

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 – Phase III – Phase IV

1 December 2021

Table of Contents:

I- Introduction.....	3
II- System Description and Constraints.....	4
III- ER Diagram for ARMY School	6
IV- Entity types	7
1. Student.....	7
2. Employee.....	8
3. Club.....	9
4. Department.....	10
5. Room.....	11
6. Utility.....	12
7. Bus.....	13
8. Sport Team.....	13
9. Guardian.....	14
10. Course.....	15
11 Event.....	15
V-Relationships.....	16
VI- ER to Relational Mapping Algorithm.....	24
Step1: Mapping of Regular Entity types	24
Step2: Mapping of Weak Entity types	26
Step3: Mapping of 1:1 Binary Relationship Types.....	27
Step4: Mapping of 1:M Binary Relationship Types.....	28
Step5: Mapping of M:N Binary Relationship Types.....	29
Step6: Mapping of Multivalued attributes.....	31
Step7: Mapping of N-ary Relationships Types.....	31
FINAL STEP: Final Display.....	32
VII- Table Structure for ARMY School.....	33
VIII- Table Description.....	43
IX- Inserting Data.....	48
X- Final Tables State	65
XI- Queries.....	76
XII- Normalization Up to The BCNF Normal Form.....	86
XIII- Conclusion.....	98
XIV- Phase 4 Report Checklist.....	99
XV- GitHub Repository	100
XVI- Instructor's Feedback and Evaluation.....	101

I-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.

II- 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, his 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.

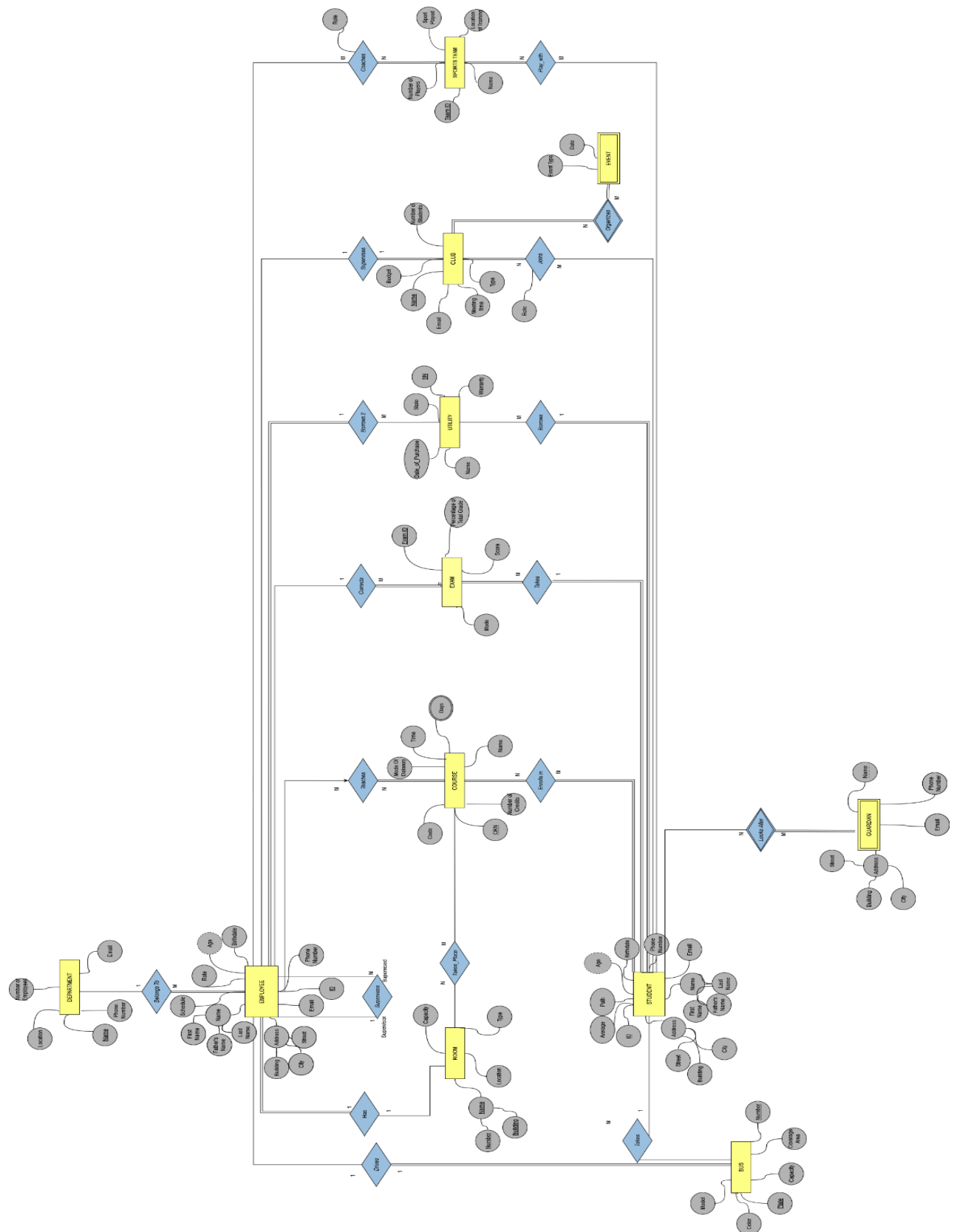
The students' privileges may also include the usages of facilities (gym, swimming pool, library, cafeteria, etc...) and utilities (computers, books, projectors, boards, etc...).

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.

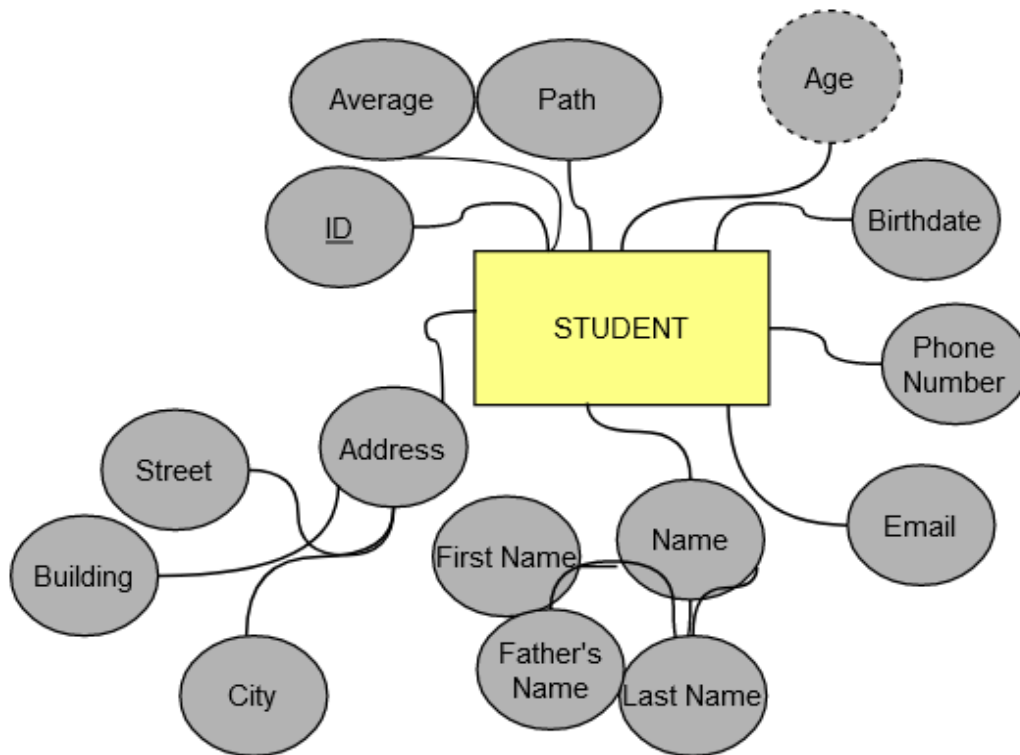
The instructors at ARMY school put all their efforts to provide a decent education for the students.

III- ER Diagram:



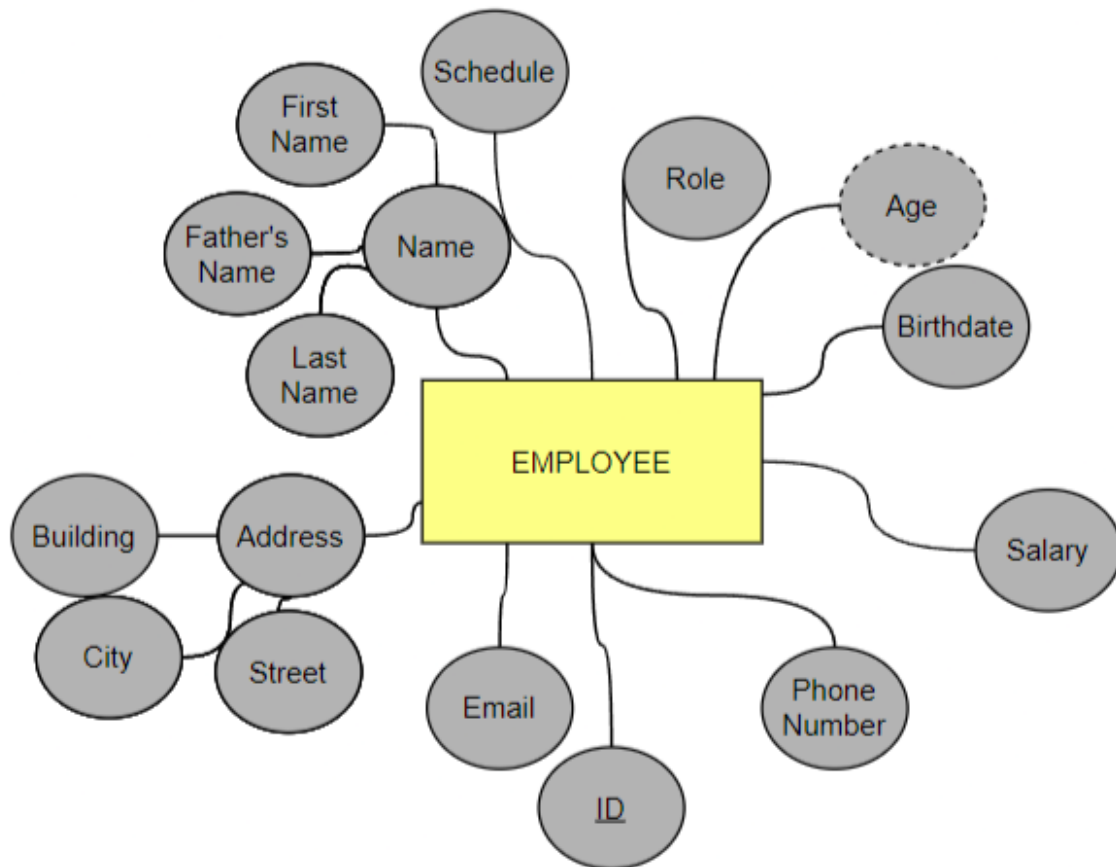
IV- Entity Types:

1) STUDENT



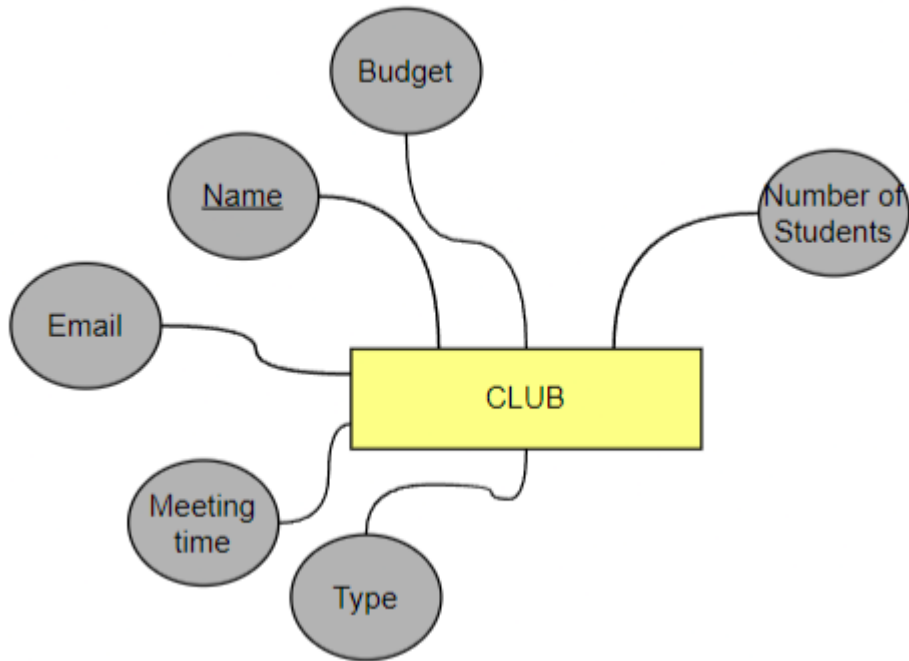
A student is the main entity type in the ARMY school database system. The student attends the school five days a week. In the database, we save the student's **name** as a composite attribute: first name, father's name, and last name. Moreover, each student has a **unique id** which is considered a key 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, in addition their **average**.

2) EMPLOYEE



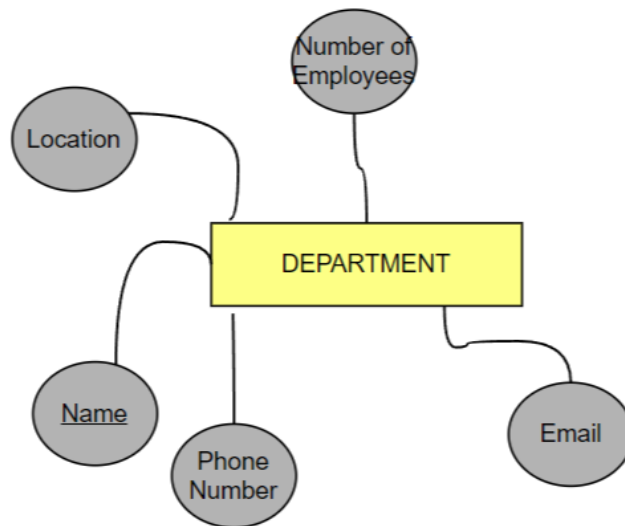
Like any institution in the world, the school has a bunch of employees. The **EMPLOYEE** entity type has a set of composite attributes including the **address** of the employee (composed of the name of the building, name of the street, and name of the city) and **name** of the employee (composed of first 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:



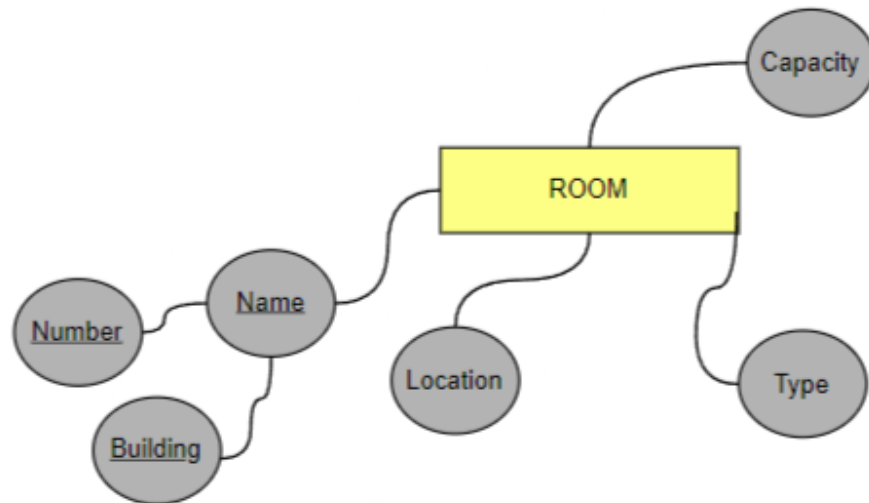
In ARMY school, the students engage in activities that satisfy their curiosity and interests so they have the ability to choose a club. The **CLUB** entity type in our database has the **name** of the club as the key attribute (it is unique) and has an **email** address to let the students and the school 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



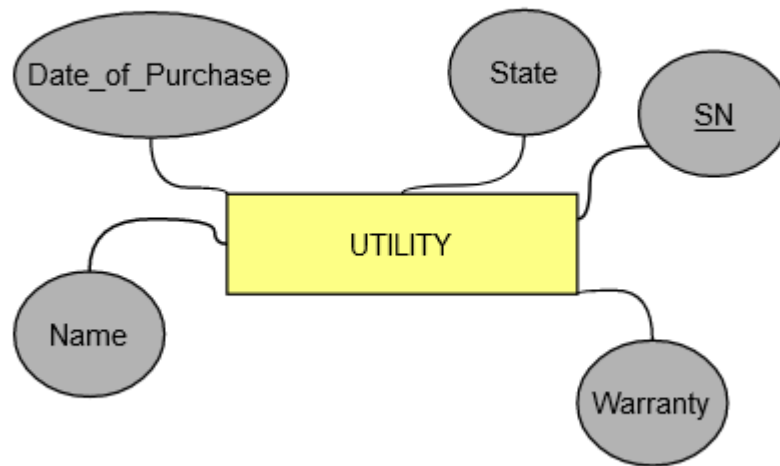
Every school has several departments which form the structure of this school. The **name** of the department is the key attribute of this entity type because each department has a unique name. Of course, every department can be reached through a **phone number** and **email address**. Finally, the **DEPARTMENT** entity type has 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



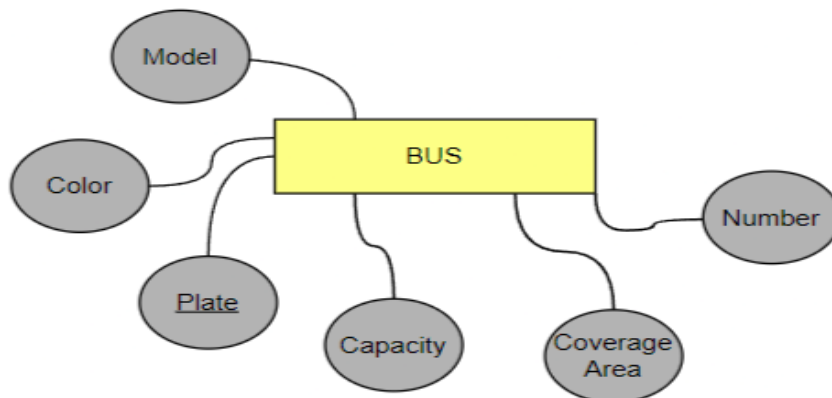
The school consists of several buildings and each building consists of several rooms. These rooms can be offices, classrooms, or any other type because this room can either be used as an office for an instructor or lab for students to experiment, or even a storage room to store our utility. Based on 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



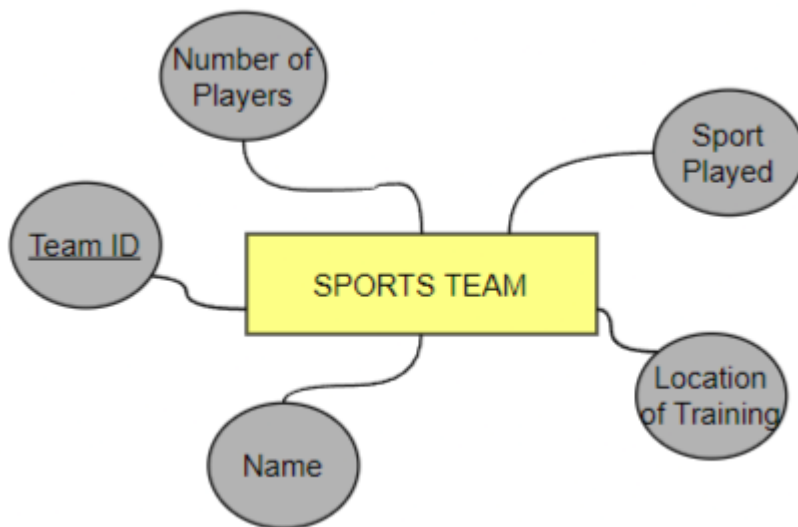
Every school has equipment and utilities. This utility can be a computer or book, a computer, a project or a whiteboard, or any other utility. The key attribute of the **UTILITY** entity type is the **serial number SN** because each utility has a unique serial number. Moreover, each utility has a certain **name**, **state** that specifies if it is good to use or needs repair, **warranty** that specifies how many years of warranty do we have from the date of purchase and finally **date_of_purchase**.

7) BUS:



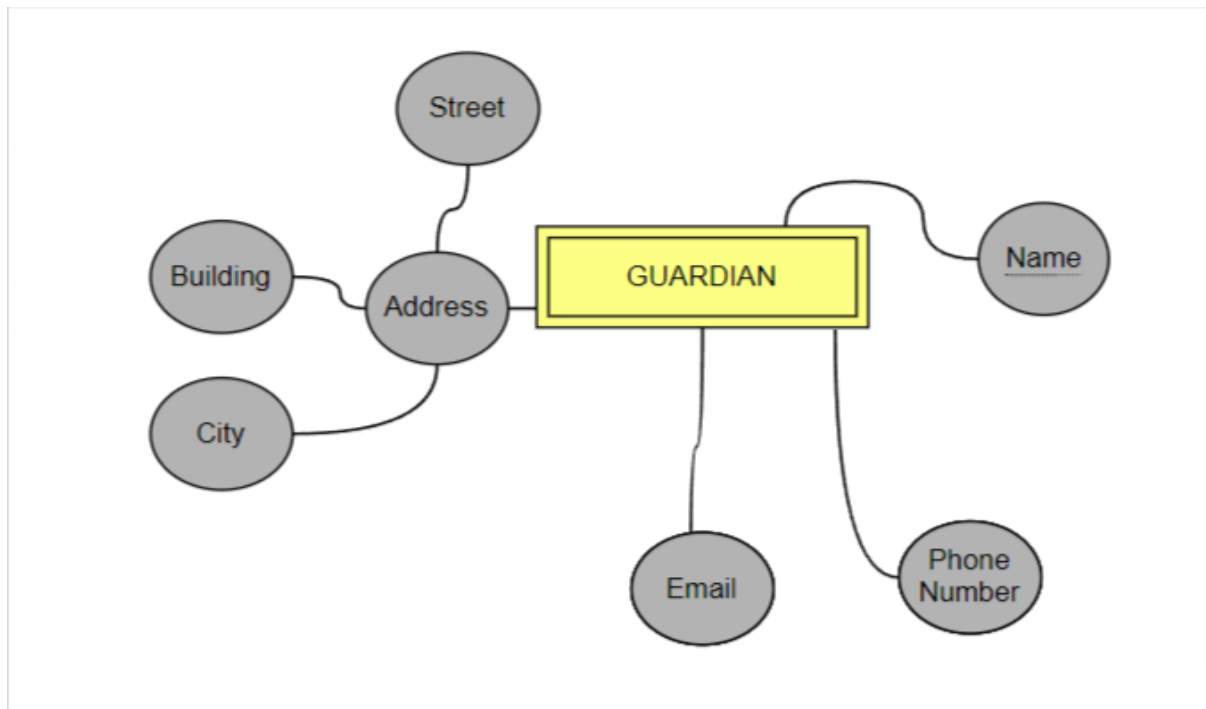
It is known that every school has a bus to take the students from their homes to the school. The **BUS** entity type has a **plate number** as the key attribute because each bus has a unique number. Of course, each bus has a maximum **number of seats** and a specific destination or **coverage area**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

8) SPORTS TEAM



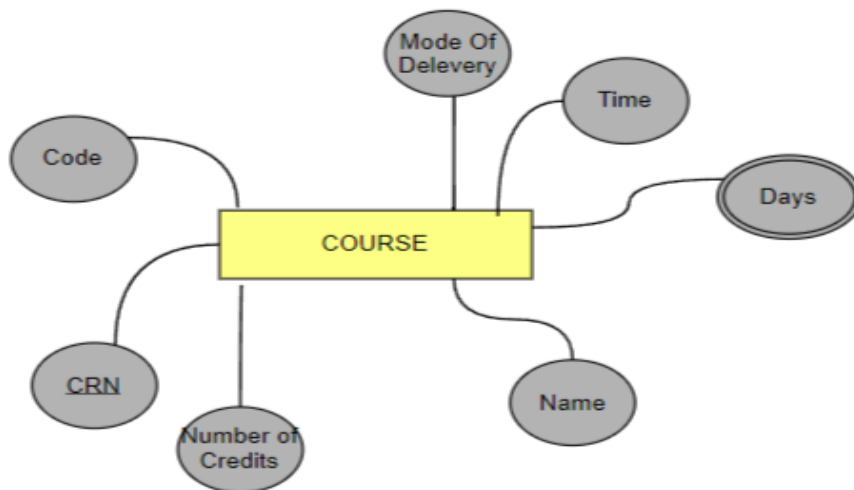
Sports teams exist in every school and they are very important because they help the students enhance their leadership skills and maintain good health. The **SPORTS TEAM** entity type has a composite **Team ID** as a key attribute. In addition, the sport team has a **name**, **Number of players**, **sport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

9) GUARDIAN



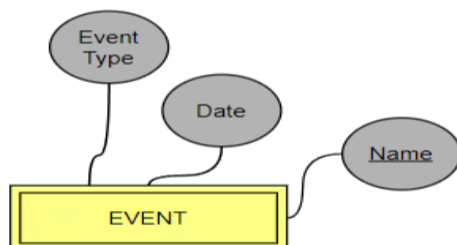
Each student in the school has a guardian or a dependent that can protect and help him/her whenever needed. The guardian entity type is weak because it doesn't have a key attribute. However, the guardian entity type is identified by the guardian's first **name** which is the partial key and the specific 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.

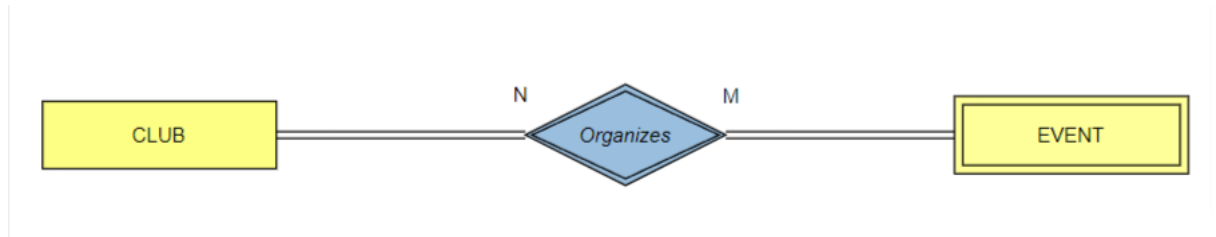
11) Event:



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:

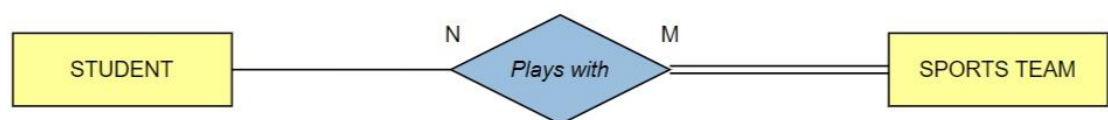
1) *Organizes:*



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.

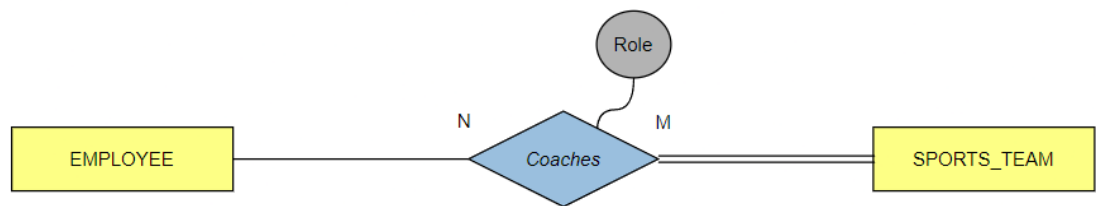
2) *Plays_with*



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).

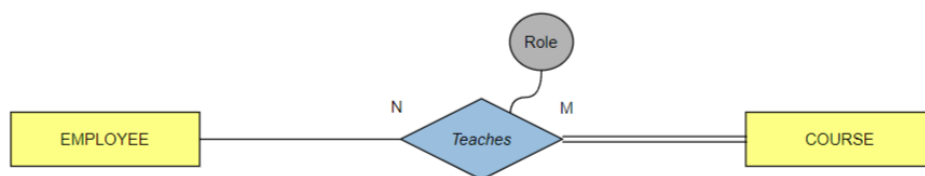
3) *Coaches*



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).

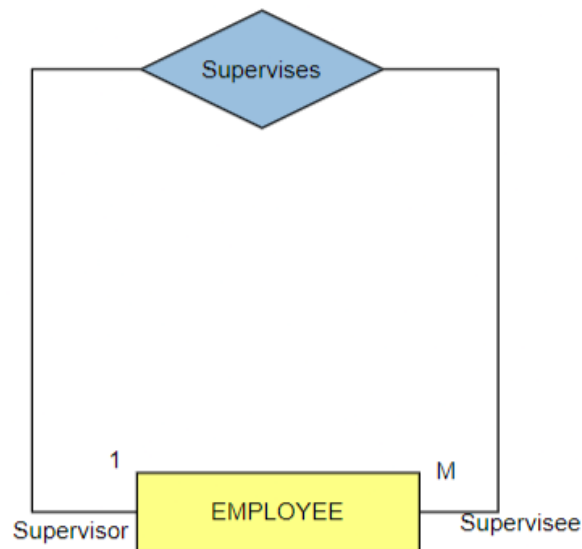
4) *Teaches*



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).

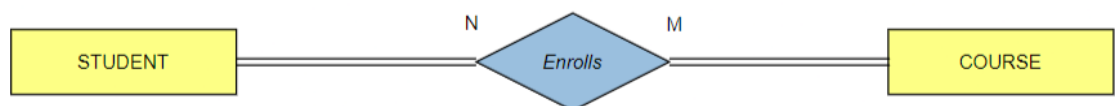
5) *Supervises*



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.

6) *Enrolls*

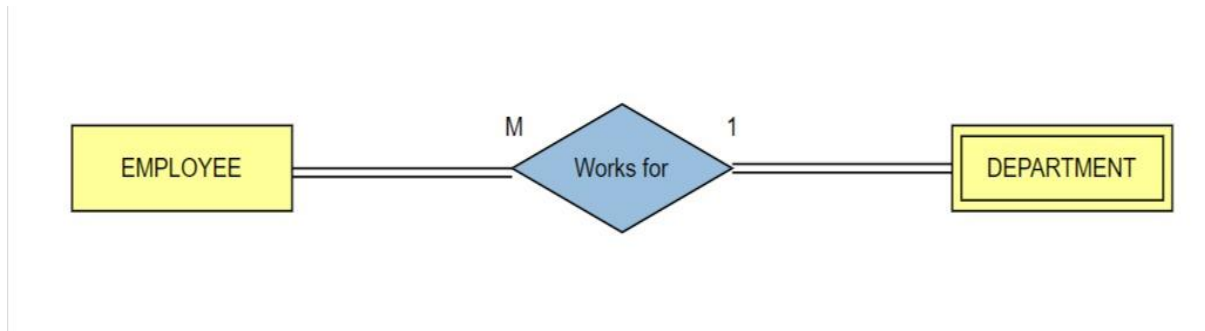
in



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.

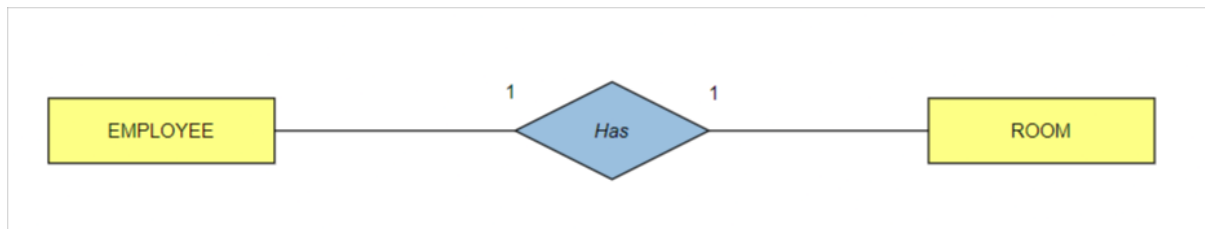
A student can enroll in one or many courses and a course can be registered by one or many students (the cardinality is many-to-many).

7) *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).

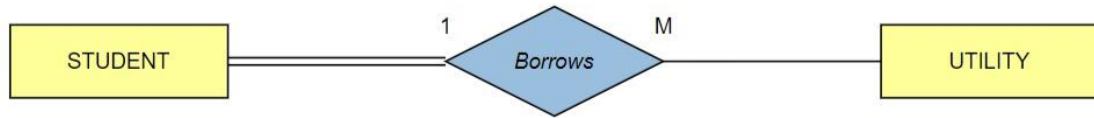
8) *Has*



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.

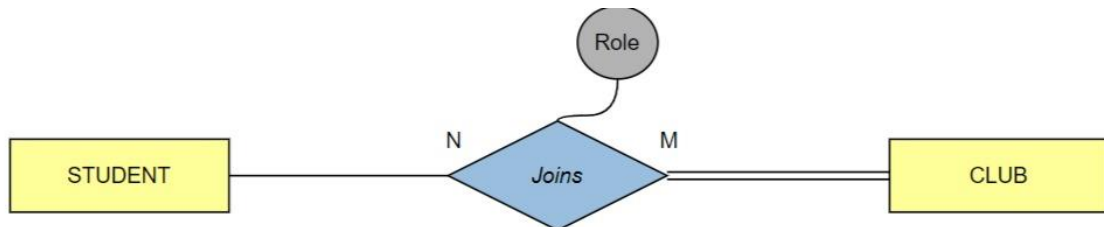
An employee has one room and a room can be related to one employee (Cardinality is one-to-one).

9) *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.

10) *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.

11) Takes



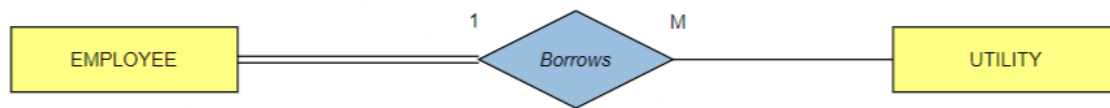
Students can *take* the bus. Therefore, there exists a “Takes” relationship between STUDENT and BUS. Not all students take the bus (partial participation) but all buses are taken by students (total participation). Moreover, one bus is taken by many students and many students take one bus only (one-to-many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

12) Looks after



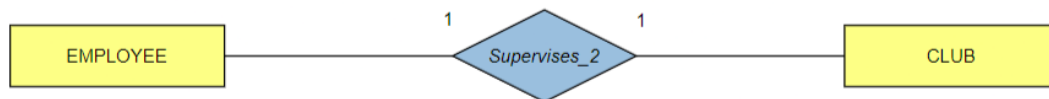
Students can have guardians that *look after* them. Therefore, there exists a relationship between the STUDENT entity type and the GUARDIAN entity type. Not all students have guardians so the relationship is partial from the side of the STUDENT entity type but all guardians are there to look after the 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.

13) Borrows



Every employee *borrow*s 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.

14) Supervises



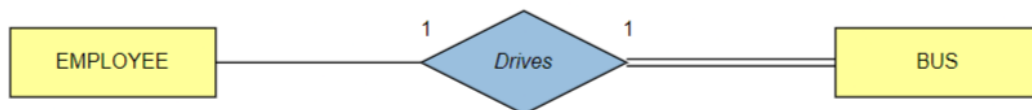
In this school some of the employees are tasked to supervise a club thus there exists a “Supervises” relationship between the EMPLOYEE and the CLUB entities. All clubs are being supervised by employees but not all employees are tasked with the supervision of clubs thus the total participation from the 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.

15) 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 from the side of the COURSE entity type and some rooms are 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.

16) 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

<u>Name</u>	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.

2. EMPLOYEE

<u>ID</u>	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.

3. ROOM

<u>Building</u>	<u>Number</u>	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.

4. COURSE

<u>CRN</u>	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_Delivery, and the primary key CRN which is underlined.

5. STUDENT

<u>ID</u>	First_Name	Fathers_Name	Last_Name	City	Street	Building
Email	Phone_Number	Birthday	Age	Path	Average	

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 First_Name, Fathers_Name, Last_Name 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, average and Path of the Student.

6. BUS

<u>Plate</u>	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.

7. UTILITY

<u>SN</u>	Name	State	Date_of_Purchase	Warranty
-----------	------	-------	------------------	----------

The UTILITY entity type contains simple attributes. Simple attributes are the primary key SN which is underlined in addition to name, state, date_of_purchase and warranty.

8. CLUB

<u>Name</u>	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.

9. SPORTS_TEAM

<u>Team_ID</u>	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

<u>StudentID</u>	<u>Name</u>	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.

2) EVENT

<u>Club_Name</u>	<u>Name</u>	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 Club_Name, the primary of the owner entity type CLUB, is included. Club_Name 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 on 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*)

<u>Plate</u>	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.

2. OFFICE(*Has*)

<u>Room_Number</u>	<u>Building_Name</u>	<u>Employee_ID</u>
--------------------	----------------------	--------------------

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.

3. CLUB (*Supervises*)

<u>Name</u>	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 should add 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

<u>ID</u>	First_Name	Fathers_Name	Last_Name	City	Street	Building
Email	Phone_Number	Birthday	Age	Path	Bus_Taken	Average

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.

2. Employee (*Belongs_to*)

<u>ID</u>	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.

3. Employee (*Supervises_1*)

<u>ID</u>	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 supervising the others knowing that ID is the primary key in the entity type EMPLOYEE.

4. UTILITY(*borrows*)

<u>SN</u>	Name	State	Date_of_purchase	Warranty	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.

5. UTILITY (*borrows_2*)

<u>SN</u>	Name	State	Date_of_purchase	Warranty	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

<u>StudentID</u>	<u>SportTeam_Name</u>
------------------	-----------------------

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 "StudentID" and "SportTeam_Name" respectively. The combination of both of these attributes constitutes the primary key of the "Plays_with" relation and they are underlined.

2. COACHES

<u>EmployeeID</u>	<u>SportTeam_Name</u>	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.

3. JOINS

<u>Student_ID</u>	<u>Club_Name</u>	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.

4. ENROLLS

<u>Student_Id</u>	<u>Course_CRN</u>
-------------------	-------------------

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.

5. TEACHES

<u>Employee_ID</u>	<u>Course_CRN</u>
--------------------	-------------------

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.

6. TAKES_PLACE

<u>Course_CRN</u>	<u>Room_Number</u>	<u>Building_Name</u>
-------------------	--------------------	----------------------

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 and building room is added to the “Takes_Place” relation and renamed Room_Name and Building_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

<u>CourseCRN</u>	<u>Days</u>
------------------	-------------

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

<u>ID</u>	First_Name	Fathers_Name	Last_Name	City	Street	Building
Age	Birthday	Phone_Number	Email	Role	Schedule	Name_of_department
		SupervisorID				

STUDENT

<u>ID</u>	First_Name	Fathers_Name	Last_Name	City	Street	Building
Email	Phone_Number	Birthday	Age	Path	Bus_Taken	Average

BUS

<u>Plate</u>	Number	Coverage_Area	Capacity	Color	Model	DriverID
--------------	--------	---------------	----------	-------	-------	----------

DEPARTMENT

<u>Name</u>	Location	Email	Phone_Number
-------------	----------	-------	--------------

UTILITY

<u>SN</u>	Name	State	Date_of_purchase	Warranty	Student_ID	Employee_ID
-----------	------	-------	------------------	----------	------------	-------------

CLUB

<u>Name</u>	Budget	Meeting_Time	Type	Email	SupervisorID
-------------	--------	--------------	------	-------	--------------

ROOM

<u>Building</u>	<u>Number</u>	Capacity	Type	Location
-----------------	---------------	----------	------	----------

COURSE

<u>CRN</u>	Name	Code	Nb_of_Credits	Time	Mode_of_Delivery
------------	------	------	---------------	------	------------------

SPORTS_TEAM

<u>Team_ID</u>	Name	Sport_Played	Location_of_Training	Time
----------------	------	--------------	----------------------	------

GUARDIAN

<u>Student_ID</u>	<u>Name</u>	Phone_Number	Email	Street	Building	City
-------------------	-------------	--------------	-------	--------	----------	------

EVENT

<u>ClubName</u>	<u>Name</u>	Date	Event_type
-----------------	-------------	------	------------

OFFICE

<u>Room Number</u>	<u>Building Name</u>	<u>Employee ID</u>
--------------------	----------------------	--------------------

PLAYS_WITH

<u>StudentID</u>	<u>SportTeam Name</u>
------------------	-----------------------

COACHES

<u>EmployeeID</u>	<u>SportTeam Name</u>	Role
-------------------	-----------------------	------

JOINS

<u>Student ID</u>	<u>Club Name</u>	Role
-------------------	------------------	------

ENROLLS_IN

<u>Student Id</u>	<u>Course CRN</u>
-------------------	-------------------

TEACHES

<u>Employee ID</u>	<u>Course CRN</u>
--------------------	-------------------

TAKES_PLACE

<u>Course CRN</u>	<u>Room Name</u>	<u>Building Name</u>
-------------------	------------------	----------------------

COURSEDAYS

<u>CourseCRN</u>	<u>Days</u>
------------------	-------------

VII- Table Structure for ARMY School:

After designing the ER diagram for ARMY school and mapping this diagram into relational database design, now it is time to start creating the actual tables for our database on the Oracle Database Server. We will start by creating all tables and then inserting data into these tables. Finally, we will execute some queries to display the importance of the database and especially in a school.

1. Department:

```
CREATE TABLE DEPARTMENT (  
  Name VARCHAR(20) NOT NULL,  
  Location VARCHAR(20) NOT NULL CHECK  
    (  
      Location IN('Beirut','Byblos')  
    ),  
  Email VARCHAR(20) NOT NULL CHECK (  
    Email LIKE '%@army.edu%'  
  ),  
  Phone_Number CHAR(8) NOT NULL,  
  PRIMARY KEY (Name)  
);
```

2. Employee:

```
CREATE TABLE EMPLOYEE (  
  ID CHAR(7) NOT NULL,  
  First_Name VARCHAR(15) NOT NULL,  
  Fathers_Name VARCHAR(15),  
  Last_Name VARCHAR(15) NOT NULL,  
  City VARCHAR(15) NOT NULL,  
  Street VARCHAR(15),  
  Building VARCHAR(15),  
  Birthday DATE,
```

```

Phone_Number VARCHAR(15),
Email VARCHAR(50) NOT NULL CHECK (
    Email LIKE '%@army.edu%'
),
Role VARCHAR(20),
Schedule VARCHAR(10) NOT NULL,
Name_of_Department VARCHAR(20) NOT NULL,
SupervisorID CHAR(7),
PRIMARY KEY (ID),
FOREIGN KEY (Name_of_Department) REFERENCES DEPARTMENT(Name),
FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE(ID)
);

```

3. Bus:

```

CREATE TABLE BUS (
Plate CHAR(6) PRIMARY KEY,
Bus_Number INT NOT NULL,
Coverage_Area VARCHAR(20),
Capacity INT NOT NULL CHECK(
    Capacity > 0
),
Color VARCHAR(20) NOT NULL,
Model VARCHAR(20) NOT NULL,
DriverID CHAR(7) NOT NULL,
FOREIGN KEY (DriverID) REFERENCES EMPLOYEE(ID),
UNIQUE (Bus_Number)
);

```

4. Student:

```

CREATE TABLE STUDENT(
ID CHAR(7) NOT NULL,
First_Name VARCHAR(15) NOT NULL,
Fathers_Name VARCHAR(15),
Last_Name VARCHAR(15) NOT NULL,

```

```

City VARCHAR(15) NOT NULL,
Street VARCHAR(15),
Building VARCHAR(15),
Birthday DATE,
Phone_Number VARCHAR(15),
Email VARCHAR(50) NOT NULL CHECK (
    Email LIKE '%@army.edu%'
),
Path CHAR(2) NOT NULL CHECK (
    Path IN('LS','GS','SE','HP')
),
Average FLOAT NOT NULL CHECK (
    Average>0 AND Average<=20
),
Bus_Taken CHAR(6),
FOREIGN KEY (Bus_Taken) REFERENCES BUS(Plate),
PRIMARY KEY (ID)
);

```

5. Club:

```

CREATE TABLE CLUB (
Name VARCHAR(30) PRIMARY KEY,
Budget INT NOT NULL CHECK (
    Budget >= 50 AND Budget <= 5000),
Meeting_Time TIMESTAMP WITH TIME ZONE NOT NULL,
Email VARCHAR(40) NOT NULL CHECK (
    Email LIKE '%@army.edu%'
),
SupervisorID CHAR(7) NOT NULL,
FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE(ID)
);

```

6. Room:

```
CREATE TABLE ROOM (  
  Building VARCHAR(30) NOT NULL,  
  RoomNumber INT NOT NULL,  
  Capacity INT NOT NULL CHECK(  
    Capacity > 0),  
  Type VARCHAR(40) NOT NULL CHECK (  
    Type IN('Class', 'Office')  
  ),  
  Location VARCHAR(30) NOT NULL CHECK (  
    Location IN('Beirut', 'Byblos')  
  ),  
  PRIMARY KEY(Building, RoomNumber)  
);
```

7. Course:

```
CREATE TABLE COURSE (  
  CRN CHAR(5) PRIMARY KEY,  
  Name VARCHAR(30) NOT NULL,  
  Code CHAR(5) NOT NULL,  
  Nbr_of_Credits INT NOT NULL CHECK  
    (  
      Nbr_of_Credits IN(1, 2, 3)  
    ),  
  Course_Time VARCHAR(5) CHECK (Course_Time  
    IN('8:00', '9:00', '10:00', '11:00', '12:00', '1:00', '2:00', '3:00')  
  ),  
  Mode_of_Delivery VARCHAR(15) NOT NULL  
);
```

8. Utility:

```
CREATE TABLE UTILITY (  
  SN VARCHAR(30) PRIMARY KEY,  
  Name VARCHAR(30) NOT NULL,  
  Date_of_Purchase DATE NOT NULL,  
  Warranty INT CHECK(  
    Warranty >= 0),  
  State VARCHAR(20) NOT NULL CHECK  
    (  
      state IN('Good condition','Needs Repair')  
    ),  
  Student_ID CHAR(7),  
  Employee_ID CHAR(7),  
  FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(ID),  
  FOREIGN KEY (Student_ID) REFERENCES STUDENT(ID)  
);
```

9. Sport Team:

```
CREATE TABLE SPORTS_TEAM (  
  Team_ID INT PRIMARY KEY,  
  Name VARCHAR(30) NOT NULL,  
  Sport_Played VARCHAR(20) NOT NULL,  
  Location_Of_Training VARCHAR(20),  
  Time_Of_Training VARCHAR(20)  
);
```

10. Event:

```
CREATE TABLE EVENT (  
  Event_Name VARCHAR(30) NOT NULL,  
  Event_Date DATE NOT NULL,  
  Event_Type VARCHAR(20) NOT NULL,
```

```
Club_Name VARCHAR(30) NOT NULL,
FOREIGN KEY (Club_Name) REFERENCES CLUB(Name),
PRIMARY KEY(Club_Name, Event_Name)
);
```

11. Guardian:

```
CREATE TABLE GUARDIAN (
Student_ID CHAR(7) NOT NULL,
Name VARCHAR(25) NOT NULL,
Phone_Number VARCHAR(15),
Email VARCHAR(50) NOT NULL,
City VARCHAR(20) NOT NULL,
Street VARCHAR(20),
Building VARCHAR(30),
FOREIGN KEY (Student_ID) REFERENCES STUDENT(ID),
PRIMARY KEY(Student_ID, Name)
);
```

12. Office:

```
CREATE TABLE OFFICE(
Room_Number INT,
Room_Building VARCHAR(30),
Employee_ID CHAR(7),
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(ID),
FOREIGN KEY (Room_Number, Room_Building) REFERENCES
ROOM(RoomNumber, Building),
PRIMARY KEY(Room_Number, Room_Building, Employee_ID)
);
```

13. Plays With:

```
CREATE TABLE PLAYS_WITH (
Student_ID CHAR(7),
Sport_team_ID INT,
```

```
FOREIGN KEY (Sport_team_ID) REFERENCES SPORTS_TEAM(Team_ID),
FOREIGN KEY (Student_ID) REFERENCES STUDENT(ID),
PRIMARY KEY (Student_ID, Sport_Team_ID)
);
```

14. Coaches:

```
CREATE TABLE COACHES(
Employee_ID CHAR(7) NOT NULL,
Sport_Team_ID INT NOT NULL,
Role VARCHAR(20) NOT NULL,
FOREIGN KEY (Sport_Team_ID) REFERENCES SPORTS_TEAM(Team_ID),
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(ID),
PRIMARY KEY (Employee_ID, Sport_Team_ID)
);
```

15. Joins:

```
CREATE TABLE JOINS (
Student_ID CHAR(7) NOT NULL,
Club_Name VARCHAR(20) NOT NULL,
Role VARCHAR(30) NOT NULL,
FOREIGN KEY (Club_Name) REFERENCES CLUB(Name),
FOREIGN KEY (Student_ID) REFERENCES STUDENT(ID),
PRIMARY KEY (Club_Name, Student_ID)
);
```

16. Enrolls-In:

```
CREATE TABLE ENROLLS_IN(
Student_ID CHAR(7) NOT NULL,
Course_CRN CHAR(5) NOT NULL,
FOREIGN KEY (Student_ID) REFERENCES STUDENT(ID),
FOREIGN KEY (Course_CRN) REFERENCES COURSE(CRN),
PRIMARY KEY (Course_CRN, Student_ID)
);
```


17. Teaches:

```
CREATE TABLE TEACHES(  
  Course_CRN CHAR(5),  
  Employee_ID CHAR(7),  
  FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(ID),  
  FOREIGN KEY (Course_CRN) REFERENCES COURSE(CRN),  
  PRIMARY KEY(Course_CRN, Employee_ID)  
);
```

18. Takes Place:

```
CREATE TABLE TAKES_PLACE(  
  Course_CRN CHAR(5) NOT NULL,  
  Room_Number INT NOT NULL,  
  Room_Building VARCHAR(30) NOT NULL,  
  FOREIGN KEY (Course_CRN) REFERENCES COURSE(CRN),  
  FOREIGN KEY (Room_Number, Room_Building) REFERENCES  
  ROOM(RoomNumber, Building),  
  PRIMARY KEY(Course_CRN, Room_Number, Room_Building)  
);
```

19. Course Days:

```
CREATE TABLE COURSE_DAYS (  
  Course_CRN CHAR(5) NOT NULL,  
  Days VARCHAR(10) NOT NULL,  
  FOREIGN KEY (Course_CRN) REFERENCES COURSE(CRN),  
  PRIMARY KEY(Course_CRN, Days)  
);
```

VIII- Table Descriptions:

After creating the tables on the oracle server we can view the data entered in each table to make sure all the entries were entered correctly. In our database we have the following tables and we used this command to get display the results:

1. Department Description:

DESC DEPARTMENT;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENT	NAME	VARCHAR2	20	-	-	1	-	-	-
	LOCATION	VARCHAR2	20	-	-	-	-	-	-
	EMAIL	VARCHAR2	20	-	-	-	-	-	-
	PHONE_NUMBER	CHAR	8	-	-	-	-	-	-

2. Employee Description:

DESC EMPLOYEE;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	ID	CHAR	7	-	-	1	-	-	-
	FIRST_NAME	VARCHAR2	15	-	-	-	-	-	-
	FATHERS_NAME	VARCHAR2	15	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	15	-	-	-	-	-	-
	CITY	VARCHAR2	15	-	-	-	-	-	-
	STREET	VARCHAR2	15	-	-	-	✓	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	BIRTHDAY	DATE	7	-	-	-	✓	-	-
	PHONE_NUMBER	VARCHAR2	15	-	-	-	✓	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	ROLE	VARCHAR2	20	-	-	-	✓	-	-
	SCHEDULE	VARCHAR2	10	-	-	-	-	-	-
	NAME_OF_DEPARTMENT	VARCHAR2	20	-	-	-	-	-	-
	SUPERVISORID	CHAR	7	-	-	-	✓	-	-

3. Student Description:

DESC STUDENT;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT	ID	CHAR	7	-	-	1	-	-	-
	FIRST_NAME	VARCHAR2	15	-	-	-	-	-	-
	FATHERS_NAME	VARCHAR2	15	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	15	-	-	-	-	-	-
	CITY	VARCHAR2	15	-	-	-	-	-	-
	STREET	VARCHAR2	15	-	-	-	✓	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	BIRTHDAY	DATE	7	-	-	-	✓	-	-
	PHONE_NUMBER	VARCHAR2	15	-	-	-	✓	-	-
	EMAIL	VARCHAR2	50	-	-	-	-	-	-
	PATH	CHAR	2	-	-	-	-	-	-
	AVERAGE	FLOAT	126	126	-	-	-	-	-
	BUS_TAKEN	CHAR	6	-	-	-	✓	-	-

4. Utility Description:

DESC UTILITY;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
UTILITY	SN	VARCHAR2	30	-	-	1	-	-	-
	NAME	VARCHAR2	30	-	-	-	-	-	-
	DATE_OF_PURCHASE	DATE	7	-	-	-	-	-	-
	WARRANTY	NUMBER	22	-	0	-	✓	-	-
	STATE	VARCHAR2	20	-	-	-	-	-	-
	STUDENT_ID	CHAR	7	-	-	-	✓	-	-
	EMPLOYEE_ID	CHAR	7	-	-	-	✓	-	-

5. Club Description:

DESC CLUB;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CLUB	NAME	VARCHAR2	30	-	-	1	-	-	-
	BUDGET	NUMBER	22	-	0	-	-	-	-
	MEETING_TIME	TIMESTAMP(6) WITH TIME ZONE	13	-	6	-	-	-	-
	EMAIL	VARCHAR2	40	-	-	-	-	-	-
	SUPERVISORID	CHAR	7	-	-	-	-	-	-

6. Sports-Team Description:

DESC SPORTS_TEAM;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SPORTS_TEAM	TEAM_ID	NUMBER	22	-	0	1	-	-	-
	NAME	VARCHAR2	30	-	-	-	-	-	-
	SPORT_PLAYED	VARCHAR2	20	-	-	-	-	-	-
	LOCATION_OF_TRAINING	VARCHAR2	20	-	-	-	✓	-	-
	TIME_OF_TRAINING	VARCHAR2	20	-	-	-	✓	-	-

7. Event Description:

DESC EVENT;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EVENT	EVENT_NAME	VARCHAR2	30	-	-	2	-	-	-
	EVENT_DATE	DATE	7	-	-	-	-	-	-
	EVENT_TYPE	VARCHAR2	20	-	-	-	-	-	-
	CLUB_NAME	VARCHAR2	30	-	-	1	-	-	-

8. Plays-with Description:

DESC PLAYS_WITH;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PLAYS_WITH	STUDENT_ID	CHAR	7	-	-	1	-	-	-
	SPORT_TEAM_ID	NUMBER	22	-	0	2	-	-	-

9. Enrolls-in Description:

DESC ENROLLS_IN;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ENROLLS_IN	STUDENT_ID	CHAR	7	-	-	2	-	-	-
	COURSE_CRN	CHAR	5	-	-	1	-	-	-

10. Takes-Place Description:

DESC TAKES_PLACE;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TAKES_PLACE	COURSE_CRN	CHAR	5	-	-	1	-	-	-
	ROOM_NUMBER	NUMBER	22	-	0	2	-	-	-
	ROOM_BUILDING	VARCHAR2	30	-	-	3	-	-	-

11. Bus Description:

DESC BUS;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BUS	PLATE	CHAR	6	-	-	1	-	-	-
	BUS_NUMBER	NUMBER	22	-	0	-	-	-	-
	COVERAGE_AREA	VARCHAR2	20	-	-	-	✓	-	-
	CAPACITY	NUMBER	22	-	0	-	-	-	-
	COLOR	VARCHAR2	20	-	-	-	-	-	-
	MODEL	VARCHAR2	20	-	-	-	-	-	-
	DRIVERID	CHAR	7	-	-	-	-	-	-

12. Course Description:

DESC COURSE;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COURSE	CRN	CHAR	5	-	-	1	-	-	-
	NAME	VARCHAR2	30	-	-	-	-	-	-
	CODE	CHAR	5	-	-	-	-	-	-
	NBR_OF_CREDITS	NUMBER	22	-	0	-	-	-	-
	COURSE_TIME	VARCHAR2	5	-	-	-	✓	-	-
	MODE_OF_DELIVERY	VARCHAR2	15	-	-	-	-	-	-

13. Coaches Description:

DESC COACHES;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COACHES	EMPLOYEE_ID	CHAR	7	-	-	1	-	-	-
	SPORT_TEAM_ID	NUMBER	22	-	0	2	-	-	-
	ROLE	VARCHAR2	20	-	-	-	-	-	-

14. Room Description:

DESC ROOM;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROOM	BUILDING	VARCHAR2	30	-	-	1	-	-	-
	ROOMNUMBER	NUMBER	22	-	0	2	-	-	-
	CAPACITY	NUMBER	22	-	0	-	-	-	-
	TYPE	VARCHAR2	40	-	-	-	-	-	-
	LOCATION	VARCHAR2	30	-	-	-	-	-	-

15. Office Description:

DESC OFFICE;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OFFICE	ROOM_NUMBER	NUMBER	22	-	0	1	-	-	-
	ROOM_BUILDING	VARCHAR2	30	-	-	2	-	-	-
	EMPLOYEE_ID	CHAR	7	-	-	3	-	-	-

16. Course-Days Description:

DESC COURSE_DAYS;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COURSE_DAYS	COURSE_CRN	CHAR	5	-	-	1	-	-	-
	DAYS	VARCHAR2	10	-	-	2	-	-	-

17. Teaches Description:

DESC TEACHES;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TEACHES	COURSE_CRN	CHAR	5	-	-	1	-	-	-
	EMPLOYEE_ID	CHAR	7	-	-	2	-	-	-

18. Guardian Description:

DESC GUARDIAN;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>GUARDIAN</u>	<u>STUDENT_ID</u>	CHAR	7	-	-	1	-	-	-
	<u>NAME</u>	VARCHAR2	25	-	-	2	-	-	-
	<u>PHONE_NUMBER</u>	VARCHAR2	15	-	-	-	✓	-	-
	<u>EMAIL</u>	VARCHAR2	50	-	-	-	-	-	-
	<u>CITY</u>	VARCHAR2	20	-	-	-	-	-	-
	<u>STREET</u>	VARCHAR2	20	-	-	-	✓	-	-
	<u>BUILDING</u>	VARCHAR2	30	-	-	-	✓	-	-

19. Joins Description:

DESC JOINS;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>JOINS</u>	<u>STUDENT_ID</u>	CHAR	7	-	-	2	-	-	-
	<u>CLUB_NAME</u>	VARCHAR2	20	-	-	1	-	-	-
	<u>ROLE</u>	VARCHAR2	30	-	-	-	-	-	-

IX- Inserting Data:

1. Department:

```
INSERT INTO DEPARTMENT
VALUES('Financial','Beirut','fn_dpirt@army.edu','01111111');

INSERT INTO DEPARTMENT VALUES('Special
Education','Beirut','spedu_dpirt@army.edu','01111112');

INSERT INTO DEPARTMENT
VALUES('Science','Byblos','sc_dpirt@army.edu','01111113');

INSERT INTO DEPARTMENT VALUES ('Literature &
Culture','Byblos','ltC_dpirt@army.edu','01111114');

INSERT INTO DEPARTMENT
VALUES('Transportation','Beirut','art_dpirt@army.edu','01111115');

INSERT INTO DEPARTMENT VALUES ('Health &
Safety','Beirut','hlthS_dpirt@army.edu','01111116');

INSERT INTO DEPARTMENT
VALUES('Sport','Byblos','sport_dpirt@army.edu','01111117');

INSERT INTO DEPARTMENT
VALUES('Admission','Beirut','adm_dpirt@army.edu','01111118');

INSERT INTO DEPARTMENT VALUES ('Human
Resources','Byblos','hr_dpirt@army.edu','01111119');

INSERT INTO DEPARTMENT
VALUES('IT','Byblos','it_dpirt@army.edu','01111120');
```

2. Employee:

```
INSERT INTO EMPLOYEE
VALUES('I202129','Elie','Joseph','Wehbe','Beirut','445','Grand',DATE
'1998-10-12','03376772','elie.wehbe@army.edu','Head Instructor','M--
>F','Science',NULL);

INSERT INTO EMPLOYEE
VALUES('I202119','Mostafa','Ali','Marsi','Beirut','963','Solh',DATE
'2001-1-30','70981236','mostafa.masri@army.edu','Manager','M--
>F','Human Resources',NULL);
```



```
INSERT INTO EMPLOYEE
VALUES('I202109','Sary','Sleiman','Fahs','Jounieh','Thirteenth','Gha
lyini',DATE'1996-7-7','76111120','sary.fahs@army.edu','Bus
Driver','M->F','Transportation',NULL);
```

```
INSERT INTO EMPLOYEE
VALUES('I202100','John','Mark','Khoury','Beirut','First','Safadi',DA
TE'1992-9-11','76111111','john.khoury@army.edu','Bus Driver','M-
>F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202101','Marie','Bassam','Ismail','Byblos','Second','Salman
',DATE'2000-2-20','76111112','marie.ismail@army.edu','Bus
Driver','M->F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202102','Yara','Amin','Harb','Tyre','Fifteenth','Solh',DATE
'1990-10-1','76111113','yara.harb@army.edu','Bus Driver','M-
>F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202103','Jad','Salim','Mayyas','Sidon','Eighth','Khalifeh',D
ATE'1992-12-5','76111114','jad.mayyas@army.edu','Bus Driver','M-
>F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE VALUES('I202104','Sophie','Jean-
Paul','Rahal','Batroun','Third','Harb',DATE'1994-5-
6','76111115','sophie.rahal@army.edu','Bus Driver','M-
>F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202105','Omar','Moustafa','Salman','Tripoli','First','Khala
f',DATE'1992-9-19','76111116','omar.salman@army.edu','Bus
Driver','M->F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202106','Mira','Jamil','Hafi','Aley','Tenth','Abedel
Baset',DATE'1995-3-4','76111117','mira.hafi@army.edu','Bus
Driver','M->F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202107','Mohamad','Kairm','Itani','Chouf','Eleventh','Zein'
,DATE'1997-3-30','76111118','mohamad.itani@army.edu','Bus
Driver','M->F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202108','George','Mark','Sidani','Zahle','Seventeenth','Ita
ni',DATE'1999-8-28','76111119','george.sidani@army.edu','Bus
Driver','M->F','Transportation','I202109');
```

```
INSERT INTO EMPLOYEE
VALUES('I202114','Omar','Ali','Itani','Byblos','758','Zeinab',DATE'1
995-7-22','70589632','omar.itani@army.edu','Head Coach','M--
>F','Sport','I202119');
```

```
INSERT INTO EMPLOYEE
VALUES('I202115','Jad','Zahi','Aridi','Byblos','164','Farah',DATE'19
83-3-3','70152369','jad.aridi@army.edu','TA','M--
>F','Sport','I202114');
```

```
INSERT INTO EMPLOYEE
VALUES('I202110','Rani','Slaiman','Salman','Beirut','785','Khawli',D
ATE'1990-9-5','70830831','rani.salman@army.edu','Coach','M--
>F','Sport','I202114');
```

```
INSERT INTO EMPLOYEE
VALUES('I202111','Dani','Mazen','Breich','Beirut','242','Ali',DATE'1
991-11-13','70963758','dani.breich@army.edu','Coach','M--
>F','Sport','I202114');
```

```
INSERT INTO EMPLOYEE
VALUES('I202112','Mazen','Mohammad','Chatila','Beirut','753','Sky',D
ATE'1990-4-1','70159863','mazen.chatila@army.edu','Coach','M--
>F','Sport','I202114');
```

```
INSERT INTO EMPLOYEE
VALUES('I202113','Hadi','Ramzi','Harb','Beirut','896','Glow',DATE'19
93-5-21','70145489','hadi.harb@army.edu','Coach','M--
>F','Sport','I202114');
```

```
INSERT INTO EMPLOYEE
VALUES('I202116','Jad','Zouhair','Salman','Byblos','854','Anouti',DA
TE'1976-8-5','70845963','jad.salman@army.edu','TA','M--
>F','Science','I202129');
```

```
INSERT INTO EMPLOYEE
VALUES('I202117','Morwan','Maziad','AbouKarroum','Byblos','148','Kha
life',DATE'1956-3-
2','70145289','morwan.aboukarroum@army.edu','Accountant','M--
>F','Financial','I202119');
```

```
INSERT INTO EMPLOYEE
VALUES('I202118','Nidal','Adel','ElAchkar','Byblos','741','Harb',DAT
E'1998-2-6','70481592','nidal.elachkar@army.edu','Secretary','M--
>F','Financial','I202119');
```

```
INSERT INTO EMPLOYEE
VALUES('I202120','Fadi','Joseph','Sarkis','Beirut','78','Sky',DATE'1
990-8-8','70514233','fadi.sarkis@army.edu','Instructor','M--
>F','Literature & Culture','I202129');
```

```
INSERT INTO EMPLOYEE
VALUES('I202121','Fadi','Marc','Costa','Beirut','774','Anouti',DATE'
1997-10-10','70448559','fadi.costa@army.edu','Instructor','M--
>F','Science','I202129');
```

```
INSERT INTO EMPLOYEE
VALUES('I202122','Joseph','Charbel','Maksoud','Byblos','778','Tall',
DATE'2001-5-6','71596324','Joseph.maksoud@army.edu','Instructor','M-
->F','Literature & Culture','I202129');
```

```
INSERT INTO EMPLOYEE
VALUES('I202123','Mazen','Habib','Karake','Beirut','159','Plaza',DAT
E'2001-7-8','03326598','mazen.karake@army.edu','Instructor','M--
>F','Human Resources','I202129');
```

3. Bus:

```
INSERT INTO BUS VALUES('B11440','1','Beirut', 30, 'White','Toyota',
'I202100');
```

```
INSERT INTO BUS VALUES('T11441','2','Beirut', 20,
'White','Toyota','I202101');
```

```
INSERT INTO BUS VALUES('S11442','3','Beirut', 35,
'Grey','Toyota','I202102');
```

```
INSERT INTO BUS VALUES('J11443','4','Beirut', 25,
'White','Toyota','I202103');
```

```
INSERT INTO BUS VALUES('B14444','5','Beirut', 25,
'White','Toyota','I202104');
```

```
INSERT INTO BUS VALUES('D11445','6','Byblos', 30,
'Grey','Toyota','I202105');
```

```
INSERT INTO BUS VALUES('S11446','7','Byblos', 35,
'White','Zonda','I202106');
```

```
INSERT INTO BUS VALUES('C11447','8','Byblos', 25,
'Grey','Zonda','I202107');
```

```
INSERT INTO BUS VALUES('Z11448','9','Byblos', 20,  
'White','Zonda','I202108');
```

```
INSERT INTO BUS VALUES('Z11449','10','Byblos', 25,  
'Grey','Zonda','I202109');
```

4. Student:

```
INSERT INTO STUDENT  
VALUES('S202100','Salim','Slaiman','Hammoud','Beirut','Ras el  
Ain','Farah',DATE'2004-12-  
5','03918111','salim.hammoud@army.edu','GS',8,'B11440');
```

```
INSERT INTO STUDENT  
VALUES('S202101','Farah','Mohamad','Itani','Byblos','Omar el  
Zeinnie','Abedel khalek',DATE'2010-9-  
22','03918112','farah.itani@army.edu','LS',12,'Z11449');
```

```
INSERT INTO STUDENT  
VALUES('S202102','Mohamad','Bassam','Makknieh','Zahle','Zouk  
II','Walid',DATE'2006-5-  
19','03918113','mohamad.makknieh@army.edu','SE',18,NULL);
```

```
INSERT INTO STUDENT  
VALUES('S202103','Sara','Hassan','Harb','Jounieh','Second  
22','Khawli',DATE'2011-6-  
13','03918114','sara.harb@army.edu','HP',14,NULL);
```

```
INSERT INTO STUDENT  
VALUES('S202104','Ibrahim','Abbas','Sleit','Sidon','Aisha bakar  
23','Itani I',DATE'2009-4-  
6','03918115','ibrahim.sleit@army.edu','GS',17,NULL);
```

```
INSERT INTO STUDENT  
VALUES('S202105','Yara','Ali','Ghalayini','Beirut','Ashrafieh  
10','Sleit',DATE'2015-3-  
10','03918116','yara.ghalayini@army.edu','LS',19,'B14444');
```

```
INSERT INTO STUDENT  
VALUES('S202106','Elias','George','Khoury','Batroun','Zouks  
34','Ghalayini',DATE'2005-7-  
18','03918117','elias.khoury@army.edu','SE',2,NULL);
```

```
INSERT INTO STUDENT  
VALUES('S202107','Rana','Farhan','Salman','Beirut','Kamil  
34','Abbas',DATE'2007-10-  
30','03918118','rana.salman@army.edu','HP',15,'S11442');
```

```
INSERT INTO STUDENT
VALUES('S202108','Assem','Ahmad','Farhat','Aley','Michel','Shine',DATE'2008-2-21','03918119','assem.farhat@army.edu','GS',17.8,NULL);
```

```
INSERT INTO STUDENT
VALUES('S202109','Roula','Jad','Attar','Byblos','Third12','Mira',DATE'2003-1-1','03918120','roula.attar@army.edu','LS',19.8,'C11447');
```

```
INSERT INTO STUDENT
VALUES('S202110','Roy','Mark','Solh','Beirut','Kouraytem111','Mark B',DATE'2004-9-11','71131211','roy.solh@army.edu','LS',19,'B11440');
```

```
INSERT INTO STUDENT
VALUES('S202111','Tony','Hisham','Zaiter','Baalbek','Hamra111','Hisham B',DATE'2004-9-11','78131211','toni.zaiter@army.edu','HP',18.7,NULL);
```

```
INSERT INTO STUDENT
VALUES('S202112','Charbel','Fadi','Naim','Zahle','BBB313','Fadi B',DATE'2003-9-11','76134511','charbel.naim@army.edu','GS',15,NULL);
```

```
INSERT INTO STUDENT
VALUES('S202113','Elias','Loay','Yaghi','Sidon','Souk12','Safadi',DATE'2003-9-11','76145611','elias.yaghi@army.edu','LS',13,NULL);
```

```
INSERT INTO STUDENT
VALUES('S202114','Omar','Ali','Shayah','Beirut','Ashrafeih21','DAWAL Kamar',DATE'2002-9-11','70131211','omar.shayah@army.edu','SE',16.5,'B11440');
```

```
INSERT INTO STUDENT
VALUES('S202115','Bakr','Omar','Sharif','Beirut','First11','Omar BB',DATE'2001-9-11','03131211','bakr.sharif@army.edu','GS',14.5,'J11443');
```

```
INSERT INTO STUDENT
VALUES('S202116','Haidar','Ali','Tarhine','Beirut','Center','AliB',DATE'2000-9-11','71112311','haidar.tarhine@army.edu','GS',3,'B14444');
```

```
INSERT INTO STUDENT
VALUES('S202117','Waleed','Othman','Khoury','Tyre','BCC123','AKSOB',DATE'2002-9-11','76131310','waleed.khoury@army.edu','LS',17,NULL);
```

```
INSERT INTO STUDENT
VALUES('S202118','Rana','Mohammad','Harb','Byblos','BYB1234','Mohamm
```

```
ad B',DATE'2001-9-11', '76111222', 'rana.harb@army.edu', 'SE',8, 'C11447');
```

```
INSERT INTO STUDENT  
VALUES('S202119', 'Rani', 'Salman', 'Salem', 'Sawfar', 'Old st21', 'Wydad  
B',DATE'2004-9-11', '71913710', 'rani.salem@army.edu', 'HP',18,NULL);
```

5. Club:

```
INSERT INTO CLUB VALUES('Math Club',200,TIMESTAMP'2021-09-17  
13:00:00', 'MathClub@army.edu.lb', 'I202110');
```

```
INSERT INTO CLUB VALUES('Physics Club',300,TIMESTAMP'2021-09-18  
11:00:00', 'PhyisecsClub@army.edu.lb', 'I202111');
```

```
INSERT INTO CLUB VALUES('Science Club',400,TIMESTAMP'2021-10-17  
12:00:00', 'ScienceClub@army.edu.lb', 'I202112');
```

```
INSERT INTO CLUB VALUES('Chemistry Club',1000,TIMESTAMP'2021-09-11  
9:00:00', 'ChemistryClub@army.edu.lb', 'I202113');
```

```
INSERT INTO CLUB VALUES('Arabic Club',200,TIMESTAMP'2021-12-17  
14:00:00', 'ArabicClub@army.edu.lb', 'I202114');
```

```
INSERT INTO CLUB VALUES('English Club',800,TIMESTAMP'2021-09-19  
13:00:00', 'EnglishClub@army.edu.lb', 'I202115');
```

```
INSERT INTO CLUB VALUES('Sports Club',200,TIMESTAMP'2021-09-17  
16:00:00', 'SportsClub@army.edu.lb', 'I202116');
```

```
INSERT INTO CLUB VALUES('Music Club',600,TIMESTAMP'2021-09-14  
13:00:00', 'MusicClub@army.edu.lb', 'I202117');
```

```
INSERT INTO CLUB VALUES('Animal Care Club',700,TIMESTAMP'2021-09-17  
14:30:00', 'AnimalCareClub@army.edu.lb', 'I202118');
```

```
INSERT INTO CLUB VALUES('Computer Club',2500,TIMESTAMP'2021-09-17  
13:30:00', 'ComputerClub@army.edu.lb', 'I202119');
```

6. Room:

```
INSERT INTO ROOM VALUES('Kinderhaus',12,30,'Class','Beirut');
INSERT INTO ROOM VALUES('Kinderhaus',13,5,'Office','Beirut');
INSERT INTO ROOM VALUES('Kinderhaus',14,5,'Office','Beirut');
INSERT INTO ROOM VALUES('Kinderhaus',15,5,'Office','Beirut');
INSERT INTO ROOM VALUES('Kinderhaus',16,30,'Class','Beirut');
INSERT INTO ROOM VALUES('Sage',22,5,'Office','Beirut');
INSERT INTO ROOM VALUES('Sage',23,5,'Office','Beirut');
INSERT INTO ROOM VALUES('Sage',24,30,'Class','Beirut');
INSERT INTO ROOM VALUES('Sage',25,30,'Class','Beirut');
INSERT INTO ROOM VALUES('Sage',26,30,'Class','Beirut');
INSERT INTO ROOM VALUES('Nicole',32,5,'Office','Byblos');
INSERT INTO ROOM VALUES('Nicole',33,5,'Office','Byblos');
INSERT INTO ROOM VALUES('Nicole',34,5,'Office','Byblos');
INSERT INTO ROOM VALUES('Nicole',35,30,'Class','Byblos');
INSERT INTO ROOM VALUES('Nicole',36,30,'Class','Byblos');
INSERT INTO ROOM VALUES('OG',42,30,'Class','Byblos');
INSERT INTO ROOM VALUES('OG',43,5,'Office','Byblos');
INSERT INTO ROOM VALUES('OG',44,5,'Office','Byblos');
INSERT INTO ROOM VALUES('OG',45,30,'Class','Byblos');
INSERT INTO ROOM VALUES('OG',46,30,'Class','Byblos');
```

7. Course:

```
INSERT INTO COURSE VALUES ('12345','Math
1','MTH20',3,'8:00','Online');

INSERT INTO COURSE
VALUES('12300','Science','SC90',3,'9:00','Online');

INSERT INTO COURSE VALUES('12045','Math
1','MTH20',3,'10:00','Online');
INSERT INTO COURSE
VALUES('02345','Physics','PHY05',3,'2:00','Hybrid');
```

```
INSERT INTO COURSE VALUES('12305','Math  
2','MTH40',3,'3:00','Hybrid');
```

```
INSERT INTO COURSE  
VALUES('00345','Economics','EC056',3,'10:00','Campus');
```

```
INSERT INTO COURSE  
VALUES('07345','Geometry','MTH21',1,'11:00','Campus');
```

```
INSERT INTO COURSE VALUES('10344','Paint','PA00',3,'2:00','Campus');
```

```
INSERT INTO COURSE  
VALUES('67842','Arabic','ARA50',3,'1:00','Campus');
```

```
INSERT INTO COURSE  
VALUES('10333','Chemistry','CHM55',3,'12:00','Campus');
```

8. Utility:

```
INSERT INTO UTILITY VALUES('FSD6789','Laptop',DATE'2020-12-  
12',2,'Good condition',NULL,'I202102');
```

```
INSERT INTO UTILITY VALUES('S45678','Mouse',DATE'2021-7-5',1,'Good  
condition',NULL,NULL);
```

```
INSERT INTO UTILITY VALUES('FS56F78S','Calculator',DATE'1995-5-  
15',0,'Needs Repair','S202101',NULL);
```

```
INSERT INTO UTILITY VALUES('FS5678SF67','Book: History of  
Lebanon',DATE'1985-1-1',0,'Good condition','S202109',NULL);  
INSERT INTO UTILITY VALUES('SF5S6FS6','Printer',DATE'2019-7-  
1',5,'Needs Repair',NULL,NULL);
```

```
INSERT INTO UTILITY VALUES('ILUYI389L','Markers',DATE'2021-2-  
15',0,'Good condition',NULL,'I202115');
```

```
INSERT INTO UTILITY VALUES('LU7Y5LY','Projector',DATE'2018-11-  
15',3,'Good condition',NULL,'I202106');
```

```
INSERT INTO UTILITY VALUES('CVN5C79','Monitor',DATE'2018-11-  
15',5,'Good condition','S202105',NULL);
```

```
INSERT INTO UTILITY VALUES('CN45CN8CV','Speaker',DATE'2015-5-  
4',5,'Good condition',NULL,'I202108');
```

```
INSERT INTO UTILITY VALUES('4YE5R6','Keyboard',DATE'2021-1-  
18',1,'Good condition',NULL,NULL);
```


9. Sport Team:

```
INSERT INTO SPORTS_TEAM VALUES(1, 'Wizards (Varsity)', 'Basketball',  
'Basketball court', 'Wednesday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(2, 'Warriors (Junior Varsity)',  
'Basketball', 'Basketball court', 'Monday 5:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(3, 'Mystics', 'Volley Ball',  
'Basketball court', 'Tuesday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(4, 'Stars (Varsity)', 'Football',  
'Football field', 'Wednesday 5:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(5, 'Suns (Junior Varsity)',  
'Football', 'Football field', 'Friday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(6, 'Blitz', 'Ultimate Frisbee',  
'Football field', 'Monday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(7, 'Rebels', 'Rugby', 'Football  
field', 'Tuesday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(8, 'Sweet Shots', 'Tennis', 'Tennis  
court', 'Tuesday 4:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(9, 'Gladiators', 'Wrestling', 'Indoor  
Gymnasium', 'Thursday 5:00 pm');
```

```
INSERT INTO SPORTS_TEAM VALUES(10, 'Fireballs', 'Hand ball',  
'Basketball court', 'Friday 5:00 pm');
```

10. Event:

```
INSERT INTO EVENT VALUES('Math Contest', DATE'2021-10-  
22', 'Competition', 'Math Club');
```

```
INSERT INTO EVENT VALUES('Math Contest', DATE'2021-10-  
22', 'Competition', 'Physics Club');
```

```
INSERT INTO EVENT VALUES('Intro to Space', DATE'2021-5-  
14', 'Educational', 'Physics Club');
```

```

INSERT INTO EVENT VALUES('Science Fair',DATE'2021-11-25','Educational','Science Club');

INSERT INTO EVENT VALUES('Group Experiments',DATE'2021-12-25','Educational','Chemistry Club');

INSERT INTO EVENT VALUES('Poerty Night',DATE'2021-12-14','Entertainment','Arabic Club');

INSERT INTO EVENT VALUES('Poerty Night',DATE'2021-12-14','Entertainment','English Club');

INSERT INTO EVENT VALUES('Group Reading',DATE'2021-9-19','Entertainment','English Club');

INSERT INTO EVENT VALUES('Football tournament', DATE'2022-1-10','Competition','Sports Club');

INSERT INTO EVENT VALUES('Concert Night', DATE'2022-12-8','Entertainment','Music Club');

INSERT INTO EVENT VALUES('Seminar on wild life', DATE'2022-10-9','Educational','Animal Care Club');

INSERT INTO EVENT VALUES('Hackathon', DATE'2021-10-19','Competition','Computer Club');

```

11. Guardian:

```

INSERT INTO GUARDIAN VALUES('S202101','Slaiman Hammoud','81456123','slaiman.hammoud@outlook.com','Beirut','654','Farah');

INSERT INTO GUARDIAN VALUES('S202102','Mohamad Itani','76985123','mohamad.itani@outlook.com','Jbeil','369','Abedel khalek');

INSERT INTO GUARDIAN VALUES('S202103','Bassam makknieh','81475555','bassam.makknieh@outlook.com','Zahle','258','Wazlid');

INSERT INTO GUARDIAN VALUES('S202104','Hassan Harb','71830831','hassan.harb@outlook.com','Jounieh','147','Khawli')
;

```

```
INSERT INTO GUARDIAN VALUES('S202105', 'Abbas Sleit', '03652148', 'abbas.sleit@outlook.com', 'Sidon', '825', 'Itani I');
```

```
INSERT INTO GUARDIAN VALUES('S202110', 'Mark Solh', '71717109', 'marksolh12@gmail.com', 'Beirut', 'Kouraytem111', 'Mark B');
```

```
INSERT INTO GUARDIAN VALUES('S202111', 'Hisham Zaiter', '71231211', 'hisham.zaiter@gmail.com', 'Baalbek', 'Nahle 111', 'Hisham B');
```

```
INSERT INTO GUARDIAN VALUES('S202112', 'Fadi Naim', '03134511', 'fadi.naim@gmail.com', 'Zahle', 'BBB313', 'Fadi B');
```

```
INSERT INTO GUARDIAN VALUES('S202113', 'Loay Yaghi', '78145611', 'loay.yaghi@gmail.com', 'Sidon', 'Souk 12', 'Safadi');
```

```
INSERT INTO GUARDIAN VALUES('S202114', 'Ali Shayah', '70111211', 'ali.shayah@gmail.com', 'Beirut', 'Ashrafeih21', 'DAW AL Kamar');
```

12. Office:

```
INSERT INTO OFFICE VALUES(13, 'Kinderhaus', 'I202129');
```

```
INSERT INTO OFFICE VALUES(14, 'Kinderhaus', 'I202123');
```

```
INSERT INTO OFFICE VALUES(15, 'Kinderhaus', 'I202122');
```

```
INSERT INTO OFFICE VALUES(22, 'Sage', 'I202121');
```

```
INSERT INTO OFFICE VALUES(23, 'Sage', 'I202120');
```

```
INSERT INTO OFFICE VALUES(32, 'Nicole', 'I202117');
```

```
INSERT INTO OFFICE VALUES(33, 'Nicole', 'I202119');
```

```
INSERT INTO OFFICE VALUES(34, 'Nicole', 'I202115');
```

```
INSERT INTO OFFICE VALUES(43, 'OG', 'I202110');
```

```
INSERT INTO OFFICE VALUES(44, 'OG', 'I202116');
```

13. Plays-With:

```
INSERT INTO PLAYS_WITH VALUES('S202110', 1);
```

```
INSERT INTO PLAYS_WITH VALUES('S202111', 10);
```

```

INSERT INTO PLAYS_WITH VALUES('S202112',2);
INSERT INTO PLAYS_WITH VALUES('S202113',3);
INSERT INTO PLAYS_WITH VALUES('S202114',4);
INSERT INTO PLAYS_WITH VALUES('S202115',5);
INSERT INTO PLAYS_WITH VALUES('S202116',6);
INSERT INTO PLAYS_WITH VALUES('S202117',7);
INSERT INTO PLAYS_WITH VALUES('S202118',8);
INSERT INTO PLAYS_WITH VALUES('S202119',9);

```

14. Coaches:

```

INSERT INTO COACHES VALUES('I202119',1,'Trainer');
INSERT INTO COACHES VALUES('I202118',2,'Trainer Assistant');
INSERT INTO COACHES VALUES('I202117',3,'Flexibility Trainer');
INSERT INTO COACHES VALUES('I202116',4,'Coach Assistant');
INSERT INTO COACHES VALUES('I202115',5,'Coach');
INSERT INTO COACHES VALUES('I202114',6,'captain');
INSERT INTO COACHES VALUES('I202113',7,'Trainer');
INSERT INTO COACHES VALUES('I202112',8,'Manager');
INSERT INTO COACHES VALUES('I202111',9,'Coach');
INSERT INTO COACHES VALUES('I202110',10,'Trainer');

```

15. Joins:

```

INSERT INTO JOINS VALUES('S202100','Math Club', 'President');
INSERT INTO JOINS VALUES('S202102','Math Club', 'Member');
INSERT INTO JOINS VALUES('S202103','Math Club', 'Member');
INSERT INTO JOINS VALUES('S202108','Math Club', 'Member');
INSERT INTO JOINS VALUES('S202114','Computer Club','President');
INSERT INTO JOINS VALUES('S202107','Computer Club','Member');
INSERT INTO JOINS VALUES('S202101','Physics Club','President');
INSERT INTO JOINS VALUES('S202112','Physics Club','Member');

```

```

INSERT INTO JOINS VALUES('S202102','Science Club','Vice President');
INSERT INTO JOINS VALUES('S202110','Science Club','President');
INSERT INTO JOINS VALUES('S202103','Chemistry Club','President');
INSERT INTO JOINS VALUES('S202115','Chemistry Club','Member');
INSERT INTO JOINS VALUES('S202104','Arabic Club','Member');
INSERT INTO JOINS VALUES('S202111','Arabic Club','President');
INSERT INTO JOINS VALUES('S202105','English Club','President');
INSERT INTO JOINS VALUES('S202106','Sports Club','Member');
INSERT INTO JOINS VALUES('S202115','Sports Club','Member');
INSERT INTO JOINS VALUES('S202112','Sports Club','President');
INSERT INTO JOINS VALUES('S202107','Music Club','Vice President');
INSERT INTO JOINS VALUES('S202113','Music Club','President');
INSERT INTO JOINS VALUES('S202108','Animal Care Club','President');
INSERT INTO JOINS VALUES('S202105','Animal Care Club','Member');
INSERT INTO JOINS VALUES('S202109','Computer Club','Secretary');

```

16. Enrolls In:

```

INSERT INTO ENROLLS_IN VALUES('S202100','12345');
INSERT INTO ENROLLS_IN VALUES('S202101','12300');
INSERT INTO ENROLLS_IN VALUES('S202102','12345');
INSERT INTO ENROLLS_IN VALUES('S202103','12305');
INSERT INTO ENROLLS_IN VALUES('S202104','02345');
INSERT INTO ENROLLS_IN VALUES('S202104','07345');
INSERT INTO ENROLLS_IN VALUES('S202105','67842');
INSERT INTO ENROLLS_IN VALUES('S202106','67842');
INSERT INTO ENROLLS_IN VALUES('S202107','12300');
INSERT INTO ENROLLS_IN VALUES('S202108','02345');
INSERT INTO ENROLLS_IN VALUES('S202109','12045');
INSERT INTO ENROLLS_IN VALUES('S202110','10333');
INSERT INTO ENROLLS_IN VALUES('S202111','10333');
INSERT INTO ENROLLS_IN VALUES('S202112','67842');

```

```

INSERT INTO ENROLLS_IN VALUES('S202113', '12045');
INSERT INTO ENROLLS_IN VALUES('S202114', '02345');
INSERT INTO ENROLLS_IN VALUES('S202114', '12305');
INSERT INTO ENROLLS_IN VALUES('S202115', '00345');
INSERT INTO ENROLLS_IN VALUES('S202116', '07345');
INSERT INTO ENROLLS_IN VALUES('S202117', '07345');
INSERT INTO ENROLLS_IN VALUES('S202118', '10344');
INSERT INTO ENROLLS_IN VALUES('S202119', '00345');
INSERT INTO ENROLLS_IN VALUES('S202118', '10333');
INSERT INTO ENROLLS_IN VALUES('S202119', '12305');
INSERT INTO ENROLLS_IN VALUES('S202104', '12345');
INSERT INTO ENROLLS_IN VALUES('S202105', '12300');
INSERT INTO ENROLLS_IN VALUES('S202110', '12345');
INSERT INTO ENROLLS_IN VALUES('S202111', '12305');
INSERT INTO ENROLLS_IN VALUES('S202107', '02345');
INSERT INTO ENROLLS_IN VALUES('S202115', '67842');
INSERT INTO ENROLLS_IN VALUES('S202110', '12300');
INSERT INTO ENROLLS_IN VALUES('S202109', '02345');
INSERT INTO ENROLLS_IN VALUES('S202110', '12045');
INSERT INTO ENROLLS_IN VALUES('S202114', '10333');
INSERT INTO ENROLLS_IN VALUES('S202118', '12045');
INSERT INTO ENROLLS_IN VALUES('S202101', '12305');
INSERT INTO ENROLLS_IN VALUES('S202102', '00345');
INSERT INTO ENROLLS_IN VALUES('S202103', '07345');
INSERT INTO ENROLLS_IN VALUES('S202105', '07345');
INSERT INTO ENROLLS_IN VALUES('S202106', '10344');
INSERT INTO ENROLLS_IN VALUES('S202107', '00345');
INSERT INTO ENROLLS_IN VALUES('S202112', '10333');
INSERT INTO ENROLLS_IN VALUES('S202113', '12305')

```

17. Teaches:

```
INSERT INTO TEACHES VALUES('12345', 'I202121');
INSERT INTO TEACHES VALUES('12300', 'I202129');
INSERT INTO TEACHES VALUES('12045', 'I202123');
INSERT INTO TEACHES VALUES('02345', 'I202121');
INSERT INTO TEACHES VALUES('12305', 'I202120');
INSERT INTO TEACHES VALUES('00345', 'I202122');
INSERT INTO TEACHES VALUES('07345', 'I202122');
INSERT INTO TEACHES VALUES('10344', 'I202123');
INSERT INTO TEACHES VALUES('67842', 'I202120');
INSERT INTO TEACHES VALUES('10333', 'I202129');
```

18. Takes Place:

```
INSERT INTO TAKES_PLACE VALUES('12345', 12, 'Kinderhaus');
INSERT INTO TAKES_PLACE VALUES('12300', 16, 'Kinderhaus');
INSERT INTO TAKES_PLACE VALUES('12045', 24, 'Sage');
INSERT INTO TAKES_PLACE VALUES('02345', 25, 'Sage');
INSERT INTO TAKES_PLACE VALUES('12305', 26, 'Sage');
INSERT INTO TAKES_PLACE VALUES('00345', 35, 'Nicole');
INSERT INTO TAKES_PLACE VALUES('07345', 36, 'Nicole');
INSERT INTO TAKES_PLACE VALUES('10344', 42, 'OG');
INSERT INTO TAKES_PLACE VALUES('67842', 45, 'OG');
INSERT INTO TAKES_PLACE VALUES('10333', 46, 'OG');
```

19. Course Days:

```
INSERT INTO COURSE_DAYS VALUES('12045', 'Monday');
INSERT INTO COURSE_DAYS VALUES('12045', 'Wednesday');
INSERT INTO COURSE_DAYS VALUES('12045', 'Friday');
INSERT INTO COURSE_DAYS VALUES('12345', 'Monday');
```

```
INSERT INTO COURSE_DAYS VALUES('12345', 'Thursday');
INSERT INTO COURSE_DAYS VALUES('12300', 'Tuesday');
INSERT INTO COURSE_DAYS VALUES('12300', 'Friday');
INSERT INTO COURSE_DAYS VALUES('02345', 'Wednesday');
INSERT INTO COURSE_DAYS VALUES('02345', 'Friday');
INSERT INTO COURSE_DAYS VALUES('12305', 'Thursday');
INSERT INTO COURSE_DAYS VALUES('12305', 'Tuesday');
INSERT INTO COURSE_DAYS VALUES('00345', 'Friday');
INSERT INTO COURSE_DAYS VALUES('00345', 'Tuesday');
INSERT INTO COURSE_DAYS VALUES('00345', 'Wednesday');
INSERT INTO COURSE_DAYS VALUES('07345', 'Tuesday');
INSERT INTO COURSE_DAYS VALUES('07345', 'Wednesday');
INSERT INTO COURSE_DAYS VALUES('10344', 'Monday');
INSERT INTO COURSE_DAYS VALUES('10344', 'Thursday');
INSERT INTO COURSE_DAYS VALUES('67842', 'Wednesday');
INSERT INTO COURSE_DAYS VALUES('10333', 'Monday');
```


X- Final Tables State:

1. Department

EDIT	NAME	LOCATION	EMAIL	PHONE_NUMBER
	Financial	Beirut	fn_dpvt@army.edu	01111111
	Special Education	Beirut	spedu_dpvt@army.edu	01111112
	Science	Byblos	sc_dpvt@army.edu	01111113
	Literature & Culture	Byblos	litC_dpvt@army.edu	01111114
	Transportation	Beirut	art_dpvt@army.edu	01111115
	Health & Safety	Beirut	hlthS_dpvt@army.edu	01111116
	Sport	Byblos	sport_dpvt@army.edu	01111117
	Admission	Beirut	adm_dpvt@army.edu	01111118
	Human Resources	Byblos	hr_dpvt@army.edu	01111119
	IT	Byblos	it_dpvt@army.edu	01111120

2. Employee

ID	FIRST_NAME	FATHERS_NAME	LAST_NAME	CITY	STREET	BUILDING	BIRTHDAY	PHONE_NUMBER	EMAIL	ROLE	SCHEDULE	NAME_OF_DEPARTMENT	SUPERVISORID
I202129	Elie	Joseph	Wehbe	Beirut	445	Grand	10/12/1998	03376772	elie.wehbe@army.edu	Head Instructor	M->F	Science	-
I202119	Mostafa	Ali	Marsi	Beirut	963	Solh	01/30/2001	70981236	mostafa.masri@army.edu	Manager	M->F	Human Resources	-
I202109	Sary	Steiman	Fahs	Jounieh	Thirteenth	Ghalyini	07/07/1996	76111120	sary.fahs@army.edu	Bus Driver	M->F	Transportation	-
I202100	John	Mark	Khoury	Beirut	First	Safadi	09/11/1992	76111111	john.khoury@army.edu	Bus Driver	M->F	Transportation	I202109
I202101	Marie	Bassam	Ismail	Byblos	Second	Salman	02/20/2000	76111112	marie.ismail@army.edu	Bus Driver	M->F	Transportation	I202109
I202102	Yara	Amin	Harb	Tyre	Fifteenth	Solh	10/01/1990	76111113	yara.harb@army.edu	Bus Driver	M->F	Transportation	I202109
I202103	Jad	Salim	Mayyas	Sidon	Eigth	Khalifeh	12/05/1992	76111114	jad.mayyas@army.edu	Bus Driver	M->F	Transportation	I202109
I202104	Sophie	Jean-Paul	Rahal	Batroun	Third	Harb	05/06/1994	76111115	sophie.rahal@army.edu	Bus Driver	M->F	Transportation	I202109
I202105	Omar	Moustafa	Salman	Tripoli	First	Khalaf	09/19/1992	76111116	omar.salman@army.edu	Bus Driver	M->F	Transportation	I202109
I202106	Mira	Jamil	Hafi	Aley	Tenth	Abedel Daset	03/04/1995	76111117	mira.hafi@army.edu	Bus Driver	M->F	Transportation	I202109
I202107	Mohamad	Kairm	Itani	Chouf	Eleventh	Zein	03/30/1997	76111118	mohamad.itani@army.edu	Bus Driver	M->F	Transportation	I202109
I202108	George	Mark	Sidani	Zahle	Seventeenth	Itani	08/28/1999	76111119	george.sidani@army.edu	Bus Driver	M->F	Transportation	I202109
I202114	Omar	Ali	Itani	Byblos	758	Zeinab	07/22/1995	70589632	omar.itani@army.edu	Head Coach	M->F	Sport	I202119
I202115	Jad	Zahi	Andi	Byblos	164	Farah	03/03/1983	70152369	jad.andi@army.edu	TA	M->F	Sport	I202114
I202110	Rani	Salman	Salman	Beirut	785	Khawli	09/05/1990	70630831	rani.salman@army.edu	Coach	M->F	Sport	I202114
ID	FIRST_NAME	FATHERS_NAME	LAST_NAME	CITY	STREET	BUILDING	BIRTHDAY	PHONE_NUMBER	EMAIL	ROLE	SCHEDULE	NAME_OF_DEPARTMENT	SUPERVISOR
I202111	Dani	Mazen	Breich	Beirut	242	Ali	11/13/1991	70963758	dani.breich@army.edu	Coach	M->F	Sport	I202114
I202112	Mazen	Mohammad	Chatila	Beirut	753	Sky	04/01/1990	70159063	mazen.chatila@army.edu	Coach	M->F	Sport	I202114
I202113	Hadi	Ramzi	Harb	Beirut	896	Glow	05/21/1993	70145489	hadi.harb@army.edu	Coach	M->F	Sport	I202114
I202116	Jad	Zouhair	Salman	Byblos	854	Anouti	08/05/1976	70845963	jad.salman@army.edu	TA	M->F	Science	I202129
I202117	Morwan	Maziad	AbouKarroum	Byblos	148	Khalife	03/02/1956	70145289	morwan.aboukarroum@army.edu	Accountant	M->F	Financial	I202119
I202118	Nidal	Adel	Elachkar	Byblos	741	Harb	02/06/1998	70481592	nidal.elachkar@army.edu	Secretary	M->F	Financial	I202119
I202120	Fadi	Joseph	Sarkis	Beirut	76	Sky	08/08/1990	70514233	fadi.sarkis@army.edu	Instructor	M->F	Literature & Culture	I202129
I202121	Fadi	Marc	Costa	Beirut	774	Anouti	10/10/1997	70448559	fadi.costa@army.edu	Instructor	M->F	Science	I202129
I202122	Joseph	Charbel	Maksoud	Byblos	778	Tall	05/06/2001	71596324	Joseph.maksoud@army.edu	Instructor	M->F	Literature & Culture	I202129
I202123	Mazen	Habib	Karake	Beirut	159	Plaza	07/08/2001	03326598	mazen.karake@army.edu	Instructor	M->F	Human Resources	I202129

3. Student

ID	FIRST_NAME	FATHERS_NAME	LAST_NAME	CITY	STREET	BUILDING	BIRTHDAY	PHONE_NUMBER	EMAIL	PATH	AVERAGE	BUS_TAKEN
S202100	Salim	Slaiman	Hammoud	Beirut	Ras el Ain	Farah	12/05/2004	03918111	salim.hammoud@army.edu	GS	8	B11440
S202101	Farah	Mohamad	Itani	Byblos	Omar el Zeinnie	Abedel khalek	09/22/2010	03918112	farah.itani@army.edu	LS	12	Z11449
S202102	Mohamad	Bassam	Makknieh	Zahle	Zouk II	Walid	05/19/2006	03918113	mohamad.makknieh@army.edu	SE	18	-
S202103	Sara	Hassan	Harb	Jounieh	Second 22	Khawli	06/13/2011	03918114	sara.harb@army.edu	HP	14	-
S202104	Ibrahim	Abbas	Sleit	Sidon	Aisha bakar 23	Itani I	04/06/2009	03918115	ibrahim.sleit@army.edu	GS	17	-
S202105	Yara	Ali	Ghalayini	Beirut	Ashrafieh 10	Sleit	03/10/2015	03918116	yara.ghalayini@army.edu	LS	19	B14444
S202106	Elias	George	Khoury	Batroun	Zouks 34	Ghalayini	07/18/2005	03918117	elias.khoury@army.edu	SE	2	-
S202107	Rana	Farhan	Salman	Beirut	Kamil 34	Abbas	10/30/2007	03918118	rana.salman@army.edu	HP	15	S11442
S202108	Assem	Ahmad	Farhat	Aley	Michel	Shine	02/21/2008	03918119	assem.farhat@army.edu	GS	17.8	-
S202109	Roula	Jad	Attar	Byblos	Third 12	Mira	01/01/2003	03918120	roula.attar@army.edu	LS	19.8	C11447
S202110	Roy	Mark	Solih	Beirut	Kouraytem 111	Mark B	09/11/2004	71131211	roy.solih@army.edu	LS	19	B11440
S202111	Tony	Hisham	Zaiter	Baalbek	Hamra 111	Hisham B	09/11/2004	78131211	toni.zaiter@army.edu	HP	18.7	-
S202112	Charbel	Fadi	Naim	Zahle	BBB313	Fadi B	09/11/2003	76134511	charbel.naim@army.edu	GS	15	-
S202113	Elias	Loay	Yaghi	Sidon	Souk 12	Safadi	09/11/2003	76145611	elias.yaghi@army.edu	LS	13	-
S202114	Omar	Ali	Shayah	Beirut	Ashrafeih 21	DAW AL Kamar	09/11/2002	70131211	omar.shayah@army.edu	SE	16.5	B11440
ID	FIRST_NAME	FATHERS_NAME	LAST_NAME	CITY	STREET	BUILDING	BIRTHDAY	PHONE_NUMBER	EMAIL	PATH	AVERAGE	BUS_TAKEN
S202115	Bakr	Omar	Sharif	Beirut	First 11	Omar BB	09/11/2001	03131211	bakr.sharif@army.edu	GS	14.5	J11443
S202116	Haidar	Ali	Tarhine	Beirut	Center	Ali B	09/11/2000	71112311	haidar.tarhine@army.edu	GS	3	B14444
S202117	Waleed	Othman	Khoury	Tyre	BCC123	AKSOB	09/11/2002	76131310	waleed.khoury@army.edu	LS	17	-
S202118	Rana	Mohammad	Harb	Byblos	BYB1234	Mohammad B	09/11/2001	76111222	rana.harb@army.edu	SE	8	C11447
S202119	Rani	Salman	Salem	Sawfar	Old st 21	Wydad B	09/11/2004	71913710	rani.salem@army.edu	HP	18	-

3. Utility

SN	NAME	DATE_OF_PURCHASE	WARRANTY	STATE	STUDENT_ID	EMPLOYEE_ID
FSD6789	Laptop	12/12/2020	2	Good condition	-	I202102
S45678	Mouse	07/05/2021	1	Good condition	-	-
FS56F78S	Calculator	05/15/1995	0	Needs Repair	S202101	-
FS5678SF67	Book: History of Lebanon	01/01/1985	0	Good condition	S202109	-
SF5S6FS6	Printer	07/01/2019	5	Needs Repair	-	-
ILUYI389L	Markers	02/15/2021	0	Good condition	-	I202115
LU7Y5LY	Projector	11/15/2018	3	Good condition	-	I202106
CVN5C79	Monitor	11/15/2018	5	Good condition	S202105	-
CN45CN8CV	Speaker	05/04/2015	5	Good condition	-	I202108
4YE5R6	Keyboard	01/18/2021	1	Good condition	-	-

4. Club

NAME	BUDGET	MEETING_TIME	EMAIL	SUPERVISORID
Math Club	400	17-SEP-21 01.00.00.000000 PM +02:00	MathClub@army.edu.lb	I202110
Physics Club	600	18-SEP-21 11.00.00.000000 AM +02:00	PhysiscsClub@army.edu.lb	I202111
Science Club	600	17-OCT-21 12.00.00.000000 PM +02:00	ScienceClub@army.edu.lb	I202112
Chemistry Club	1200	11-SEP-21 09.00.00.000000 AM +02:00	ChemistryClub@army.edu.lb	I202113
Arabic Club	400	17-DEC-21 02.00.00.000000 PM +02:00	ArabicClub@army.edu.lb	I202114
English Club	960	19-SEP-21 01.00.00.000000 PM +02:00	EnglishClub@army.edu.lb	I202115
Sports Club	400	17-SEP-21 04.00.00.000000 PM +02:00	SportsClub@army.edu.lb	I202116
Music Club	900	14-SEP-21 01.00.00.000000 PM +02:00	MusicClub@army.edu.lb	I202117
Animal Care Club	1050	17-SEP-21 02.30.00.000000 PM +02:00	AnimalCareClub@army.edu.lb	I202118
Computer Club	3000	17-SEP-21 01.30.00.000000 PM +02:00	ComputerClub@army.edu.lb	I202119

5. Sports-Team

TEAM_ID	NAME	SPORT_PLAYED	LOCATION_OF_TRAINING	TIME_OF_TRAINING
1	Wizards (Varsity)	Basketball	Basketball court	Wednesday 4:00 pm
2	Warriors (Junior Varsity)	Basketball	Basketball court	Monday 5:00 pm
3	Mystics	Volley Ball	Basketball court	Tuesday 4:00 pm
4	Stars (Varsity)	Football	Football field	Wednesday 5:00 pm
5	Suns (Junior Varsity)	Football	Football field	Friday 4:00 pm
6	Blitz	Ultimate Frisbee	Football field	Monday 4:00 pm
7	Rebels	Rugby	Football field	Tuesday 4:00 pm
8	Sweet Shots	Tennis	Tennis court	Tuesday 4:00 pm
9	Gladiators	Wrestling	Indoor Gymnasium	Thursday 5:00 pm
10	Fireballs	Hand ball	Basketball court	Friday 5:00 pm


























6. Event
















EVENT_NAME	EVENT_DATE	EVENT_TYPE	CLUB_NAME
Math Contest	10/22/2021	Competition	Math Club
Math Contest	10/22/2021	Competition	Physics Club
Intro to Space	05/14/2021	Educational	Physics Club
Science Fair	11/25/2021	Educational	Science Club
Group Experiments	12/25/2021	Educational	Chemistry Club
Poerty Night	12/14/2021	Entertainment	Arabic Club
Poerty Night	12/14/2021	Entertainment	English Club
Group Reading	09/19/2021	Entertainment	English Club
Football tournament	01/10/2022	Competition	Sports Club
Concert Night	12/08/2022	Entertainment	Music Club
Seminar on wild life	10/09/2022	Educational	Animal Care Club
Hackathon	10/19/2021	Competition	Computer Club

7. Plays-With

STUDENT_ID	SPORT_TEAM_ID
S202110	1
S202111	10
S202112	2
S202113	3
S202114	4
S202115	5
S202116	6
S202117	7
S202118	8
S202119	9

8. Enrolls-In

EDIT	STUDENT_ID	COURSE_CRN
	S202102	00345
	S202107	00345
	S202115	00345
	S202119	00345
	S202104	02345
	S202107	02345
	S202108	02345
	S202109	02345
	S202114	02345
	S202103	07345
	S202104	07345
	S202105	07345
	S202117	07345
	S202110	10333
	S202111	10333
EDIT	STUDENT_ID	COURSE_CRN
	S202113	12305
	S202114	12305
	S202119	12305
	S202100	12345
	S202102	12345
	S202104	12345
	S202110	12345
	S202105	67842
	S202112	67842
	S202115	67842

EDIT	STUDENT_ID	COURSE_CRN
	S202112	10333
	S202114	10333
	S202118	10333
	S202118	10344
	S202109	12045
	S202110	12045
	S202113	12045
	S202118	12045
	S202101	12300
	S202105	12300
	S202107	12300
	S202110	12300
	S202101	12305
	S202103	12305
	S202111	12305

9. Takes-Place

COURSE_CRN	ROOM_NUMBER	ROOM_BUILDING
00345	35	Nicole
02345	25	Sage
07345	36	Nicole
10333	46	OG
10344	42	OG
12045	24	Sage
12300	16	Kinderhaus
12305	26	Sage
12345	12	Kinderhaus
67842	45	OG

10. Bus

PLATE	BUS_NUMBER	COVERAGE_AREA	CAPACITY	COLOR	MODEL	DRIVERID
B11440	1	Beirut	30	White	Toyota	I202100
T11441	2	Beirut	20	White	Toyota	I202101
S11442	3	Beirut	35	Grey	Toyota	I202102
J11443	4	Beirut	25	White	Toyota	I202103
B14444	5	Beirut	25	White	Toyota	I202104
D11445	6	Byblos	30	Grey	Toyota	I202105
S11446	7	Byblos	35	White	Zonda	I202106
C11447	8	Byblos	25	Grey	Zonda	I202107
Z11448	9	Byblos	20	White	Zonda	I202108
Z11449	10	Byblos	25	Grey	Zonda	I202109

11. Course

CRN	NAME	CODE	NBR_OF_CREDITS	COURSE_TIME	MODE_OF_DELIVERY
12345	Math 1	MTH20	3	8:00	Online
12300	Science	SC90	3	9:00	Online
12045	Math 1	MTH20	3	10:00	Online
02345	Physics	PHY05	3	2:00	Hybrid
12305	Math 2	MTH40	3	3:00	Hybrid
00345	Economics	ECO56	3	10:00	Campus
07345	Geometry	MTH21	1	11:00	Campus
10344	Paint	PA00	3	2:00	Campus
67842	Arabic	ARA50	3	1:00	Campus
10333	Chemistry	CHM55	3	12:00	Campus




12. Coaches

EMPLOYEE_ID	SPORT_TEAM_ID	ROLE
I202119	1	Trainer
I202118	2	Trainer Assistant
I202117	3	Flexibility Trainer
I202116	4	Coach Assistant
I202115	5	Coach
I202114	6	captain
I202113	7	Trainer
I202112	8	Manager
I202111	9	Coach
I202110	10	Trainer

13. Room

BUILDING	ROOMNUMBER	CAPACITY	TYPE	LOCATION
Kinderhaus	12	30	Class	Beirut
Kinderhaus	13	5	Office	Beirut
Kinderhaus	14	5	Office	Beirut
Kinderhaus	15	5	Office	Beirut
Kinderhaus	16	30	Class	Beirut
Sage	22	5	Office	Beirut
Sage	23	5	Office	Beirut
Sage	24	30	Class	Beirut
Sage	25	30	Class	Beirut
Sage	26	30	Class	Beirut
Nicole	32	5	Office	Byblos
Nicole	33	5	Office	Byblos
Nicole	34	5	Office	Byblos
Nicole	35	30	Class	Byblos
Nicole	36	30	Class	Byblos
BUILDING	ROOMNUMBER	CAPACITY	TYPE	LOCATION
OG	42	30	Class	Byblos
OG	43	5	Office	Byblos
OG	44	5	Office	Byblos
OG	45	30	Class	Byblos
OG	46	30	Class	Byblos

14. Office

EDIT	ROOM_NUMBER	ROOM_BUILDING	EMPLOYEE_ID
	13	Kinderhaus	I202129
	14	Kinderhaus	I202123
	15	Kinderhaus	I202122
	22	Sage	I202121
	23	Sage	I202120
	32	Nicole	I202117
	33	Nicole	I202119
	34	Nicole	I202115
	43	OG	I202110
	44	OG	I202116

15. Course Days

COURSE_CRN	DAYS		
00345	Friday		
00345	Tuesday		
00345	Wednesday		
02345	Friday		
02345	Wednesday		
07345	Tuesday		
07345	Wednesday		
10333	Monday		
10344	Monday		
10344	Thursday	COURSE_CRN	DAYS
12045	Friday	12305	Thursday
12045	Monday	12305	Tuesday
12045	Wednesday	12345	Monday
12045	Friday	12345	Thursday
12300	Friday	67842	Wednesday

16. Guardian

STUDENT_ID	NAME	PHONE_NUMBER	EMAIL	CITY	STREET	BUILDING
S202101	Slaiman Hammoud	81456123	slaiman.hammoud@outlook.com	Beirut	654	Farah
S202102	Mohamad Itani	76985123	mohamad.itani@outlook.com	Jbeil	369	Abedel khalek
S202103	Bassam maknleh	81475555	bassam.maknleh@outlook.com	Zahle	258	WazlId
S202104	Hassan Harb	71830831	hassan.harb@outlook.com	Jounieh	147	Khawli
S202105	Abbas Sleit	03652148	abbas.sleit@outlook.com	Sidon	825	Itani I
S202110	Mark Solh	71717109	marksolh12@gmail.com	Beirut	Kouraytem111	Mark B
S202111	Hisham Zaiter	71231211	hisham.zaiter@gmail.com	Baalbek	Nahle 111	Hisham B
S202112	Fadi Naim	03134511	fadi.naim@gmail.com	Zahle	BBB313	Fadi B
S202113	Loay Yaghl	78145611	loay.yaghl@gmail.com	Sidon	Souk 12	Safadi
S202114	Ali Shayah	70111211	ali.shayah@gmail.com	Beirut	Ashrafeih21	DAW AL Kamar

17. Event

EVENT_NAME	EVENT_DATE	EVENT_TYPE	CLUB_NAME
Math Contest	10/22/2021	Competition	Math Club
Math Contest	10/22/2021	Competition	Physics Club
Intro to Space	05/14/2021	Educational	Physics Club
Science Fair	11/25/2021	Educational	Science Club
Group Experiments	12/25/2021	Educational	Chemistry Club
Poerty Night	12/14/2021	Entertainment	Arabic Club
Poerty Night	12/14/2021	Entertainment	English Club
Group Reading	09/19/2021	Entertainment	English Club
Football tournament	01/10/2022	Competition	Sports Club
Concert Night	12/08/2022	Entertainment	Music Club
Seminar on wild life	10/09/2022	Educational	Animal Care Club
Hackathon	10/19/2021	Competition	Computer Club

18. Teaches

COURSE_CRN	EMPLOYEE_ID
00345	I202122
02345	I202121
07345	I202122
10333	I202129
10344	I202123
12045	I202123
12300	I202129
12305	I202120
12345	I202121
67842	I202120

19. Joins

STUDENT_ID	CLUB_NAME	ROLE
S202100	Math Club	President
S202114	Computer Club	President
S202101	Physics Club	President
S202102	Science Club	Vice President
S202110	Science Club	President
S202103	Chemistry Club	President
S202104	Arabic Club	Member
S202111	Arabic Club	President
S202105	English Club	President
S202106	Sports Club	Member
S202115	Sports Club	Member
S202112	Sports Club	President
S202107	Music Club	Vice President
S202113	Music Club	President
S202108	Animal Care Club	President

STUDENT_ID	CLUB_NAME	ROLE
S202109	Computer Club	Secretary
S202102	Math Club	Member
S202103	Math Club	Member
S202108	Math Club	Member
S202107	Computer Club	Member
S202112	Physics Club	Member
S202115	Chemistry Club	Member

XI- Queries:

1. Raise in the club budgets

In the light of financial crisis that happens in Lebanon. Army School decides to increase the budgets of its clubs. Clubs should have a big budget in order to organize events and resume their normal work. However, not all the clubs will acquire the same rate of raise. The update for the budgets will be as follows: New budget=

- 1) $1.2 \times \text{old budget}$ for clubs that have an existing budget exceeds 700\$.
- 2) $1.5 \times \text{old budget}$ for the clubs that have an existing budget between 300\$ and 700\$.
- 3) $2 \times \text{old budget}$ for the clubs that have an existing budget less than 300\$.

- SQL Query:

```
UPDATE CLUB
```

```
SET
```

```
    Budget = Budget * 1.2
```

```
WHERE
```

```
    Budget > 700;
```

```
UPDATE CLUB
```

```
SET
```

```
    Budget = Budget * 1.5
```

```
WHERE
```

```
    300 < Budget AND Budget <= 700;
```

```
UPDATE CLUB
```

```
SET
```

```
    Budget = Budget * 2
```

```
WHERE
```

```
    Budget <= 300;
```

- OUTPUT:

```
3 row(s) updated. 0.00 seconds
```

```
3 row(s) updated. 0.00 seconds
```

```
4 row(s) updated. 0.01 seconds
```

2. Injured student in the football team

Charbel Naim is one of the most active and talented students at Army school. He loves Football games very much and his friends call him "Messi version of Lebanon". In one of his football games, Charbel's leg was broken due to an aggressive tackle by his friend Tony Yaghi. The management of the school immediately demanded the full information about the guardian of Charbel in order to contact him. In addition, the management needs the id of the sport team where Charbel played and the location of the training of this team.

- **SQL Query:**

```
SELECT GUARDIAN.Name, GUARDIAN.Email, GUARDIAN.Phone_Number,  
SPORTS_TEAM.Location_Of_Training  
FROM STUDENT, SPORTS_TEAM, PLAYS_WITH, GUARDIAN  
WHERE STUDENT.First_Name = 'Charbel'  
      AND STUDENT.Last_Name = 'Naim'  
      AND STUDENT.ID = PLAYS_WITH.Student_ID  
      AND GUARDIAN.Student_ID = STUDENT.ID  
      AND PLAYS_WITH.Sport_Team_ID = SPORTS_TEAM.TEAM_ID
```

- **OUTPUT:**

NAME	EMAIL	PHONE_NUMBER	LOCATION_OF_TRAINING
Fadi Naim	fadi.naim@gmail.com	03134511	Basketball court

3. Urgent Meeting with the clubs

Due to an urgent matter that invaded the school, a meeting is needed to be scheduled with the presidents and vice president of the clubs. To notify them with the meeting an email should be sent. Therefore, to send the email, the first name, last name, email, club name, and role of each president and vice president is required.

- SQL Query:

```
SELECT STUDENT.First_Name, STUDENT.Last_Name,  
STUDENT.Email, JOINS.Club_Name, JOINS.Role  
FROM STUDENT, JOINS  
WHERE STUDENT.ID=JOINS.Student_ID  
      AND (JOINS.Role='President' OR JOINS.Role='Vice  
President');
```

- OUTPUT:

FIRST_NAME	LAST_NAME	EMAIL	CLUB_NAME	ROLE
Salim	Hammoud	salim.hammoud@army.edu	Math Club	President
Farah	Itani	farah.itani@army.edu	Physics Club	President
Mohamad	Makknieh	mohamad.makknieh@army.edu	Science Club	Vice President
Sara	Harb	sara.harb@army.edu	Chemistry Club	President
Yara	Ghalayini	yara.ghalayini@army.edu	English Club	President
Rana	Salman	rana.salman@army.edu	Music Club	Vice President
Assem	Farhat	assem.farhat@army.edu	Animal Care Club	President
Roy	Solh	roy.solh@army.edu	Science Club	President
Tony	Zaiter	toni.zaiter@army.edu	Arabic Club	President
Charbel	Naim	charbel.naim@army.edu	Sports Club	President
Elias	Yaghi	elias.yaghi@army.edu	Music Club	President
Omar	Shayah	omar.shayah@army.edu	Computer Club	President

4. Argument between the parent and instructor

One Day, one of the receptionists of the school got a phone call from an angry parent asking to get the contact information of an instructor who, according to her son, was intentionally giving him bad grades after a personal conflict between the two. The receptionist knew that an instructor would never do that but still had to obey the parent and give her the contact information of this instructor.

We will write a query that gets us the email, ID, phone number and office of an instructor named Mazen Karake.

- SQL Query:

```
SELECT EMPLOYEE.Email, EMPLOYEE.ID,  
EMPLOYEE.Phone_Number, OFFICE.Room_Number,  
OFFICE.Room_Building  
FROM EMPLOYEE, OFFICE  
WHERE EMPLOYEE.First_Name = 'Mazen' AND  
EMPLOYEE.Last_Name='Karake' AND  
EMPLOYEE.Role='Instructor'  
AND EMPLOYEE.ID = OFFICE.Employee_ID
```

- OUTPUT:

EMAIL	ID	PHONE_NUMBER	ROOM_NUMBER	ROOM_BUILDING
mazen.karake@army.edu	I202123	03326598	14	Kinderhaus

5. Failing students

In the school, we found that some of the students were not serious enough and did not cooperate with any of our plans to guide them in order to get better grades. So, the school decided to make these students drop their courses and make them take an entrance exam to be able to take courses again later. We will write a query that make students that have an average less than 5 to drop all their current courses.

- **SQL Query:**

```
DELETE FROM ENROLLS_IN
WHERE ENROLLS_IN.STUDENT_ID IN
(
    SELECT STUDENT.ID FROM STUDENT WHERE
    STUDENT.Average<5
);
```

- **OUTPUT:**

3 row(s) deleted. 0.01 seconds

6. Honor Students

The school decided to make a gesture and acknowledge all the students, that are currently enrolled in two or more courses and have an average of 16 and above as honor students. These students will have a graduation of their own and a different diploma that mentions their outstanding performance in school.

We will write a query that displays all students that are taking more than one course and have an average of 16 and above.

- **SQL Query:**

```
CREATE TABLE Student_CourseCount AS
SELECT STUDENT.ID AS ID, COUNT(STUDENT.ID) AS
Nb_Of_Courses
FROM STUDENT, ENROLLS_IN
WHERE STUDENT.ID=ENROLLS_IN.Student_ID AND
STUDENT.Average>=16
GROUP BY STUDENT.ID;

SELECT STUDENT.First_Name, STUDENT.Last_name,
STUDENT.Email
FROM STUDENT, Student_CourseCount
WHERE STUDENT.ID=Student_CourseCount.ID AND
Student_CourseCount.Nb_Of_Courses>=3;
```


DROP TABLE Student_CourseCount;

- **OUTPUT:**

Table created. 0.01 seconds

FIRST_NAME	LAST_NAME	EMAIL
Ibrahim	Sleit	ibrahim.sleit@army.edu
Yara	Ghalayini	yara.ghalayini@army.edu
Roy	Solh	roy.solh@army.edu
Omar	Shayah	omar.shayah@army.edu

4 rows selected. 0.01 seconds

Table dropped. 0.01 seconds

7. **Coronavirus case**

During the week, news broke out that one of the students in ARMY school caught Covid-19. The school was able to contact this student and get his ID number which is 'S202110'. To abide by the covid-19 health and safety measures, the school administration needs to urge the people that may have been in contact with the student to take PCR tests and stay at home. This query's job is to present the school administration with the contact information of the teachers and students that teach and attend the same courses at the Covid-19 case.

- **SQL Query:**

```
A. SELECT DISTINCT EMPLOYEE.First_Name, EMPLOYEE.Last_Name,
EMPLOYEE.Email, EMPLOYEE.Phone_Number
FROM EMPLOYEE, TEACHES
WHERE EMPLOYEE.ID=TEACHES.Employee_ID
AND TEACHES.Course_CRN IN
(
SELECT ENROLLS_IN.Course_CRN
FROM ENROLLS_IN
```

```
WHERE ENROLLS_IN.Student_ID='S202110'
);
```

```
B. SELECT DISTINCT STUDENT.First_Name, STUDENT.Last_Name,
STUDENT.Email, STUDENT.Phone_Number
FROM STUDENT, ENROLLS_IN
WHERE STUDENT.ID=ENROLLS_IN.STUDENT_ID AND
ENROLLS_IN.Student_ID!='S202110'
AND ENROLLS_IN.Course_CRN IN
(
SELECT ENROLLS_IN.Course_CRN
FROM ENROLLS_IN
WHERE ENROLLS_IN.Student_ID='S202110'
);
```

• OUTPUT:

A.

FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
Elie	Wehbe	elie.wehbe@army.edu	03376772
Fadi	Costa	fadi.costa@army.edu	70448559
Mazen	Karake	mazen.karake@army.edu	03326598

B.

FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
Charbel	Naim	charbel.naim@army.edu	76134511
Roula	Attar	roula.attar@army.edu	03918120
Ibrahim	Sleit	ibrahim.sleit@army.edu	03918115
Tony	Zaiter	toni.zaiter@army.edu	78131211
Rana	Harb	rana.harb@army.edu	76111222
Salim	Hammoud	salim.hammoud@army.edu	03918111
Yara	Ghalayini	yara.ghalayini@army.edu	03918116
Rana	Salman	rana.salman@army.edu	03918118
Omar	Shayah	omar.shayah@army.edu	70131211
Mohamad	Makknieh	mohamad.makknieh@army.edu	03918113
Farah	Itani	farah.itani@army.edu	03918112
Elias	Yaghi	elias.yaghi@army.edu	76145611

8. Utility check

The school wants to check on the state of its equipment and utilities to figure out which of them needs to be repaired. After detecting the ones that needs to be repaired, one of the employees remembers that some of them might still have warranty which might save the school some money, but he chose to keep this information to himself in order to save the money of the repair to his pocket. At night, the employee sneaks to the database of the school and gets a list of the utility that needs repair and split them into two categories. Those that still have warranty where he will take the case of the repair and the utility that no longer have warranty. This query will check if a utility needs repair and if it still has warranty.

- SQL Query:

A. `SELECT UTILITY.SN, UTILITY.Name
FROM UTILITY
WHERE UTILITY.State='Needs Repair'
AND EXTRACT(YEAR FROM
UTILITY.Date_Of_Purchase)+WARRANTY>=2021`

B. `SELECT UTILITY.SN, UTILITY.Name
FROM UTILITY
WHERE UTILITY.State='Needs Repair'
AND EXTRACT(YEAR FROM UTILITY.Date_Of_Purchase) +
WARRANTY<2021`

- OUTPUT:

A.

SN	NAME
SF5S6FS6	Printer

B.

SN	NAME
FS56F78S	Calculator

9. Math Contest Event

Clubs in ARMY school are very active and they always organize events. Army school administration needs to ensure that students are participating in these events and the budget of the clubs is not useless. Math contest is one of the activities that was organized in the school by the clubs. The manager of the school demands the full information about the students who attended this activity as well as to which clubs these students belong. In our database each event has the name(s) of the clubs that organize it; however, the manager needs information about the students. This query will solve the problem.

- SQL Query:

```
SELECT DISTINCT STUDENT.First_Name, STUDENT.Last_Name,  
STUDENT.ID, CLUB.Name  
FROM CLUB JOIN EVENT ON Name = Club_Name, STUDENT JOIN  
JOINS ON ID = Student_ID  
WHERE EVENT.Event_Name = 'Math Contest' AND  
EVENT.Club_Name=JOINS.Club_Name
```

- OUTPUT:

FIRST_NAME	LAST_NAME	ID	NAME
Salim	Hammoud	S202100	Math Club
Assem	Farhat	S202108	Math Club
Mohamad	Makknieh	S202102	Math Club
Sara	Harb	S202103	Math Club
Farah	Itani	S202101	Physics Club
Charbel	Naim	S202112	Physics Club

10. Third Party Bus Company

ARMY's school buses mainly transport students that are living in Beirut and Byblos. So a third-party bus company came with the proposal to take charge of the students living outside of Beirut and Byblos. This company doesn't have access to the school's database. So to offer their services, it contacted the school

administration and asked for the contact information and the city of students residing outside of Beirut and Byblos. This query will check if a student already takes a bus by checking if the referenced-key bus taken from the STUDENT table is null. Then it will check if the student lives outside of Beirut and Byblos.

- **SQL Query:**

```
SELECT STUDENT.First_Name, STUDENT.Last_Name, STUDENT.Email,  
STUDENT.Phone_Number, STUDENT.City  
FROM STUDENT  
WHERE (STUDENT.Bus_Taken IS NULL)  
      AND (STUDENT.City != 'Beirut' AND STUDENT.City  
          =!'Byblos')
```

- **OUTPUT:**

FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	CITY
Mohamad	Makknieh	mohamad.makknieh@army.edu	03918113	Zahle
Sara	Harb	sara.harb@army.edu	03918114	Jounieh
Ibrahim	Sleit	ibrahim.sleit@army.edu	03918115	Sidon
Elias	Khoury	elias.khoury@army.edu	03918117	Batroun
Assem	Farhat	assem.farhat@army.edu	03918119	Aley
Tony	Zaiter	toni.zaiter@army.edu	78131211	Baalbek
Charbel	Naim	charbel.naim@army.edu	76134511	Zahle
Elias	Yaghi	elias.yaghi@army.edu	76145611	Sidon
Waleed	Khoury	waleed.khoury@army.edu	76131310	Tyre
Rani	Salem	rani.salem@army.edu	71913710	Sawfar

XII-Normalization Up to The BCNF Normal Form:

After creating all relations, normalization is going to be carried out so that the resulting designs are of high quality and meet the desirable properties. Here we are going to normalize our database up to the Boyce-Codd Normal Form. On each relation we are going to apply the four normal forms. We start with the first then second, third and at last the BCNF normal form. Let us first start by a general description of each normal form.

First Normal Form:

The first normal form disallows any multivalued and composite attributes as well as attributes whose values for an individual tuple are non-atomic, known as nested relations. Therefore, the only attribute values permitted are single atomic values.

Second Normal Form:

Definitions:

- A **Prime attribute** must be a member of some candidate key
- A **Non-prime attribute** is not a prime attribute—that is, it is not a member of any candidate key.
- **Functional Dependency:** an FD $Y \rightarrow Z$, where removal of any attribute from Y means the FD does not hold anymore.

The Second normal form is based on the concept of full functional dependency. A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key.

Third Normal form:

Definitions:

- **Transitive functional dependency:** an FD $X \rightarrow Z$ that can be derived from two FDs $X \rightarrow Y$ and $Y \rightarrow Z$. In $X \rightarrow Y$ and $Y \rightarrow Z$, with X as the primary key, we consider this a problem only if Y is not a candidate key. When Y is a candidate key, there is no issue with the transitive dependency.

The third normal form is based on the concept of transitive dependency. A relation schema R is in third normal form (3NF) if it is in 2NF and no non-prime attribute A in R is transitively dependent on the primary key.

A relation schema R is in third normal form (3NF) if whenever a FD $X \rightarrow A$ holds in R, then either:

(a) X is a superkey of R

OR

(b) A is a prime attribute of R

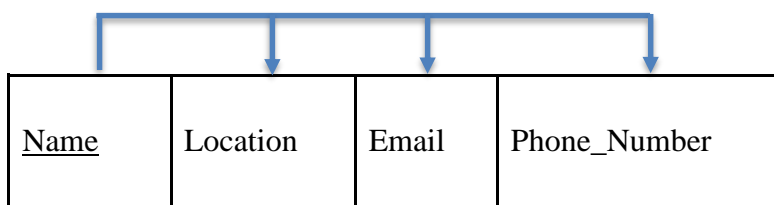
Condition (a) catches two types of violations:

- ❖ One where a prime attribute functionally determines a non-prime attribute. This catches 2NF violations due to non-full functional dependencies.
- ❖ Second, where a non-prime attribute functionally determines a non-prime attribute. This catches 3NF violations due to a transitive dependency.

Boyce-Codd Normal Form:

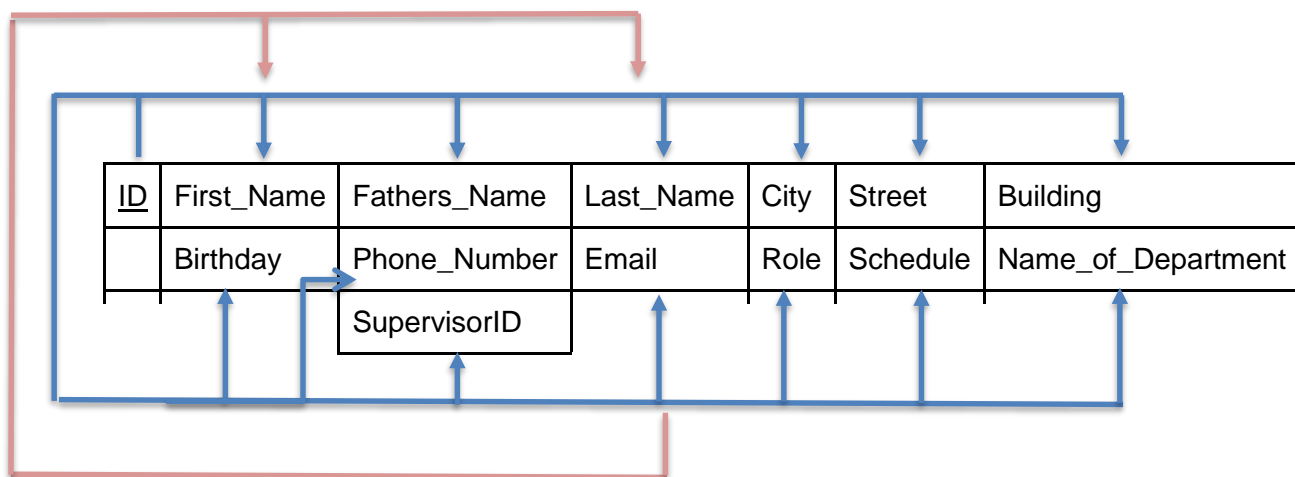
The Boyce-Codd normal form is a stricter form than the third normal form. The BCNF differs from the definition of the third normal form in only one condition. The third normal form allows the right-hand side of the functional dependency to be a prime attribute, while BCNF does not allow that.

1. Department:



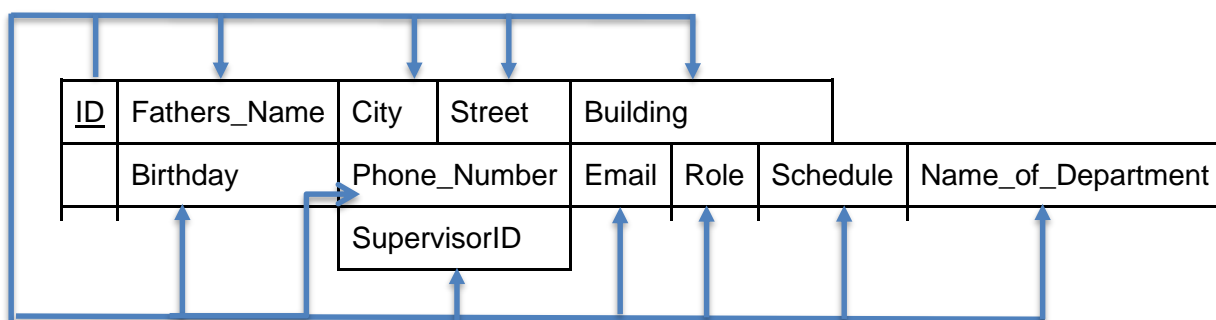
- A. The **DEPARTMENT** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **DEPARTMENT** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Name”.
- C. The **DEPARTMENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “Name”.
- D. The **DEPARTMENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

2. EMPLOYEE



Normalize:

a.EMPLOYEE_ONE

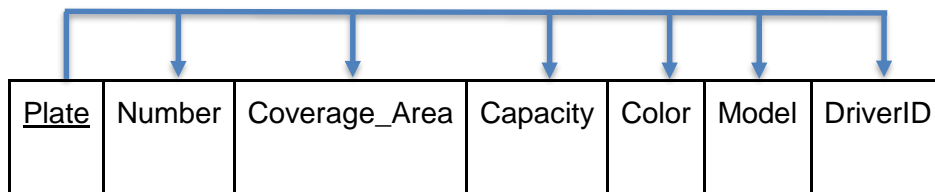


b. EMPLOYEE_TWO

<u>Email</u>	First_Name	Last_Name
--------------	------------	-----------

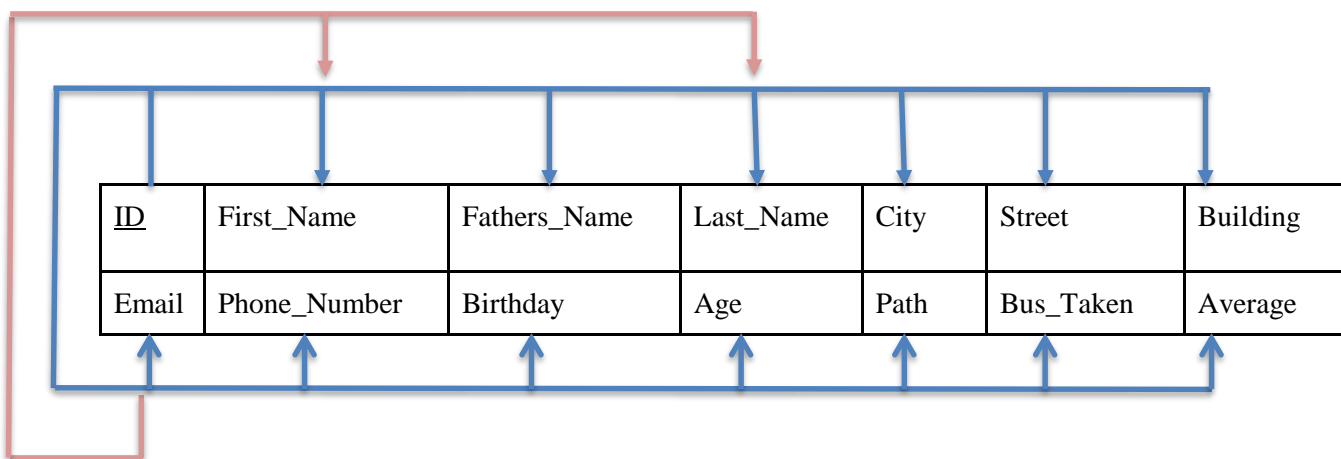
- A. The **EMPLOYEE** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **EMPLOYEE** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “ID”.
- C. Although the **EMPLOYEE** relation schema satisfies the 2NF, it doesn’t satisfy all conditions of the 3NF because “Email” determines “First_Name” and “Last_Name” (FD: Email → First_Name and Last_Name); however, “Email” is not a super key and “First_Name” and “Last_Name” are not prime attributes.
- D. The **EMPLOYEE** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

3. BUS



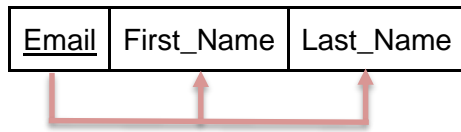
- A. The **BUS** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **BUS** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Plate”.
- C. The **BUS** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “Plate”.
- D. The **BUS** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

4. STUDENT

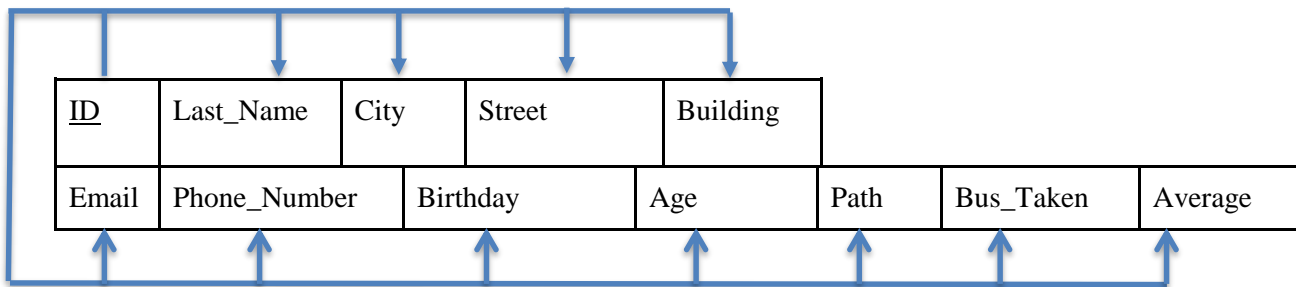


Normalize:

a. STUDENT_ONE




b. STUDENT_TWO



- A. The **STUDENT** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **STUDENT** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “ID”.
- C. Although the **STUDENT** relation schema satisfies the 2NF, it doesn’t satisfy all conditions of the 3NF because Email determines “First_Name” and “Last_Name” (FD: Email → First_Name and Last_Name); however, “Email” is not a super key and the “First_Name” and “Last_Name” are not prime attributes.
- D. The **STUDENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

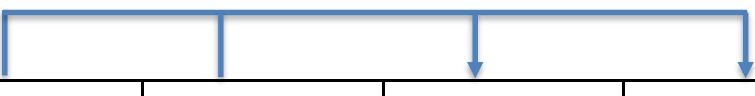
5. CLUB



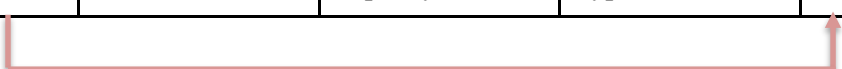
<u>Name</u>	Budget	Meeting_Time	Type	Email	SupervisorID
-------------	--------	--------------	------	-------	--------------

- A. The **CLUB** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **CLUB** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Name”.
- C. The **CLUB** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “Name”.
- D. The **CLUB** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

6. ROOM

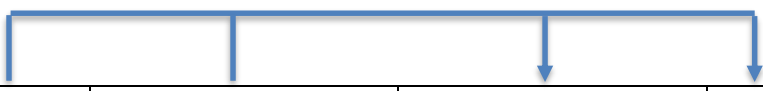


<u>Building</u>	<u>Number</u>	Capacity	Type	Location
-----------------	---------------	----------	------	----------




Normalize:

a. ROOM_ONE



<u>Building</u>	<u>Number</u>	Capacity	Type
-----------------	---------------	----------	------

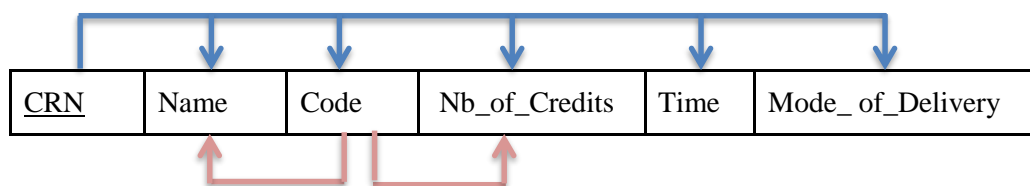
b. ROOM_TWO



<u>Building</u>	Location
-----------------	----------

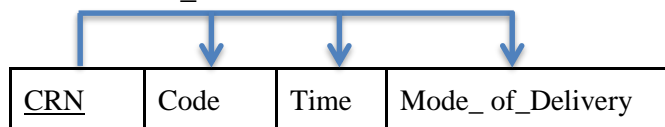
- A. The **ROOM** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **ROOM** relation schema doesn't satisfy all conditions of the 2NF because not every non-prime attribute is fully functionally dependent on the primary key.
"Location" attribute (Ex: Beirut campus or Byblos) is not a prime attribute, and it is dependent only on the "Building" attribute. Thus, "Location" attribute is partially dependent on the primary key which is the combination of "Building and number".
- C. The **ROOM** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key "Building" and "Number".
- D. The **ROOM** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

7. COURSE

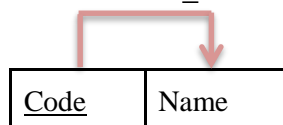


Normalize:

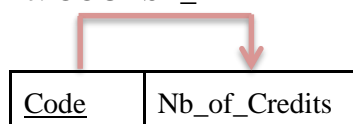
a. COURSE_ONE



b. COURSE_TWO

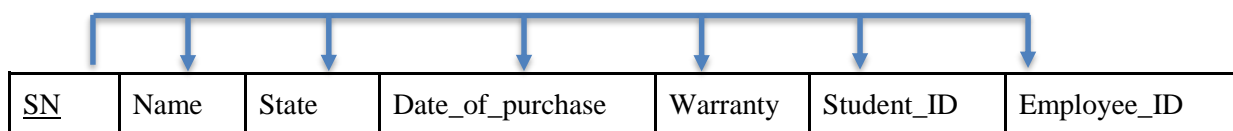


c. COURSE_THREE



- A. The **COURSE** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **COURSE** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “CRN”.
- C. Although the **COURSE** relation schema satisfies the 2NF, it doesn’t satisfy all conditions of the 3NF because “Code (Ex: MTH20, CHM55) determines “Nb_of_Credits” (FD1: Code \rightarrow Nb_of_Credits) and “Code” determines “Name” (FD2: Code \rightarrow Name); however, “Code” is not a super key and the “Nb_of_Credits” and “Name” are not prime attributes.
- D. The **COURSE** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

8. UTILITY



- A. The **UTILITY** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **UTILITY** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “SN”.
- C. The **UTILITY** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “SN”.
- D. The **UTILITY** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

9. SPORTS_TEAM

<u>Team_ID</u>	Name	Sport_Played	Location_of_Training	Time
----------------	------	--------------	----------------------	------

Normalize:

a. SPORTS_TEAM_ONE

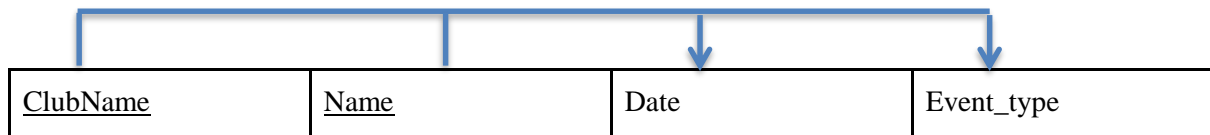
<u>Sport_Played</u>	Location_of_Training
---------------------	----------------------

b. SPORTS_TEAM_TWO

<u>Team_ID</u>	Name	Sport_Played	Time
----------------	------	--------------	------

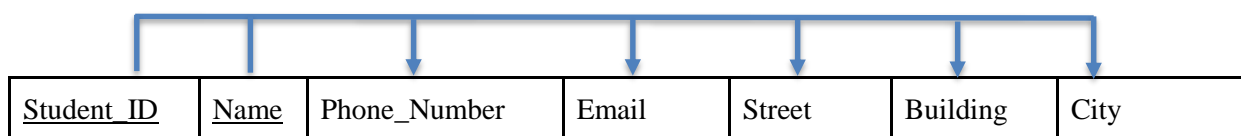
- A. The **SPORTS_TEAM** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **SPORTS_TEAM** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Team_ID”.
- C. Although the **SPORTS_TEAM** relation schema satisfies the 2NF, it doesn’t satisfy all conditions of the 3NF because “Sport_Played” determines “Location_of_Training” (FD: Sport_Played \rightarrow Location_of_Training); however, “Sport_Played” is not a super key and the “Location_of_Training” is not a prime attribute.
- D. The **SPORTS_TEAM** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

10. EVENT



- A. The **EVENT** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **EVENT** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “ClubName” and “Name”.
- C. The **EVENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “ClubName” and “Name”.
- D. The **EVENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

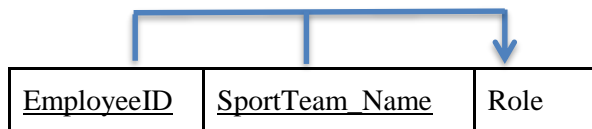
11. GUARDIAN



- A. The **GUARDIAN** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **GUARDIAN** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Student_ID” and “Name”.
- C. The **GUARDIAN** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “Student_ID” and “Name”.

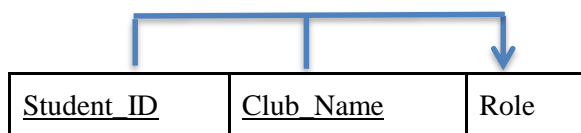
- D. The **GUARDIAN** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

12. COACHES



- A. The **COACHES** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **COACHES** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “EmployeeID” and “SportTeam_Name”.
- C. The **COACHES** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “EmployeeID” and “SportTeam_Name”.
- D. The **COACHES** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

13. JOINS



- A. The **JOINS** relation schema satisfies all conditions of the 1NF because it contains single and atomic attributes. This relation has neither multivalued attributes nor composite attributes.
- B. The **JOINS** relation schema satisfies all conditions of the 2NF because every non-prime attribute is fully functionally dependent on the primary key “Student_ID” and “Club_Name”.

- C. The **JOINS** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there are no non-prime attributes that are transitively dependent on the primary key “Student_ID” and “Club_Name”.
- D. The **JOINS** relation schema satisfies all conditions of the BCNF because there exists no functional dependency $X \rightarrow A$ where X is not a super key or A is a prime attribute and X not a super key.

Relation Schemas without non-prime attributes:

14. OFFICE

<u>Room_Number</u>	<u>Building_Name</u>	<u>Employee_ID</u>
--------------------	----------------------	--------------------

15. PLAYS_WITH

<u>Student_ID</u>	<u>SportTeam_Name</u>
-------------------	-----------------------

16. ENROLLS_IN

<u>Student_ID</u>	<u>Course_CRN</u>

17. TEACHES

<u>Employee_ID</u>	<u>Course_CRN</u>
--------------------	-------------------

18. TAKES_PLACE

<u>Course_CRN</u>	<u>Room_Name</u>	<u>Building_Name</u>
-------------------	------------------	----------------------

19. COURSEDAYS

<u>CourseCRN</u>	<u>Days</u>
------------------	-------------

XIII-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.

XIV- Phase III 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.

Corrected phase I and II: Yes.

Improved phase I and II: Yes.

Translated the ER diagram into relational schemas: Yes.

Build database on the Oracle Server: Yes.

At least ten tuples per relation on the database server: Yes.

10 transactions to query database: Yes.

Express queries in SQL: Yes.

Execute SQL queries on the database implemented using ORACLE: Yes.

Print the results: Yes.

Normalize the database up to the BCNF Normal Form: Yes

Explain the normalization process: Yes

XV- GitHub Repository

<https://github.com/mhmdkhlf/School-CSC375>

XVI- Instructor's feedback and evaluation

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.

[illegible]