# Informatics 1: Data & Analysis

## Lecture 2: Entities and Relationships

Ian Stark

School of Informatics
The University of Edinburgh

Friday 18 January 2013
Semester 2 Week 1

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

# 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

## Data Representation

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

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

## 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 the 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

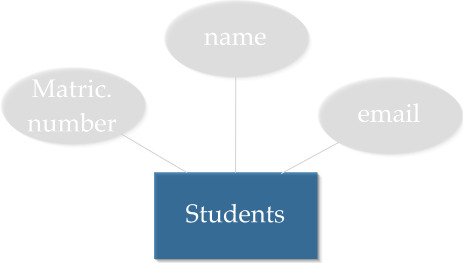
- How is it used?
  As a graphical notation for visualising the structure of data, to clarify and communicate that structure.

# 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*.

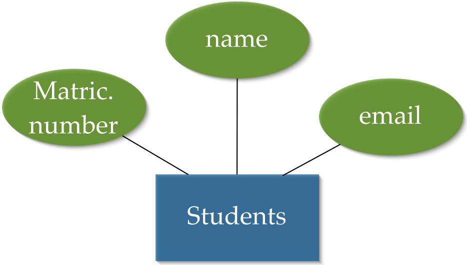
An *ER diagram* shows entity sets as boxes, labelled with the name of the entity set (not the name of an individual entity).



# Attributes

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

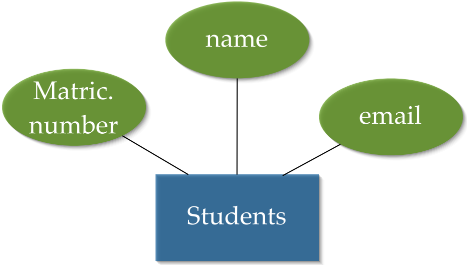
An ER diagram shows attributes as ovals, labelled with the attribute's name, connected to the appropriate entity set.



# 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 name and email might be “strings of up to 64 characters”.

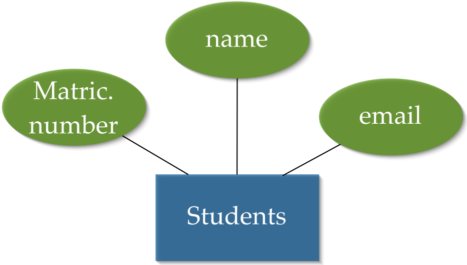


# Keys

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*.

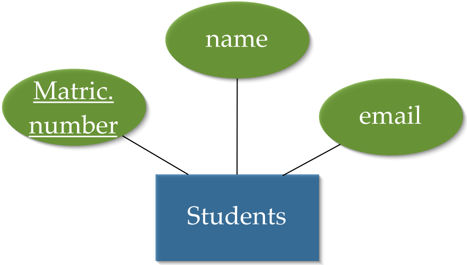
Here both {Matric. number} and {email} are candidate keys.



# Primary Keys

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

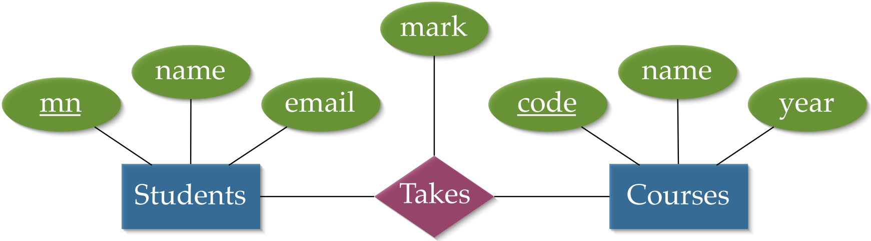
The attributes occurring in the primary key are underlined in the ER diagram. If there are several then they form a *composite key*.



# Relationships and Relationship Sets

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

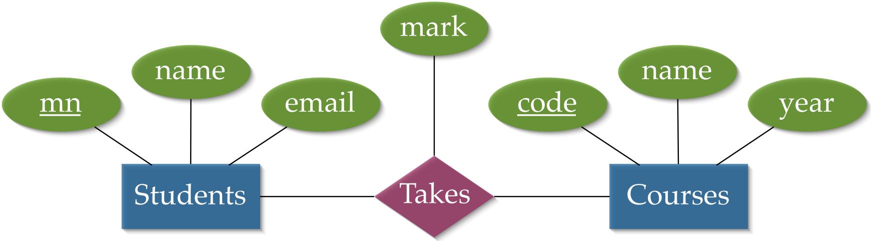
Relationships are grouped into *relationship sets*.



# 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.

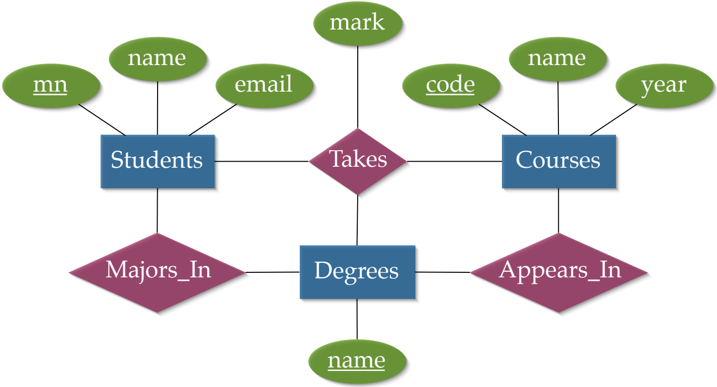
Relationships may have attributes of their own.



# 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:

( 123, "Informatics 1: Data & Analysis", 2012 )

A relationship instance from the Takes relationship set:

( 123, "Winston", "wsmith@example.org",
08015, "Informatics 1: Data & Analysis", 2012, 88 )

# 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 Database Systems course. It's a large book, with thorough and extensive material on a wide range of database topics.

It is *not* necessary to buy or read this book for Inf1-DA. However, you might be interested to look at the copies in the University Library.