Task 1

BCNF and dependency persistence It is not always possible to achieve both BCNF and dependency persistence.

Consider the circuit:

dept_advisor (s_ID, i_ID, department_name)

With function dependencies: i_ID -> dept_name s_ID, dept_name -> i_ID

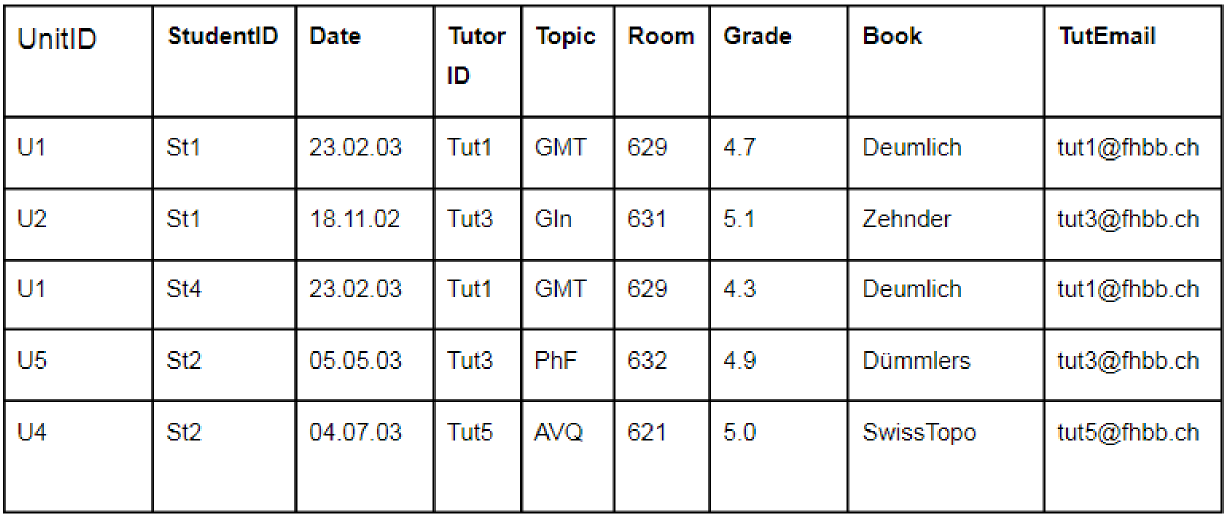
dept_advisor is missing from BCNF i_ID is not a superkey.

A Any dept_advisor decomposition will not include all attributes in

s_ID, dept_name -> i_ID

So composition does NOT preserve dependencies.

Task 2



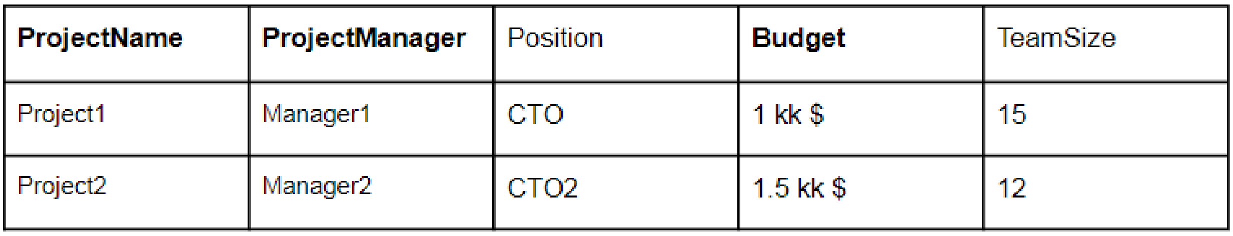
UnitID	StudentID	Date	Tutor ID	Topic	Room	Grade	Book	TutEmail
U1	St1	23.02.03	Tut1	GMT	629	4.7	Deumlich	tut1@fhbb.ch
U2	St1	18.11.02	Tut3	Gln	631	5.1	Zehnder	tut3@fhbb.ch
U1	St4	23.02.03	Tut1	GMT	629	4.3	Deumlich	tut1@fhbb.ch
U5	St2	05.05.03	Tut3	PhF	632	4.9	Dümmlers	tut3@fhbb.ch
U4	St2	04.07.03	Tut5	AVQ	621	5.0	SwissTopo	tut5@fhbb.ch

UnituID	Date	Topic	Room	Book
U1	23.02.03	GMT	629	Deumlich
U2	18.11.02	Gln	631	Zehnder
U5	05.05.03	PhF	632	Dummlers
U4	04.07.03	AVQ	621	SwissTopo

UnitID	StudentID	TutorID	Grade
U1	St1	Tut1	4.7
U2	St1	Tut3	5.1
U1	St4	Tut1	4.3
U5	St2	Tut3	4.9
U4	St2	Tut5	5.0

TutorID	TutEmail
Tut1	Tut1@fhbb.ch
Tut3	Tut3@fhbb.ch
Tut5	Tut5@fhbb.ch

Task 3



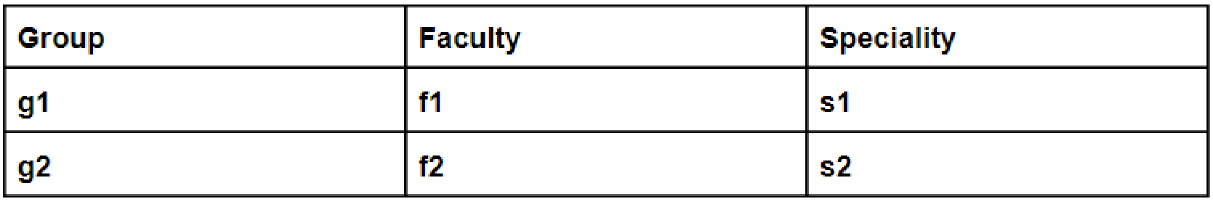
ProjectName	**ProjectManager**	Position	**Budget**	TeamSize
Project1	Manager1	CTO	1 kk $	15
Project2	Manager2	CTO2	1.5 kk $	12

ProjectManager	Position
Manager1	CTO
Manager2	CTO2

ProjectName	Budget	TeamSize
Project1	1 kk $	15
Project2	1.5 kk $	12

ProjectName	ProjectManager
Project1	Manager1
Project2	Manager2

Task 4

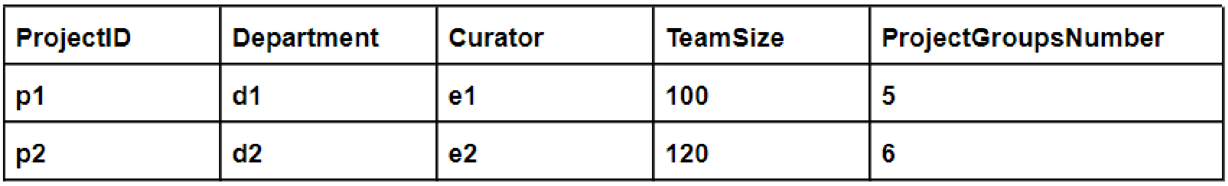


Group	**Faculty**	**Speciality**
g1	**f1**	**s1**
g2	**f2**	**s2**

Group	Specialty
g1	s1
g2	s2

Specialty	Faculry
s1	f1
s2	f2

Task 5



ProjectID	Department	Curator	TeamSize	ProjectGroupsNumber
p1	d1	e1	100	5
p2	d2	e2	120	6

ProjectID	Curator	Department
p1	e1	d1
p2	e2	d2

ProjectID	Department	ProjectGroupsNumber	TeamSize
p1	d1	5	100
p2	d2	6	120

Task 6

List of design goals: Lossless decomposition

Decomposition with preservation of dependencies

BCNF

Insertion Anomaly Suppose for a new admission, until the student selects the branch, the student data cannot be inserted, otherwise we would have to set the branch information to NULL. In addition, if we need to insert data on 100 students of the same department, then the department information will be repeated for all these 100 students. These scripts are nothing more than insertion anomalies.

Update Anomaly What if Mr. X leaves college? or is no longer the HOD of the Faculty of Computer Science? In this case, all student records must be updated, and if we miss a record by mistake, it will lead to data inconsistencies. This is an update anomaly.

Delete Anomaly Our "Students" table stores two different information: student information and branch information. Therefore, at the end of the school year, if student records are deleted, we will also lose branch information. This is a deletion anomaly.