# On Campus Accommodation Office

Nima Khani Amin Abadi

Aayush Maharjan

Rasila Thapa

Muaaz Wahid

San Francisco Bay University

CS 457: Database Technologies Lab

Summer 2024

Prof. Dharmale

July 23, 2024

# Table of Contents

## Scenario

The director of the ***On Campus Accommodation Office*** requires you to design a database to assist with the administration of the office. The requirements collection and analysis phase of the database design process has provided the following data requirements specification for the *On Campus Accommodation Office* database followed by examples of query transactions that should be supported by the database.

**Students**

The data stored for each full-time student includes: the banner number, name (first and last name), home address (street, city, postcode), mobile phone number, email, date of birth, gender, category of student (for example, first-year undergraduate, postgraduate), nationality, special needs, any additional comments, current status (placed/waiting), major, and minor.

The student information stored relates to those currently renting a room and those on the waiting list. Students may rent a room in a hall of residence or student apartment.

When a student joins the university, he or she is assigned to a member of staff who acts as his or her Adviser. The Adviser is responsible for monitoring the student's welfare and academic progression throughout his or her time at the university.

The data held on a student's Adviser includes full name, position, name of department, internal telephone number, email, and room number.

**Halls of residence**

Each hall of residence has a name, address, telephone number, and a hall manager,who supervises the operation of the hall. The halls provide only single rooms, which have a room number, place number, and monthly rent rate.

The place number uniquely identifies each room in all halls controlled by the Residence Office and is used when renting a room to a student.

**Student flats**

The Residence Office also offers student apartments. These are fully furnished and provide single-room accommodation for groups of three, four, or five students. The information held on student apartments includes an apartment number, address, and the number of single bedrooms available in each apartment. The flat number

uniquely identifies each apartment.

Each bedroom in an apartment has a monthly rent rate, room number, and a place number. The place number uniquely identifies each room available in all student apartments and is used when renting a room to a student.

**Leases**

A student may rent a room in a hall or student apartment for various periods of time. New lease agreements are negotiated at the start of each academic year, with a minimum rental period of one semester and a maximum rental period of one year, which includes semesters 1 and 2 and the summer semester. Each individual lease agreement between a student and the Residence Office is uniquely identified using a lease number.

The data stored on each lease includes the lease number, duration of the lease (given as semesters), student's name and banner number, place number, room number, address details of the hall or student apartment, and the date the student wishes to enter the room, and the date the student wishes to leave the room (if known).

**Invoices**

At the start of each semester, each student is sent an invoice for the following rental period. Each invoice has a unique invoice number.

The data stored on each invoice includes the invoice number, lease number, semester, payment due, student's full name and banner number, place number, room number, and the address of the hall or apartment. Additional data is also held regarding the payment of the invoice and includes the date the invoice was paid, the method of payment (check, cash, Visa, and so on), the date the first and second reminder was sent (if necessary).

**Student apartment inspections**

Student apartments are inspected by staff on a regular basis to ensure that the accommodation is well maintained. The information recorded for each inspection is the name of the member of staff who carried out the inspection, the date of inspection, an indication of whether the property was found to be in a satisfactory condition (yes or no), and any additional comments.

**Residence staff**

Some information is also held on members of staff of the Residence Office and includes the staff number, name (first and last name), email, home address (street, city, postcode), date of birth, gender, position (for example, Hall Manager, Administrative Assistant, Cleaner) and location (for example, Residence Office or Hall).

**Courses**

The Residence Office also stores a limited amount of information on the courses offered by the university, including the course number, course title (including year), course instructor, instructor's on-campus telephone number, email, room number, and department name. Each student is also associated with a single programme of studies.

**Next-of-kin**

Whenever possible, information on a student's next-of-kin is stored, which includes the name, relationship, address (street, city, postcode), and contact telephone number.

## Abstract

The On Campus Accommodation Office database system is designed to streamline and enhance the management of student housing within a university setting. The primary goal of this project is to develop a database system that addresses various operational needs related to student accommodation, including tracking student occupancy, managing leases, handling financial transactions, and ensuring proper maintenance of housing facilities thus facilitating students to excel.

## Problem Statement

The current administrative process for managing student accommodation is fragmented and inefficient, leading to difficulties in tracking student occupancy, managing lease agreements, and ensuring timely maintenance of accommodation facilities. The lack of an integrated system results in data redundancy, delayed processing, and potential inaccuracies in student records and financial transactions.

## Approach

To address these challenges, we propose a  database system that consolidates all relevant information into a unified platform. The database will include tables for students, advisers, halls of residence, student apartments, leases, invoices, inspections, residence staff, courses, and next-of-kin.

Module1:

1. **Data Entry and Update through forms** : Implement user-friendly forms for entering and updating data, ensuring that information is accurately recorded and easily accessible.

2. **Data Reporting**: Generate various reports such as lease reports, invoice reports, financial summaries, and special needs occupancy reports to facilitate effective decision-making and ensure timely processing of administrative tasks.

Module 2:

1. **Designing Data Dictionary**: Provides precise definitions and attributes for each data element, ensuring consistency and accuracy in data management. It simplifies maintenance and querying by offering a clear reference for understanding and using each data field.
2. Creating Cross reference table: maps out the relationships between different tables/forms, facilitating accurate and efficient multi-table queries.

# Forms

**Student**

- BannerNumber (Primary key)
- Name (First and last name)
- Address (Street, City, Postcode)
- MobilePhoneNumber
- Email
- DateOfBirth
- Gender
- Category (e.g., First-year Undergraduate, Postgraduate)
- Nationality
- SpecialNeeds
- Comments
- CurrentStatus (Placed/Waiting)
- Major
- Minor
- Room (“Hall of residence” or “Student apartment”)
- AdviserID (Foreign Key to Adviser)
- NextOfKinID (Foreign Key to NextOfKin)

- **Adviser**
  - AdviserID (Primary Key)
  - Name
  - Position
  - DepartmentName
  - InternalPhoneNumber
  - Email
  - RoomNumber

- **Hall Of Residence**
  - HallName (UNIQUE)
  - Address (Street, City, Postcode)
  - MobilePhoneNumber
  - StaffNumber (Foreign Key to ResidenceStaff)

- **Room**
    - PlaceNumber (Unique across all rooms PRIMARY KEY)
    - RoomNumber
    - MonthlyRent
    - Type (“Hall of Residence” or “Student Apartment”)

- **Student Apartment**
    - Apartment Number(Primary Key)
    - Address (Street, City, Postcode)
    - Number of rooms

- **Lease**
    - Lease Number (Primary Key)
    - Duration (Semesters 1, 2 and the summer semester)
    - BannerNumber (Foreign Key to Student)
    - StudentName
    - PlaceNumber (Foreign Key to Room)
    - RoomNumber
    - Address
    - StartDate
    - EndDate (Null depending on if the student knows the move out date)

- **Invoice**
    - InvoiceNumber (Primary Key)
    - LeaseNumber(Foreign Key to Lease)
    - Semester
    - Name
    - BannerNumber (Foreign Key to Student)
    - PlaceNumber
    - RoomNumber
    - Address
    - PaymentDue
    - MethodOfPayment (check, cash, Visa, and so on or can be Null)
    - PaymentDate (Nullable)
    - FirstReminderDate (Nullable)
    - SecondReminderDate (Nullable)

- **Inspection**
    - StaffNumber (Foreign Key to ResidenceStaff)
    - Name
    - ApartmentNumber
    - InspectionDate
    - Satisfactory (Yes/No)
    - Comments

- **Residence Staff**
    - StaffNumber (Primary Key)
    - Name (first and last name)
    - Email
    - Address (Street, City, Postcode)
    - DateOfBirth
    - Gender
    - Position (for example, Hall Manager, Administrative Assistant, Cleaner)
    - Location (Residence Office or Hall)

- **Course (DONE)**
    - CourseNumber
    - CourseTitle (Including year)
    - InstructorName
    - InstructorPhoneNumber
    - InstructorEmail
    - InstructorRoomNumber
    - DepartmentName

**Next Of Kin**

- NextOfKinID (Primary Key)
- Name
- Relationship
- Address (Street, City, Postcode)
- ContactPhoneNumber

# Student Form

Banner number: _____

Name: _____

Home Address: _____

Mobile Phone Number: _____

Email: _____

Date Of Birth: _____

Gender: _____

Category: Undergraduate [ ] Graduate [ ]

Nationality: _____

Special Needs: _____

Comments: _____

Current Status: Placed [ ] Waiting[ ]

Major: _____

Minor: _____

Room: Hall of Residence [ ] Student Apartment [ ]

Adviser ID: _____

NextOfKin ID: _____

# Adviser Form

Adviser ID: _____

Name: _____

Position: _____

Department Name: _____

Internal Phone Number: _____

Email: _____

Room Number: _____

# Hall of Residence Form

Hall Name: _____

Address: _____

Mobile Phone Number: _____

Staff Number: _____

# Student Apartment Form

Apartment Number: _____

Address: _____

Number of Rooms: _____

# Room Form

Place Number: _____

Room Number: _____

Monthly Rent: _____

Type: Hall of Residence [ ] Student Apartment [ ]

# Invoice Form

Invoice Number: _____

Lease Number: _____

Semester: _____

Name: _____

Banner Number: _____

Place Number: _____

Room number: _____

Address: _____

Payment Due: _____

Method of payment: check [ ] cash [ ] Visa [ ] others [ ]

Payment Date: _____

First Reminder Date: _____

Second Reminder Date: _____

# Lease Form

Lease Number: _____

Duration: _____

Banner Number: _____

Student's Name: _____

Place Number: _____

Room Number: _____

Address: _____

Start Date: _____

End Date: _____

# Inspection Form

Staff Number: _____

Name: _____

Apartment Number: _____

Inspection Date: _____

Satisfactory: Yes [ ] No [ ]

Comments: _____

# Residence Staff Form

Staff Number: _____

Name: _____

Email: _____

Address: _____

Date Of Birth: _____

Gender: _____

Position: _____

Location: Residence Office [ ] Hall [ ]

# Course Form

Course Number: _____

Course Title: _____

Instructor Name: _____

Instructor Phone Number: _____

Instructor Email: _____

Instructor Room Number: _____

Department Name: _____

# Next of Kin Form

Next Of Kin ID: _____

Name: _____

Relationship: _____

Address: _____

Contact Phone Number: _____

# Reports

## On Campus Accommodation

### Student Status Report

| Student Banner | Student Name | Status | Place Number | Room Type | Adviser Name | Comments |
|---|---|---|---|---|---|---|
| 1001 | Diane Feneck | Placed | 101 | Hall of Residence | Dr. Smith | None |
| 1010 | Brachett Lee | Waiting | | | Dr. Johnson | Awaiting room |

# On Campus Accommodation

## Lease Report

| Lease Number | Student Banner | Student Name | Place Number | Room Number | Room Type | Address | Duration | Start Date | End Date |
|---|---|---|---|---|---|---|---|---|---|
| 001 | 1001 | Daine Feneck | 101 | H1 | Hall of Residence | 123 Hall St, City | 1 | 2024-01-15 | 2024-05-15 |
| 002 | 1002 | Mellisa Black | 202 | A2 | Student Apartment | 456 Apartment Rd, City | 2 | 2024-01-15 | N/A |

# On Campus Accommodation

## Room Inspection Report

| Staff Number | Name | Apartment Number | Inspection Date | Satisfactory | Comments |
|---|---|---|---|---|---|
| 001 | Alice Johnson | 101 | 2024-01-15 | Yes | |
| 002 | Bob Smith | 102 | 2024-01-10 | No | Leak in bathroom |
| 001 | Alice Johnson | 101 | 2024-03-15 | Yes | |
| 002 | Bob Smith | 102 | 2024-03-10 | Yes | Fixed leak |

# On Campus Accommodation

## Invoice Report

| Invoice Number | Lease Number | Semester | Student Name | Student Banner | Place Number | Room Number | Room Type | Address | Payment Due | Method of Payment | Payment Date | First Reminder Date | Second Reminder Date |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1001 | L001 | Spring | Diane Feneck | 1001 | 101 | H1 | Hall of Residence | 123 Hall St, City | $800 | Cash | 2024-01-20 | 2024-01-27 | 2024-02-03 |
| 1002 | L002 | Spring | Mellissa Black | 1002 | 202 | A2 | Student Apartment | 456 Apartment Rd, City | $900 | Visa | 2024-01-22 | 2024-01-29 | 2024-02-05 |

# On Campus Accommodation

## Special Needs Occupancy Report

| Banner Number | Student Name | Place Number | Room Type | Address | Special Needs | Adviser Name | Current Status |
|---|---|---|---|---|---|---|---|
| 3001 | Tom Daniels | 301 | Hall of Residence | 123 Hall St, City | Mobility Impairment | Dr. Smith | Occupied |
| 3003 | Jane Smith | 302 | Student Apartment | 456 Apartment Rd, City | Visual Impairment | Dr. Johnson | Waiting |
| 3004 | Sarah Clark | 303 | Student Apartment | 789 Apartment Blvd, City | Hearing Impairment | Dr. Lee | Occupied |

### The input and output screens for routine transactions

a. View Student Occupancy Details

Input Screen:

- Student Banner Number: [Text Field] - Enter the banner number or student ID to search for occupancy details.
- View Student Occupancy Button: [Button] - Initiates the search for the student's occupancy details.

Output Screen:

- Students Banner Number: - Displays the student's banner number.
- Student Name: - Displays the student's name.
- Current Status: - Displays the current status (e.g., Placed/Waiting).
- Major: - Displays the student's major.
- Minor: - Displays the student's minor.
- Room Number:- Displays the room number the student is occupying.
- Place Number: - Displays the place number of the room.
- Room Type: - Displays the type of room (e.g., Hall of Residence, Student Apartment).
- Lease Number: - Displays the lease number associated with the occupancy.
- Address: - Displays the address of the hall or apartment.

- Start Date: - Displays the start date of the lease.
- End Date: - Displays the end date of the lease (if known)

b. View Room Occupancy Summary

Input Screen:

- Place Number:  - Enter the Place number of the room to view occupancy details.
- View Room Occupancy SummaryButton: [Button] - Initiates the search for occupancy details of the specified room.

Output Screen:

- Room Number: - Displays the room number.
- Place Number: - Displays the place number of the room.
- Room Type: - Displays the type of room (e.g., Hall of Residence, Student Apartment).
- Current Occupant: - Displays the name of the current occupant, if any.
- Student Banner Number - Displays the banner number of the current occupant.
- Lease Number: - Displays the lease number for the current occupancy.
- Lease Start Date:  - Displays the start date of the current lease.
- Lease End Date: - Displays the end date of the current lease (if known).
- Status: - Displays the status of the room (e.g., Occupied, Vacant).

c. Generate Invoice Report

Input Screen:

- Semester: [Dropdown Menu] - Select the semester (e.g., Fall, Spring, Summer).
- Year: - Enter the academic year.
- Generate Report Button: [Button] - Initiates the generation of the invoice report.

Output Screen:

- Invoice Number: - Displays the invoice number.
- Lease Number:- Displays the lease number.
- Semester:- Displays the semester.
- Student Name:- Displays the student's name.
- Student ID: - Displays the student’s banner number.
- Place Number:- Displays the place number.
- Room Number: - Displays the room number.
- Address:  - Displays the address of the hall or apartment.

- Payment Due: - Displays the amount due.
- Payment Date: - Displays the payment date (if paid).
- Method of Payment: - Displays the payment method (e.g., cash, check).

## List of assumption

- Adviser: Every student is assigned to a member of staff who is his/her adviser.
- Banner Number: Banner number is unique to each student and can be used as primary key for student form.
- HallOfResidence: HallOfResidence provides single rooms.
- Student Apartment: Student Apartment has many single rooms.
- Room: Each room has a unique place number, room number, and monthly rent rate. A room is either in HallOfResidence or it is in StudentFlats.
- Leases can be identified by lease number which is unique. Duration of lease is in semesters and maximum is 1 year and minimum is one semester.
- Each Lease is associated with one Room.
- An Invoice is generated for each Lease. Invoice number uniquely identifies an invoice.
- An Inspection is conducted for a StudentApartment by a ResidenceStaff member.
- ResidenceStaff member can be uniquely identified by StaffNumber.
- Courses are also offered.
- A Student is associated with one NextOfKin.

# Data Dictionary

Write out a user-oriented data dictionary, consisting of an alphabetical list of every data item referenced in any report or routine transaction, and an informal definition for each term.

| | | |
|---|---|---|
| BannerNumber | Student's unique identification number. | Integer |
| Name | First and last name of a person. | String |
| HomeAddress | Location of a person and it includes street, city and postal code. | String |
| MobilePhoneNumber | Phone number of a person that can be used to contact them. | String |
| Email | Email address of a person where university can send out important mails. | String |
| DateOfBirth | Date when a person was born on. | Date |
| Gender | Gender of a person. | String |
| Category | Student's year and level of study (e.g., First-year Undergraduate, Postgraduate) | String |
| Nationality | Student's place of birth | String |
| SpecialNeeds | Any needs/accommodations the student has and/or requires | String |
| Comments | Additional comments added. | String |
| CurrentStatus | Student's current status of admission (Placed/Waiting) | String |
| Major | Student's primary field of study | String |
| Minor | Student's specialized sub-focus within their major | String |
| Room | Student's type of living space ("Hall of Residence" or "Student Apartment") | String |
| AdviserID | Adviser's unique identification number | Integer |
| NextOfKinID | Next of Kin's unique identification number | Integer |
| Position | Position of a person in the university. | String |
| DepartmentName | Name of a department (Eg: Computer science, Biology) | String |
| InternalPhoneNumber | Adviser's phone number. | String |
| HallName | Unique name for the hall of residence managed by the Residence Office. | String |
| StaffNumber | A staff member's unique identification number | Integer |
| PlaceNumber | The room's unique identification number | Integer |
| RoomNumber | The room's identification number | String |
| MonthlyRent | The monthly payment to stay in the room | Decimal |
| Location | Residence Office or Hall | String |
| Type | Which building the room belongs to ("Hall of Residence" or "Student | String |

| | | |
|---|---|---|
| | Apartment”) | |
| ApartmentNumber | The unique number of the apartment managed by the Residence Office | Integer |
| NumberOfRooms | Number of single rooms available in an apartment. | Integer |
| LeaseNumber | Lease’s unique identification number. | Integer |
| Duration | Time period when lease is valid. | String |
| Address | The living space’s location including street, city, & postal code. | String |
| StartDate | Lease start date. | Date |
| EndDate | Lease end date. | Date |
| InvoiceNumber | Invoice’s unique identification number | Integer |
| Semester | Which term the invoice is for | String |
| PaymentDue | Amount of money due | Decimal |
| PaymentDate | Date the payment was made | Date |
| PaymentMethod | Medium of payment | String |
| FirstReminderDate | Date for first reminder of payment | Date |
| SecondReminderDate | Date for second reminder of payment | Date |
| InspectionDate | Date inspection took place | Date |
| Satisfactory | Passed or failed inspection | String |
| CourseNumber | Course’s unique identification number. | Integer |
| CourseTitle | Course title including year, name, and level (upper/lower division). | String |
| InstructorName | Course teacher’s name. | String |
| InstructorPhoneNumber | Teacher’s phone number. | String |
| InstructorRoomNumber | Teacher’s office number. | Integer |
| Relationship | Next of kin’s relationship to student | String |

# Cross Reference Table

| Term | Student | Adviser | HallOfResidence | Room | StudentApartment | Lease | Invoice | Inspection | ResidenceStaff | Course | NextOfKin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BannerNumber | ✔ | ✖ | ✖ | ✖ | ✖ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ |
| Name | ✔ | ✔ | ✖ | ✖ | ✖ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Address | ✔ | ✖ | ✔ | ✖ | ✔ | ✔ | ✔ | ✖ | ✔ | ✖ | ✔ |
| PhoneNumber | ✔ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |
| Email | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✔ | ✖ |
| DateOfBirth | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ |
| Gender | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ |
| Category | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Nationality | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| SpecialNeeds | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Comments | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ |
| CurrentStatus | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Major | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Minor | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Room | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| AdviserID | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| NextOfKinID | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Position | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ |
| DepartmentName | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RoomNumber | ✖ | ✔ | ✖ | ✔ | ✖ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ |
| HallName | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| StaffNumber | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✔ | ✔ | ✖ | ✖ |
| PlaceNumber | ✖ | ✖ | ✖ | ✔ | ✖ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ |
| MonthlyRent | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| Type | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| ApartmentNumber | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ |
| NumberOfRooms | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| LeaseNumber | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ |
| Duration | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ |
| StartDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ |
| EndDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ |
| InvoiceNumber | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| Semester | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| PaymentDue | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| MethodOfPayment | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| PaymentDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| FirstReminderDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| SecondReminderDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ | ✖ |
| InspectionDate | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ |
| Satisfactory | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ | ✖ |
| Location | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ | ✖ |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CourseNumber | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |
| CourseTitle | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |
| InstructorName | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |
| InstructorRoomNumber | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ | ✖ |
| NextOfKinID | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ |
| Relationship | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✔ |

# List of all entities and their associated attributes

**Entities:**

Student, NextOfKin, Adviser, Room, StudentApartment, Inspection, ResidenceStaff, HallOfResidence, Invoice, & Course

**Attributes for Student:**

BannerNumber, FirstName, LastName, HomeAddress, MobilePhoneNumber, Email, DateOfBirth, Gender, Category, Nationality, SpecialNeeds, Comments, CurrentStatus, Major, Minor, & Room

**Attributes for NextOfKin:**

NextOfKinID, FirstName, LastName, Address, ContactPhoneNumber, & Relationship

**Attributes for Adviser:**

AdviserID, FirstName, LastName, Email, DepartmentName, Position, RoomNumber, & InternalPhoneNumber

**Attributes for Room:**

RoomNumber, PlaceNumber, MonthlyRent, & Type

**Attributes for HallOfResidence:**

HallName, MobilePhoneNumber, & Address

**Attributes for StudentApartment:**

ApartmentNumber, Address, & NumberOfRooms

**Attributes for Inspection:**

InspectionDate, Satisfactory,  FirstName, LastName, Comments, & ApartmentNumber

**Attributes for ResidenceStaff:**

StaffNumber, FirstName, LastName, Gender, Address, Location, Position, DateOfBirth, & Email

**Attributes for Invoice:**

InvoiceNumber, Semester PaymentDue, PlaceNumber, MethodOfPayment, FirstReminderDate, SecondReminderDate, Address,  PaymentDue, Name, & RoomNumber

**Attributes for Course:**

InstructorName, DepartmentName, CourseNumber, CourseTitle, InstructorPhoneNumber, InstructorEmail, & InstructorRoomNumber

# List of relationships to be represented, and any descriptive attributes for them.

**Student to Next of Kin**
Relationship Type: One-to-Zero or One
Description: Each student may have at most one next-of-kin registered for emergency contact purposes.
Descriptive Attributes: None directly; typically connected through a foreign key NextOfKinID in the Student entity, which may be nullable.

**Student to Enroll in Course**
Relationship Type: Many-to-Many
Description: Each student can enroll in multiple courses during their academic tenure. But a course can be taken by multiple students.
Descriptive Attributes: Managed through an enrollment join table that includes StudentID and CourseID.

**Student to Invoice**
Relationship Type: One-to-One
Description: Each student receives a unique invoice, which consolidates all applicable fees for a term or year.
Descriptive Attributes: Includes InvoiceNumber, and may include details like TotalAmount, DueDate, etc.

**Student to Room**
Relationship Type: One-to-One
Description: Each student is assigned to exactly one room, assuming individual room assignments.
Descriptive Attributes: RoomNumber or RoomID linking to the specific room details.

**Student to Lease**
Relationship Type: One-to-One
Description: Each student signs a lease agreement specific to their room assignment.
Descriptive Attributes: Includes LeaseID, StartDate, EndDate, and terms of the lease.

**Student to Adviser**
Relationship Type: Many-to-One
Description: Several students are assigned to one adviser, who supports their academic and personal needs.

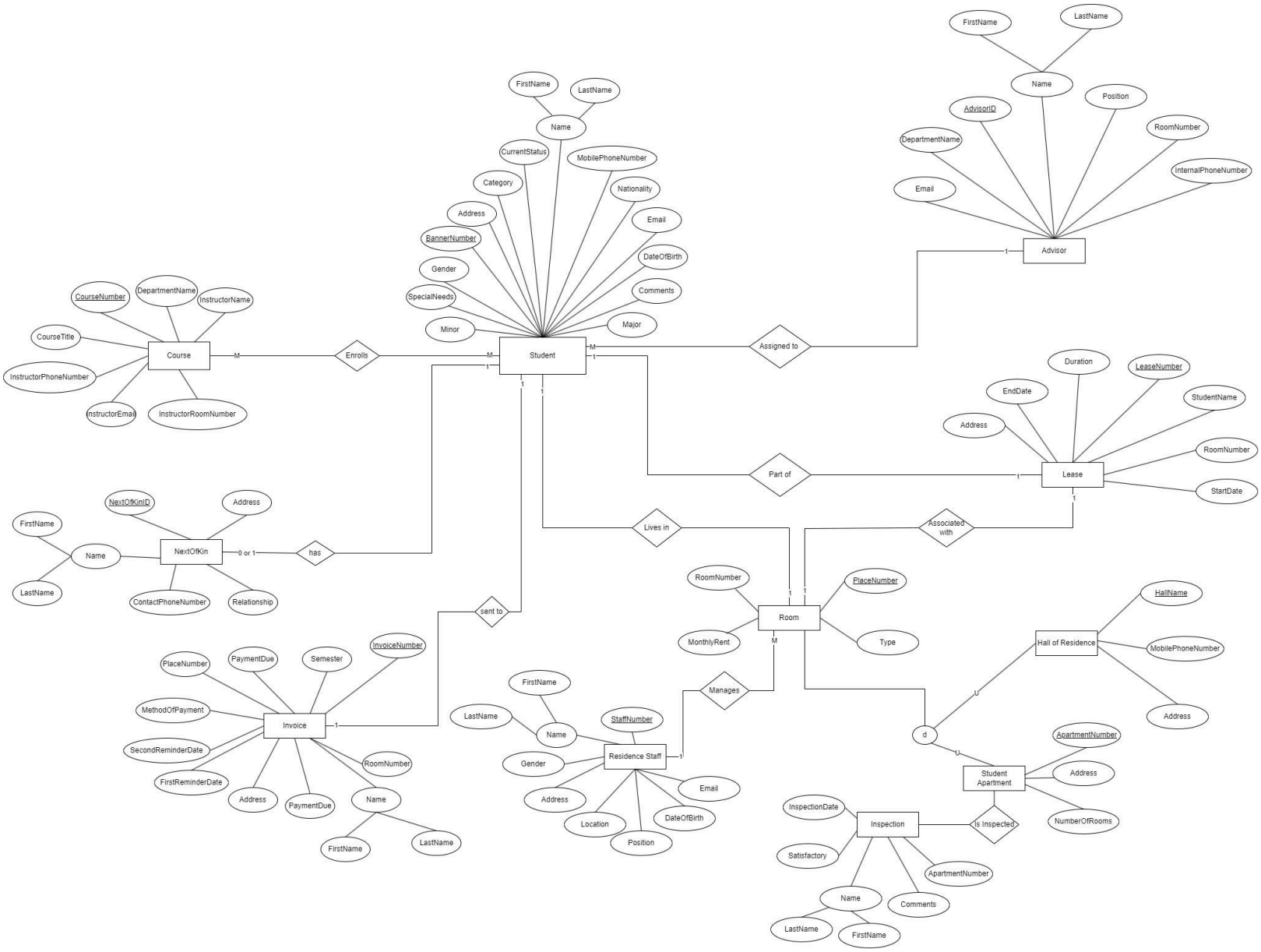
Descriptive Attributes: Adviser is linked through AdviserID in the Student entity.

**Room to Lease**
Relationship Type: One-to-One
Description: Each room is linked to a specific lease detailing the terms of occupation.
Descriptive Attributes: LeaseID directly associated with the room.

**Room to Residence Staff**
Relationship Type: Many-to-One
Description: Several rooms can be under the management or maintenance of a single residence staff member.
Descriptive Attributes: Staff is linked through StaffID, indicating which staff member manages the rooms.

**Room Associated with Lease**
Relationship Type: One-to-One
Description: Direct association where each room is explicitly linked to a lease defining its use.
Descriptive Attributes: LeaseID uniquely identifies the lease associated with each room.

# E-R diagram:


FirstName
LastName
Name
Position
AdvisorID
RoomNumber
DepartmentName
InternalPhoneNumber
Email
Advisor
FirstName
LastName
Name
CurrentStatus
MobilePhoneNumber
Category
Nationality
Address
Email
BannerNumber
DateOfBirth
Gender
SpecialNeeds
Comments
Minor
Major
Student
CourseNumber
DepartmentName
InstructorName
CourseTitle
Course
InstructorPhoneNumber
InstructorEmail
InstructorRoomNumber
Enrolls
Assigned to
Duration
LeaseNumber
EndDate
StudentName
Address
RoomNumber
Part of
Lease
StartDate
NextOfKinID
Address
FirstName
Name
NextOfKin
0 or 1
has
LastName
ContactPhoneNumber
Relationship
Lives in
Associated with
RoomNumber
PlaceNumber
Room
MonthlyRent
Type
sent to
HallName
Hall of Residence
MobilePhoneNumber
Address
InvoiceNumber
PlaceNumber
PaymentDue
Semester
MethodOfPayment
Invoice
SecondReminderDate
FirstReminderDate
RoomNumber
Address
PaymentDue
Name
FirstName
LastName
Manages
FirstName
LastName
Name
StaffNumber
Residence Staff
Gender
Address
Email
Location
DateOfBirth
Position
d
ApartmentNumber
Address
Student Apartment
NumberOfRooms
InspectionDate
Inspection
Is Inspected
Satisfactory
ApartmentNumber
Name
Comments
LastName
FirstName

# Updated data dictionary and list of assumptions

| | | |
|---|---|---|
| BannerNumber | Student's unique identification number. | Integer |
| LastName | Last name of a person. | String |
| FirstName | First name of a person. | String |
| | | |
| HomeAddress | Location of a person and it includes street, city and postal code. | String |
| MobilePhoneNumber | Phone number of a person that can be used to contact them. | String |
| Email | Email address of a person where university can send out important mails. | String |
| DateOfBirth | Date when a person was born. | Date |
| Gender | Gender of a person. | String |
| Category | Student's year and level of study (e.g., First-year Undergraduate, Postgraduate) | String |
| Nationality | Student's place of birth | String |
| SpecialNeeds | Any needs/accommodations the student has and/or requires | String |
| Comments | Additional comments added. | String |
| CurrentStatus | Student's current status of admission (Placed/Waiting) | String |
| Major | Student's primary field of study | String |
| Minor | Student's specialized sub-focus within their major | String |
| Room | Student's type of living space ("Hall of Residence" or "Student Apartment") | String |
| AdviserID | Adviser's unique identification number | Integer |
| NextOfKinID | Next of Kin's unique identification number | Integer |
| Position | Position of a person in the university. | String |
| DepartmentName | Name of a department (Eg: Computer science, Biology) | String |
| InternalPhoneNumber | Adviser's phone number. | String |
| RoomNumber | Adviser's room number. | Integer |
| HallName | Unique name for the hall of residence managed by the Residence Office. | String |
| StaffNumber | A staff member's unique identification number | Integer |
| PlaceNumber | The room's unique identification number | Integer |
| RoomNumber | The room's identification number | String |
| MonthlyRent | The monthly payment to stay in the room | Decimal |

| | | |
|---|---|---|
| Type | Which building the room belongs to (“Hall of Residence” or “Student Apartment”) | String |
| ApartmentNumber | The unique number of the apartment managed by the Residence Office | Integer |
| NumberOfRooms | Number of single rooms available in an apartment. | Integer |
| LeaseNumber | Lease’s unique identification number. | Integer |
| Duration | Time period when lease is valid. | String |
| Address | The living space’s location including street, city, & postal code. | String |
| StartDate | Lease start date. | Date |
| EndDate | Lease end date. | Date |
| InvoiceNumber | Invoice’s unique identification number | Integer |
| Semester | Which term the invoice is for | String |
| PaymentDue | Amount of money due | Decimal |
| PaymentDate | Date the payment was made | Date |
| PaymentMethod | Medium of payment | String |
| FirstReminderDate | Date for first reminder of payment | Date |
| SecondReminderDate | Date for second reminder of payment | Date |
| InspectionDate | Date inspection took place | Date |
| Satisfactory | Passed or failed inspection | String |
| CourseNumber | Course’s unique identification number. | Integer |
| CourseTitle | Course title including year, name, and level (upper/lower division). | String |
| InstructorName | Course teacher’s name. | String |
| InstructorPhoneNumber | Teacher’s phone number. | String |
| InstructorRoomNumber | Teacher’s office number. | Integer |
| Relationship | Next of kin’s relationship to student | String |

Updates:

- Address: Into streetName, CityName, StateName, Zip, CountryName.
- Split the Name into First and Last
- Lease: StudentName: Split
- Add foreign key in lease for placeNumber

# Relational Schema


Schema 'On Campus Accomodatiopn Office

# Creating Database

Creating tables and inserting values:

Table Advisor

```
1 •  SELECT * FROM housing_database.advisor;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| FirstName | LastName | AdvisorID | Position | DepartmentName | InternalPhoneNumber | Email | RoomNumber |
|---|---|---|---|---|---|---|---|
| John | Doe | 1 | Professor | Computer Science | 1234 | johndoe@university.edu | 101 |
| Jane | Smith | 2 | Associate Professor | Mathematics | 1235 | janesmith@university.edu | 102 |
| Alice | Johnson | 3 | Lecturer | Physics | 1236 | alicejohnson@university.edu | 103 |
| Bob | Brown | 4 | Senior Lecturer | Chemistry | 1237 | bobbrown@university.edu | 104 |
| Carol | White | 5 | Assistant Professor | Biology | 1238 | carolwhite@university.edu | 105 |
| David | Black | 6 | Professor | Engineering | 1239 | davidblack@university.edu | 106 |
| Eve | Green | 7 | Senior Lecturer | Economics | 1240 | evegreen@university.edu | 107 |
| Frank | Blue | 8 | Lecturer | History | 1241 | frankblue@university.edu | 108 |
| Grace | Yellow | 9 | Associate Professor | Literature | 1242 | graceyellow@university.edu | 109 |
| Hank | Purple | 10 | Professor | Philosophy | 1243 | hankpurple@university.edu | 110 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Course

SQL File 9* | advisor | course | Limit to 1000 rows

```
1 •  SELECT * FROM housing_database.course;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| CourseNumber | CourseTitle | InstructorFirstName | InstructorLastName | InstructorPhoneNumber | InstructorEmail | InstructorRoomNumber | DepartmentName |
|---|---|---|---|---|---|---|---|
| 1 | Database Systems | Alice | Johnson | (710) 234-6789 | alicejohnson@university.edu | 102 | Computer Science |
| 2 | Calculus | David | Smith | (669) 345-6790 | davidsmith@university.edu | 103 | Mathematics |
| 3 | Physics 101 | Robert | Brown | (345) 456-6791 | robertbrown@university.edu | 104 | Physics |
| 4 | Chemistry 101 | Michael | White | (134) 567-6792 | michaelwhite@university.edu | 105 | Chemistry |
| 5 | Biology 101 | Laura | Black | (298) 678-6793 | laurablack@university.edu | 106 | Biology |
| 6 | Engineering 101 | Chris | Green | (471) 789-6794 | chrisgreen@university.edu | 107 | Engineering |
| 7 | Economics 101 | Sarah | Blue | (562) 890-6795 | sarahblue@university.edu | 108 | Economics |
| 8 | History 101 | James | Yellow | (680) 901-6796 | jamesyellow@university.edu | 109 | History |
| 9 | Literature 101 | Emily | Purple | (710) 012-6797 | emilypurple@university.edu | 110 | Literature |
| 10 | Philosophy 101 | Anna | Orange | (499) 123-6798 | annaorange@university.edu | 111 | Philosophy |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

HallOfResidence

```
SELECT * FROM housing_database.hallofresidence;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| HallId | StreetName | CityName | StateName | Zip | CountryCode | MobilePhoneNumber |
|---|---|---|---|---|---|---|
| 1 | 123 College Ave | Berkeley | CA | 94704 | US | (510) 123-4567 |
| 2 | 456 University Blvd | Berkeley | CA | 94705 | US | (510) 234-5678 |
| 3 | 789 Dormitory Ln | Berkeley | CA | 94720 | US | (510) 345-6789 |
| 4 | 101 Campus Dr | Berkeley | CA | 94703 | US | (510) 456-7890 |
| 5 | 202 Student Way | Berkeley | CA | 94709 | US | (510) 567-8901 |
| 6 | 303 Quad Cir | Berkeley | CA | 94707 | US | (510) 678-9012 |
| 7 | 404 Academic Rd | Berkeley | CA | 94702 | US | (510) 789-0123 |
| 8 | 505 Learning St | Berkeley | CA | 94710 | US | (510) 890-1234 |
| 9 | 606 Study Ln | Berkeley | CA | 94708 | US | (510) 901-2345 |
| 10 | 707 Knowledge Ave | Berkeley | CA | 94706 | US | (510) 012-3456 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Inspection

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| StaffNumber | StudentFirstName | StudentLastName | ApartmentNumber | InspectionDate | Satisfactory | Comments |
|---|---|---|---|---|---|---|
| 1 | NULL | NULL | 1 | NULL | NULL | NULL |
| 2 | NULL | NULL | 2 | NULL | NULL | NULL |
| 3 | James | Smith | 3 | 2024-08-17 | No | Plumbing issues detected. |
| 4 | NULL | NULL | 4 | NULL | NULL | NULL |
| 5 | NULL | NULL | 5 | NULL | NULL | NULL |
| 6 | Sarah | Black | 6 | 2024-08-20 | No | Electrical issues detected. |
| 7 | NULL | NULL | 7 | NULL | NULL | NULL |
| 8 | NULL | NULL | 8 | NULL | NULL | NULL |
| 9 | Michael | Yellow | 9 | 2024-08-23 | Yes | Well maintained. |
| 10 | NULL | NULL | 10 | NULL | NULL | NULL |

Invoice

```
SELECT * FROM housing_database.invoice;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| Inv | Semester | FirstName | LastName | BannerNu | PlaceNu | RoomNumber | StreetName | CityName | StateName | Zip | CountryCode | PaymentDue | PaymentMethod | PaymentDate | FirstReminderDate | SecondReminderD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Fall 2024 | Mark | Taylor | 1 | 1 | 201A | 123 College Rd | University City | CA | 94540 | US | 2000.00 | Credit Card | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 2 | Fall 2024 | Lisa | Brown | 2 | 2 | 202B | 125 College Rd | University City | CA | 94541 | US | 1900.00 | Bank Transfer | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 3 | Fall 2024 | James | Smith | 3 | 3 | 203C | 127 College Rd | University City | CA | 94542 | US | 2100.00 | Credit Card | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 4 | Fall 2024 | Emily | Johnson | 4 | 4 | 204D | 129 College Rd | University City | CA | 94543 | US | 1800.00 | Check | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 5 | Fall 2024 | David | White | 5 | 5 | 205E | 131 College Rd | University City | CA | 94544 | US | 1950.00 | Credit Card | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 6 | Fall 2024 | Sarah | Black | 6 | 6 | 206F | 133 College Rd | University City | CA | 94545 | US | 2150.00 | Bank Transfer | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 7 | Fall 2024 | Chris | Green | 7 | 7 | 207G | 135 College Rd | University City | CA | 94546 | US | 2200.00 | Credit Card | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 8 | Fall 2024 | Laura | Blue | 8 | 8 | 208H | 137 College Rd | University City | CA | 94547 | US | 2000.00 | Check | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 9 | Fall 2024 | Michael | Yellow | 9 | 9 | 209I | 139 College Rd | University City | CA | 94548 | US | 2050.00 | Credit Card | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| 10 | Fall 2024 | Anna | Purple | 10 | 10 | 210J | 141 College Rd | University City | CA | 94549 | US | 1950.00 | Bank Transfer | 2024-09-01 | 2024-09-10 | 2024-09-20 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

## Lease

```
1 •  SELECT * FROM housing_database.lease;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| LeaseNumber | Duration | BannerNumber | StudentFirstName | StudentLastName | PlaceNumber | RoomNumber | StreetName | CityName | StateName | Zip | CountryCode | StartDate | EndDate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12 months | 1 | Mark | Taylor | 1 | 201A | 123 College Rd | University City | CA | 94540 | US | 2024-08-01 | 2025-07-31 |
| 2 | 12 months | 2 | Lisa | Brown | 2 | 202B | 125 College Rd | University City | CA | 94541 | US | 2024-08-01 | 2025-07-31 |
| 3 | 12 months | 3 | James | Smith | 3 | 203C | 127 College Rd | University City | CA | 94542 | US | 2024-08-01 | 2025-07-31 |
| 4 | 12 months | 4 | Emily | Johnson | 4 | 204D | 129 College Rd | University City | CA | 94543 | US | 2024-08-01 | 2025-07-31 |
| 5 | 12 months | 5 | David | White | 5 | 205E | 131 College Rd | University City | CA | 94544 | US | 2024-08-01 | 2025-07-31 |
| 6 | 12 months | 6 | Sarah | Black | 6 | 206F | 133 College Rd | University City | CA | 94545 | US | 2024-08-01 | 2025-07-31 |
| 7 | 12 months | 7 | Chris | Green | 7 | 207G | 135 College Rd | University City | CA | 94546 | US | 2024-08-01 | 2025-07-31 |
| 8 | 12 months | 8 | Laura | Blue | 8 | 208H | 137 College Rd | University City | CA | 94547 | US | 2024-08-01 | 2025-07-31 |
| 9 | 12 months | 9 | Michael | Yellow | 9 | 209I | 139 College Rd | University City | CA | 94548 | US | 2024-08-01 | 2025-07-31 |
| 10 | 12 months | 10 | Anna | Purple | 10 | 210J | 141 College Rd | University City | CA | 94549 | US | 2024-08-01 | 2025-07-31 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

## NextOfKin

```
1 •  SELECT * FROM housing_database.nextofkin;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| NextOfKinID | FirstName | LastName | StreetName | CityName | StateName | Zip | CountryCode | Relationship | ContactPhoneNumber |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Jane | Doe | 123 Main St | Berkeley | CA | 94704 | US | Mother | (510) 123-4567 |
| 2 | Peter | Smith | 456 Oak St | Tracy | CA | 95376 | US | Father | (209) 234-5678 |
| 3 | Mary | Johnson | 789 Pine St | Castro Valley | CA | 94546 | US | Guardian | (510) 345-6789 |
| 4 | Paul | Brown | 101 Maple St | Richmond | CA | 94801 | US | Sibling | (510) 456-7890 |
| 5 | Lucy | White | 202 Birch St | Oakland | CA | 94612 | US | Mother | (510) 567-8901 |
| 6 | James | Black | 303 Cedar St | San Leandro | CA | 94577 | US | Father | (510) 678-9012 |
| 7 | Emily | Green | 404 Spruce St | Hayward | CA | 94541 | US | Guardian | (510) 789-0123 |
| 8 | Michael | Blue | 505 Elm St | Dublin | CA | 94568 | US | Sibling | (925) 890-1234 |
| 9 | Sarah | Yellow | 606 Fir St | Fremont | CA | 94536 | US | Mother | (510) 901-2345 |
| 10 | Tom | Purple | 707 Redwood St | Pleasanton | CA | 94566 | US | Father | (925) 012-3456 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

## ResidenceStaff

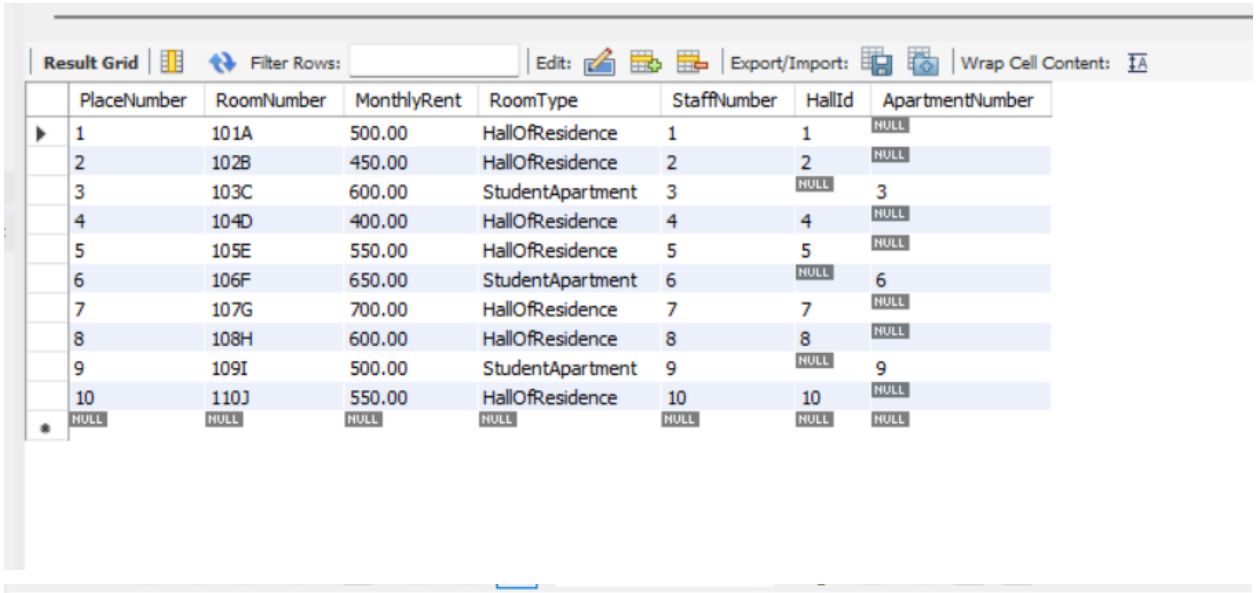
```
1 •  SELECT * FROM housing_database.residencestaff;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

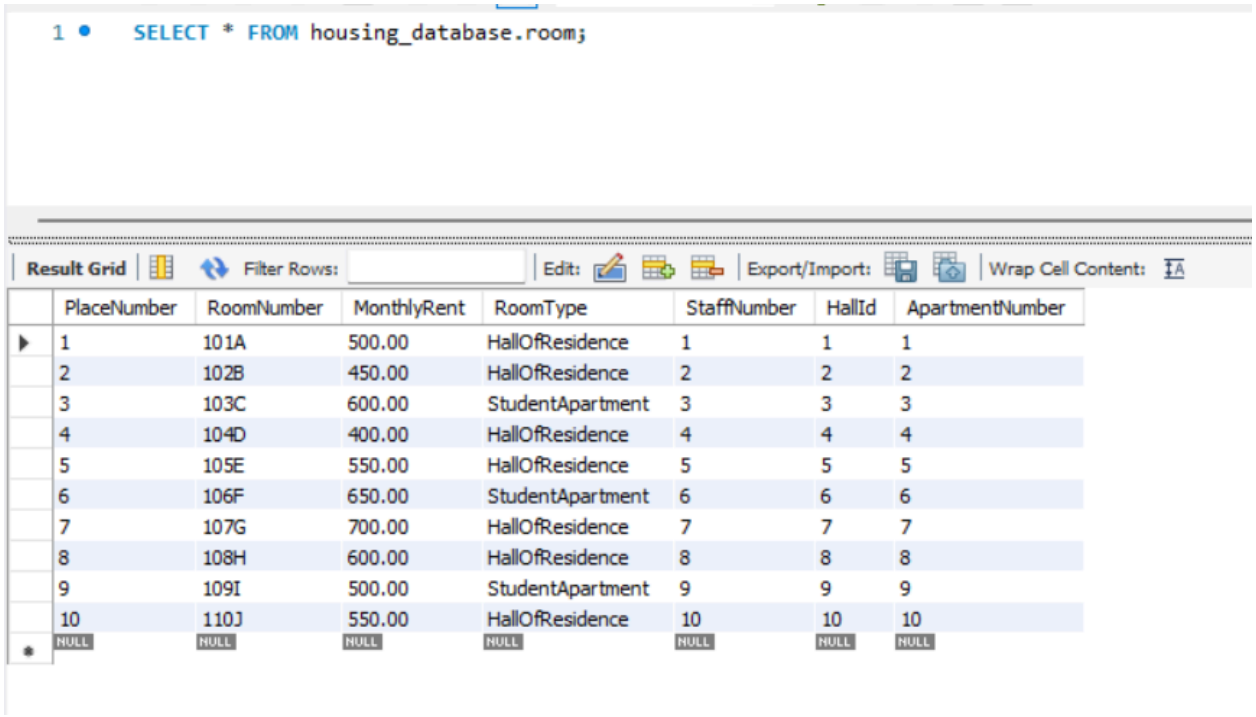
| StaffNumber | FirstName | LastName | Email | StreetName | CityName | StateName | Zip | CountryCode | DateOfBirth | Gender | Position | Location |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Mike | Smith | mikesmith@university.edu | 101 Campus Block A | Berkeley | CA | 94720 | US | 1970-01-01 | Male | Manager | Campus Block A |
| 2 | Laura | Brown | laurabrown@university.edu | 102 Campus Block A | Berkeley | CA | 94720 | US | 1980-02-02 | Female | Coordinator | Campus Block A |
| 3 | Chris | White | chriswhite@university.edu | 103 Campus Block B | Berkeley | CA | 94720 | US | 1975-03-03 | Male | Supervisor | Campus Block B |
| 4 | Anna | Black | annablack@university.edu | 104 Campus Block B | Berkeley | CA | 94720 | US | 1985-04-04 | Female | Director | Campus Block B |
| 5 | James | Green | jamesgreen@university.edu | 105 Campus Block C | Berkeley | CA | 94720 | US | 1990-05-05 | Male | Assistant | Campus Block C |
| 6 | Sarah | Blue | sarahblue@university.edu | 106 Campus Block C | Berkeley | CA | 94720 | US | 1978-06-06 | Female | Manager | Campus Block C |
| 7 | Robert | Yellow | robertyellow@university.edu | 107 Campus Block D | Berkeley | CA | 94720 | US | 1982-07-07 | Male | Coordinator | Campus Block D |
| 8 | Emily | Purple | emilypurple@university.edu | 108 Campus Block D | Berkeley | CA | 94720 | US | 1989-08-08 | Female | Supervisor | Campus Block D |
| 9 | Paul | Orange | paulorange@university.edu | 109 Campus Block E | Berkeley | CA | 94720 | US | 1972-09-09 | Male | Director | Campus Block E |
| 10 | Diana | Pink | dianapink@university.edu | 110 Campus Block E | Berkeley | CA | 94720 | US | 1987-10-10 | Female | Assistant | Campus Block E |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Room

Updated Room Table:



| PlaceNumber | RoomNumber | MonthlyRent | RoomType | StaffNumber | HallId | ApartmentNumber |
|---|---|---|---|---|---|---|
| 1 | 101A | 500.00 | HallOfResidence | 1 | 1 | NULL |
| 2 | 102B | 450.00 | HallOfResidence | 2 | 2 | NULL |
| 3 | 103C | 600.00 | StudentApartment | 3 | NULL | 3 |
| 4 | 104D | 400.00 | HallOfResidence | 4 | 4 | NULL |
| 5 | 105E | 550.00 | HallOfResidence | 5 | 5 | NULL |
| 6 | 106F | 650.00 | StudentApartment | 6 | NULL | 6 |
| 7 | 107G | 700.00 | HallOfResidence | 7 | 7 | NULL |
| 8 | 108H | 600.00 | HallOfResidence | 8 | 8 | NULL |
| 9 | 109I | 500.00 | StudentApartment | 9 | NULL | 9 |
| 10 | 110J | 550.00 | HallOfResidence | 10 | 10 | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |



```
1 •  SELECT * FROM housing_database.room;
```

| PlaceNumber | RoomNumber | MonthlyRent | RoomType | StaffNumber | HallId | ApartmentNumber |
|---|---|---|---|---|---|---|
| 1 | 101A | 500.00 | HallOfResidence | 1 | 1 | 1 |
| 2 | 102B | 450.00 | HallOfResidence | 2 | 2 | 2 |
| 3 | 103C | 600.00 | StudentApartment | 3 | 3 | 3 |
| 4 | 104D | 400.00 | HallOfResidence | 4 | 4 | 4 |
| 5 | 105E | 550.00 | HallOfResidence | 5 | 5 | 5 |
| 6 | 106F | 650.00 | StudentApartment | 6 | 6 | 6 |
| 7 | 107G | 700.00 | HallOfResidence | 7 | 7 | 7 |
| 8 | 108H | 600.00 | HallOfResidence | 8 | 8 | 8 |
| 9 | 109I | 500.00 | StudentApartment | 9 | 9 | 9 |
| 10 | 110J | 550.00 | HallOfResidence | 10 | 10 | 10 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Student

```
1 •  SELECT * FROM housing_database.student;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| Bann | FirstName | LastName | StreetName | CityN | State | Zip | Country | MobilePhoneNumber | Email | DateOfBirt | Genc | Categor | Nat | Special | Comments | Curren | Major | Minor | Advis | Nex | Placef | Course | LeaseNumber |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Mark | Taylor | 101 Univers... | Be... | CA | 94704 | US | 510-555-1111 | marktaylor@university.edu | 2000-0... | Male | Under... | US | None | Active and thriving | Active | Computer Science | Mathematics | 1 | 1 | 1 | 1 | NULL |
| 2 | Lisa | Brown | 102 Univers... | Be... | CA | 94704 | US | 510-555-2222 | lisabrown@university.edu | 2000-0... | F... | Under... | US | Low... | Needs assistance ... | Active | Mathematics | Physics | 2 | 2 | 2 | 2 | NULL |
| 3 | James | Smith | 103 Univers... | Be... | CA | 94704 | US | 510-555-3333 | jamessmith@university.edu | 2000-0... | Male | Under... | US | None | No comments | Active | Physics | Chemistry | 3 | 3 | 3 | 3 | NULL |
| 4 | Emily | Johnson | 104 Univers... | Be... | CA | 94704 | US | 510-555-4444 | emilyjohnson@university.edu | 2000-0... | F... | Under... | US | Limb... | Requires accessib... | Active | Chemistry | Biology | 4 | 4 | 4 | 4 | NULL |
| 5 | David | White | 105 Univers... | Be... | CA | 94704 | US | 510-555-5555 | davidwhite@university.edu | 2000-0... | Male | Under... | US | None | Active participant... | Active | Biology | Engineering | 5 | 5 | 5 | 5 | NULL |
| 6 | Sarah | Black | 106 Univers... | Be... | CA | 94704 | US | 510-555-6666 | sarahblack@university.edu | 2000-0... | F... | Under... | US | Hea... | Uses hearing aids... | Active | Engineering | Economics | 6 | 6 | 6 | 6 | NULL |
| 7 | Chris | Green | 107 Univers... | Be... | CA | 94704 | US | 510-555-7777 | chrisgreen@university.edu | 2000-0... | Male | Under... | US | None | Participates in sp... | Active | Economics | History | 7 | 7 | 7 | 7 | NULL |
| 8 | Laura | Blue | 108 Univers... | Be... | CA | 94704 | US | 510-555-8888 | laurablue@university.edu | 2000-0... | F... | Under... | US | None | Active in student ... | Active | History | Literature | 8 | 8 | 8 | 8 | NULL |
| 9 | Michael | Yellow | 109 Univers... | Be... | CA | 94704 | US | 510-555-9999 | michaelyellow@university.edu | 2000-0... | Male | Under... | US | None | Volunteer at local ... | Active | Literature | Philosophy | 9 | 9 | 9 | 9 | NULL |
| 10 | Anna | Purple | 110 Univers... | Be... | CA | 94704 | US | 510-555-0000 | annapurple@university.edu | 2000-1... | F... | Under... | US | None | Excels in academics | Active | Philosophy | Computer S... | 10 | 10 | 10 | 10 | NULL |

StudentApartment

```
1 •  SELECT * FROM housing_database.studentapartment;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| ApartmentNumber | StreetName | CityName | StateName | Zip | CountryCode | NumberOfRooms |
|---|---|---|---|---|---|---|
| 1 | 123 Campus Dr | Berkeley | CA | 94720 | US | 4 |
| 2 | 124 Campus Dr | Berkeley | CA | 94720 | US | 3 |
| 3 | 125 Campus Dr | Berkeley | CA | 94720 | US | 2 |
| 4 | 126 Campus Dr | Berkeley | CA | 94720 | US | 1 |
| 5 | 127 Campus Dr | Berkeley | CA | 94720 | US | 5 |
| 6 | 128 Campus Dr | Berkeley | CA | 94720 | US | 4 |
| 7 | 129 Campus Dr | Berkeley | CA | 94720 | US | 3 |
| 8 | 130 Campus Dr | Berkeley | CA | 94720 | US | 2 |
| 9 | 131 Campus Dr | Berkeley | CA | 94720 | US | 1 |
| 10 | 132 Campus Dr | Berkeley | CA | 94720 | US | 5 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

# SQL statements that process 5 non routine requests:

Query1:
-- Retrieves students with special needs, along with their Room and Advisor's information:

Select s.BannerNumber, S.FirstName, S.LastName, s.SpecialNeeds, s.CurrentStatus,
Concat(a.FirstName, ' ' , a.LastName) AS AdvisorName,
r.RoomType, r.PlaceNumber
From Student s JOIN Room r on s.PlaceNumber = r.PlaceNumber
Join Advisor a ON s.AdvisorID = a.AdvisorID
where s.SpecialNeeds != 'None';

```
4   -- Retrives students with specialneeds, along with thier Room and Advisor's informartion
5   Select s.BannerNumber, S.FirstName, S.LastName, s.SpecialNeeds, s.CurrentStatus,
6   Concat(a.FirstName, ' ' , a.LastName) AS AdvisorName,
7   r.RoomType, r.PlaceNumber, r.RoomNumber
8   From Student s JOIN Room r on s.PlaceNumber = r.PlaceNumber
9   Join Advisor a ON s.AdvisorID = a.AdvisorID
10  where s.SpecialNeeds != 'None';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | FirstName | LastName | SpecialNeeds | CurrentStatus | AdvisorName | RoomType | PlaceNumber | RoomNumber |
|---|---|---|---|---|---|---|---|---|
| 2 | Lisa | Brown | Low vision | Active | Jane Smith | HallOfResidence | 2 | 102B |
| 4 | Emily | Johnson | Limb leg | Active | Bob Brown | HallOfResidence | 4 | 104D |
| 6 | Sarah | Black | Hearing impairment | Active | David Black | StudentApartment | 6 | 106F |

Query 2:
-- Retrieves apartments with more than 2 rooms along with students info

SELECT s.BannerNumber, s.FirstName, s.LastName, s.PlaceNumber,
sa.ApartmentNumber, sa.NumberOfRooms

From Student s Join Room r On s.PlaceNumber = r.PlaceNumber
Join StudentApartment sa ON r.ApartmentNumber = sa.ApartmentNumber
where  sa.NumberOfRooms >2;

```
12    -- Retrieves apartmemts with more thyan 2 rooms along with students info
13    SELECT s.BannerNumber, s.FirstName, s.LastName, s.PlaceNumber,
14    sa.ApartmentNumber, sa.NumberOfRooms
15    From Student s Join Room r On s.PlaceNumber = r.PlaceNumber
16    Join StudentApartment sa ON r.ApartmentNumber = sa.ApartmentNumber
17    where  sa.NumberOfRooms >2;
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | FirstName | LastName | PlaceNumber | ApartmentNumber | NumberOfRooms |
|---|---|---|---|---|---|
| 1 | Mark | Taylor | 1 | 1 | 4 |
| 2 | Lisa | Brown | 2 | 2 | 3 |
| 5 | David | White | 5 | 5 | 5 |
| 6 | Sarah | Black | 6 | 6 | 4 |
| 7 | Chris | Green | 7 | 7 | 3 |
| 10 | Anna | Purple | 10 | 10 | 5 |

Query3:
– Retrieve students whose Next of kin are their parents.

Select s.BannerNumber, Concat(s.FirstName, ' ',  s.LastName) AS StudentName,
n.NextofKinID, concat(n.FirstName, ' ', n.LastName) AS NextOfKinName, n.Relationship
from Student s Join NextOfKin n
ON s.NextofKinID = n.NextOfKinID
where n.Relationship = 'Mother' or n.Relationship = 'Father';

```
21    -- Retrieve students whose Next of kin are their parents.
22    Select s.BannerNumber, Concat(s.FirstName, ' ',  s.LastName) AS StudentName,
23    n.NextofKinID, concat(n.FirstName, ' ', n.LastName) AS NextOfKinName, n.Relationship
24    from Student s Join NextOfKin n
25    ON s.NextofKinID = n.NextOfKinID
26    where n.Relationship = 'Mother' or n.Relationship = 'Father';
27
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | StudentName | NextofKinID | NextOfKinName | Relationship |
|---|---|---|---|---|
| 1 | Mark Taylor | 1 | Jane Doe | Mother |
| 2 | Lisa Brown | 2 | Peter Smith | Father |
| 5 | David White | 5 | Lucy White | Mother |
| 6 | Sarah Black | 6 | James Black | Father |
| 9 | Michael Yellow | 9 | Sarah Yellow | Mother |
| 10 | Anna Purple | 10 | Tom Purple | Father |

Query4:
-- Retrieve info about residence staff with rooms they are assign to along with their position and location
Select r.RoomNumber, r.PlaceNumber, r.MonthlyRent,

rs.FirstName, rs.LastName, rs.Position, rs.Location
From Room r Join ResidenceStaff rs
ON r.staffNumber = rs.StaffNumber;

```
29    -- Query4
30    -- Retrieve info about residence staff with rooms they are assign to along with their position and location
31    Select r.RoomNumber, r.PlaceNumber, r.MonthlyRent,
32    rs.FirstName, rs.LastName, rs.Position, rs.Location
33    From Room r Join ResidenceStaff rs
34    ON r.staffNumber = rs.StaffNumber;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| RoomNumber | PlaceNumber | MonthlyRent | FirstName | LastName | Position | Location |
|---|---|---|---|---|---|---|
| 101A | 1 | 500.00 | Mike | Smith | Manager | Campus Block A |
| 102B | 2 | 450.00 | Laura | Brown | Coordinator | Campus Block A |
| 103C | 3 | 600.00 | Chris | White | Supervisor | Campus Block B |
| 104D | 4 | 400.00 | Anna | Black | Director | Campus Block B |
| 105E | 5 | 550.00 | James | Green | Assistant | Campus Block C |
| 106F | 6 | 650.00 | Sarah | Blue | Manager | Campus Block C |
| 107G | 7 | 700.00 | Robert | Yellow | Coordin | Block D |
| 108H | 8 | 600.00 | Emily | Purple | Supervisor | Campus Block D |
| 109I | 9 | 500.00 | Paul | Orange | Director | Campus Block E |
| 110J | 10 | 550.00 | Diana | Pink | Assistant | Campus Block E |

Coordinator

-- Query5
-- Show apartments which issues found during inspection, along with staff info who conducted the inspection

SELECT i.InspectionDate, i.StaffNumber, CONCAT(s.FirstName, ' ', s.LastName) AS StaffName,
CONCAT(i.StudentFirstName, ' ', i.StudentLastName) AS StudentName,
i.ApartmentNumber, i.Satisfactory, i.Comments
FROM Inspection i
JOIN ResidenceStaff s ON i.StaffNumber = s.StaffNumber
WHERE i.Satisfactory = 'NO';

```
36    -- Query5
37    -- Show apartments which issues found during inspection, along with staff info who conducted the inspection
38    SELECT i.InspectionDate, i.StaffNumber, CONCAT(s.FirstName, ' ', s.LastName) AS StaffName,
39    CONCAT(i.StudentFirstName, ' ', i.StudentLastName) AS StudentName,
40     i.ApartmentNumber, i.Satisfactory, i.Comments
41    FROM  Inspection i
42    JOIN ResidenceStaff s ON i.StaffNumber = s.StaffNumber
43    WHERE i.Satisfactory = 'NO';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| InspectionDate | StaffNumber | StaffName | StudentName | ApartmentNumber | Satisfactory | Comments |
|---|---|---|---|---|---|---|
| 2024-08-17 | 3 | Chris White | James Smith | 3 | No | Plumbing issues detected. |
| 2024-08-20 | 6 | Sarah Blue | Sarah Black | 6 | No | Electrical issues detected. |

# Module 5:

**Step 6.1 Begin with the list of the tables that the entities and relationships from the ER diagram mapped to naturally. For each table on the list, identify functional dependencies and normalize the relation to BCNF. Then decide whether the resulting tables should be implemented in that form. If not, explain why.**

## Foreign keys in Italics.

**Student Table Functional Dependencies:**
BannerNumber → {{FirstName, LastName, StreetName, CityName, Zip, CountryCode, MobilePhoneNumber, Email, DateOfBirth, Gender, Category, Nationality, SpecialNeeds, Comments, CurrentStatus, Major, Minor, *AdvisorID, NextOfKinID, PlaceNumber, CourseNumber, LeaseNumber*}, Email -> {FirstName, LastName}, MobilePhoneNumber -> {FirstName, LastName}, SpecialNeeds-> {Comments}}

Functional dependencies = {BN -> {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN }, E -> {F, L}, P -> {F, L}, SN -> {CO}}

This table is in 1NF because every value is atomic.
Let's make candidate key:

1. BN F L St CN SN Z CC MPN E DOB G CA N CO CS MA MI AID NOKID PN CON LN = {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN }

   Canceling F and L because they are determined by E and P.
2. BN St CN SN Z CC MPN E DOB G CA N CO CS MA MI AID NOKID PN CON LN = {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN}

   Canceling CO because it is determined by SN.
3. BN St CN SN Z CC MPN E DOB G CA N CS MA MI AID NOKID PN CON LN = {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN}

   Canceling out every other attributes because BN can define them
4. BN = {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN}

   Now, let's find the closure of proper subsets:

   BN = {BN}

Therefore, BN is a candidate key.

There is no partial dependency here so it is in 2NF.

Now, we can see that there is no transitive dependencies. Thus, this table is in 3NF.

For BCNF, we have BN -> {F, L, St, CN, SN, Z, CC, MPN, E, DOB, G, CA, N, CO, CS, MA, MI, AID, NOKID, PN, CON, LN} and BN is the super key.

**Course Table Functional Dependencies:**

CourseNumber → {CourseTitle, InstructorFirstName, InstructorLastName, InstructorPhoneNumber, InstructorEmail, InstructorRoomNumber, DepartmentName}
InstructorEmail → {InstructorFirstName, InstructorLastName, InstructorPhoneNumber, InstructorRoomNumber, DepartmentNumber}
InstructorPhoneNumber → {InstructorFirstName, InstructorLastName, InstructorEmail, InstructorRoomNumber, DepartmentNumber}

We can write this in an abbreviated form:
CN → {CT, IFN, ILN, IPN, IE, IRM, DN}
IE → {IFN, ILN, IPN, IRN, DN}
IPN → {IFN, ILN, IE, IRN, DN}

Now, the table is in 1Nf because every attribute is atomic.

The table is in 2NF.
Finding the closure:

CN CT TFN TLN IPN IE IRN $DN^{+}$ → {CN, CT, IFN, ILN, IPN, IE, IRN, DN}

Cancleing out IFN, ILN, IPN, IRN, DN because they can be determined by IE

CN CT $IE^{+}$ → {CN, CT, IFN, ILN, IPN, IE, IRN, DN}
Canceling out CT and IE because they can be determined by CN
CN $^{+}$ → {CN, CT, IFN, ILN, IPN, IE, IRN, DN}

So, CN is the candidate key because property subset is itself.

There is no partial dependency here.

Therefore, the table is in 2NF form.

Now, we can make this table in 3Nf by doing this

First table: CN and IE
Second table: IE, IFN, ILN, IP, IRN, DN

And then we can make sure that this is in 3NF.

Finally, we can make sure that these relations are in BCNF because they have a X→ A where X is a super key.

**Advisor Table Functional Dependencies:**
AdvisorID → {FirstName, LastName, Position, DepartmentName, InternalPhoneNumber, Email, RoomNumber}
Email → {FirstName, LastName, Position, DepartmentName, InternalPhoneNumber, RoomNumber}
InternalPhoneNumber → {FirstName, LastName, Position, DepartmentName, Email, RoomNumber}

Writing it in Abbreviated form:
AID → {FN, LN, P, DN, IPN, E, RN}
E → {FN, LN, P, DN, IPN, RN}
IPN → {FN, LN, P, DN, E, RN}

Now, the table is in 1NF.
Lets make it in 2NF.

Closure:
AID FN LN P DN IPN E RN $^{+}$ → {AID, FN, LN, P, DN, IPN, E, RN}

Canceling out FN LN P DN IPN and RN as E determines them
AID E $^{+}$ → {AID, FN, LN, P, DN, IPN, E, RN}

Canceling out E as AID determines it
AID $^{+}$ → {AID, FN, LN, P, DN, IPN, E, RN}

Now the proper subset of AID = {AID}
Therefore, AID is a candidate key.

It is in 2NF as there is no partial dependency only one key.

Now it is in 3NF and to remove that we can do the following:
First Table: AID, E

Second Table: E, FN, LN, P, IPN, RN

This way it will be in 3NF and also in BCNF.

**NextOfKin Table Functional Dependencies:**
NextOfKinID → {FirstName, LastName, Relationship, StreetName, CityName, Zip, CountryCode, ContactPhoneNumber}
NextOfKin table is in BCNF.

**Lease Table Functional Dependencies:**
LeaseNumber → {Duration, StudentFirstName, StudentLastName, PlaceNumber, RoomNumber, StreetName, CityName, Zip, CountryCode, StartDate, EndDate, *BannerNumber*}
Lease table is in BCNF.

**Invoice Table Functional Dependencies:**
InvoiceNumber → {Semester, FirstName, LastName, RoomNumber, StreetName, CityName, Zip, CountryCode, PaymentDue, PaymentMethod, PaymentDate, FirstReminderDate, SecondReminderDate, *BannerNumber, PlaceNumber*}
Invoice table is in BCNF.

**Room Table Functional Dependencies:**
PlaceNumber → {ApartmentNumber, RoomNumber, MonthlyRent, RoomType, StaffNumber, *HallID*}
Room table is in BCNF.

**HallOfResidence Table Functional Dependencies:**
HallID → {StreetName, CityName, Zip, CountryCode, MobilePhoneNumber}
HallOfResidence table is in BCNF.

**StudentApartment Table Functional Dependencies:**
ApartmentNumber → {StreetName, CityName, Zip, CountryCode, NumberOfRooms}
StudentApartment table is in BCNF.

**Inspection Table Functional Dependencies:**
No primary/candidate key so no functional dependencies.

**ResidenceStaff Table Functional Dependencies:**
StaffNumber → {FirstName, FirstName, LastName, Email, StreetName, CityName, StateName, Zip, CountryCode, DateOfBirth, Gender, Position, Location}
ResidenceStaff table is in BCNF.

**Step 6.2 Update the data dictionary and list of assumptions as needed.**
No need to update DDL and list of assumptions.

**Step 6.3 For each table, write the table name and write out the names, data types, and sizes of all the data items, Identify any constraints, using the conventions of the DBMS you will use for implementation.**

Student

| Names | Data Types & Sizes | Constraints |
|---|---|---|
| BannerNumber | INT | |
| FirstName | VARCHAR(50) | |
| LastName | VARCHAR(50) | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| MobilePhoneNumber | VARCHAR(20) | |
| Email | VARCHAR(70) UNIQUE | |
| DateOfBirth | DATE | |
| Gender | VARCHAR(10) | |
| Category | VARCHAR(50) | |
| Nationality | VARCHAR(50) | |
| SpecialNeeds | TEXT | |
| Comments | TEXT | |
| CurrentStatus | VARCHAR(20) | |

| | | |
|---|---|---|
| Major | VARCHAR(30) | |
| Minor | VARCHAR(30) | |
| AdvisorID | INT | fk_Student_Advisor_AdvisorID |
| NextOfKinID | INT | fk_Student_NextOfKin_NextOfKinID |
| PlaceNumber | INT | fk_Student_Room_PlaceNumber |
| CourseNumber | INT | fk_Student_Course_CourseNumber |
| LeaseNumber | INT | fk_Student_Lease_LeaseNumber |

Advisor

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| AdvisorID | INT | |
| FirstName | VARCHAR(50) | |
| LastName | VARCHAR(50) | |
| Position | VARCHAR(50) | |
| DepartmentName | VARCHAR(50) | |
| InternalPhoneNumber | VARCHAR(20) | |
| Email | VARCHAR(70) UNIQUE | |
| RoomNumber | VARCHAR(30) | |

NextOfKin

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|

| | | |
|---|---|---|
| NextOfKinID | INT | |
| FirstName | VARCHAR(50) | |
| LastName | VARCHAR(50) | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| Relationship | VARCHAR(50) | |
| ContactPhoneNumber | VARCHAR(30) | |

Course

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| CourseNumber | INT | |
| CourseTitle | VARCHAR(30) | |
| InstructorFirstName | VARCHAR(30) | |
| InstructorLastName | VARCHAR(30) | |
| StreetNameInstructorLastName | VARCHAR(30) | |
| InstructorPhoneNumber | VARCHAR(20) | |
| InstructorEmail | VARCHAR(20) | |

| | | |
|---|---|---|
| InstructorRoomNumber | INT | |
| DepartmentName | VARCHAR(50) | |

## Lease

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| LeaseNumber | INT | |
| Duration | VARCHAR(20) | |
| BannerNumber | INT | fk_Lease_Student_BannerNumber |
| StudentFirstName | VARCHAR(20) | |
| StudentLastName | VARCHAR(20) | |
| PlaceNumber | INT | fk_Lease_Room_PlaceNumber |
| RoomNumber | VARCHAR(30) | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(50) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| StartDate | DATE | |
| EndDate | DATE | |

## Invoice

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|

| | | |
|---|---|---|
| InvoiceNumber | INT | |
| Semester | VARCHAR(30) | |
| FirstName | VARCHAR(30) | |
| LastName | VARCHAR(30) | |
| BannerNumber | INT | fk_Invoice_Student_BannerNumber |
| PlaceNumber | INT | |
| RoomNumber | VARCHAR(30) | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| PaymentDue | DECIMAL(10, 2) | |
| PaymentMethod | VARCHAR(50) | |
| PaymentDate | DATE | |
| FirstReminderDate | DATE | |
| SecondReminderDate | DATE | |

## Room

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| PlaceNumber | INT | |

| | | |
|---|---|---|
| RoomNumber | VARCHAR(30) | |
| MonthlyRent | DECIMAL(10, 2) | |
| RoomType | VARCHAR(30) | |
| StaffNumber | INT | fk_Room_ResidenceStaff_StaffNumber |
| HallId | INT | fk_Room_HallOfResidence_HallId |
| ApartmentNumber | INT | fk_Room_StudentApartment_ApartmentNumber |

## HallOfResidence

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| HallId | INT | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| MobilePhoneNumber | VARCHAR(20) | |

## StudentApartment

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| ApartmentNumber | INT | |

| | | |
|---|---|---|
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| NumberOfRooms | INT | |

## Inspection

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| StaffNumber | INT | |
| StudentFirstName | VARCHAR(50) | |
| StudentLastName | VARCHAR(50) | |
| ApartmentNumber | INT | fk_Inspection_StudentApartment_ApartmentNumber |
| InspectionDate | DATE | |
| Satisfactory | VARCHAR(20) | |
| Comments | TEXT | |

## ResidenceStaff

| **Names** | **Data Types & Sizes** | **Constraints** |
|---|---|---|
| StaffNumber | INT | |
| FirstName | VARCHAR(30) | |

| | | |
|---|---|---|
| LastName | VARCHAR(30) | |
| Email | VARCHAR(70) UNIQUE | |
| StreetName | VARCHAR(30) | |
| CityName | VARCHAR(20) | |
| StateName | VARCHAR(20) | |
| Zip | INT | |
| CountryCode | VARCHAR(10) | |
| DateOfBirth | DATE | |
| Gender | VARCHAR(10) | |
| Position | VARCHAR(20) | |
| Location | VARCHAR(20) | |

**Step 6.4 Write and execute SQL statements to create all the tables needed to implement the design.**

```
CREATE TABLE CourseInstructor (
        CourseNumber INT PRIMARY KEY,
   CourseTitle VARCHAR(30),
   InstructorEmail VARCHAR(70) UNIQUE
);


CREATE TABLE InstructorDetail (
   InstructorFirstName VARCHAR(30),
   InstructorLastName VARCHAR(30),
   InstructorPhoneNumber VARCHAR(20),
   InstructorEmail VARCHAR(70) UNIQUE,
   InstructorRoomNumber INT,
   DepartmentName VARCHAR(50),
   FOREIGN KEY(InstructorEmail) REFERENCES CourseInstructor(InstructorEmail)
);
```

```
INSERT INTO CourseInstructor (CourseNumber, CourseTitle, InstructorEmail)
VALUES (1, 'Database Systems', 'alicejohnson@university.edu'),
(2, 'Calculus', 'davidsmith@university.edu'),
(3, 'Physics 101', 'robertbrown@university.edu'),
(4, 'Chemistry 101', 'michaelwhite@university.edu'),
(5, 'Biology 101', 'laurablack@university.edu' ),
(6, 'Engineering 101', 'chrisgreen@university.edu' ),
(7, 'Economics 101', 'sarahblue@university.edu'),
(8, 'History 101', 'jamesyellow@university.edu'),
(9, 'Literature 101', 'emilypurple@university.edu'),
(10, 'Philosophy 101', 'annaorange@university.edu');


INSERT INTO InstructorDetail (InstructorFirstName, InstructorLastName,
InstructorPhoneNumber, InstructorEmail, InstructorRoomNumber, DepartmentName)
VALUES ('Alice', 'Johnson', '(710) 234-6789', 'alicejohnson@university.edu', 102, 'Computer
Science'),
('David', 'Smith', '(669) 345-6790', 'davidsmith@university.edu', 103, 'Mathematics'),
('Robert', 'Brown', '(345) 456-6791', 'robertbrown@university.edu', 104, 'Physics'),
('Michael', 'White', '(134) 567-6792', 'michaelwhite@university.edu', 105, 'Chemistry'),
('Laura', 'Black', '(298) 678-6793', 'laurablack@university.edu', 106, 'Biology'),
('Chris', 'Green', '(471) 789-6794', 'chrisgreen@university.edu', 107, 'Engineering'),
('Sarah', 'Blue', '(562) 890-6795', 'sarahblue@university.edu', 108, 'Economics'),
('James', 'Yellow', '(680) 901-6796', 'jamesyellow@university.edu', 109, 'History'),
('Emily', 'Purple', '(710) 012-6797', 'emilypurple@university.edu', 110, 'Literature'),
('Anna', 'Orange', '(499) 123-6798', 'annaorange@university.edu', 111, 'Philosophy');


SELECT * FROM CourseInstructor;
SELECT * FROM InstructorDetail;
```

After execution:

| CourseNumber | CourseTitle | InstructorEmail |
|---|---|---|
| 1 | Database Systems | alicejohnson@university.edu |
| 2 | Calculus | davidsmith@university.edu |
| 3 | Physics 101 | robertbrown@university.edu |
| 4 | Chemistry 101 | michaelwhite@university.edu |
| 5 | Biology 101 | laurablack@university.edu |
| 6 | Engineering 101 | chrisgreen@university.edu |
| 7 | Economics 101 | sarahblue@university.edu |
| 8 | History 101 | jamesyellow@university.edu |
| 9 | Literature 101 | emilypurple@university.edu |
| 10 | Philosophy 101 | annaorange@university.edu |
| NULL | NULL | NULL |

| InstructorFirstName | InstructorLastName | InstructorPhoneNumber | InstructorEmail | InstructorRoomNumber | DepartmentName |
|---|---|---|---|---|---|
| Alice | Johnson | (710) 234-6789 | alicejohnson@university.edu | 102 | Computer Science |
| David | Smith | (669) 345-6790 | davidsmith@university.edu | 103 | Mathematics |
| Robert | Brown | (345) 456-6791 | robertbrown@university.edu | 104 | Physics |
| Michael | White | (134) 567-6792 | michaelwhite@university.edu | 105 | Chemistry |
| Laura | Black | (298) 678-6793 | laurablack@university.edu | 106 | Biology |
| Chris | Green | (471) 789-6794 | chrisgreen@university.edu | 107 | Engineering |
| Sarah | Blue | (562) 890-6795 | sarahblue@university.edu | 108 | Economics |
| James | Yellow | (680) 901-6796 | jamesyellow@university.edu | 109 | History |
| Emily | Purple | (710) 012-6797 | emilypurple@university.edu | 110 | Literature |
| Anna | Orange | (499) 123-6798 | annaorange@university.edu | 111 | Philosophy |

```
CREATE TABLE AdvisorEmail (
   AdvisorID INT PRIMARY KEY,
   Email VARCHAR(70) UNIQUE
);

CREATE TABLE AdvisorDetail (
        Email VARCHAR(70) UNIQUE,
        FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Position VARCHAR(50),
   DepartmentName VARCHAR(50),
   InternalPhoneNumber VARCHAR(20),
   RoomNumber VARCHAR(30)
);


INSERT INTO AdvisorEmail(AdvisorID, Email)
VALUES (1, 'johndoe@university.edu'),
(2, 'janesmith@university.edu'),
(3, 'alicejohnson@university.edu'),
(4, 'bobbrown@university.edu'),
```

```
(5,'carolwhite@university.edu'),
(6, 'davidblack@university.edu'),
(7, 'evegreen@university.edu'),
(8, 'frankblue@university.edu'),
(9, 'graceyellow@university.edu'),
(10, 'hankpurple@university.edu');

INSERT INTO AdvisorDetail (Email, FirstName, LastName, Position, DepartmentName,
InternalPhoneNumber, RoomNumber)
VALUES ('johndoe@university.edu', 'John', 'Doe', 'Professor', 'Computer Science', '1234', '101'),
('janesmith@university.edu','Jane', 'Smith', 'Associate Professor', 'Mathematics', '1235', '102'),
('alicejohnson@university.edu', 'Alice', 'Johnson', 'Lecturer', 'Physics', '1236','103'),
('bobbrown@university.edu', 'Bob', 'Brown', 'Senior Lecturer', 'Chemistry', '1237', '104'),
('carolwhite@university.edu', 'Carol', 'White', 'Assistant Professor', 'Biology', '1238', '105'),
('davidblack@university.edu','David', 'Black', 'Professor', 'Engineering', '1239', '106'),
('evegreen@university.edu', 'Eve', 'Green', 'Senior Lecturer', 'Economics', '1240', '107'),
('frankblue@university.edu',  'Frank', 'Blue', 'Lecturer', 'History', '1241', '108'),
('graceyellow@university.edu', 'Grace', 'Yellow', 'Associate Professor', 'Literature', '1242', '109'),
('hankpurple@university.edu', 'Hank', 'Purple', 'Professor', 'Philosophy', '1243', '110');

SELECT * FROM AdvisorEmail;
SELECT * FROM AdvisorDetail;
```

Afer Execution:

| AdvisorID | Email |
|---|---|
| 3 | alicejohnson@university.edu |
| 4 | bobbrown@university.edu |
| 5 | carolwhite@university.edu |
| 6 | davidblack@university.edu |
| 7 | evegreen@university.edu |
| 8 | frankblue@university.edu |
| 9 | graceyellow@university.edu |
| 10 | hankpurple@university.edu |
| 2 | janesmith@university.edu |
| 1 | johndoe@university.edu |
| NULL | NULL |

| Email | FirstName | LastName | Position | DepartmentName | InternalPhoneNumber | RoomNumber |
|---|---|---|---|---|---|---|
| johndoe@university.edu | John | Doe | Professor | Computer Science | 1234 | 101 |
| janesmith@university.edu | Jane | Smith | Associate Professor | Mathematics | 1235 | 102 |
| alicejohnson@university.edu | Alice | Johnson | Lecturer | Physics | 1236 | 103 |
| bobbrown@university.edu | Bob | Brown | Senior Lecturer | Chemistry | 1237 | 104 |
| carolwhite@university.edu | Carol | White | Assistant Professor | Biology | 1238 | 105 |
| davidblack@university.edu | David | Black | Professor | Engineering | 1239 | 106 |
| evegreen@university.edu | Eve | Green | Senior Lecturer | Economics | 1240 | 107 |
| frankblue@university.edu | Frank | Blue | Lecturer | History | 1241 | 108 |
| graceyellow@university.edu | Grace | Yellow | Associate Professor | Literature | 1242 | 109 |
| hankpurple@university.edu | Hank | Purple | Professor | Philosophy | 1243 | 110 |

**Step 6.6 Insert about five records in each table, preserving all constraints. Put in enough data to demonstrate how the database will function.**

**Answer:**

```
INSERT INTO AdvisorEmail(AdvisorID, Email)
VALUES (1, 'johndoe@university.edu'),
(2, 'janesmith@university.edu'),
(3, 'alicejohnson@university.edu'),
(4, 'bobbrown@university.edu'),
(5,'carolwhite@university.edu'),
(6, 'davidblack@university.edu'),
(7, 'evegreen@university.edu'),
(8, 'frankblue@university.edu'),
(9, 'graceyellow@university.edu'),
(10, 'hankpurple@university.edu');

INSERT INTO AdvisorDetail (Email, FirstName, LastName, Position, DepartmentName,
InternalPhoneNumber, RoomNumber)
VALUES ('johndoe@university.edu', 'John', 'Doe', 'Professor', 'Computer Science', '1234', '101'),
('janesmith@university.edu','Jane', 'Smith', 'Associate Professor', 'Mathematics', '1235', '102'),
('alicejohnson@university.edu', 'Alice', 'Johnson', 'Lecturer', 'Physics', '1236','103'),
('bobbrown@university.edu', 'Bob', 'Brown', 'Senior Lecturer', 'Chemistry', '1237', '104'),
('carolwhite@university.edu', 'Carol', 'White', 'Assistant Professor', 'Biology', '1238', '105'),
('davidblack@university.edu','David', 'Black', 'Professor', 'Engineering', '1239', '106'),
('evegreen@university.edu', 'Eve', 'Green', 'Senior Lecturer', 'Economics', '1240', '107'),
('frankblue@university.edu',  'Frank', 'Blue', 'Lecturer', 'History', '1241', '108'),
('graceyellow@university.edu', 'Grace', 'Yellow', 'Associate Professor', 'Literature', '1242', '109'),
('hankpurple@university.edu', 'Hank', 'Purple', 'Professor', 'Philosophy', '1243', '110');

INSERT INTO CourseInstructor (CourseNumber, CourseTitle, InstructorEmail)
VALUES (1, 'Database Systems', 'alicejohnson@university.edu'),
(2, 'Calculus', 'davidsmith@university.edu'),
(3, 'Physics 101', 'robertbrown@university.edu'),
(4, 'Chemistry 101', 'michaelwhite@university.edu'),
```

(5, 'Biology 101', 'laurablack@university.edu' ),
(6, 'Engineering 101', 'chrisgreen@university.edu' ),
(7, 'Economics 101', 'sarahblue@university.edu'),
(8, 'History 101', 'jamesyellow@university.edu'),
(9, 'Literature 101', 'emilypurple@university.edu'),
(10, 'Philosophy 101', 'annaorange@university.edu');

INSERT INTO InstructorDetail (InstructorFirstName, InstructorLastName, InstructorPhoneNumber, InstructorEmail, InstructorRoomNumber, DepartmentName)
VALUES ('Alice', 'Johnson', '(710) 234-6789', 'alicejohnson@university.edu', 102, 'Computer Science'),
('David', 'Smith', '(669) 345-6790', 'davidsmith@university.edu', 103, 'Mathematics'),
('Robert', 'Brown', '(345) 456-6791', 'robertbrown@university.edu', 104, 'Physics'),
('Michael', 'White', '(134) 567-6792', 'michaelwhite@university.edu', 105, 'Chemistry'),
('Laura', 'Black', '(298) 678-6793', 'laurablack@university.edu', 106, 'Biology'),
('Chris', 'Green', '(471) 789-6794', 'chrisgreen@university.edu', 107, 'Engineering'),
('Sarah', 'Blue', '(562) 890-6795', 'sarahblue@university.edu', 108, 'Economics'),
('James', 'Yellow', '(680) 901-6796', 'jamesyellow@university.edu', 109, 'History'),
('Emily', 'Purple', '(710) 012-6797', 'emilypurple@university.edu', 110, 'Literature'),
('Anna', 'Orange', '(499) 123-6798', 'annaorange@university.edu', 111, 'Philosophy');

**Step 6.7 Write SQL statements that will process five non-routine requests for information from the database just created. For each, write the request in English, followed by the corresponding SQL command**

Query 1: Identify all vulnerable students with special needs and their housing situation. Get banner number (id), name, any special needs, any comments, the type of room each student is staying in, rent, and place number.

```
11    # Module 5
12    # Query 1: Get all students banner number, name, special needs, and comments.
13    # additionally pull Advisor name and the type of room they are staying in, and place number
14 •  SELECT s.BannerNumber, CONCAT(s.LastName, ', ' , s.FirstName) AS StudentName, s.SpecialNeeds, s.Comments,
15    r.RoomType, r.MonthlyRent, r.PlaceNumber
16    FROM Student s
17    JOIN Room r ON s.PlaceNumber = r.PlaceNumber;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | StudentName | SpecialNeeds | Comments | RoomType | MonthlyRent | PlaceNumber |
|---|---|---|---|---|---|---|
| 1 | Taylor, Mark | None | Active and thriving | HallOfResidence | 500.00 | 1 |
| 2 | Brown, Lisa | Low vision | Needs assistance with reading materials | HallOfResidence | 450.00 | 2 |
| 3 | Smith, James | None | No comments | StudentApartment | 600.00 | 3 |
| 4 | Johnson, Emily | Limb leg | Requires accessible seating | HallOfResidence | 400.00 | 4 |
| 5 | White, David | None | Active participant in clubs | HallOfResidence | 550.00 | 5 |
| 6 | Black, Sarah | Hearing impairment | Uses hearing aids in class | StudentApartment | 650.00 | 6 |
| 7 | Green, Chris | None | Participates in sports | HallOfResidence | 700.00 | 7 |
| 8 | Blue, Laura | None | Active in student government | HallOfResidence | 600.00 | 8 |
| 9 | Yellow, Michael | None | Volunteer at local shelters | StudentApartment | 500.00 | 9 |
| 10 | Purple, Anna | None | Excels in academics | HallOfResidence | 550.00 | 10 |

Query 2:
List all inspections on August 16, 2024 where inspection noted “Minor cleaning required.”
Get date, staff name, student name, apartment number, satisfactory or not, and any comments.

```
65 •  SELECT i.InspectionDate, CONCAT(r.FirstName, ' ', r.LastName) AS StaffName,
66    CONCAT(i.StudentFirstName, ' ', i.StudentLastName) AS StudentName,
67     i.ApartmentNumber, i.Satisfactory, i.Comments
68    FROM Inspection i
69    JOIN ResidenceStaff r ON i.StaffNumber = r.StaffNumber
70    WHERE i.Comments = "Minor cleaning required."
71    AND i.InspectionDate = "2024-08-16";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| InspectionDate | StaffName | StudentName | ApartmentNumber | Satisfactory | Comments |
|---|---|---|---|---|---|
| 2024-08-16 | Laura Brown | Lisa Brown | 2 | Yes | Minor cleaning required. |

Query 3:
List all students next of kin who have the same last name as another resident staff number.
Pull student’s BannerNumber, name, nationality, next of kin relationship to student, next of kin full name, and resident staff full name.

```
74 •  SELECT s.BannerNumber, CONCAT(s.FirstName, ' ', s.LastName) AS StudentName, s.Nationality,
75    kin.Relationship AS KinRelationship, CONCAT(kin.FirstName, ' ', kin.LastName) AS NextOfKinName,
76    CONCAT(rs.FirstName, ' ', rs.LastName) AS ResidentStaffName
77    FROM Student s
78    JOIN NextOfKin kin ON kin.NextOfKinID = s.NextOfKinID
79    JOIN ResidenceStaff rs ON rs.LastName = kin.LastName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | StudentName | Nationality | KinRelationship | NextOfKinName | ResidentStaffName |
|---|---|---|---|---|---|
| 2 | Lisa Brown | US | Father | Peter Smith | Mike Smith |
| 4 | Emily Johnson | US | Sibling | Paul Brown | Laura Brown |
| 5 | David White | US | Mother | Lucy White | Chris White |
| 6 | Sarah Black | US | Father | James Black | Anna Black |
| 7 | Chris Green | US | Guardian | Emily Green | James Green |
| 8 | Laura Blue | US | Sibling | Michael Blue | Sarah Blue |
| 9 | Michael Yellow | US | Mother | Sarah Yellow | Robert Yellow |
| 10 | Anna Purple | US | Father | Tom Purple | Emily Purple |

Query 4:
Identify all NextOfKin that do not live in the same city as the student.
Get student banner number, name, city, zip code, next of kin relationship, next of kin name,

```
26    # Query 4
27 •  SELECT s.BannerNumber, CONCAT(s.FirstName, ' ', s.LastName) AS StudentName, s.CityName, s.Zip,
28    kin.Relationship, CONCAT(kin.FirstName, ' ', kin.LastName) AS NextOfKinName, kin.CityName, kin.Zip
29    FROM Student s
30    JOIN NextOfKin kin ON kin.NextOfKinID = s.NextOfKinID
31    WHERE s.CityName != kin.CityName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| BannerNumber | StudentName | CityName | Zip | Relationship | NextOfKinName | CityName | Zip |
|---|---|---|---|---|---|---|---|
| 2 | Lisa Brown | Berkeley | 94704 | Father | Peter Smith | Tracy | 95376 |
| 3 | James Smith | Berkeley | 94704 | Guardian | Mary Johnson | Castro Valley | 94546 |
| 4 | Emily Johnson | Berkeley | 94704 | Sibling | Paul Brown | Richmond | 94801 |
| 5 | David White | Berkeley | 94704 | Mother | Lucy White | Oakland | 94612 |
| 6 | Sarah Black | Berkeley | 94704 | Father | James Black | San Leandro | 94577 |
| 7 | Chris Green | Berkeley | 94704 | Guardian | Emily Green | Hayward | 94541 |
| 8 | Laura Blue | Berkeley | 94704 | Sibling | Michael Blue | Dublin | 94568 |
| 9 | Michael Yellow | Berkeley | 94704 | Mother | Sarah Yellow | Fremont | 94536 |
| 10 | Anna Purple | Berkeley | 94704 | Father | Tom Purple | Pleasanton | 94566 |

Query 5:
Get a list of all students and resident staff living on the same street who are of the same gender.
Get both student and staff name, street, & gender as well as type of room.

```
34 •  SELECT rm.RoomType,
35    CONCAT(s.FirstName, ' ', s.LastName) AS StudentName, s.StreetName AS StudentStreet, s.Gender,
36    CONCAT(rs.FirstName, ' ', rs.LastName) AS StaffName, rs.StreetName AS StaffStreet, rs.Gender
37    FROM Student s
38    JOIN Room rm ON rm.PlaceNumber = s.PlaceNumber
39    JOIN ResidenceStaff rs ON rs.StaffNumber = rm.StaffNumber
40    WHERE s.Gender = rs.Gender;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| RoomType | StudentName | StudentStreet | Gender | StaffName | StaffStreet | Gender |
|---|---|---|---|---|---|---|
| HallOfResidence | Mark Taylor | 101 University Ave | Male | Mike Smith | 101 Campus Block A | Male |
| HallOfResidence | Lisa Brown | 102 University Ave | Female | Laura Brown | 102 Campus Block A | Female |
| StudentApartment | James Smith | 103 University Ave | Male | Chris White | 103 Campus Block B | Male |
| HallOfResidence | Emily Johnson | 104 University Ave | Female | Anna Black | 104 Campus Block B | Female |
| HallOfResidence | David White | 105 University Ave | Male | James Green | 105 Campus Block C | Male |
| StudentApartment | Sarah Black | 106 University Ave | Female | Sarah Blue | 106 Campus Block C | Female |
| HallOfResidence | Chris Green | 107 University Ave | Male | Robert Yellow | 107 Campus Block D | Male |
| HallOfResidence | Laura Blue | 108 University Ave | Female | Emily Purple | 108 Campus Block D | Female |
| StudentApartment | Michael Yellow | 109 University Ave | Male | Paul Orange | 109 Campus Block E | Male |
| HallOfResidence | Anna Purple | 110 University Ave | Female | Diana Pink | 110 Campus Block E | Female |