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 systems 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 Dormataries were made simple.**
6. **Keeping a track of the guests(who and how’s) at the Dormatries 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 are 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 theMain 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 -

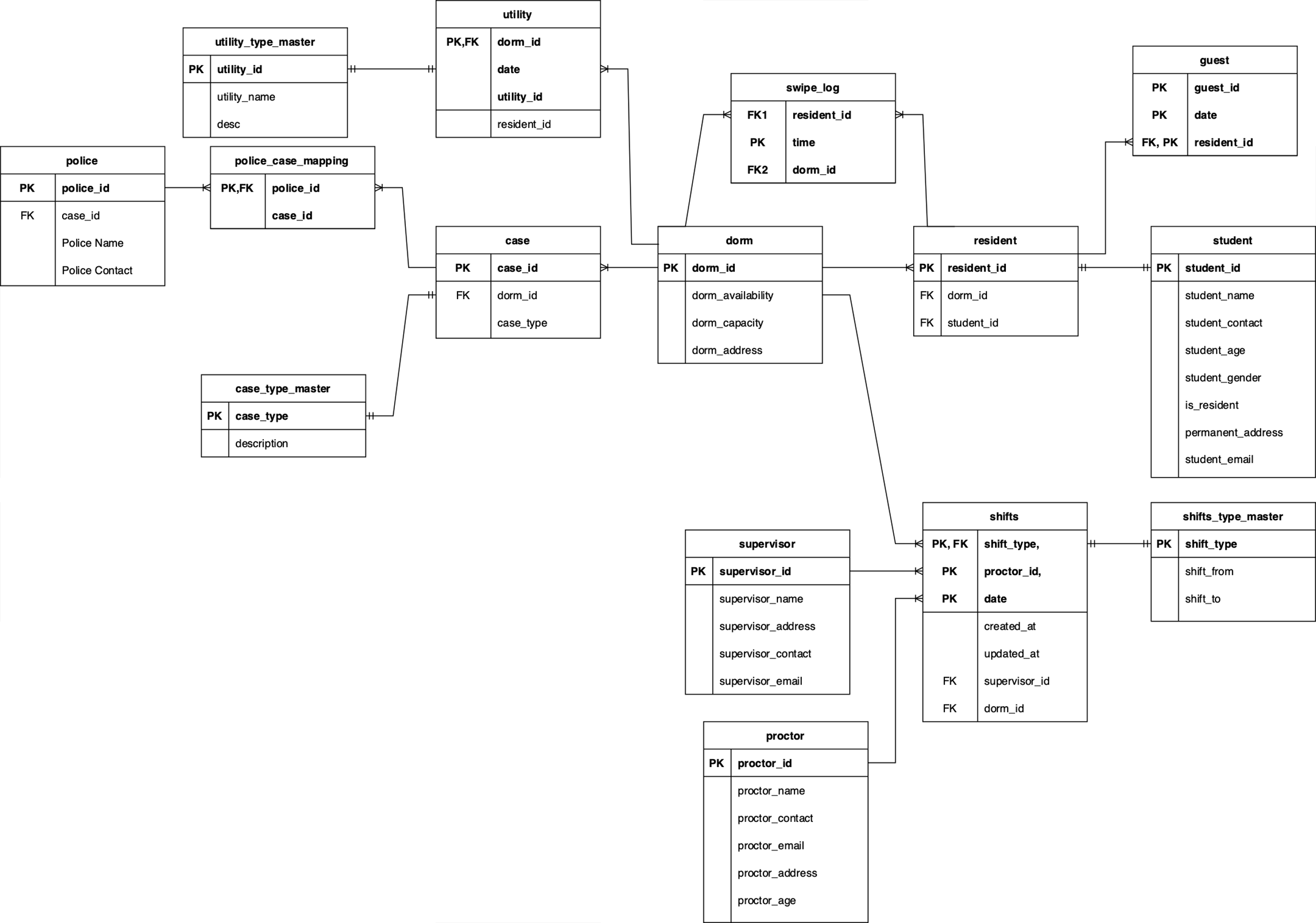
|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Attributes** | **Description/Meaning** |
| Proctor | proctor\_id | Proctor ID is a unique identifier assigned to each proctor. |
| Dorm | dorm\_id | Dorm ID is a unique identifier assigned to each dorm. |
| Supervisor | supervisor\_id | Supervisor 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:**

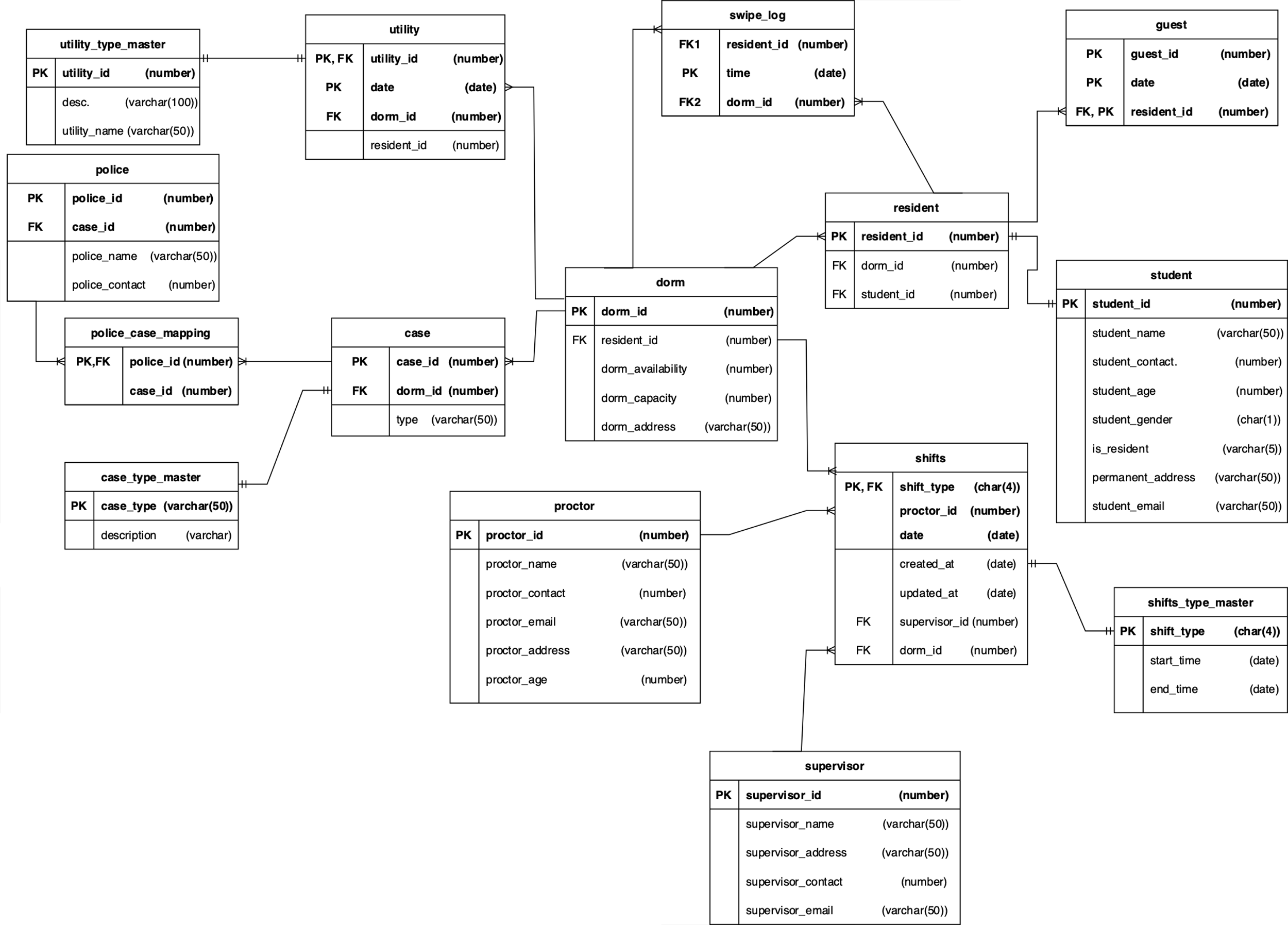
Diagram

Description automatically generated

**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 |