Task 1. Will the conversion to BCNF be dependency preserving in any case? Proofyour statement and give a reasoning for choosing BCNF design.

Proof:

We only need to give a counter example:

Consider the following schema:

> a b c and c->b Clearly the above schema is in 3NF, because ab->c is a superkey dependency and from c->b we can see that b-c=b, which is a subset of the primary key (such dependency is also allowed in 3NF).

But the above schema is not in BCNF because c->b is neither super-key nor trivial dependency. So we decompose above schema, keeping it lossless. Only possible lossless decomposition is: ac and cb. (because their intersection c is primary key for the 2nd table). But clearly the dependency ab->c is lost.

Hence, proved.

Q.E.D

Task 2. Given table in 1NF, convert to 3NF if PK is {UnitID, StudentID}:

StudentID	UnitID	Grade
St1	U1	4.7
St1	U2	5.1
St4	U1	4.3
St2	U5	4.9
St2	U4	5.0

UnitID	TutorID	TutEmail	Room	Date
U1	Tut1	tut1@fhbb.ch	629	23.02.03
U2	Tut3	tut3@fhbb.ch	631	18.11.02
U5	Tut3	tut3@fhbb.ch	632	05.05.03
U4	Tut5	Tut5@fhbb.ch	621	04.07.03

UnitID	Book	Topic
U1	Deumlich	GMT
U2	Zehnder	Gln
U1	Deumlich	GMT
U5	Dummlers	PhF
U4	Swiss Topo	AVQ

Task 3. Given table in 1NF, convert to 2NF if PK is {ProjectName, ProjectManager}, use decomposition:

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

ProjectManager	Position
Manager1	CTO
Manager2	CTO2

Task 4. Given table, convert to 3NF if PK is Group, use decomposition:
Faculties have a number of specialties; each specialty consists of a set of particular groups.

Specialty	**Group**
s1	g1
s2	g2

Faculty	**Specialty**
f1	s1
f2	s2

Task 5. Given table, convert to BCNF if PK is {ProjectID, Department}, use decomposition:

Curator depends on ProjectID and related departments, TeamSize directly relates to project and related departments, ProjectGroupsNumber depends on TeamSize.

ProjectID	**TeamID**	**Department**	**Curator**
p1	1	d1	e1
p2	2	d2	e2

TeamID	**TeamSize**	**ProjectGroupsNumber**
1	100	5
2	120	6

Task 6. List the three design goals for relational databases and explain why each is desirable. Give an example of both desirable and undesirable types of decompositions.

Three design goals for relational databases:

1. Lossless Decomposition (no information is lost)
2. Dependency Preservation (to check the updates without computing the natural join)
3. Lack of Data Redundancy (to use as little space as possible)

Lossless Decomposition (desirable decomposition)

Decomposition is lossless if it is feasible to reconstruct relation R from decomposed tables using Joins. This is the preferred choice. The information will not lose from the relation when decomposed. The join would result in the same original relation.

Example:

EmpInfo (Emp_ID, Emp_Name, Emp_Age, Emp_Location, Dept_ID, Dept_Name)
Decompose the above table into two tables:
EmpDetails (Emp_ID, Emp_Name, Emp_Age, Emp_Location)
DeptDetails (Dept_ID, Emp_ID, Dept_Name)
Now, if we join above two tables we receive the initial table EmpInfo. Therefore, the above relation had lossless decomposition i.e. no loss of information.

Lossy Decomposition (undesirable decomposition)

As the name suggests, when a relation is decomposed into two or more relational schemas, the loss of information is unavoidable when the original relation is retrieved.

Example:

EmpInfo (Emp_ID, Emp_Name, Emp_Age, Emp_Location, Dept_ID, Dept_Name)
Decompose the above table into two tables:

EmpDetails (Emp_ID, Emp_Name, Emp_Age, Emp_Location)
DeptDetails (Dept_ID, Dept_Name)

Now, you won't be able to join the above tables, since **Emp_ID** isn't part of the **DeptDetails** relation.

Therefore, the above relation has lossy decomposition