Normalisation

Designing databases

Normalisation

Overview

- Why normalise?
- How to normalise
- Where normalisation fits in

Why normalise?

- ensures the design of the database is sound
  - ➢tables are properly structured
- ensures **anomalies** can’t occur
- 3 types of anomalies
  - Insertion
  - Modification
  - Deletion

What are anomalies?

Student No	Student	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	Cert IV	Cert 4 of IT	31	Minnie Mouse
100234	Huey	BIT3	Bach of IT	45	Mary Poppins
100234	Huey	Cert IV	Cert 4 of IT	31	Minnie Mouse

Consider these situations

- Add a new coordinator
- Minnie Mouse changes her name
- Huey leaves the university

Inovoice tables

- Properly structured
  - Primary/foreign keys
  - Referential integrity: every foreign key entry must have a matching primary key entry
- See invoice tables
- Not properly structured

What are anomalies?

Student No	Student	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	Cert IV	Cert 4 of IT	31	Minnie Mouse
100234	Huey	BIT3	Bach of IT	45	Mary Poppins
100234	Huey	Cert IV	Cert 4 of IT	31	Minnie Mouse

Consider these situations

- Add a new coordinator
- Minnie Mouse changes her name
- Huey leaves the university

Properly structured

Student

No	Name
783917	Charlie Chaplin
983403	Pluto
100234	Huey

Course

Course	CourseName	CoordId
BIT99	Bach of IT	23
BIT3	Bach of IT	45
Cert IV	Cert IV of IT	31

Studying

No	Course
783917	BIT99
983403	BIT99
983403	Cert IV
100234	BIT3
100234	Cert IV

Coordinator

CoordId	CoordName
23	Fred Flintstone
45	Mary Poppins
31	Minnie Mouse

Normalisation

Normalisation (and SQL) has roots in

- Set theory
- Relational algebra
- All based around tables

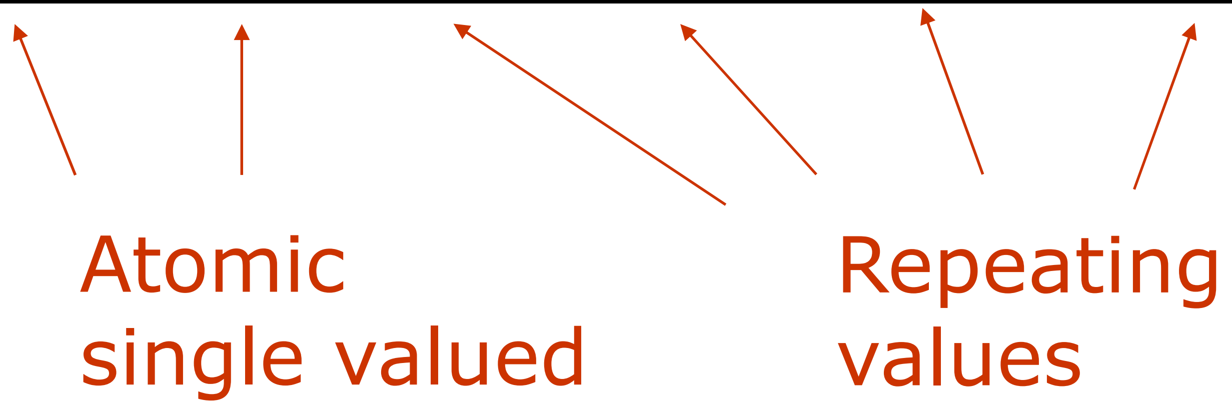
➢ Anomalies usually resolved by splitting tables

Normal Forms

- First Normal Form (1NF)
  - All fields are atomic
- Second Normal Form (2NF)
  - 1NF and no partial dependencies
- Third Normal Form (3NF)
  - 2NF and no transitive dependencies
- Higher forms exist but apply only in very specialised circumstances

1NF – all fields atomic

No	Name	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99 Cert IV	Bach of IT Cert 4 of IT	23 31	Fred Flintstone Minnie Mouse
100234	Huey	BIT3 Cert IV	Bach of IT Cert 4 of IT	45 31	Mary Poppins Minnie Mouse



Atomic single valued

Repeating values

1NF

- If not in 1NF then
  - Expand/duplicate data so that each row in the relation contains atomic values
- Identify the Primary Key
  - Candidate keys: contender for the primary keys
  - Primary key: the candidate key selected

No	Name	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99 Cert IV	Bach of IT Cert 4 of IT	23 31	Fred Flintstone Minnie Mouse
100234	Huey	BIT3 Cert IV	Bach of IT Cert 4 of IT	45 31	Mary Poppins Minnie Mouse

All values atomic

No	Name	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	Cert IV	Cert 4 of IT	31	Minnie Mouse
100234	Huey	BIT3	Bach of IT	45	Mary Poppins
100234	Huey	Cert IV	Cert 4 of IT	31	Minnie Mouse

What is the primary key?

primary key

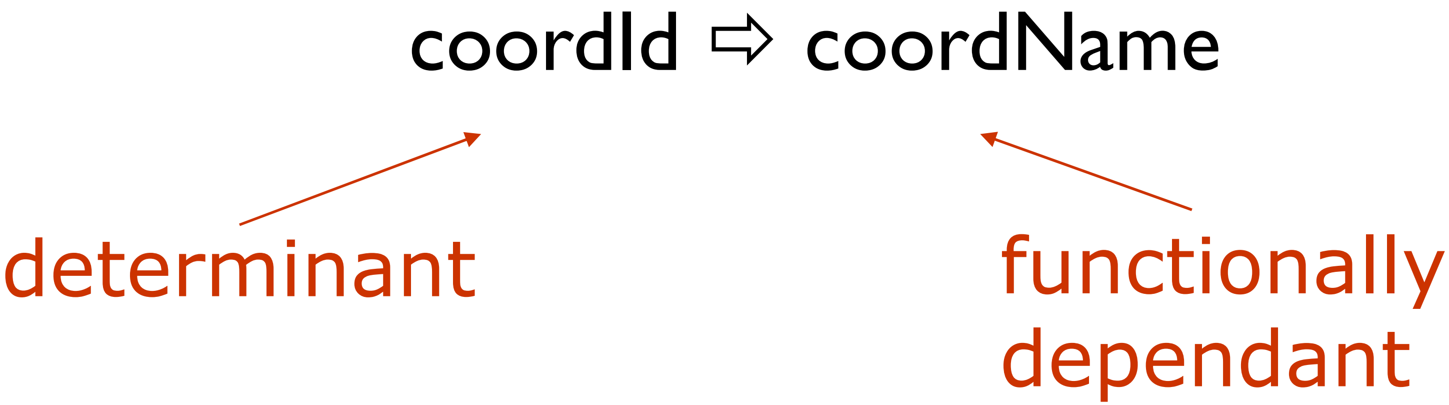
Determinants

No	Name	Course	CourseName	CoordId	CoordName
783917	Charlie Chaplin	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	BIT99	Bach of IT	23	Fred Flintstone
983403	Pluto	Cert IV	Cert 4 of IT	31	Minnie Mouse
100234	Huey	BIT3	Bach of IT	45	Mary Poppins
100234	Huey	Cert IV	Cert 4 of IT	31	Minnie Mouse

A determinant is a field (or fields) whose value determines the value of another field

Functionally dependant

- A field is functionally dependant on another field (or fields) if that field is a determinant for it.



coordId ⇨ coordName

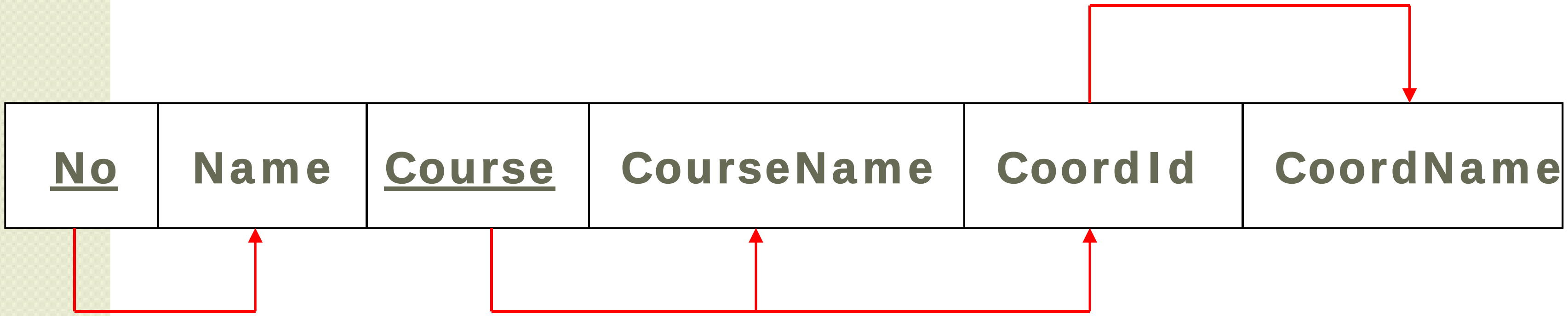
determinant

functionally dependant

Functional Dependencies

- A constraint between two sets of attributes
- For any Relation R, Attribute B is functionally dependent on Attribute A if, for every valid instance of A, that value of A uniquely determines the value of B
  - A ⇨ B
- Examples
  - StudentNo ⇨ StudentName, DOB
  - StudentNo, Unit, Semester ⇨ Result
  - Unit, Semester ⇨ Lecturer [business rules]
  - AccountNo ⇨ Branch

Other Functional Dependencies?



No	Name	Course	CourseName	CoordId	CoordName

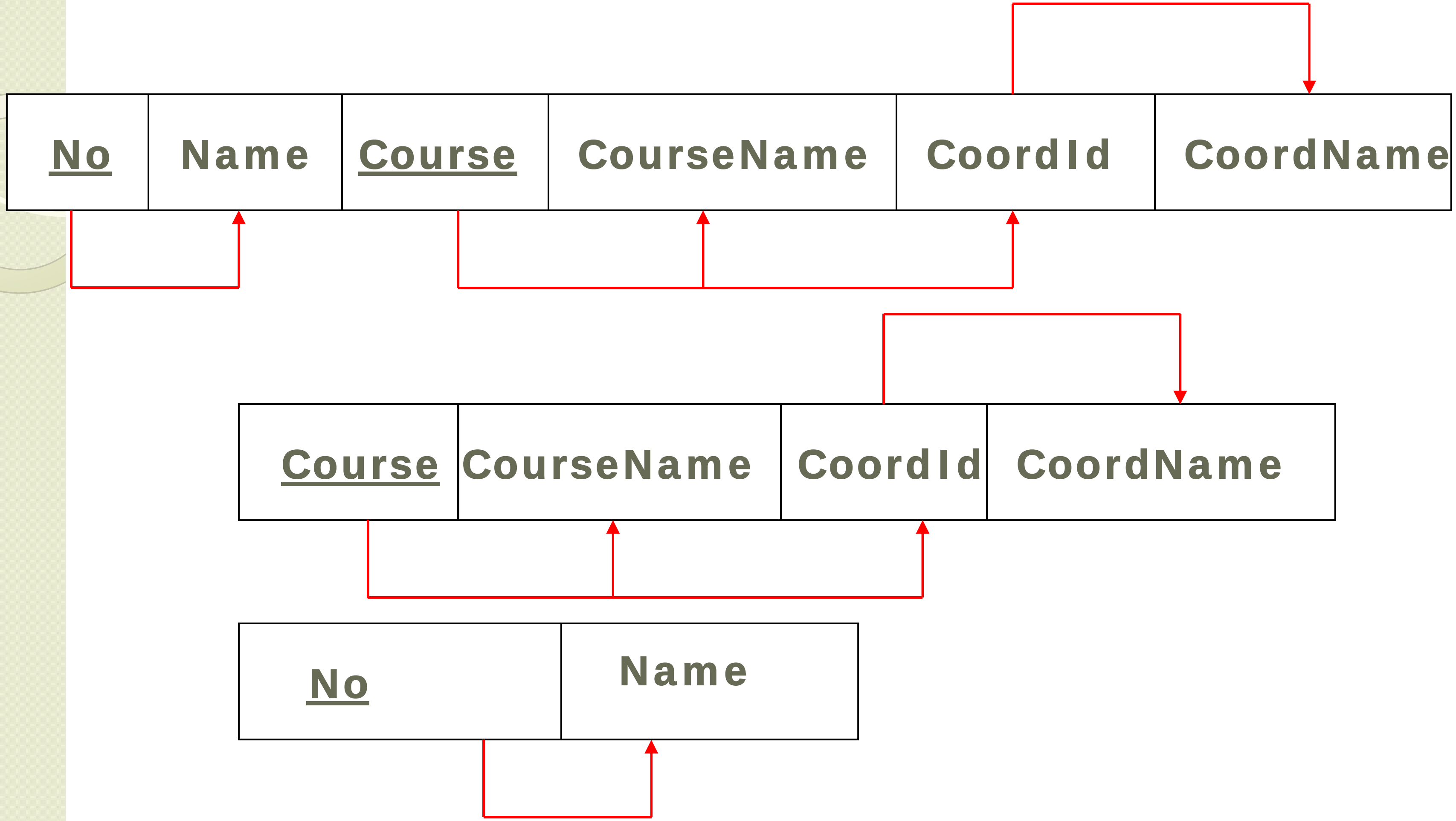
CoordID => CoordName

Course => CourseName,CoordID

No => Name

2NF: No Partial dependencies

- Partial dependency: a non-key field is functionally dependant on the ***part*** of the primary key (not the entire primary key)
- Split table
  - Determinant + attributes
  - Determinants stay in original relation



No	Name	Course	CourseName	CoordId	CoordName

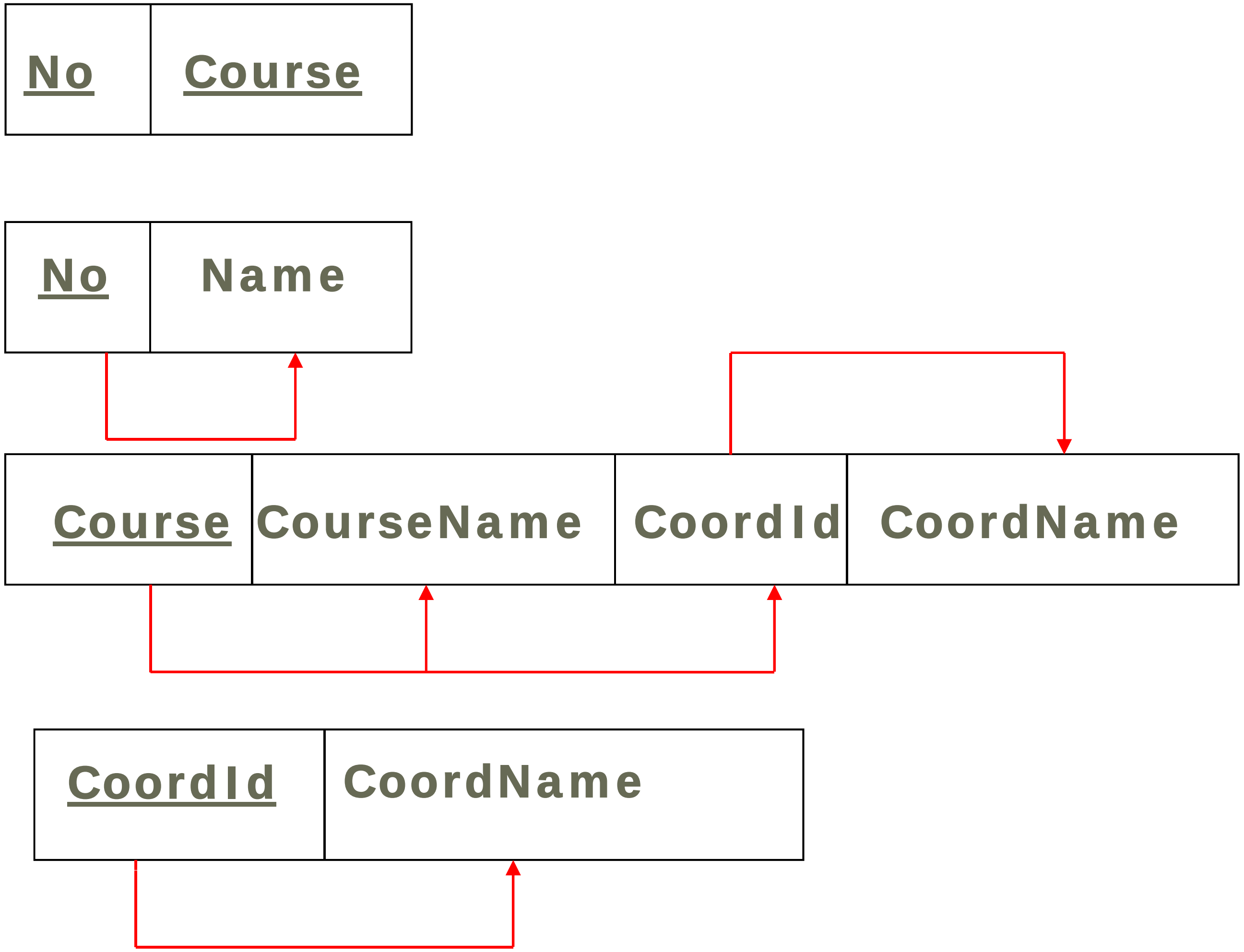
Course	CourseName	CoordId	CoordName

No	Name

3NF: No Transitive Dependencies

- Transitive dependencies: functional dependency between 2 non-key fields
- Split table
  - Determinant + attributes
  - Determinants stays in old relation

Third Normal Form (3NF)



No Course

No Name

Course CourseName CoordId CoordName

CoordId CoordName

Split the tables

No	**Name**

Student

No	Name
783917	Charlie Chaplin
983403	Pluto
100234	Huey

Course	**CourseName**	**CoordId**

Course

Course	CourseName	CoordId
BIT99	Bach of IT	23
BIT3	Bach of IT	45
Cert IV	Cert IV of IT	31

No	**Course**

Studying

No	Couse
783917	BIT99
983403	BIT99
983403	Cert IV
100234	BIT3
100234	Cert IV

CoordId	**CoordName**

Coordinator

CoordId	CoordName
23	Fred Flintstone
45	Mary Poppins
31	Minnie Mouse

Recap

- First Normal Form (1NF)
  - All fields are atomic
- Second Normal Form (2NF)
  - 1NF and no partial dependencies
- Third Normal Form (3NF)
  - 2NF and no transitive dependencies
- Functional Dependencies

Invoice table

For the following

1. Determine functional dependencies
2. Show 1NF, 2NF, 3NF

Questions

- Invoice tables not structured
- Complete the question sheets