Normalisation

Database Systems

Michael Pound

SEM Feedback

- Mostly positive feedback
- Some issues:
  - Some examples don't map to real life scenarios
  - Large gap until the 4pm lecture
  - 9am labs are too early!

SEM Feedback

- More detail on SQL insertion/injection attacks
  - I'll be showing you how to do (and defend against) SQL insertion attacks in the security lecture.
- More information on data storage
  - I'll cover data storage in the lecture on efficiency and storage
- Solutions to labs
  - Will appear on the website
- Dim the lights a little!

This Lecture

- Normalisation
  - Data Redundancy
  - Functional Dependencies
  - Normal Forms
  - First, Second and Third Normal Forms
- Further reading
  - The Manga Guide to Databases, Chapter 3
  - Database Systems, Chapter 14

Redundancy and Normalisation

- Redundant data
  - Can be determined from other data in the database
  - Leads to various problems
    - INSERT Anomalies
    - UPDATE Anomalies
    - DELETE Anomalies
- Normalisation
  - Aims to reduce data redundancy
  - Redundancy is expressed in terms of functional dependencies
  - Normal forms are defined that don't contain specific types of functional dependency

Normalisation

- Normalisation is a formal process for something that you will often do naturally
  - When you create your tables, they'll often be normalised already
- E/R diagrams help produce normalised tables
- Despite being somewhat common sense, it's good to have a formal process we can use

First Normal Form

- In most definitions of the relational model
  - All data values should be atomic
  - This means that table entries should be single values, not sets or composite objects
  - Simplifies queries and data comparisons
- A relation is said to be first normal form (1NF) if all data values are atomic

Normalisation to 1NF

- To convert any relation into 1NF, split any non-atomic values

Unnormalised

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D2	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

Problems with 1NF

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

- INSERT Anomalies
  - Can't add a module with no texts
- UPDATE Anomalies
  - To change the lecture for M1, we will need to update two rows
- DELETE Anomalies
  - If we remove M3, we will remove L2 as well

Functional Dependencies

- Redundancy is often caused by a functional dependency
- A functional dependency (FD) is a link between two sets of attributes in a relation
- We can normalise a relation by removing undesirable FDs
- A set of attributes, A, functionally determines another set, B, if whenever two rows of the relation have the same values for all the attributes in A, then they also have the same values for all the attributes in B.
- In this case, we can say there exists a functional dependency between A and B ($A \rightarrow B$),

Example

- Three notable functional dependencies exist in this relation:
  - {ID} → {First, Last}
  - {moduleCode} → {moduleName}
  - {ID, moduleCode} → {First, Last, moduleName}

ID	First	Last	moduleCode	moduleName
111	Joe	Smith	G51PRG	Programming
222	Anne	Jones	G51DBS	Databases

Properties of FDs

- In any relation
  - The primary key functionally determines any set of attributes in that relation

    $K \rightarrow X$
    - K is the primary key, X is a set of attributes
  - Same for candidate keys
  - Any set of attributes is FD on itself

    $X \rightarrow X$
- Rules for FDs
  - Reflexivity: If B is a subset of A then

    $A \rightarrow B$
  - Augmentation: If $A \rightarrow B$ then

    $A \cup C \rightarrow B \cup C$
  - Transitivity: If $A \rightarrow B$ and $B \rightarrow C$ then

    $A \rightarrow C$

FDs and Normalisation

- We define a set of 'normal forms'
  - Each normal form has fewer FDs than the last
  - Since FDs represent redundancy, each normal form has less redundancy than the last
- Not all FDs cause a problem
  - We identify various sorts of FD that do
  - Each normal form removes a type of FD that causes problems

FD Example

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

- The Primary Key is {Module, Text} so
  - {Module, Text} → {Dept, Lecturer}
- 'Trivial' FDs, eg:
  - {Text, Dept} → {Text}
  - {Module} → {Module}
  - {Dept, Lecturer} → { }

FD Example

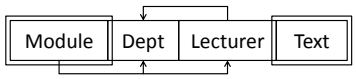
1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

- Other FDs are
  - {Module} → {Lecturer}
  - {Module} → {Dept}
  - {Lecturer} → {Dept}
  - These are non-trivial and the determinants (left hand side of the dependency) are not candidate keys.

FD Diagrams

- Rather than an entire table, FDs can be represented simply using the headings:



- {Module , Text} is a candidate key, so we put a double box around them
- {Lecturer} → {Dept}, so we have an arrow from Lecturer to Dept
- {Module} → {Dept} and {Module} → {Lecturer} , so we have {Module} → {Dept, Lecturer}

Note: Trivial FDs and FDs dependent on an entire candidate key are not included

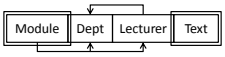
Second Normal Form

- Partial FDs:
  - A FD, $A \rightarrow B$ is a partial FD, if some attribute of A can be removed and the FD still holds
  - Formally, there is some proper subset of A, $C \subset A$, such that $C \rightarrow B$
- Second normal form:
  - A relation is in second normal form (2NF) if it is in 1NF and no non-key attribute is partially dependent on a candidate key
  - In other words, no $C \rightarrow B$ where C is a strict subset of a candidate key and B is a non-key attribute.

Normalising to 2NF

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6



- '1NF' is not in 2NF
  - We have the FD {Module, Text} → {Lecturer, Dept}
  - But also {Module} → {Lecturer, Dept}
  - And so Lecturer and Dept are partially dependent on the primary key

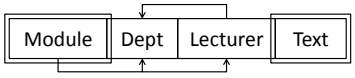
Normalising to 2NF

- Suppose we have a relation R with scheme S and the FD $A \rightarrow B$ where

  $A \cap B = \{\}$
- Let $C = S - (A \cup B)$
- In other words:
  - A – attributes on the left hand side of the FD
  - B – attributes on the right hand side of the FD
  - C – all other attributes
- It turns out that we can split R into two parts:
  - R1, with scheme $A \cup C$
  - R2, with scheme $A \cup B$
- The original relation can be recovered as the natural join of R1 and R2:
  - R = R1 NATURAL JOIN R2

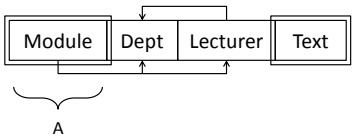
Normalising to 2NF

- We need to remove FD $A \rightarrow B$ in order to convert the relation to 2NF



Normalising to 2NF

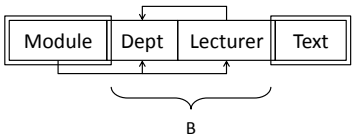
- We need to remove FD $A \rightarrow B$ in order to convert the relation to 2NF



A – The determinant of the functional dependency

Normalising to 2NF

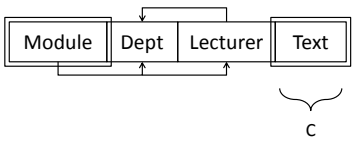
- We need to remove FD $A \rightarrow B$ in order to convert the relation to 2NF



B – The dependant attributes of the functional dependency

Normalising to 2NF

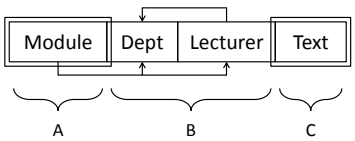
- We need to remove FD $A \rightarrow B$ in order to convert the relation to 2NF



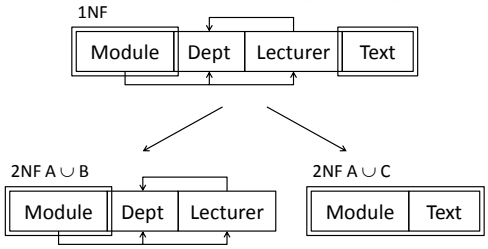
C – All remaining attributes in the relation

Normalising to 2NF

- To convert to 2NF, create two relations $A \cup B$ and $A \cup C$



Normalising to 2NF



Normalising to 2NF

2NFa

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4

2NFb

Module	Text
M1	T1
M1	T2
M2	T1
M2	T3
M3	T4
M4	T1
M4	T5
M5	T6

Problems Resolved in 2NF

- INSERT Anomalies
  - We can now add a module without texts
- UPDATE Anomalies
  - We only need to change a single row when changing a module lecturer

2NFa

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4

Problems Remaining in 2NF

- INSERT Anomalies
  - We can't add lecturers who don't currently teach modules
- UPDATE Anomalies
  - To change the department for L1, we must change two rows
- DELETE Anomalies
  - To delete module M3, we must delete L2

2NFa

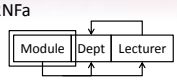
Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4

Transitive FDs and 3NF

- Transitive FDs:
  - A FD, $A \rightarrow C$ is a transitive FD, if there is some set B such that $A \rightarrow B$ and $B \rightarrow C$ are non-trivial FDs
  - $A \rightarrow B$ non-trivial means: B is not a subset of A
  - Essentially $A \rightarrow B \rightarrow C$
- Third normal form
  - A relation is in third normal form (3NF) if it is in 2NF and no non-key attribute is transitively dependent on a candidate key

Normalising to 3NF

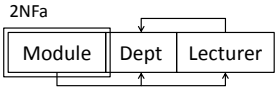
2NFa



- 2NFa is not in 3NF
  - There are FDs
    {Module} → {Lecturer}
    {Lecturer} → {Dept}
  - So there is a transitive FD from Primary key {Module} to {Dept}
- To move a relationship from 2NF to 3NF:
  - Given the transitive FD $A \rightarrow B \rightarrow C$
  - We split the relation into two new relations
  - The first contains all of the columns contained in B and C
  - The second contains all of the columns which are not contained in A, B or C and the columns contained in A and B

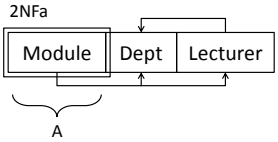
Normalising to 3NF

- We need to remove FD A → B → C in order to convert the relation to 3NF



Normalising to 3NF

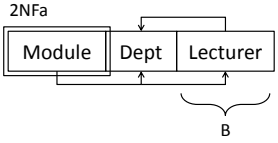
- We need to remove FD A → B → C in order to convert the relation to 3NF



A – The determinant of the functional dependency

Normalising to 3NF

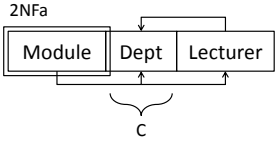
- We need to remove FD A → B → C in order to convert the relation to 3NF



B – The dependant attributes of the functional dependency

Normalising to 3NF

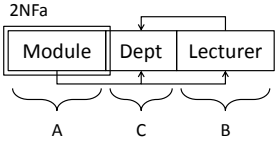
- We need to remove FD A → B → C in order to convert the relation to 3NF



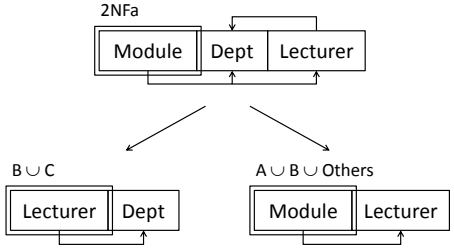
B – The transitively dependant attributes of the functional dependency

Normalising to 3NF

- To convert to 3NF, create two relations B ∪ C and A ∪ B ∪ Other Columns



Normalising to 3NF



Normalising to 3NF

3NFa

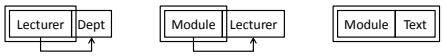
Lecturer	Dept
L1	D1
L2	D1
L3	D2
L4	D2

3NFb

Module	Lecturer
M1	L1
M2	L1
M3	L2
M4	L3
M5	L4

2NFb

Module	Text
M1	T1
M1	T2
M2	T1
M2	T3
M3	T4
M4	T1
M4	T5
M5	T6



Problems Resolved in 3NF

- Problems resolved in 3NF
  - INSERT – We can now add Lecturers who don't teach any modules
  - UPDATE – We need only change a single row to update the department for L1
  - DELETE – We can delete M3 while preserving L2

3NFa

Lecturer	Dept
L1	D1
L2	D1
L3	D2
L4	D2

3NFb

Module	Lecturer
M1	L1
M2	L1
M3	L2
M4	L3
M5	L4

Normalisation and Design

- Normalisation is related to Database design
  - A database should normally be in 3NF at least
  - If your design leads to a non-3NF database, then you might want to revise it
- When you find you have a non-3NF database
  - Identify the FDs that are causing a problem
  - Think if they will lead to any insert, update, or delete anomalies
  - Try to remove them

Next Lecture

- More Normalisation
  - Lossless decomposition
  - BCNF
- Further Reading
  - The Manga Guide to Databases, Chapter 3
  - Database Systems, Chapters 14 and 15