# Task1.

In BCNF, every non-prime attribute should be functionally dependent on any of super key in schema. If there exists any FD, which do not follow this, then for that case we have to separate it into new relation. Now if any of other FD uses previous FD, then this creates non presentation of FD in BNCF.

# Task2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UnitID** | TutorID | Topic | Room | Date |
| U1 | Tut1 | GMT | 629 | 23.02.03 |
| U2 | Tut3 | Gln | 631 | 18.11.02 |
| U1 | Tut1 | GMT | 629 | 23.02.03 |
| U5 | Tut3 | PhF | 632 | 05.05.03 |
| U4 | Tut5 | AVQ | 621 | 04.07.03 |

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

|  |  |
| --- | --- |
| **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) |

|  |  |
| --- | --- |
| **Topic** | Book |
| GMT | Deumlich |
| Gln | Zehnder |
| PhF | Dummlers |
| AVQ | SwissTopo |

**Task3.**

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

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

|  |  |
| --- | --- |
| **Group** | Specialty |
| g1 | s1 |
| g2 | s2 |

|  |  |
| --- | --- |
| **Specialty** | Faculty |
| s1 | f1 |
| s2 | f2 |

# Task5.

|  |  |
| --- | --- |
| **ProjectID** | **Department** |
| p1 | d1 |
| p2 | d2 |

|  |  |  |
| --- | --- | --- |
| **ProjectID** | Curator | TeamSize |
| p1 | e1 | 100 |
| p2 | e2 | 120 |

|  |  |
| --- | --- |
| **TeamSize** | ProjectGrouosNumber |
| 100 | 5 |
| 120 | 6 |

**Task6.**

# Lossless Join Decomposition

-If the information is not lost from the relation that is decomposed, then the decomposition will be lossless.

-The lossless decomposition guarantees that the join of relations will result in the same relation as it was decomposed.

-The relation is said to be lossless decomposition if natural joins of all the decomposition give the original relation.

# Dependency Preserving Decomposition

-It is an important constraint of the database.

-In the dependency preservation, at least one decomposed table must satisfy every dependency.

-If a relation R is decomposed into relation R1 and R2, then the dependencies of R either must be a part of R1 or R2 or must be derivable from the combination of functional dependencies of R1 and R2.

**Repetition of information** – condition in database, that the values of one attribute are

determined by the values of another attribute in the same relation, and both values are repeated throughout the relation.