CSCU9B3 Database Principles and Applications Autumn 2018

Tutorial 2: Normalisation

1. Define Functional Dependency, 2nd Normal Form, 3rd Normal Form and Non-Loss Decomposition.

2. This problem comes from Ritchie (3rd Edition, page 130). A Financial consultancy provides consultants to work on clients' projects. Each Consultant works on only one project at a time, but a project may employ more than one consultant. The following (un-normalized) table gives sample data about current assignments.

Consultant ID (C#)	Project no. (P#)	Hours (Hours)	Project Name (Pname)	Consultant Name (Cname)	Project Location (Ploc)	Fee Rate (Fee)
C1	P1	450	Apollo	Gray	Glasgow	100
C2	P2	90	Zeus	Brown	Edinburgh	90
C3	P2	20	Zeus	White	Edinburgh	95
C4	P3	135	Mercury	Green	Aberdeen	150

Ritchie tells us that the Hours value is the total hours worked on the project so far, per consultant. He also tells us that the fee rate value depends on the combination of the consultant involved and the project.

(a) Using this table as an example, explain how anomalies can arise from the use of un-normalised data.

(b) What are the functional dependencies in the above table? What are the candidate keys?

(c) Design a decomposition to produce relations in 3NF identifying the primary keys and foreign keys.

(d) The consultancy is restructured so that consultants are now allowed to work on more than one project at a time. How does this affect the functional dependencies and therefore the decomposition? Show the new decomposition.

3. A relation has four attributes, as in the following table:

Property Number(N)	**Street Address (A)**	**City (C)**	**Postcode (P)**
P105	10 Acacia Avenue	Edinburgh	EH10 1ZX
P972	11 Acacia Avenue	Edinburgh	EH10 1ZX
P655	10 Acacia Avenue	Perth	PH9 1AB
P390	27 Union Street	Stirling	FK9 7BN
P217	14 Drover Lane	Glasgow	G12 7AS

All street addresses are unique within cities. As one would expect, the area corresponding to a postcode falls within only one city (e.g. FK9 7BN is completely contained in Stirling and none of it is in Falkirk).

(a) Identify all functional dependencies among the attributes.

(b) Identify all key attributes (all those which are part of a candidate key).

(c) In which normal form is this relation?

For all parts, be careful to consider and state any assumptions made.

Sample Solutions

Q1. Bookwork:

Functional dependency: a column (or set of columns) Y is functionally dependent on a column (or set of columns) X if at any time exactly one Y value is associated with each X value. (Or, X uniquely identifies Y)

2nd Normal Form: a relation is in 2NF if it is in 1NF and every non-key column is fully functionally dependent on the primary key.

3rd Normal Form: a relation is in 3NF if it is in 2NF and every non-key column is non-transitively fully functionally dependent on the primary key.

Non-Loss decomposition: a relation is replaced by multiple relations representing part of the data from which the original relation can be re-created.

Q2. The question from Ritchie:

(a) There are three types of anomaly: update, delete, insert. Information about projects is repeated (e.g. name, location). An update anomaly occurs when we change the value for the location for a project in one row but not the other (e.g. change the location of Zeus to Falkirk in row 2 but not row 3). A deletion anomaly occurs if the last consultant on a project is reassigned and as a consequence we lose all the project information (e.g. delete row 1 because Gray leaves the company, but now we've also lost information about project Apollo). An insertion anomaly occurs if we want to add a new project, because we can't do this without also assigning a consultant (e.g. new project Athena based in Stirling must also have consultant Black with hours and fees).

(b) Obviously we have:

C# -> Cname (1)

P# -> (Pname, Ploc) (2)

The sentence "The fee rate value depends on the combination of the Consultant involved and the project" suggests:

(C#,P#) -> (Hours, Fee) (3)

but this is misleading because we have:

C# -> P# (4)

because each consultant only works on one project. Then (3) should really be:

C# -> (Hours, Fee) (3a)

The candidate key is C#

(c) Decomposition into 3NF will give (C#, Cname, P#, Hours, Fee) and (P#, Pname, Ploc).

(d) If consultants can work on multiple projects then (4) and (3a) no longer hold, and we go back to (3) above. The Key is now (C#,P#). So our decomposition is: (C#, Cname), (C#, P#, Hours, Fee) and (P#, Pname, Ploc).

Q3.

(a) The full functional dependencies present are:

N →A,C,P (a property number identifies a unique address, city and postcode)

P → C (a postcode identifies a unique city)

A,C →P (street address together with city determine a unique postcode)

It may be the case that we can have different properties at the same address (eg, an apartment building). If this is NOT allowed, then we also have

A,C → N (street address and city determine a unique property number)

A,P → N (street address and postcode determine a unique property number)

(b) Again, this depends on whether we allow different properties at the same address.

If NO, then there are three candidate keys: N and (A,C), and (A,P), so the key attributes are N, A, C and P (ie, all of them).

If YES, then there is only one candidate key, N, and the only key attribute is N.

(c) Again, this depends on whether there can be different properties at the same address:

If NO, then there are no non-key attributes, so it is trivially in 3NF. (We cannot have transitive dependencies involving non-key attributes if there are no non-key attributes!)

If YES, then it is in 2NF because every non-key attribute is fully functionally dependent on the primary key (N), but it is not in 3NF because P (a non-key attribute) is transitively dependent on N (the primary key) via A,C (non-key attributes). In other words, we have this transitive dependency:

N → A,C → P