Normalisation 1 Review

unnormalised => 1NF => 2NF => 3NF => BCNF

Chapter 4.1

V3.01

Dr Gordon Russell

INVESTOR IN PEOPLE

NAPIER UNIVERSITY
EDINBURGH

Terminology

- R (matric_no, firstname, surname, tutor_no, tutor_name)

  tutor_no -> tutor_name

  - A given tutor_no uniquely identifies (AKA *functionally determines*) a tutor_name.
  - Tutor_name is dependent on tutor_no
  - Tutor_no is the *determinant*

  - An implied determinant (underlined) is also present in R:
    - matrix_no -> firstname, surname, tutor_no, tutor_name

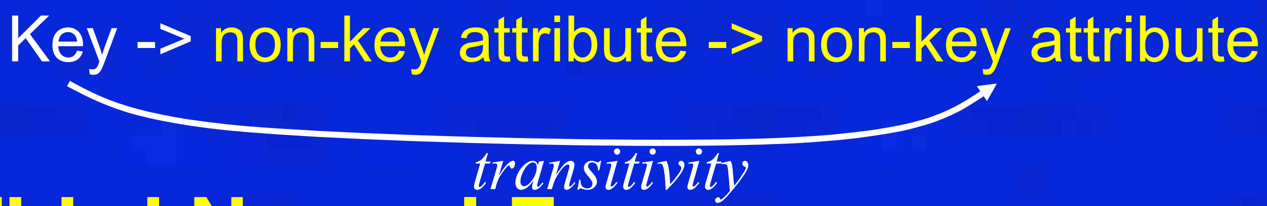
First Normal Form

- A relation is in 1NF if, and only if, it contains no repeating attributes or groups of attributes (must be atomic values).
- A table with repeating groups is not in 1NF
  - it is an `unnormalised table'.
- To remove repeating groups, either:
  - flatten the table and extend the key, or
  - decompose (split) the relation

KeyPart1 + KeyPart2 => attribute1, attribute2, …

Second Normal Form

- A relation is in 2NF if, and only if, it is in 1NF **and every** non-key attribute is fully functionally dependent on the whole key.
- Thus all non-key attributes must depend on the whole key. Another way of saying this is that there must be no partial key dependencies (PKDs).
- Problems arise only when there is a compound key


Key -> non-key attribute -> non-key attribute

transitivity

Third Normal Form

- 3NF removes virtually all the redundant data
- A relation is in 3NF if, and only if,
  - it is in 2NF and
  - there are no transitive functional dependencies
- A transitive functional dependency can only occur if there is more than one non-key field
- A non-key field must provide a fact about the key, the whole key (2NF) and nothing but the key (3NF).

Summary: 1NF

Primary Key for the whole relation

- A relation is in 1NF if it contains no repeating groups
- **Remember** to put the primary key from the original relation into both new relations.

R ($\underline{a}$,b, ($\underline{c}$,d)) becomes

R($\underline{a}$,b)

R1($\underline{a}$,$\underline{c}$,d)

Summary: 2NF

- A relation is in 2NF if it is in 1NF and has no partial key functional dependencies
- NOTE: A relation in 1NF with a single key field **must** (inevitably) be in 2NF
- DECOMPOSE:
  - One relation for the attributes that are fully dependent upon the key.
  - One relation for each part of the key that has partially dependent attributes

  R (a,b,c,d)

  a->c becomes

  R (a,b,d) and R1 (a,c)

Summary: 3NF

- A relation is in 3NF if it it is in 2NF and has no transitive functional dependencies
- NOTE: A relation in 2NF with only one non-key attribute **must** (inevitably) be in 3NF
- DECOMPOSE To remove transitive functional dependencies, remove the attributes involved in the transitive dependency to a new relation

3NF continued

$R(\underline{a,b},c,d)$

c -> d Becomes

$R(\underline{a,b},c)$

$R1(\underline{c},d)$