

Lecture 7

- Database normalisation
- 1st, 2nd and 3rd normal form (NF)

Database normalisation

- Why normalise a database?
- Minimise duplication of information in the DB
- Make queries easier

First normal form (1NF)

- A table is said to be in first normal form (1NF) if:
 - the rows are in no particular order
 - the columns are in no particular order
 - every field in every record of the table contains **one** value in accordance with the field definition. Most definitions of 1NF also require that the field cannot be empty (i.e., NULL)
 - each row occurs only once – no duplicates
 - all fields contain actual data, i.e., there are no fields that identify the position of the row in the table

A table that isn't in 1NF

Animal	Continent	No. of feet	Keeper 1	Keeper 2
Elephant	Africa, Asia	4	Fred	Zoe
Snake	Africa, America, Australia	0	Steph	
Snake	Europe	0	Steph	
Tiger	Asia	4	Jenny	George
Emu	Australia	2	Martin	

Why isn't it in 1NF?

Animal	Continent	No. of feet	Keeper1	Keeper2
Elephant	Africa, Asia	4	Fred	Zoe
Snake	Africa, America, Australia	0	Steph	
Snake	Europe	0	Steph	
Tiger	Asia	4	Jenny	George
Emu	Australia	2	Martin	

No unique key – table can have duplicate records

Why isn't it in 1NF?

Animal	Continent	No. of feet	Keeper1	Keeper2
Elephant	Africa, Asia	4	Fred	Zoe
Snake	Africa, America, Australia	0	Steph	
Snake	Europe	0	Steph	
Tiger	Asia	4	Jenny	George
Emu	Australia	2	Martin	

Multiple values from the set of continents in the same field

Why isn't it in 1NF?

Animal	Continent	No. of feet	Keeper1	Keeper2
Elephant	Africa, Asia	4	Fred	Zoe
Snake	Africa, America, Australia	0	Steph	
Snake	Europe	0	Steph	
Tiger	Asia	4	Jenny	George
Emu	Australia	2	Martin	

Multiple columns with the same data (carers)

The same in 1NF

Animal ID	Animal	No. of feet	Animal ID	Continent	Animal ID	Keeper
1	Elephant	4	1	Africa	1	Fred
1	Elephant	4	1	Asia	1	Zoe
3	Snake	0	3	Africa	3	Steph
4	Tiger	4	3	America	4	Jenny
6	Snake	0	3	Asia	4	George
7	Emu	2	4	Asia	6	Steph
			6	Europe	7	Martin
			7	Australia		

NB: A unique key in the two tables on the right could be formed from Animal ID and carer name (presuming for a moment that the carer name is unique) – but you could equally argue that both tables need their own primary keys to be 1NF

Advantages of 1NF here

- Can query for carers without having to decide which field to query
- Can query for continents without having to split the field value from the result set
- No risk of duplicate rows and hence possibly inconsistent/ambiguous updates

Second normal form (2NF)

- A table cannot be in 2NF unless it's already in 1NF
- Any table with a primary key that is defined across one field only is automatically 2NF
- If a key is defined across several fields, then all fields that are not part of the key must depend on the entire key and not just on part of the key

A table that's not 2NF

Who feeds what?

Animal ID	Keeper	Food	Feeding time
1	Fred	Hay	Morning
1	Zoe	Hay	Evening
3	Steph	Mice	Midday
4	Jenny	Beef	Morning
4	George	Beef	Evening
6	Steph	Mice	Midday
7	Martin	Grass	Midday

A table that's not 2NF

Who feeds what?

Animal ID	Keeper	Food	Feeding time
1	Fred	Hay	Morning
1	Zoe	Hay	Evening
3	Steph	Mice	Midday
4	Jenny	Beef	Morning
4	George	Beef	Evening
6	Steph	Mice	Midday
7	Martin	Grass	Midday

Composite key fields

A table that's not 2NF

Who feeds what?

Animal ID	Keeper	Food	Feeding time
1	Fred	Hay	Morning
1	Zoe	Hay	Evening
3	Steph	Mice	Midday
4	Jenny	Beef	Morning
4	George	Beef	Evening
6	Steph	Mice	Midday
7	Martin	Grass	Midday

Food depends on animal only – just part of the key

Making it 2NF

- Move food field to animal table (where the animal id is the primary key)
- (Possibly) move the feeding time into a separate table if it depends on the animal or on the keeper's shift only

Advantages of 2NF

- Avoids unnecessary duplication of field values across rows (animals eat the same stuff no matter who feeds them)

Third normal form (3NF)

- A table in 3NF must be in 2NF already
- All fields in the table must depend on the primary key only
- That means: an update of a field other than the primary key must not require the update of another field

A table that's not in 3NF

Animal ID	Animal	No. of feet
1	Elephant	4
3	Snake	0
4	Tiger	4
6	Snake	0
7	Emu	2

A table that's not in 3NF

Animal ID	Animal	No. of feet
1	Elephant	4
3	Snake	0
4	Tiger	4
6	Snake	0
7	Emu	2

Primary key

A table that's not in 3NF

Animal ID	Animal	No. of feet
1	Elephant	4
3	Snake	0
4	Tiger	4
6	Snake	0
7	Emu	2

Updating the number of feet requires update of animal (and often vice versa)

Getting it into 3NF

- Need another table that defines the animal species (as primary key) and the number of feet that go with it

Advantages of 3NF

- Can update columns independently

Why normalisation is sometimes not good for you

- Increasing normalisation of all and everything increases the number of tables
- Means we need complex joins (or at least complex WHERE clauses) more often – computationally intensive for the RDBMS
- More complex for the developer – easy not to see the data relationships for the tables

Today's lab sheet

- ...is on the web.
- Today: Normalising a database from <NF to 3NF