**Databases**

**Laboratory work 5**

**Maiwand Alozi**

**Task 1. Will the conversion to BCNF be dependency preserving in any case? Proof your statement and give a reasoning for choosing BCNF design.**

**Answer:** Every conversion into BCNF may not be dependency preserving.

**Proof:** We only need to give a counter example: Consider the following schema;

* a b c and c 🡪 b
* Clearly the above schema 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.

Hence, proved.

**Task 2. Given table in 1NF, convert to 3NF if PK is UnitID and StudentID:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UnitID** | **StudentID** | **Date** | **TutorID** | **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 |

**Solution:**

To convert this table to 3NF, first of all, we should convert it to the 2nd normal form. That is:

1. The table should be in the First Normal Form.
2. There should be no Partial Dependency.

2nd NF of the tables:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **UnitID** | **StudentID** | **Date** | **Topic** | **Room** | **Grade** | **Book** | **TutorID** |
| U1 | St1 | 23.02.03 | GMT | 629 | 4.7 | Deumlich | Tut1 |
| U2 | St1 | 18.11.02 | GIn | 631 | 5.1 | Zehnder | Tut3 |
| U1 | St4 | 23.02.03 | GMT | 629 | 4.3 | Deumlich | Tut1 |
| U5 | St2 | 05.05.03 | PhF | 632 | 4.9 | Dümmlers | Tut3 |
| U4 | St2 | 04.07.03 | AVQ | 621 | 5.0 | SwissTopo | Tut5 |

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

Then, for a table to be in the third normal form:

1. It should be in the Second Normal form.
2. And it should not have Transitive Dependency.

3nd NF of the tables:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **UnitID** | **StudentID** | **Date** | **Room** | **Grade** | **TopicID** | **TutorID** |
| U1 | St1 | 23.02.03 | 629 | 4.7 | Top1 | Tut1 |
| U2 | St1 | 18.11.02 | 631 | 5.1 | Top2 | Tut3 |
| U1 | St4 | 23.02.03 | 629 | 4.3 | Top1 | Tut1 |
| U5 | St2 | 05.05.03 | 632 | 4.9 | Top3 | Tut3 |
| U4 | St2 | 04.07.03 | 621 | 5.0 | Top4 | Tut5 |

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

|  |  |  |
| --- | --- | --- |
| **TopicID** | **Topic** | **Book** |
| Top1 | GMT | Deumlich |
| Top2 | GIn | Zehnder |
| Top3 | PhF | Dümmlers |
| Top4 | AVQ | SwissTopo |

**Task 3. Given table in 1NF, convert to 2NF if PK is {ProjectName, ProjectManager}, use decomposition:**

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

**Solution:**

Here, **Position** depends on **ProjectManager** but not **ProjectName**. So, that is a partial dependency. To reduce it we use a decomposition method like shown below.

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

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

**Task 4. Given table, convert to 3NF if PK is Group, use decomposition:**

*Faculties have a number of specialties; each specialty consists of a set of particular groups.*

|  |  |  |
| --- | --- | --- |
| **Group** | **Faculty** | **Specialty** |
| g1 | f1 | s1 |
| g2 | f2 | s2 |

**Solution:**

|  |  |
| --- | --- |
| **Group** | **S\_F\_ID** |
| g1 | 1 |
| g2 | 2 |

|  |  |  |
| --- | --- | --- |
| **S\_F\_ID** | **Faculty** | **Specialty** |
| 1 | f1 | s1 |
| 2 | f2 | s2 |

**Task 5. Given table, convert to BCNF if PK is {ProjectID, Department}, use decomposition:**

*Curator depends on* ***ProjectID*** *and related departments,* ***TeamSize*** *directly relates to project and related departments,* ***ProjectGroupsNumber*** *depends on* ***TeamSize****.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ProjectID** | **Department** | **Curator** | **TeamSize** | **ProjectGroupsNumber** |
| p1 | d1 | e1 | 100 | 5 |
| p2 | d2 | e2 | 120 | 6 |

**Solution:**

In this task, first of all, we see a transitive dependency, i.e. **ProjectGroupsNumber** depends on **TeamSize**, that depends on {**ProjectID**, **Department**}. From the definition of Boyce-Codd Normal Form (BCNF), we have:

1. It should be in the Third Normal Form.
2. And, for any dependency A → B, A should be a super key.

Hence, before BCNF, we should convert actual table to 3rd NF, because of transitive dependency. The reduced form looks like:

|  |  |  |  |
| --- | --- | --- | --- |
| **ProjectID** | **Department** | **Curator** | **TeamID** |
| p1 | d1 | e1 | t1 |
| p2 | d2 | e2 | t2 |

|  |  |  |
| --- | --- | --- |
| **TeamID** | **TeamSize** | **ProjectGroupsNumber** |
| t1 | 100 | 5 |
| t2 | 120 | 6 |

Then, if we look at the first table, we can see that,

{**ProjectID**, **Department**} 🡪 **Curator**

**Curator** 🡪 **ProjectID**

So, here we have that **Curator** is not a super key, therefore the first table is not in BCNF. To convert it to BCNF we should decompose it:

|  |  |
| --- | --- |
| **Department** | **Curator** |
| d1 | e1 |
| d2 | e2 |

|  |  |  |
| --- | --- | --- |
| **Curator** | **ProjectID** | **TeamID** |
| e1 | p1 | t1 |
| e2 | p2 | t2 |

|  |  |  |
| --- | --- | --- |
| **TeamID** | **TeamSize** | **ProjectGroupsNumber** |
| t1 | 100 | 5 |
| t2 | 120 | 6 |

**Task 6.** **List the three design goals for relational databases, and explain why each is desirable. Give an example of both desirable and undesirable types of decompositions.**

**Answer:** 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. In Figure 1, there is an example for undesirable decomposition, i.e. **lossy decompositions.** We shall refer to such decompositions as being **lossy decompositions**, and, conversely, to those that are not as **lossless decompositions**.

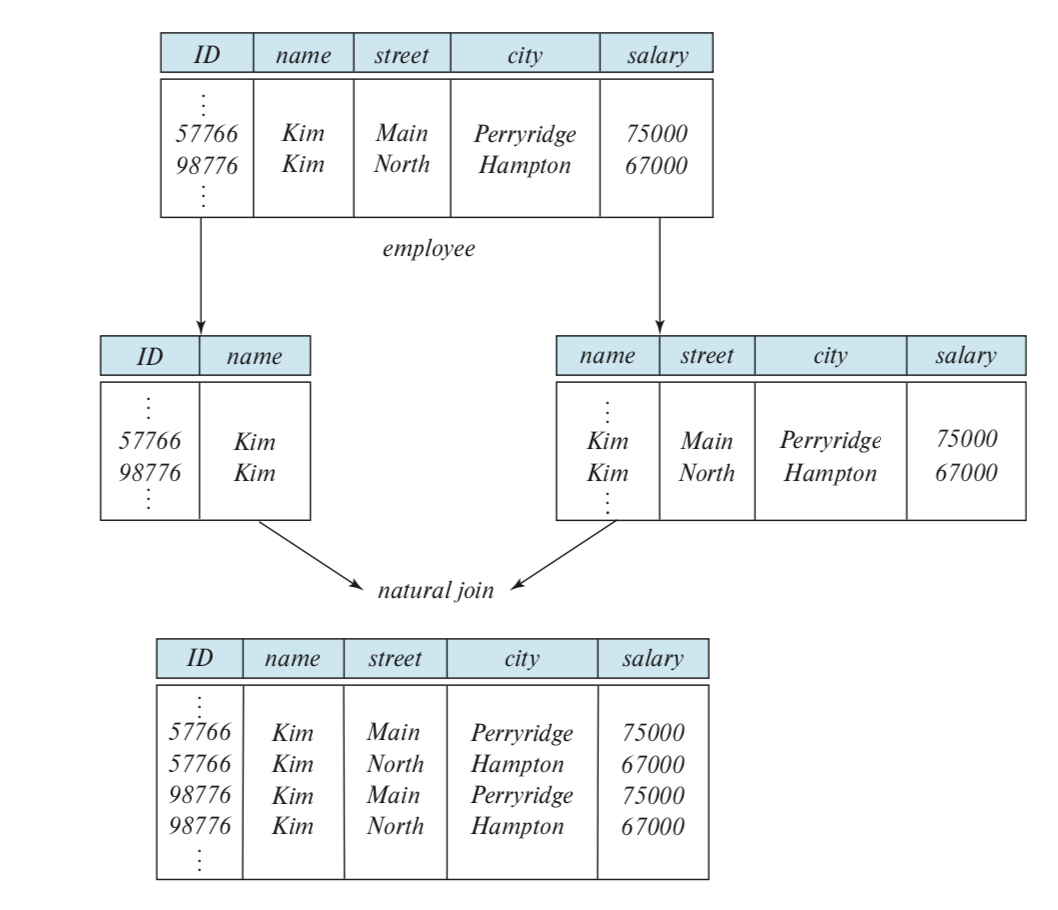
****

Figure 1 - Loss of information via a bad decomposition.

We say that the decomposition is a **lossless decomposition** if there is no loss of information by replacing R with two relation schemas R1 and R2. In Figure 2, we can notice the lossless decomposition.

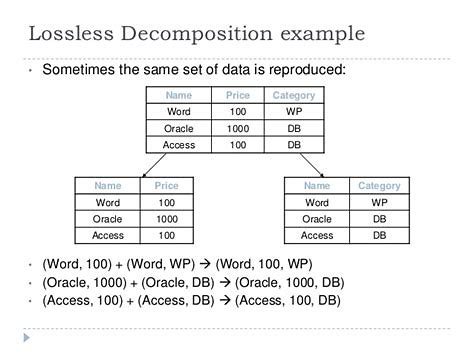


Figure 2 – Example of lossless decomposition