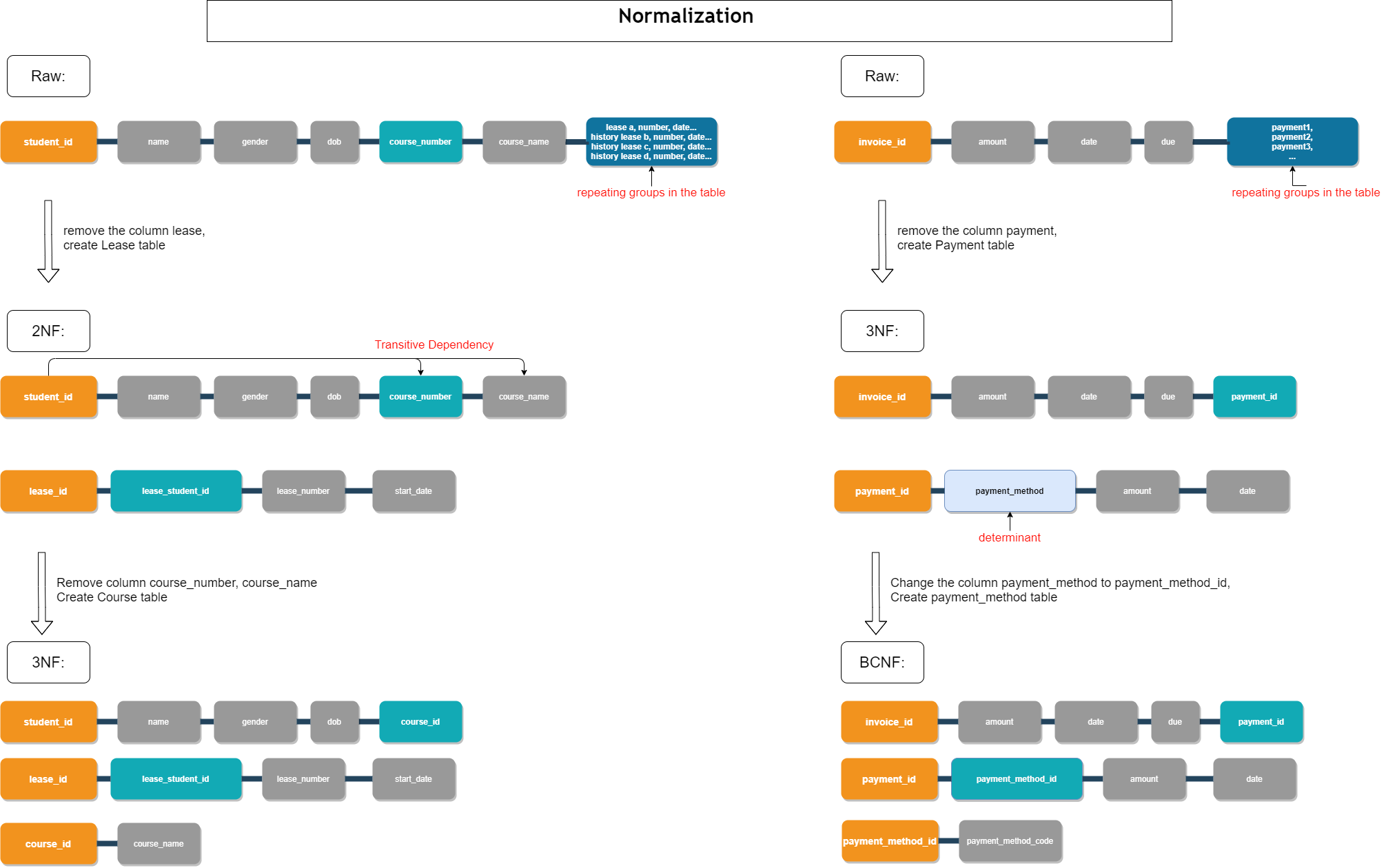
Assumption:

* The basic relationship of this case is Property -> Room -> Student. We have 3 types of property (Hall, Flat, Studio) which are saved in the table PROPERTY\_TYPE. Particularly, if a Property is Flat, it has an attribute FLAT\_NUMBER, so we make it unique and nullable in table PROPERTY.
* Entity Room has 3 unique and nullable attributes as the unique identification: HALL\_PLACE\_NUMBER, FLAT\_PLACE\_NUMBER, STUDIO\_UNIT\_NUMBER.
* One student can sign multiple leases in different time period, so lease can not be an identification to determine a room’s current student. Also, it is possible that a student rent multiple rooms. So, we give Student-Room one-to-many relationship.
* We use table COMM\_MASTER to store all contact details. COMM\_MASTER has an attribute Type (such as PHONE\_WORK, PHONE\_HOME, FAX, MOB, ADDRESS, POST), and it is stored in the table COMM\_TYPE.
* COMM\_MASTER is in many-to-many relationships with entities that have contacts such as PROPERTY and NEXT\_OF\_KIN. We use third tables to deal with many-to-many relationships, such as the table PROPERTY\_COMM and NEXT\_OF\_KIN\_COMM.
* Appendix A does not give information about whether an Advisor live with student, so we give it a varchar type for room number, instead of room id.
* The degree course entity also has an attribute Room Number, and it's the classroom in school which shall has different attributes with accommodation rooms, so we put a varchar type for it at this stage.
* The entity Advisor and entity Course both have attribute Department. We assume that they are the same: the departments of school. So, we link them to a same entity Department.
* Invoice is not directly linked to Room or Student. Instead, we link Invoice to Lease. Invoice will get Room and Student information from Lease; and get Property information from Room.

Business Rules:

1. Advisor & Department: Each Advisor belongs to only one Department; Each Department has many Advisors.
2. Advisor & Student: Each Advisor advises many Students; each Student has only one Advisor.
3. Communication & Communication Type: Each Communication Type belongs to many Communications; each Communication has only one Communication Type.
4. Communication & Property: Each Communication belongs to many Properties; each Property has many Communications.
5. Communication & Next of Kin: Each Communication belongs to many Properties; each Next of Kin has many Communications.
6. Course & Department: Each Course belongs to only one Department; each Department has many Courses.
7. Inspection & Property: Each Property has many Inspection; each Inspection belongs to only one Property.
8. Inspection & Staff: Each Staff hold many Inspections; each Inspection is held by only one Staff.
9. Invoice & Lease: Each Invoice belongs to only one Lease; each Lease generates many Invoice.
10. Invoice & Payment: Each Invoice is paid by many Payments; each Payment pay to only one Invoice.
11. Lease & Student: Each Lease is signed by only one Student; each Student can sign many Leases.
12. Lease & Room: Each Lease is allocated to one Room; each Room can be allocated to many Leases.
13. Next of Kin & Student: Student & Next of Kin: Each Student has many Next of Kin; each Next of Kin belongs to only one Student.
14. Payment & Payment Type: Each Payment Type is held by many Payments; each Payment has only one Payment Type.
15. Property & Manager: Each Manager manages many Properties; each Property has only one Manager.
16. Property & Type: Each Type is held by many Properties; each Property has only one Type.
17. Property & Room: Each Property has many Rooms; each Room belongs to only one Property.
18. Furnished Status & Room: Each Furnished Status is held by many Rooms; each Room has only one Furnished Status.
19. Student & Room: Each Student can have many Rooms; each Room belongs to only one Student.
20. Student & Degree Course: Each Degree Course is studied by many Students; each Student studies only one Degree course.

Normalize process:

Content of tables:

