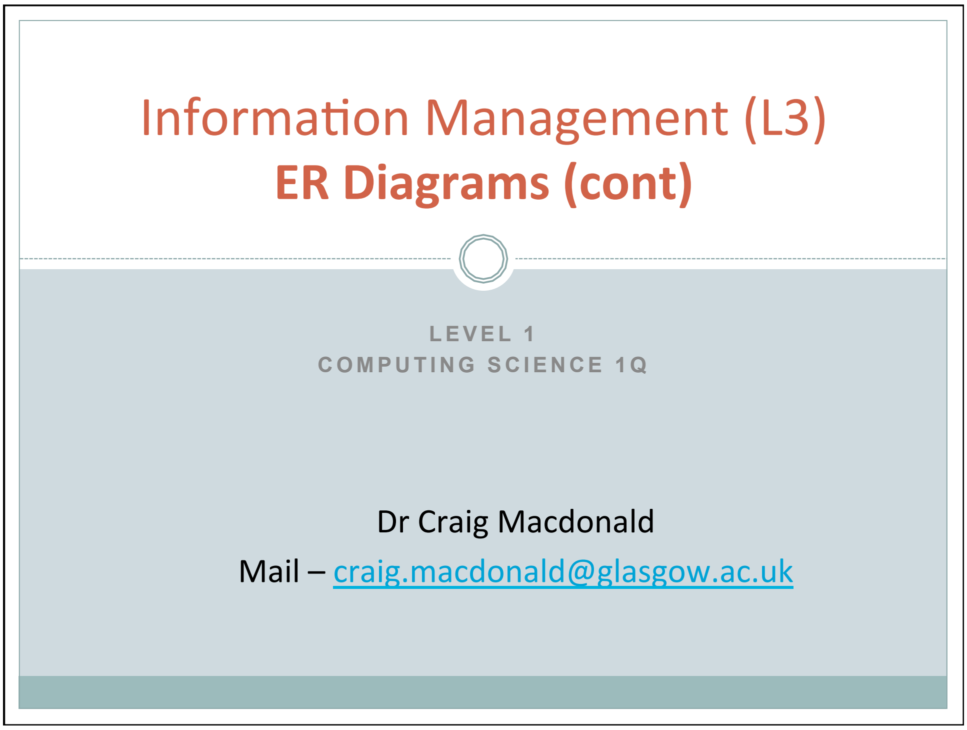
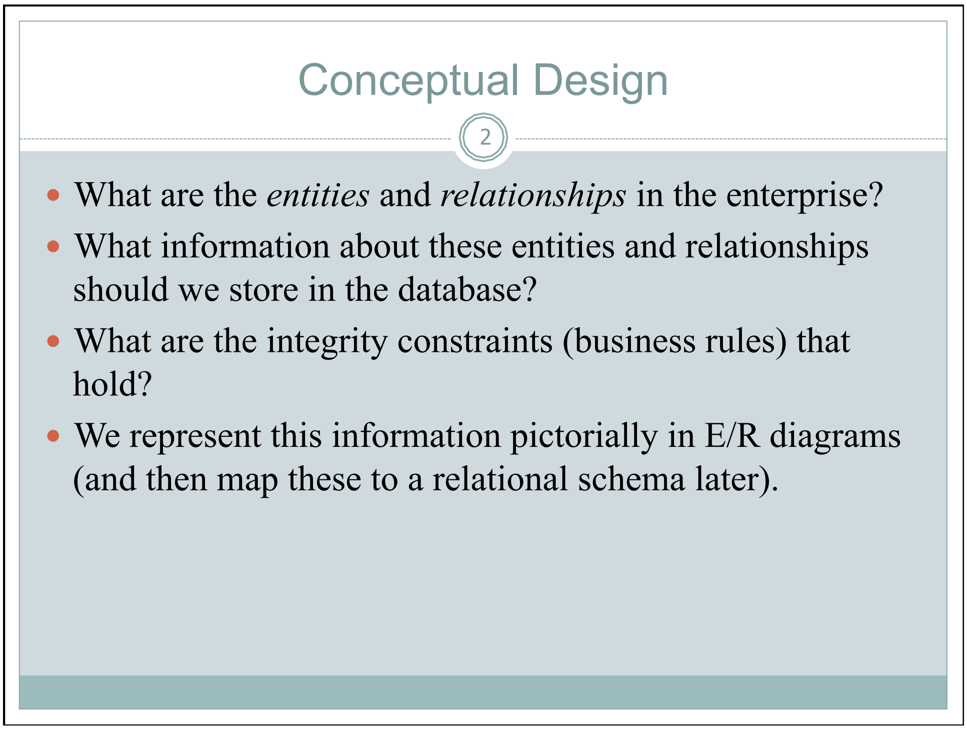


Information Management (L3)
ER Diagrams (cont)

LEVEL 1
COMPUTING SCIENCE 1Q

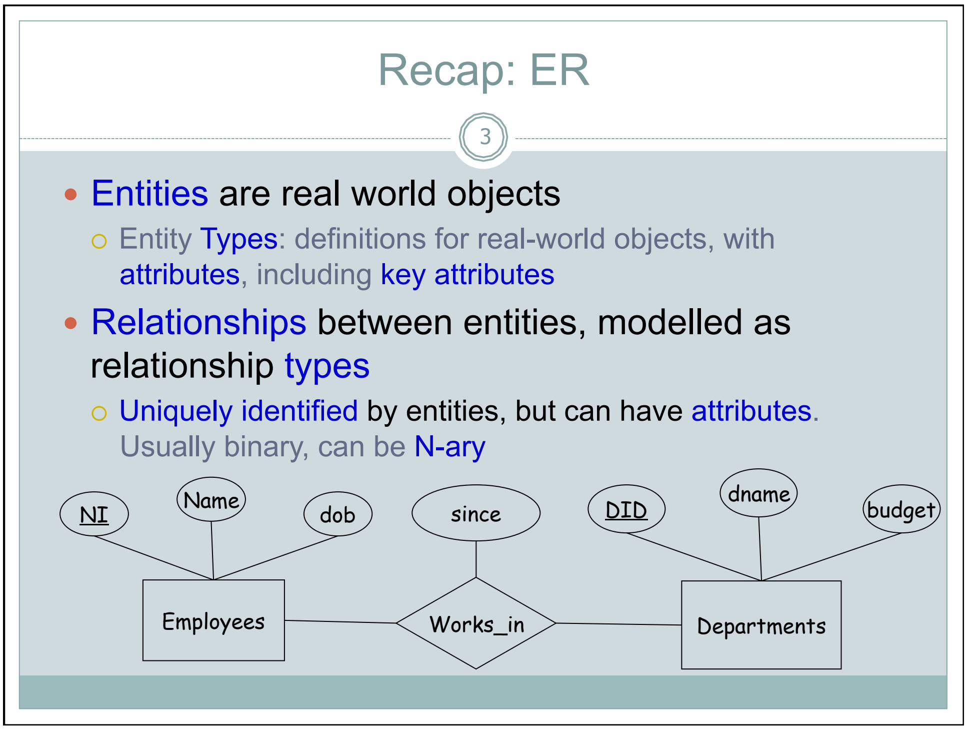
Dr Craig Macdonald
Mail – craig.macdonald@glasgow.ac.uk



Conceptual Design

2

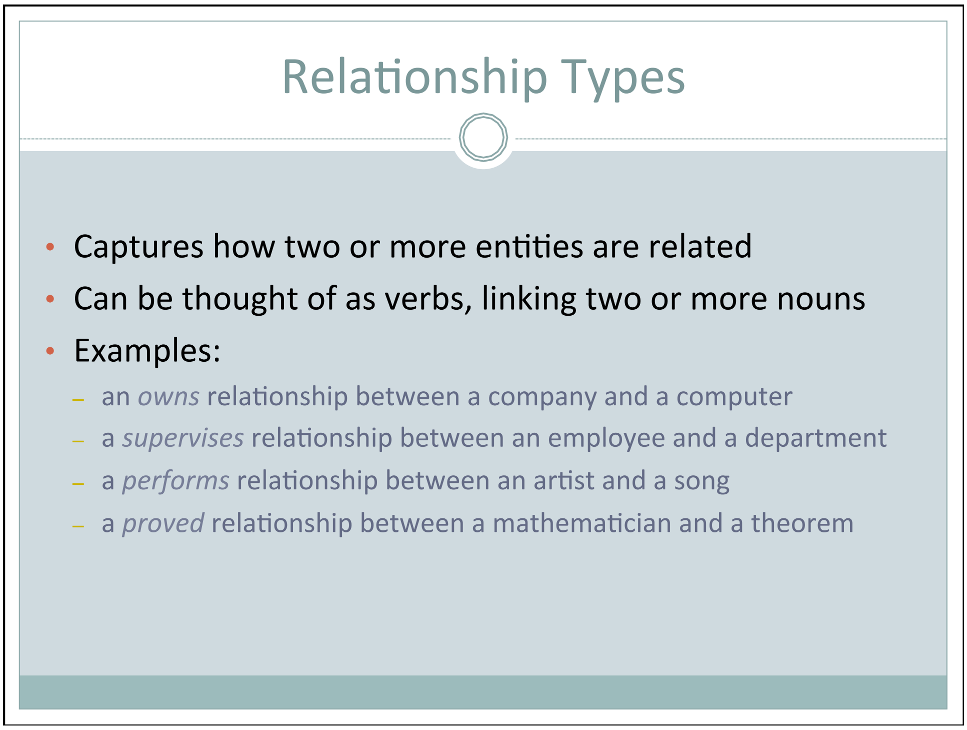
- What are the *entities* and *relationships* in the enterprise?
- What information about these entities and relationships should we store in the database?
- What are the integrity constraints (business rules) that hold?
- We represent this information pictorially in E/R diagrams (and then map these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



Recap: ER

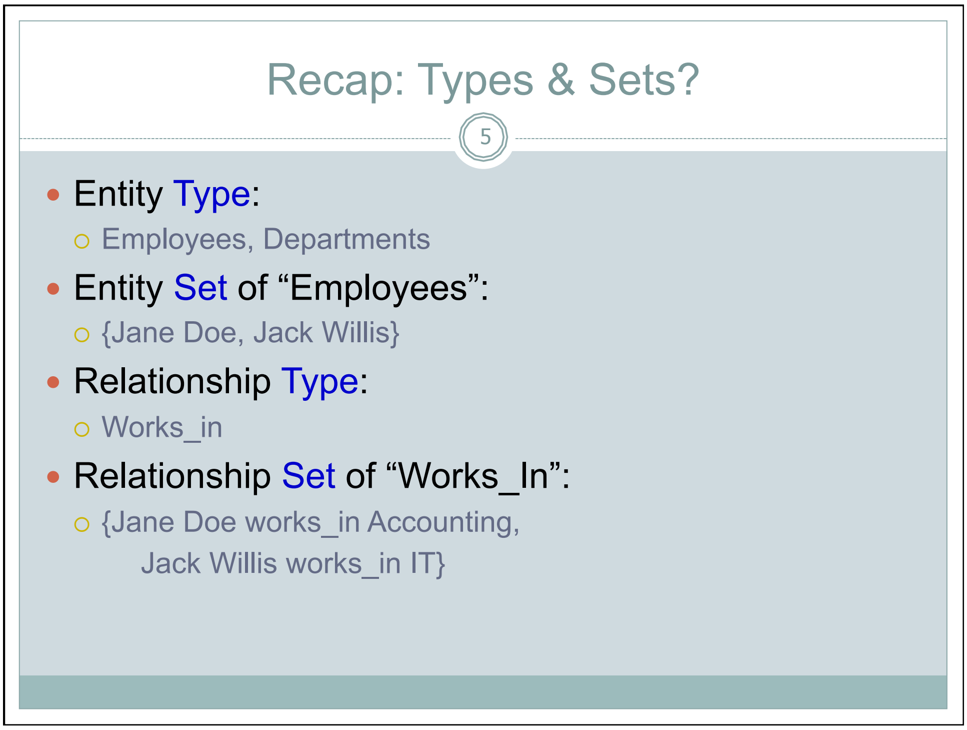
3

- 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

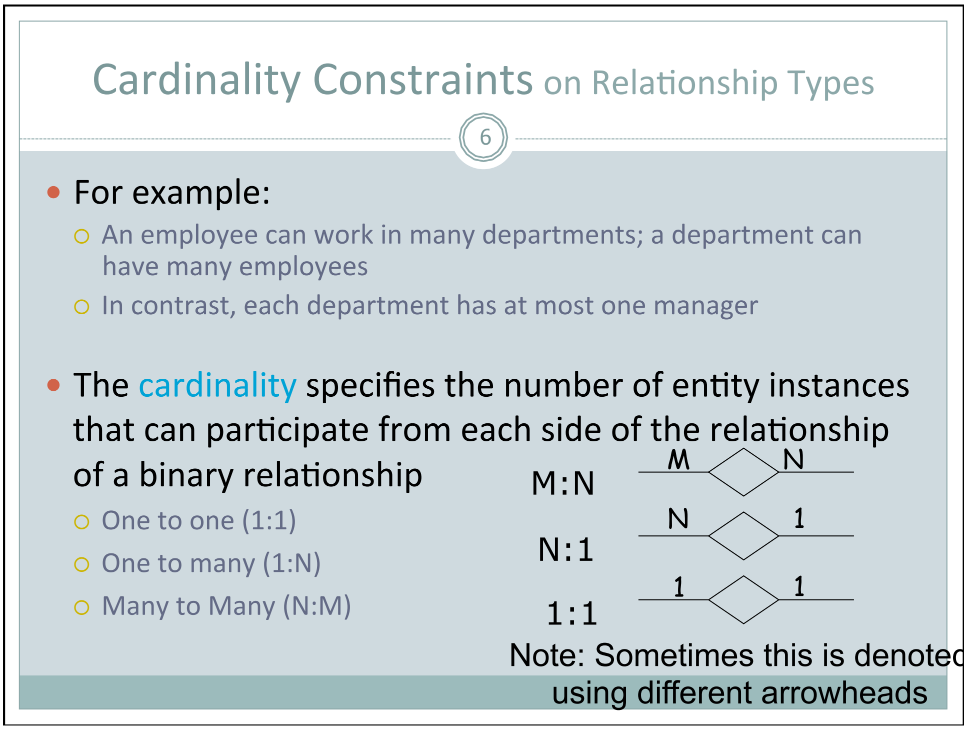
- Captures how two or more entities are related
- Can be thought of as verbs, linking two or more nouns
- Examples:
  - an *owns* relationship between a company and a computer
  - a *supervises* relationship between an employee and a department
  - a *performs* relationship between an artist and a song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



Recap: Types & Sets?

5

- 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

6

- 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)

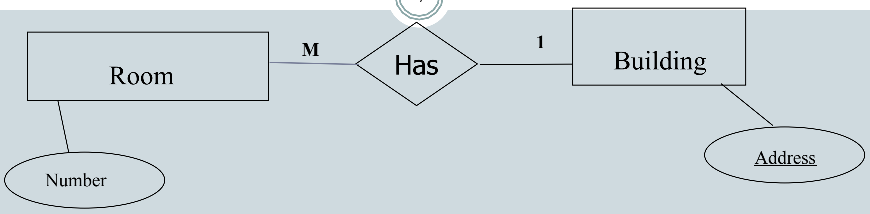
M:N

N:1

1:1

Note: Sometimes this is denoted
using different arrowheads

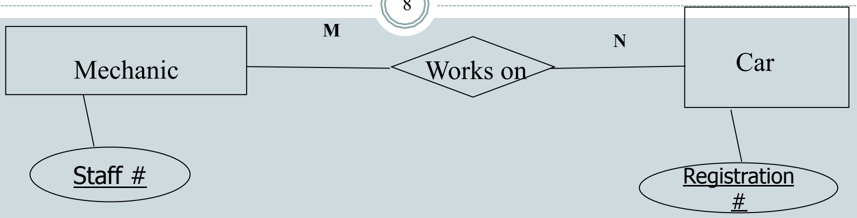
Example of 1 to N cardinality



For every ONE room, there is ONE building

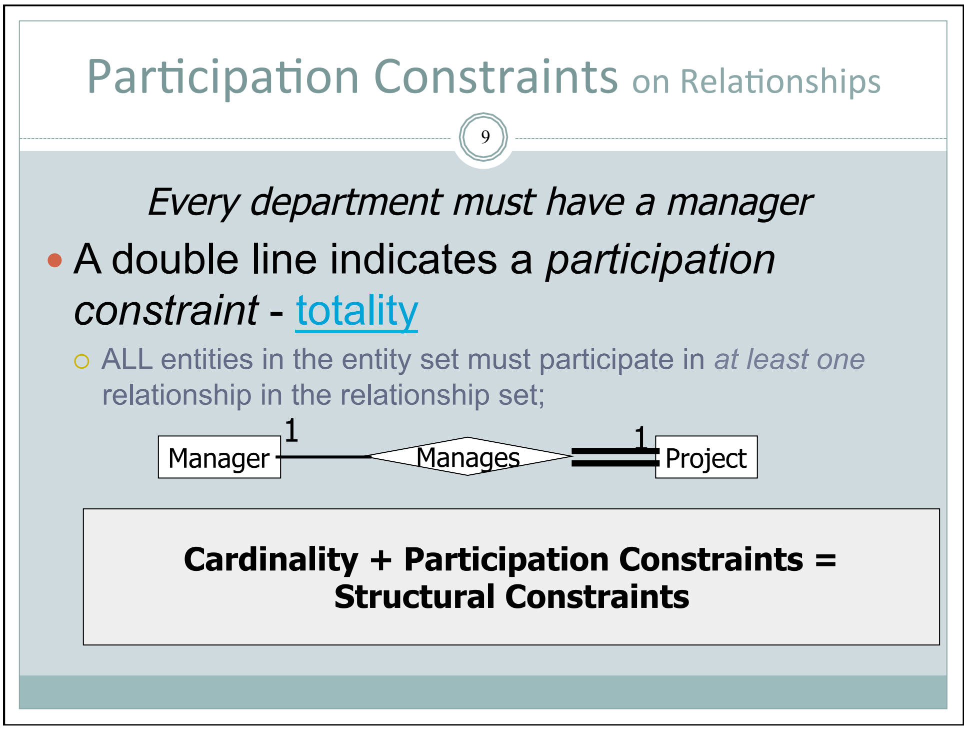
For every ONE building, there are ANY NUMBER of rooms

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



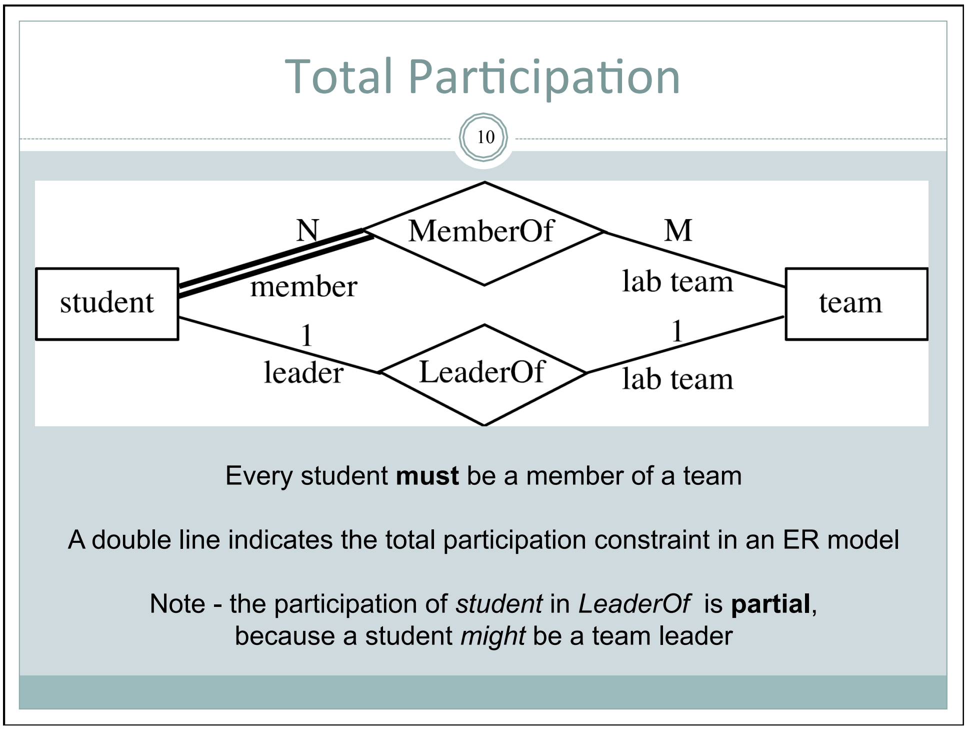
Participation Constraints on Relationships

9

Every department must have a manager

- A double line indicates a *participation constraint* - totality
  - ALL entities in the entity set must participate in *at least one* relationship in the relationship set;

Cardinality + Participation Constraints = Structural Constraints



Total Participation

10

Every student **must** be a member of a team

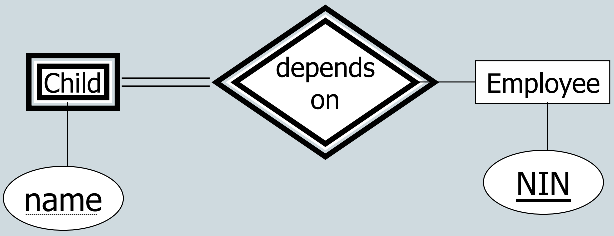
A double line indicates the total participation constraint in an ER model

Note - the participation of *student* in *LeaderOf* is **partial**, because a student *might* be a team leader

Weak Entity Types

- Depend on other entities to guarantee uniqueness
- Do not have primary key (attributes) of their own

weak entity

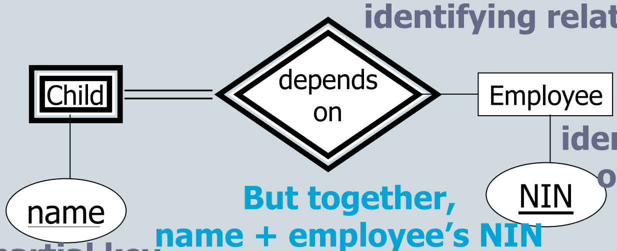


Weak Entity Types

- Depend on other entities to guarantee uniqueness
- Do not have primary key (attributes) of their own

identifying relationship

weak entity



identifying owner

name not guaranteed unique

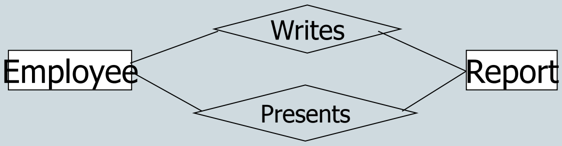
partial key (dotted ul)

But together, name + employee's NIN can be used as key

- Weak entity set must have total participation in this identifying relationship set.

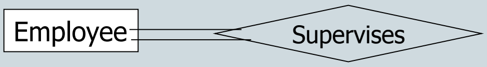
More on relationships - 1

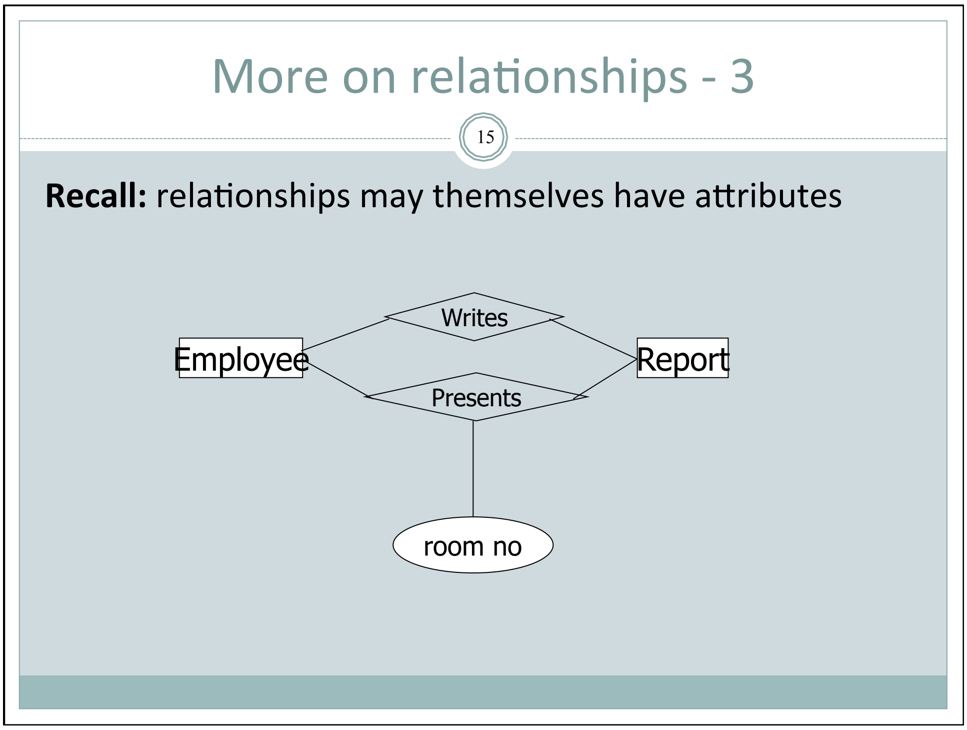
- There may be more than one relationship between entity types



More on relationships - 2

- An entity type may be in a relationship with itself
  - this is a recursive relationship

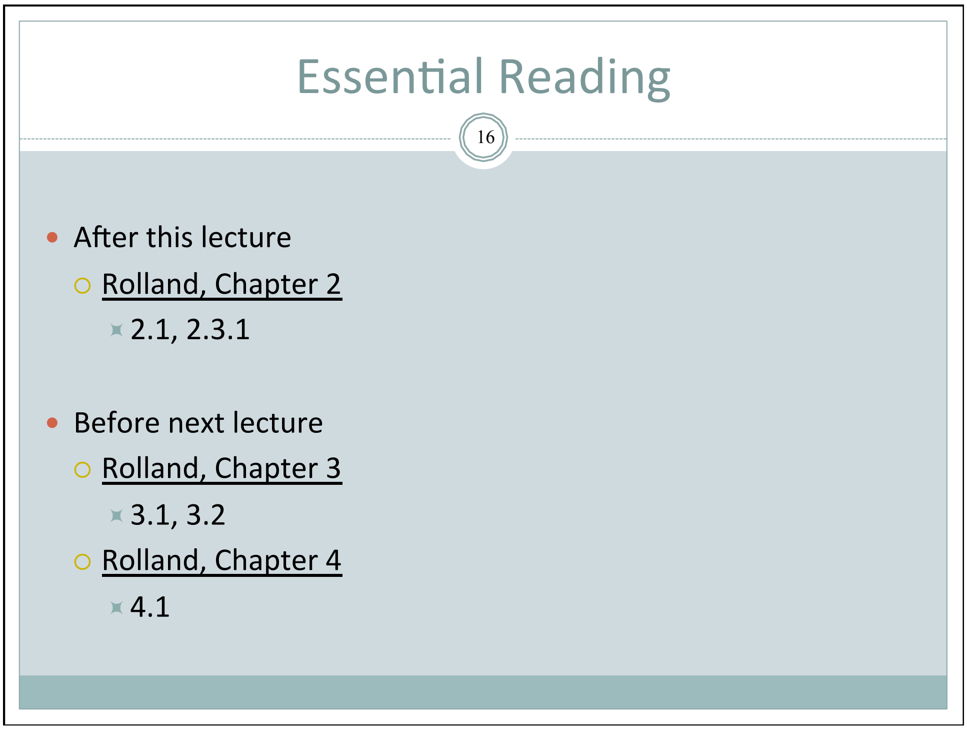




More on relationships - 3

15

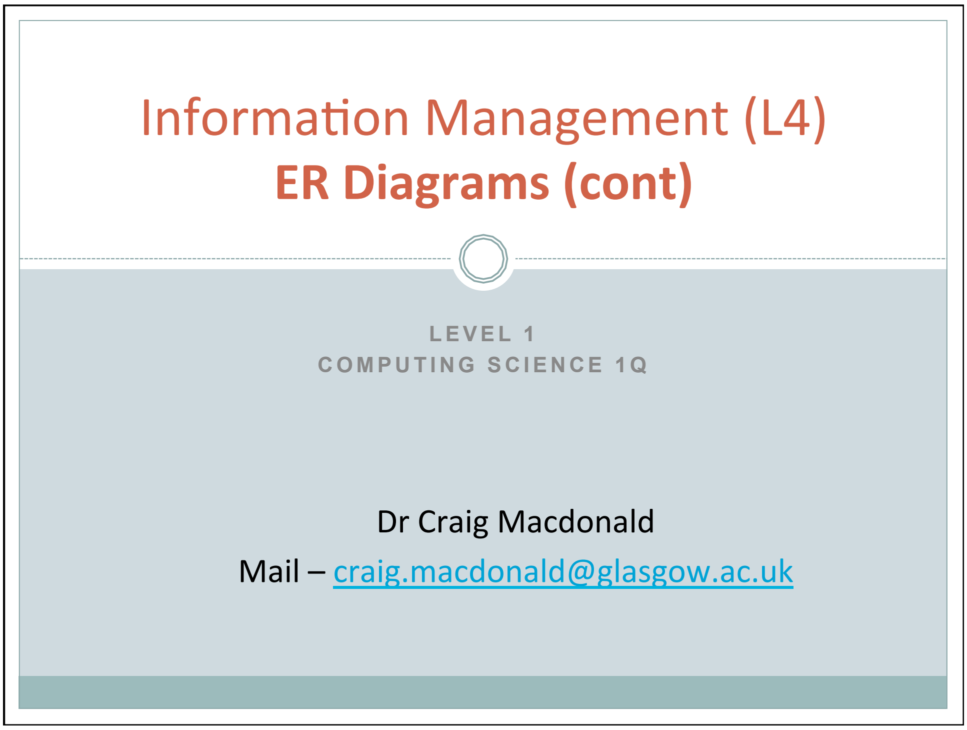
Recall: relationships may themselves have attributes



Essential Reading

16

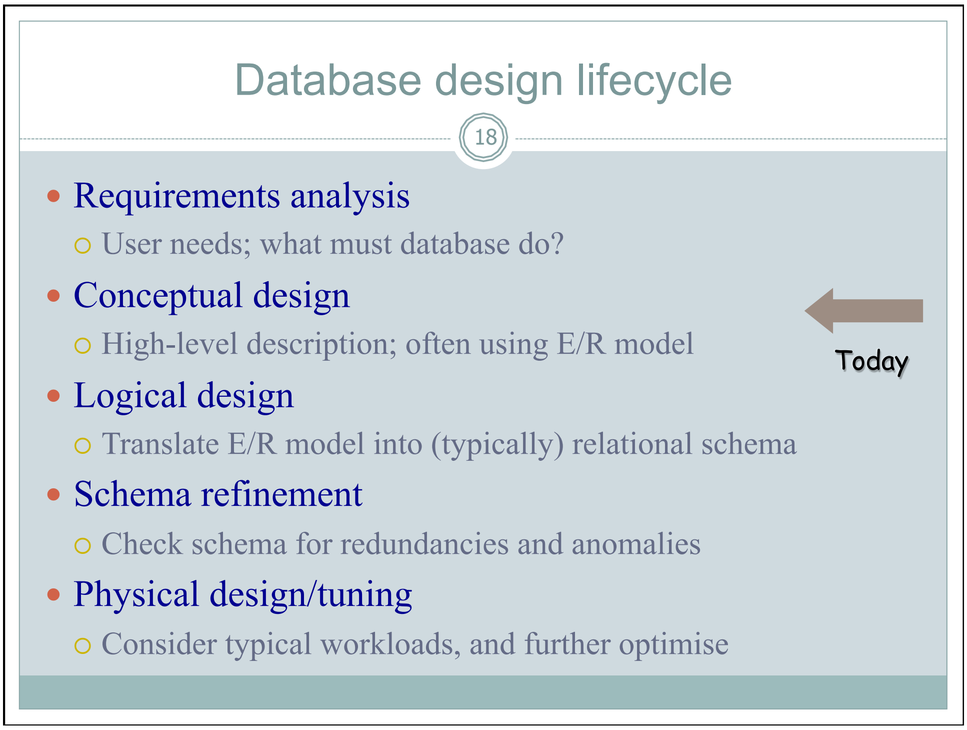
- After this lecture
  - Rolland, Chapter 2
    - 2.1, 2.3.1
- Before next lecture
  - Rolland, Chapter 3
    - 3.1, 3.2
  - Rolland, Chapter 4
    - 4.1



Information Management (L4)
ER Diagrams (cont)

LEVEL 1
COMPUTING SCIENCE 1Q

Dr Craig Macdonald
Mail – craig.macdonald@glasgow.ac.uk



Database design lifecycle

18

- 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

Today

From a written scenario to an ER Model

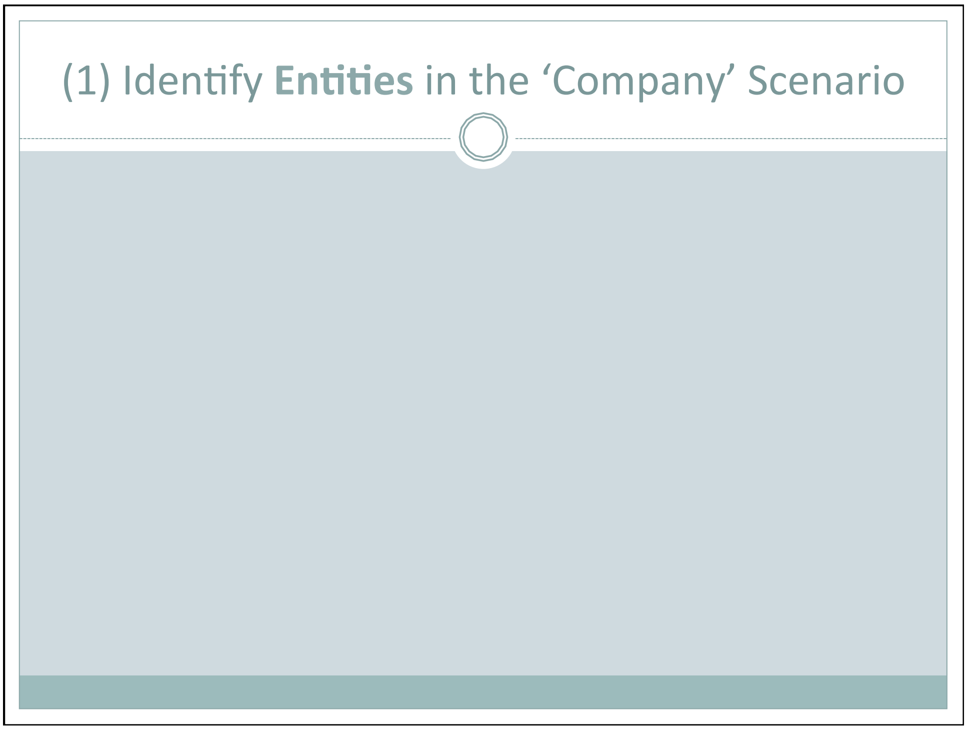
19

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

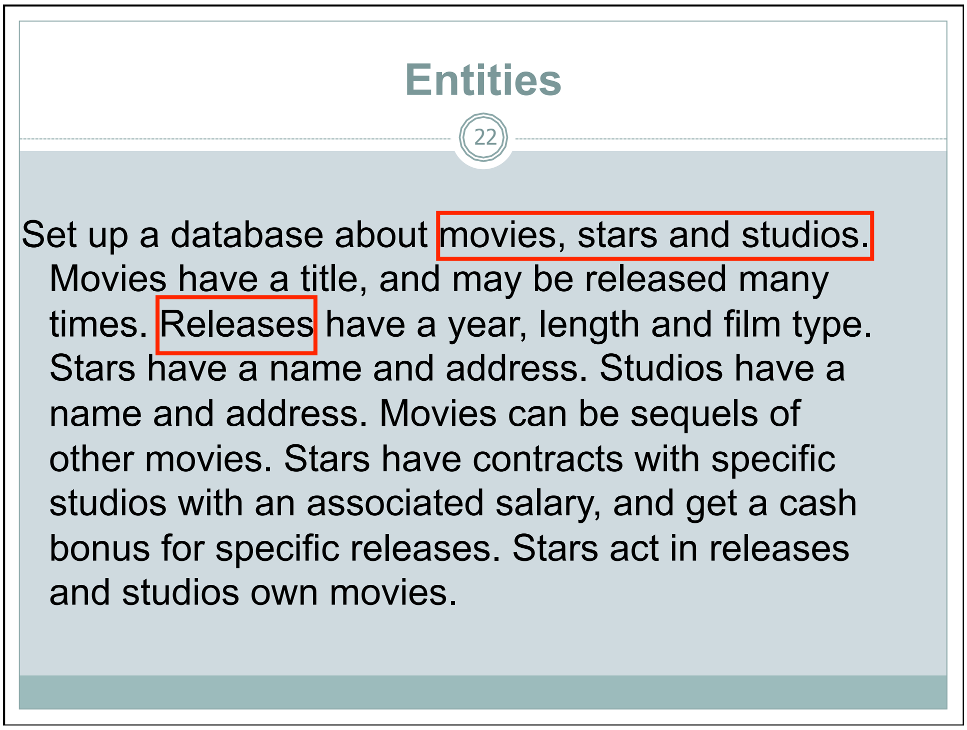
Constructing an ER diagram

20

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)



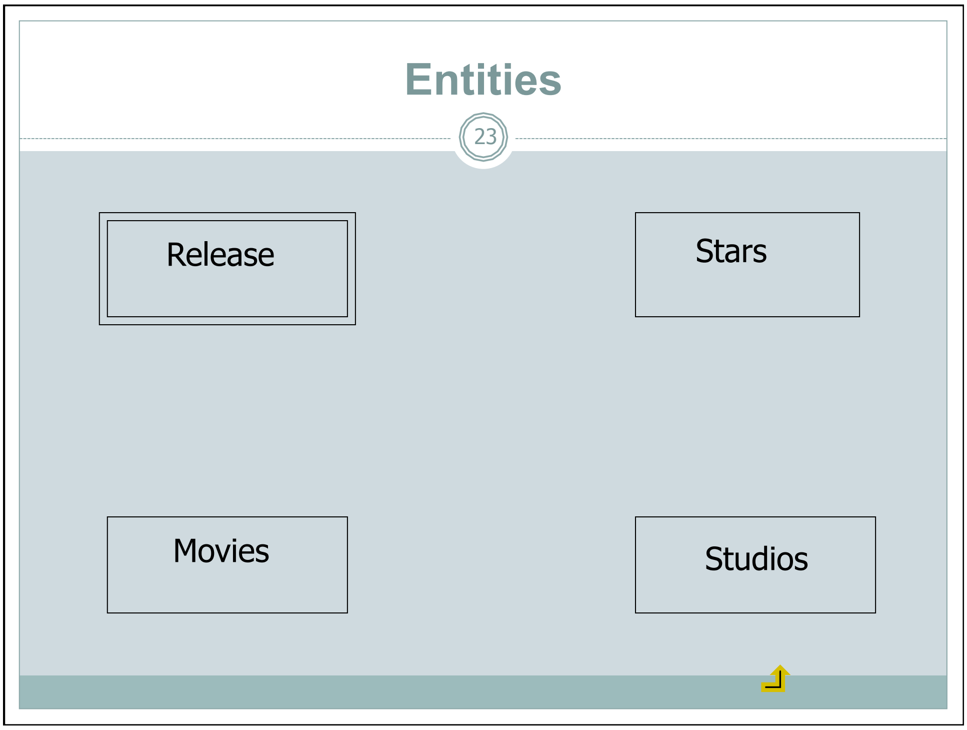
(1) Identify **Entities** in the ‘Company’ Scenario



Entities

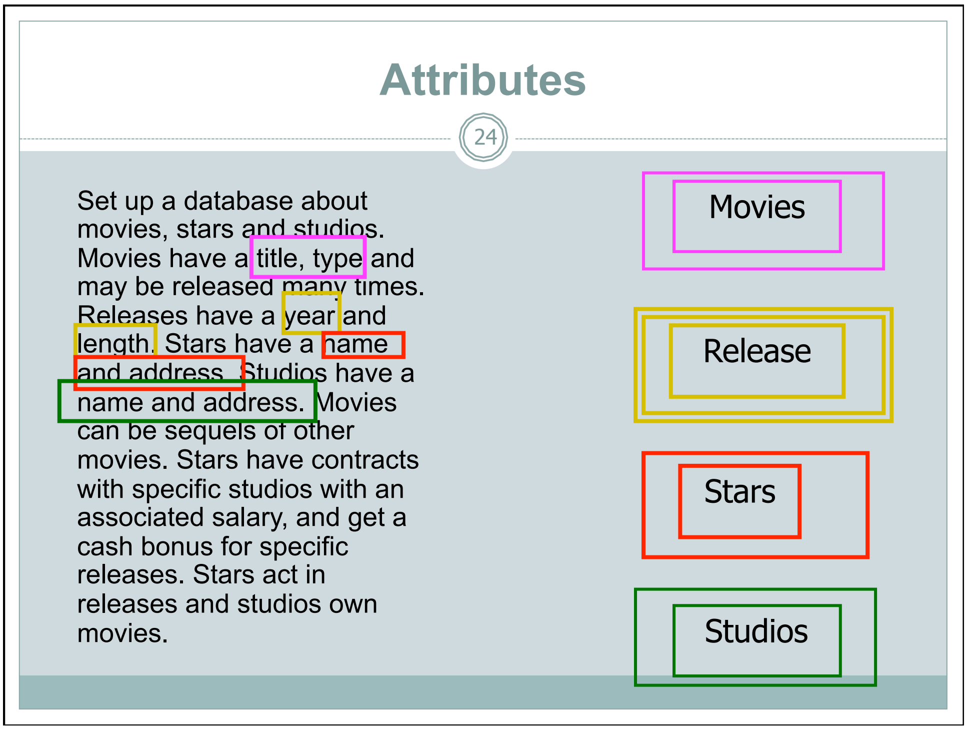
22

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. 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

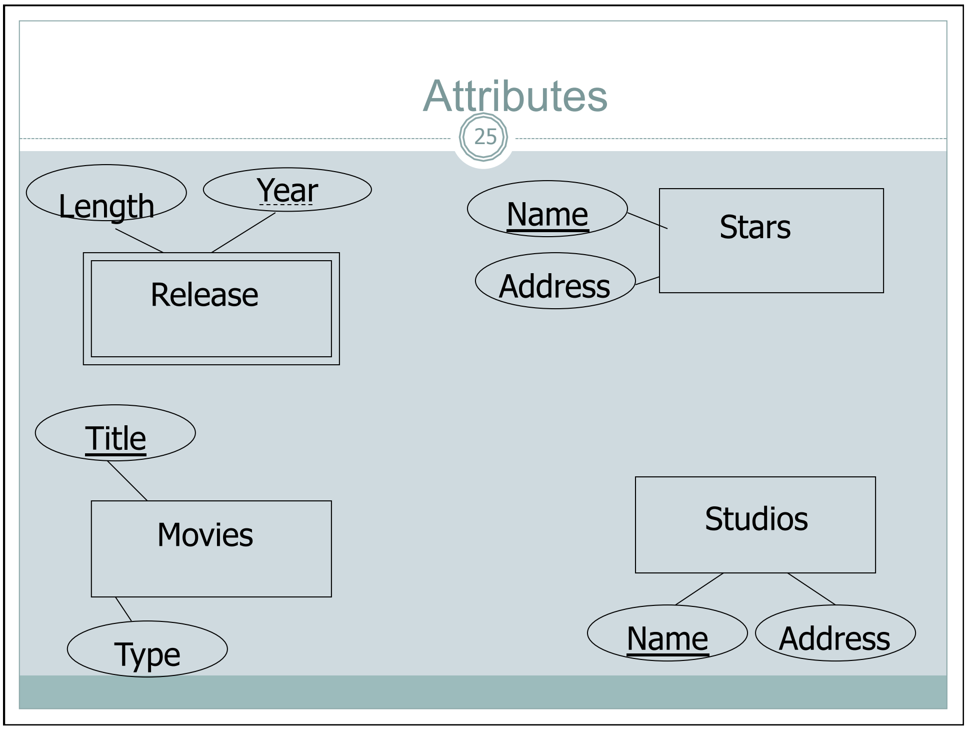
23



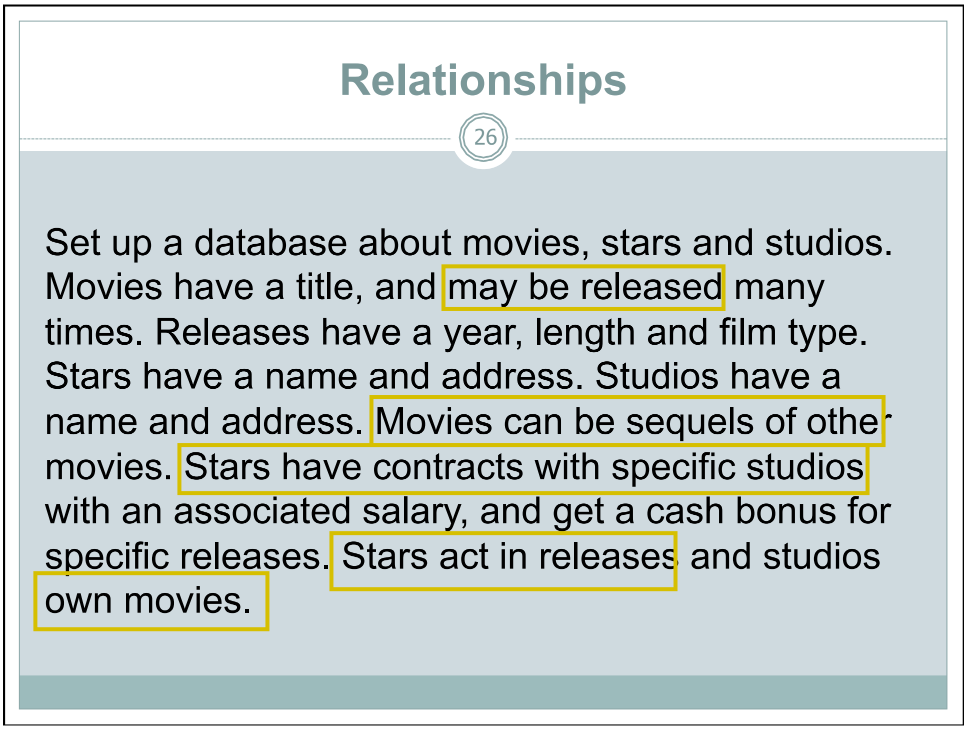
Attributes

24

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. 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.



Attributes

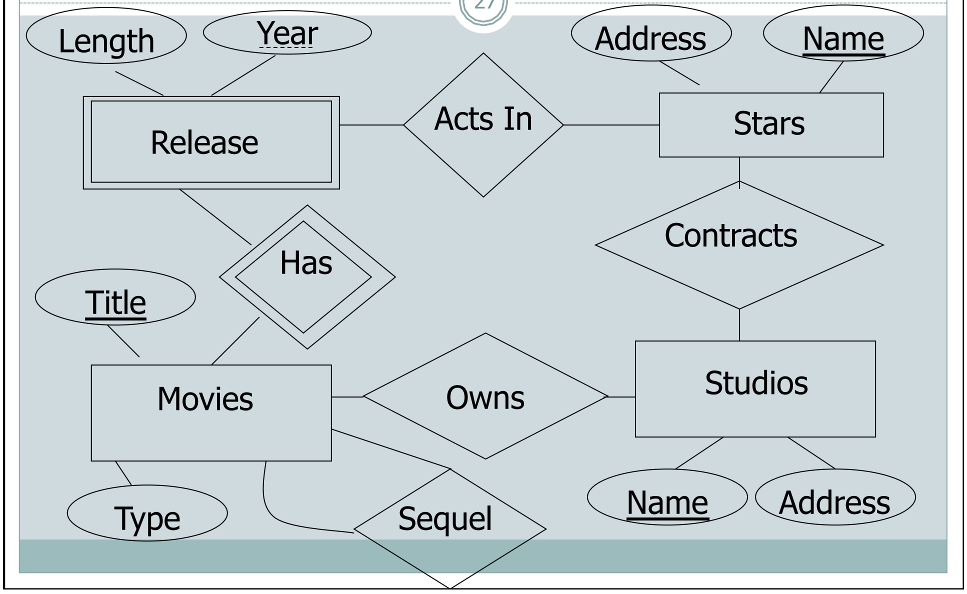


Relationships

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. 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

27



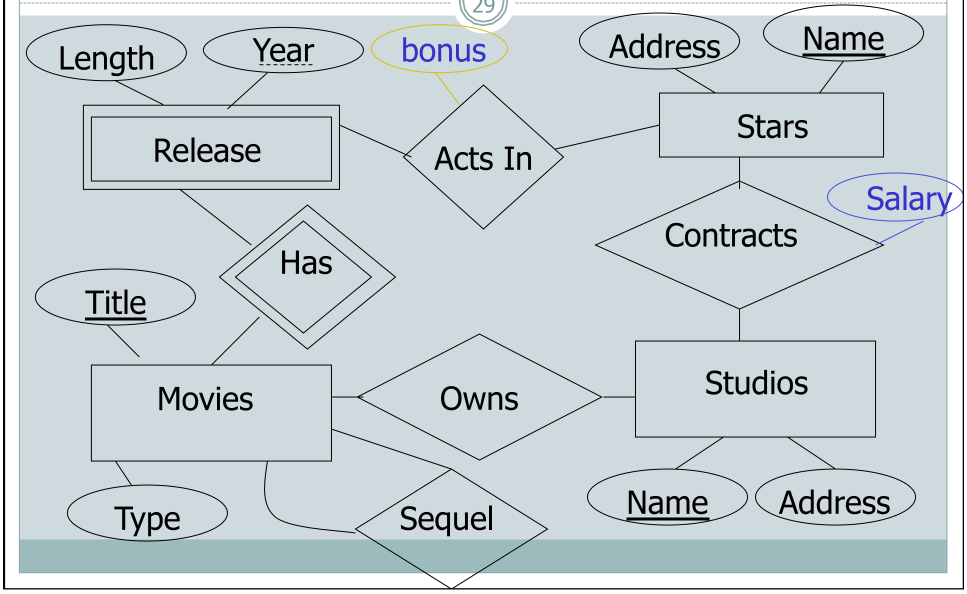
Relationship Attributes

28

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. 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.

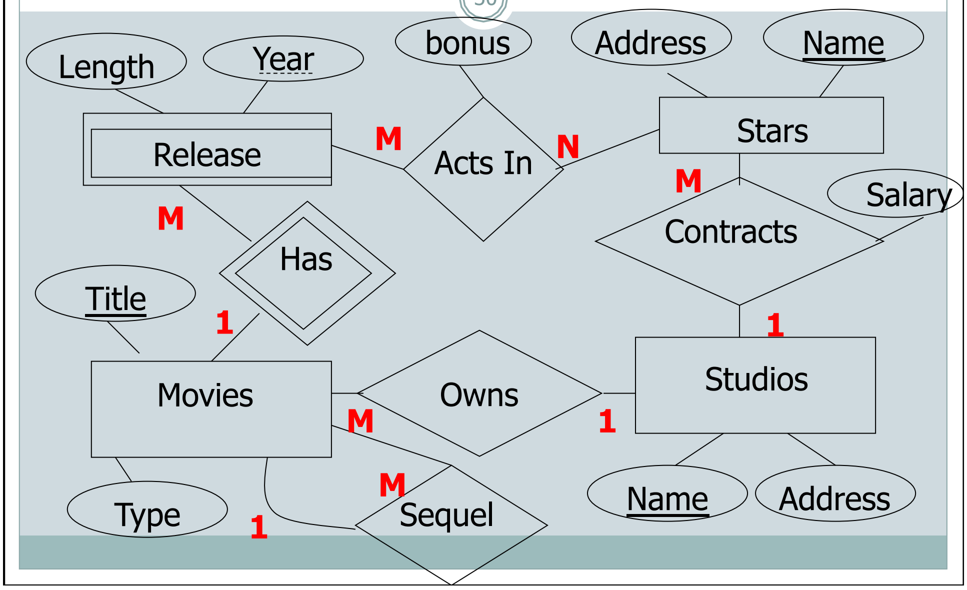
Relationship Attributes

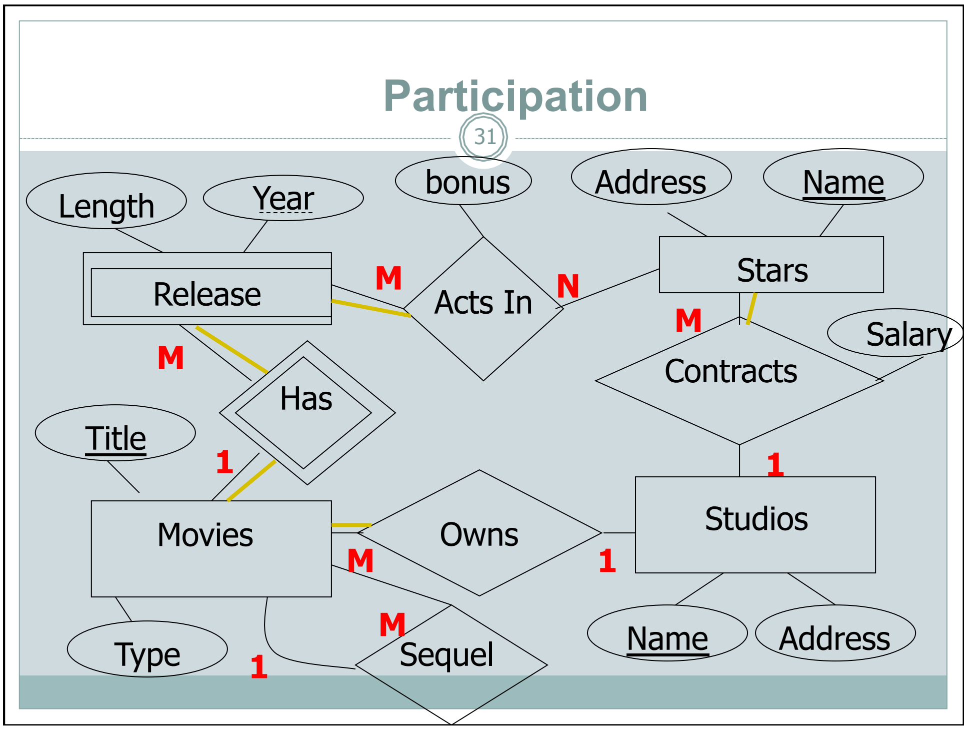
29



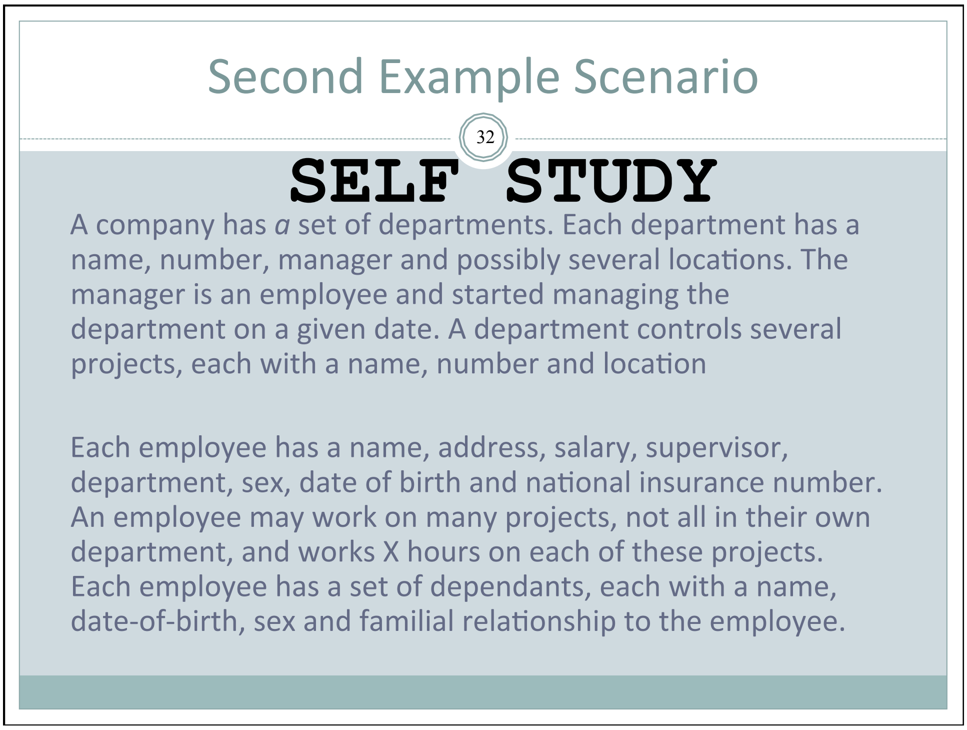
Cardinality

30





Participation



Second Example Scenario

SELF STUDY

A company has *a* set of departments. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A department controls several projects, each with a name, number and location

Each employee has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a name, date-of-birth, sex and familial relationship to the employee.

The Example Scenario

A company has *a* set of **departments**. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A department controls several **projects**, each with a name, number and location

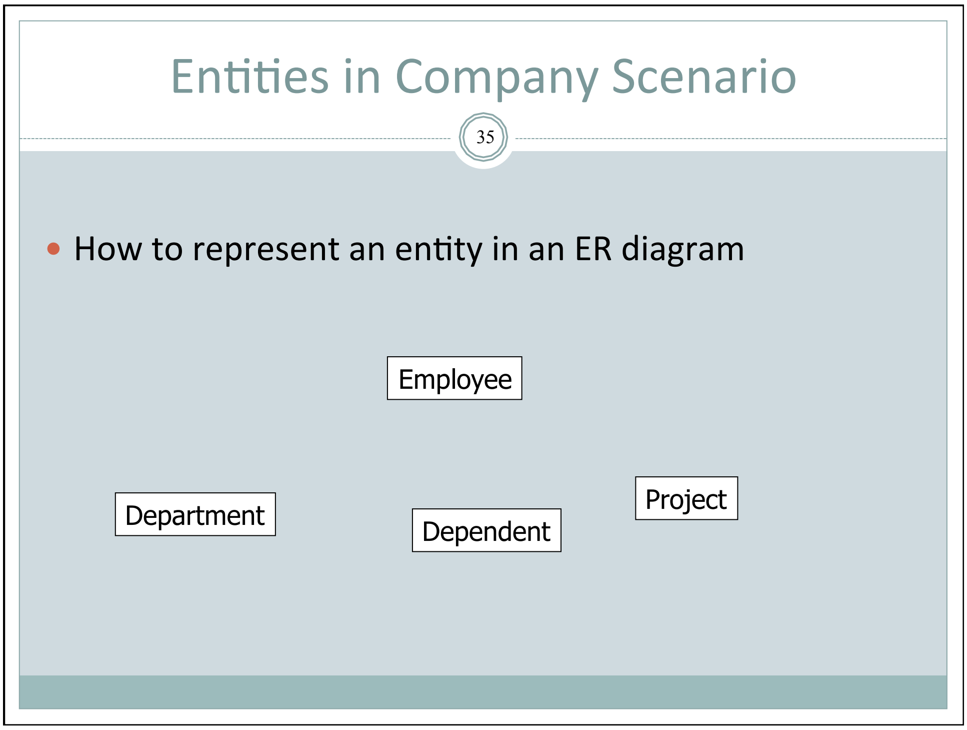
Each **employee** has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of **dependants**, each with a name, date-of-birth, sex and familial relationship to the employee.

Entities in the Company Scenario

Departments, Employees, Projects, Dependants

Notes

- Company is **not** an entity type - it is the whole database
- Some things are relationships rather than entities themselves
  - Managers ? “The **manager** *is an* **employee** “
  - Supervisors ? “Each **employee** has a **supervisor”**

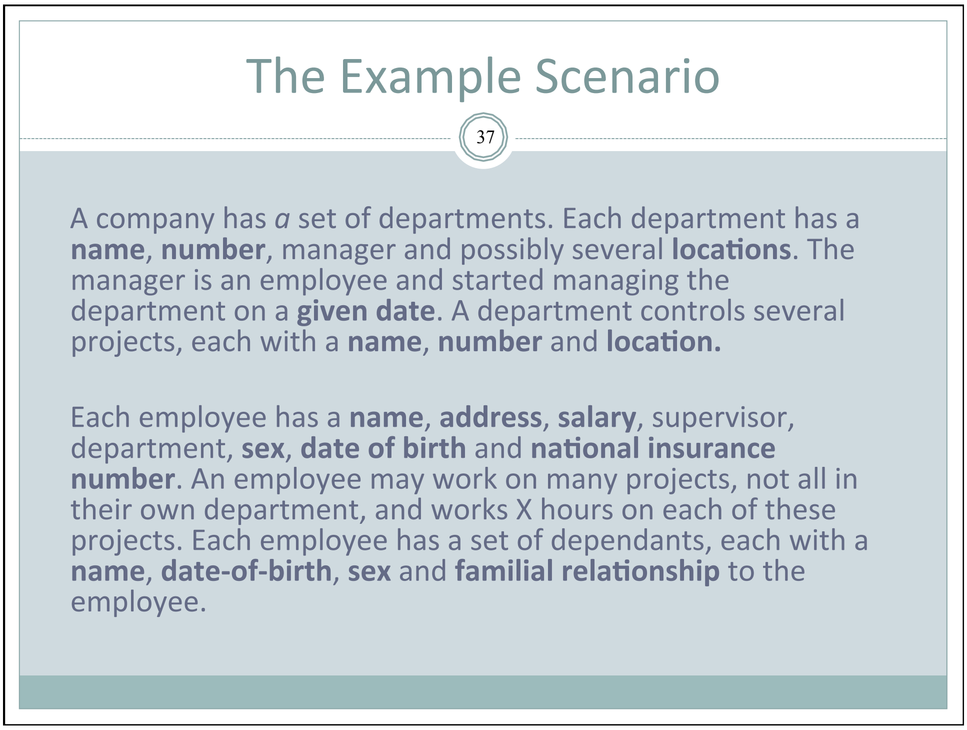


Entities in Company Scenario

- How to represent an entity in an ER diagram

Employee
Department
Dependent
Project

(2) Identify **Attributes** in Company Scenario

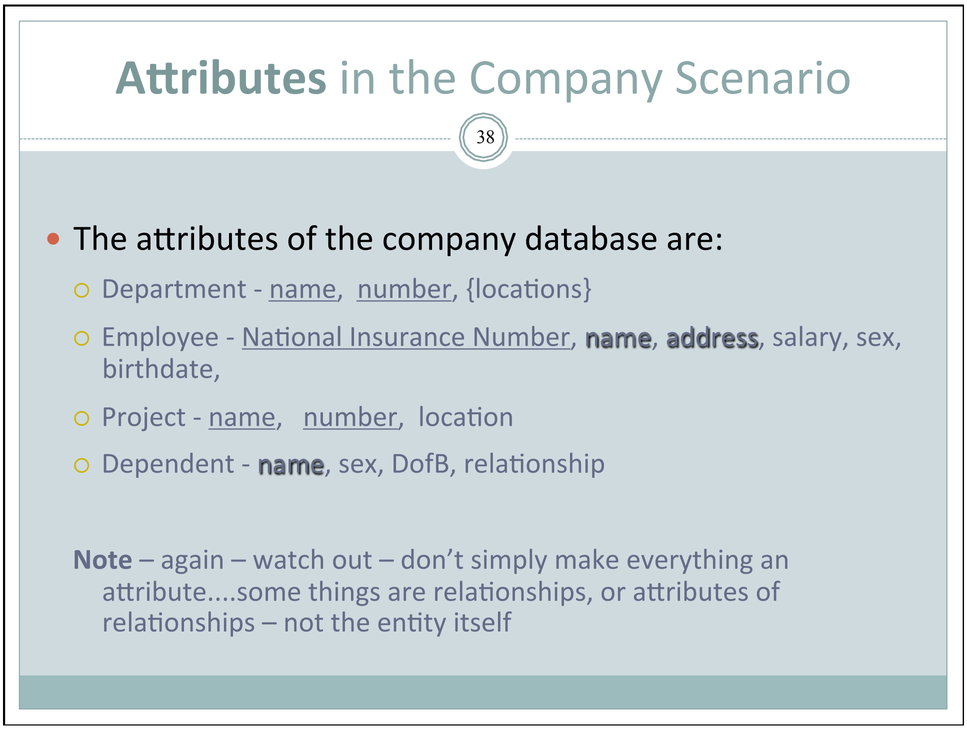


The Example Scenario

37

A company has *a* set of departments. Each department has a **name**, **number**, manager and possibly several **locations**. The manager is an employee and started managing the department on a **given date**. A department controls several projects, each with a **name**, **number** and **location.**

Each employee has a **name**, **address**, **salary**, supervisor, department, **sex**, **date of birth** and **national insurance number**. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a **name**, **date-of-birth**, **sex** and **familial relationship** to the employee.

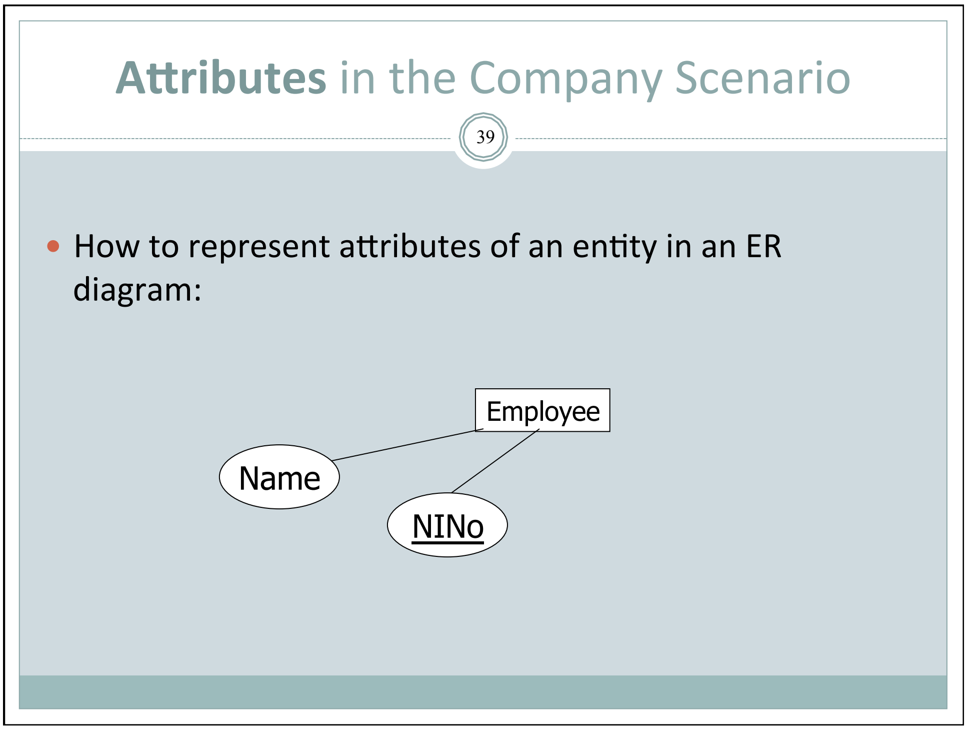


Attributes in the Company Scenario

38

- The attributes of the company database are:
  - Department - name, number, {locations}
  - Employee - National Insurance Number, **name**, **address**, salary, sex, birthdate,
  - Project - name, number, location
  - Dependent - **name**, sex, DofB, relationship

Note – again – watch out – don't simply make everything an attribute....some things are relationships, or attributes of relationships – not the entity itself

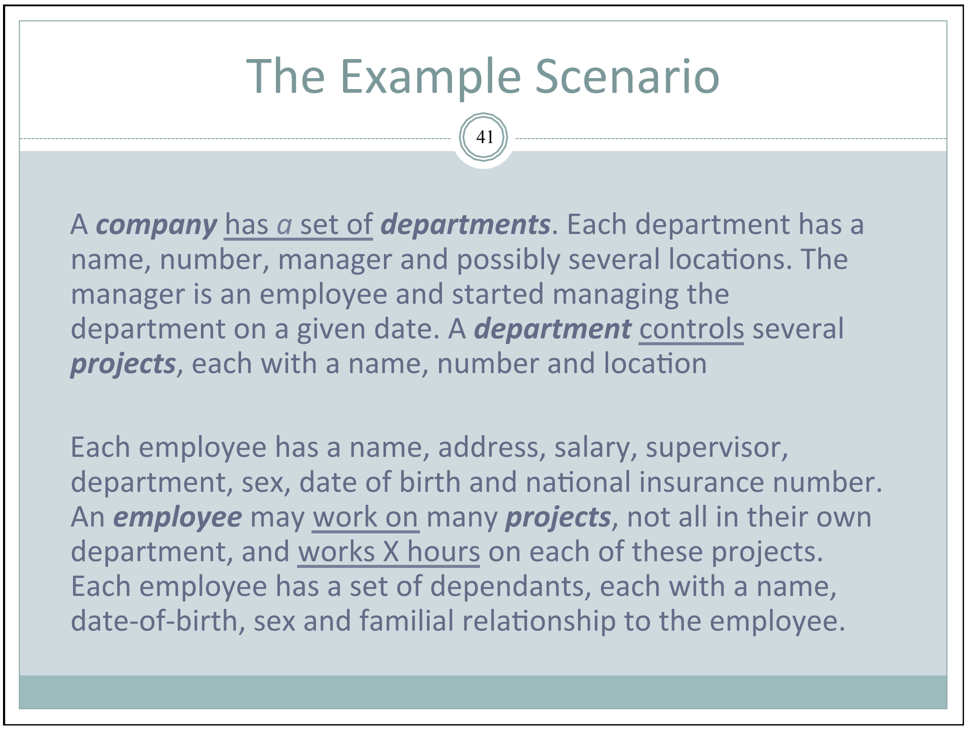


Attributes in the Company Scenario

39

- How to represent attributes of an entity in an ER diagram:

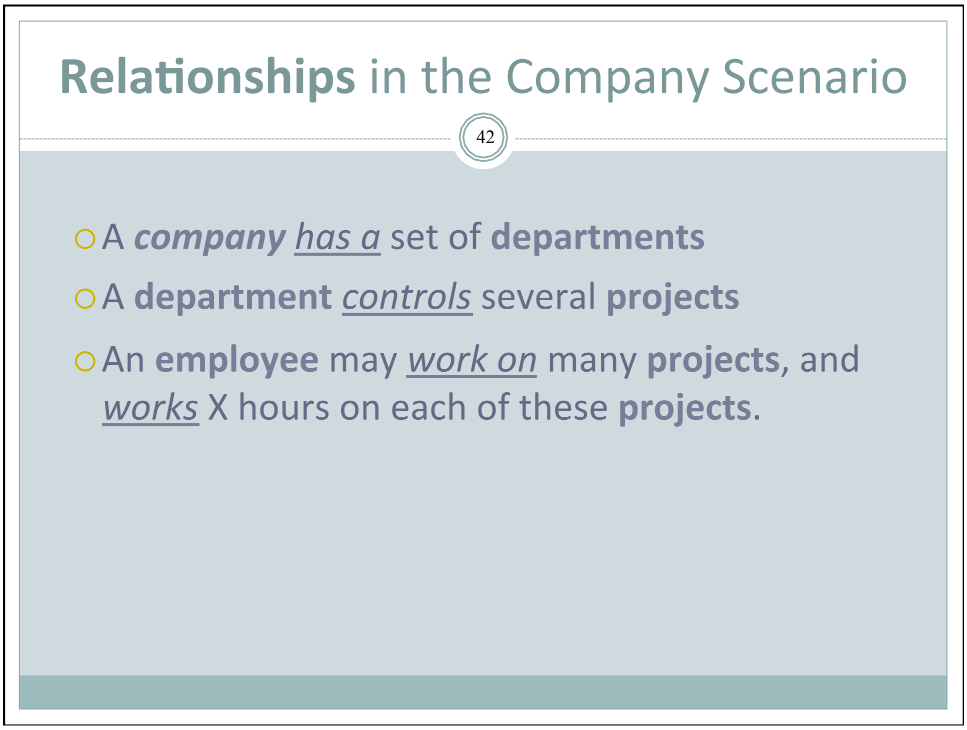
(3) Identify Relationships in Company Scenario



The Example Scenario

A ***company*** has *a* set of ***departments***. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A ***department*** controls several ***projects***, each with a name, number and location

Each employee has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An ***employee*** may work on many ***projects***, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a name, date-of-birth, sex and familial relationship to the employee.



Relationships in the Company Scenario

- A ***company*** *has a* set of **departments**
- A **department** *controls* several **projects**
- An **employee** may *work on* many **projects**, and *works* X hours on each of these **projects**.

Relationships in the Company Scenario

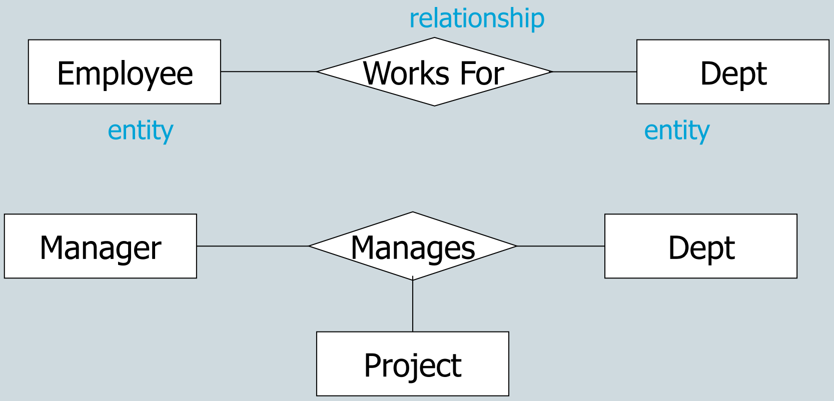
43

Relationships with their own attributes

- Each **employee** *has a* set of **dependants**, each with a **name, date-of-birth, sex and familial relationship** to the **employee**.
- The **manager** *is an* **employee** and *started managing* the **department** on a given date

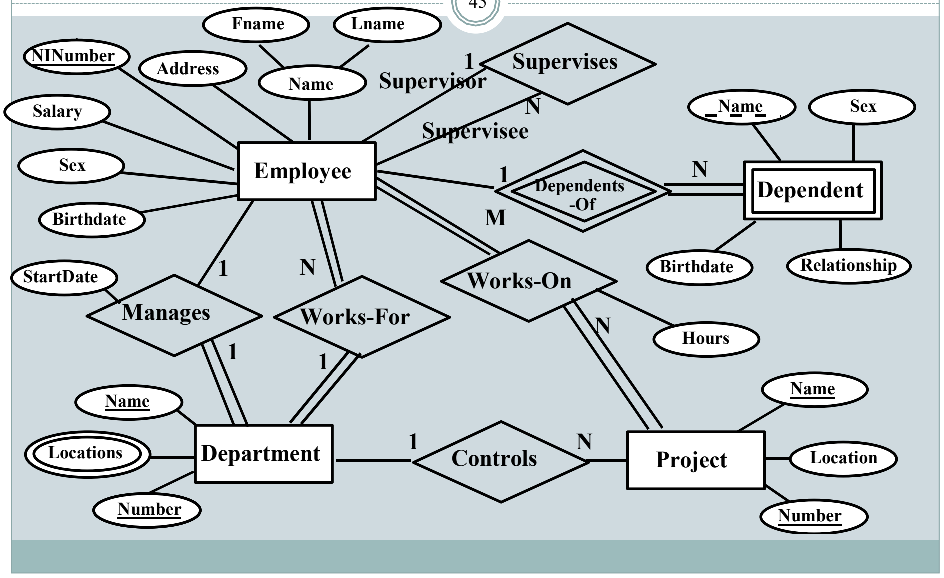
Representing Relationships in an ER Diagram

44

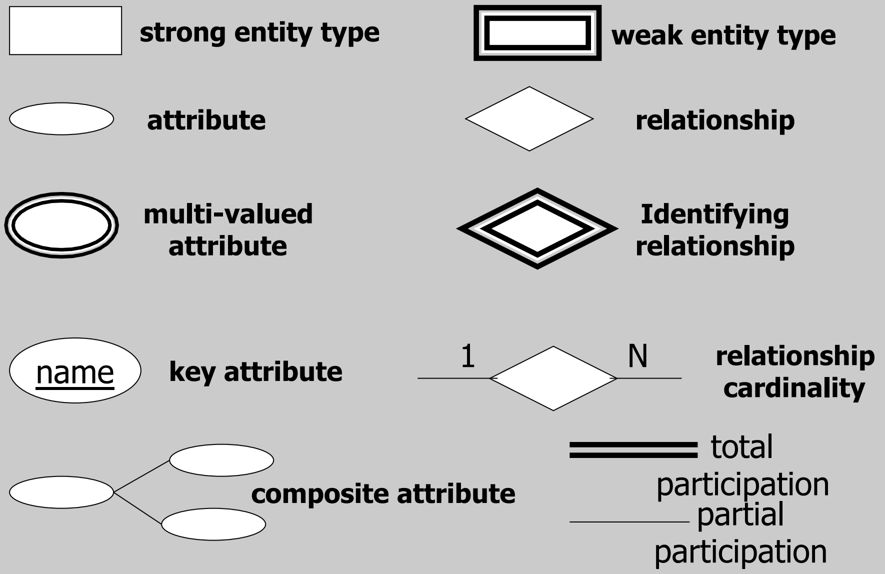


ER Diagram for the Company Database

45



ER Diagram Notation



strong entity type

weak entity type

attribute

relationship

multi-valued attribute

Identifying relationship

name

key attribute

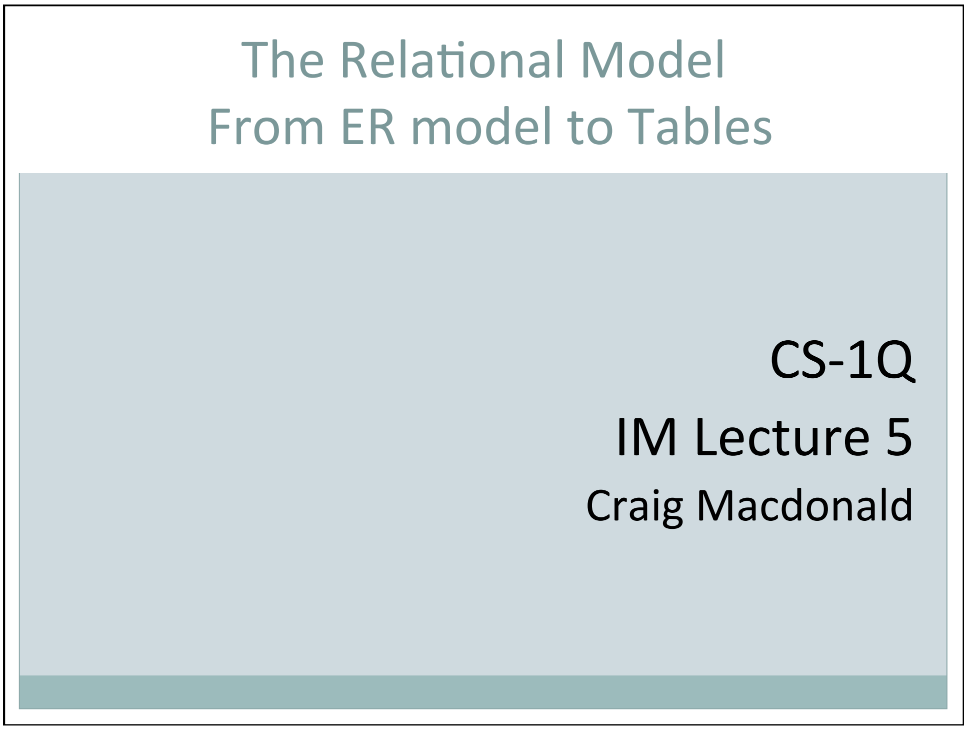
1 N

relationship cardinality

composite attribute

total participation

partial participation

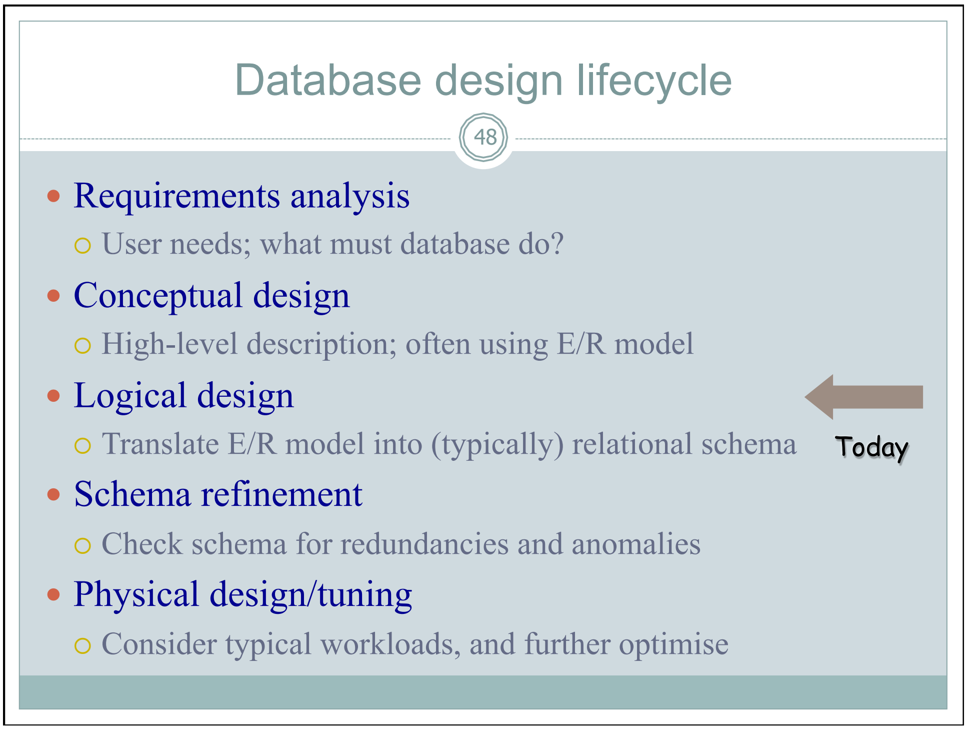


The Relational Model
From ER model to Tables

CS-1Q

IM Lecture 5

Craig Macdonald

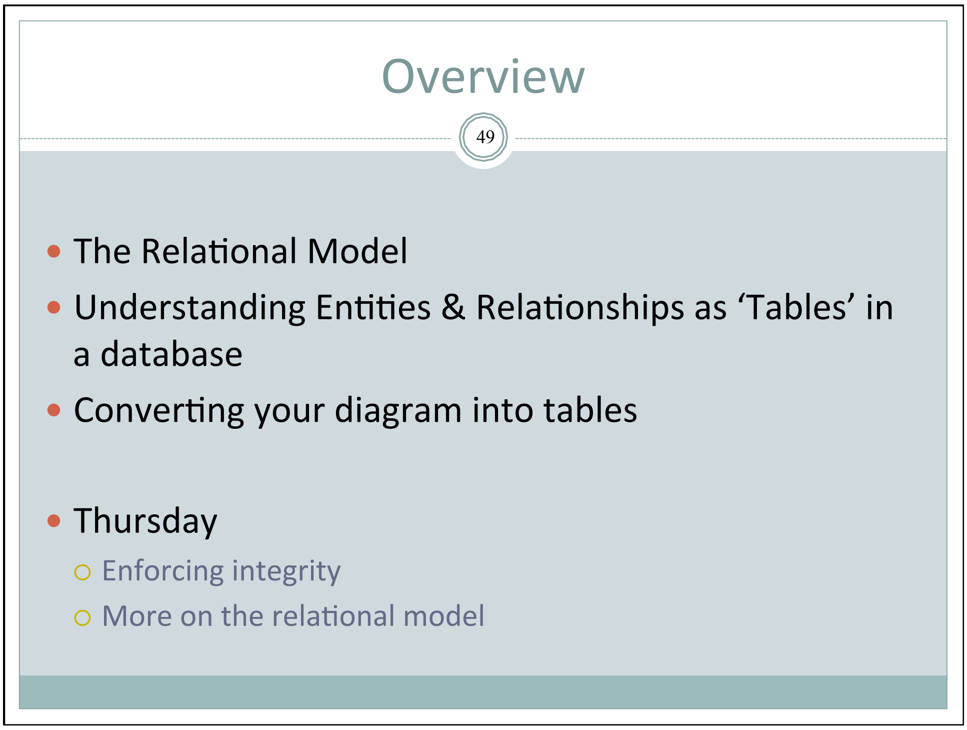


Database design lifecycle

48

- 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

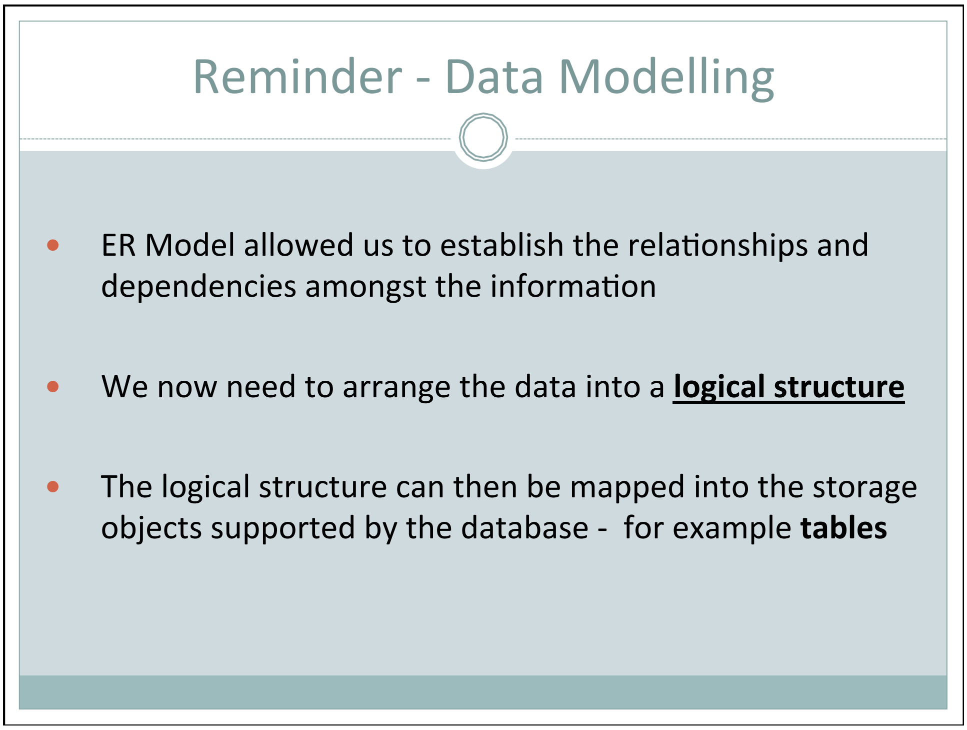
Today



Overview

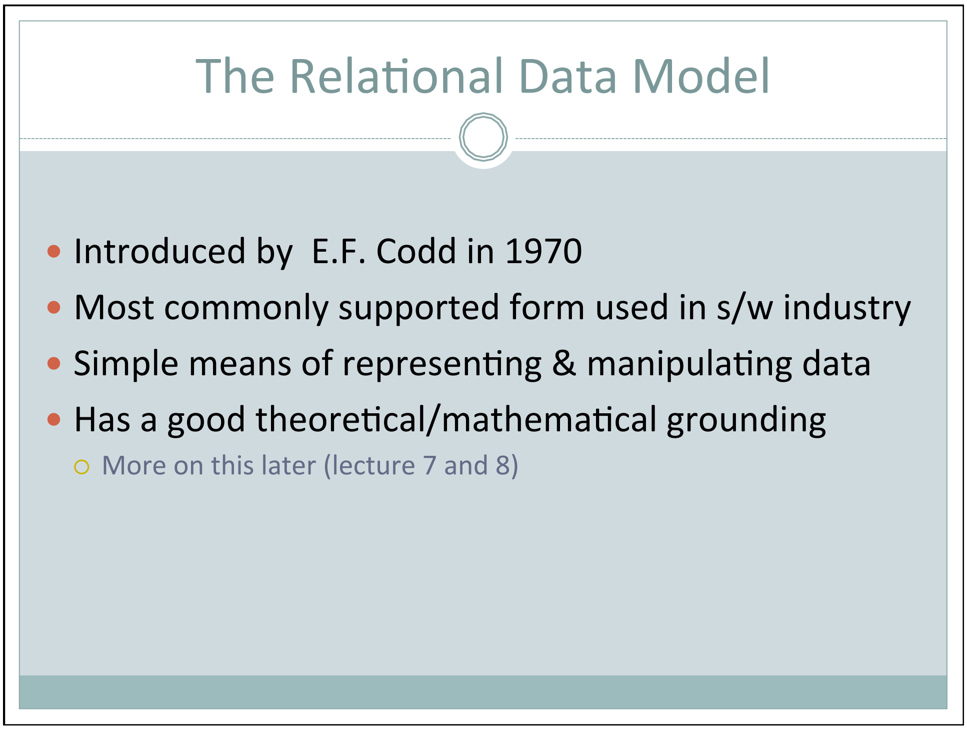
49

- The Relational Model
- Understanding Entities & Relationships as ‘Tables’ in a database
- Converting your diagram into tables
- Thursday
  - Enforcing integrity
  - More on the relational model



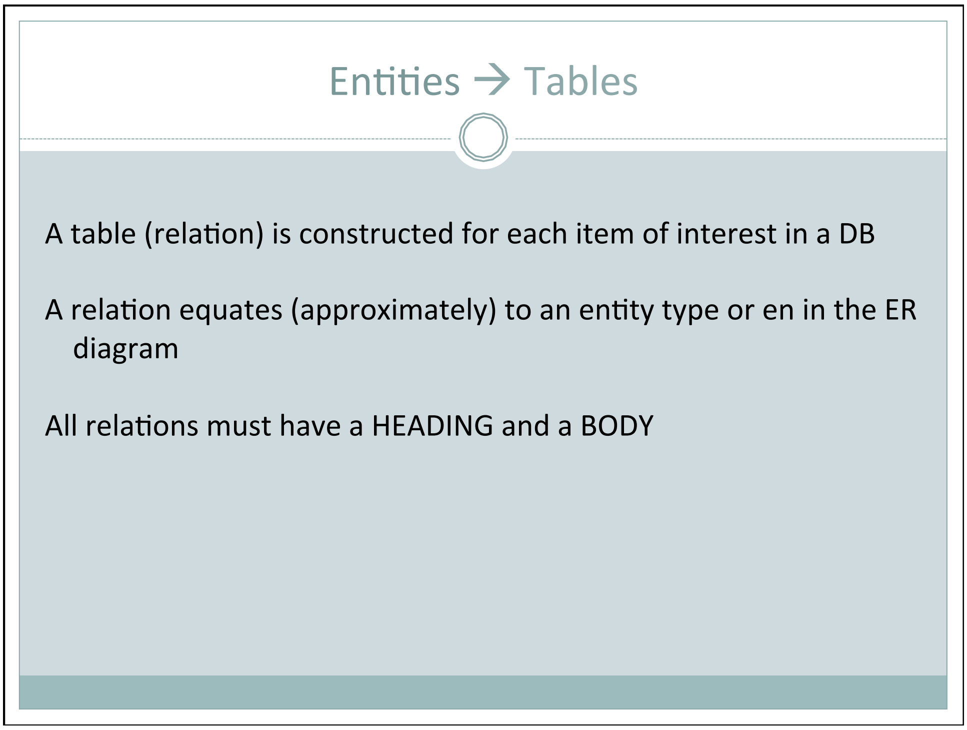
Reminder - Data Modelling

- 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**



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

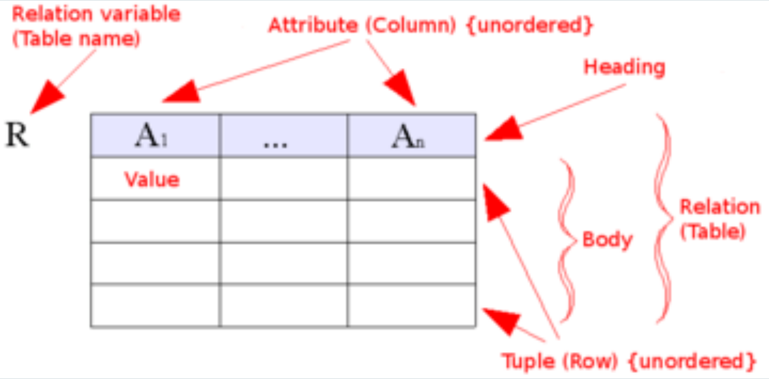
A table (relation) is constructed for each item of interest in a DB

A relation equates (approximately) to an entity type or en in the ER diagram

All relations must have a HEADING and a BODY

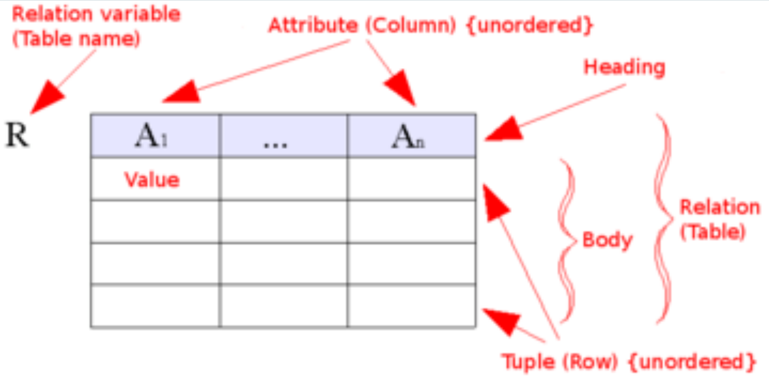
Structure of Data Objects in the Relational Model

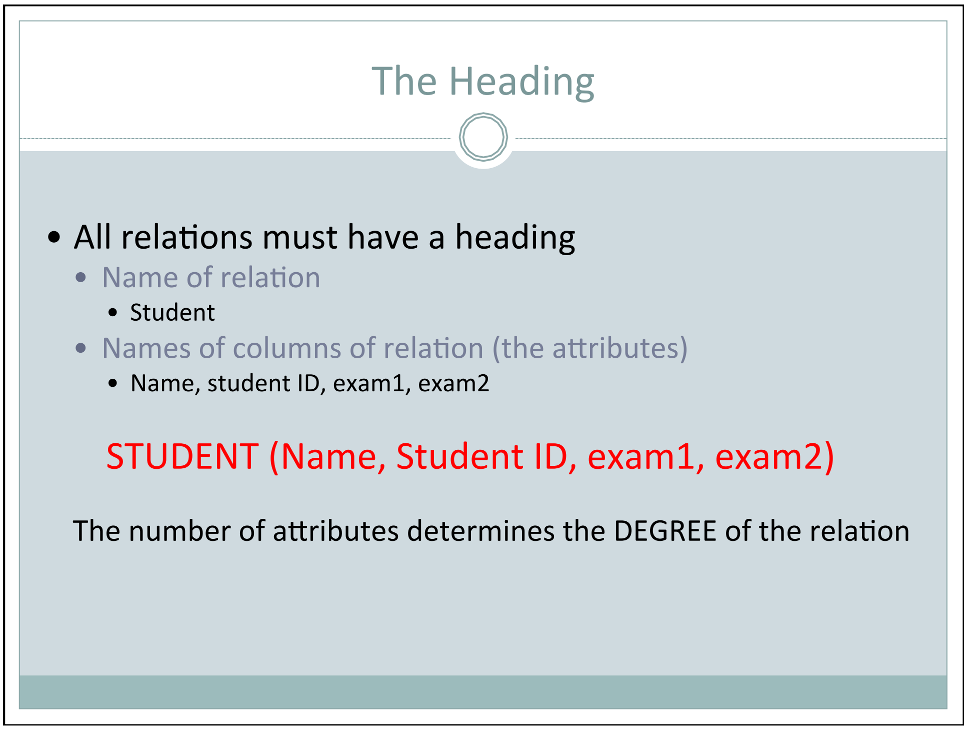
- Data is represented in two dimesional TABLES (relations)



Structure of Data Objects in the Relational Model

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



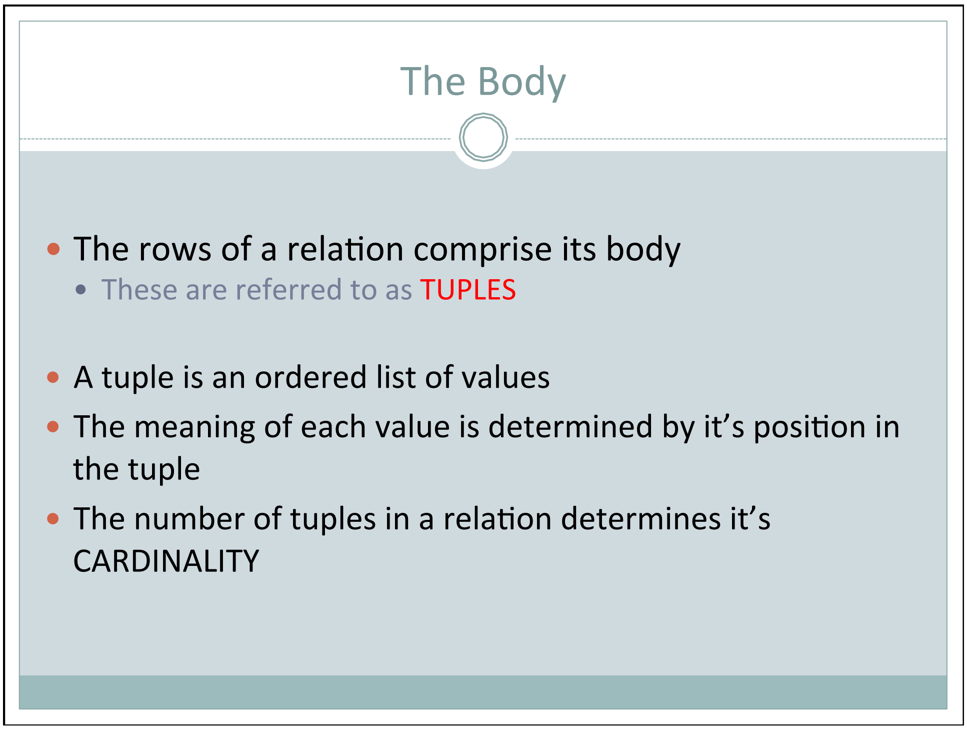


The Heading

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

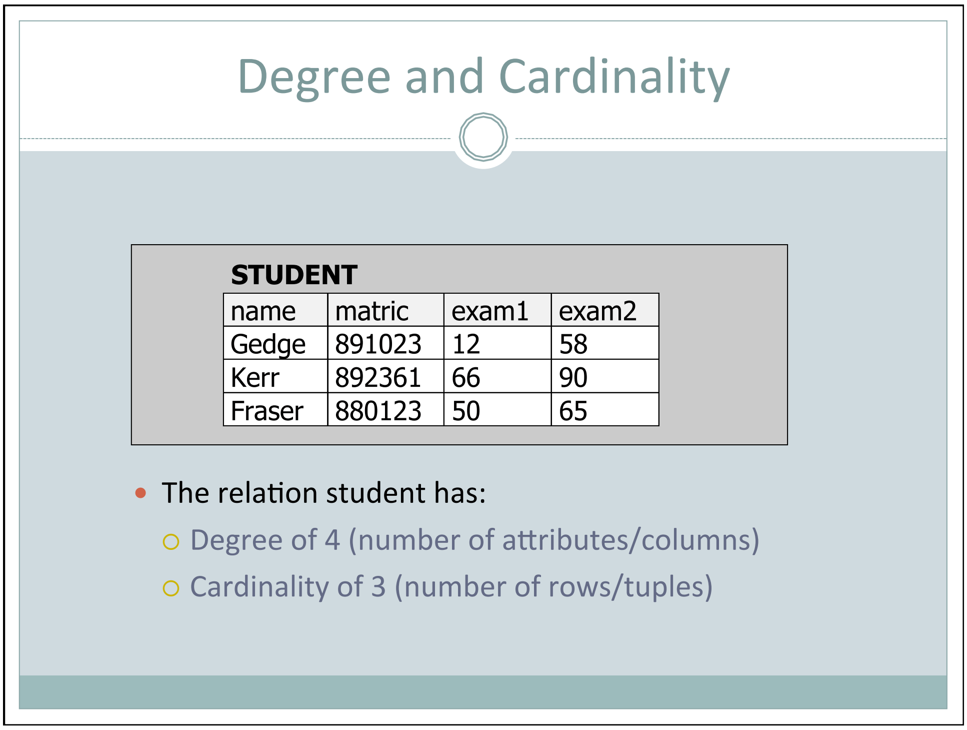
STUDENT (Name, Student ID, exam1, exam2)

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 it’s position in the tuple
- The number of tuples in a relation determines it’s CARDINALITY

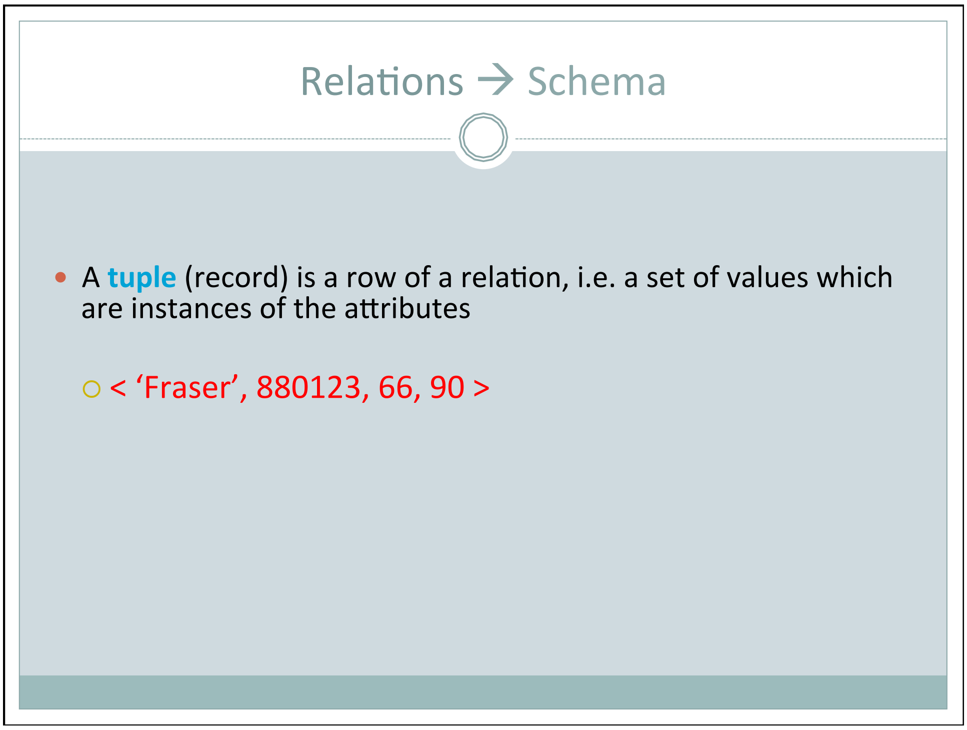


Degree and Cardinality

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)



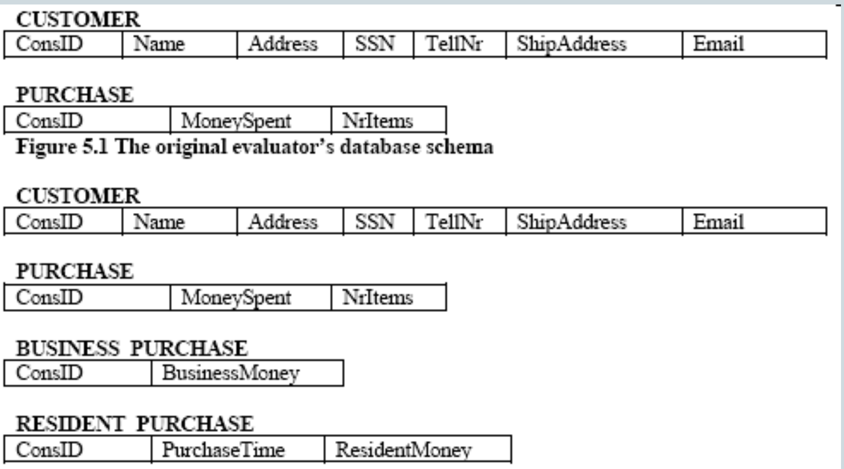
Relations → Schema

- 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, $A_{2,\ldots}$ A_n) e.g.
  - Student (name: Text, matric: Number, ex1: Number, ex2: Number)
- A **relational database schema** is a set of these relation schemas

Relational Schema



CUSTOMER

ConsID	Name	Address	SSN	TelNr	ShipAddress	Email

PURCHASE

ConsID	MoneySpent	NrItems

Figure 5.1 The original evaluator's database schema

CUSTOMER

ConsID	Name	Address	SSN	TelNr	ShipAddress	Email

PURCHASE

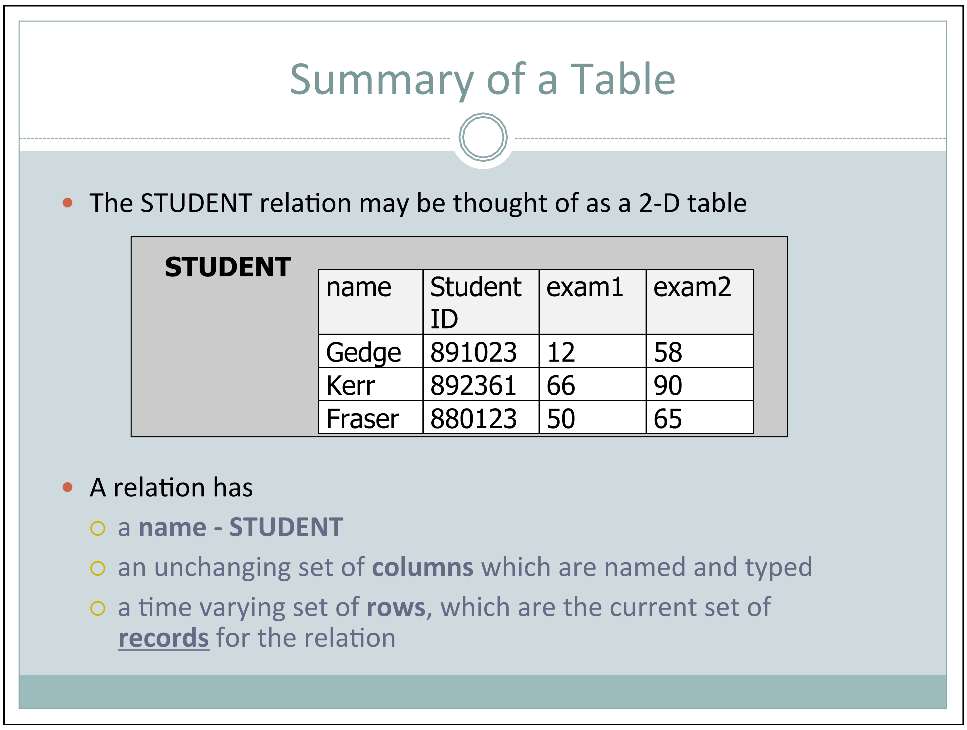
ConsID	MoneySpent	NrItems

BUSINESS PURCHASE

ConsID	BusinessMoney

RESIDENT PURCHASE

ConsID	PurchaseTime	ResidentMoney



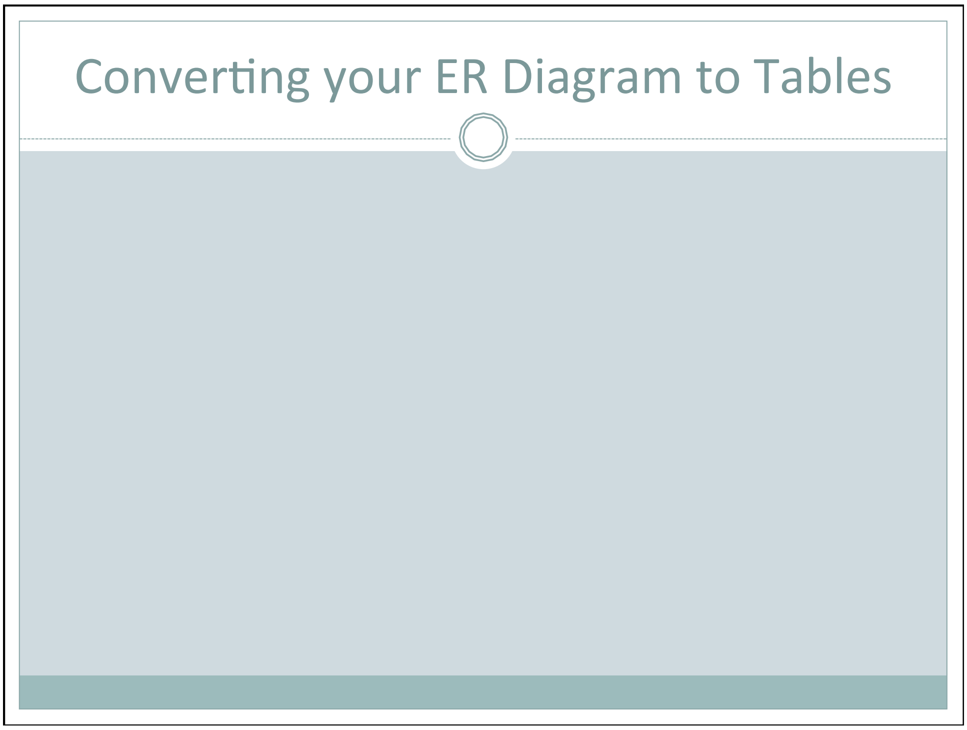
Summary of a Table

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

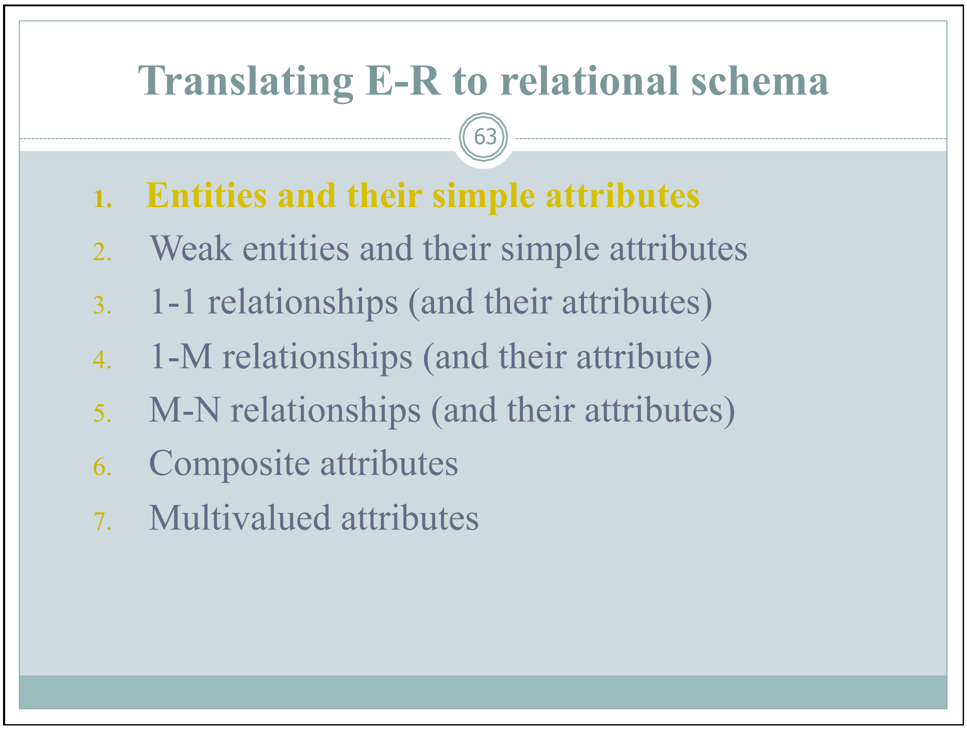
STUDENT

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



Converting your ER Diagram to Tables



Translating E-R to relational schema

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