Informatics 1: Data & Analysis

Lecture 2: Entities and Relationships

Ian Stark

School of Informatics
The University of Edinburgh

Friday 17 January 2014
Semester 2 Week 1

http://www.inf.ed.ac.uk/teaching/courses/inf1/da

Please Complete on Sheet Provided

Anonymous Survey

Q1: How many hours do you estimate you slept last night?

Q2: About how many hours of physical exercise do you usually do in a week?

Q3: Elaborate question about operating systems.

My Z is:

Elaborate Question About Operating Systems

This question is about what students use to carry out coursework and connect remotely to Informatics accounts. If you routinely use multiple devices for this, running multiple different operating systems, then please choose the one which you use most often.

Which operating system do you expect to be using while working on this course when away from the computer labs?

L Linux (Mint, Ubuntu, Fedora, Debian, SUSE, . . .)

C ChromeOS (Chrome, ChromeBox, ChromeBase, . . .)

W Microsoft Windows

X OS X

Z Something else (BSD, AmigaOS, z/OS, Difference Engine,...)

N None — I expect to do all my work on lab DICE machines.

If Z, then please write a more specific answer in the space provided.

Yes, I know that ChromeOS is a Linux variant; in this case I want to distinguish it.

Tutorials and Exercise Sheets

Tutorials start in week 3 of semester and and continue each week until the end of semester, except for Innovative Learning Week which falls between lecturing weeks 5 and 6.

Your *course tutor* leads your tutorial group; this is not the same as your *personal tutor*. Please check your group, when and where it meets:

http://www.inf.ed.ac.uk/teaching/courses/inf1/da/#tutorials

If you wish to move to a different tutorial group, ask the ITO:

http://www.inf.ed.ac.uk/teaching/contact

All students studying Inf1-DA are required to attend the weekly tutorials. If you are ill or otherwise unable to attend one week then email your course tutor, and if possible attend another tutorial group in the same week.

Coursework, Feedback, and Examination

Information about tutorial exercises, written assignments, assessment, feedback and the final exam is all available on the course web page:

- http://www.inf.ed.ac.uk/teaching/courses/inf1/da/#coursework
- http://www.inf.ed.ac.uk/teaching/courses/inf1/da/#feedback
- http://www.inf.ed.ac.uk/teaching/courses/inf1/da/#exam

Homework

Read Sections 2.1–2.4 of the handout, which is from this textbook on databases.

R. Ramakrishnan and J. Gehrke.
Database Management Systems.
McGraw-Hill, third edition, 2003.

This is the recommended textbook for the third-year course Database Systems. It's a large book, with thorough and extensive material on a wide range of database topics.

It is *not* necessary to buy this book for Inf1-DA. However, I recommend looking at the copies in the University Library HUB to read more.

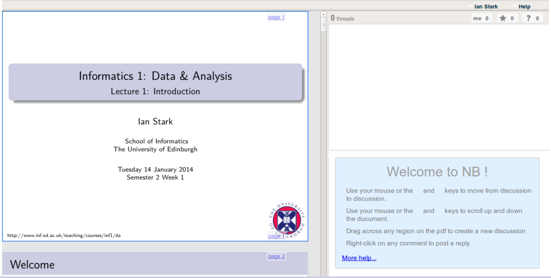
Nota Bene: Online Collaborative Annotation

MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) is developing a tool for collaborative document annotation called **NB**.

We are trialling this in Inf1-DA; it has already been used with some success in other Informatics courses at Edinburgh.

I shall put all lecture slides and tutorial exercise sheets onto the **NB** server: you are invited to add comments, questions, answers and discussions.

Both I and Areti as course TA will follow submissions to NB and respond to questions and comments. Your course tutors may also do this.



Informatics 1: Data & Analysis
Lecture 1: Introduction

Ian Stark

School of Informatics
The University of Edinburgh

Tuesday 14 January 2014
Semester 2 Week 1

http://www.inf.ed.ac.uk/teaching/courses/inf1/da

Welcome

Welcome to NB !

Use your mouse or the and keys to move from discussion to discussion.

Use your mouse or the and keys to scroll up and down the document.

Drag across any region on the pdf to create a new discussion

Right-click on any comment to post a reply

Getting started with NB

To use NB, follow links in the "Slides" section of the Inf1-DA web page.

- To view a particular slide set, click the NB link by the lecture title.
- To add a new comment, click and drag to outline part of a slide.
- You can view and respond to other comments and discussions.
- By default you will be viewing and commenting as "Guest".
- To create an account:
  - If you are already viewing slides in NB as a guest, select the "Guest" drop-down menu in the top right-hand corner and click "Register".
  - Otherwise, click the link "Create an NB account and subscribe to Inf1-DA" on the course web page.
- You can add comments under your own name or anonymously.

Find out more about NB at the project page http://nb.mit.edu/welcome

Visibility and Privacy in NB

I have chosen to put the Inf1-DA materials openly on NB: you don't need an account to view or comment on them, and they are visible to the world.

Comments are by default public, and anyone can see them. You can also post individual comments as "staff-only" or "myself-only", and while I expect this to work as described there is no strong guarantee of privacy.

NB is an educational research project, and the research team at MIT have complete access to everything posted on the service.

All material is hosted on servers in Cambridge, Massachusetts, USA.

I expect that most students will create accounts on NB under their own, recognizable name; some may explicitly link these to existing Google or Facebook identities.

However, we do not require this: you can use a pseudonym if you wish, or simply post as Guest. For stronger anonymity guarantees I recommend the Tor Browser Bundle. http://www.torproject.org/torbrowser

Structured Data

Some application domains involve handling quantities of data that can be very strictly organised. For example:

- The University of Edinburgh records some standardized pieces of information about each of several thousand students.
- A supermarket chain will maintain information on tens of thousands of product lines, and the stock in each shop where they are sold.
- A web browser will keep details of passwords and preferences for all websites a user visits.

What's central to this structure is that we are working with the same information about many different individuals. Even when there are different kinds of individual (product lines, shops, staff, ...) there are far more items of each kind than there are different kinds.

Structured Data

Some application domains involve handling quantities of data that can be very strictly organised. For example:

- The University of Edinburgh records some standardized pieces of information about each of several thousand students.
- A supermarket chain will maintain information on tens of thousands of product lines, and the stock in each shop where they are sold.
- A web browser will keep details of passwords and preferences for all websites a user visits.

As well as individuals or entities, it's usually important to also work with the relationship between individuals: which students take which course, or which shop stocks which product.

Structured Data

Some application domains involve handling quantities of data that can be very strictly organised. For example:

- The University of Edinburgh records some standardized pieces of information about each of several thousand students.
- A supermarket chain will maintain information on tens of thousands of product lines, and the stock in each shop where they are sold.
- A web browser will keep details of passwords and preferences for all websites a user visits.

It turns out — perhaps unexpectedly — to be very effective to concentrate more on the relations between things than on the things themselves.

"The fundamental interconnectedness of all things"
Douglas Adams, Dirk Gently's Holistic Detective Agency

Lecture Plan for Weeks 1–4

Data Representation

This first course section starts by presenting two common data representation models.

- The *entity-relationship (ER)* model
- The *relational* model

Note slightly different naming:
-relation**ship** vs. relationa**l**

Data Manipulation

This is followed by some methods for manipulating data in the relational model and using it to extract information.

- *Relational algebra*
- The *tuple-relational calculus*
- The query language *SQL*

Database Design

1. Requirements analysis

   Understand what data is to be stored in the database and what operations on it are likely to be needed.

2. Conceptual design

   Develop a high-level description of data to be stored, and the constraints that apply to it.

   This is the level where we might use an ER data model.

3. Logical design

   Implement the conceptual design by mapping it to a specific data representation. The outcome is a *logical schema.*

   For example, implementation can be performed by translating the ER data model into a relational data model.

The ER Data Model

- What is it?

  The ER model is a way to organise description of *entities* (individual things in the real world) and the *relationships* between them.

- Why is it useful?

  It readily maps into different *logical data models*, such as the relational model

- How is it used?

  As a graphical notation for visualising the structure of data, to clarify and communicate that structure.

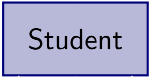
P. P. Chen
The Entity-Relationship Model – Toward a Unified View of Data.
ACM Transactions on Database Systems 1(1):9–36.

Entities and Entity Sets

Any individual object (for example, in the real world) can be an *entity*.

A collection of the same sort of entities is an *entity set*.

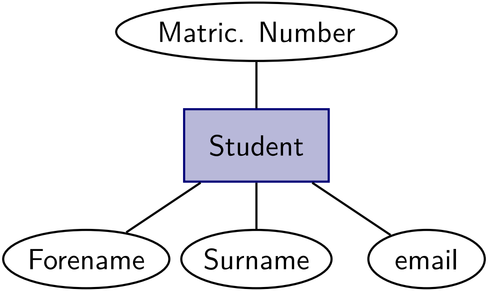
An *ER diagram* shows entity sets as boxes, labelled with the kind of entity belonging to that set.



Attributes

Each entity from the same entity set has characteristic *attributes*.

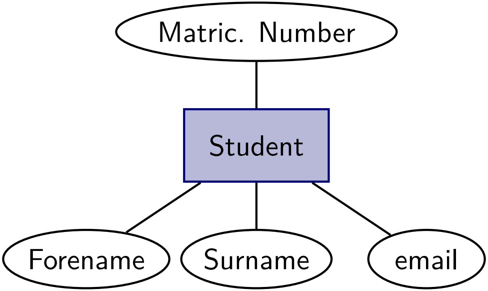
An ER diagram shows attributes as ovals, labelled with the attribute's name, connected to the appropriate entity set.



Domains

Each attribute has a *domain* of allowed values, similar to the use of types in Haskell or Java.

For example, Matric. Number has domain "positive integer", while the domain for email might be "strings of up to 254 characters".

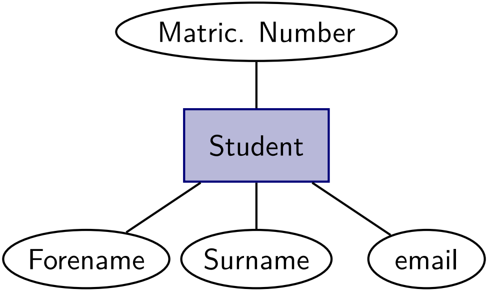


Keys

A *key* is a minimal set of attributes whose values uniquely identify each entity in an entity set.

Where there is more than one such set, they are called *candidate keys*.

Here both {Matric. Number} and {email} are candidate keys.

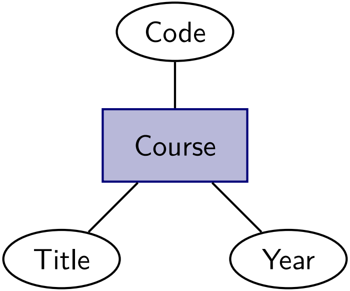


Keys

A *key* is a minimal set of attributes whose values uniquely identify each entity in an entity set.

Where there is more than one such set, they are called *candidate keys*.

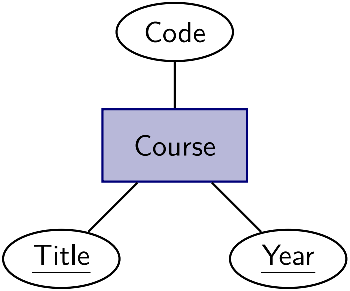
In this case {Title, Year} and {Code, Year} are both candidate keys.



Primary Keys

If there are several candidate keys, we choose just one and make it the *primary key*.

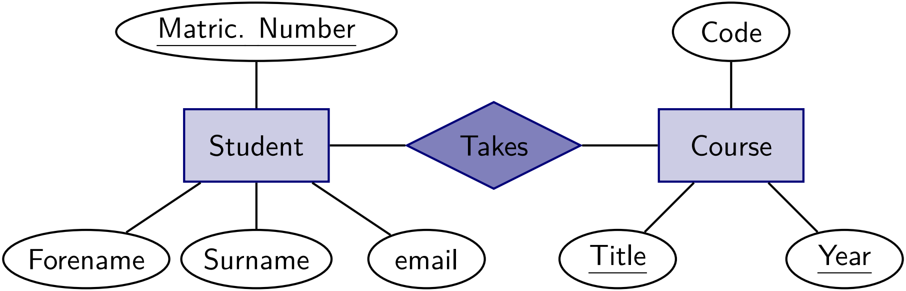
The attributes occurring in the primary key are underlined in the ER diagram. If there are several then they form a *composite key*.



Relationships and Relationship Sets

A *relationship* is an association between individual entities from particular entity sets.

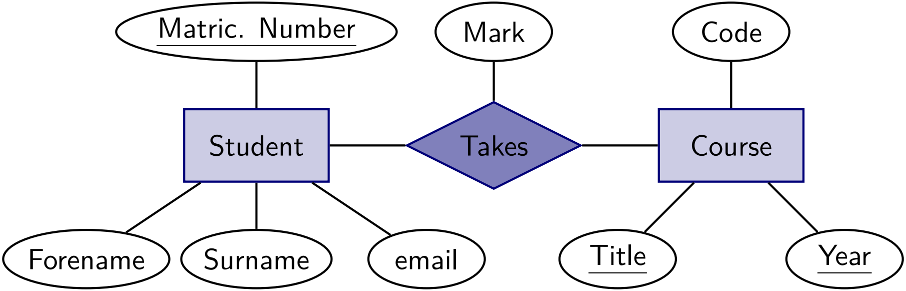
Relationships are grouped into *relationship sets*.



Relationships and Relationship Sets

ER diagrams show relationship sets as diamonds, labelled with the name of the relationship set and connected to all the participating entity sets.

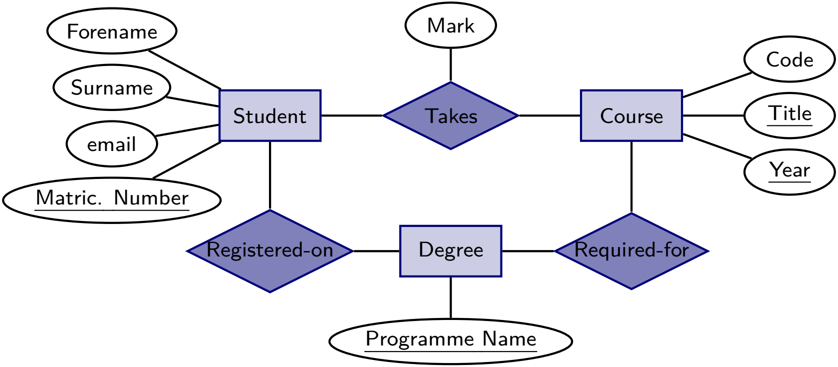
Relationships may have attributes of their own.



Relationships and Relationship Sets

There is no bound on the number of entities participating in a relationship.

An entity may be involved in any number of different relationships.



Instances

An *entity instance* is the set of attribute values for a particular entity in an entity set: these values *instantiate* the attributes.

In the same way, a collection of attributes defines a *relationship instance*.

Examples

An entity instance from the Students entity set:

(123, “Winston”, “wsmith@example.org”)

An entity instance from the Courses entity set:

(INFR08015, “Informatics 1: Data & Analysis”, 2013)

A relationship instance from the Takes relationship set:

(123, “Winston”, “wsmith@example.org”,
INFR08015, “Informatics 1: Data & Analysis”, 2013, 88)