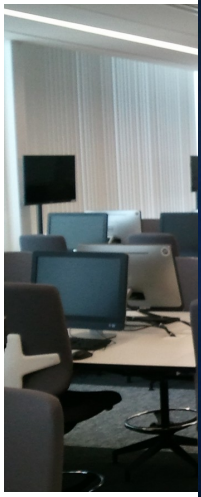
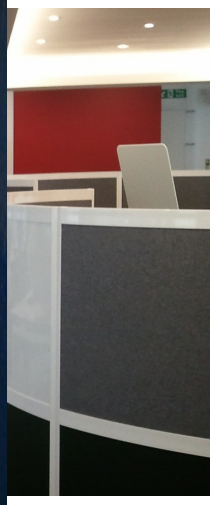
Information Management (L4)
ER Diagrams & Relations

CS1F

Dr Craig Macdonald

Firstly, some announcements…





Why be a class rep?

1. improve your study experience
2. get free advocacy training from SRC
3. gain credit on your CV and uni transcript
4. influence policy at the School of Computing Science

How to be a class rep:

1. Email Helen.Border@glasgow.ac.uk and volunteer – tell her your CompSci classes and lab group numbers
2. If you are selected…
   - Attend SRC training session
   - Solicit feedback from your class peers
   - Attend two staff/student meetings (one per semester) and present feedback to the School of Comp Sci

Database design lifecycle

5

- Requirements analysis
  - User needs; what must database do?
- Conceptual design
  - High-level description; often using E/R model
- Logical design
  - Translate E/R model into (typically) relational schema
- Schema refinement
  - Check schema for redundancies and anomalies
- Physical design/tuning
  - Consider typical workloads, and further optimise

Conceptual Design

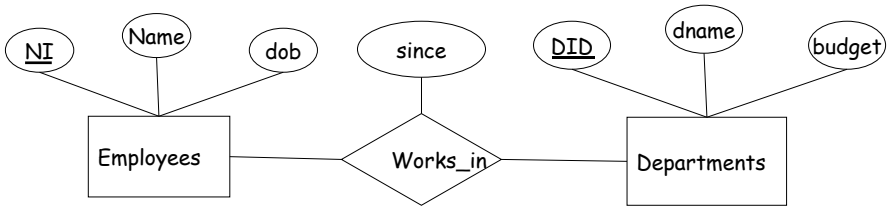
6

- What are the *entities* and *relationships* in the enterprise?
- What information about these entities and relationships should we store in the database?
- What are the integrity constraints (business rules) that hold?
- We represent this information pictorially in E/R diagrams (and then map these to a relational schema later).

Recap: ER

7

- Entities are real world objects
  - Entity Types: definitions for real-world objects, with attributes, including key attributes
- Relationships between entities, modelled as relationship types
  - Uniquely identified by entities, but can have attributes. Usually binary, can be N-ary



Relationship Types

- Captures how two or more entities are related
- Can be thought of as verbs, linking two or more nouns
- Examples:
  - an *owns* relationship between a company and a computer
  - a *supervises* relationship between an employee and a department
  - a *performs* relationship between an artist and a song
  - a *proved* relationship between a mathematician and a theorem

Types & Sets?

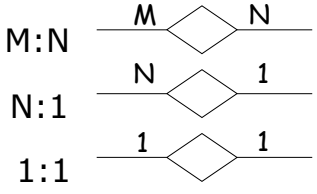
9

- Entity Type:
  - Employees, Departments
- Entity Set of "Employees":
  - {Jane Doe, Jack Willis}
- Relationship Type:
  - Works_in
- Relationship Set of "Works_In":
  - {Jane Doe works_in Accounting,
    Jack Willis works_in IT}

Cardinality Constraints on Relationship Types

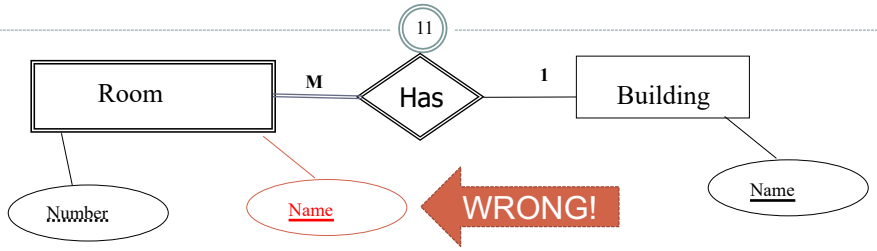
10

- For example:
  - An employee can work in many departments; a department can have many employees
  - In contrast, each department has at most one manager
- The cardinality specifies the number of entity instances that can participate from each side of the relationship of a binary relationship
  - One to one (1:1)
  - One to many (1:N)
  - Many to Many (N:M)



Note: Sometimes this is denoted using different arrowheads

Example of 1 to N cardinality

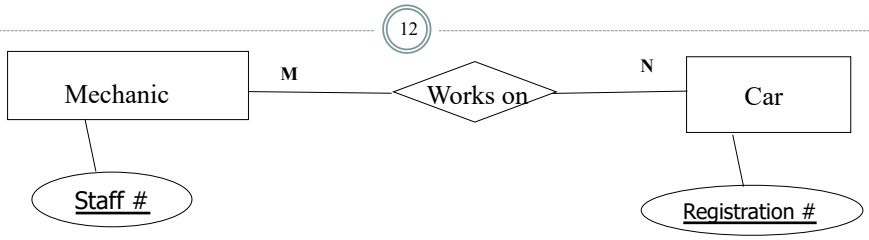


For every ONE room, there is ONE building

For every ONE building, there are ANY NUMBER of rooms

Reminder: We do not "move" (key) attributes across relationships

Example of M to N cardinality



For every ONE car, there are ANY NUMBER of mechanics

For every ONE mechanic, there are ANY NUMBER of cars

Recap: Other ER Notation

13

- Entity Types:
  - Subtypes and Supertypes
    - One entity type can *inherit* properties of another type
    - Subtypes without attributes or relationships are pointless
  - Weak Entities
    - Insufficient attributes to form a primary key
    - Must have a *relationship* with an identifying "strong" entity
- Relationships
  - Total vs partial participation of a relationship
    - MUST vs CAN
  - Recursive Relationships
    - An entity type can be in a relationship with itself
    - For example, one employee manages another
      - Not necessarily the same entity instance – i.e. I am not my own manager!

From a written scenario to an ER Model

14

- Identify the **Entities**, their **Attributes**, and all **Relationships** involved in any given scenario
- Represent this in an Entity-Relationship Diagram
  - Our conceptual model
- ER Diagram (and model) can then be used to design and implement the actual relationship tables in the database itself
  - Our conceptual schema
  - (we will do this in the lab in week 3)

Constructing an ER diagram

15

1. Identify the entity types (in boxes)
2. Identify each entity types' properties
3. Decide which properties are attributes (connected to entity in oval)
4. Decide which attributes could be keys
5. Select primary key (underlined attribute)
6. Determine which properties infer relationships (labelled diamond between the participating entities)
7. Decide on the cardinality and participation of the relationship (numbers at entities involved in relationship; single line Vs double line at entity)

The Movies Scenario

We're going to the movies!

The Movies Scenario

17

Set up a database about movies, stars and studios. Movies have a title, and may be released many times. Releases have a year, length and film type. Stars have a name and address. Studios have a name and address. Movies can be sequels of other movies. All stars have contracts with specific studios with an associated salary, and get a cash bonus for specific releases. Stars act in releases and studios own movies.

1. Identify the entity types

18

Set up a database about movies, stars and studios. Movies have a title, and may be released many times. Releases have a year, length and film type. Stars have a name and address. Studios have a name and address. Movies can be sequels of other movies. All stars have contracts with specific studios with an associated salary, and get a cash bonus for specific releases. Stars act in releases and studios own movies.

Entities

19

Release

Stars

Movies

Studios

Attributes

20

Set up a database about movies, stars and studios. Movies have a title, type and may be released many times. Releases have a year and length. Stars have a name and address. Studios have a name and address. Movies can be sequels of other movies. All stars have contracts with specific studios with an associated salary, and get a cash bonus for specific releases. Stars act in releases and studios own movies.

Movies

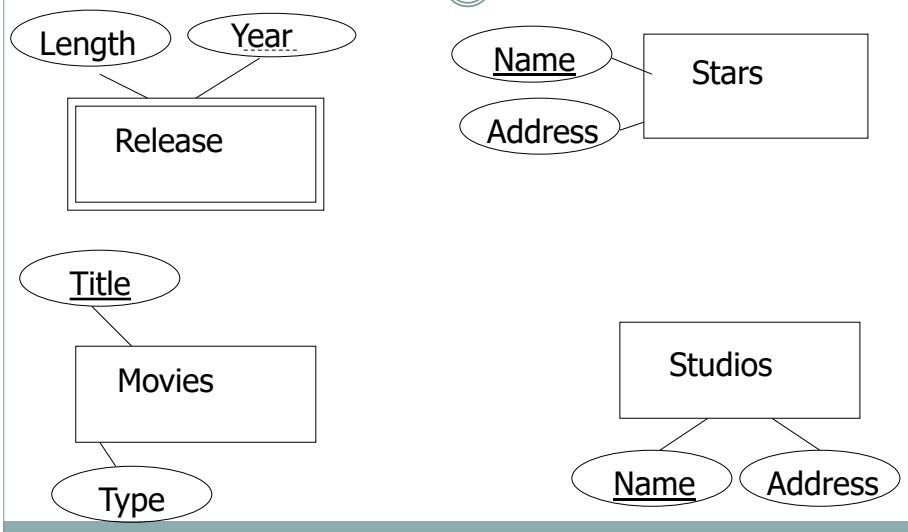
Release

Stars

Studios

Attributes

21



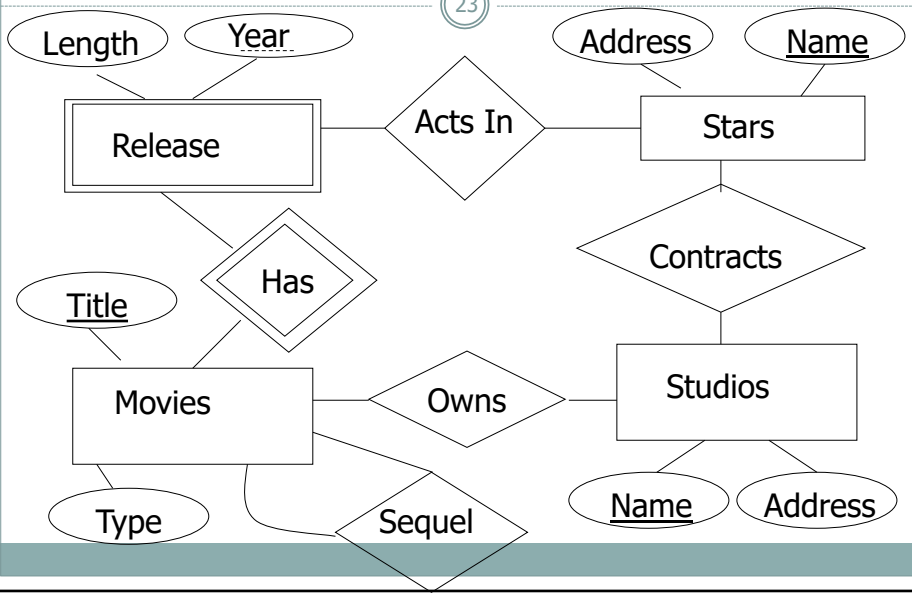
Relationships

22

Set up a database about movies, stars and studios. Movies have a title, and may be released many times. Releases have a year, length and film type. Stars have a name and address. Studios have a name and address. Movies can be sequels of other movies. All stars have contracts with specific studios with an associated salary, and get a cash bonus for specific releases. Stars act in releases and studios own movies.

Relationships

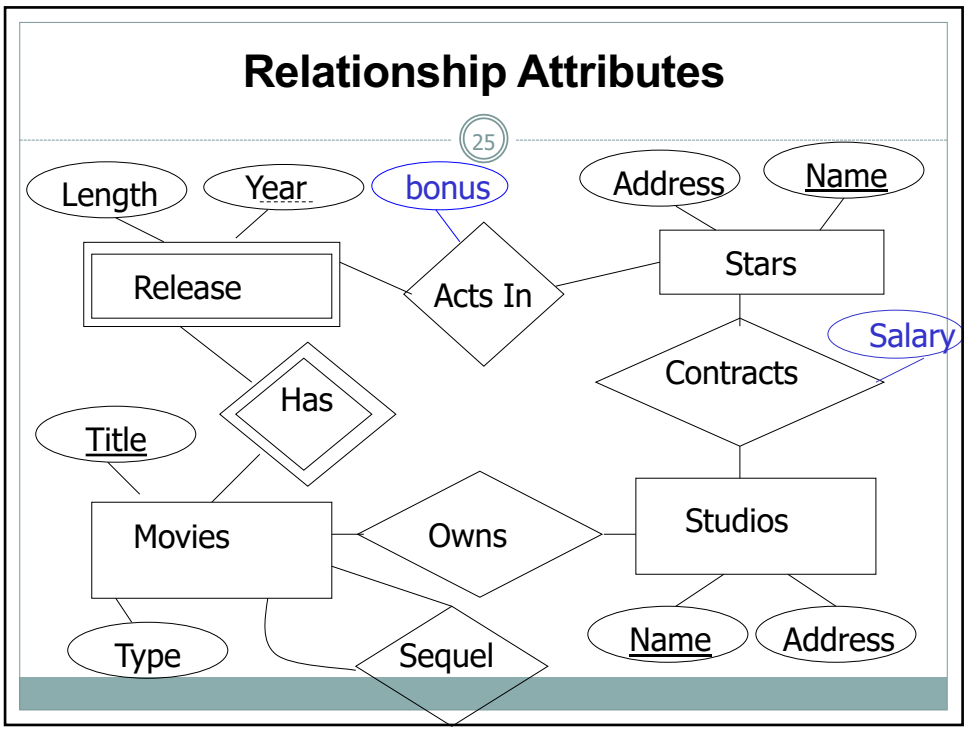
23

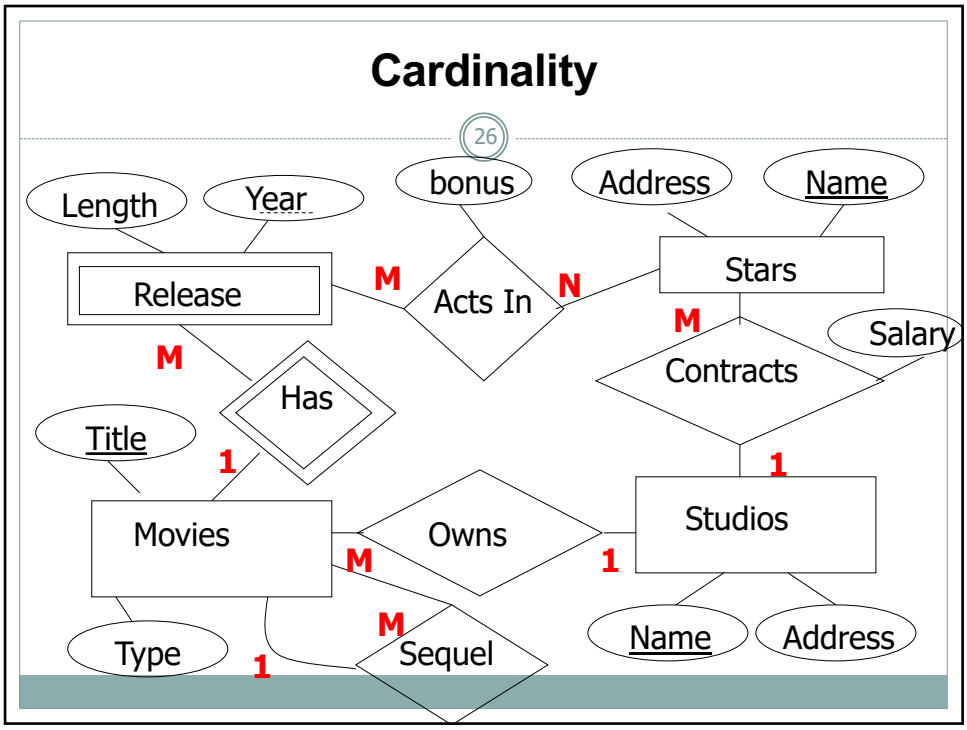


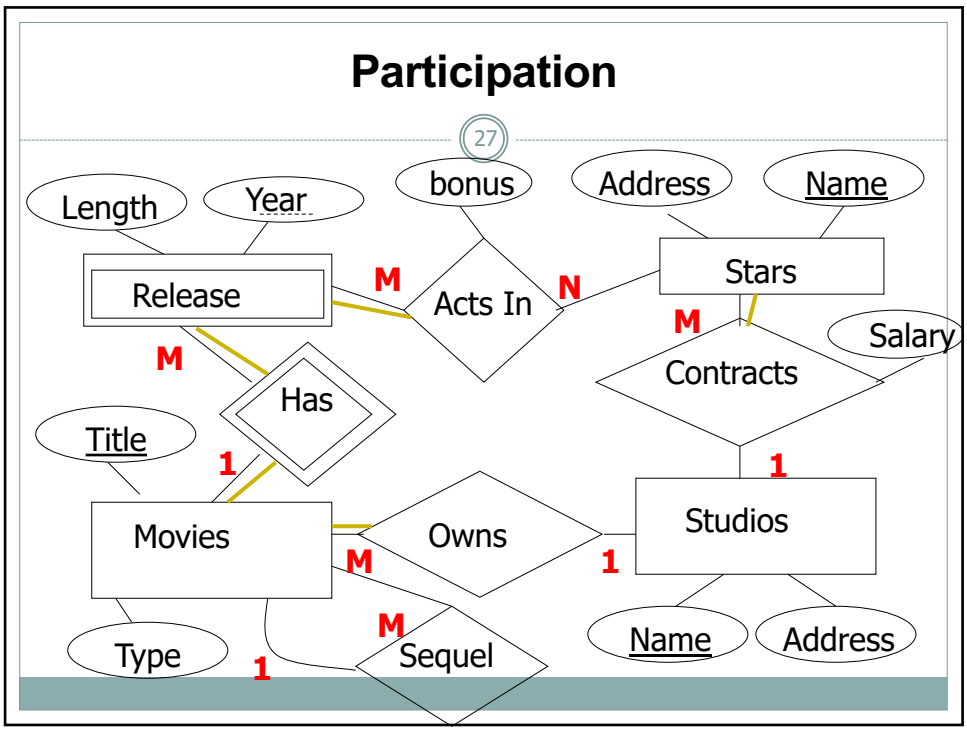
Relationship Attributes

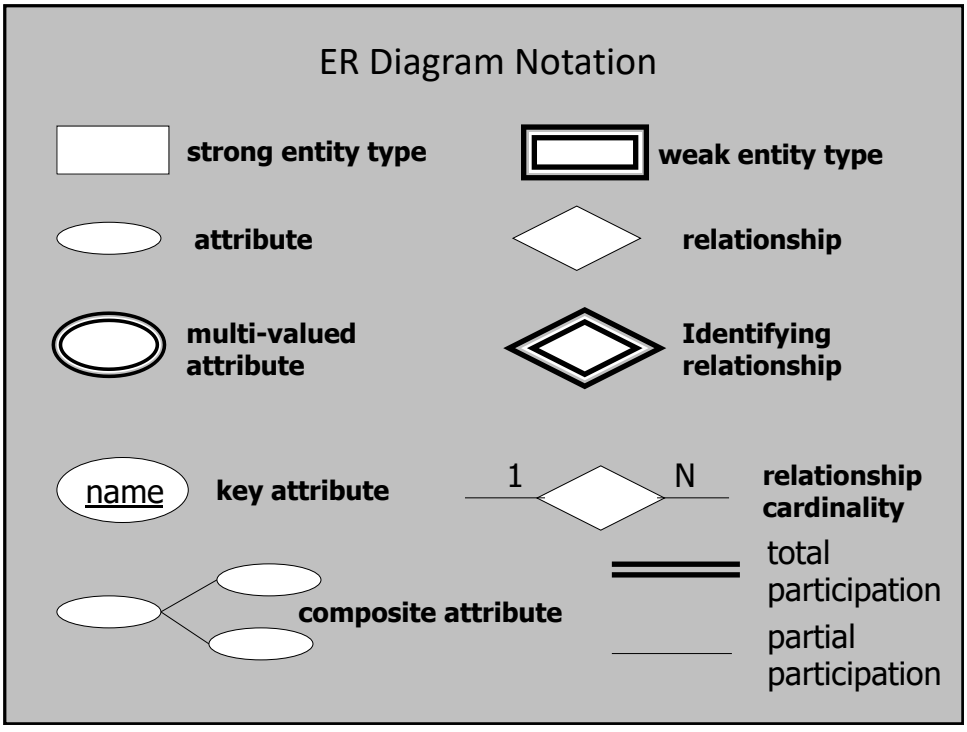
24

Set up a database about movies, stars and studios. Movies have a title, and may be released many times. Releases have a year, length and film type. Stars have a name and address. Studios have a name and address. Movies can be sequels of other movies. All stars have contracts with specific studios with an associated salary, and get a cash bonus for specific releases. Stars act in releases and studios own movies.


Relationship Attributes
25
Length
Year
bonus
Address
Name
Release
Acts In
Stars
Salary
Contracts
Has
Title
Movies
Owns
Studios
Type
Sequel
Name
Address


Cardinality
26
Length
Year
bonus
Address
Name
Release
M
Acts In
N
Stars
M
Salary
Contracts
M
Has
Title
1
1
Movies
Owns
M
1
Studios
M
Type
1
Sequel
Name
Address


Participation
27
Length
Year
bonus
Address
Name
Release
Acts In
Stars
M
N
M
M
Has
Contracts
Salary
Title
1
1
Movies
Owns
Studios
M
1
M
Sequel
Name
Address
Type
1


ER Diagram Notation
strong entity type
weak entity type
attribute
relationship
multi-valued attribute
Identifying relationship
name
key attribute
1
N
relationship cardinality
composite attribute
total participation
partial participation

The Relational Model
From ER model to Tables

CS1F

IM Lecture 5 (an intro)

Craig Macdonald

Database design lifecycle

30

- Requirements analysis
  - User needs; what must database do?
- Conceptual design
  - High-level description; often using E/R model
- Logical design
  - Translate E/R model into (typically) relational schema

- Schema refinement
  - Check schema for redundancies and anomalies
- Physical design/tuning
  - Consider typical workloads, and further optimise

Overview

31

- The Relational Model
- Understanding Entities & Relationships as 'Tables' in a database

- Next:
  - Converting your diagram into tables
  - Enforcing integrity
  - More on the relational model

Reminder - Data Modelling

- ER Model allowed us to establish the relationships and dependencies amongst the information
- We now need to arrange the data into a **logical structure**
- The logical structure can then be mapped into the storage objects supported by the database - for example **tables**
- So from *conceptual design to conceptual schema*

The Relational Data Model

- Introduced by E.F. Codd in 1970
- Most commonly supported form used in s/w industry
- Simple means of representing & manipulating data
- Has a good theoretical/mathematical grounding
  - More on this later (lecture 7 and 8)

Entities → Tables

A table (relation) is constructed for each item of interest in a DB

A relation equates (approximately) to an entity type or *some* relationships in an ER diagram

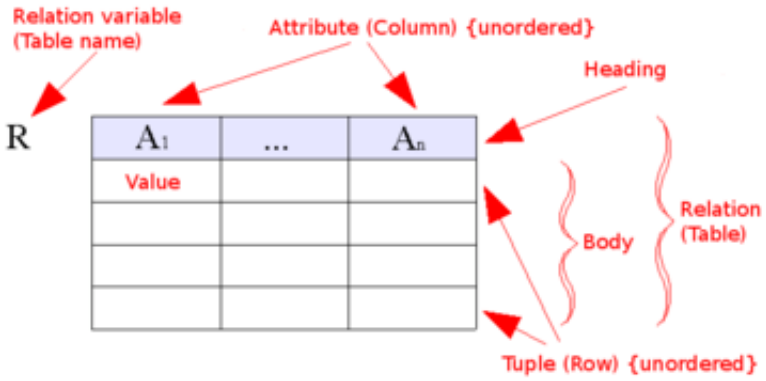
Student:

Student ID#	Firstname	Surname	…
2019582	Joe	Bloggs	
2058292	Jane	Bloggs	
…			

All relations must have a HEADING and a BODY

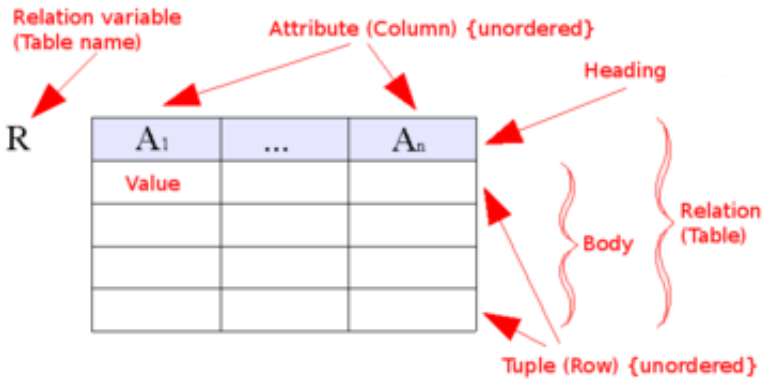
Structure of Data Objects in the Relational Model

- Data is represented in two dimensional TABLES (relations)



Structure of Data Objects in the Relational Model

- Each table has ROWS (tuples) and COLUMNS (attributes)



The Heading

- All relations must have a heading
  - Name of relation
    - Student
  - Names of columns of relation (the attributes)
    - Name, student ID, exam1, exam2

STUDENT (Name, Student ID, exam1, exam2)

The number of attributes determines the DEGREE of the relation

The Body

- The rows of a relation comprise its body
  - These are referred to as TUPLES
- A tuple is an ordered list of values
- The meaning of each value is determined by its position in the tuple
- The number of tuples in a relation determines it's CARDINALITY

Degree and Cardinality of a Relation

STUDENT

name	matric	exam1	exam2
Gedge	891023	12	58
Kerr	892361	66	90
Fraser	880123	50	65

- The relation student has:
  - Degree of 4 (number of attributes/columns)
  - Cardinality of 3 (number of rows/tuples)

GOTCHA: Do not confuse this with the cardinality of a relationSHIP type in an E/R diagram

Relations → Schema

- A **tuple** (record) is a row of a relation, i.e. a set of values which are instances of the attributes
  - < 'Fraser', 880123, 66, 90 >

Relations → Schema

- A **relation schema** is a set of attributes
  - written R (A_1, A_2,. . .A_n) e.g.
  - Student (name: Text, matric: Number, ex1: Number, ex2: Number)
- A **relational database schema** is a set of these relation schemas
- A relational database – relational database schema with **data**

Relational Database Schema

CUSTOMER

ConsID	Name	Address	SSN	TelNr	ShipAddress	Email

PURCHASE

ConsID	MoneySpent	NrItems

CUSTOMER

ConsID	Name	Address	SSN	TelNr	ShipAddress	Email

PURCHASE

ConsID	MoneySpent	NrItems

BUSINESS PURCHASE

ConsID	BusinessMoney

RESIDENT PURCHASE

ConsID	PurchaseTime	ResidentMoney

Summary of a Table

- The STUDENT relation may be thought of as a 2D table

STUDENT

name	Student ID	exam1	exam2
Gedge	891023	12	58
Kerr	892361	66	90
Fraser	880123	50	65

- A relation has
  - a **name - STUDENT**
  - an unchanging set of **columns** which are named and typed
  - a time varying set of **rows**, which are the current set of **records** for the relation

Translating E-R to relational schema

44

1. **Entities and their simple attributes**
2. Weak entities and their simple attributes
3. 1-1 relationships (and their attributes)
4. 1-M relationships (and their attribute)
5. M-N relationships (and their attributes)
6. Composite attributes
7. Multivalued attributes

NEXT WEEK

Reading

45

- Garcia Molina
  - Chapter 7, Section 5 except 5.3 (pages 153-160)

OR

- Chapter 12 of

https://www.tutorialspoint.com/dbms/dbms_tutorial.pdf
(pages 33-36)