*TASK 1:*

It is not always possible to achieve both BCNF and dependency preservation

Proof with schema

- R=(A, B, C)

With function dependencies:

- B → C

- A, C →B

R is not in BCNF

- B is not a superkey.

Any decomposition of R will not include all the attributes in

- A, C → B

Thus, the composition is not be dependency preserving

advantages: we may have to use null values to represent some of the possible meaningful

relationships among data items. There is the problem of repetition of information.

*TASK 2:*

|  |  |  |  |
| --- | --- | --- | --- |
| StudentID | UnitID | Grade | TutorID |
| St1 | U1 | 4,7 | Tut1 |
| St1 | U2 | 5,1 | Tut3 |
| St2 | U4 | 5 | Tut5 |
| St2 | U5 | 4,9 | Tut3 |
| St4 | U1 | 4,3 | Tut1 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Date | Room | Topic | Book |
| U1 | 23.02.03 | 629 | GMT | Deumlich |
| U2 | 18.11.02 | 631 | Gln | Zehnder |
| U4 | 04.07.03 | 621 | AVQ | SwissTopo |
| U5 | 05.05.03 | 632 | PhF | Dummlers |

*TASK 3:*

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

|  |  |  |
| --- | --- | --- |
| ProjectName | Budget |  |
| ProjectName | Budget | TeamSize |
| Project1 | 1 kk $ | 15 |
| Project2 | 1,5 kk $ | 12 |

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

*TASK 4:*

|  |  |
| --- | --- |
| Faculty | Speciality |
| F1 | S1 |
| F2 | S2 |

|  |  |
| --- | --- |
| Speciality | Groups |
| S1 | G1 |
| S2 | G2 |

*TASK 5:*

|  |  |
| --- | --- |
| ProjectID | Department |
| P1 | D1 |
| P2 | D2 |

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

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

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

*TASK 6:*

1. Minimization of information repetition.

When we perform update in such relation, it will be checked easily whether it suits to our relation.

1. Dependency preserving decomposition.

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

1. 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.

1. Lossless Decomposition

In this type of decomposition there is no loss of information when we replace relation R by two relations R1 and R2.