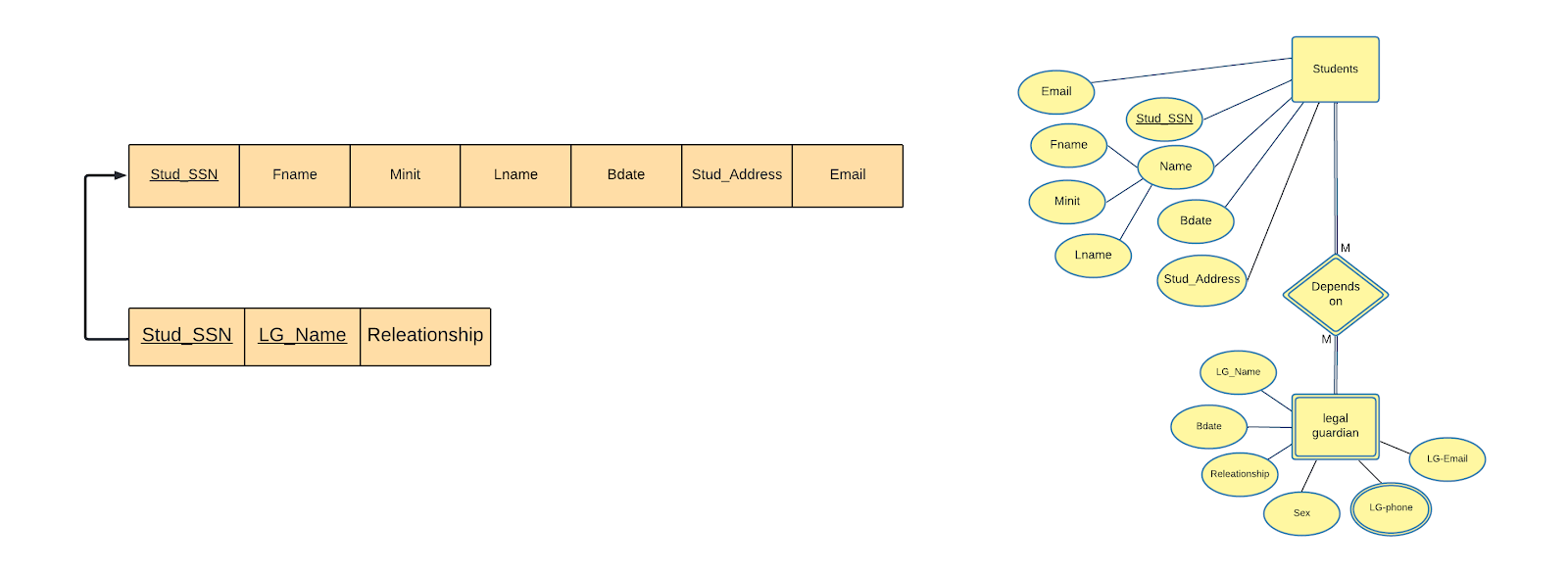








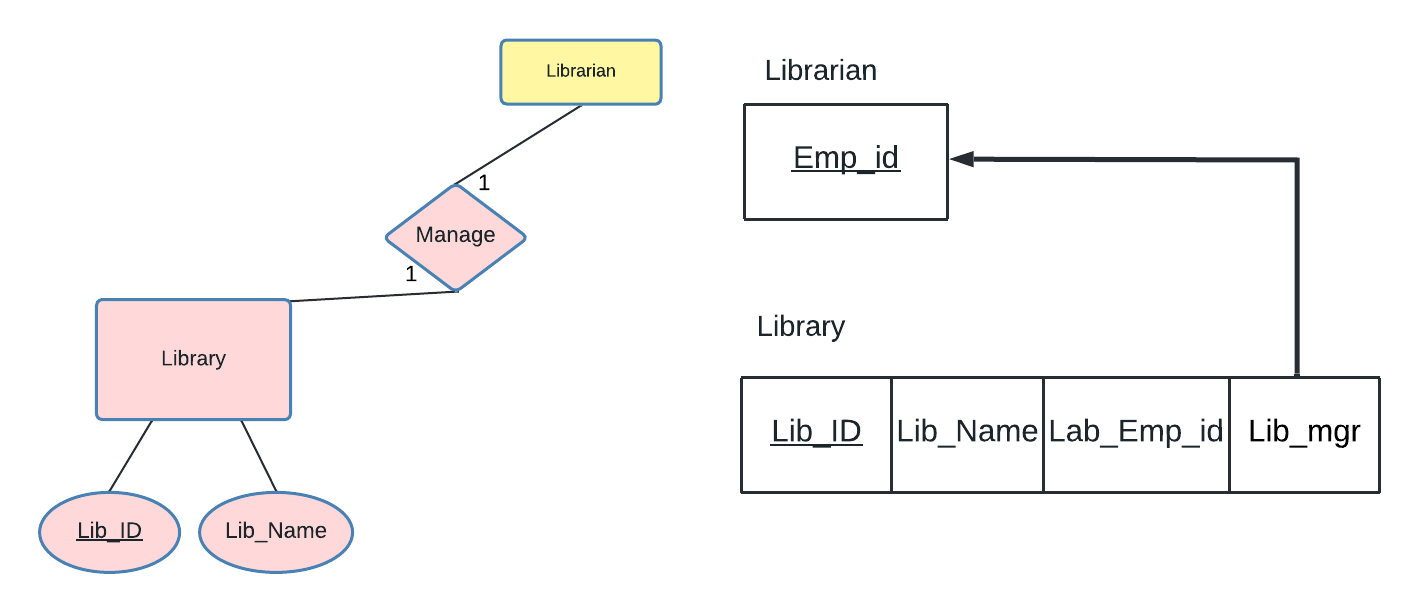
3.2 Mapping of Weak Entity Types

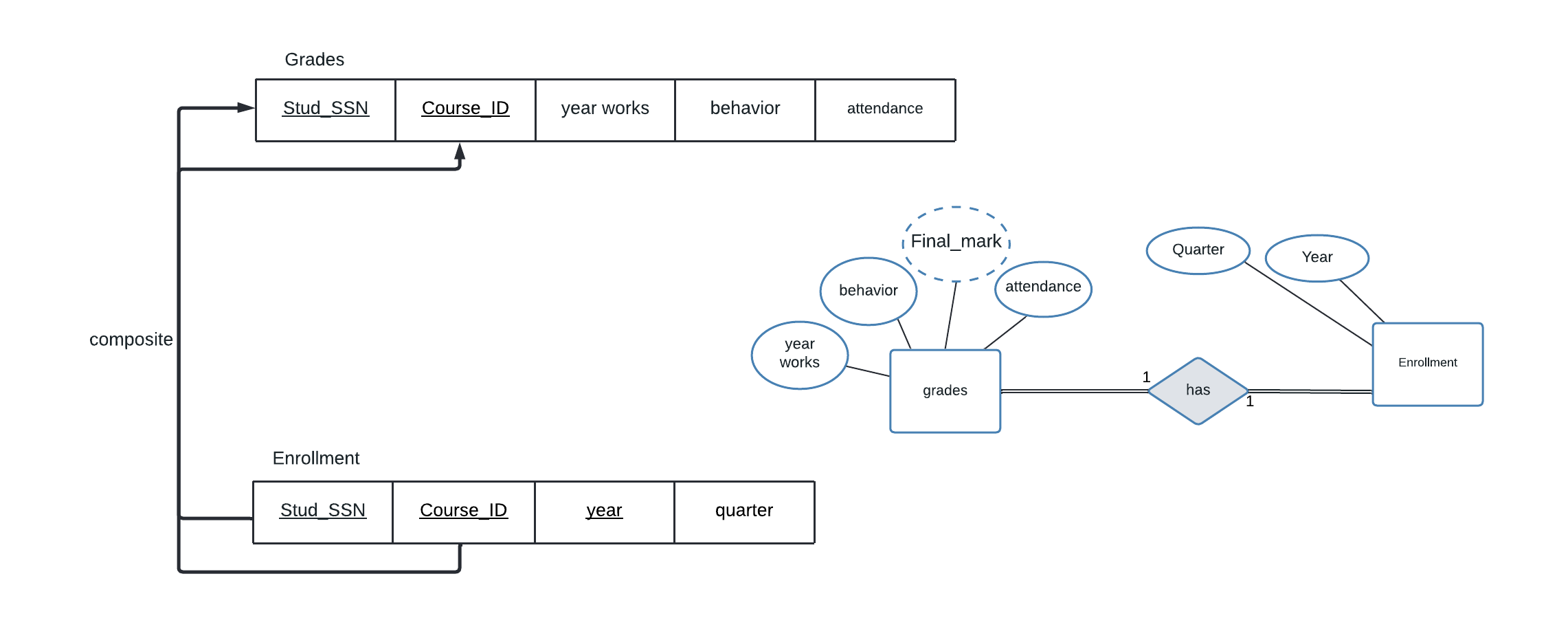


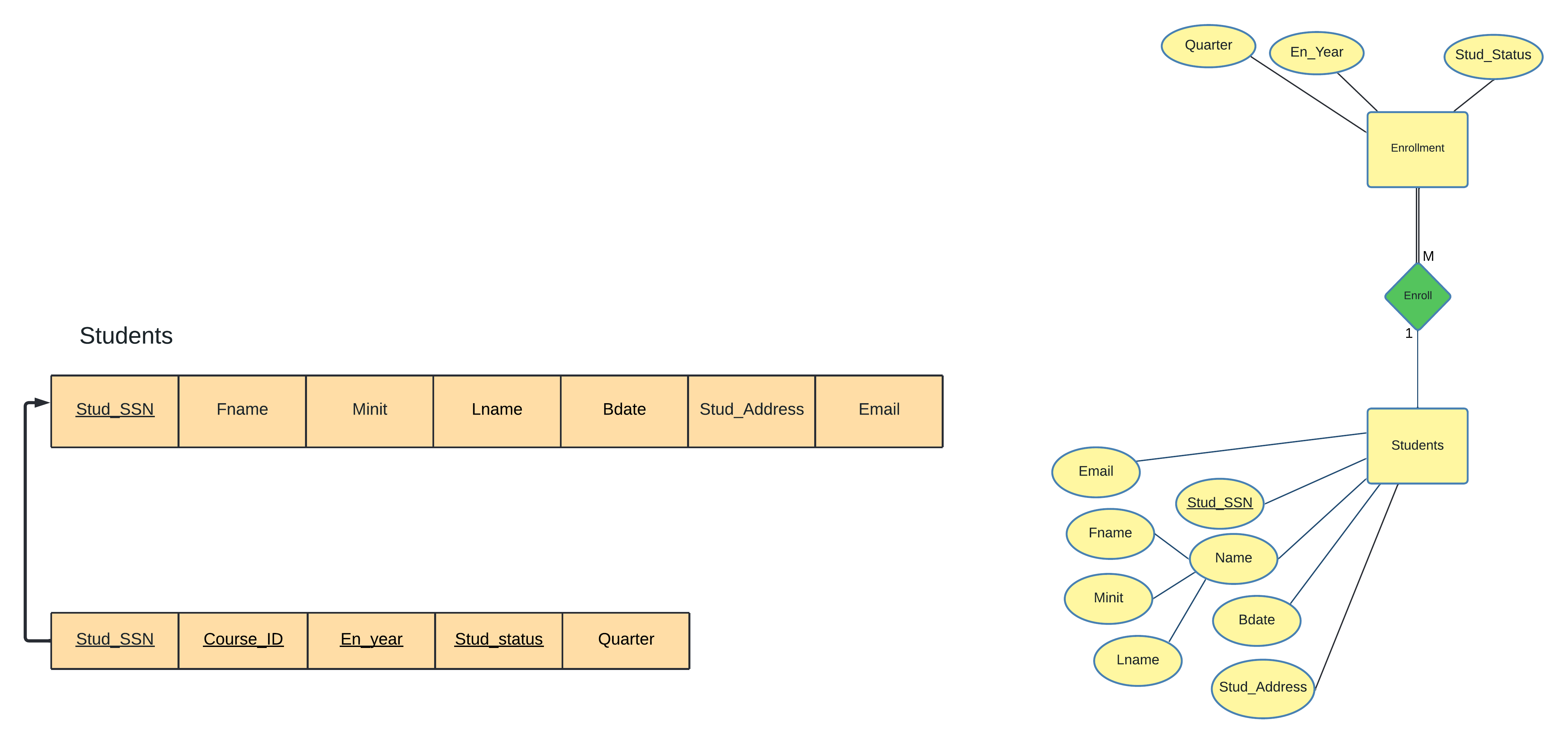
Students

Legal guardian

3.3 Mapping of binary 1-1 relationship types

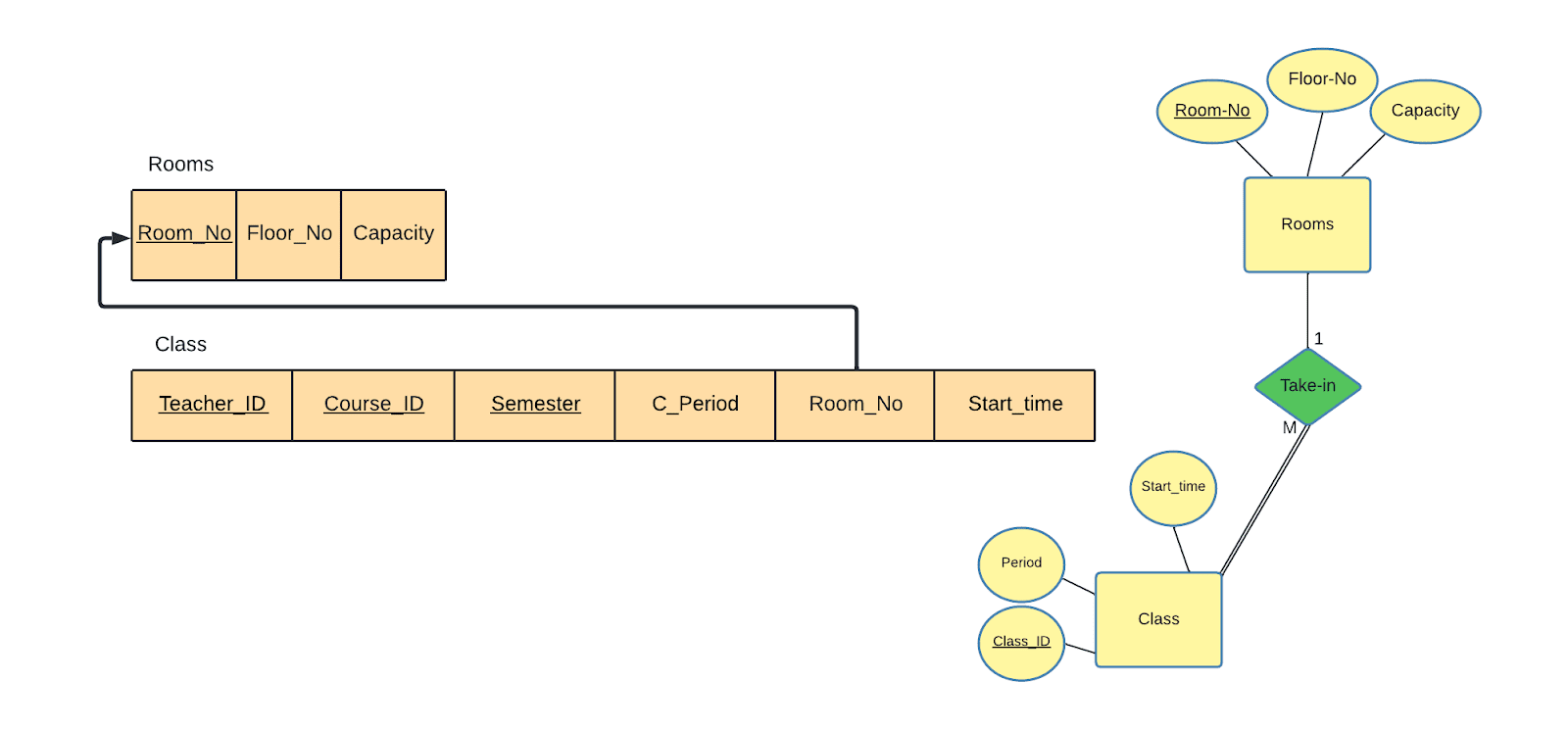


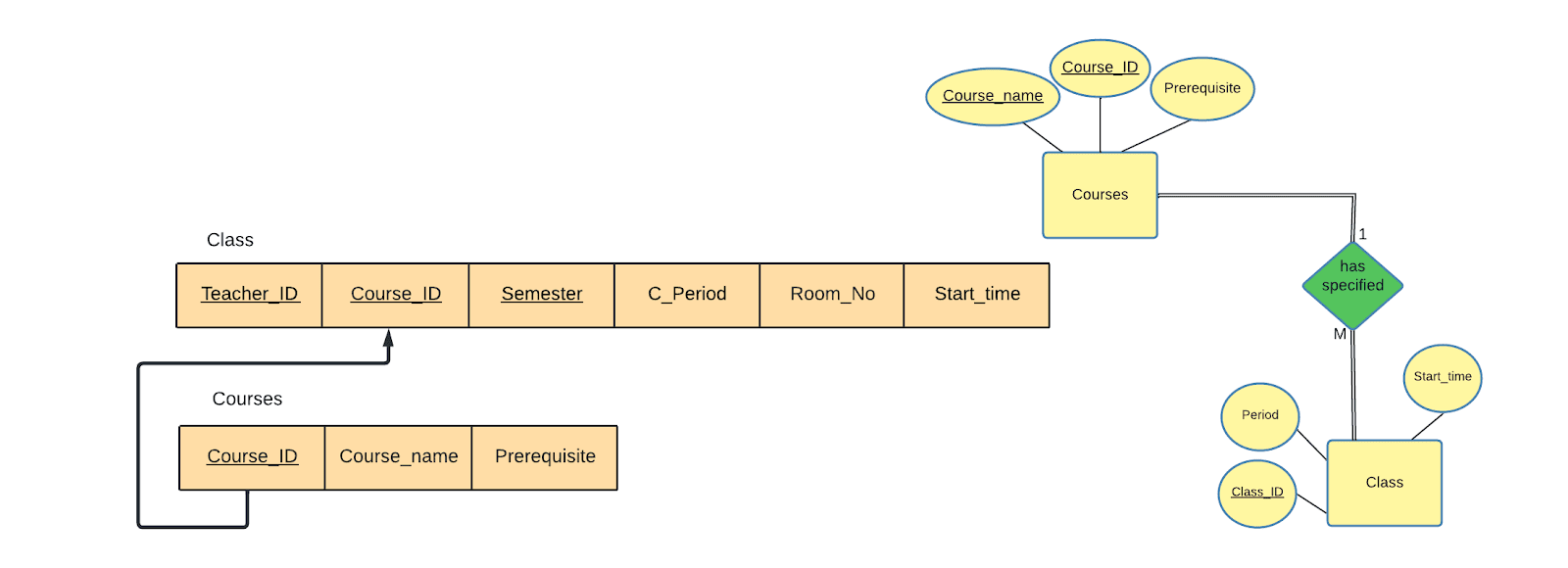


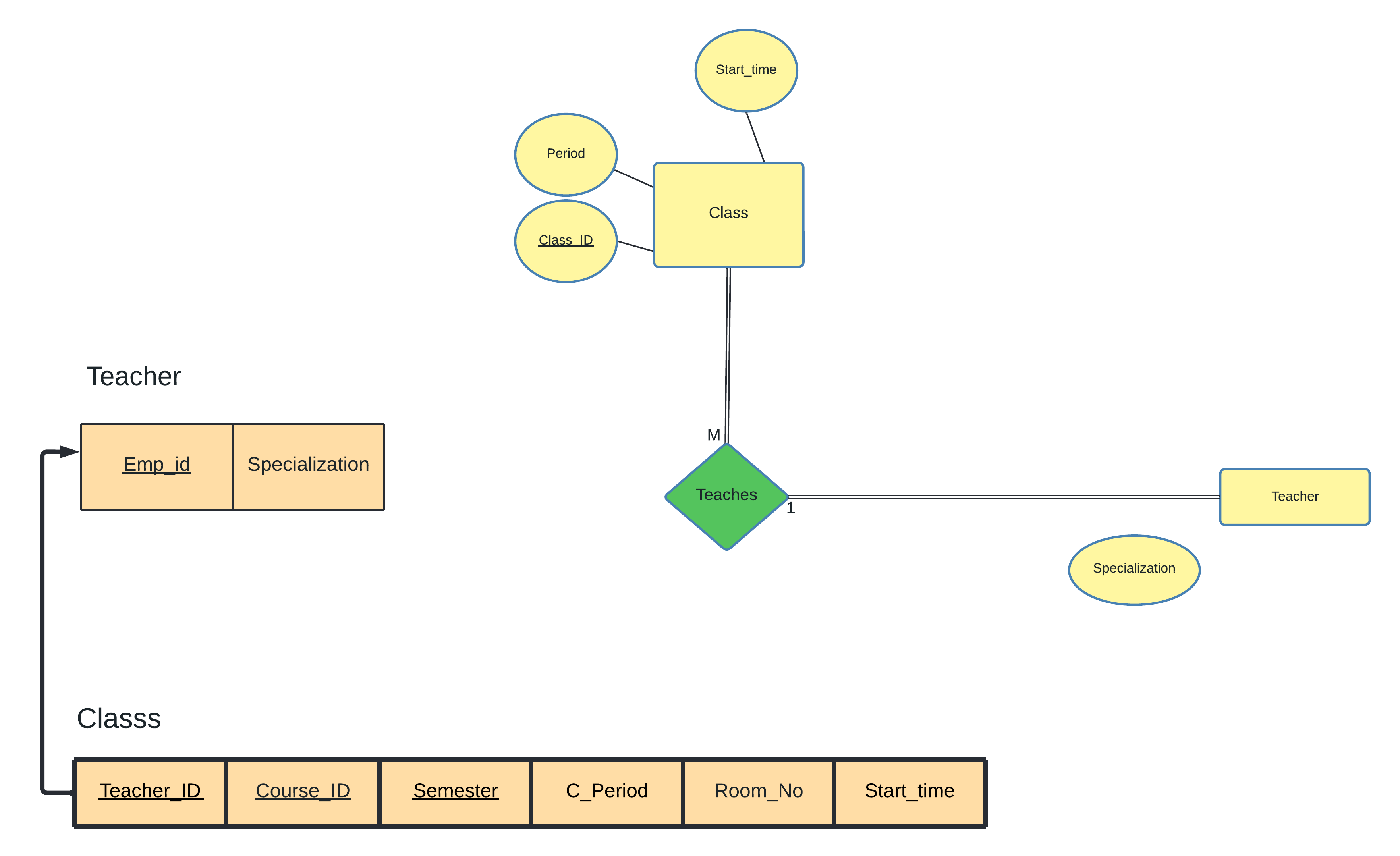


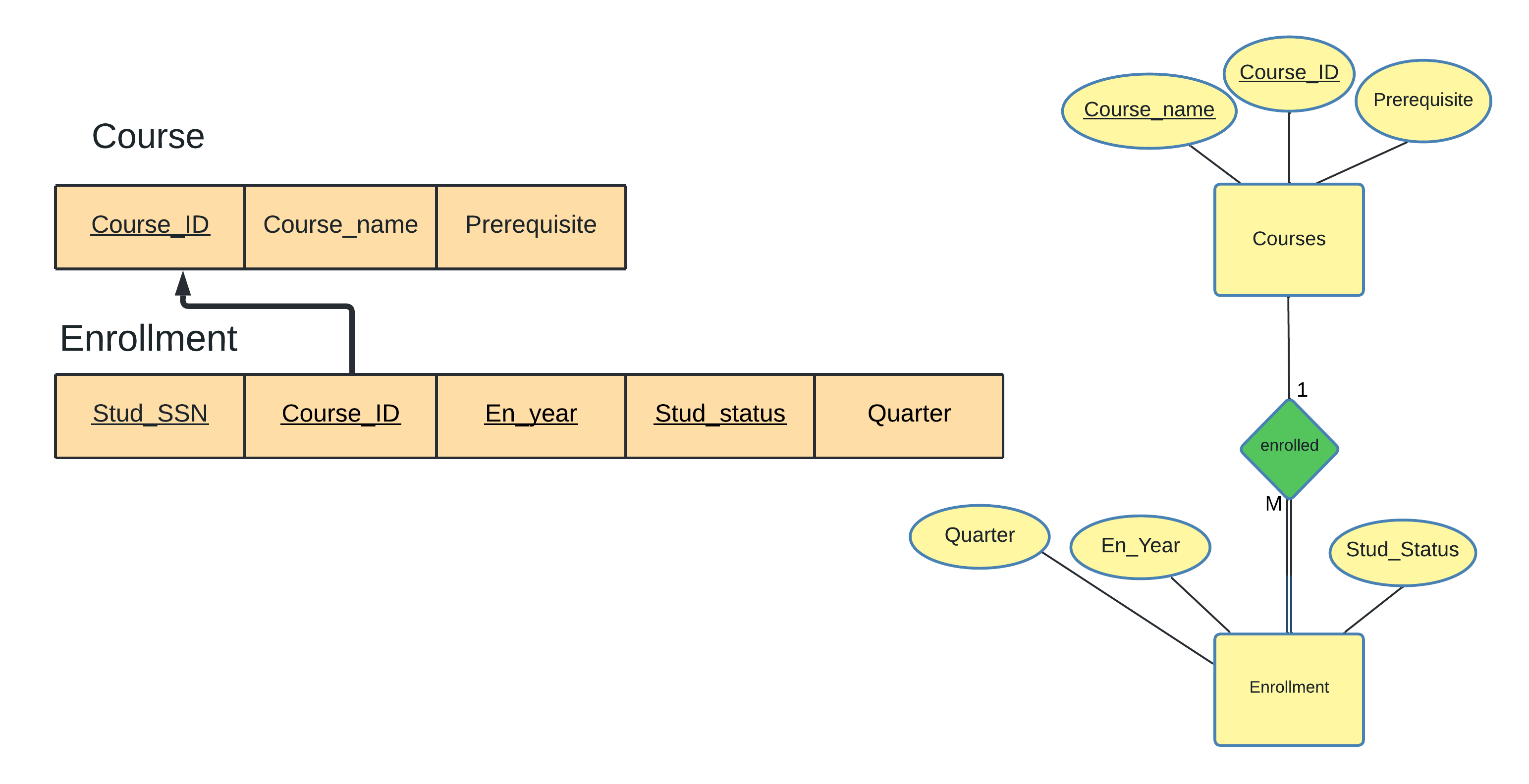
Enrollment

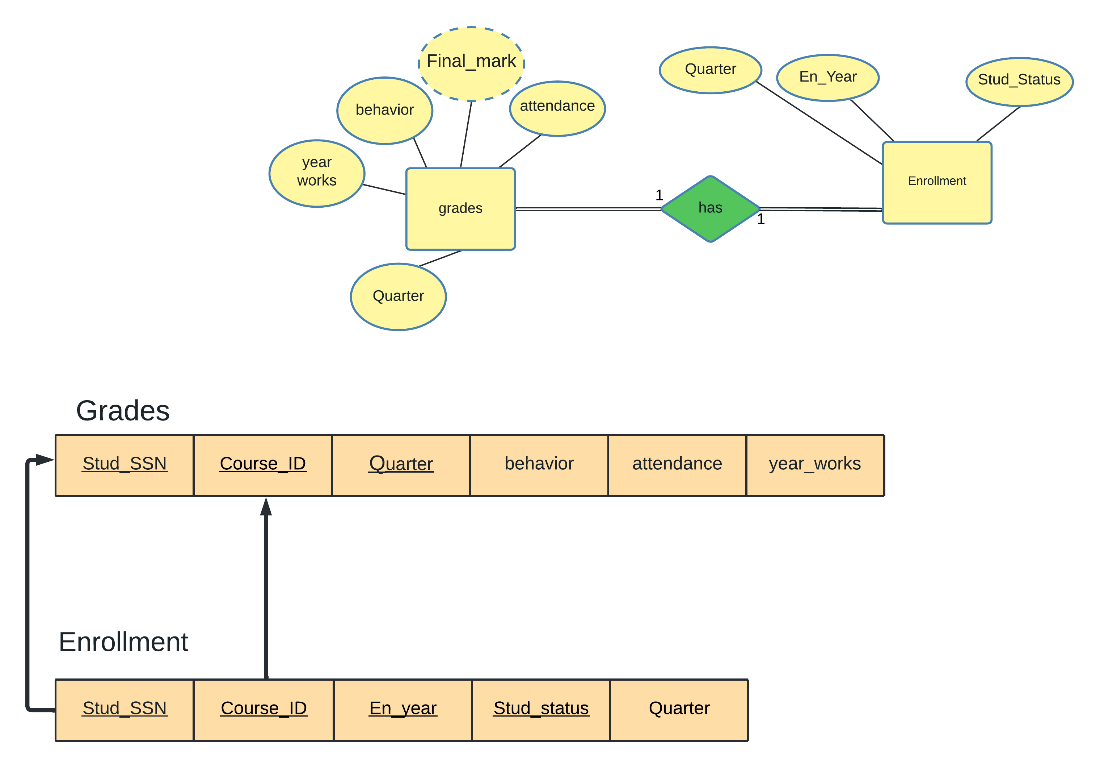
3.4 Mapping of binary 1-N relationship types

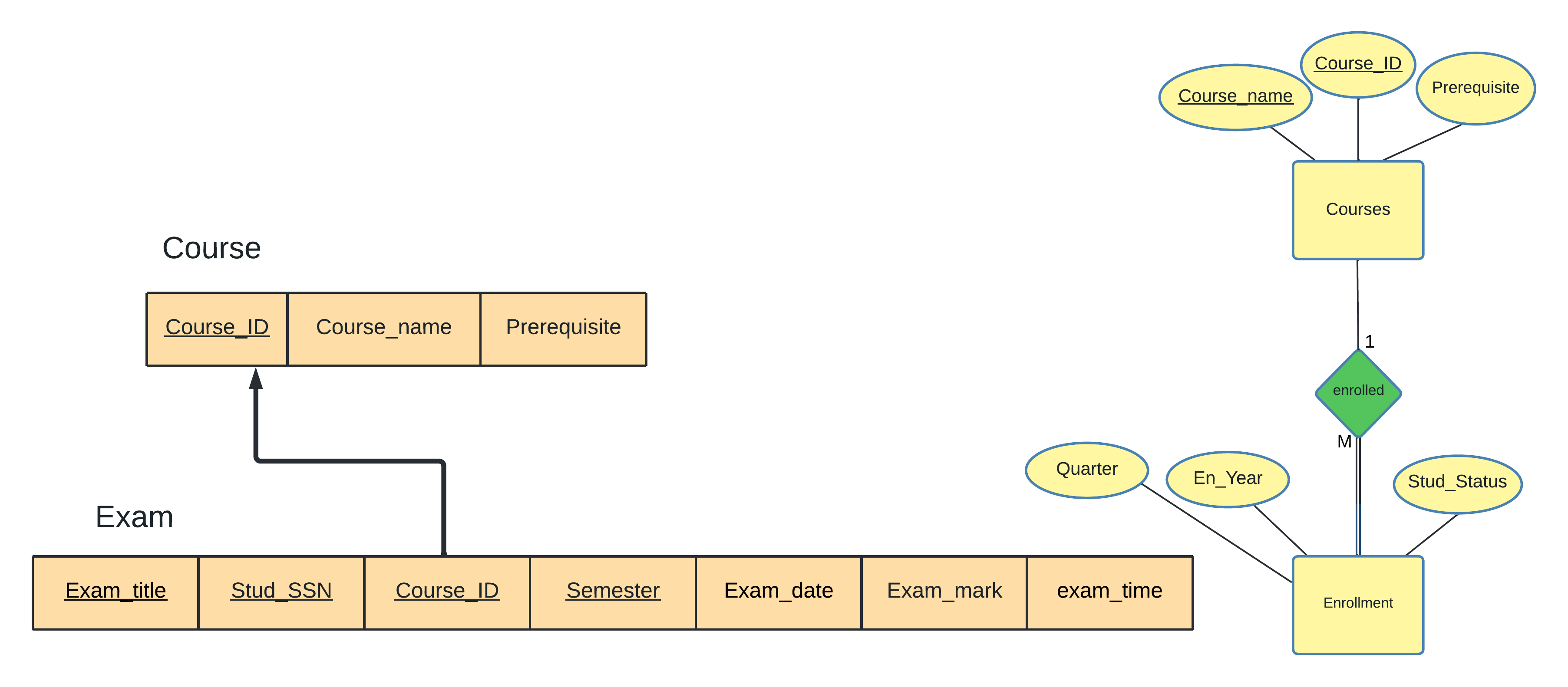
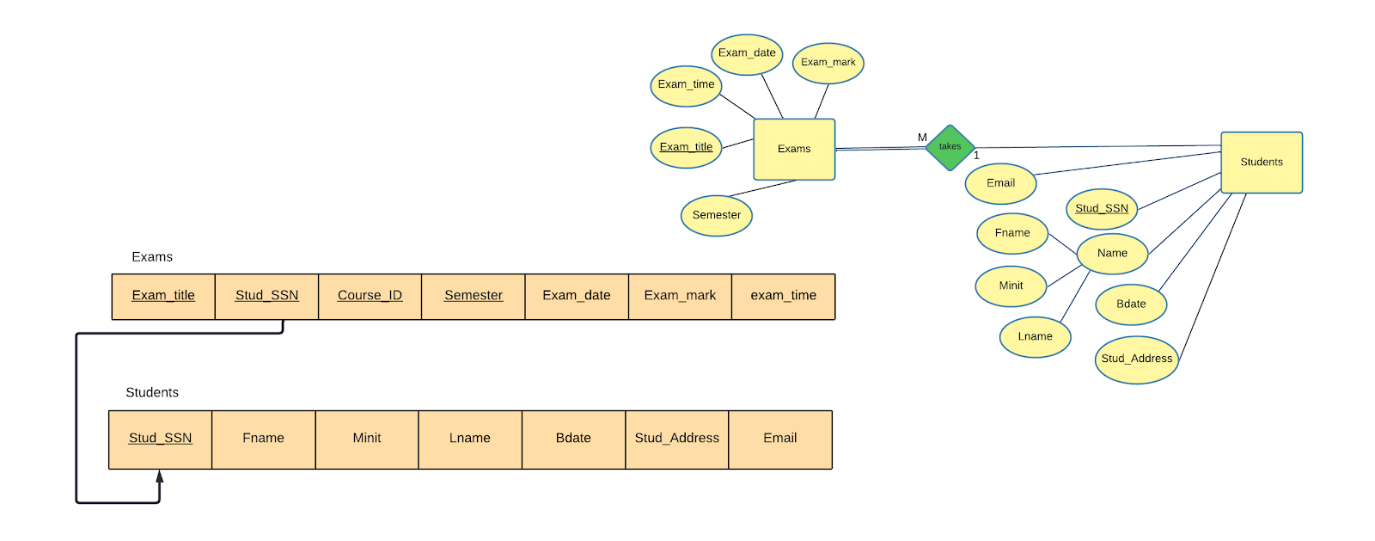






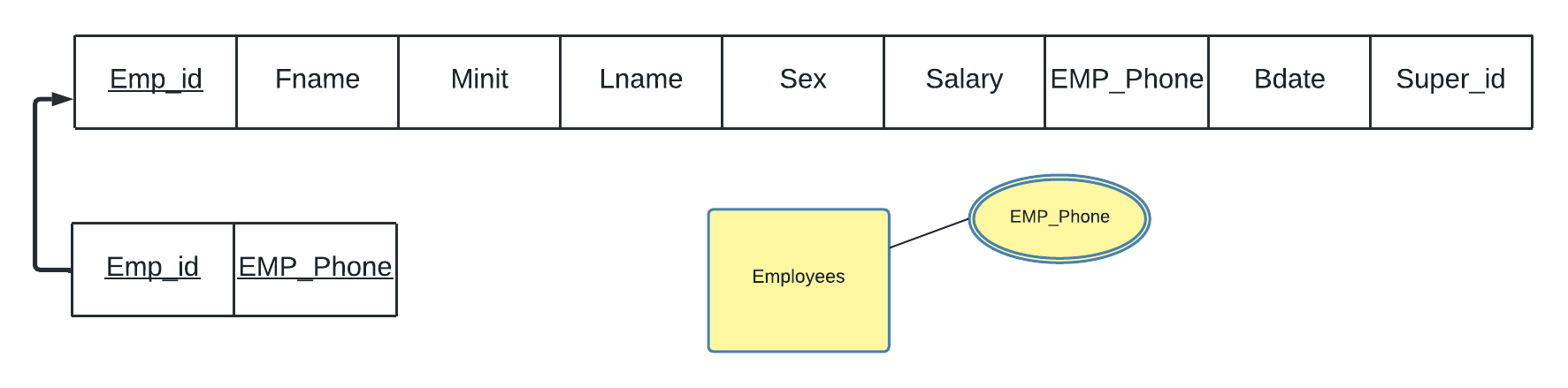


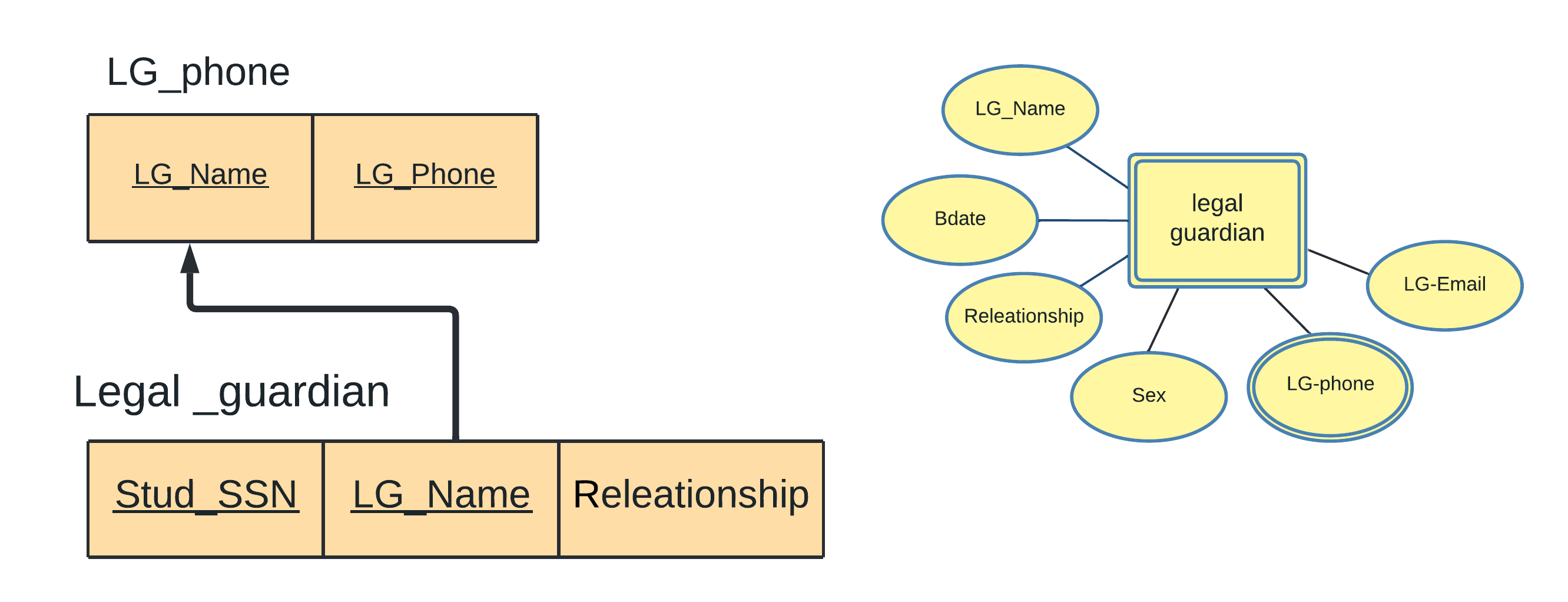




3.5 Mapping of binary M-N relationship types

[NONE]

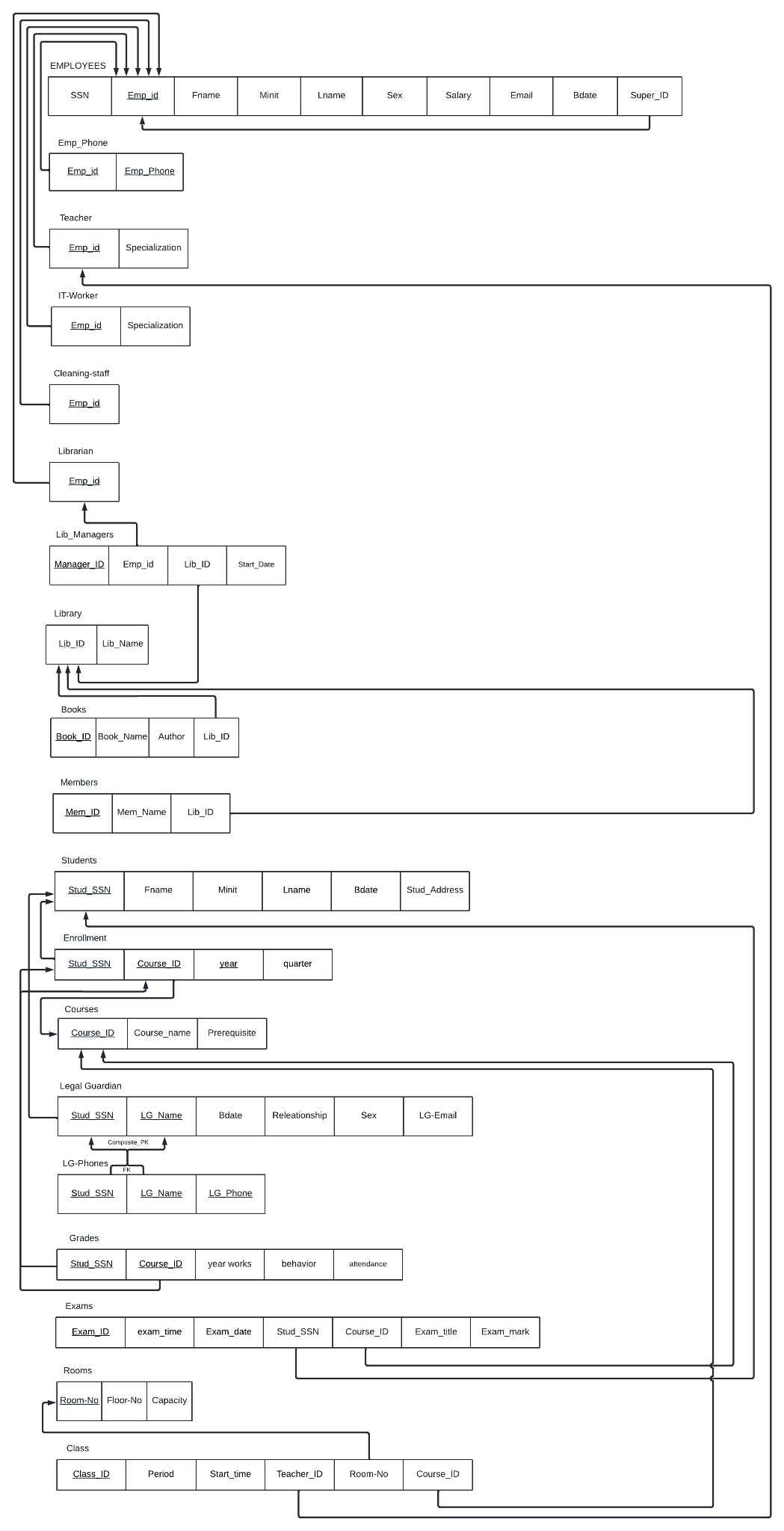
3.6 Mapping of multivalued attributes



3.7 Mapping of n-ary relationship types

[NONE]

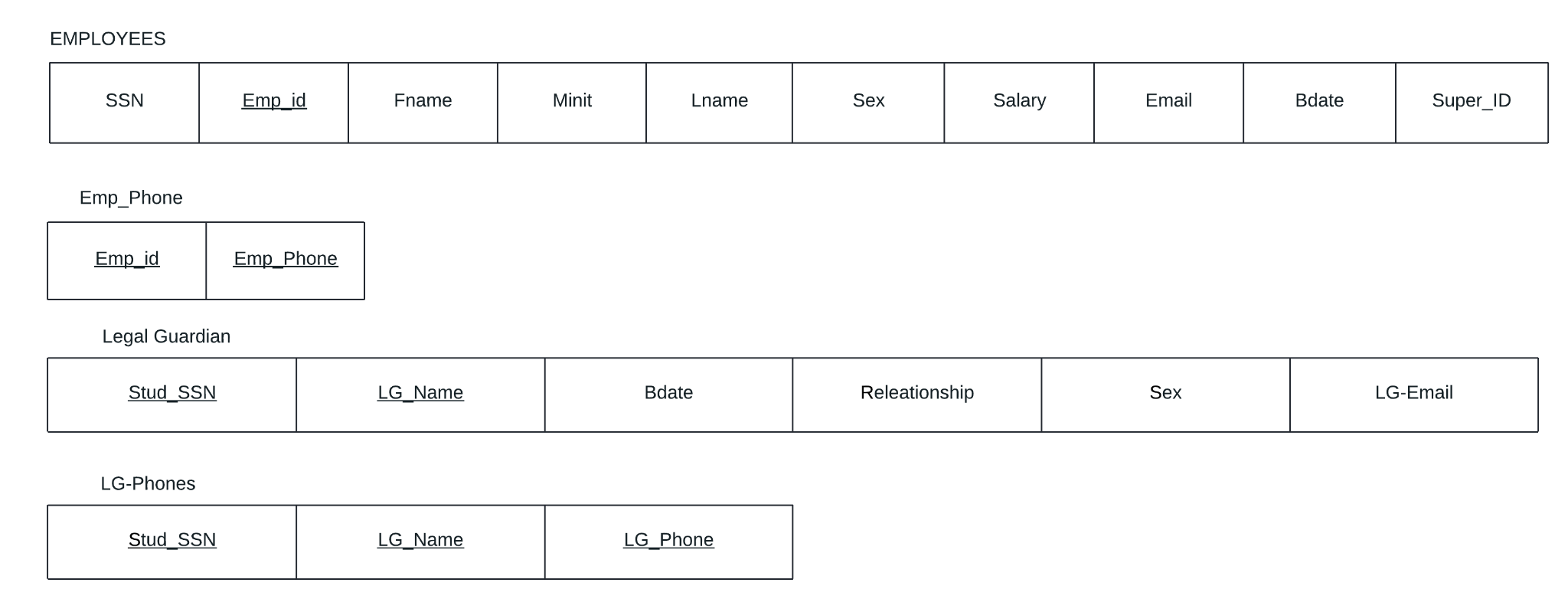
3.8 Schema Diagram

**

4 Normalization

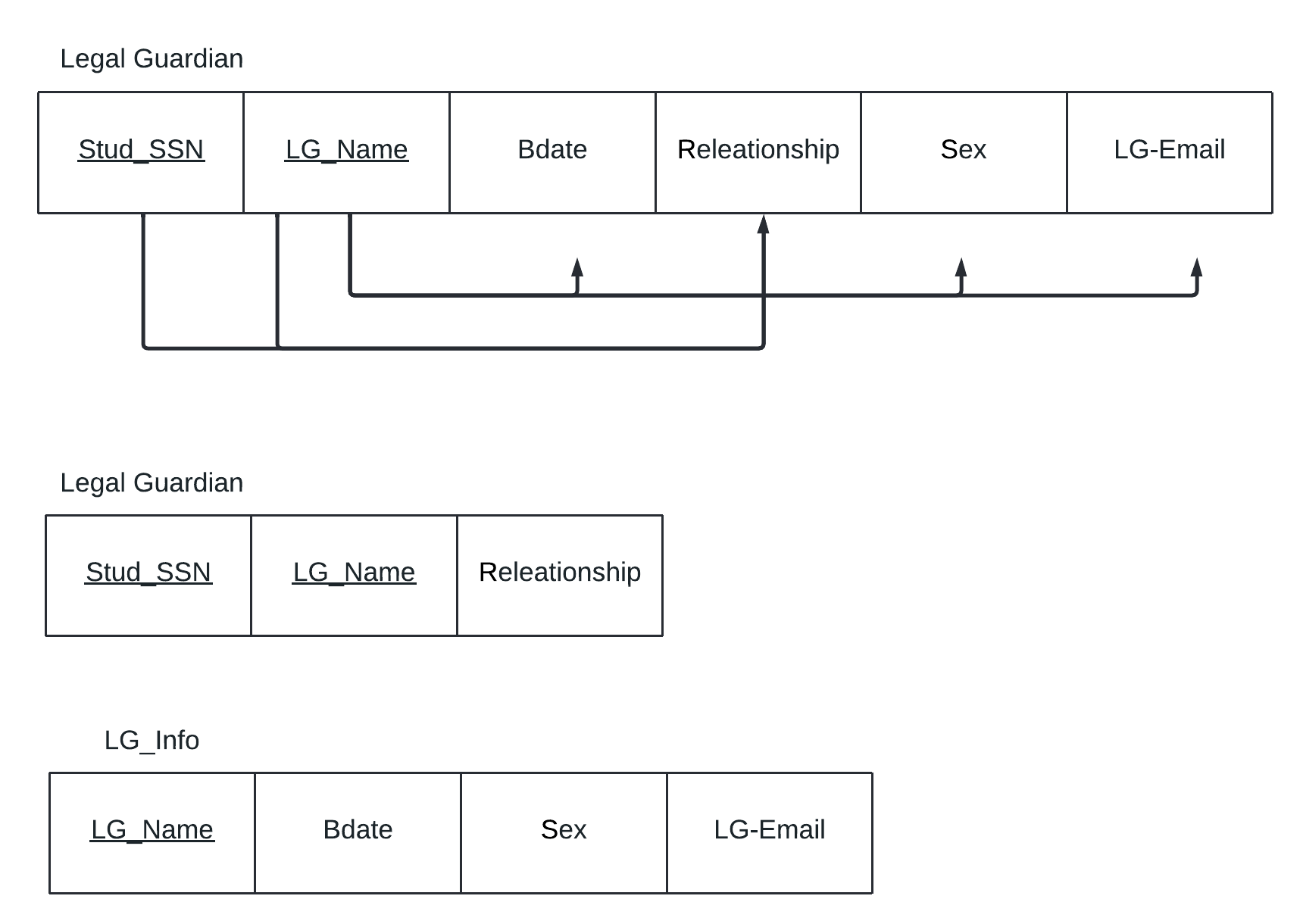
4.1 First Normal Form

First normal form requires not having composite and multivalued attributes. We have two multivalued attributes in the employee entity and in the Legal guardian entity. Since we transformed the mentioned attributes to a relation, we no longer have anything that goes against the guidelines for the first normalization Form.



4.2 Second Normal Form

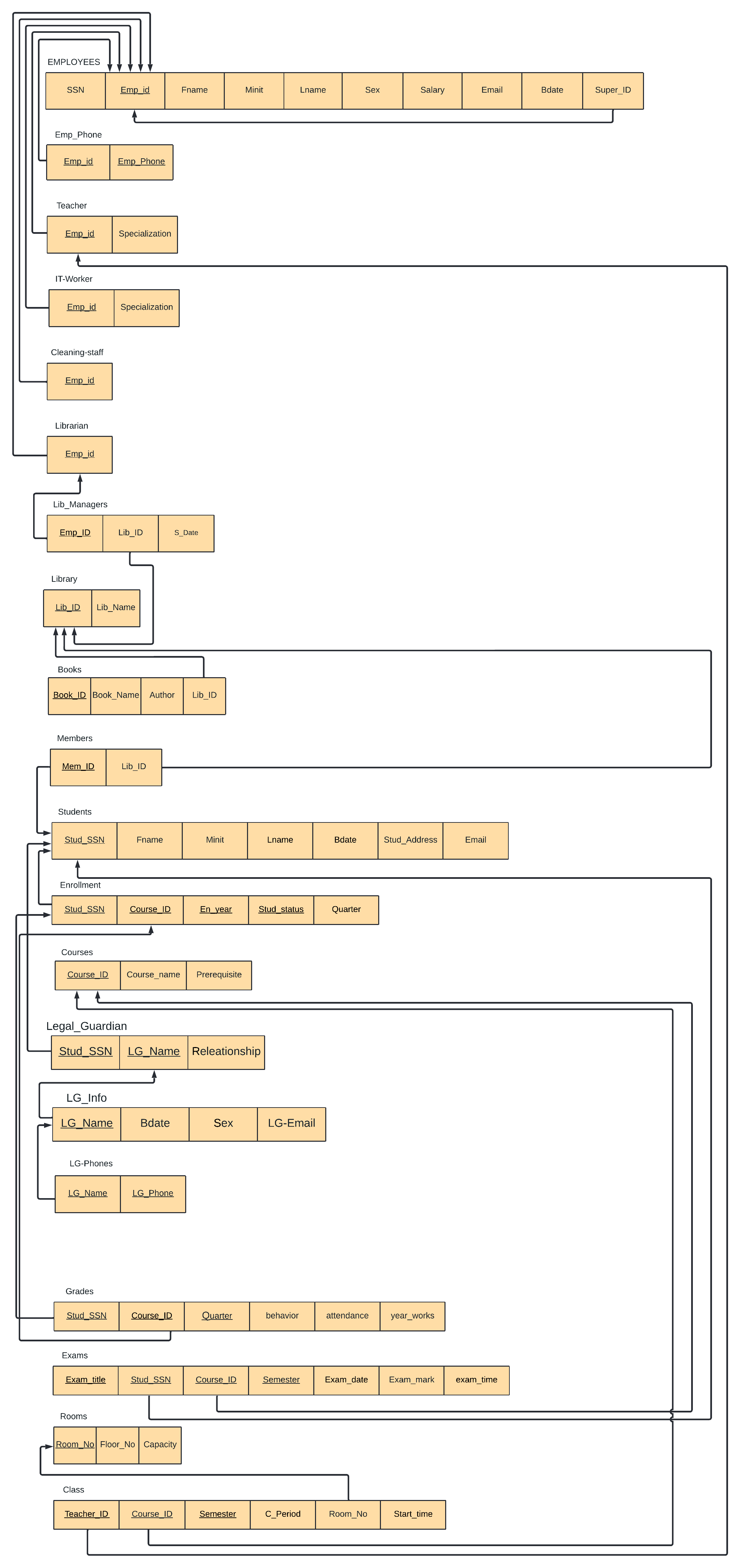
To make our relational schema in Second normal form, every non-prime attribute should be fully functionally dependent on the primary key. All our relations are in the 2NF except Legal Guardian Relation:



4.3 Third Normal Form

[NONE]

5 Final DB Schema Diagram



PART III: IMPLEMENTATION

6 Table Creation Script

*In this section, list your table creation scripts. The tables should all be in 3NF. If you have 10 tables then you should have 10 subsections. Please highlight (in yellow) all the constraints in the script including referential integrity and constraints on attributes. Please be organized!*

6.1 <*Employee*> TABLE



6.2 <*Emp\_Phone*> TABLE



6.3 <*Teacher*> TABLE



6.4 <*IT\_Worker*> TABLE



6.5 <*Cleaning\_Staff*> TABLE



6.6 <*librarian*> TABLE



6.7 <*lib\_managers*> TABLE



6.8 <*Library*> TABLE



6.9 <*Books*> TABLE



6.10 <*Members*> TABLE



6.11 <*Students*> TABLE



6.12 <*courses*> TABLE



6.13 <*Enrollment*> TABLE



6.14 <*Legal\_Guardian*> TABLE



6.15 <*LG\_Info*> TABLE

create table LG\_Info (

    LG\_Name varchar2(15),

    Bdata DATE,

    Sex CHAR,

    LG\_Email VARCHAR2(25),

    CONSTRAINTS LG\_Info\_PK PRIMARY KEY (LG\_Name),

    CONSTRAINTS LG\_Info\_FK FOREIGN KEY(LG\_Name) REFERENCES Legal\_Guardian(LG\_Name) on delete CASCADE

);

6.16 <*LG\_Phones*> TABLE



6.17 <*Grades*> TABLE



6.18 <*Exams*> TABLE

6.19 <*Rooms*> TABLE



6.20 <Class> TABLE



7 Constraints Script

*In this subsection, show how the business rules have been translated into SQL script. Refer to section 2.2.*

|  |  |  |
| --- | --- | --- |
| Business Rule | SQL Script | Table |
| Each employee has a  unique ID | “Emp\_ID NUMBER(10),”  “CONSTRAINTS Employee\_Pk PRIMARY KEY (Emp\_ID)” | Employee |
| Each student can enrol many courses even if he fail the course he can re enrol it next semester | Stud\_SSN NUMBER(10),  Course\_ID NUMBER(10),  En\_year varchar2(15),  Stud\_Status varchar2(15),  Quarter VARCHAR2(15),  CONSTRAINTS Enrollment\_PK PRIMARY KEY (Stud\_SSN , Course\_ID , En\_year , Stud\_Status),  CONSTRAINTS Enrollment\_FK1 FOREIGN KEY(Stud\_SSN) REFERENCES Students(Stud\_SSN),  CONSTRAINTS Enrollment\_FK2 FOREIGN KEY(Course\_ID) REFERENCES courses(Course\_ID) | enrollment |
| 1 student can have only one legal guardian | Stud\_SSN NUMBER(10) UNIQUE , | Legal\_guardian |
|  |  |  |

8 Queries

*In the following subsections, write down five different SQL queries which implements five of the indented output of your system (section 1.4).*

8.1 *<Year\_Works library members>*

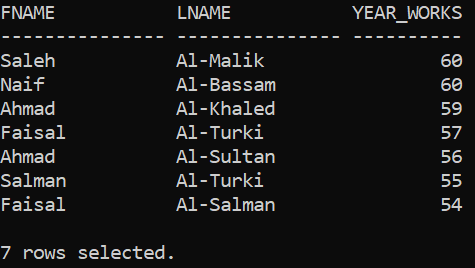
**Query in natural language (ENGLISH)**

Select all library members and order by their grades from high to low

**SQL script**



**Caption of the first five rows of the output**



8.2 *<Covid-19 students>*

**Query in natural language (ENGLISH)**

- List the names of the students who took the same subjects in the year 2020

**SQL script**



**Caption of the first five rows of the output**



8.3 *<Non-Physics 1 teachers>*

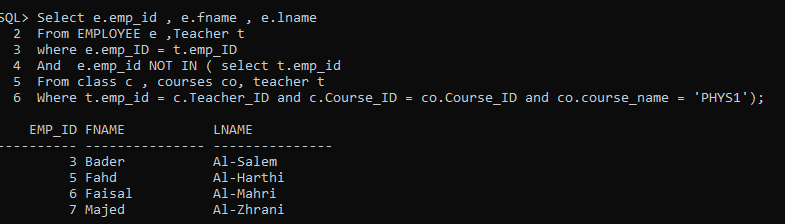
**Query in natural language (ENGLISH)**

Print the Teacher\_ID,first name, last name, of those who do not give a PHYS1 course.

**SQL script**



**Caption of the first five rows of the output**



8.4 *<All failed students in 2021>*

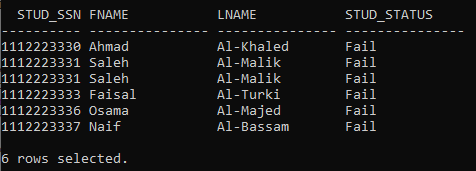
**Query in natural language (ENGLISH)**

Print all the students who fail in 2021

**SQL script**



**Caption of the first five rows of the output**



8.5 *<Fundamental year students>*

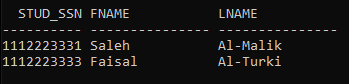
**Query in natural language (ENGLISH)**

- list all students who took MATH1 and in 2021

**SQL script**



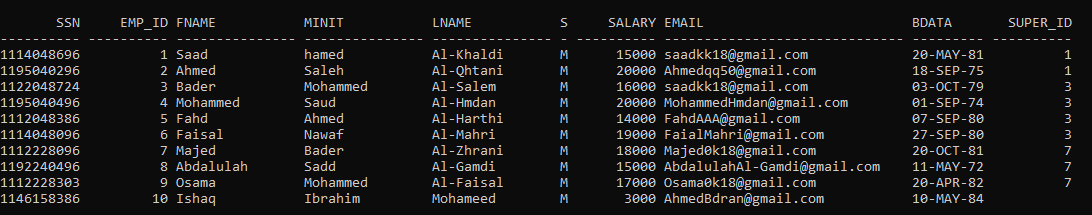
**Caption of the first five rows of the output**



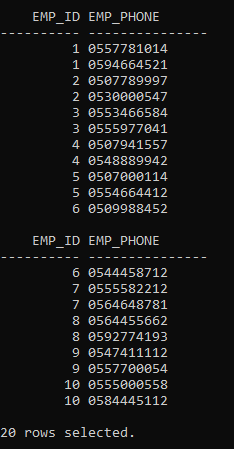
APPENDIX

*For each table, list all the rows (organize!).*

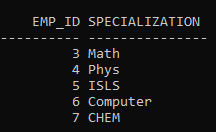
**Employee Table**



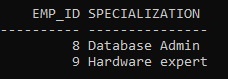
**Emp\_Phone**



**Teacher**



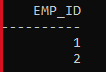
**IT\_Worker**



**Cleaning\_Staff**



**Librarian**



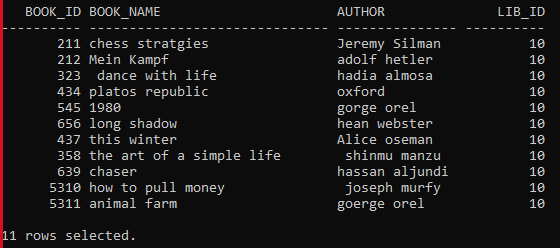
**lib\_managers**



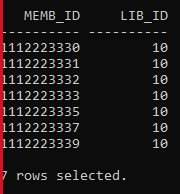
**Library**



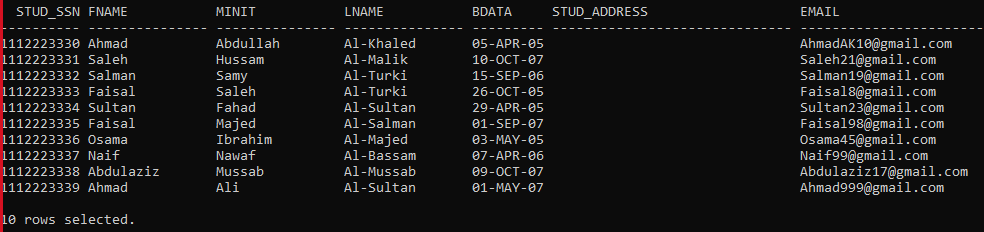
**Books**



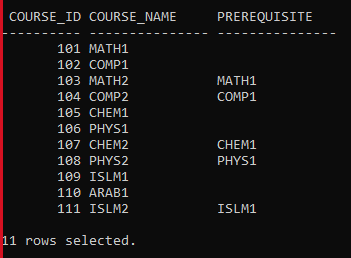
**Members**



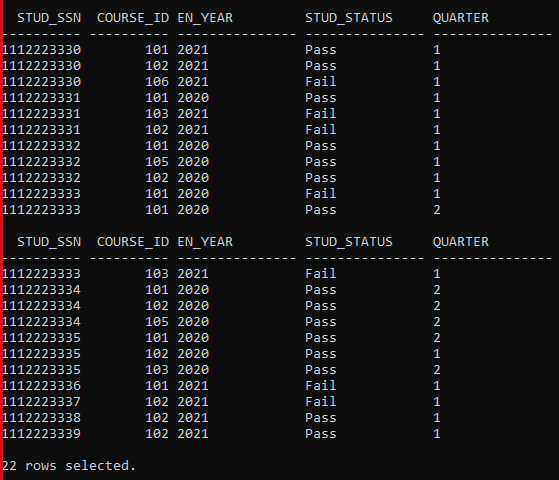
**Students**



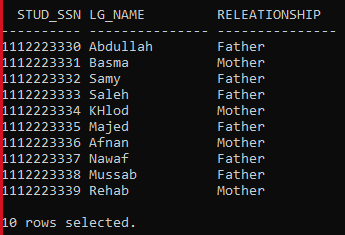
**Courses**



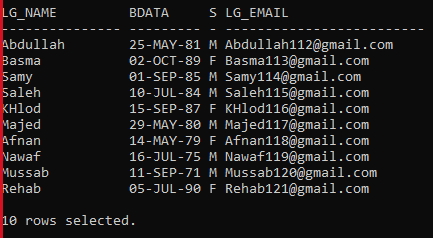
**Enrollment**



**Legal\_Guardian**



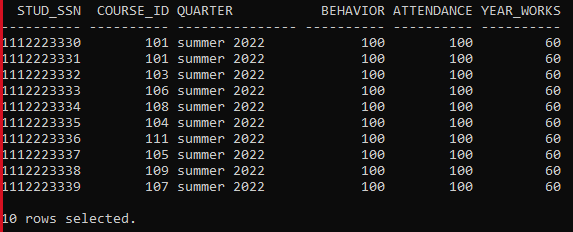
**LG\_Info**



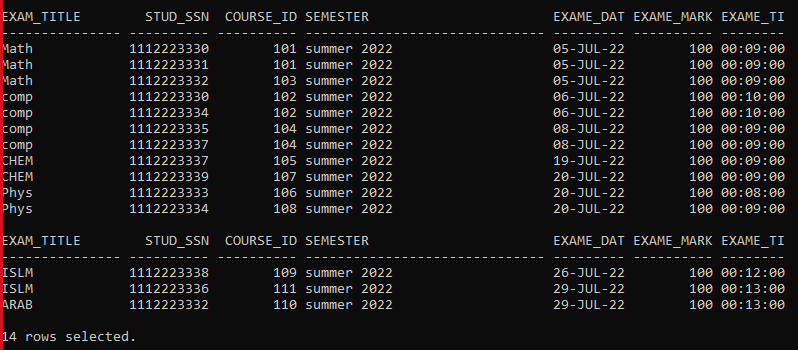
**LG\_Phones**



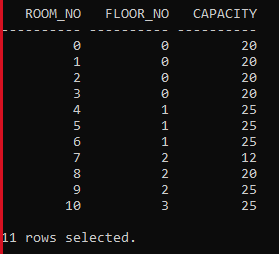
**Grades**



**Exams**



**Rooms**



**Class**

