**Task1.** Will the conversion to BCNF be dependency preserving in any case? Proof your statement and give reasoning for choosing BCNF design.

We can prove it by giving 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 .

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.

So, it BCNF can not be dependency preserving in all cases.

**Task2.** Given table in 1NF, convert to 3NF if PK is {UnitID,StudentID}:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UnitID | StudentID | Date | TutorID | Topic | Room | Grade | Book | TutEmail |
| U1 | St1 | 23.02.03 | Tut1 | GMT | 629 | 4.7 | Deumlich | [tut1@fhbb.ch](mailto:tut1@fhbb.ch) |
| U2 | St1 | 18.11.02 | Tut3 | Gin | 631 | 5.1 | Zehnder | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| U1 | St4 | 23.02.03 | Tut1 | GMT | 629 | 4.3 | Deumlich | [tut1@fhbb.ch](mailto:tut1@fhbb.ch) |
| U5 | St2 | 05.05.03 | Tut3 | PhF | 632 | 4.9 | Dummlers | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| U4 | St2 | 04.07.03 | Tut5 | AVG | 621 | 5.0 | SwissTopo | [tut5@fhbb.ch](mailto:tut5@fhbb.ch) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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](mailto:tut1@fhbb.ch) |
| Tut3 | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| Tut5 | tut5@fhbb.ch |

**Task3.** Given table in 1NF, convert to 2NF if PK is {ProjectName, ProjectManager}, use decomposition:

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

|  |  |
| --- | --- |
| ProjectManager | Position |
| Manager1 | CTO |
| Manager2 | CTO2 |

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

|  |  |
| --- | --- |
| ProjectName | ProjectManager |
| Project1 | Manager1 |
| Project2 | Manager2 |

**Task4.**  Given table, convert to 3NF if PK is Group, use decomposition:

|  |  |  |
| --- | --- | --- |
| Group | Faculty | Speciality |
| G1 | F1 | S1 |
| G2 | F2 | S2 |

|  |  |
| --- | --- |
| Faculty | Specialty |
| F1 | S1 |
| F2 | S2 |

|  |  |
| --- | --- |
| Speciality | Group |
| S1 | G1 |
| S2 | G2 |

**Task5.** Given table convert to BCNF if PK is {ProjectID, Department}, use decomposition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ProjectID | Department | Curator | TeamSize | ProjectGroupsNumber |
| P1 | D1 | E1 | 100 | 5 |
| P2 | D2 | E2 | 120 | 6 |

|  |  |
| --- | --- |
| ProjectID | Curator |
| P1 | E1 |
| P2 | E2 |

|  |  |
| --- | --- |
| Curator | Department |
| E1 | D1 |
| E2 | D2 |

|  |  |
| --- | --- |
| ProjectID | TeamSize |
| P1 | 100 |
| P2 | 120 |

|  |  |
| --- | --- |
| TeamSize | ProjectGroupsNumber |
| 100 | 5 |
| 120 | 6 |

**Task6.** 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.

1)Minimization of information repetition.

When we perform update in such relation, it will be checked easily whether it suits to our relation.  
2)Dependency preserving decomposition.

By this way we can maintain an accurate relations in our database.  
3)Lossless join decomposition

The smallest possible amount of space is used for storing the information.

Types of decomposition:

1)A Lossy Decomposition  
In this type of decomposition we may lose some information from initial table.  
2)Lossless Decomposition  
In this type of decomposition there is no loss of information when we replace relation R by two relations R1 and R2.