COMP2001 Database Design & Modelling

Martin Read

This Lecture

Aim:

- Introduce the key principles of data modelling

Learning Outcomes:

- Identify entities and their relationships and express these in an Entity Relationship Diagram
- Be able to normalise data from a scenario
- Understand how to model data from requirements
- To be able to create tables from the entities

Databases

- Every time you search for a product on **Amazon**
- Send messages to friends in **Facebook**
- Watch a video in **YouTube**
- Or search for directions in **GoogleMaps**
  - you are using a database

What is data?

- Individual facts about something or somebody that have not been organised
  - (sometimes called ‘raw data’)

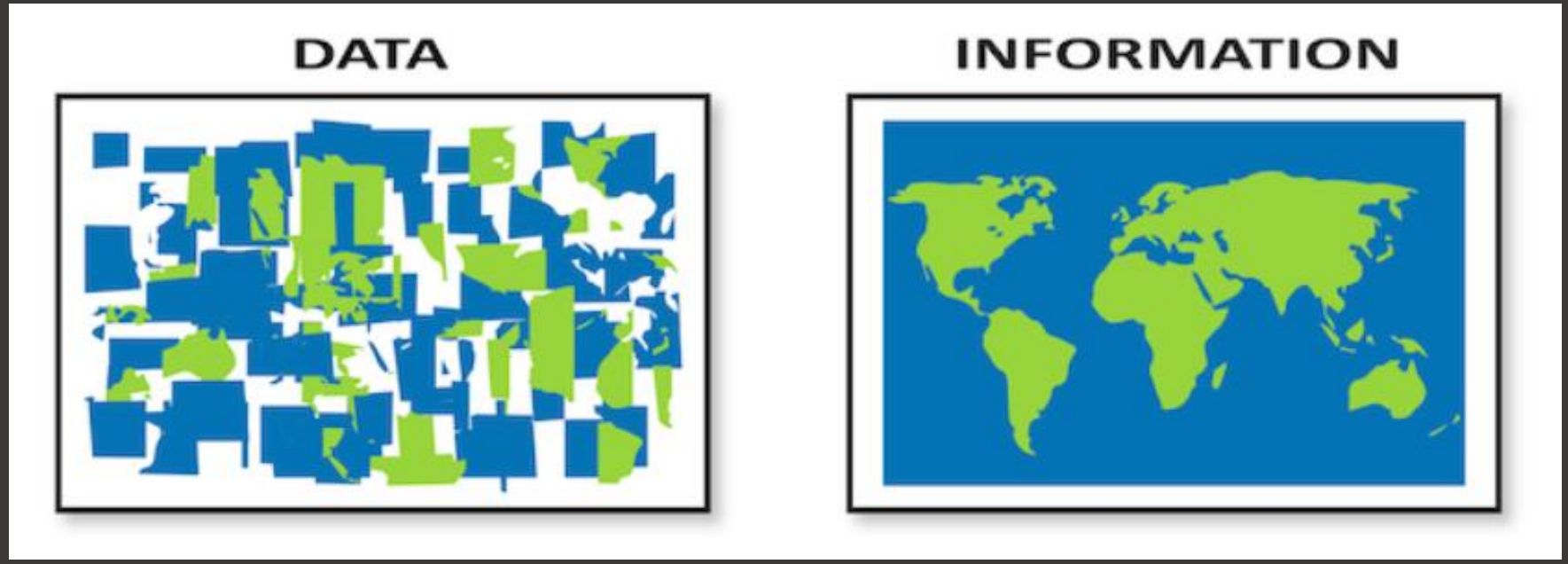
e.g. a random collection of names & telephone numbers

What is information?

- Data which has been organised in some way
  - so as to be useful to somebody

e.g. telephone directory – data which has been organised into alphabetical order so that a telephone number can be found from the name of the individual

Data vs information



Data is raw, unorganised facts that need to be processed

Information refers to the meaning of data as understood by a person

How can data be processed into information?

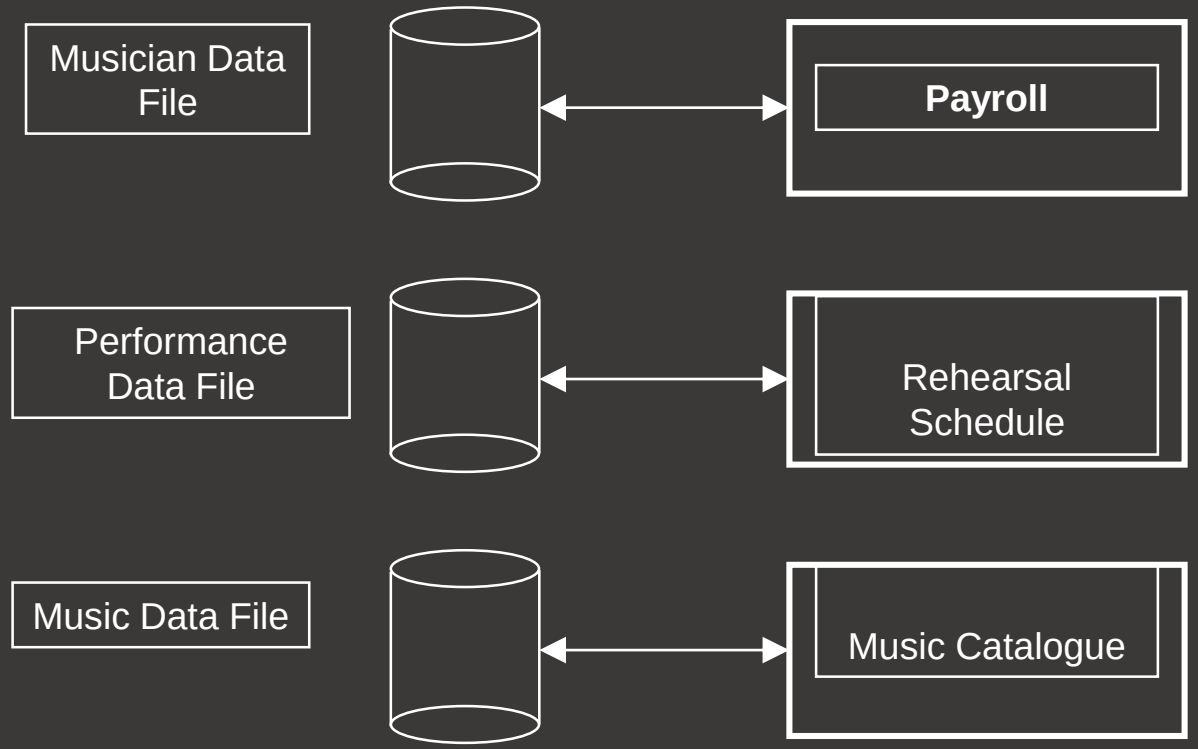
- Sorting
- Searching
- Filtering
- Aggregating
- Performing additional calculations

Characteristics of useful information

- Up to date
- On time
- **Consistent**
- Relevant
- Accessible to authorised personnel
- Secured against unauthorised access
- Complete
- Presented in a useable form

Conventional File System

- Separate applications and **DATA**
- Data stored according to the needs of each application



Flat files: Data Duplication

Track

Track Title	Artist Name	Country
Paranoid	Black Sabbath	UK
Falling in Love	Aerosmith	US
Pink	Aersmith	US
Love in an Elevator	Aerosmith	US
Smooth Criminal	Alien Ant Farm	US
Meaning of Life	Disturbed	US
The Game	Disturbed	US
Voices	Disturbed	US
Down with the Sickness	Disturbed	US

What is a database?

- An integrated collection of data organised to meet the needs of one or more users
- Any collection of related information grouped together

- **Relational databases** group data using common attributes found in the data set

What is CRUD?

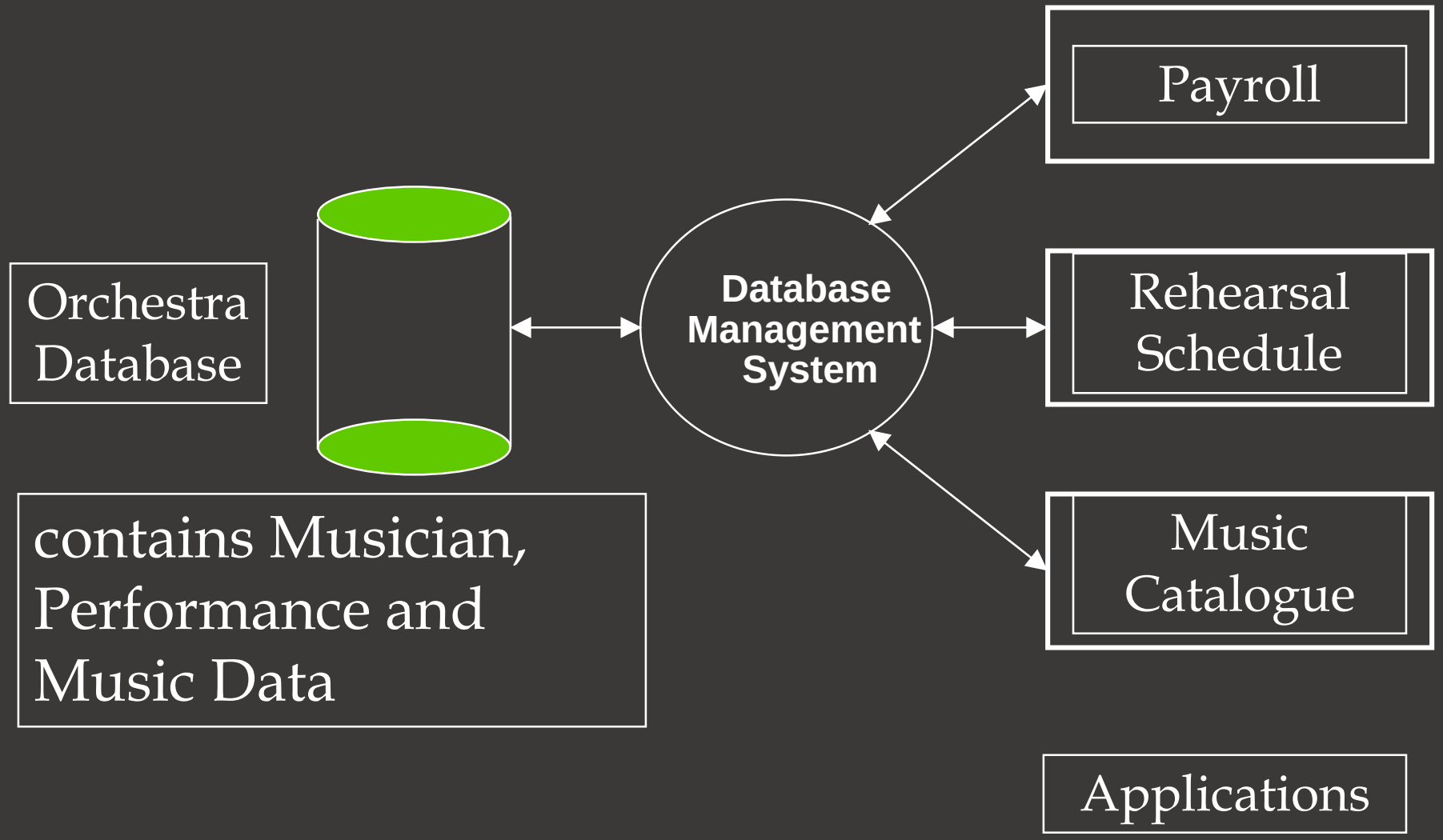
- Describes the elemental functions of a database. Databases allows users to:
  - Create
  - Read
  - Update
  - Delete

    the data in some manner
- These functions are also descriptive of the data life cycle

Why is database design important?

“If the database design doesn't provide a solid foundation for the rest of the project to build upon, the application as a whole will fail”

Database Management System



Analysis

- How can we model the present system?
  - What is it you want the database to do?
  - What inputs/output?
- What data is available? How are they related?
- What properties need to be stored to support the application?
- What other data is required to complete the application?
- Are there samples of real world data (forms, reports, screens…) which can be used?

What is data modelling?

- Technique for describing information (or data) structures

- Concerned with:
  - What data does the system need to store/have?
  - What is the most efficient way of organising the data?

Data modelling principles

- Techniques:
  - Entity-Relationship (E-R) modelling

    Or
  - Logical Data Structure/Model (LDM)
  - Relational Data Analysis (Normalization)

Entity-Relationship (E-R) modelling

- Technique for describing data (or information) in a structured way
- Conceptual/abstract view
  - start with the basic elements of the system and build them up into a coherent model
- E-R modelling does **not** represent the flow of data through a system

…what do we need to store data about?…

Data Modelling

- Information models represent:
  - Things – **entities**
  - Properties of things – **attributes**
  - Associations between things - **relationships**

What is an Entity?

Abstractions of real world things...

Definition:

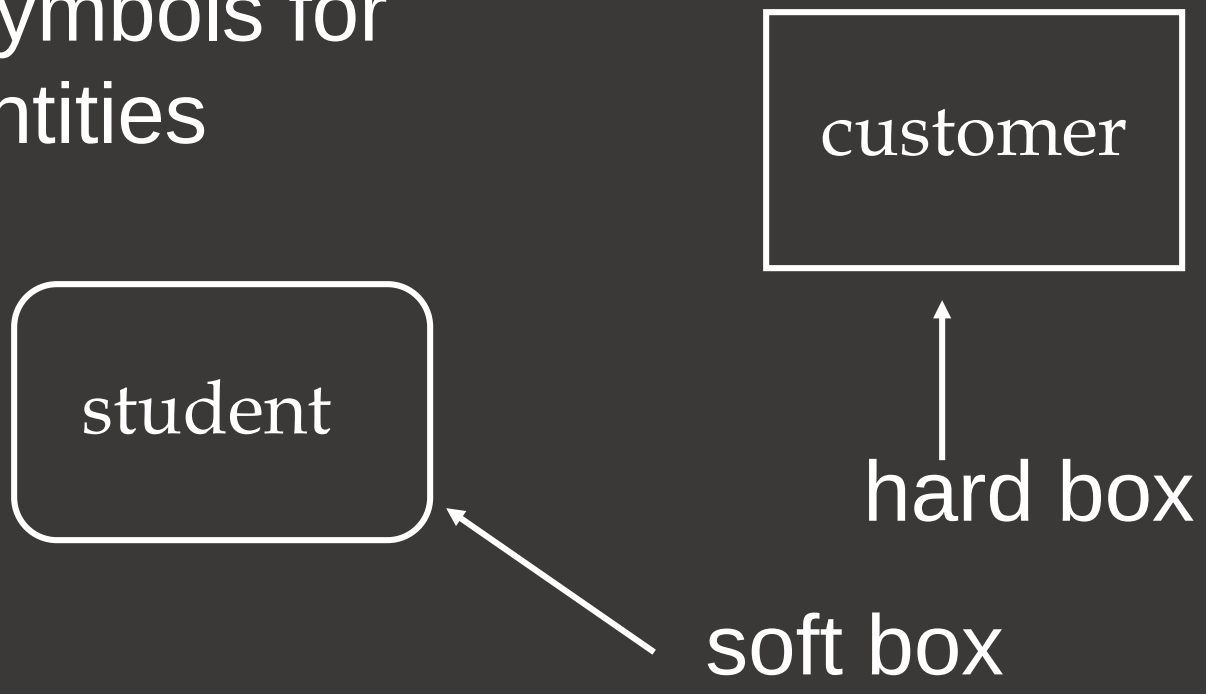
…“Something” of significance to the system about which information is to be held. It must have the capability to be uniquely identified…

e.g. Retail systems - customer, product, order, etc...

...any distinguishable person, place, thing, event or concept about which information is kept

Diagrammatic Representation

Symbols for entities



What is an Attribute?

… are the “data items” or “elements” that make up the entity…

- Must be significant to the system

e.g customer’s name, *customer’s phone number*, *customer’s address*, etc

Values

“contents” of the attributes for a particular entity occurrence

e.g. A. Smith is a value for the *customer name* attribute

Logical Data Model

Customer Entity

Attribute	Value
Customer Number	L0132
Name	Joseph Smith
Address	1 Larch Rd Plymouth
Telephone Number	01752 365874
Credit Limit	£5000

Identifiers/keys

- A **key** is a data item that allows us to **tell records apart**

A **key** is an attribute (or a set of attributes) that **uniquely identifies a record/row**

What is a relationship?

Definition:

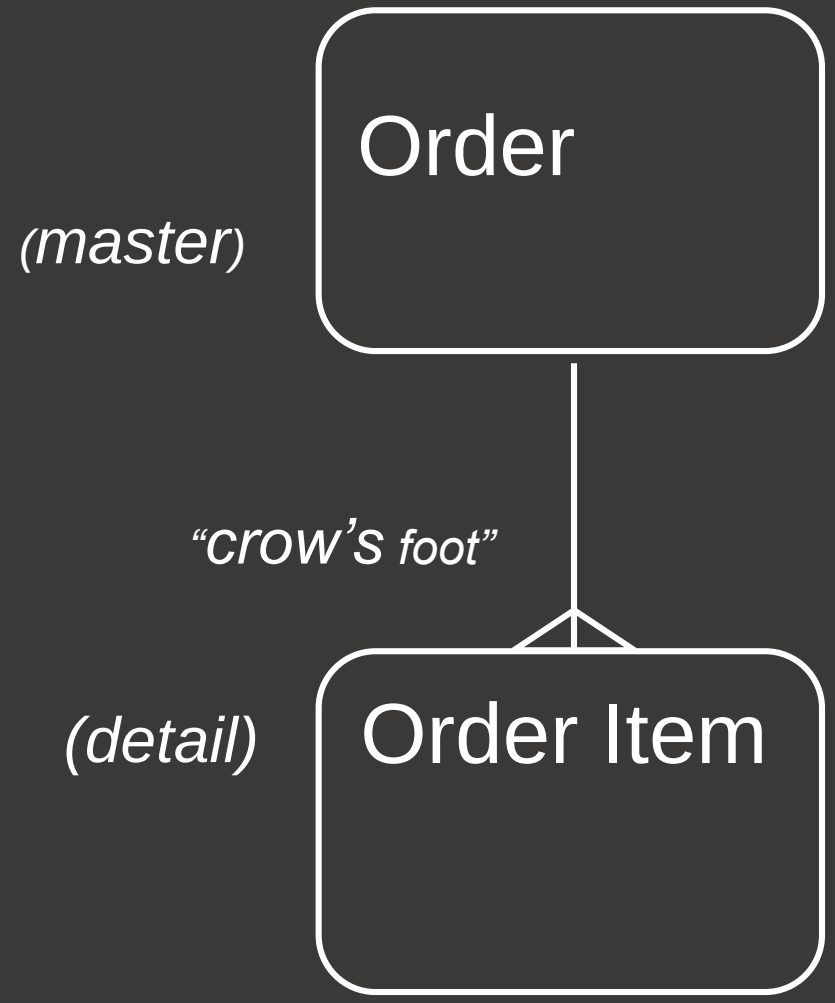
…is a link(association) between two entities which is significant for the system…

e.g. a relationship exists between a customer and their order/s

a CUSTOMER *places* an ORDER

Lecturers teach Students

Entity Relationship Diagram (ERD)



each order is a list of one or more items

each order item is listed on one and only one order

Relationship Types

- One-to-one

- One-to-many

- Many-to-one

- Many-to-many

Examples

- The University may merge the student system and the library system. What is the relationship between a students university record and their library record?
- What is the relationship between a student and the modules they take?
- What is the relationship between a module leader and a module?

Scenario 1

- A University library holds books for its student members to borrow. Each book may be attributed to one or more authors. Any one author, of course, may have written several books. Up to 10 copies may be held of popular titles
- A member may borrow up to six books at a time
- If no copies of a wanted book are currently in stock, a member may make a reservation for the title until it is available

Scenario 1

- A University library holds **books** for its **student** members to **borrow**. Each book may be attributed to one or more **authors**. Any one author, of course, may have written several books. Up to 10 **copies** may be held of popular titles
- A member may borrow up to six books at a time
- If no copies of a wanted book are currently in stock, a member may make a **reservation** for the title until it is available

Scenario 2

Roundel Print Ltd.

Roundel Print carries out a wide range of single and multi-colour printing jobs using various types of paper purchased from a number of suppliers. Each purchase order placed on a supplier account may contain various paper sets.

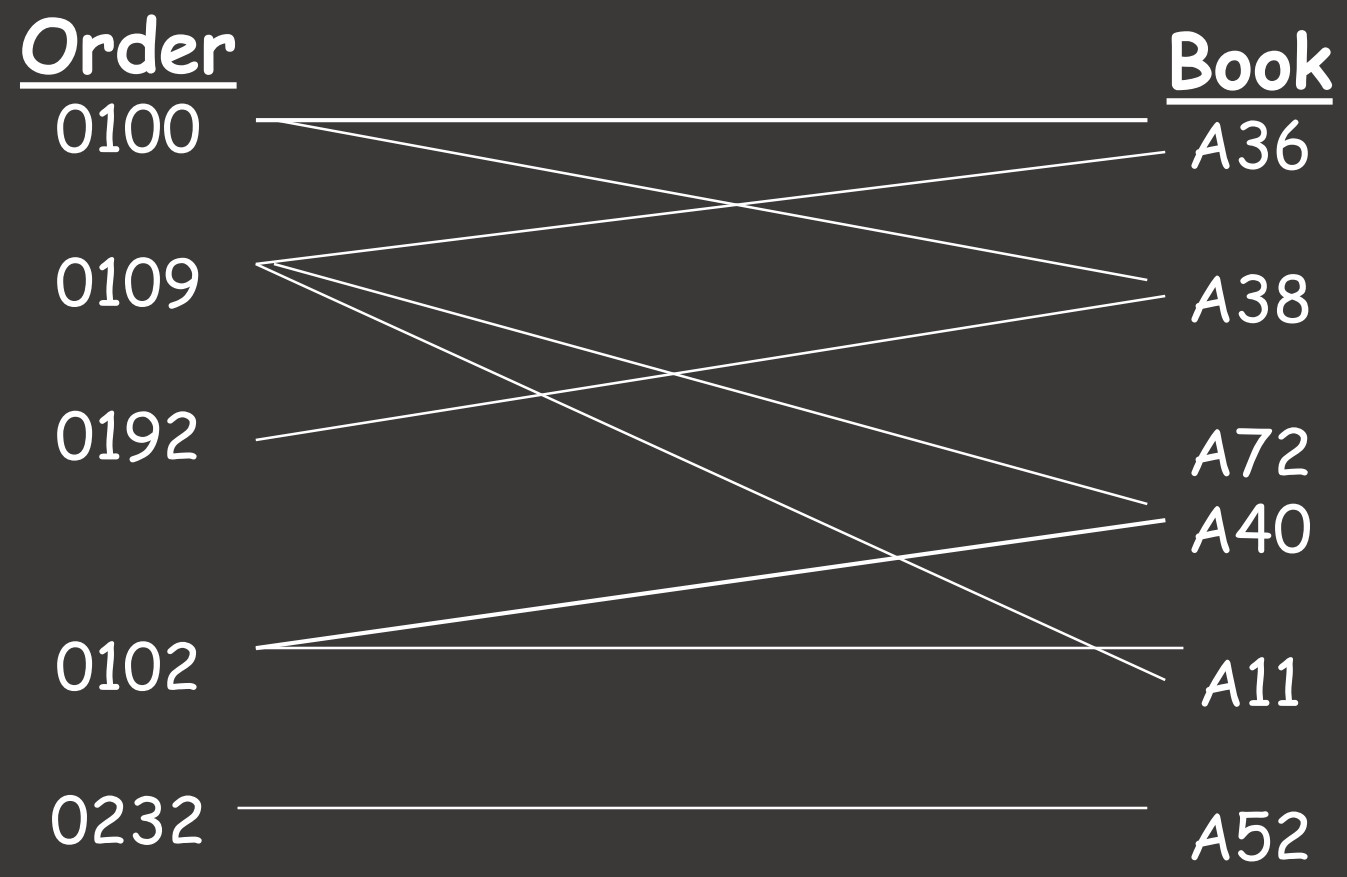
Roundel gives estimates for work when enquiries are received from customers. When a customer places an order the various costs for each job on the order are recorded. For analysis purposes each cost is assigned to a cost-centre. These costs are used to calculate the customer invoice which is charged to the customer account when the order is complete.

Scenario 2

Roundel Print Ltd.

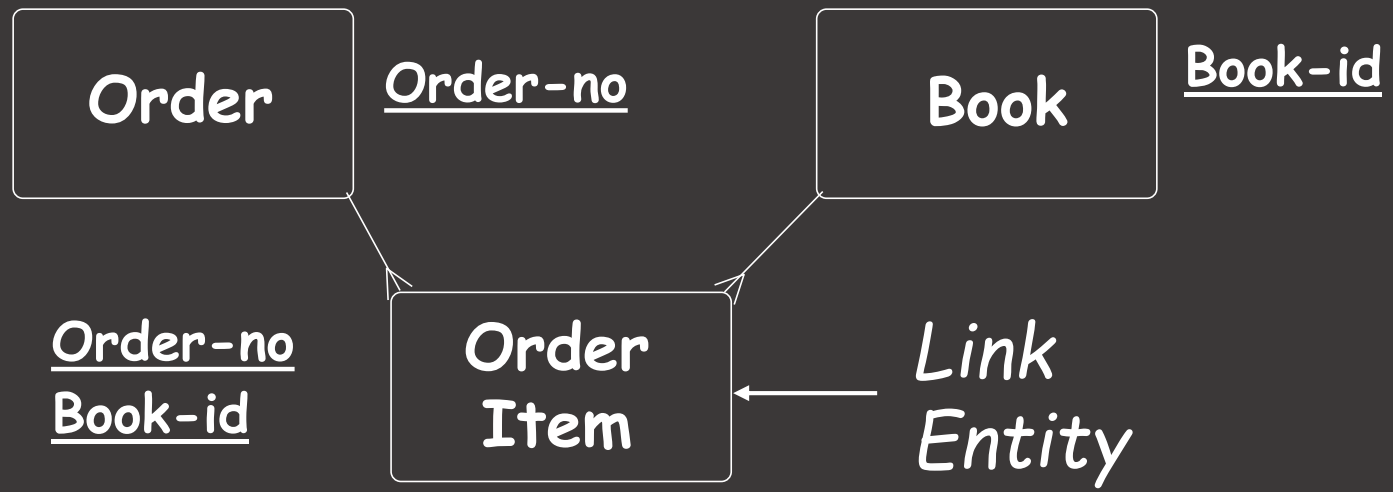
Roundel Print carries out a wide range of single and multi-colour **printing jobs** using various types of **paper** purchased from a number of **suppliers**. Each **purchase order** placed on a **supplier account** may contain various **paper sets**.

Roundel gives **estimates** for work when **enquiries** are received from **customers**. When a customer places an **order** the various **costs** for each **job** on the order are recorded. For analysis purposes each cost is assigned to a **cost-centre**. These costs are used to calculate the **customer invoice** which is charged to the **customer account** when the order is complete.


Order
0100
0109
0192
0102
0232
Book
A36
A38
A72
A40
A11
A52

Many to Many Relationships

Resolving Many-to-Many Relationships



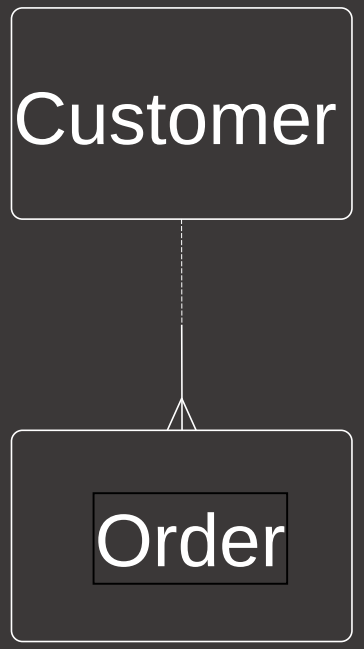
Create a link (or association) entity which is a **detail** to both the original entities

What are the key identifiers for the link entity?

Optional Relationships

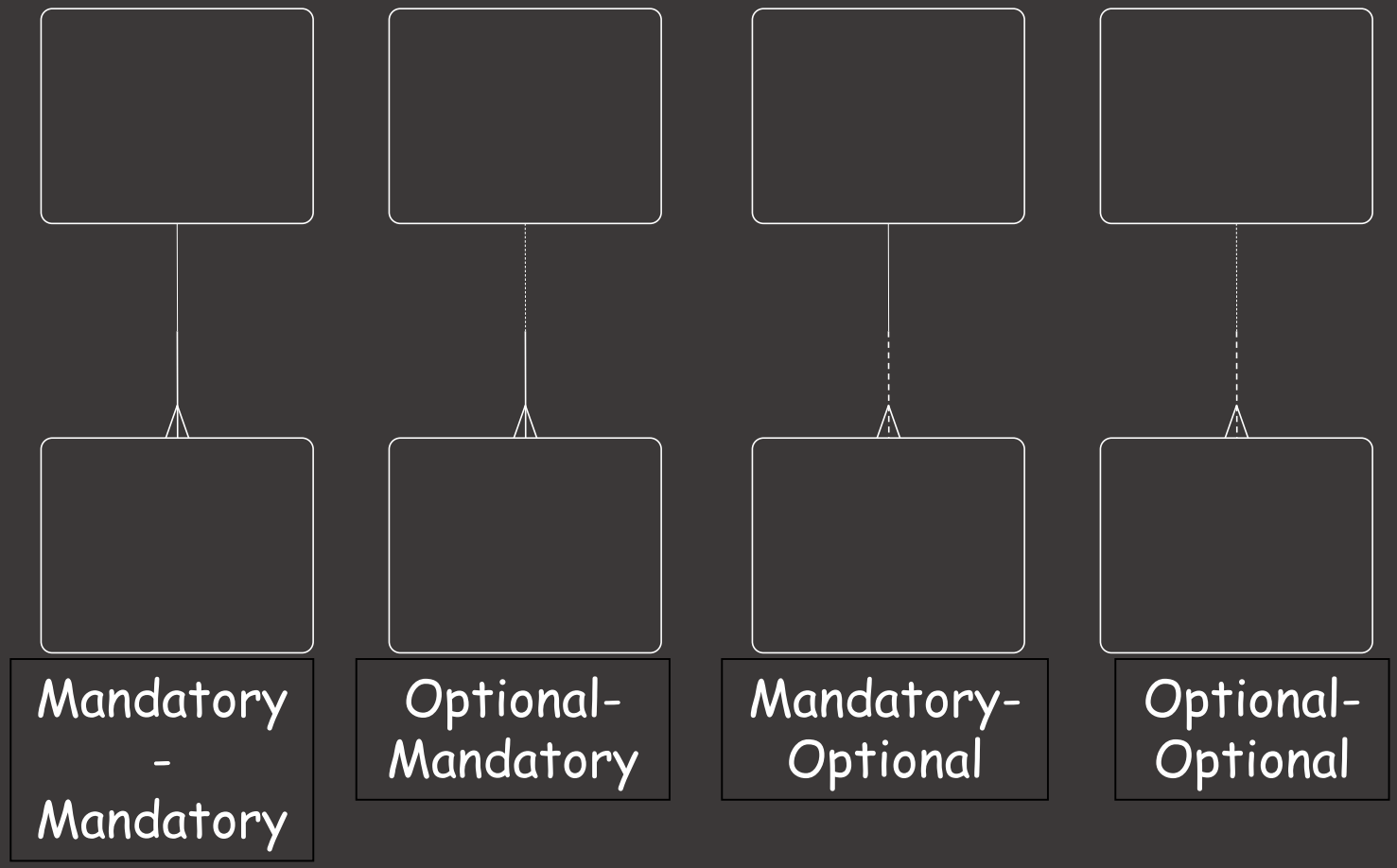
a Customer **may** send one or more orders

an Order **must** be from one and only one Customer



optional at the Customer end but mandatory at the Order end

Membership of Relationship



Optionality

Check each relationship
- can they exist independently at either end

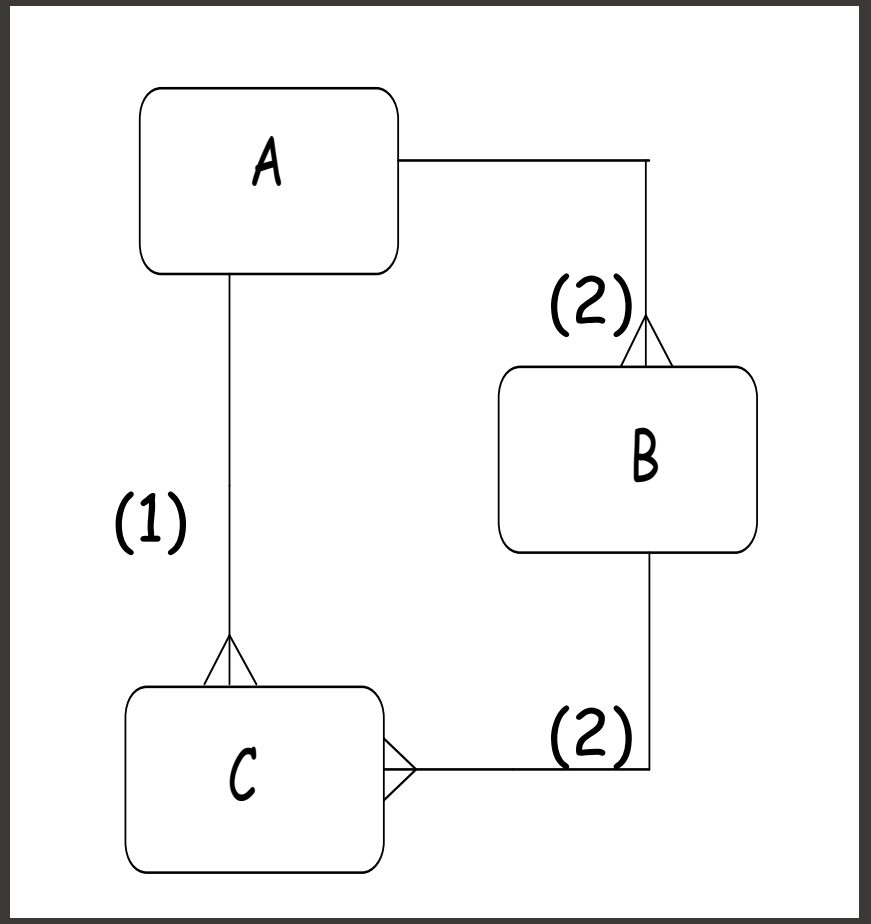
- If occurrence must always be a member of a relationship
  - a mandatory relationship
- If occurrence does not always have to be a member of a relationship
  - an optional relationship

Defining Relationship Link Phrases:

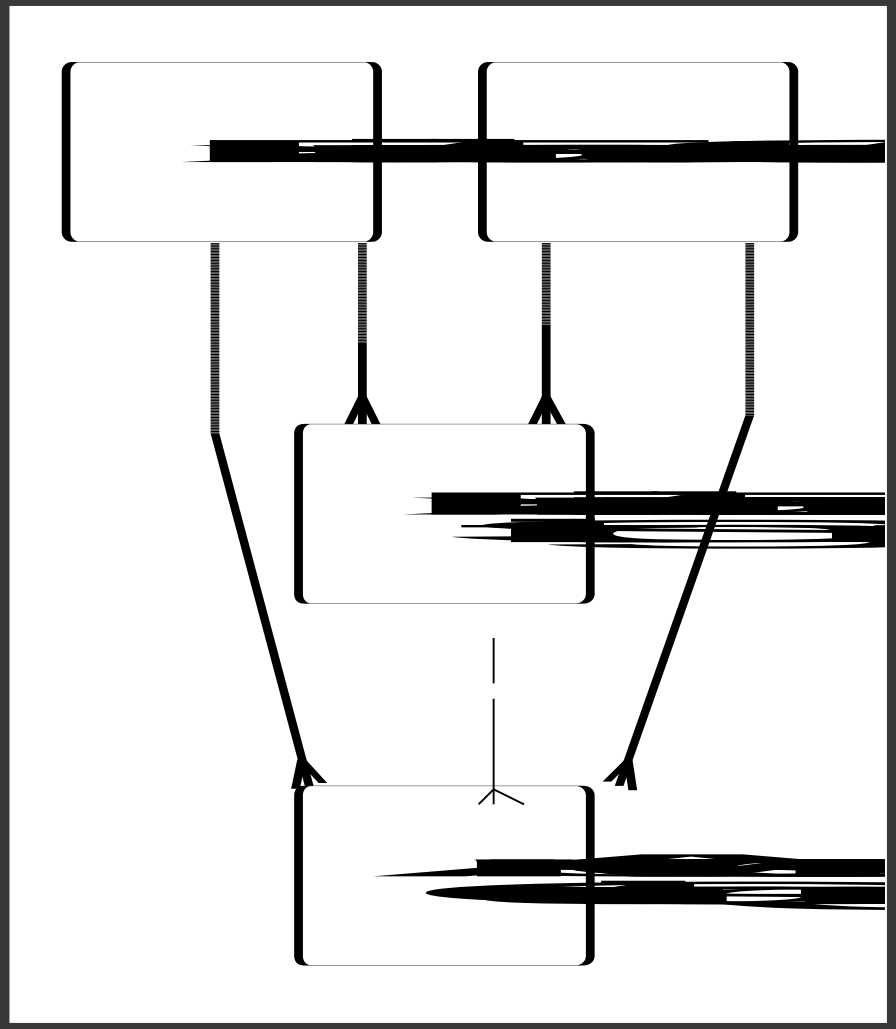
each <subject entity> **must be | may be** <link phrase>
one and only one | one or more <object entity>

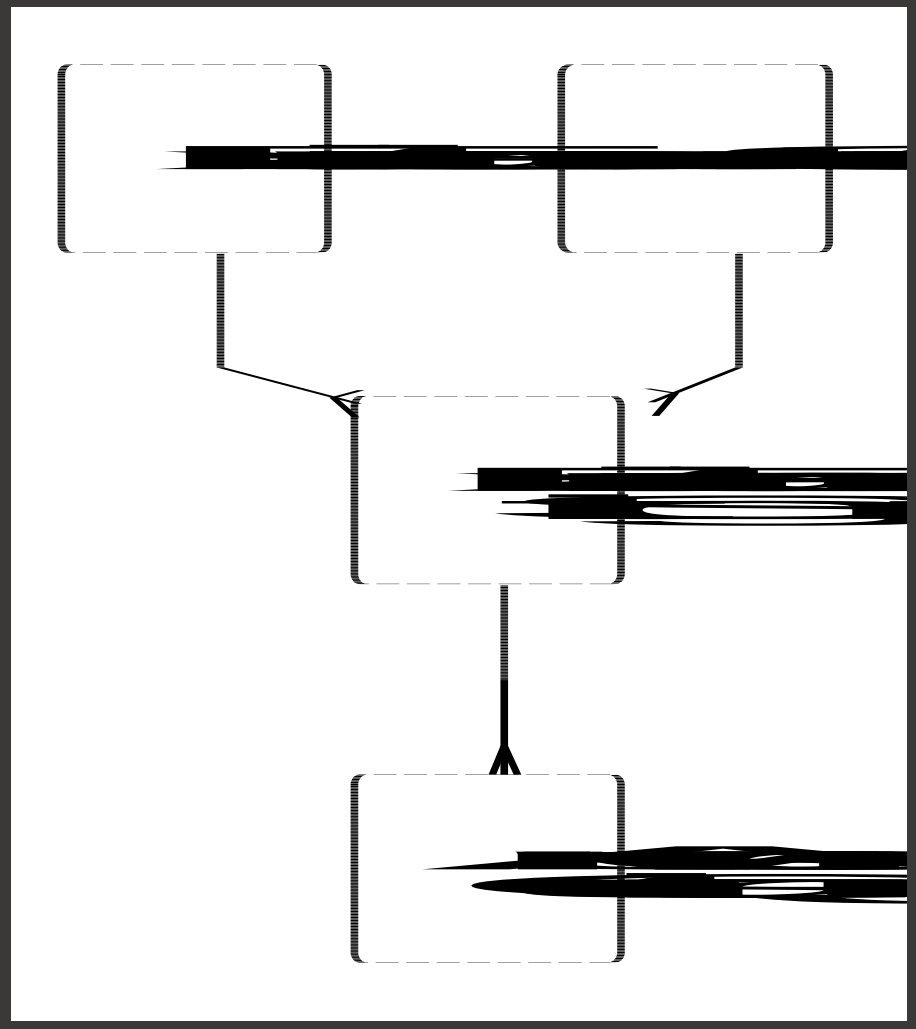
Dealing with possible redundant relationships

- There are two paths between entities A & C
- Which is redundant?
- Route 1 may seem simple & quicker but is redundant, because…
- Route 2 can satisfy the requirement A-C via A-B and B-C



Redundant relationships





Deriving a E-R model

- Analyse business procedures/processes
- Look for nouns/noun phrases & verbs
  (remember: some may be attributes!)
- Interview users
- Use experience
- Consider what entities *ought* or *need* to exist
- Review Data Flow Model

Deriving a E-R model (2)

- Carry out further checks
  - is the ‘entity’ a ‘must’ for the business application?
  - is it really an entity or an attribute?
  - is it another name for an entity already identified (a ‘synonym’)?
  - is it something that, although not an entity, may inform us about business rules/processing and therefore ought to be kept on one side for further analysis?

Developing an E-R model

- Identify possible entities
- Identify relationship between the entities
- Draw initial **entity relationship diagram**
- Add membership for each relationship
- **Resolve One-to-One** and **Many-to-Many relationships**
- Remove redundant relationships
- Tidy up - check optionality of relationships
- Validate the model with **Relational Data Analysis**

Summary

Introduction to the key principles of data modelling

- What data modelling is
- The need for data analysis and data modelling
- The techniques, definitions and concepts of data modelling

Relational Data Analysis (Normalisation)

Martin Read

Lecture Objectives

Aim:

- Introduction to normalisation

Learning Outcomes:

- Be able to normalise data from a scenario

What is Relational Data Analysis (RDA)?

Relational Data Analysis (RDA) is a technique for deriving data structures based on the mathematics of set theory.

The process is underpinned by Relational Algebra.

We will use a method of partitioning samples of data attributes into relations known as Normalisation.

Normalisation

- Objective is to structure real world data into relations
- Ensures efficiency of file organisation
  - processing performance
- Removes redundant (duplicate) data
  - threat to integrity as possible to update different occurrences of the same data item
- Provides a systematic approach to database design by following a set of rules
- 7 forms – we go only to 3NF

Normalisation

- Result of normalisation is to identify *entity types* along with their *attributes*
- One or more of the attributes will constitute the *key* of the entity
- Any attribute(s) which determines another attribute(s) is called a key
- All non-key attributes of a given entity must be “fully functionally dependent” on the key

Some concepts and terms...

Relation

- 2-dimensional array (table)
- Consisting of rows and columns

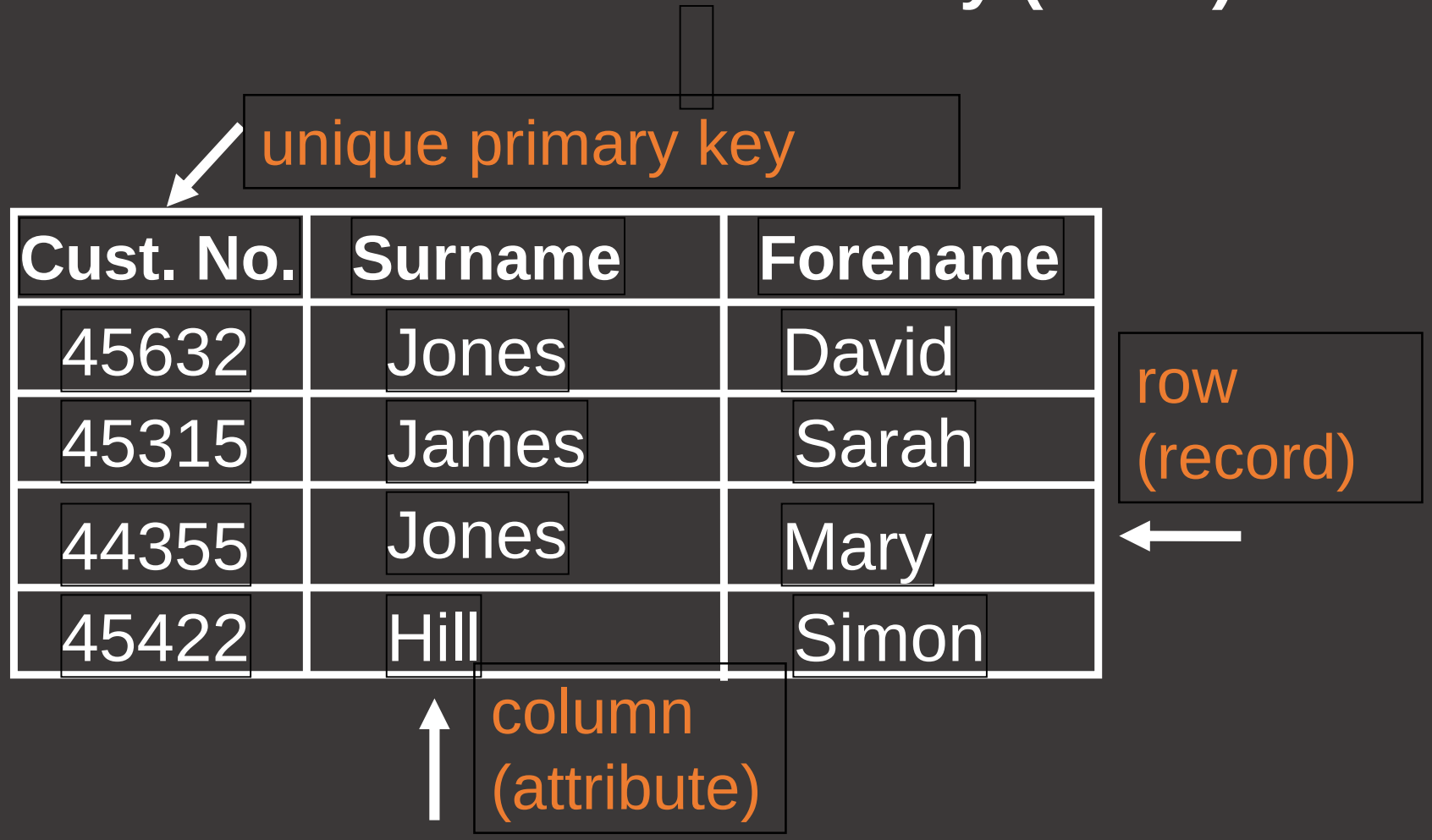
Keys

- Primary key
- Foreign key
- Simple key
- Compound key
- Composite key

Normal forms: Unnormalised, 1st, 2nd, 3rd…

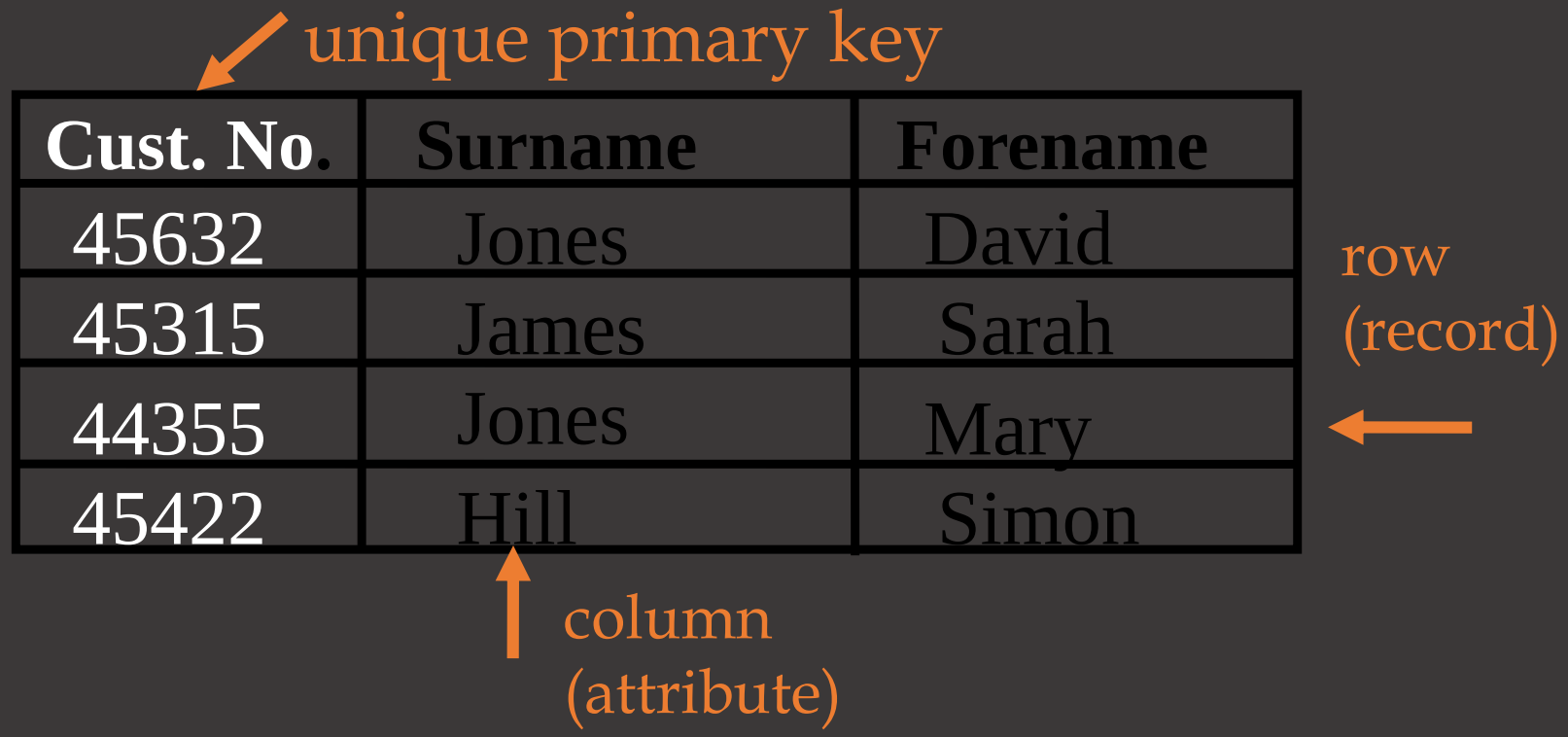
A Relation

A two-dimensional array (table)



Primary Key

One or more attributes whose value(s) are unique for each occurrence (record)



Types of keys

- **Primary key**: best choice from all candidate keys for a relation
- **Simple key**: a key consisting of one attribute
- **Compound key**: a key consisting of two or more independent attributes
- **Composite key**: a key (or part of a key) consisting of an independent attribute plus a dependent attribute – composite keys cannot be separated for part-key dependency checking in moving from 1NF to 2NF
- **Foreign key**: a key which is the primary key of one relation and non-key attribute in another relation

Keys ‘prove’ the relationships

Unnormalised Form (UNF)

- List all identifiable attributes from the given sample and order into a list (a single column)
- Attributes which are multivalued or are in repeating groups can be shown
  - Indented
  - (or in parentheses)
- A multivalued attribute is any for which there is more than one **value** for a given attribute **name**

  e.g. Product Code on an Order
- Choose an initial key

  e.g. Order Number for an Order

Order Document

Order Number : 17835

Depot Number : 5
Depot Name : Plymouth

Customer Number : 946
Martin Engineering

Parkway Industrial Estate
Bodmin
Cornwall

Telephone : 01208 76543

Product No.	Description	Quantity	Price Each
658-1400	ZX Pump	2	85.50
3658	22mm Pipe	10	12.75
89023-3	Radiator	5	126.00
732911	Thermostat	1	56.40

Assumptions

Order No, *Customer No*, *Depot No* and *Product No* are unique

Price Each depends on *Product No*

Product No can only be used for one item on the same order

N.B. the Depot is the place where the order is processed

UNF of Order Document

<u>Order No.</u>
Customer No.
Customer Name
Customer Address
Customer Tel No.
Depot No.
Depot Name
(Product No.
Product Name
Product Qty
Product Price)

First Normal Form (1NF)

- Separate out multivalued attributes
  - (i.e. repeating groups)
- Choose a **primary** key for the new relation
  - and propagate the key of the relation from which the new relation has been filtered

    i.e. copy the key of the originating relation as the first part of what will now be either a compound or composite key
- Any remaining attributes in the Unnormalised column are copied across using the initial key as the **primary** key of this relation

UNF to 1NF

UNF

Order No.
Customer No.
Customer Name
Customer Address
Customer Tel No.
Depot No.
Depot Name
(Product No.
Product Name
Product Qty
Product Price)

1NF

Order No.
Customer No.
Customer Name
Customer Address
Customer Tel No.
Depot No.
Depot Name

Order No.
Product No.
Product Name
Product Qty
Product Price

Second Normal Form (2NF)

- Separate out attributes which are dependent on only **part** of a key
  - These are called part-key dependencies

- New relation will contain the part of the key from the original relation plus dependent attributes

- Part key is also retained in the original relation
- Any relation with a key that is just a single attribute (a simple key)
  - is already in 2NF and should be copied across

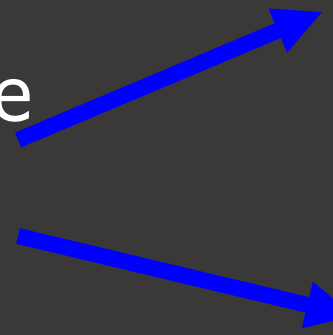
1NF to 2NF Changes

- Product name and price (which is the same for any order) are dependent on product number

  i.e. given product no. we can determine product name and price regardless of the customer order number.

1NF

Order No.
Product No.
Product Name
Product Qty
Product Price



2NF

Order No.
Product No.
Product Qty

Product No.
Product Name
Product Price

Complete 2NF of Document

Order No.	Order No.
Customer Number	Product No.
Customer Name	Product Qty
Customer Address	
Customer Tel No.	
Depot No.	Product No.
Depot Name	Product Name
	Product Price

Third Normal Form (3NF)

- Separate out attributes which are dependent on non-key attributes
  - ‘transitive’ dependencies

- Key & non-key attributes are treated as distinct ‘groups’ within each relation
  - checks are made for determinacy within these

Third Normal Form (3NF)

For the ‘non-key’ (group of) attribute(s):

- Identify & remove to a new relation any attributes which are dependent on any other attribute(s)
- The new relation will contain the new key attribute(s) and its dependent attribute(s)
- The new key will also remain in the original relation as a non-key attribute
  - marked with an asterisk as a foreign key
- Relations which are unchanged after 3NF inspection should be copied across as they are

Third Normal Form (3NF)

For the ‘key’ (group of) attribute(s):

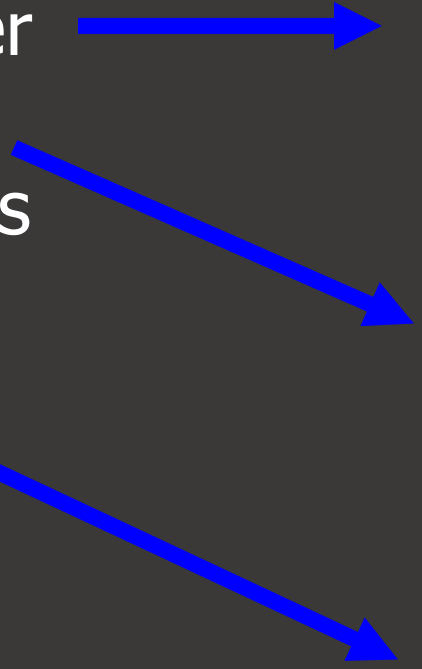
- Examine the attributes to see if any of them (singly or in combination) are a determinant of any given attribute(s) in the key
  - If they are, demote the dependent attribute(s) to non-key status
  - Check demoted attributes to see if they are ‘foreign’ keys - mark them with an asterisk

- Note that simple keys are already in 3NF

2NF to 3NF Changes

2NF

Order No.
Customer Number
Customer Name
Customer Address
Customer Tel No.
Depot No.
Depot Name



3NF

Order No.
*Customer No.
*Depot No.

Customer No.
Customer Name
Customer Address
Customer Tel No.

Depot No.
Depot Name

Complete 3NF of Document

Order No.
*Customer No.
*Depot No.

CustomerNo.
Customer Name
Customer Address
Customer Tel No.

Depot No.
Depot Name

Order No.
Product No.
Product Qty

Product No.
Product Name
Product Price

Normalisation - summary

UNF	1NF	2NF	3NF
Order No. Customer No. Customer Name Customer Address Customer Tel No. Depot No. Depot Name Product No. Product Name Product Qty Product Price	Order No. Customer No. Customer Name Customer Address Customer Tel No. Depot No. Depot Name Order No. Product No. Product Name Product Qty Product Price	Order No. Customer No. Customer Name Customer Address Customer Tel No. Depot No. Depot Name Order No. Product No. Product Qty Product No. Product Name Product Price	Order No. *Customer No. *Depot No. Customer No. Customer Name Customer Address Customer Tel No. Depot No. Depot Name Order No. Product No. Product Qty Product No. Product Name Product Price

Relations in 3NF are Data Entities

- Results of RDA are true to the data currently in the system
  - should be preferred to the results of general analysis
- RDA gives no information regarding optionality or relationship link phrases
  - these should be retained or added
- Normalisation alone is unlikely to produce a good data model
- Normalisation does identify many problems and clarifies the attributes

Converting 3NF to an EM

- By 3NF all relationships between entities (relations) are one-to-many
- A set of 3NF relations can be expressed as a partial LDS or Entity Model
- The ‘master’ entity is “one” end of the relationship
- The ‘detail’ is the “many” end

Normalisation & Relationships

- The attributes used for keys indicate the *relationships* between entities

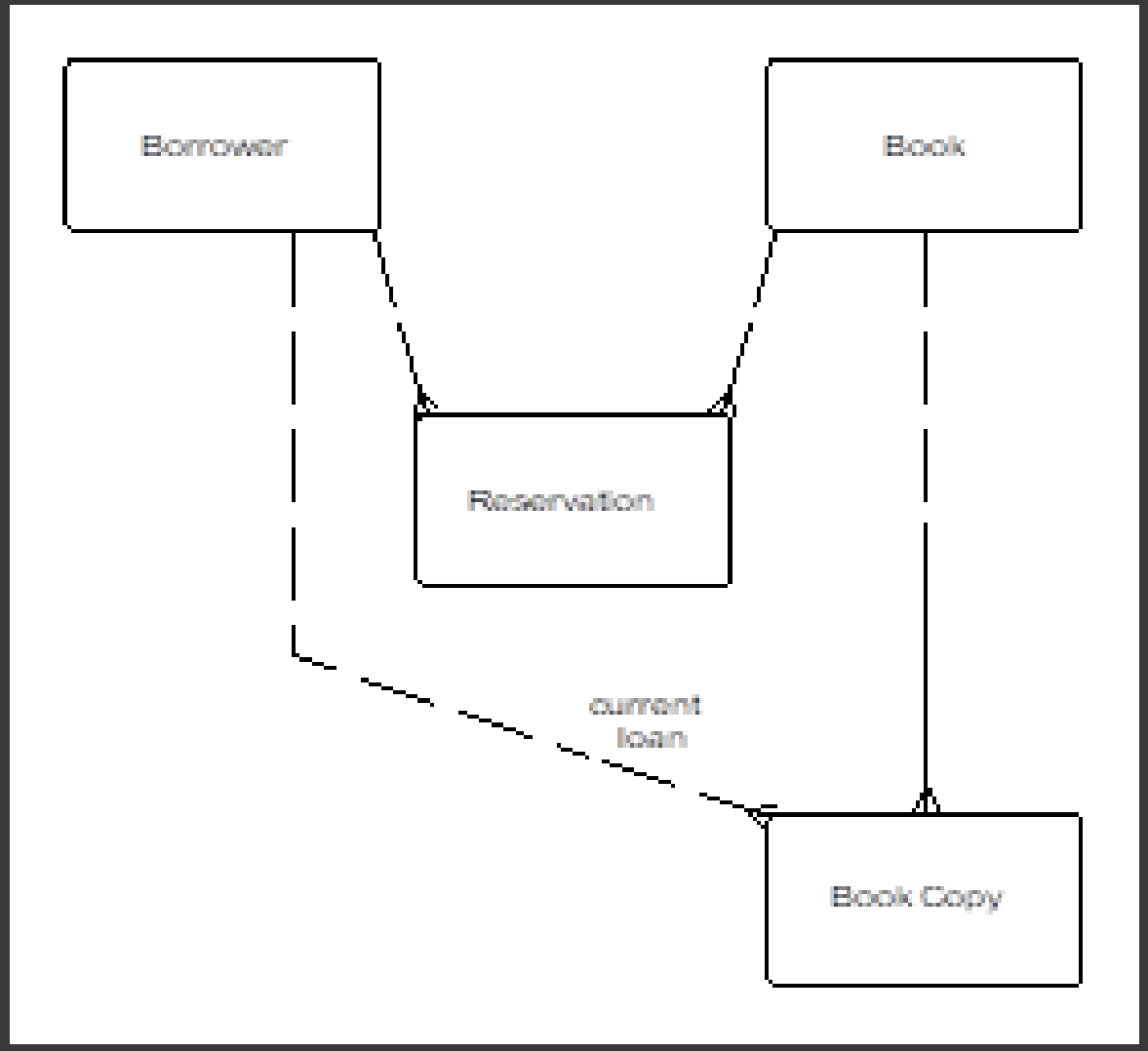
  e.g. Customer entity is related to the Order entity

- A relationship is based on one or more attributes of the keys being present in the related entities
- By Third Normal Form, all relationships are one-to-many
  - outputs from normalisation can be applied directly to an LDS/ERD
- Relationship Link Phrases may need to be added

Merging diagrams

- Merge Data Models from different sources
- Merge with E-R Diagram

Library Example



Borrower & Current Loans

Borrower ID No. 45632	**Address** 10 New St. Crownhill Plymouth
Name George Jones	

Current Loans

Issue date	Return due	No. of renewals	Accession No.	ISBN	Title
3/12/04	17/12/04	0	678956	0077099745	Software System Developn
3/12/04	17/12/04	0	815642	0077095855	Database Design
22/11/04	20/12/04	1	823124	0201708574	Database Systems

Borrower & Current Loans

UNF	
Borrower ID No.	
Name	
Address	
(Issue date	
Return due	
No. of renewals	
Accession No.	
ISBN	
Title)	

Borrower & Current Loans

UNF	1NF
Borrower ID No.	**Borrower ID No.**
Name	Name
Address	Address
(Issue date	
Return due	**Borrower ID No.**
No. of renewals	**Accession No.**
Accession No.	Issue date
ISBN	Return due
Title)	No. of renewals
	ISBN
	Title

Borrower & Current Loans

UNF	1NF	2NF
Borrower ID No.	**Borrower ID No.**	**Borrower ID No.**
Name	Name	Name
Address	Address	Address
(Issue date		
Return due	**Borrower ID No.**	**Borrower ID No.**
No. of renewals	**Accession No.**	**Accession No.**
Accession No.	Issue date	
ISBN	Return due	**Accession No**
Title)	No. of renewals	Issue date
	ISBN	Return due
	Title	No. of renewals
		ISBN
		Title

Borrower & Current Loans

UNF	1NF	2NF	3NF
Borrower ID No.	**Borrower ID No.**	**Borrower ID No.**	**Borrower ID No.**
Name	Name	Name	Name
Address	Address	Address	Address
(Issue date			
Return due	**Borrower ID No.**	**Borrower ID No.**	**Accession No.**
No. of renewals	**Accession No.**	**Accession No.**	*Borrower ID No.
Accession No.	Issue date		
ISBN	Return due	**Accession No**	**Accession No**
Title)	No. of renewals	Issue date	Issue date
	ISBN	Return due	Return due
	Title	No. of renewals	No. of renewals
		ISBN	*ISBN
		Title	
			ISBN
			Title

Borrower & Current Loans

UNF	1NF	2NF	3NF	Optimised 3NF
Borrower ID No.	**Borrower ID No.**	**Borrower ID No.**	**Borrower ID No.**	**Borrower ID No.**
Name	Name	Name	Name	Name
Address	Address	Address	Address	Address
(Issue date				
Return due	**Borrower ID No.**	**Borrower ID No.**	**Accession No.**	**Accession No.**
No. of renewals	**Accession No.**	**Accession No.**	*Borrower ID No.	*Borrower ID No.
Accession No.	Issue date			Issue date
ISBN	Return due	**Accession No**	**Accession No**	Return due
Title)	No. of renewals	Issue date	Issue date	No. of renewals
	ISBN	Return due	Return due	*ISBN
	Title	No. of renewals	No. of renewals	
		ISBN	*ISBN	**ISBN**
		Title		Title
			ISBN	
			Title	

Understanding the data

Normalisation gives us a set of rules for determining whether we have grouped our data items correctly

BUT

cannot, in itself, ensure that we have captured all the data the system may require, or that we have fully understood it

Only looks at **part** of the system, not all of the data

Summary

Introduction to the key principles of Normalisation

- Introduced the concepts of normalisation
- Showed how to carry out normalisation


Any Questions?