

A DATABASE DESIGN FOR ARMY SCHOOL

by

Ali Yousef Solh

202004405

Rani Salman

202001002

Mohamad Khalifeh

202000865

Yara Harb

202001058

A REPORT

Submitted to DR. RAMZI HARATY in partial fulfillment of the requirements

for the course Database Management Systems in Computer Science

DB Crew - Phase I - Phase II

27 October 2021

**Table of contents:**

I- Phase2 updates…………..................................................................................................3

II- Introduction…………......................................................................................................3

III- System Description and Constraints..............................................................................4

IV- ER Diagram for ARMY School …...............................................................................6

V- Entity types ....................................................................................................................7

1. Student................................................................................................................ 7

2. Employee............................................................................................................ 8

3. Club.................................................................................................................... 9

4. Departmen….......................................................................................................10

5. Room.................................................................................................................. 11

6. Utility.................................................................................................................. 12

7. Bus.......................................................................................................................12

8. Sport Team...........................................................................................................13

9. Guardian................................................................................................................13

10. Course.................................................................................................................14

11 Event....................................................................................................................14

VI-Relationships....................................................................................................................15

VII- ER to Relational Mapping Algorithm............................................................................22

Step1: Mapping of Regular Entity types .................................................................22

Step2: Mapping of Weak Entity types ....................................................................24

Step3: Mapping of 1:1 binary relationship types.....................................................25

Step4: Mapping of 1:M binary relationship types...................................................26

Step5: Mapping of M:N binary relationship types..................................................27

Step6: Mapping of Multivalued attributes...............................................................29

Step7: Mapping of N-ary relationships types..........................................................29

FINAL STEP: Final display..................................................................................................30

VIII- Conclusion ...................................................................................................................32

IX- Phase 2 Report Checklist...............................................................................................33

X- Instructor’s feedback and evaluation................................................................................34

**I- Phase II Updates**

* Added relational schemas to the report.
* Improved ER diagram (changes in entity types, formatting).
* Improved language.
* Fixed grammar mistakes.
* Improved table of content’s structure and content.
* Added page numbers.

**II-Introduction:**

“DB Crew” is a company created by Yara Harb, Mohamad Khalifeh, Rani Salman, and Ali Youssef Solh. This company offers a designed database for a school and aims to satisfy the user’s preferences and requirements. Thus, an ER diagram is needed to build this database and work diligently on every detail required. To construct the ER diagram included in the report, we referred to “Draw.io” which is a platform similar to Microsoft Visio.

This report will tackle and summarize the work done from attributes, entity types, and relationships. Every entity type with its attributes and relationships is then discussed individually to ensure its clarity.

We have the pleasure to take part in this project as we will not only learn different skills in the process but also help build a school that emerges a well-educated society.

**III- System Description and Constraints:**

Ever since the parents know they are having a baby, a sense of worried emotions comes rushing to them. Every couple would be concerned about putting their child in a decent and respectful school. For this, ARMY School is built to provide the appropriate environment for the children while making them feel at home.

ARMY is not designed to teach students only book-related information; instead, it is intended to teach them life skills, preparing them for life. For instance, ARMY teaches the students speaking and leadership skills through clubs and programs that are provided from other schools or universities (e.g. MAL and MUN in LAU where the students should represent a certain country and suggest the possible solutions to the given problem through a speech and form afterward a group to lead in case of crisis).

ARMY school has different campuses, one in Beirut, one in Dbayeh, and one in Baalbek. Each campus has its students, instructors, employees; however, they are all connected.

Once the child registers in one of the campuses, her name, address, phone number, email, ID, etc. are added to the system. In addition, all information related to the student’s guardian is collected. Afterward, the student will proceed by registering in courses according to their chosen path.

A specific instructor and a room are assigned for each course. Other employees may supervise these employees. The employee when hired, as the student, has a specific ID, phone number, email, name, etc.

The students then have the chance to join a club (e.g. Robotics Club, Chess Club, Drama Club, Dance Club, etc…) and sports teams. These clubs and sports teams are supervised and guided by the employee.

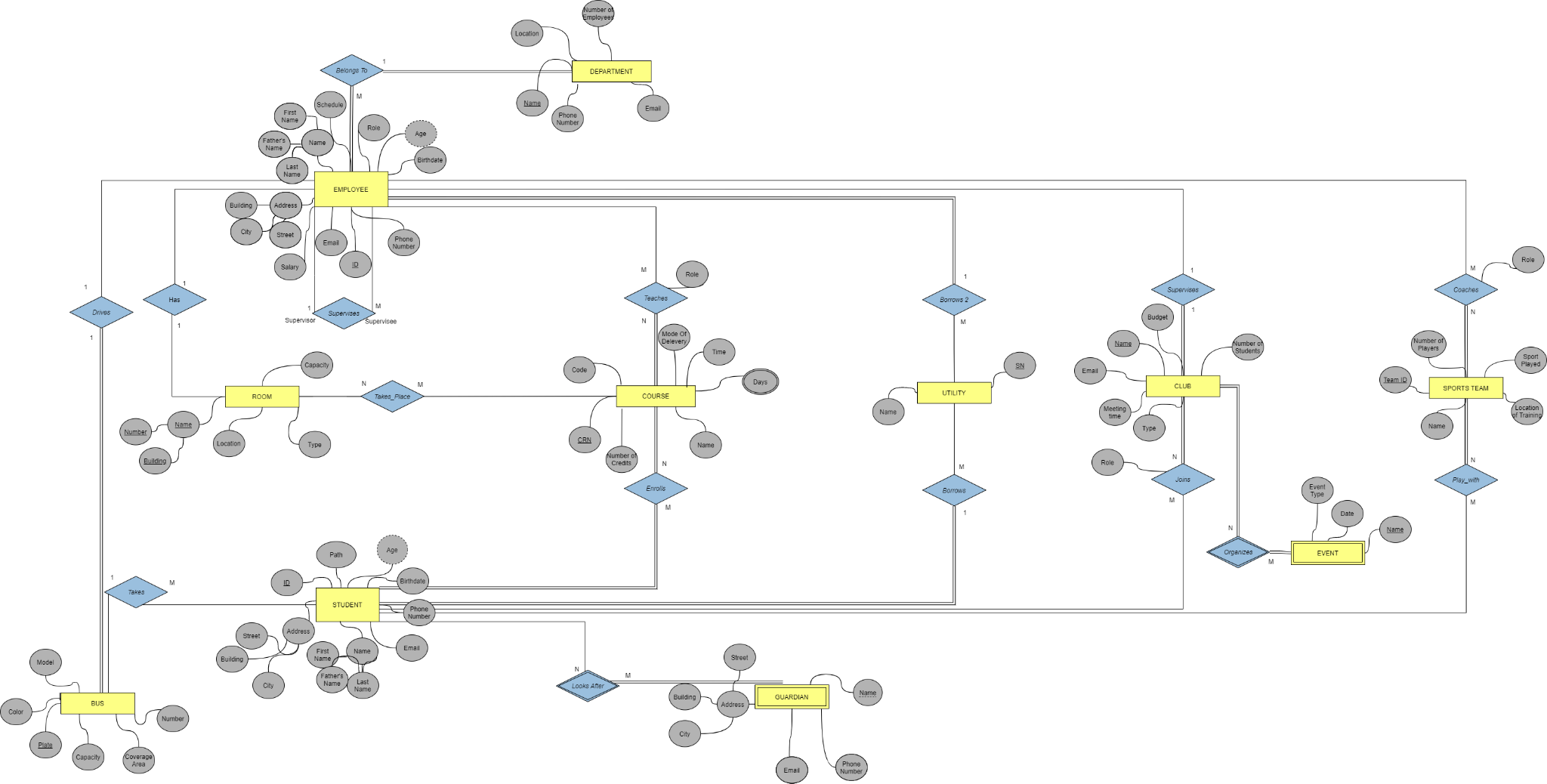
The students’ privileges may also include the usages of facilities (gym, swimming pool, library, cafeteria, etc…) and utilities (computers, books, projectors, boards, etc…).

Within one campus, the school is divided into departments, each having its employees that work in. These departments may include the Financial Department, Special Education Department, etc…

As the day ends, when the students want to go home, they have the option to go by bus that is provided by the ARMY school.

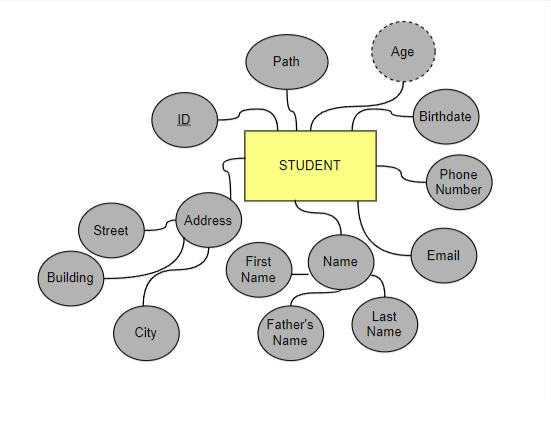
The instructors at ARMY school put all their efforts to provide a decent education for the students.

**IV- ER Diagram:**



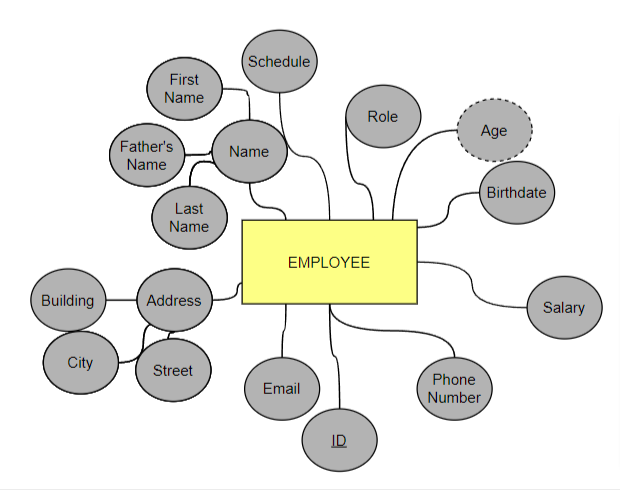
**IV- Entity Types:**

**1) STUDENT**

s

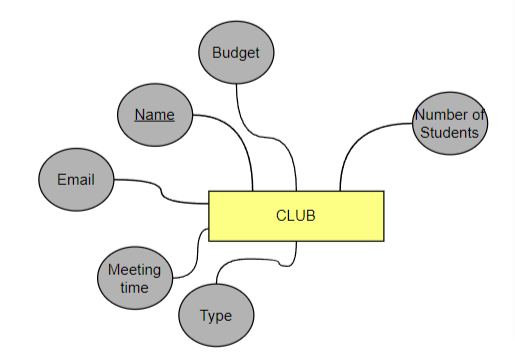
A student is the main entity type in the ARMY school database system. The student attends the school five days a week. In the database, we save the student’s name as a composite attribute: first name, father’s name, and last name. Moreover, each student has a **unique id** which is considered a key attribute in the **STUDENT** entity type. The **STUDENT** entity type has many other attributes such as birthdate and age which is derived from birthdate. The school should collect data about the student’s location. Therefore, the **STUDENT** entity type has an address which is composed of the name of the building, name of the city, and name of the street where he lives. The school also collects some information to communicate with the student in case of any emergency. Thus the **STUDENT** entity type has additional attributes such as phone number and email. The school helps the student reach his maximum potential so we make sure each student chooses the path that lets him/her explore his/her farthest potential.

**2) EMPLOYEE**



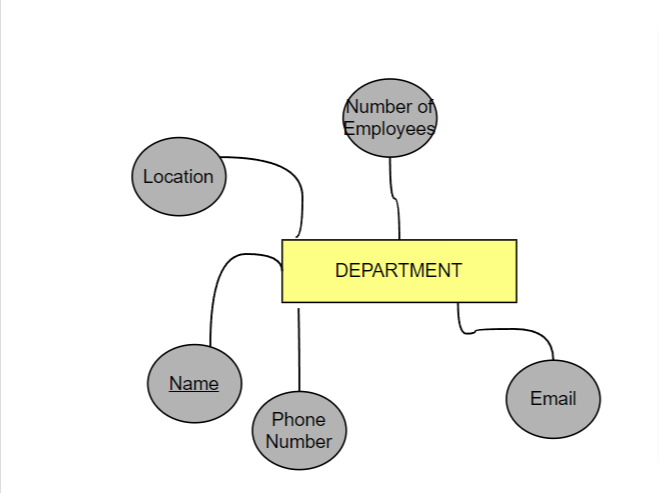
Like any institution in the world, the school has a bunch of employees. The **EMPLOYEE** entity type has a set of composite attributes including the address of the employee (composed of the name of the building, name of the street, and name of the city) and name of the employee ( composed of first name, middle name, and last name). To contact the employee, the school should have the email and the phone number of the employee. Of course, the **EMPLOYEE** entity type has a birthdate attribute and age derived from it. The key attribute for this entity type is the **ID** because each employee has a unique ID. Moreover, the **EMPLOYEE** entity type has a salary as an attribute in return to the employees’ work and a schedule attribute to specify the start and end of their work. The employees work in the school in different areas, and thus the **EMPLOYEE** entity type has a role attribute that specifies the role assigned to the employee. This entity type includes the instructor, janitor, security, etc…

**3) CLUB:**



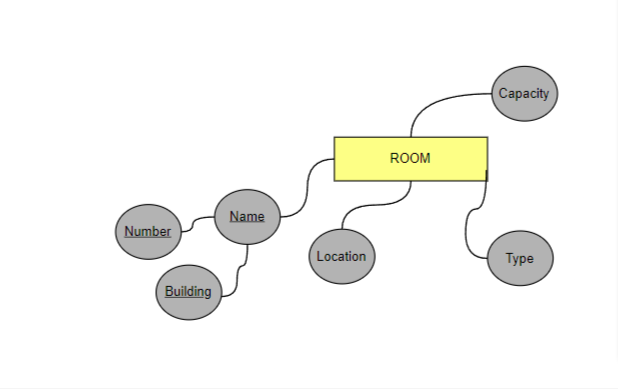
In ARMY school, the students engage in activities that satisfy their curiosity and interests so they have the ability to choose a club. The **CLUB** entity type in our database has the name of the club as the key attribute (it is unique) and has an email address to let the students and the school communicate with the club. Moreover, the **CLUB** entity type has the number of members as an attribute to indicate the number of students enrolled in the club in addition to the type of the club, its budget and its meeting time.

**4) DEPARTMENT**



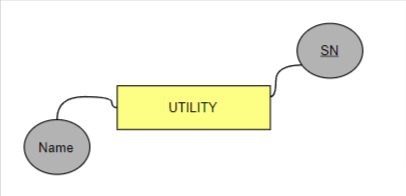
Every school has several departments which form the structure of this school. The name of the department is the key attribute of this entity type because each department has a unique name. Of course, every department can be reached through a phone number and email address. Finally, the **DEPARTMENT** entity typehas a certain number of employees that work in it. Since ARMY is an international school, the **DEPARTMENT** entity type has a location attribute to know to which campus the department belongs.

**5) ROOM**

****

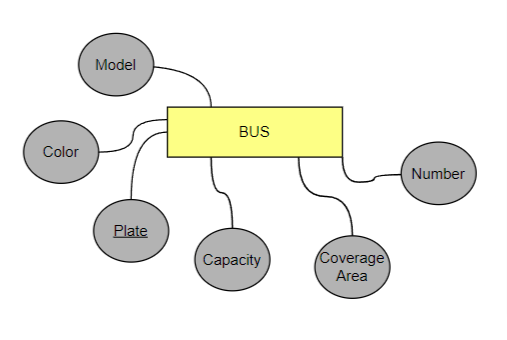
The school consists of several buildings and each building consists of several rooms. These rooms can be offices, classrooms, or any other type because this room can either be used as an office for an instructor or lab for students to experiment, or even a storage room to store our utility. Based on this, the **ROOM** entity type has a type as an attribute as well as location and capacity (number of seats). Moreover, the room has a unique composite attribute which is the name. The name of the room consists of the room number and the name of the building. Together the room number and the building form a unique name for any room in the school.

**6) UTILITY**



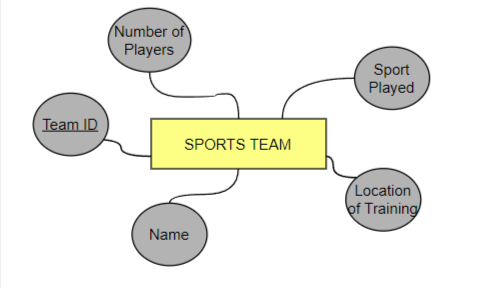
Every school has equipment and utilities. This utility can be a computer or book, a computer, a project or a whiteboard, or any other utility. The key attribute of the **UTILITY** entity type is the **serial number SN** because each utility has a unique serial number. Moreover, each utility has a certain name.

**7) BUS:**



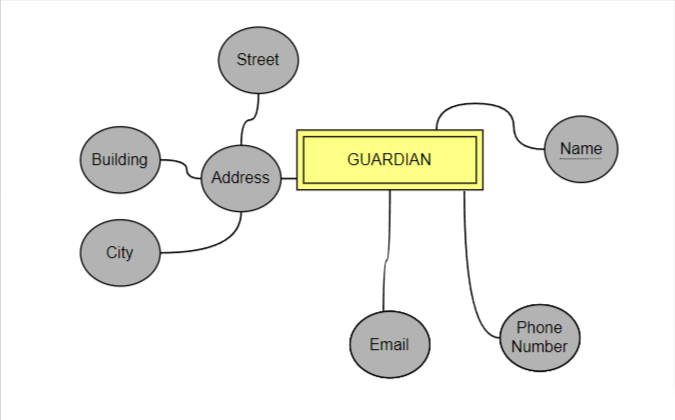
It is known that every school has a bus to take the students from their homes to the school. The **BUS** entity type has a plate number as the key attribute because each bus has a unique number. Of course, each bus has a maximum number of seats and a specific destination or coverage area (from which region the bus translates the students), in addition to a bus number, color and model.

**8) SPORTS TEAM**

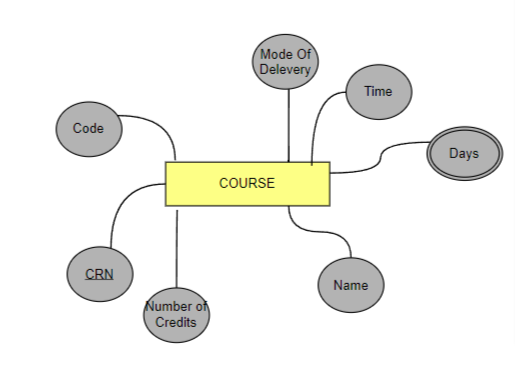
\

Sports teams exist in every school and they are very important because they help the students enhance their leadership skills and maintain good health. The **SPORTS TEAM** entity type has a composite Team ID as a key attribute. In addition, the sport team has a name, Number of players, sport played, Training location as attributes.

**9) GUARDIAN**

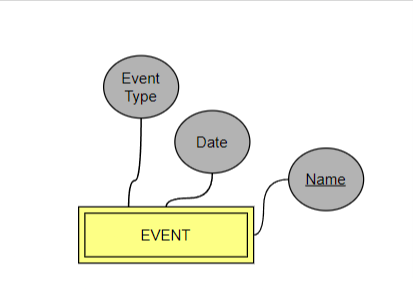


Each student in the school has a guardian or a dependent that can protect and help him/her whenever needed. The guardian entity type is weak because it doesn’t have a key attribute. However, the guardian entity type is identified by the guardian’s first name which is the partial key and the specific STUDENT with whom the guardian is related (through identifying relationship). The **GUARDIAN** entity type has the address of the guardian as a composite attribute (composite of the name of the building, name of the street, and name of the city). Of course, the school has contact info about the guardian, so the **GUARDIAN** entity type has attributes including the guardian’s phone number and email address.

**10) COURSE**

The main aim of the school is to invest the knowledge of the instructor to help the student acquire this knowledge. This can be done through courses that the instructor gives to the student. The **COURSE** entity type has a **unique CRN** and thus CRN is a key attribute for the COURSE entity type. Moreover, the course also has some other attributes such as the number of credits, the code or abbreviation of the course (such as CSC, MTH…), the full name of the course, and the mode of delivery (whether the course is delivered online or on-campus).Moreover, The COURSE entity type has a days and a time in which the course is delivered. The Days attribute is a multivalued attribute since the course can be delivered in many days not in a single one day.

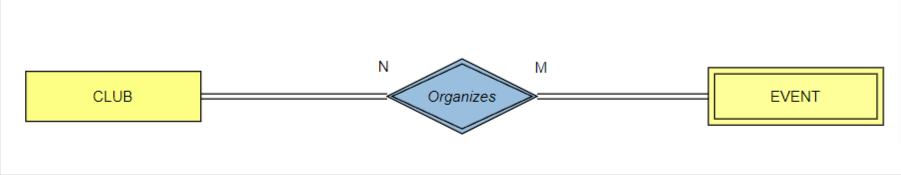
**11) Event:**



The EVENT entity type is a weak entity type with its name as a partial key. In addition this entity type has the event type and date as attributes. Also it is identified by the specific CLUB in which the activity is related to.

**V- Relationships:**

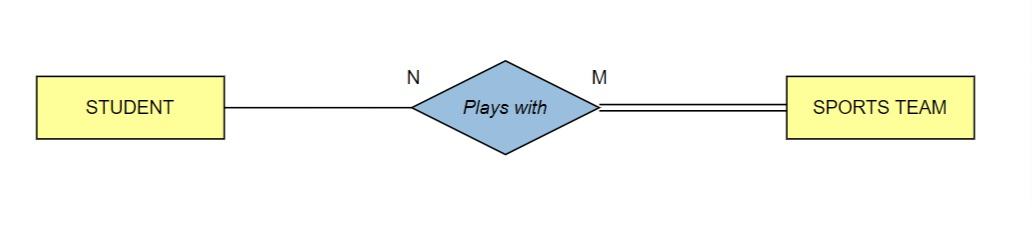
1. *Organizes:*



The club organizes events in the school. Thus there exists an “*organizes*” relation between the event entity type and club entity type. All clubs organize events and all the club events are organized by the clubs. Thus the participation is total from both sides of the relationship.

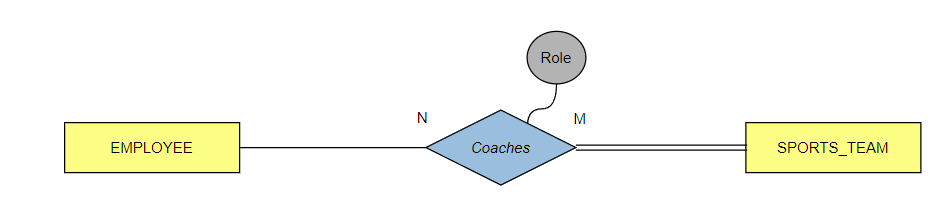
A club organizes one or more events, and an event can be organized by one or more clubs. (Many-to-many). The “*Organizes”* relationship acts as the identifying of the weak entity type EVENT with its identifying entity type CLUB.

1. *Plays\_with*

There exists a “plays with” relationship between the STUDENT entity type and the SPORTS TEAM entity type. All sports teams in the school have students; however, not all students belong to sports teams. Thus participation is total from the side of the sports team and partial from the side of students.

A student can join one or more sports teams and a sports team has many players (Many-to-many cardinality).

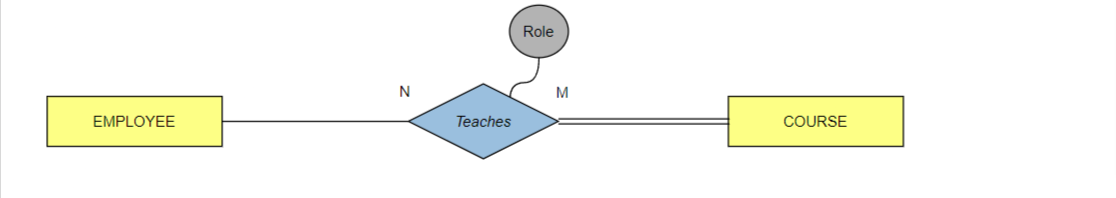
1. *Coaches*



It is known that every sports team has a coach. There exists a “*Coaches*” relationship between the and the EMPLOYEE and the SPORTS\_TEAM entity type and with partial participation on the side of the EMPLOYEE entity type and a total participation from the side of the SPORTS\_TEAM entity type.

An EMPLOYEE can train one or more SPORTS TEAM and a SPORTS TEAM can have one or more coaches. (Cardinality is many-to-many).

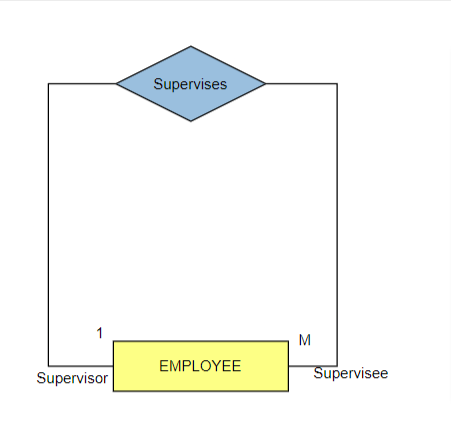
1. *Teaches*



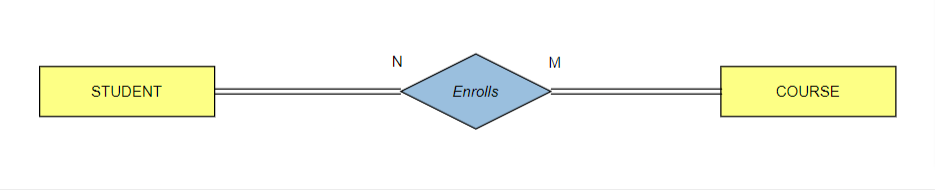
The courses in the school are taught by the instructors. Thus there exists a “*teaches”* relationship between the EMPLOYEE and the COURSE entities. All courses are taught by employees, but not all employees are instructors so not all of them teach courses. Thus, the participation is total only from the course side and partial from the employee side.

An employee can teach one or more courses and a course can be taught by one or more employees (ex: primary and secondary professor) (cardinality is many to many).

1. *Supervises*



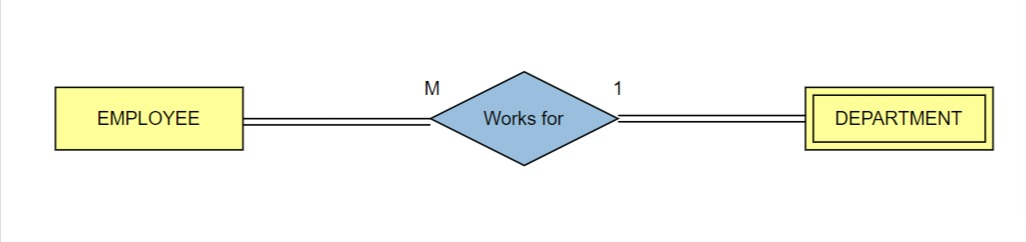
The employee can be supervised by another employee. This is a recurrence relationship. One employee can supervise many employees and the employees have only one employee that supervises them so the cardinality would be one-to-many. Not all employees are tasked with supervising other employees and not all employees can be supervised since some of them might be in higher positions so the relationship has partial participation from both sides.

1. *Enrolls i*n

The students learn in the school by enrolling in courses. Thus, there exists a “enrolls” relationship between the STUDENT entity type and COURSE entity type. All students are enrolled in courses and all courses are registered via students. Thus the participation is total on both sides.

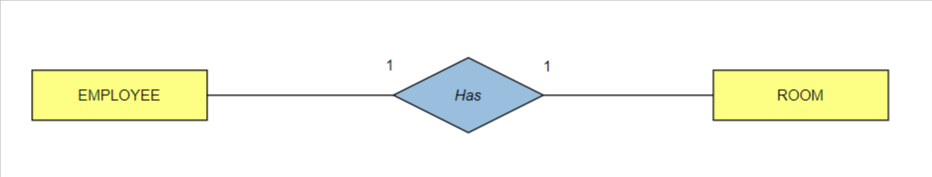
A student can enroll in one or many courses and a course can be registered by one or many students (the cardinality is many-to-many).

1. *Works for*



All EMPLOYEES *work* in a DEPARTMENT of the school and all DEPARTMENT have EMPLOYEES that *work* in them. Thus, the participation is total from both sides. An EMPLOYEE can *work* in one department, and a DEPARTMENT can have many EMPLOYEES. (Cardinality is one-to-many).

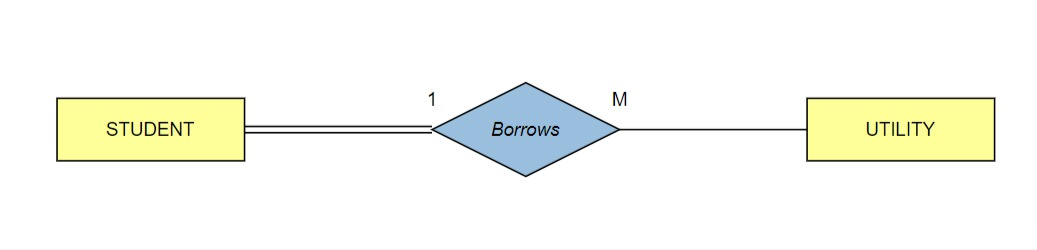
1. *Has*



Some of the employees have a room in the school that they can use as their office. Thus, there exists a “Has” relationship between the EMPLOYEE entity type and ROOM entity type. Some employees have rooms and not all rooms are used as offices to the instructors. Then the participation is partial from both sides of the relationship.

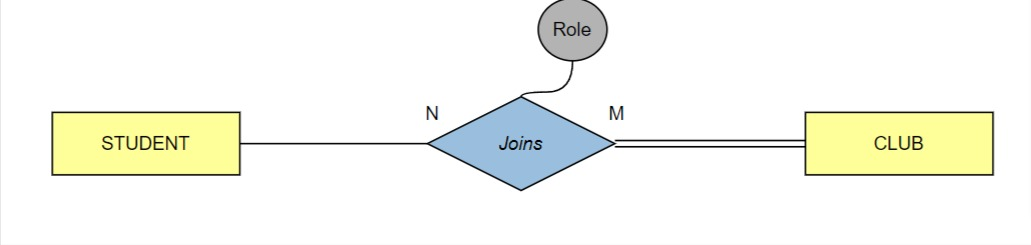
An employee has one room and a room can be related to one employee (Cardinality is one-to-one).

1. *Borrows*



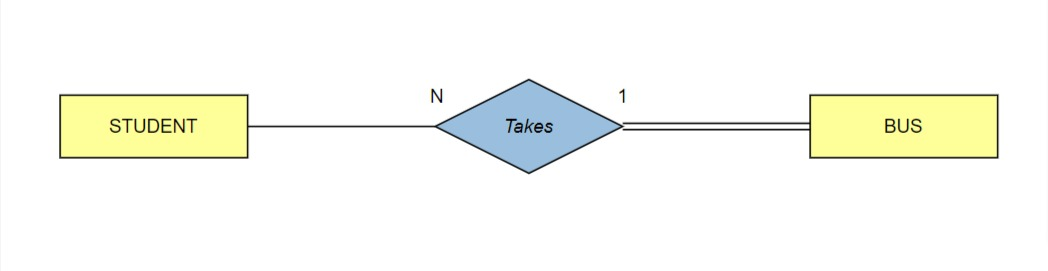
Every student *Borrows* utilities like computers and laptops. Therefore there exists the relation “Borrows” between STUDENT and UTILITY. All students borrow utilities (total participation) but not all utilities are used by students (partial participation). One student can borrow many utilities, but each utility can be borrowed by one student at a time thus the one-to-many cardinality.

1. *Joins*

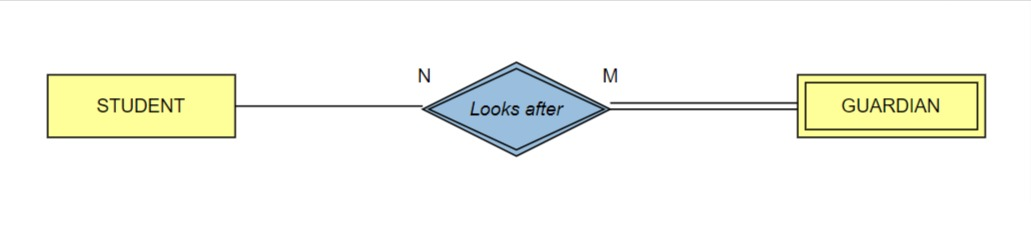


Students can choose to *join* clubs. Therefore there exists a relationship between the STUDENT and the CLUB entity type. Students are not obliged to join a club therefore the relation is partial from the side of the student. However, all the members of the club are students so participation has to be total from the side of the club. A student can join many clubs and a club usually has many members; therefore, the many to many cardinality of this relationship. When joining the club, the student can choose to be more than a regular member like a club president or vice-president so the club relationship has a role attribute.

1. *Takes*

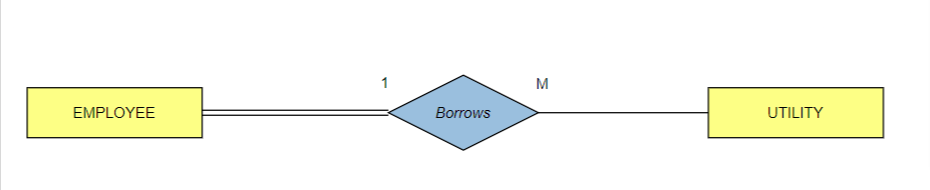
Students can *take* the bus. Therefore there exists a “Takes” relationship between STUDENT and BUS. Not all students take the bus (partial participation) but all buses are taken by students (total participation). Moreover, one bus is taken by many students and many students take one bus only (one-to-many cardinality).

1. *Looks after*



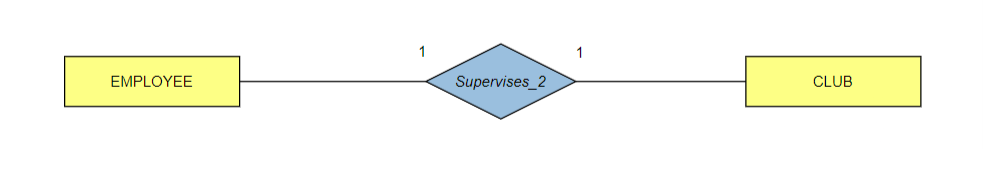
Students can have guardians that *look after* them. Therefore there exists a relationship between the STUDENT entity type and the GUARDIAN entity type. Not all students have guardians so the relationship is partial from the side of the STUDENT entity type but all guardians are there to look after the students thus the total participation from the side of the GUARDIAN entity type. A student can have more than one guardian that looks after them and the same guardian can have more than one student to look after thus the many-to-many cardinality. The “*Looks After”* relationship acts as the identifying of the weak entity type GUARDIAN with its identifying entity type STUDENT.

1. Borrows



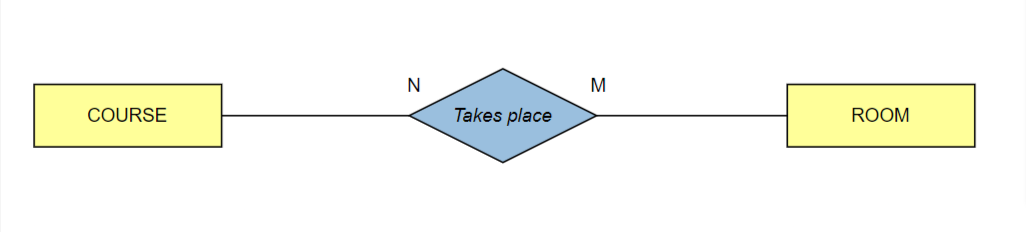
Every employee *borrows* utilities like computers and laptops. Therefore there exists the relation “Borrows 2” between EMPLOYEE and UTILITY. All employees borrow utilities (total participation) but not all utilities are used by employees (partial participation). One employee can borrow many utilities, but each utility can be borrowed by one employee at a time thus the one-to-many cardinality.

1. *Supervises*



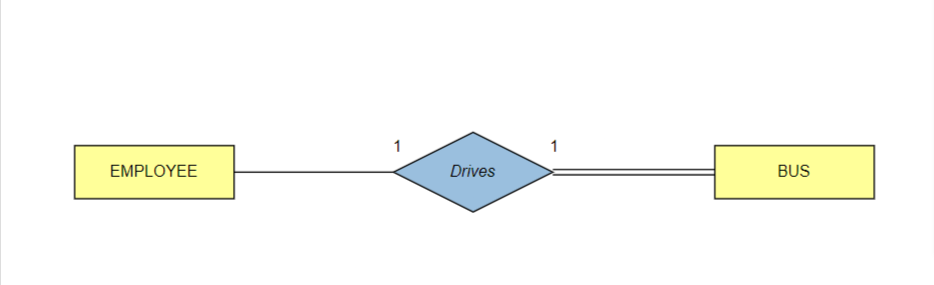
In this school some of the employees are tasked to supervise a club thus there exists a “Supervises” relationship between the EMPLOYEE and the CLUB entities. All clubs are being supervised by employees but not all employees are tasked with the supervision of clubs thus the total participation from the side of the CLUB entity type and the partial participation from the side of the EMPLOYEE entity type. Each club has one employee that supervises it and each instructor is in charge of supervising one club therefore the one-to-one cardinality.

1. *Takes Place*



Courses can be given in a physical room therefore there should be a “Takes place” relationship between the ROOM entity type and the COURSE entity type. Some courses can be given online therefore they have no need for a room and the participation would be partial form the side of the COURSE entity type and some rooms and not meant to be lecture rooms like offices and therefore the partial participation from the side of the ROOM entity type too. A course can be given in many rooms for example a lab room and a lecture room and a room can be used to teach many courses therefore the many-to-many cardinality.

1. *Drives*



Some employees work as bus drivers thus the relationship “Drives” Between the EMPLOYEE entity type and the BUS entity type. Not all employees drive buses but all the buses are driven by employees; and therefore, the participation is partial from the side of the EMPLOYEE entity type and total from the side of the BUS entity type. Each employee drives one bus and each bus has one driver thus the one-to-one cardinality.

**VI- ER to Relational Mapping Algorithm**

After designing the ER schema and having displayed the database for ARMY school as a

system of entities, attributes, and relationships, this high-level design must be translated into a relational database design. In order to map the ER design to a relational database design, a seven-step algorithm needs to be followed. The following is a detailed description of applying the different steps to our database design.

**STEP 1: Mapping of Regular Entity Types**

In the first step, the regular entity types must be mapped into relations. Each regular entity type is going to have its own relation that includes all of its simple attributes and a single primary key which is underlined. The regular entity types in this database are DEPARTMENT, EMPLOYEE, ROOM, COURSE, STUDENT, BUS, UTILITY, CLUB, and SPORTS\_TEAM.

1. DEPARTMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Location | Email | Phone\_Number |

The DEPARTMENT entity type contains only simple attributes so we simply include in this relation these attributes: Location, Email, Phone, and the primary key Name which is underlined.

1. EMPLOYEE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Age | Birthday | Phone\_Number | Email | Role | Schedule |  |

The EMPLOYEE entity type contains simple, derived, and composite attributes. The derived attribute Age is not represented in this relation. This relation only includes all simple attributes and the primary key ID which is underlined. The EMPLOYEE entity type has Named as a composite attribute of which only the simple attributes FirstName, LastName, and Father’s name are included in the relation.

1. ROOM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Building | Number | Capacity | Type | Location |

The ROOM entity type contains only simple attributes so we included these attributes which are: Capacity, Type, Location, Building, and Number. The attributes Building and Number together constitute the primary key to this Entity type which is why they are underlined.

1. COURSE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CRN | Name | Code | Nb\_of\_Credits | Time | Mode\_ of\_Delivery |

The COURSE entity type contains only simple attributes so we included these attributes which are: Name, Code, Nb\_of\_Credits, Time, Mode\_of\_Delevery, and the primary key CRN which is underlined.

1. STUDENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Email | Phone\_Number | Birthday | Age | Path |  |  |

The STUDENT entity type contains simple, derived, and composite attributes. This relation only includes all simple attributes and the primary key ID which is underlined. The STUDENT entity type has Name as a composite attribute of which only the simple attributes FirstName, Fathers\_Name, LastName are included in the table. Similarly, this entity type has also Address as a composite attribute and it is composed of simple attributes City, street, and building that are included in the table. Finally, the STUDENT entity type has also other simple attributes such as Phone\_number, Birthday, and Path of the Student.

1. BUS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Plate | Number | Coverage Area | Capacity | Color | Model |

The BUS entity type has a Primary Key which is the plate number of the bus. Moreover, this entity type has other simple attributes that are included in the table such as the number of this bus, coverage area, Capacity of the bus, color, and Model of the Bus.

1. UTILITY

|  |  |
| --- | --- |
| SN | Name |

The UTILITY entity type contains simple attributes. Simple attributes are the primary key SN which is underlined and Name.

1. CLUB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Budget | Meeting\_Time | Type | Email |

The CLUB entity type contains simple attributes. The simple attributes are Name (primary key), Budget, Meeting\_Time, Type, and Email. This relation only includes all simple attributes and the primary key Name which is underlined.

1. SPORTS\_TEAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team\_ID | Name | Sport\_Played | Location\_of\_Training | Time |

The SPORTS\_TEAM entity type contains simple attributes. The simple attributes are Team\_ID (primary key), Name, Sport\_Played, Location\_of\_Training, and Time. This relation only includes all simple attributes and the primary key Team\_ID which is underlined.

**STEP 2: Mapping of Weak Entity Types**

In this step, the weak entity types are mapped into relations. Only the simple attributes are included in the relations and not multivalued or derived attributes. Moreover, the primary key of the owner entity type is the foreign key attribute for the weak entity type. The primary key of the relation is the combination of the foreign key added and the partial key of the weak entity type. The weak entity types are GUARDIAN and EVENT.

1. GUARDIAN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| StudentID | Name | Phone\_Number | Email | Street | Building | City |

The weak entity type GUARDIAN does not have any derived or multivalued attributes. The simple attributes: Name (the partial key), Phone\_Number, Email, Street, Building, and City are included. Moreover, the StudentID, the primary of the owner entity type STUDENT, is included. StudentID and the partial key Name are combined to represent the primary key of this relation. The GUARDIAN entity type has Address as a composite attribute of which only the simple attributes Street, Building, and City are included in the relation.

1. EVENT

|  |  |  |  |
| --- | --- | --- | --- |
| ClubName | Name | Date | Event\_type |

The weak entity type EVENT includes only simple attributes. The simple attributes: Name (the partial key), Event\_type, and Date are included. Moreover, the ClubName, the primary of the owner entity type CLUB, is included. ClubName and the partial key Name are combined to represent the primary key of this relation.

**STEP 3: Mapping of Binary 1:1 Relationship Types**

In the third step, we map the one-to-one relationships into tables. There are three methods that can be followed in order to accomplish this step.

i) Foreign key (2 relations): choose one of relations that are involved, most preferably one that has total participation, and add a foreign key to that relation (this foreign key is the primary key of the other relation)

ii) Merged relation (1 relation): merge the two entity types into one relation. The primary key of this relation would be the combination of the key attribute of each entity type. (This approach can be appropriate if the relationship one both sides is total)

iii) Cross-reference or relationship-relation (3 relations): make a third relation in addition to the two entities that became relations. This third relation would include the primary keys of the two previous relations, and it would include the relationship attributes.

1. BUS (*Drives*)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plate | Number | Coverage\_Area | Capacity | Color | Model | DriverID |

The “drives” relationship links the EMPLOYEE entity type and the DEPARTEMENT entity type. The BUS entity type is on the total participation side. Thus, we add to its relation the foreign key which is the primary key of the EMPLOYEE entity type and we rename it DriverID.

1. OFFICE(*Has)*

|  |  |
| --- | --- |
| Room\_Number | Employee\_ID |

The ‘has’ relationship links EMPLOYEE entity type to ROOM entity type. A third relation is created that includes the primary keys of both entities which are the Room\_Number from the ROOM entity type and the Employee\_ID from the EMPLOYEE entity type.

1. CLUB (Supervises)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Budget | Meeting\_Time | Type | Email | SupervisorID |

The “supervises” relationship links the EMPLOYEE entity type and the CLUB entity type. The CLUB entity type is on the total participation side. Thus, we add to its relation the foreign key which is the primary key of the EMPLOYEE entity type and we rename it SupervisorID.

**STEP 4: Mapping of Binary 1:M Relationship Types**

In this step, we are going to map the binary one-to-many relationships. We shouldadd to the participating entity type (the one that is on the M side of the relation) a foreign key that is the primary key of the other relation. Also, we must add the relationship attributes to the participating entity type.

1. STUDENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Email | Phone\_Number | Birthday | Age | Path | Bus\_Taken |  |

Many students belong to one bus. The Takes relationship links the STUDENT entity type and the DEPARTEMENT entity type. The STUDENT entity type is on the “many'' side. Thus, we add to its relation the foreign key ID which is the primary key of the BUS entity type and we rename it Bus\_Taken.

1. Employee *(Belongs\_to)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Age | Birthday | Phone\_Number | Email | Role | Schedule | Name\_of\_department |

Many employees belong to one department. The ‘belongs to’ relationship links the EMPLOYEE entity type and the DEPARTEMENT entity type. The EMPLOYEE entity type is on the “many'' side. Thus, we add to its relation the foreign key ID which is the primary key of the DEPARTMENT entity type and we rename it Name\_Of\_Department.

1. Employee *(Supervises\_1)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Age | Birthday | Phone\_Number | Email | Role | Schedule | Name\_of\_department |
|  |  | SupervisorID |  |  |  |  |

Many employees are being supervised by one other employee. The “Supervises\_1” relationship links the EMPLOYEE entity type with itself and the supervisee is on the “many”

side so, we added a foreign key SupervisorID which is the ID of the employee supervision the others knowing that ID is the primary key in the entity type EMPLOYEE.

1. UTILITY*(borrows)*

|  |  |  |
| --- | --- | --- |
| SN | Name | Student\_ID |

A student may borrow many utilities. The *Borrows\_1* relationship links the STUDENT entity type and the UTILITY entity type. The part that has the many (M) is the UTILITY, so we add to its relation the foreign key ID which is the primary key of the STUDENT entity type and we rename it Student\_ID.

1. UTILITY *(borrows\_2)*

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Name | STUDENT\_ID | EMPLOYEE\_ID |

An employee may borrow many utilities. The *Borrows\_2* relationship links the EMPLOYEE entity type and the UTILITY entity type. The part that has the many (M) is the UTILITY, so we add to its relation the foreign key ID which is the primary key of the EMPLOYEE entity type and we rename it Employee\_ID.

**STEP 5: Mapping of Binary M:N Relationship Types**

In this step, we are going to map the binary many-to-many relationships. We should create a new relation (a relationship-relation), include in it as foreign key attributes the two primary keys of the two relations, the combination of these two will form the primary key of the created relation. Also, add the relationship’s simple attributes to the third relation.

1. PLAYS\_WITH

|  |  |
| --- | --- |
| StudentID | SportTeam\_Name |

Many students can join many sports teams. The “Plays\_with” relationship links the STUDENT entity type with the SPORT\_TEAM entity type. The relationship “Plays\_with” includes the primary key of the STUDENT and SPORT\_TEAM entity types which are “StuendID” and “SportTeam\_Name” respectively. The combination of both of these attributes constitutes the primary key of the “Plays\_with” relation and they are underlined

.

1. COACHES

|  |  |  |
| --- | --- | --- |
| EmployeeID | SportTeam\_Name | Role |

Many employees can teach many sports teams. The “Coaches” relationship links the EMPLOYEE entity type with the SPORT\_TEAM entity type. The relationship “Coaches” includes the primary key of the EMPLOYEE and SPORT\_TEAM entity types which are “EmployeeID” and “SportTeam\_Name” respectively. The combination of both of these attributes constitutes the primary key of the “Plays\_with” relation and they are underlined. In addition, we added the role attribute which was an attribute on the “Coaches” relationship.

1. JOINS

|  |  |  |
| --- | --- | --- |
| Student\_ID | Club\_Name | Role |

Many students join many clubs. The *Joins* relationship links the STUDENT entity type and the CLUB entity type. We create a new relation called “Joins” in which we include the primary keys of the STUDENT and CLUB entity types. The primary key of the CLUB entity type is Name which is added to the table and renamed to Club\_Name. Also, ID is the primary key of the STUDENT entity type that is added and renamed to Student\_ID. The combination of added keys represents the primary key of the “Joins” relation and thus is underlined. We also have an attribute for the relationship *joins* so we added it to the table without underlining it.

1. ENROLLS

|  |  |
| --- | --- |
| Student\_Id | Course\_CRN |

Many students enroll in many courses. The “Enrolls” relationship links the STUDENT entity type and the COURSE entity type. A new relation was called “Enrolls” in which we include the primary keys of the STUDENT and COURSE entities. The primary key of the STUDENT entity type, ID, is added to the “Enrolls” relation and renamed Student\_id. Also, the primary key of the COURSE entity type, CRN, is added and renamed Course\_CRN. The combination of both keys added represents the primary key of the “Enrolls” relation and thus are underlined.

1. TEACHES

|  |  |
| --- | --- |
| Employee\_ID | Course\_CRN |

Many employees (Instructors) teach many courses. The “Teaches” relationship links the EMPLOYEE entity type and the COURSE entity type. A new relation was called “Teaches” in which we include the primary keys of the Employee and COURSE entities. The primary key of the Employee entity type, ID, is added to the “teaches” relation and renamed Employee\_id. Also, the primary key of the COURSE entity type, CRN, is added and renamed Course\_CRN. The combination of both keys added represents the primary key of the ‘teaches” relation and thus are underlined.

1. TAKES\_PLACE

|  |  |
| --- | --- |
| Course\_CRN | Room\_Name |

A course may be taught in many rooms, and a room can be the location of many courses. The “Takes\_Place” relationship links the ROOM entity type and the COURSE entity type. The primary key of the entity type Room, Name, is added to the “Takes\_Place” relation and renamed Room\_Name. Also, the primary key of the COURSE entity type, CRN, is added and renamed Course\_CRN. The combination of both keys added represents the primary key of the ‘Takes\_Places” relation and thus are underlined.

**STEP 6: Mapping of Multivalued attributes**

For **multivalued attributes**, create a relation for each one of them, including in that relation an attribute corresponding to the multivalued attribute. Also add to this relation a foreign key, which is the primary key of the relation that owns the multivalued attribute. The primary key of this relation is the combination of the previous two attributes. If a multivalued attribute is composite, then include its simple components in the relation.

1. COURSEDAYS

|  |  |
| --- | --- |
| CourseCRN | Days |

The multivalued attribute Days belongs to the COURSE entity type. To represent it, we create a relation called “COURSEDAYS”. Its primary key is composed of CourseCRN, the primary key of the course entity type, and the attribute Days which represents the different days in which this course is taught.

**STEP 7: Mapping of N-ary relationships**

Identify all **N-ary relationship types** (n>2), create a new relation for each relationship, then include as foreign key attributes all the key attributes of the participating entity-types (relations). The combination of these foreign keys will be the primary key of the created relation. Also, add the relationship’s simple attributes to the relation.

**FINAL STEP: Final Displays**

EMPLOYEE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Age | Birthday | Phone\_Number | Email | Role | Schedule | Name\_of\_department |
|  |  | SupervisorID |  |  |  |  |

STUDENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | First\_Name | Fathers\_Name | Last\_Name | City | Street | Building |
| Email | Phone\_Number | Birthday | Age | Path | Bus\_Taken |  |

BUS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plate | Number | Coverage\_Area | Capacity | Color | Model | DriverID |

DEPARTMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Location | Email | Phone\_Number |

UTILITY

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Name | STUDENT\_ID | EMPLOYEE\_ID |

CLUB

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Budget | Meeting\_Time | Type | Email | SupervisorID |

ROOM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Building | Number | Capacity | Type | Location |

COURSE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CRN | Name | Code | Nb\_of\_Credits | Time | Mode\_ of\_Delivery |

SPORTS\_TEAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team\_ID | Name | Sport\_Played | Location\_of\_Training | Time |

GUARDIAN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| StudentID | Name | Phone\_Number | Email | Street | Building | City |

EVENT

|  |  |  |  |
| --- | --- | --- | --- |
| ClubName | Name | Date | Event\_type |

OFFICE

|  |  |
| --- | --- |
| Room\_Number | Employee\_ID |

PLAYS\_WITH

|  |  |
| --- | --- |
| StudentID | SportTeam\_Name |

COACHES

|  |  |  |
| --- | --- | --- |
| EmployeeID | SportTeam\_Name | Role |

JOINS

|  |  |  |
| --- | --- | --- |
| Student\_ID | Club\_Name | Role |

ENROLLS\_IN

|  |  |
| --- | --- |
| Student\_Id | Course\_CRN |

TEACHES

|  |  |
| --- | --- |
| Employee\_ID | Course\_CRN |

TAKES\_PLACE

|  |  |
| --- | --- |
| Course\_CRN | Room\_Name |

COURSEDAYS

|  |  |
| --- | --- |
| CourseCRN | Days |

**VIII-Conclusion:**

Databases are a vital component in our life as they will organize all the essential and crucial information and data needed by a specific institution. Indeed, schools are one of these institutions that require a database to save all their data. Moreover, this database will be a necessity for the administration and functioning of the school from the salaries of instructors, courses taken by students, tuition of students, etc... Schools are educational institutions designed to provide learning spaces and learning environments for the teaching of students under the direction of instructors. As the definition of the dictionary, the school not only provides information, but also discipline and skills that are later implemented in life.

**IX- Phase II Report Checklist:**

Font Type: Times New Roman.

Team of three: Yes.

Plain English: Yes.

Group named: Yes.

Group leader: Yes.

Real-life database application: Yes.

Challenging and interesting database application: Yes.

Font Size: 11.

1.5 Spaced: Yes.

Typed: Yes.

At least 10 essential pages: Yes.

At least 10 entities: Yes.

At least 1 multivalued attribute: Yes.

At least 1 weak entity type: Yes.

ER Model built: Yes

Deadline respected: Yes.

**IX. Instructor’s feedback and evaluation** T

This page should be used by our instructor DR. RAMZI HARATY to write any comments, feedback and suggestions for our database system. This page will be used to improve our final report.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_