

ER Diagrams

Entity Relationship Diagrams

DB Concepts

Data Modelling - ER Diagram

1

- This is a **Top-Down** approach to data analysis.
- It is a model of 'Things of interest' to an organisation.
- The ER model refers to a collection of **data** requirements.
- The ER model contains ***Entities, Relationships*** and ***attributes***.

DB Concepts

Data Modelling - ER Diagram

2

ER Diagram Notation

The end product of E-R diagramming is a graphical model of the entities and relationships of a particular domain.

There is no one standard notation for E-R diagramming.

We will use the crows foot (Martin) notation:

DB Concepts

Data Modelling - ER Diagram

3

There are several notations used for producing ER diagrams:

- Chen
- **Crows Foot (Martin)**
- Bachman
- IDEF1X
- Min/Max ISO
- UML (Class Diagram)

DB Concepts

Data Modelling - ER Diagram

4

Entities

An **entity** may be defined as:

- A 'thing' (person / place/object)
- Which the organisation recognises as being capable of an independent existence
- Which can be uniquely identified
- An abstraction of some domain
- Some aspect of the real world

DB Concepts

Data Modelling - ER Diagram

5

Examples.....

- A physical object e.g. A house, a car, a hotel room, a restaurant table, a premiership team
- An event e.g. Sale of a house, rental of a car
- A concept e.g. An order
- Represented on ER diagram as a rectangle (square corners)



DB Concepts

Data Modelling - ER Diagram

6

Entity Vs Entity Type

We must distinguish between an *Entity* and an *Entity Type*.

An *entity type* is a category of a ‘thing’ or object. For example, *Students*, *Houses*, *Rentals*

An *entity* is an *instance* of a given entity type.

Examples....

Consider the entity type *Lecturers* .

An instance of this entity type (ie.an entity) might be:

- *Catherine Woods*
- *John Walsh*

An entity type *Cars* might include instances:

- 12_KY-1234*
- 07_C-6589*

Relationships

A relationship is some association between entities.

We are primarily concerned (for now) with *binary* relationships (i.e. involves only two entities)

More than 1 relationship can exist between two entities.

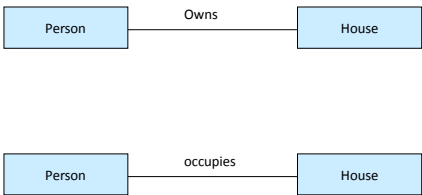
Represented on an ER diagram using a line



Examples...

The entity types *House* and *Person* can be related by:

- *Ownership*
A house *is owned by* a person
(A person owns a house)
- *Occupation* (lives in)
A person *occupies* a house
(A house is occupied by a person)



In practice, many entities are *unrelated*.

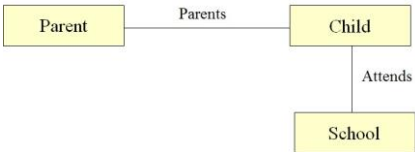
The objective of the ER model is to demonstrate *only direct* relationships.

Consider the entities *Parent*, *Child* and *School*.

What direct relationships might exist between these entities?

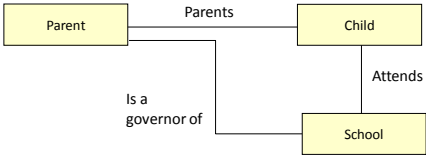
- A Parent *parents* a child.
- A child *attends* a school.

This might be modelled as follows:



If the relationship between Parent and School ‘*Parent is a governor of a school*’ then this is a direct relationship.

This relationship might be represented as follows:



This suggests that every parent is the governor of a school. ???

Attributes

An entity is characterised by a number of *attributes*.

The values assigned to attributes are used to identify one entity instance from another.

- One student from another student
- One car from another car
- One hotel reservation from another reservation

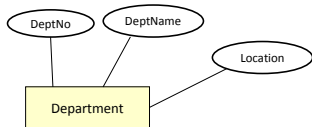
For example, the entity type *Department* might have attributes:

- *DeptNo*
- *DeptName*
- *Location*

Tributes are represented on an ER diagram using an *oval* attached to the entity of which it is an attribute



This might look as follows:



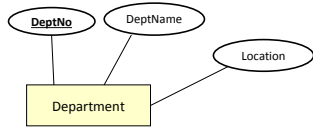
DB Concepts

Data Modelling - ER Diagram

19

One (or more attributes) might uniquely identify each instance of the entity
Department eg. ***DeptNo***

The attribute is underlined.



DB Concepts

Data Modelling - ER Diagram

20

If an entity type has many attributes, the diagram may become overcrowded.

In this event, show ***only*** the identifying attribute on the diagram and include a listing of ***all*** attributes with the diagram.

Diagram is less informative!

DB Concepts

Data Modelling - ER Diagram

21

A simple example

Stock is identified by a stock Number and has a description, cost price, sale price and quantity on hand. Suppliers have a supplier number, a name and a telephone number. Stock is supplied by a supplier.

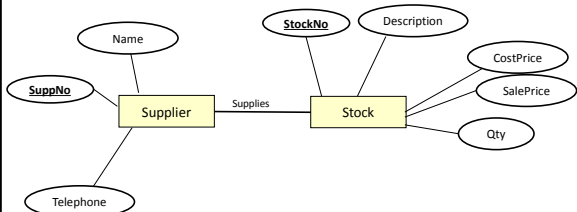
Draw a simple ER Diagram to represent the above data requirements.

DB Concepts

Data Modelling - ER Diagram

22

A solution might be:



DB Concepts

Data Modelling - ER Diagram

23

Semantic Modelling

Data to be represented on an ER model can be classified into different categories:

- Entity
- Relationship
- Attribute

Data might be represented as any of the above

DB Concepts

Data Modelling - ER Diagram

24

Consider information on *marriages* to be stored in the information system.

Marriage may be considered as an *entity type* with attributes Date, Time, Place, Bride and Groom.

Marriage may be considered as an *attribute* (marital status) associated with the entity type *Person*.

Marriage may be considered as a *relationship* between entity types *Man* and *Woman*.

DB Concepts

Data Modelling - ER Diagram

25

It is the job of the Data Modeller (analyst) to decide which of these viewpoints is most appropriate for the data under consideration.

Hence, Top-Down analysis is often referred to as *Semantic Modelling*

DB Concepts

Data Modelling - ER Diagram

26

Crows Foot Notation

As in all graphical representations, meaningful labels should be used for all elements included.

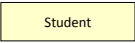
DB Concepts

Data Modelling - ER Diagram

27

Entity

Represented on the diagram by a *rectangular* box:



A meaningful description of the entity is written into this box.

Use a singular noun e.g. *Student* rather than *Students*

DB Concepts

Data Modelling - ER Diagram

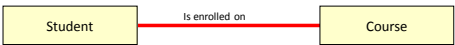
28

Relationship

A relationship between entities is represented by a solid line between the relevant entities.

No direction is shown on this line.

Relationships are usually labelled



DB Concepts

Data Modelling - ER Diagram

29

Relationship labels should be meaningful and *brief*.

This can sometimes be difficult!

Best described by verbs which imply some direction.

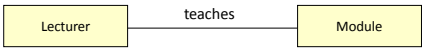
DB Concepts

Data Modelling - ER Diagram

30

The relationship between LECTURER and MODULE might read:

A lecturer *teaches* a module
or
A module *is taught by* a teacher



If a diagram is complex (lots of entities and relationships!)

It might be better to give each relationship a number eg. R1, R2, R3,...etc

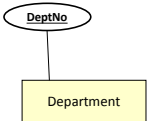
This number can then identify a description of the relationship in the data dictionary
eg. R1: A lecturer teaches a module

The diagram is not as meaningful

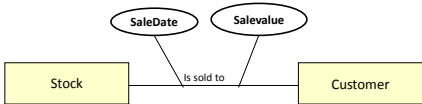
Attribute

An attribute is represented by an *oval* attached by a line to the appropriate entity.

The entity identifier is **underlined**.



A relationship can also have attributes



Rules Defining ER Model

There are two sets of rules that must be added to any relationship:

- Cardinality
- Participation

Cardinality

Cardinality concerns the number of *instances* of an entity type involved in a relationship.

A relationship can have the following cardinality:

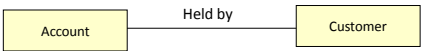
1:1 1:M M:1 M:N

Expressing the cardinality of a relationship means expressing two assertions about it.

The relationship between Bank Accounts and Customers can be said to be 1:1(one-to-one)

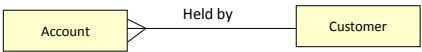
if

- A bank account is held by at most one customer
- A customer may hold at most one bank account



The relationship between Bank Accounts and Customers can be said to be 1:M (one-to-many) if it is defined as:

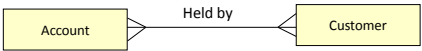
- A customer holds many bank accounts.
- A bank account is held by at most one customer.



The relationship between Bank Accounts and Customers can be said to be M:N (many-to-many) if it is defined as:

- A customer holds many bank accounts.
- A bank account is held by many customers.

Our notation uses a 'Crow's Foot' on the many end of a relationship



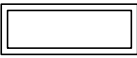
Many-to-many relationships *can not* be transformed into a database schema (we will discuss this later)

Remove any M:N relationships from the diagram.

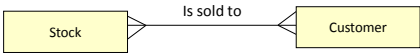
To do this we introduce a **Weak Entity Type** and two 1:M relationships

A **Weak Entity Type** is an entity type whose existence depends on the existence of another two entities.

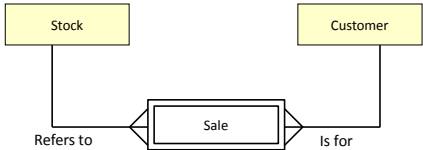
Represented on the diagram as:



The following M:N relationship:



Replaced with:



Participation

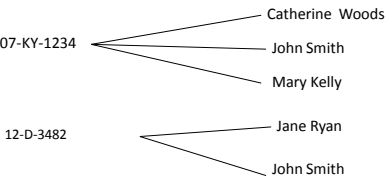
A relationship instance is two entities of one or two types associated by virtue of a defined relationship between them.

An *occurrence* of a relationship.

eg. Catherine Woods *teaches* Software Engineering
eg. 07_KY-123 is rented to John Smith

Example

A department is assigned many lecturers



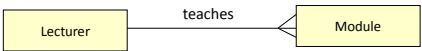
This example illustrates:

2 Entity types: Cars, Customers
7 Entity occurrences (2 cars; 5 customers)
1 relationship type: Is rented by
5 relationship occurrences

Mandatory Membership

Membership of an entity type in a relationship is **mandatory** if each entity of the type **must** participate in an instance under that relationship.

eg. A lecturer teaches *at least* one module



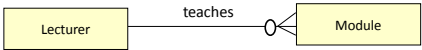
Optional Membership

Membership of an entity type in a relationship is **optional** if entities of the type can exist **without** participating in an instance under that relationship

eg. A lecturer *may* teach some modules
eg. A lecturer may not teach any modules

The **default** participation class of entities in a relationship is **mandatory**.

To indicate optional participation of an entity in a relationship on an ER diagram use the 'optional indicator'



A lecturer teaches none or several modules



A lecturer teaches none or one module

Min / Max Participation

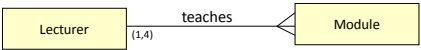
Consider the information:

A lecturer teaches at least 1 module but not more than 4 modules

How might this be represented on an ER diagram?

Place a *Min/max* indicator on the diagram.

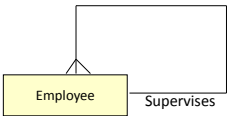
(1, 4) indicates a minimum of 1 and a maximum of 4 participations by an entity in a relationship.



Involuted Relationships

A relationship in which the two entity types of the relationship are from the same entity type

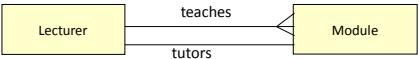
- An employee supervises many employees
- An employee is supervised by an employee



Parallel Relationships

Two or more relationships which exist between the same two entity types.

- A lecturer tutors one module
- A lecturer teaches many modules



Super/Sub Entity Types

A collection of entities of the same type to which:

- a ‘narrower’ definition applies
- additional attributes and/or relationships apply.

A person is either a lecturer or a student.
Student and *Lecturer* are *subtypes* of *Person*.
Person is known as a *supertype*.

All of the attributes of *Person* apply to both *Lecturer* and *Student*.
The converse is not true!

A lecturer might have attributes that a student does not have and vice-versa.

Consider the following attributes:

Person Id, Name, Address

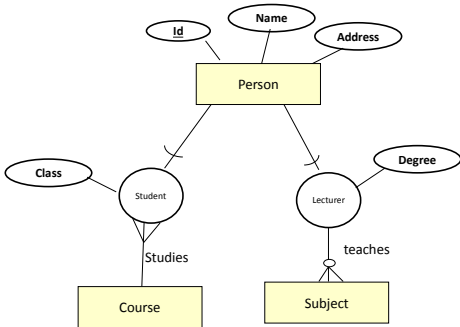
Lecturer Id, Name, Address, Degree,...

Student Id, Name, Address, Course,...

Lecturer would participate in the relationship *teaches*

Student would participate in the relationship *studies*

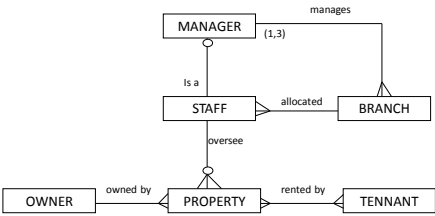
Diagram.....



Example:

Staff are allocated to a branch. A manager manages at least one and not more than 3 branches. Some staff members are managers. Staff oversee properties for rent. Some staff oversee no properties. A property is owned by an owner. An owner may have more than one property. A tenant rents at least one property. A property is rented by several tenants over time.

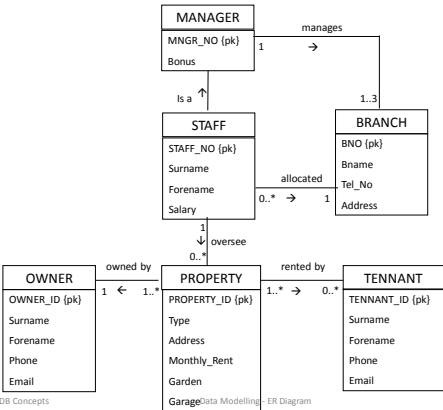
Draw an ER diagram to represent this system



Note: have not included attributes

One form of the UML Class diagram allows us to list the attributes for each entity (class) in a concise way.

See the following diagram:



In either event, there is a many-to-many relationship between *Property* and *Tenant*.

This must be replaced with a weak entity type if we are to transform the model to a relational data model.

