1NF:
-Each column should contain atomic values
-A column should contain values that are of the same type
-Each column should have a unique name
-Order in which data is saved doesn't matter

2NF:
-It should be in 1NF
-It shouldn't have any Partial Dependency (part of primary key->non-prime attribute)

3NF:
-It should be in 2NF
-It shouldn't have Transitive Dependency (non-prime attribute->non-prime attribute)

BCNF:
-It should be in 3NF
-For every dependency A->B, A should be a super key (✕ non-prime attribute->prime attribute)

Task-1.

Will the conversion to BCNF be dependency preserving in any case? Proof your statement and give reasoning for choosing BCNF design.

It is not always possible to achieve both BCNF and dependency preservation
Consider a schema:
dept_advisor(s_ID, i_ID, department_name)
With function dependencies:
i_ID→dept_name
s_ID, dept_name→i_ID
dept_advisor is not in BCNF
i_ID is not a superkey.
Any decomposition of dept_advisor will not include all the attributes in
s_ID, dept_name→i_ID
Thus, the composition is NOT be dependency preserving

Student	Dept_name	Subject
Amina	FIT	PP1
Dayana	SMC	Calculus1
Alina	BF	Business Negotiations

1NF – satisfied
2NF – satisfied ({Student, Dept_name} -> Subject) no partial dependency

3NF – satisfied (there is only one non-prime attribute)

BCNF – not satisfied ({Subject} -> Dept_name)

To satisfy all criteria for BCNF, we can decompose it like this:

Student	Subject
Amina	PP1
Dayana	Calculus1
Alina	Business Negotiations

Subject	Dept_name
PP1	FIT
Calculus1	SMC
Business Negotiations	BF

But here we can see that dependency {Student, Dept_name} -> Subject is lost/

Hence it is proved.

Task-2.

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

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	GIn	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

Answer:

UnitID	Date	TutorID	Room	Topic
U1	23.02.03	Tut1	629	GMT
U2	18.11.02	Tut3	631	Gin
U4	04.07.03	Tut5	621	AVG
U5	05.05.03	Tut3	632	PhF

Topic	Book
GMT	Deumlich
Gin	Zehnder
Phf	Dummlers
AVG	SwissTopo

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

TutorID	TutEmail
Tut1	tut1@fhbb.ch
Tut3	tut3@fhbb.ch
Tut5	tut5@fhbb.ch

Task-3.

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

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

Answer:

If budget depends on Manager:

ProjectManager	Position
Manager1	CTO
Manager2	CTO2

ProjectName	Budget	TeamSize	ProjectManager
Project1	1 kk $	15	Manager1
Project2	1.5 kk $	12	Manager2

If budget doesn't depend on Manager:

ProjectManager	Position
Manager1	CTO
Manager2	CTO2

ProjectName	ProjectManager
Project1	Manager1
Project2	Manager2

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

Task-4.

Given table, convert to 3NF if PK is Group, use decomposition:

Faculties have a number of specialities, each speciality consists of a set of particular groups.

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

Answer:

Group	Specialty
g1	s1
g2	s2

Specialty	Faculty
s1	f1
s2	f2

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

Answer:

ProjectID	Department
p1	d1
p2	d2

ProjectID	Curator	TeamSize
p1	e1	100
p2	e2	120

TeamSize	ProjectGroupsNumber
100	5
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.

Answer:

1. Minimalization of repetition of information. Repetition of information causes usage of more space for our DB, also it causes data redundancy, which has issues like insertion, deletion and updating anomalies.
2. Lossless join decomposition. In order to answer all questions and queries, we must ensure that after joining the relations we will not lose any necessary data.
3. Dependency Preservation. For using as less space as possible and maintain most accurate relations in our database.

Examples of desirable and undesirable types of decompositions:

1. Lossy decomposition –

<EmpInfo>

Emp_ID	Emp_Name	Emp_Age	Emp_Location	Dept_ID	Dept_Name
E001	Jacob	29	Alabama	Dpt1	Operations
E002	Henry	32	Alabama	Dpt2	HR
E003	Tom	22	Texas	Dpt3	Finance

Decompose the above table into two tables –

<EmpDetails>

Emp_ID	Emp_Name	Emp_Age	Emp_Location
E001	Jacob	29	Alabama
E002	Henry	32	Alabama
E003	Tom	22	Texas

<DeptDetails>

Dept_ID	Dept_Name
Dpt1	Operations
Dpt2	HR
Dpt3	Finance

2. Lossless decomposition –

R:

A	B	C
1	1	1
2	1	2
3	2	1
4	3	2

R_1:

A	B
1	1
2	1
3	2
4	3

R_2:

A	C
1	1
2	2
3	1
4	2

R_1 JOIN R_2:

A	B	C
1	1	1
2	1	2
3	2	1
4	3	2