Task 1. Will the conversion to BCNF be dependency preserving in any case? Proof your statement and give a 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) instructor can be associated with only a single department, and a student may have more than one advisor, but no more than one from a given department.

The additional constraint that “an instructor can act as advisor for only a single department”.

With function dependencies: i\_ID → dept\_name s\_ID, dept\_name → i\_ID The first functional dependency follows from our requirement that “an instructor can act as an advisor for only one department.” The second functional dependency follows from our requirement that “a student may have at most one advisor for a given department.”

* 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

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](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 | Dümmlers | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| U4 | St2 | 04.07.03 | Tut5 | AVQ | 621 | 5.0 | SwissTopo | [tut5@fhbb.ch](mailto:tut5@fhbb.ch) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UnitID | StudentID | Dr\_Id | Grade | Teaches\_Id |
| U1 | St1 | Dr1 | 4.7 | T\_1 |
| U2 | St1 | Dr2 | 5.1 | T\_2 |
| U1 | St4 | Dr1 | 4.3 | T\_1 |
| U5 | St2 | Dr3 | 4.9 | T\_3 |
| U4 | St2 | Dr4 | 5.0 | T\_4 |

|  |  |  |
| --- | --- | --- |
| Date\_Room\_Id | Date | Room |
| Dr1 | 23.02.03 | 629 |
| Dr2 | 18.11.02 | 631 |
| Dr3 | 05.05.03 | 632 |
| Dr4 | 04.07.03 | 621 |

|  |  |  |
| --- | --- | --- |
| Teaches\_Id | SubjectID | TutorId |
| T\_1 | Subj1 | Tut1 |
| T\_2 | Subj2 | Tut3 |
| T\_3 | Subj3 | Tut3 |
| T\_4 | Subj4 | Tut5 |

|  |  |  |
| --- | --- | --- |
| SubjectID | Topic | Book |
| Subj1 | GMT | Deumlich |
| Subj2 | Gln | Zehnder |
| Subj3 | PhF | Dümmlers |
| Subj4 | AVQ | SwissTopo |

|  |  |
| --- | --- |
| **TutorID** | **TutEmail** |
| Tut1 | [tut1@fhbb.ch](mailto:tut1@fhbb.ch) |
| Tut3 | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| Tut5 | [tut5@fhbb.ch](mailto: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 |

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

|  |  |  |
| --- | --- | --- |
| **ProjectManager** | Position | Team size |
| Project1 | CTO | 15 |
| Project2 | CTO2 | 12 |

|  |  |
| --- | --- |
| **ProjectName** | **Budget** |
| Project1 | 1 kk $ |
| Project2 | 1.5 kk $ |

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. Faculty -> Speciality -> Group***

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

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

|  |  |
| --- | --- |
| Speciality | 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** |

|  |  |  |
| --- | --- | --- |
| **ProjectID** | **Curator** | **TeamID** |
| **p1** | **e1** | **T\_1** |
| **p2** | **e2** | **T\_2** |

|  |  |  |
| --- | --- | --- |
| **TeamID** | **TeamSize** | **ProjectGroupsNumber** |
| T\_1 | 100 | 5 |
| T\_2 | 120 | 6 |

|  |  |
| --- | --- |
| **Curator\_id** | **Department** |
| **e1** | **d1** |
| **e2** | **d2** |

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.

1. Dependency preserving decomposition.

This permits the validity of an update to be tested without the need to compute a join of relations in the decomposition.

1. Lossless join decomposition

By this way we can maintain an accurate relations in our database.

1. Minimization of information repetition

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

Desirable type: **Lossless Decomposition**

By lossless decomposition it becomes feasible to reconstruct the relation R from decomposed tables R1 and R2 by using Joins.

Undesirable type: **Lossy Decomposition**

We cannot reconstruct the original relation