Task 1

BCNF and dependency persistence It is not always possible to achieve both BCNF and dependency persistence.

Consider the circuit:

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

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

dept\_advisor is missing from BCNF i\_ID is not a superkey.

A Any dept\_advisor decomposition will not include all attributes in

s\_ID, dept\_name -> i\_ID

So composition does NOT preserve dependencies.

Task 2

Изображение выглядит как стол

Автоматически созданное описание

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

|  |  |  |  |
| --- | --- | --- | --- |
| UnitID | StudentID | TutorID | Grade |
| U1 | St1 | Tut1 | 4.7 |
| U2 | St1 | Tut3 | 5.1 |
| U1 | St4 | Tut1 | 4.3 |
| U5 | St2 | Tut3 | 4.9 |
| U4 | St2 | Tut5 | 5.0 |

|  |  |
| --- | --- |
| TutorID | TutEmail |
| Tut1 | Tut1@fhbb.ch |
| Tut3 | Tut3@fhbb.ch |
| Tut5 | Tut5@fhbb.ch |

Task 3

Изображение выглядит как стол

Автоматически созданное описание

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

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

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

Task 4

Изображение выглядит как стол

Автоматически созданное описание

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

|  |  |
| --- | --- |
| Specialty | Faculry |
| s1 | f1 |
| s2 | f2 |

Task 5

Изображение выглядит как стол

Автоматически созданное описание

|  |  |  |  |
| --- | --- | --- | --- |
| ProjectID | Department | ProjectGroupsNumber | Curator |
| p1 | d1 | 5 | e1 |
| p2 | d2 | 6 | e2 |

|  |  |  |  |
| --- | --- | --- | --- |
| ProjectID | Department | ProjectGroupsNumber | TeamSize |
| p1 | d1 | 5 | 100 |
| p2 | d2 | 6 | 120 |

Task 6

List of design goals: Lossless decomposition

Decomposition with preservation of dependencies

BCNF

Insertion Anomaly Suppose for a new admission, until the student selects the branch, the student data cannot be inserted, otherwise we would have to set the branch information to NULL. In addition, if we need to insert data on 100 students of the same department, then the department information will be repeated for all these 100 students. These scripts are nothing more than insertion anomalies.

Update Anomaly What if Mr. X leaves college? or is no longer the HOD of the Faculty of Computer Science? In this case, all student records must be updated, and if we miss a record by mistake, it will lead to data inconsistencies. This is an update anomaly.

Delete Anomaly Our "Students" table stores two different information: student information and branch information. Therefore, at the end of the school year, if student records are deleted, we will also lose branch information. This is a deletion anomaly.