AIT-524-003

Homework 3

***Description of the Company:***

Student Housing and Residence Life is part of University Housing and Maintenance Department providing on campus living to the college students. Students can enroll in the various housing options provided. The database keeps tracks of accommodation provided to students and helps maintain information regarding issues in the houses. This organization helps students have an amazing experience during their stay in college.

***Need for a database:***

A well-planned database is needed by the Housing department to keep track of students who enroll in on-campus stay. There are dorms that need immediate attention which are attended by supervisors as directed by managers. All the data related to student accommodation and their issues will have to be stored in a database.

***Entities:***

1. **Student**- This entity collects each student data to keep track of them on campus stay.

**Attributes-** STUDENT(STU\_ID, STU\_FIRSTNAME, STU\_LASTNAME, STU\_ADMIT, STU\_DOB, STU\_PHNO)

**Description of attributes:**

-STU\_ID- This attribute mentions student’s ID**(Primary key).**

**-**STU\_FIRSTNAME: This attribute mentions student’s first name.

**-**STU\_LASTNAME: This attribute mentions student’s last name.

**-**STU\_ADMIT :This attribute mentions if a student is freshmen/junior/sophomore /senior.

**-**STU\_DOB: This attribute mentions student’s date of birth.

**-**STU\_PHNO: This attribute mentions student’s phone number.

**Data types included:**

-STU\_FIRSTNAME, STU\_LASTNAME, STU\_ADMIT: varchar

-STU\_DOB: date

-STU\_PHNO: int

1. **Manager**- This entity holds the manager data who handles if there is any issue in the dorms.

**Attributes**- MANAGER(MAN\_ID, MAN\_FIRSTNAME, MAN\_LASTNAME, MAN\_PHNO, MAN\_LOCATION, MAN\_ROOMNO)

**Description of attributes:**

MAN\_ID: This attribute mentions manager ID. **(Primary key).**

MAN\_FIRSTNAME: This attribute mentions manager first name.

MAN\_LASTNAME: This attribute mentions manager last name.

MAN\_PHNO: This attribute mentions manager phone number.

MAN\_LOCATION: This attribute mentions manager location(his office).

MAN\_ROOMNO: This attribute mentions manager room no.

**Data types included:**

-MAN\_FIRSTNAME, MAN\_LASTNAME, MAN\_LOCATION: varchar

-MAN\_ID, MAN\_PHNO, MAN\_ROOMNO: int

1. **Supervisor-** This entity records the supervisor data who attends the issue in the dorms and reports to manager.

**Attributes-**SUPERVISOR(SUP\_ID, SUP\_FIRSTNAME, SUP\_LASTNAME, SUP\_PHNO)

**Description of attributes:**

SUP\_ID: This attribute mentions supervisor ID.**(Primary key).**

SUP\_FIRSTNAME: This attribute mentions supervisor first name.

SUP\_LASTNAME: This attribute mentions supervisor last name.

SUP\_PHNO: This attribute mentions supervisor phone number.

**Data types included:**

-SUP\_FIRSTNAME, SUP\_LASTNAME: varchar

-SUP\_ID, SUP\_PHNO: int

1. **Dorm-** This entity holds information of housing where the students reside.

**Attributes-** DORM(DORM\_ROOMNO, DORM\_LOCATION, DORM\_NAME, DORM\_TYPE, DORM\_PRICE)

**Description of attributes:**

DORM\_LOCATION: This attribute mentions the location of the dorm.

DORM\_ROOMNO: This attribute mentions the room number of the dorm. **(Primary key).**

DORM\_NAME: This attribute mentions the name of dorm.

DORM\_TYPE: This attribute mentions the number of rooms.

DORM\_PRICE: This attribute mentions the price of the dorm.

**Data types included:**

-DORM\_LOCATION, DORM\_NAME, DORM\_TYPE- varchar

-DORM\_ROOMNO, DORM\_PRICE- int

1. **Sup\_Dorm-** This entity is a bridging entity between Supervisor and Dorm.

**Attributes-** Sup\_Dorm(SupDorm\_ID,Sup\_ID, Dorm\_Roomno)

**Description of attributes:**

SUPDORM\_ID- This attribute the ID for Supervisor and Dorm.

SUP\_ID: This attribute mentions supervisor ID.

DORM\_ROOMNO: This attribute mentions the room number of the dorm.

**Data types included:**

SupDorm\_ID,Sup\_ID, Dorm\_Roomno- int

1. **Service\_Request-** This entity holds information for the services request by students.

**Attributes-** Service\_Request(Service\_ID, Service\_Name)

**Description of attributes:**

Service\_ID – This is the ID associated with the service. **( Primary key)**

Service\_Name- This attribute mentions the name of the service request put by the student.

**Data types included:**

Service\_ID- int

Service\_Name- varchar

1. **Stud\_Service-** This entity is a bridging entity between Student and Service Request.

**Attributes-** Stud\_Service(Stud\_ID, Service\_ID)

**Description of attributes-**

Service\_ID- This attribute mentions the name of the service request put by the student**(Primary key, Foreign Key).**

STU\_ID- This attribute mentions student’s ID**(Primary key, Foreign Key).**

***Business Rules:***

***Rule 1: The entities of this relationship are student and supervisor.***

Many students are reported to one supervisor, so here it is M:1 relation and one supervisor will attend to many students, so here it is 1:M relation. After evaluating the relations from both sides, it is M:1 relation. The relationship between Student and Supervisor is strong because entity Student is dependent on Supervisor. The students “must” report to Supervisor. So, it is mandatory on both sides because many students report to Supervisor and one Supervisor attends to many students. According to business rules, the cardinality on the student side is (1,M) and Supervisor side is (1,1) and is mandatory on both sides.

***Rule 2: The entities of this relationship are Manager and Supervisor.***

Each manager manages many supervisors, so it is 1:M relation. Many supervisors are managed by one manager, it is M:1 relation. After evaluating from both sides, it is 1:M relation. The relationship between Manager and Supervisor is strong Supervisor is dependent on Manager. The Supervisor “must” report to his Manager. It is mandatory on both sides every supervisor reports to his manager and every manager manages his supervisor. According to business rules, the cardinality on the Manager side is (1,1) and on Supervisor side is (1,M) and mandatory on both sides.

***Rule 3: The entities of this relationship are Supervisor and Dorm.***

One supervisor attends to work in many dorms, so it is 1:M relation. Work in one dorm is attended by many supervisors, it is 1:M relation. After evaluating from both sides, it is M:N relation. It is mandatory on both sides. A bridging table Sup\_Dorm is constructed between Supervisor and Dorm entities to resolve M:N relationship since M:N cannot exist in a relational database. Hence, the relationship between supervisor and Sup\_Dorm is weak and , the relationship between dorm and Sup\_Dorm is weak. According to business rules, the cardinality for Supervisor and Sup\_Dorm is (1,1) on Supervisor side and (1,M) on Sup\_Dorm side, mandatory on both sides. The cardinality for Dorm and Sup\_Dorm is (1,1) on Dorm side and (1,M) on Sup\_Dorm side, mandatory on both sides.

***Rule 4: The entities of this relationship are Student and Service Request.***

One Student can place many Service Requests, so it is 1:M relation. One Service Request can be placed by many Students so it is 1:M relation. After evaluating from both sides, it is M:N relation. There is optionality between Student and Stud\_Service. A bridging table Stud\_Service is constructed between Student and Service\_Request entities to resolve M:N relationship since M:N cannot exist in a relational database. Hence, the relationship between Student and Stud\_Service is weak and , the relationship between Stud\_Service and Service\_Request is weak. According to business rules, the cardinality for Student and Stud\_Service is (1,1) on Student side and (1,M) on Stud\_service side, optional on Stud\_Service side because a Student can place zero or many requests. The cardinality for Stud\_Service and Service\_Request is (1,1) on Service\_Request side and (1,M) on Stud\_Service side, mandatory on both sides.

***Rule 5: Sub Entities.***

All students share common attributes(first name, last name, admit, phone number) but few students have unique attributes(for instance, labs and thesis). Student is a parent entity with Undergraduate and Graduate entities as sub entities. The sub entities show a partial participation because there are many other student types like Ph. D students. This is a disjoint sub entity because the attributes of sub entities do not overlap.

***DBMS Architecture:***

For the Student Housing and Residence Life database, a three-tier architecture would be apt. It is because, this database needs data to be accessed, storing data and maintenance. A three-tier architecture also provides a great flexibility to the team managing the database if there is any need to update data in database. I would not consider other architectures because of the performance issues on increasing data among users, cost inefficiency and other architecture’s do not provide remote access.

**Visio diagram:**

**A screenshot of a computer

Description automatically generated**