Part 1

T1.1

Relation A:

Superkeys: - set of keys
{EmpID}, {SSN}, {Email}, {Phone}, {EmpID, Name}, {SSN, Department}

Candidate keys: - unique keys
{EmpID}, {SSN}, {Email}, {Phone}

Primary key: - the main I chose
For p.k I chose {EmpID}, because it is an artificial key which doesn't get implemented by humans, unchangeable,short and unique.

As *Phone Number* is identified as *Candidate Key* - then two people can not have the same phone number in database

Relation B:
{StudentID, CourseCode, Section, Semester, Year} - I chose as the minimum attributes. Because each attribute describes smth. unique and related for a student
StudentID identifies the student unique feature
CourseCode identifies the course the student is taking
Section distinguishes different sections of the same course
Semester needed because the same course can be repeated in different semesters
Year shows academic years

T 1.2
Student.AdvisorID = Professor.Prof.ID (every stud has prof as an advisor)
Student.Major = Department.DeptCode / DeptName
Professor.Department = Department.DeptCode (the same)
Course.DepartmentCode= Department.DeptCode (the same)
Department.ChairID = Professor.ProfID
Enrollment.StudentID = Student.StudentID
Enrollment.CourseID = Course.CourseID

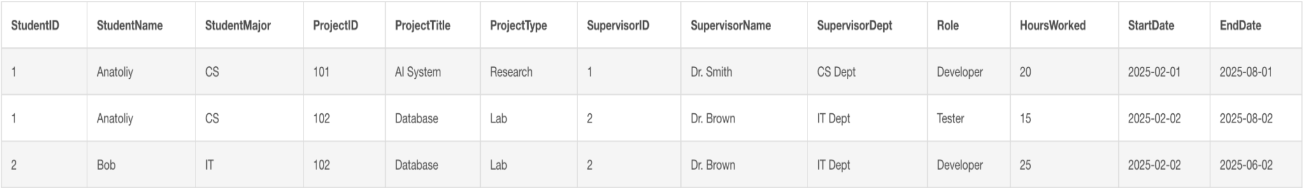
Part 4

T 4.1

1. StudentID -> StudentName , StudentMajor. ProjectID -> ProjectTitle,ProjectType. SupervisorID -> SupervisorName,SupervisorDept, StudentID -> Role. ProjectID -> Start/EndDate
2. **Update** problem: If a student changes their major, it must be updated in multiple rows. Forgetting to update one row leads to inconsistencies.
   Delete problem: If the last student working on a project is removed, the project information (ProjectTitle, ProjectType) is lost, If the last project under a supervisor is removed, the supervisor's information is also lost.

For 1NF: all attributes must be **unique** (no repeating valued attributes).In the original `StudentProject` table, all attributes such as `StudentName`, `ProjectTitle`, and

`SupervisorName` are already unique.



StudentID	StudentName	StudentMajor	ProjectID	ProjectTitle	ProjectType	SupervisorID	SupervisorName	SupervisorDept	Role	HoursWorked	StartDate	EndDate
1	Anatoliy	CS	101	AI System	Research	1	Dr. Smith	CS Dept	Developer	20	2025-02-01	2025-08-01
1	Anatoliy	CS	102	Database	Lab	2	Dr. Brown	IT Dept	Tester	15	2025-02-02	2025-08-02
2	Bob	IT	102	Database	Lab	2	Dr. Brown	IT Dept	Developer	25	2025-02-02	2025-06-02

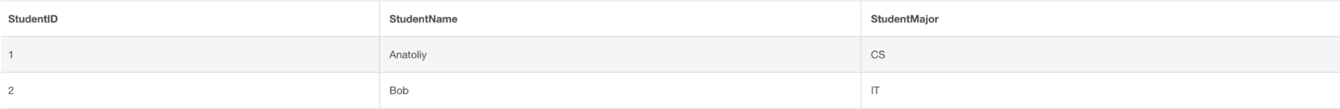
here is an example of nit 1NF table (some values repeat many times)

For 2NF:

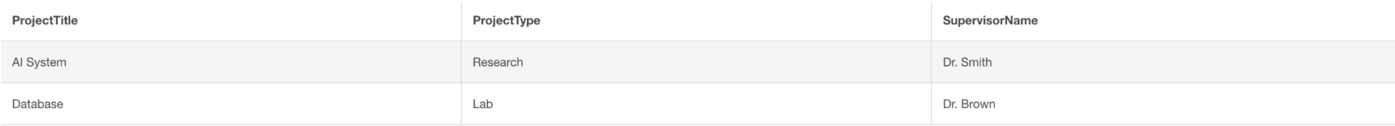
the table must be in 1NF and there should be **no partial dependency** — meaning that non-key attributes should depend on the **whole primary key**, not just part of it.

In our case, the composite primary key is `(StudentID, ProjectID)`

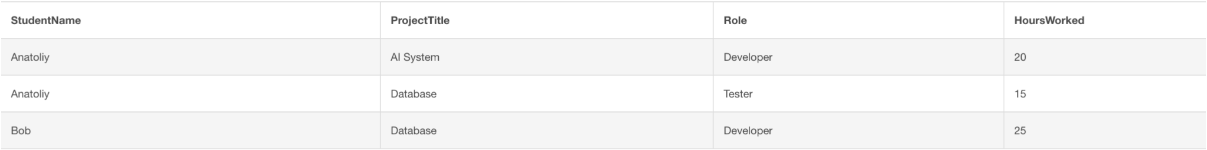
As a decision we divide the total table for parts (Student, Project, Supervisor)



StudentID	StudentName	StudentMajor
1	Anatoliy	CS
2	Bob	IT



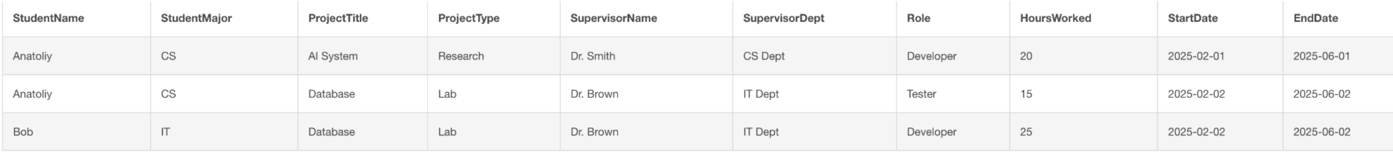
ProjectTitle	ProjectType	SupervisorName
AI System	Research	Dr. Smith
Database	Lab	Dr. Brown



StudentName	ProjectTitle	Role	HoursWorked
Anatoliy	AI System	Developer	20
Anatoliy	Database	Tester	15
Bob	Database	Developer	25

For 3NF:

Remove transitive dependencies = SupervisorDept is stored in Supervisor, not mixed into other tables.



StudentName	StudentMajor	ProjectTitle	ProjectType	SupervisorName	SupervisorDept	Role	HoursWorked	StartDate	EndDate
Anatoliy	CS	AI System	Research	Dr. Smith	CS Dept	Developer	20	2025-02-01	2025-06-01
Anatoliy	CS	Database	Lab	Dr. Brown	IT Dept	Tester	15	2025-02-02	2025-06-02
Bob	IT	Database	Lab	Dr. Brown	IT Dept	Developer	25	2025-02-02	2025-06-02

T 4.2

1)Primary Key: (StudentID, CourseID, TimeSlot, Room)

- 2) StudentID → StudentMajor
- CourseID → CourseName
- InstructorID → InstructorName
- (TimeSlot, Room) → Building
  (CourseID, TimeSlot, Room) → InstructorID
- (StudentID, CourseID, TimeSlot, Room) → all other attributes

3)A relation is in BCNF if for every dependency **X** → **Y**, X is a superkey.

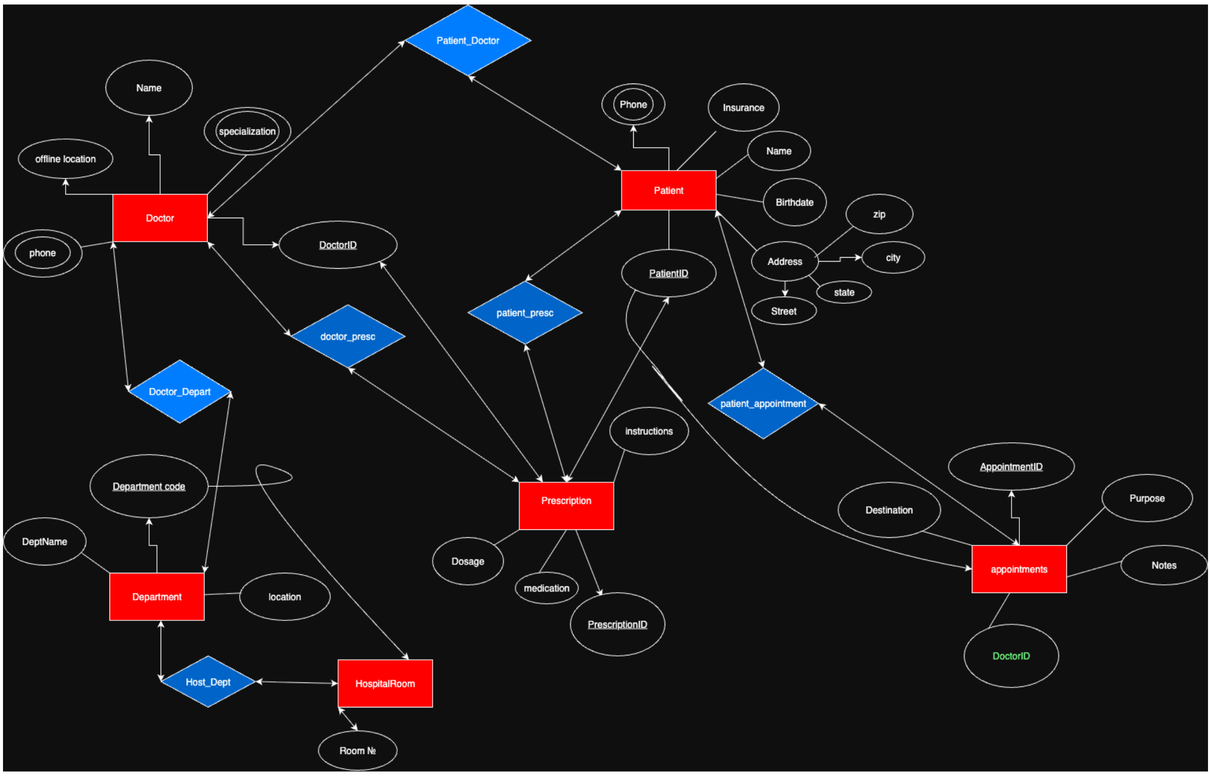
StudentID != StudentMajor (StudentID is not a superkey)
CourseID != CourseName (CourseID is not a superkey)
InstructorID != InstructorName (InstructorID is not a superkey)
(TimeSlot, Room) != Building (not a superkey)
(CourseID, TimeSlot, Room) != InstructorID (not a superkey)
4) We decompose the table into multiple relations that satisfy BCNF:

1. **Student(StudentID, StudentMajor)**
2. **Course(CourseID, CourseName)**
3. **Instructor(InstructorID, InstructorName)**
4. **Room(TimeSlot, Room, Building)**
5. **CourseSection(CourseID, TimeSlot, Room, InstructorID)**
6. **Enrollment(StudentID, CourseID, TimeSlot, Room**

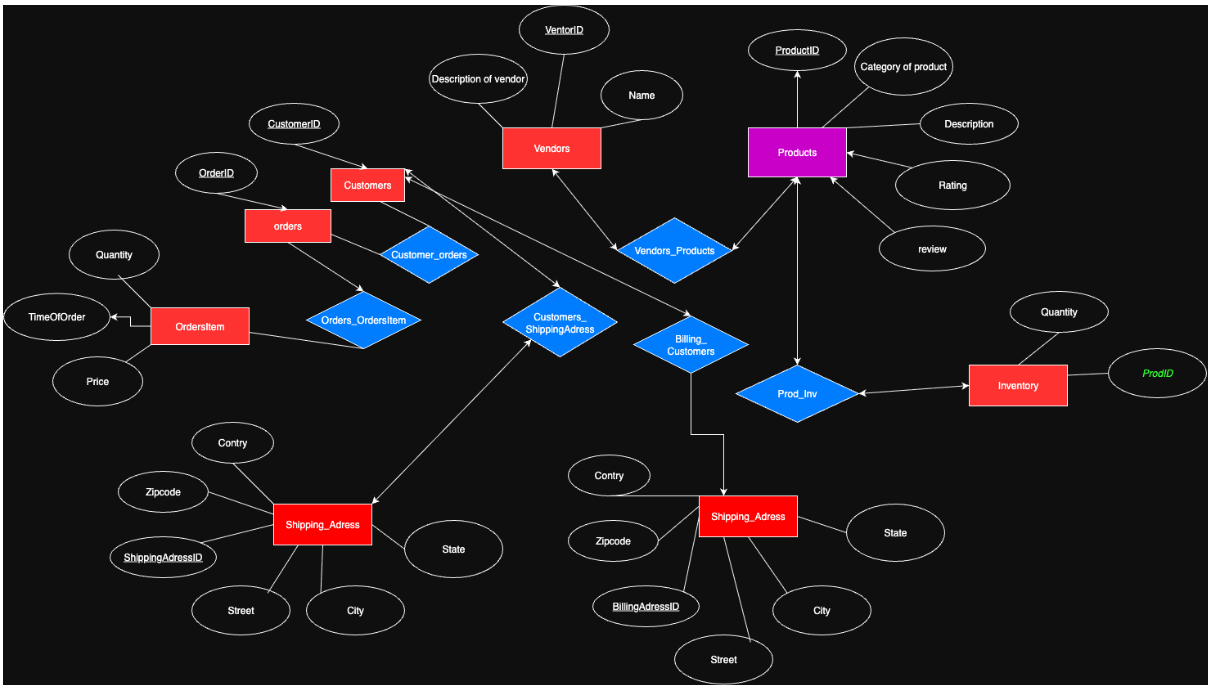
5) Loss of information: No information is lost during the decomposition, because all attributes and dependencies are preserved.

ER DIAGRAMS

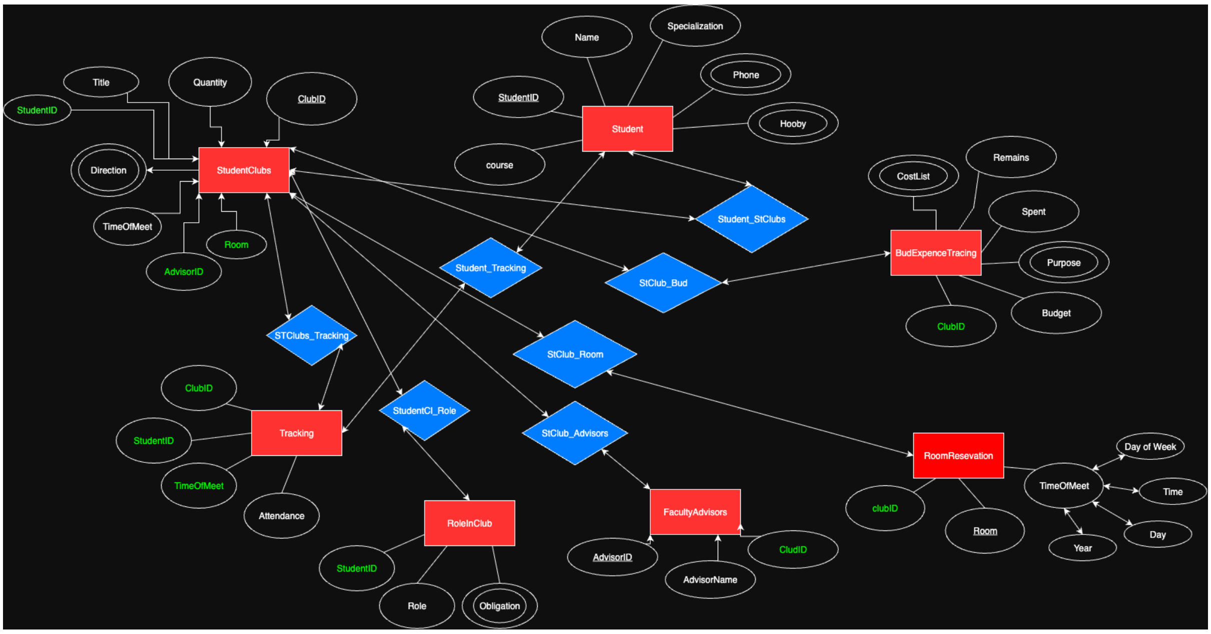
T2.1



T2.2


VentorID
Description of vendor
Name
Vendors
ProductID
Category of product
Description
Products
Rating
review
CustomerID
OrderID
Customers
orders
Customer_orders
Vendors_Products
Quantity
TimeOfOrder
OrdersItem
Orders_OrdersItem
Customers_
ShippingAdress
Billing_
Customers
Price
Prod_Inv
Quantity
Inventory
ProdID
Contry
Zipcode
Shipping_Adress
ShippingAdressID
State
Street
City
Contry
Shipping_Adress
Zipcode
State
BillingAdressID
City
Street

T5.1


Name
Specialization
Phone
Title
Quantity
ClubID
StudentID
Hooby
StudentID
Student
course
StudentClubs
Direction
Remains
CostList
Spent
TimeOfMeet
Room
Student_StClubs
AdvisorID
Student_Tracking
StClub_Bud
BudExpenceTracing
Purpose
Budget
ClubID
STClubs_Tracking
StClub_Room
ClubID
StudentCl_Role
StudentID
Tracking
StClub_Advisors
RoomResevation
Day of Week
TimeOfMeet
TimeOfMeet
Time
Attendance
clubID
RoleInClub
FacultyAdvisors
Room
Day
Year
StudentID
CludID
AdvisorID
AdvisorName
Role
Obligation