Task1

It is not always possible to achieve BOTH BCNF and dependency preservation (DP)

Advisor(s_id, i_id, department)

i_id → department

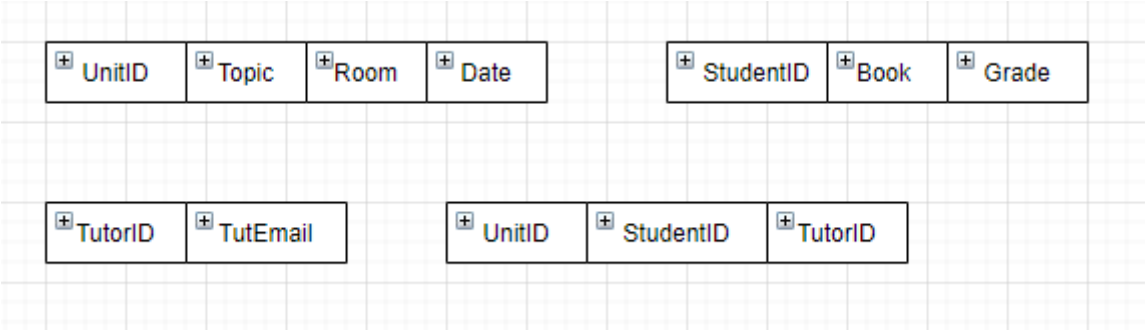
s_id, department → i_id

i_id is not a superkey and any decomposition of Advisor will not include all the attributes in s_id, department → i_id, thus Advisor not in BCNF, and it will NOT be DP.

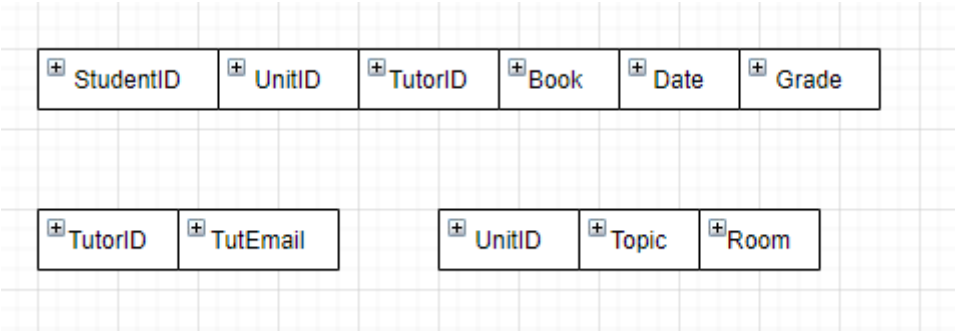
Functional depencies are important in database design because they allow us to eliminate redundancies.

Task2

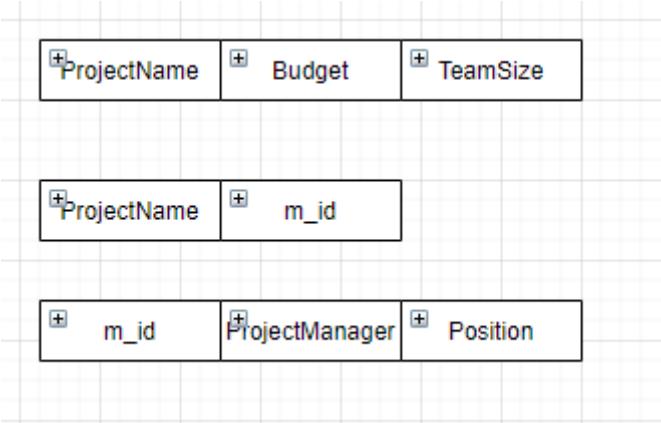
Version #1:



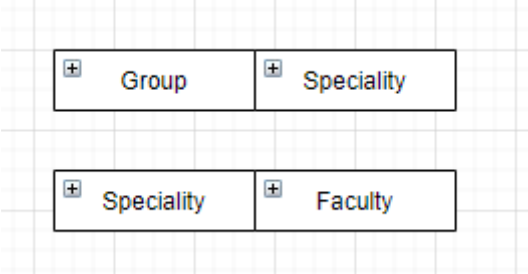
Version #2:



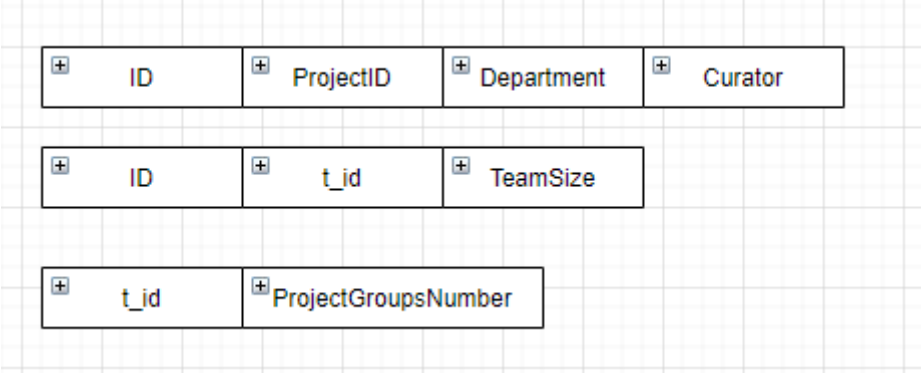
Task3



Task4



Task5



Task6

1. Lossless-join decomposition
2. Dependency preserving decompositions
3. Minimization pf repetition of the information

We can keep an precise, correct database, which is easy to maintain, navigate. Use small amount of space.

Undesirable:

Let us decompose **employee(ID, name, street, city, salary)**

Into **e1(ID, name)** and **e2(name, street, city, salary).** If there is repetition of name then we lose information, hence this is lossy decomposition.

Desirable:

R = (A, B, C) into **r1 = (A, B)** and **r2 = (B, C)**, then we don't lose information, thus it is lossless decomposition.