**Task 1.**

Consider the schema: a b c and c->b that is in 3NF, because ab->c is a super key 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. Proved.

**Task 2.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Unit ID** | **Student ID** | **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 |

|  |  |  |
| --- | --- | --- |
| **Unit ID** | **Student ID** | **Grade** |
| U1 | St1 | 4.7 |
| U2 | St1 | 5.1 |
| U1 | St4 | 4.3 |
| U5 | St2 | 4.9 |
| U4 | St2 | 5.0 |

|  |  |
| --- | --- |
| **Tutor ID** | **TutEmail** |
| Tut1 | tut1@fhbb.ch |
| Tut3 | tut3@fhbb.ch |
| Tut5 | tut5@fhbb.ch |

|  |  |
| --- | --- |
| **Book** | **Topic** |
| Deumlich | GMT |
| Zehnder | GIn |
| Dümmlers | PhF |
| SwissTopo | AVQ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit ID** | **Date** | **Tutor ID** | **Room** |
| U1 | 23.02.03 | Tut1 | 629 |
| U2 | 18.11.02 | Tut3 | 631 |
| U5 | 05.05.03 | Tut3 | 632 |
| U4 | 04.07.03 | Tut5 | 621 |

|  |  |
| --- | --- |
| **Unit ID** | **Topic** |
| U1 | GMT |
| U2 | GIn |
| U5 | PhF |
| U4 | AVQ |

**Task 3**

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

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

|  |  |  |
| --- | --- | --- |
| **ProjectName** | **ProjectManager** | **Position** |
| Project1 | Manager1 | CTO |
| Project2 | Manager2 | CTO2 |

**Task 4**

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

|  |  |  |
| --- | --- | --- |
| **Group** | **Faculty** | **Specialty** |
| G1 | F1 | S1 |
| G2 | F2 | S2 |

|  |  |
| --- | --- |
| **Specialty** | **Group** |
| S1 | G1 |
| S2 | G2 |

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

**Task 5**

Curator depends on projectID and related departments, teamSize directly relates to project and related departments, ProjectGroupsNumber depends on TeamSize

|  |  |  |  |
| --- | --- | --- | --- |
| **ProjectID** | **Department** | **TeamID** | **Curator** |
| P1 | D1 | T1 | E1 |
| P2 | D2 | T2 | E2 |

|  |  |  |
| --- | --- | --- |
| **TeamID** | **TeamSize** | **ProjectGroupsNumber** |
| T1 | 100 | 5 |
| T2 | 120 | 6 |

**Task 6**

 The three design goals are lossless-join decompositions, dependency preserving decompositions, and minimization of repetition of information. They are desirable so we can maintain an accurate database, check correctness of updates quickly, and use the smallest amount of space 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.