Task1

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

Consider a schema:

dept\_advisor(s\_ID, i\_ID, department\_name)

With function dependencies: i\_ID -> dept\_name s\_ID, dept\_name -> i\_ID

dept\_advisor is not in BCNF i\_ID is not a superkey.

A 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

-----------------------------------------------------------------------------------

Task2

Unit\_table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UintID | StudentID | TutorID | Room | Date |
| U1 | St1 | Tut1 | 629 | 23.02.03 |
| U2 | St1 | Tut3 | 631 | 18.11.02 |
| U1 | St4 | Tut1 | 629 | 23.02.03 |
| U5 | St2 | Tut3 | 632 | 05.05.03 |
| U4 | St2 | Tut5 | 621 | 04.07.03 |

Tutor\_Table

|  |  |  |  |
| --- | --- | --- | --- |
| TutorID | Topic | Book | TutEmail |
| Tut1 | GMT | Deumlich | [tut1@fhbb.ch](mailto:tut1@fhbb.ch) |
| Tut3 | Gin | Zehnder | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| Tut1 | GMT | Deumlich | [tut1@fhbb.ch](mailto:tut1@fhbb.ch) |
| Tut3 | PhF | Dümmlers | [tut3@fhbb.ch](mailto:tut3@fhbb.ch) |
| Tut5 | AVQ | SwissTopo | [tut5@fhbb.ch](mailto:tut5@fhbb.ch) |

|  |  |
| --- | --- |
| StudentID | Grade |
| St1 | 4.7 |
| St1 | 5.1 |
| St4 | 4.3 |
| St2 | 4.9 |
| St2 | 5.0 |

3 Normal form

|  |  |
| --- | --- |
| StudentID | Date |
| St1 | 23.02.03 |
| St1 | 18.11.02 |
| St4 | 23.02.03 |
| St2 | 05.05.03 |
| St2 | 04.07.03 |

|  |  |
| --- | --- |
| Room | Date |
| 629 | 23.02.03 |
| 632 | 18.11.02 |
| 629 | 23.02.03 |
| 632 | 05.05.03 |
| 621 | 04.07.03 |

Task3

|  |  |  |
| --- | --- | --- |
| Project  Name | Budget | Team size |
| Project1 | 1 kk $ | 15 |
| Project2 | 1.5 kk$ | 12 |

|  |  |
| --- | --- |
| Project Manager | Position |
| Manager1 | CTO |
| Manager2 | CTO2 |

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

--------------------------------------------------------------------------------------------------------------------------------

Task4

|  |  |
| --- | --- |
| Group | Faculty |
| g1 | f1 |
| g2 | f2 |

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

Task5

|  |  |  |  |
| --- | --- | --- | --- |
| ProjectID | Department | ProjectGroupNumber | TeamsSize |
| P1 | D1 | 5 | 100 |
| P2 | D2 | 6 | 120 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ProjectID | Department | ProjectGroupNumber | Curator |  |  |
| P1 | D1 | 5 | E1 |  |  |
| P2 | D2 | 6 | E2 |  |  |

Task6

List of design goals:

\*Lossless-join decomposition

\*Dependency preserving decomposition

\*BCNF

Insertion Anomaly Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as NULL. Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students. These scenarios are nothing but Insertion anomalies.

Updation Anomaly What if Mr. X leaves the college? or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency. This is Updation anomaly.

Deletion Anomaly In our Student table, two different informations are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is Deletion anomaly.