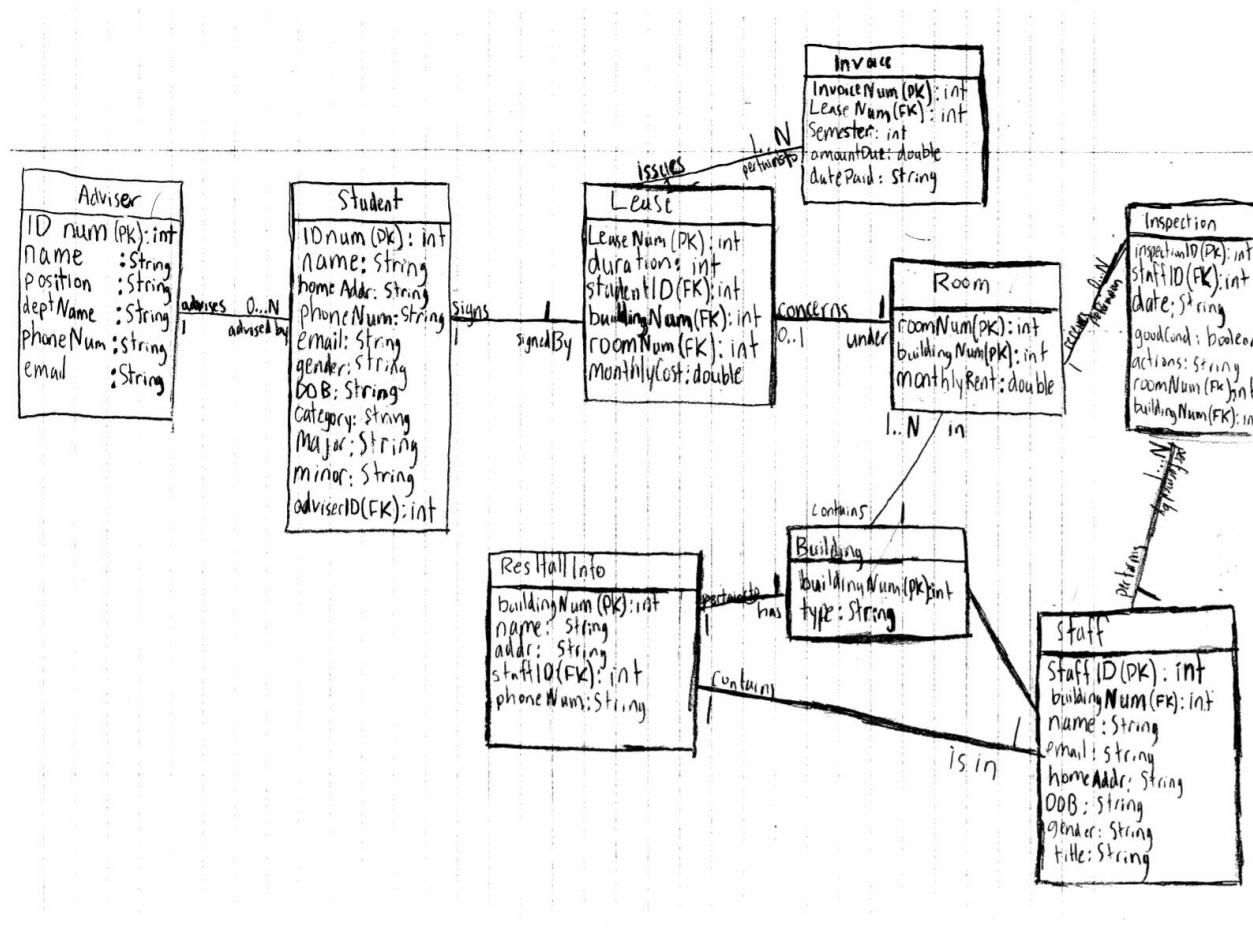


Conceptual Database Design:

Final E-R Diagram:



Logical Database Design:

Room(room_num, building_num, monthly_rent)

Inspection(inspection_ID, staff_ID, date, good_cond, actions, room_num, building_num)

Building(building_num, type)

ResHallInfo(building_num, name, addr, staff_ID, phone_num)

Staff(staff_ID, building_num, name, email, home_addr, dob, gender, title)

Adviser(id_num, name, position, dept_name, phone_num, email);

Student(id_num, name, home_addr, phone_num, email, gender, dob, category, major, minor, adviser_id);

Lease(lease_num,duration,student_id,building_num,room_num,monthly_cost);
Invoice(invoice_num,lease_num,semester,amount_due,date_paid);

Normalization Analysis:

Functional Dependencies:

In the Adviser relation:

IDnum \rightarrow name

IDnum \rightarrow position

IDnum \rightarrow deptName

IDnum \rightarrow phoneNum

IDnum \rightarrow email

In the Student relation:

IDnum \rightarrow name

IDnum \rightarrow homeAddr

IDnum \rightarrow phoneNum

IDnum \rightarrow email

IDnum \rightarrow gender

IDnum \rightarrow dob

IDnum \rightarrow category

IDnum \rightarrow major

IDnum \rightarrow minor

IDnum \rightarrow adviserID

In the Lease relation:

leaseNum \rightarrow duration

leaseNum \rightarrow studentID

leaseNum \rightarrow buildingNum

leaseNum \rightarrow roomNum

leaseNum \rightarrow monthlyCost

In the Invoice relation:

invoiceNum \rightarrow leaseNum

invoiceNum \rightarrow semester

invoiceNum \rightarrow amountDue

invoiceNum \rightarrow datePaid

In the Room relation:

{roomNum, buildingNum} \rightarrow monthlyRent

In the Inspection relation:

inspectionID \rightarrow staffID
inspectionID \rightarrow date
inspectionID \rightarrow goodCond
inspectionID \rightarrow actions
inspectionID \rightarrow roomNum
inspectionID \rightarrow buildingNum

In the ResHallInfo relation:

buildingNum \rightarrow name
buildingNum \rightarrow addr
buildingNum \rightarrow staffID
buildingNum \rightarrow phoneNum

In the Building relation:

buildingNum \rightarrow type

In the Staff relation:

staffID \rightarrow buildingName
staffID \rightarrow name
staffID \rightarrow email
staffID \rightarrow homeAddr
staffID \rightarrow dob
staffID \rightarrow gender
staffID \rightarrow title

Normalization:

1NF is achieved in the tables through a constraint limiting attributes like “name,” “email,” and “phone” to include only one value. 2NF is trivial in Adviser, Student, Lease, Invoice, Inspection, Staff, Building, ResHallInfo since each has only a single key as its primary key (and they are in 1NF). Room is in 2NF because its non-prime attribute, monthlyRent, cannot be determined from just the roomNum or just the buildingNum. Knowing that the table is in 2NF, we verify that it is in 3NF by checking that there are no transitive functional dependencies. We know from our list of functional dependencies that the attributes on the left (which are all primary keys) do not appear on the right side of a dependency $X \rightarrow Y$. So there must not exist transitive functional dependencies. The tables are in Boyce-Codd normal form (BCNF) since for every functional dependency $X \rightarrow Y$, X is a superkey (a candidate key, in fact) for the schema.

Query Description

These two queries provide an insight as to how many students may be in a certain major and how many students each advisor may be advising. In a real world scenario, this feature would be useful to even workloads between advisors and keep major evenly distributed with students.