Northeastern University
DAMG6210 - Data Management and Database Design
Fall ‘21 - **TEAM 7**

Project Title: A database design to keep track of the **Campus Housing System**.

Problem Statement:

The Campus Housing System wants to create a database to keep track of the Management system which consists of **five** major entities viz. Dorms, Residents, Management Admin, Proctors, and Police Department. The Campus Housing System **helps** schools to **manage** on-campus housing and regulates the management hierarchy in a properly aligned fashion.

Objectives:

In the Campus Housing System, Admin can allocate students to specific dorms based on the availability of the student. Also, the proctor allocation takes place automatically at the first instance of every week.

The **hierarchy** goes as follows:

- Dorms are supervised by Proctors.
- Management Admin supervises Proctor and Dorms.
- The Police Department manages Residents, Dorms, Proctors, and Admins in case of emergencies.
- Residents' entries are handled by Proctors.
- Also, Proctors manage Guests in Dormitories.
- Utilities like Laundry, Gym, Kitchen, Break Rooms, Study Rooms are supervised by Proctors.

Team Information :

TEAM 7		
NAME	**E-MAIL**	**NUID**
Vidhi Popat	popat.v@northeastern.edu	002135187
Prathamesh Nemade	nemade.p@northeastern.edu	002139730
Urvang Patel	patel.ur@northeastern.edu	001568243
Milind Sharma	sharma.mil@northeastern.edu	001090358

Northeastern University

DAMG6210 - Data Management and Database Design
Fall ‘21 - **TEAM 7**

Project Title: A database design to keep track of the **Campus Housing System**.

Purpose:
The purpose of this database design is to help the university to **manage and maintain** the on-campus houses for the students. The design takes into account the student assignment to dorms, provides proctors to the dorm, and handles emergencies.

Problem Statement:
Institutes need to manage and keep the track of the dorms and the residents staying on campus. Institutes need help with assigning the students to the dorms and managing the security. The students living on campus should be provided with good living conditions and manage all the facilities/utilities. Therefore, this model - On-Campus Housing helps the institutes/clients to manage their housing system.

The major problems that can be resolved or can be reconciled by this system are as follows:

1. **Allocation of the students from institutes as residents made easy.**
2. **Management of tracking residents’ logs at their respective residents can be achieved by this system.**
3. **Allocation of shifts based on the type (related to time, based on the part of the day) to the proctor by the top management committee for the specific date and time interval made feasible.**
4. **In case of emergencies, escalation of the issue to the relevant hierarchy body on time can be worked out.**
5. **Proper management and usage of the utilities at respective Dormitories were made simple.**
6. **Keeping a track of the guests (who and how’s) at the Dormitories can be made possible with this model.**

Problem Solution:
Following are the solutions to the above problems that can be resolved by the usage of this model, respectively -

1. Ideally, every student won’t be a resident, the only students are the residents who have their **is_resident** flag enabled from the **Student** table. This flag segregates the students from the residents.
2. Activity (enter time) of the Residents can be tracked from the **Shift Logs** table based on the **time** and **resdient_id** attributes.
3. Mainly the allocation of shifts from a pre-defined (also, can be configurable as well) table **Shift Type Master** can be done by the Supervisor from the Supervisor table to the Proctors can be done by in the **Shifts** table.

4. Based on the emergencies which have been leveraged/raised from the **Dorm** table to the **Case** table, the available Police office can be mapped in the **Case Mapping** table.
5. Usage of the utilities present from the **Utilities Master** table for the respective dormitories can be allocated to the resident in the **Utility** table.

Guests visiting the dorm can be shadowed from the **Guest's** table. Precisely, **date** and the **resident_id** attributes facilitate keeping track of the residents and guests visited.

Database Design Decision:

- When a student enrolls in university and opts for on-campus housing, the student will be automatically assigned a dorm room. Dorm mappings are stored in the **Resident** table.
- The student can access the assigned dorm by swiping their ID cards. The login data is stored in the **Swipe Log** table.
- Students can bring guests with them, and they need to log the guests in before entering the dorm. Guest details will be stored in the **Guest** table.
- Dorms will have some utilities which can be accessed by the residents. Utility details will be stored in the **Utility** table.
- Supervisors will manage the Shift assignments for the dorms and proctors. The assigned shift for proctors and dorm details are stored in the **Shifts** table.
- In emergencies, a case will be raised for the respective dorm. The details will be stored in the **Case** table.
- For each case raised, one or more police will be assigned to handle the situation. The details of this mapping will be stored in the Police **Case Mapping** table.

Facts:

- Dorms are supervised by Proctors.
- The Police Department manages Residents, Dorms, Proctors, and Admins in case of emergencies.
- Residents' entries are handled by Proctors.
- Proctors manage Guests in Dormitories.
- The student won't be admitted to a dorm if the capacity of the respective dorm exceeds.
- Students can bring in multiple guests.

Assumptions:

1. Students can have access to only one assigned dorm.
2. Multiple proctors can be assigned to a single dorm.
3. One resident is allocated to one dorm only, no switching of the dorm is allowed.
4. Supervisors can issue/assign multiple shifts to a proctor.
5. Not all students are residents.
6. Students are not proctors and vice-versa.
7. Shift types, Utility types, and Case types come from a predefined set of data respectively.

Database Identification:

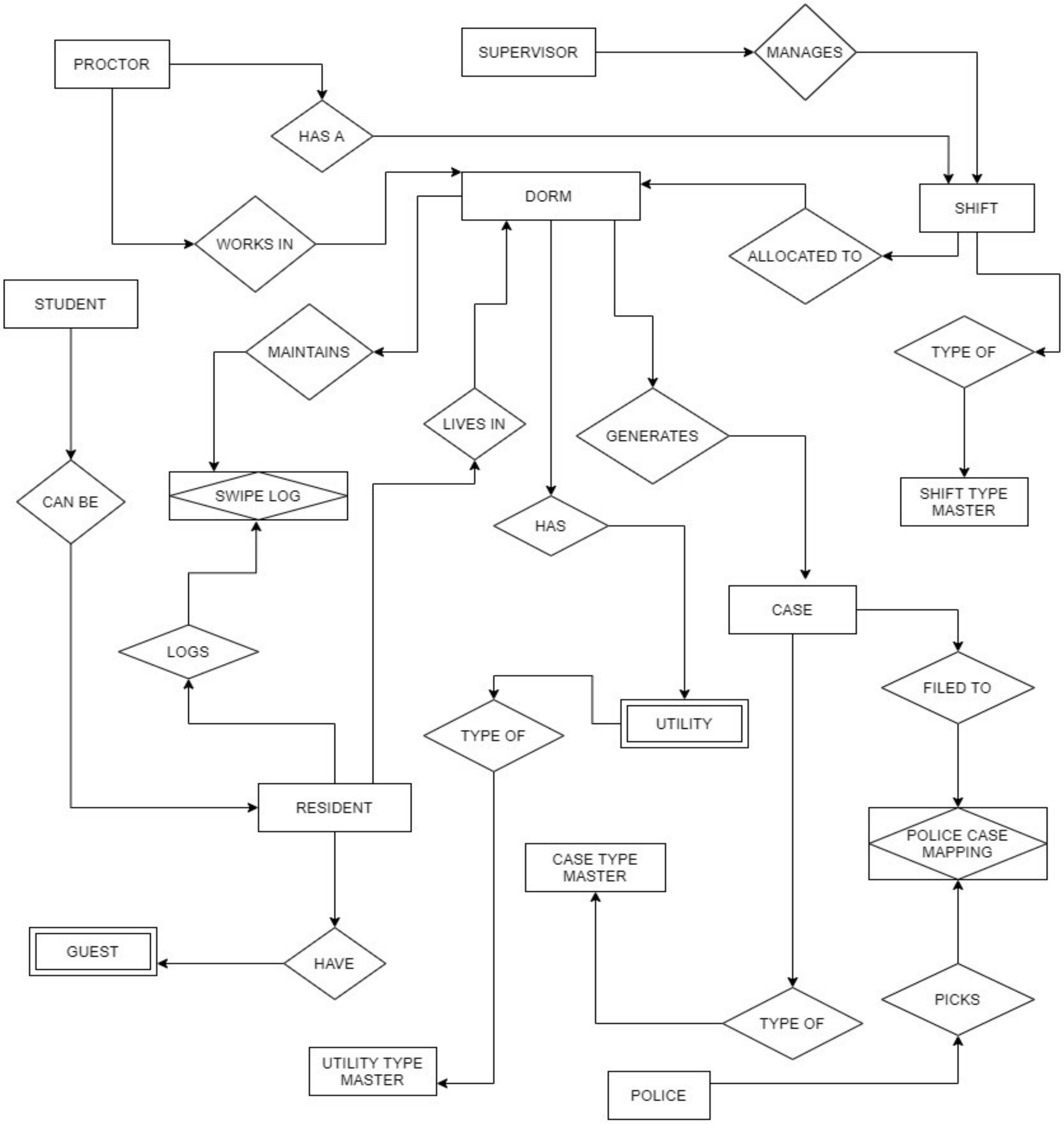
Following are the Main Entities -

Dorms, Proctors, Supervisor, Police, and Residents.

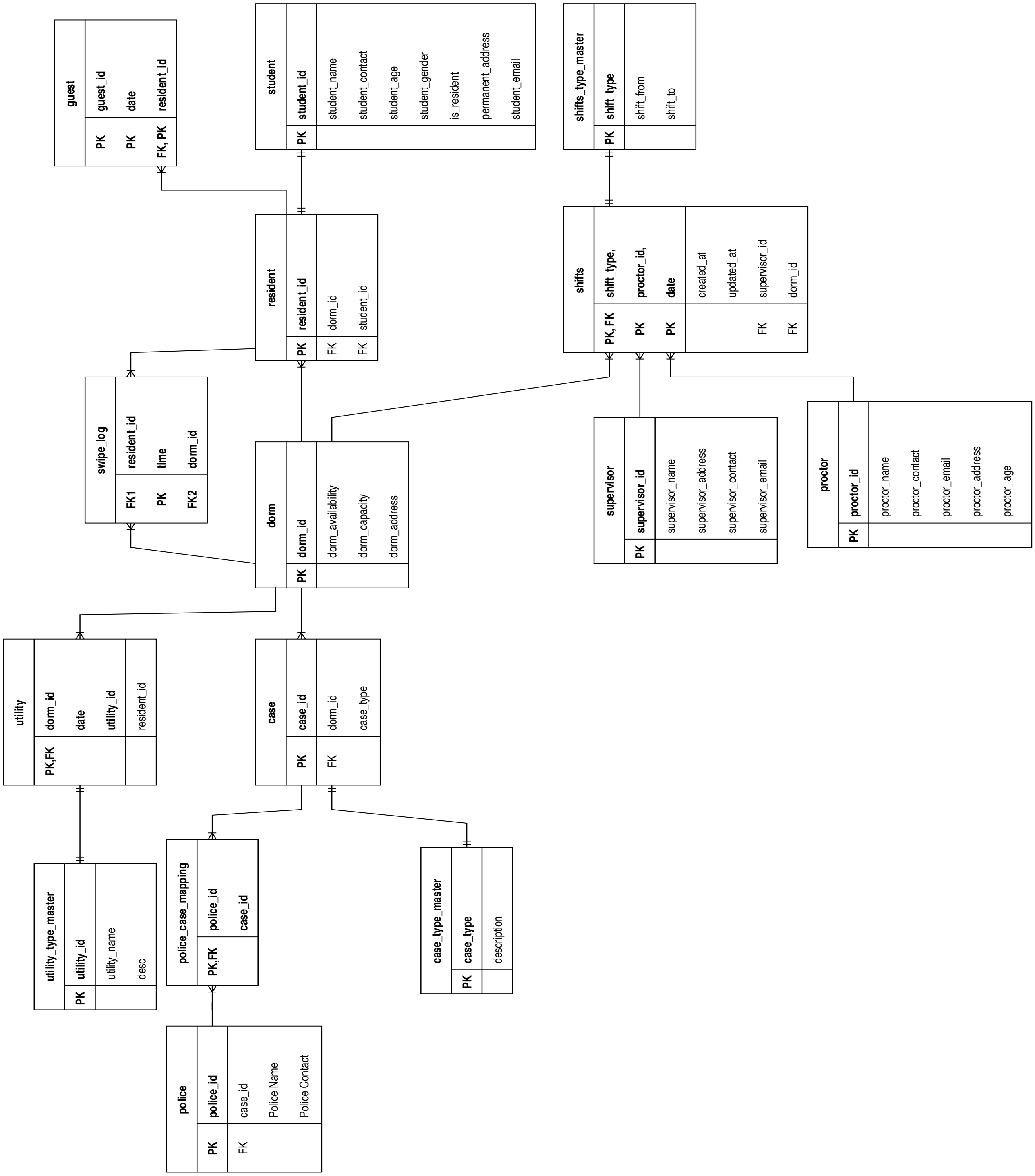
Moreover, the above entities defined a few major attributes which are the crucial ones for working of the On-Campus Housing Model, they are -

Entity Name	**Attributes**	**Description/Meaning**
Proctor	proctor_id	Proctor ID is a unique identifier assigned to each proctor.
Dorm	dorm_id	Dorm ID is a unique identifier assigned to each dorm.
Supervisor	supervisor_id	Supervisor ID is a unique identifier assigned to each supervisor.
Police	police_id	Police ID is a unique identifier assigned to each police.
Resident	resident_id	Resident ID is a unique identifier assigned to each resident.
Utility	utility_id	Utility ID is a unique identifier assigned to each utility.
Shift	shift_type	Shift type is used to uniquely identify selected shifts. For example shift_type = M01 means morning shift 1.
Swipe_Log	resident_id + dorm_id + time	All three attributes i.e resident_id, dorm_id, and time together are used to uniquely identify residents swipe in. This entity is used to maintain the log of each resident.

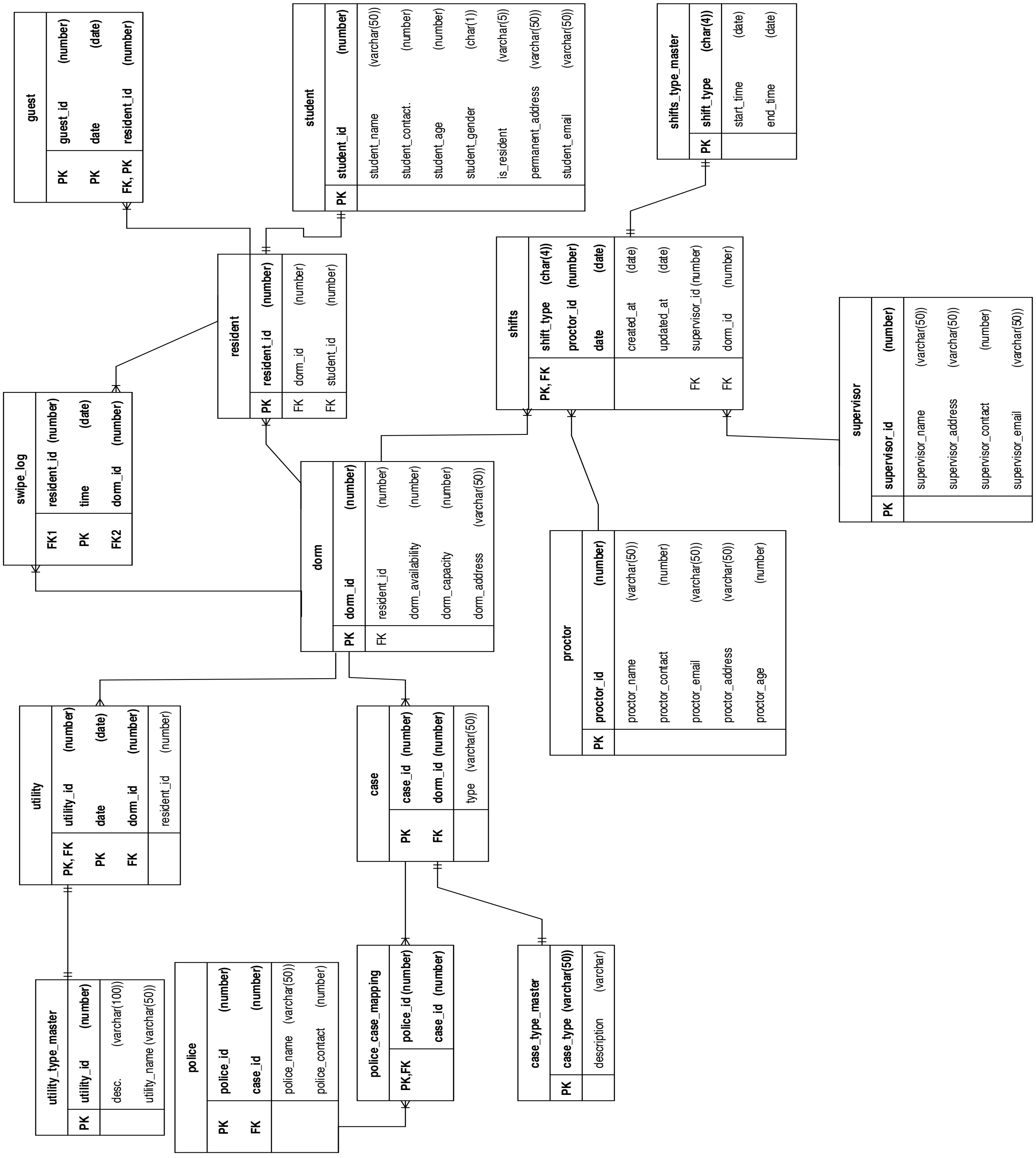
Conceptual Diagram:



Logical Diagram:



Physical Diagram:



Following are the Entities and Attributes used to develop an On-Campus Housing System -

Dorm:

The entity that stores the details about all the dorm rooms in the institute.

Key	Attribute Name	Description	Data Type
PK	dorm_id	A primary key, unique value allocated to each dormitory	number
FK	resident_id	A foreign key, unique value is allocated to each resident	number
	dorm_availablity	Dorm Availability is the current state of the dorm. This conveys the total consummation to date	number
	dorm_capacity	Dorm capacity returns the total capacity the respective dorm holds	number
	dorm_address	Dorm address is the location/brief address of the respective dormitory	varchar(50)

Proctor:

The entity that stores the details about all the proctors employed by the institute.

Key	Attribute Name	Description	Data Type
PK	proctor_id	A primary key, unique value allocated to each proctor	number
	proctor_name	Name of the proctor	varchar(50)
	proctor_contact	The contact number of the proctor	number
	proctor_email	Email address of the proctor	varchar(50)
	prcotor_address	Address of the proctor	varchar(50)
	proctor_age	Age of the proctor	number

Student:

A master data of all the students enrolled in the institute.

Key	**Attribute Name**	**Description**	**Data Type**
PK	student_id	Students unique ID number	number
	student_name	Name of the student	varchar(50)
	student_contact	The contact number of the student	number
	student_age	Age of the student	number
	student_gender	Gender of the student	char(1)
	is_resident	Did the student opt for on-campus residence	varchar(5)
	permanent _address	Students permanent address	varchar(50)
	student_email	Students email address	varchar(50)

Resident:

The entity that maps the residents with their respective assigned dorm rooms.

Key	**Attribute Name**	**Description**	**Data Type**
PK	resident_id	A unique value is allocated to each resident	number
FK1	dorm_id	A foreign key, unique value allocated to each dormitory	number
FK2	student_id	A foreign key, unique value allocated to the student	number

Supervisor:

The entity that stores all details of the supervisor who allocates shifts to each proctor.

Key	**Attribute Name**	**Description**	**Data Type**
PK	supervisor_id	A unique value is allocated to each supervisor	number
	supervisor_name	Name of the supervisor	varchar(50)
	supervisor_address	Address of the supervisor	varchar(50)
	supervisor_contact	The contact number of the supervisor	number
	supervisor_email	Email address of the supervisor	varchar(50)

Shifts:

The entity which stores details about the shifts includes: which proctor has been allocated which shift, at what time, which dorm, and which supervisor assigned that shift.

Key	Attribute Name	Description	Data Type

PK	shift_type	Shift type can be following:
		M01 = Morning shift 1
		M02 = Morning shift 2
		M03 = Morning shift 3
		A01 = Afternoon shift 1
		E01 = Evening shift 1
		E02 = Evening shift 2
			char(4)

PK, FK	proctor_id	ID of the proctor who was assigned the shift	number
PK, FK	date	Shift date	date
	created_at	Shift creation date	date
	updated_at	Shift updated date	date
FK	supervisor_id	The ID of the supervisor who assigned the shift	number
FK	dorm_id	Dorm ID in which the shift is assigned	number

Shift Master:

A master data of all the shift type's start and end times.

Key	Attribute Name	Description	Data Type

PK	shift_type	Shift type can be following:
		M01 = Morning shift 1
		M02 = Morning shift 2
		M03 = Morning shift 3
		A01 = Afternoon shift 1
		E01 = Evening shift 1
		E02 = Evening shift 2
			char(4)

	shift_from	Start date and time of the sift	date
	shfit_to	End date and time of the sift	date

Utility:

The entity that keeps track of the residents who use the utility at which dorm.

Key	Attribute Name	Description	Data Type
PK	utility_id	A unique value is allocated to each utility	number
FK1	resident_id	A unique value is allocated to each resident	number
FK2	dorm_id	A unique value is allocated to each dormitory	number

Utility Master:

A master data of utility used to store names and IDs of it.

Key	Attribute Name	Description	Data Type
PK	utility_id	A unique value is allocated to each utility	number
	utilit_name	Name of the utility	varchar(100)

Case:

The entity which stores the details of the emergency case raised, and in which dorm it was raised.

Key	Attribute Name	Description	Data Type
PK	case_id	A unique case ID for police cases	number
FK	dorm_id	Dorm ID which is related to the case	number
	type	Type of emergency	varchar(50)

Case Master:

Master dataset which stores the detailed descriptions of each case type.

Key	Attribute Name	Description	Data Type
PK	case_type	Type of emergency	number
	description	Description of all available utilities	varchar(50)

Guest:

The entity that stores details about the residents and their guests.

Key	Attribute Name	Description	Data Type
PK	guest_id	A unique value is allocated to each guest	number
PK	date	Guest arrived date	date
FK, PK	resident_id	A unique value is allocated to each resident	number

Police:

The entity that stores the details about the police officers employed by the institute.

Key	Attribute Name	Description	Data Type
PK	police_id	A unique value allocated to each police	number
FK	case_id	A unique value allocated to each case type	number
	police_name	Name of the Police officer	varchar(50)
	police_contact	The contact number of the Police officer	number

Swipe Logs:

The entity which logs the resident's activity. A new log is created every time a resident enters the dorm.

Key	Attribute Name	Description	Data Type
PK	time	Log timing - Time of the entry made	date
FK1	resident_id	A unique value is allocated to each resident	number
FK2	dorm_id	A unique value is allocated to each dormitory	number

Police Case Mapping:

Bridge table used to map the police with the ongoing cases

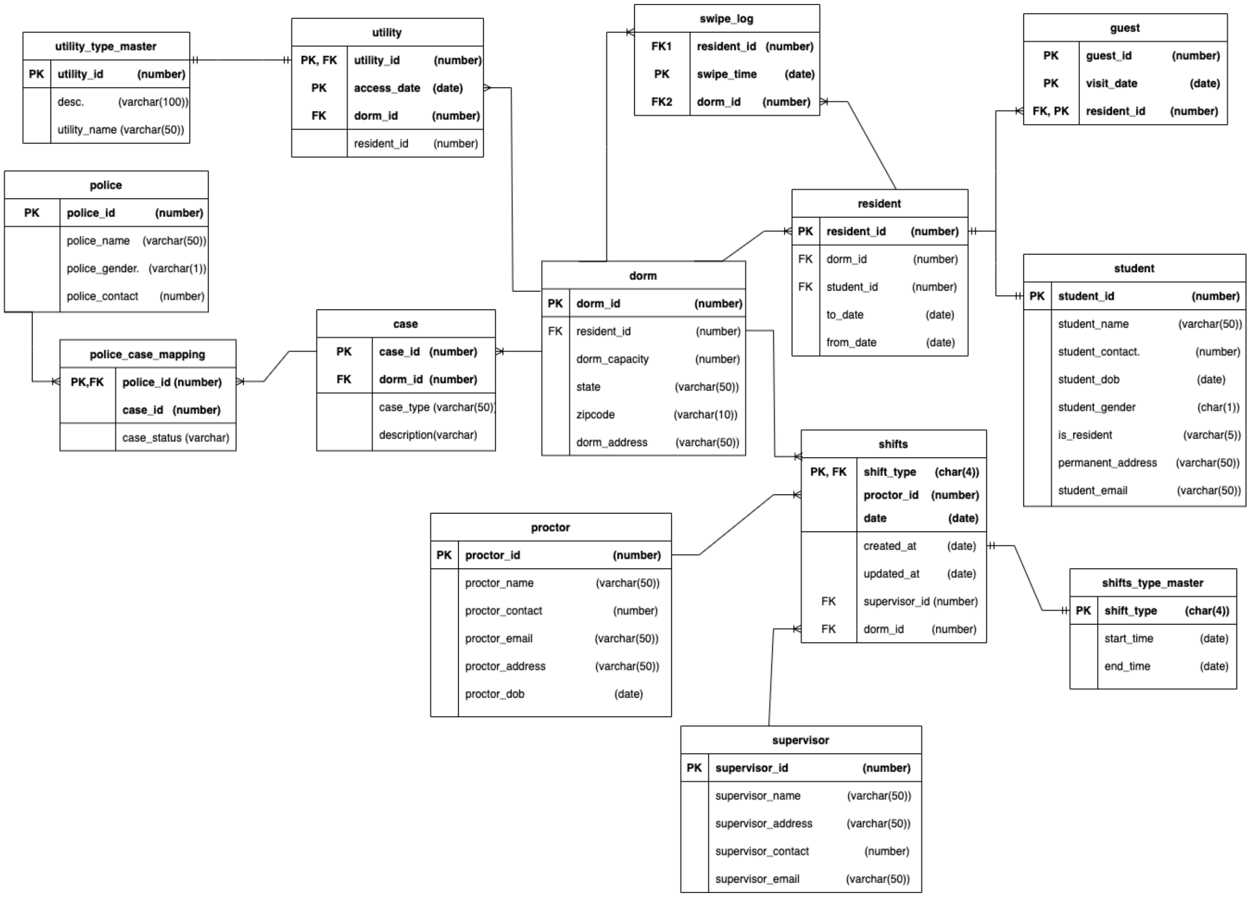
Key	Attribute Name	Description	Data Type
PK, FK	police_id	A unique value is allocated to each police	number
	case_id	A unique value is allocated to each case type	number

Northeastern University

DAMG6210 - Data Management and Database Design
Fall ‘21 - **TEAM 7**

Project Title: A database design to keep track of the **Campus Housing System**.

Final ER Diagram:



Business Rules:

1. Only a **STUDENT** can be a **RESIDENT**. **RESIDENTS** stay for a specific period.
2. **RESIDENT** is allocated to a specific **DORM**. Only one **DORM** is assigned to a **STUDENT**.
3. Every **DORM** has its own capacity and bases on this a **STUDENT** will be allocated.
4. **PROCTORS** are allocated a shift by a **SUPERVISOR**. Every **SHIFT** is of a specific type which are pre-defined in **SHIFT_TYPE_MASTER**. Ex. S1 – 9:00AM – 12:00PM, S2 – 12:00PM – 15:00PM, etc.
5. **PROCTOR** is not **STUDENT**.
6. **SHIFTS** are only allocated to **PROCTORS**.
7. **PROCTOR** is responsible for capturing **SWIPE_LOGS.** Also, they are responsible to track the **GUESTS**, accompanied by **RESIDENTS**.
8. **SUPERVISOR** are the only ones who allocated **SHIFTS** to **PROCTORS**. Also, they are the ones who manages the capacity of **DORM**.
9. Every **DORM** has its own **UTILITY**. The utilities can only be used which are pre-defined from **UTILITY_TYPE_MASTER.** Ex. Laundry, Gaming Room, etc.
10. In case of emergency, the **CASES** are being raised from specific **DORM**.
11. Every **CASE** is allocated to first available **POLICE** officer.

User level access and permissions:

1. Admin:

- ADMIN is the root user of the Database.
- ADMIN has full access to modify the Database.
- ADMIN can CREATE, VIEW, UPDATE and DELETE the tables.

	Create	View	Update	Delete
Student	✅	✅	✅	✅
Resident	✅	✅	✅	✅
Guest	✅	✅	✅	✅
Shifts	✅	✅	✅	✅
Supervisor	✅	✅	✅	✅
Swipe Logs	✅	✅	✅	✅
Proctor	✅	✅	✅	✅
Dorms	✅	✅	✅	✅
Utility / Utility Master	✅	✅	✅	✅
Case	✅	✅	✅	✅
Police	✅	✅	✅	✅

2. Supervisor

- Supervisors can only VIEW the list of enrolled students.
- Supervisor can allocate/create dorms to residents, VIEW the allocations, UPDATE them, and DELETE them.
- Supervisors can monitor/VIEW the guest lists.
- Supervisor assigns/create shifts, VIEW them, UDPATE the shifts if needed and DELETE them.
- Supervisors can VIEW all the other supervisors name and contact number.
- Supervisors can monitor/VIEW the residents swipe logs.
- Supervisors hire/create new proctors, VIEW the proctors list, UPDATE the proctors details and DELETE the proctors.
- Supervisor can create new dorms, VIEW the list of dorms and their availability, UPDATE the dorm, and DELETE the dorm.
- Supervisor can create utility, VIEW them, UPDATE and DELETE the utilities.
- Supervisors can VIEW the police cases raised.
- Supervisors can VIEW the police name and their contact numbers.

	Create	View	Update	Delete
Student		✅		
Resident	✅	✅	✅	✅
Guest		✅		
Shifts	✅	✅	✅	✅
Supervisor		✅		
Swipe Logs		✅		
Proctor	✅	✅	✅	✅
Dorms	✅	✅	✅	✅
Utility / Utility Master	✅	✅	✅	✅
Case		✅		
Police		✅		

3. Resident:

- RESIDENTS can only VIEW the Students, Residents, Dorms and Utility's data.
- RESIDENTS has no privilege to modify any kind of data.
- RESIDENTS can view the data of other residents staying in the same or a different dorm. The only data will be visible which is allowed by a VIEW created. Ex. Only a Name, Contact and Email will be visible.
- Similarly, RESIDENTS can only VIEW the data of Utility and Dorms based on the independent views created.
- Ex: A resident can only view the attributes - dorm capacity, dorm address attributes.

	Create	View	Update	Delete
Student		✅		
Resident		✅		
Guest				
Shifts				
Supervisor				
Swipe Logs				
Proctor				
Dorms		✅		
Utility / Utility Master		✅		
Case				
Police				

4. Proctor

- PROCTOR can only VIEW the data based on the Views created separately for the sensitive entities. Ex: Contact no., Address, duration of residence, etc. won’t be displayed to a Proctor.
- PROCTOR can VEIW, CREATE and UPDATE a Guest’s entry based on the action needed. Once an entry is created, PROCTOR cannot DELETE it.
- PROCTOR can only VIEW and CREATE a swipe log once a swipe made by a RESIDENT. Once a log is captured, PROCTOR has no privilege to UPDATE or DELETE it.
- Based on the Emergencies, PROCTOR can CREATE a case with a proper description and type. PROCTOR cannot DELETE a case, once created.

	Create	View	Update	Delete
Student		✅		
Resident		✅		
Guest	✅	✅	✅	
Shifts		✅		
Supervisor		✅		
Swipe Logs	✅	✅		
Proctor		✅		
Dorms		✅		
Utility / Utility Master		✅		
Case	✅	✅	✅	
Police				

5. Police

- POLICE can VIEW the supervisor names and contact information.
- POLICE can VIEW the Proctors names and contact information.
- POLICE can VIEW the dorm details.
- POLICE can VIEW the cases and UPDATE them when the case is resolved.

	Create	View	Update	Delete
Student				
Resident				
Guest				
Shifts				
Supervisor		✅		
Swipe Logs				
Proctor		✅		
Dorms		✅		
Utility / Utility Master				
Case		✅	✅	
Police				