

Information engineering

THE RELATIONAL DATABASE

An excel database as a single list

<u>Course No</u>	<u>Course Title</u>	<u>Start Date</u>	<u>Attendee 1</u>	<u>Attendee 1 Dept</u>	<u>Attendee 2</u>	<u>Attendee 2 Dept</u>	<u>Attendee 3</u>	<u>Attendee 3 Dept</u>	<u>Attendee 4</u>	
IND1	Staff Induction	25/11/2019	Fredd Jones	Accounts	Jane Smith	HR	Janet John	IT setvices		
FIR1	Fire Safety	05/11/2019	David Davie	Production	Joanne Blog	Marketing				
ACCN1	Accounts system	10/09/2019	Fred Blogs	Accounts	Mary Smith	HR	David Davie	Production	Sally Smith	
IND1	Staff Induction	02/01/2020								

Problems with This approach

Redundant Data

Inconsistent Data

Inflexibility

Limited sharing of data

No inherent control of data quality

Changing data can lead to unexpected results

A relational database approach

To increase your chances of implementing a quality information try following some basic steps

- Decide what is the purpose of your database
- Determine the information requirements of the database
- Produce a logical model of the information requirements of the database
- Convert the data model to a physical data model appropriate to the technology being used
- Implement the physical design using the chosen product.

Decide what is the purpose of your database

Relational databases are the most widely used by organisations of all sizes to store data. They offer simplicity of understanding for users and processing power. Many individuals also use relational database packages, such as Microsoft Access, on personal computers, as a means of storing and organising information relating to their personal affairs. The remainder of this text refers only to Relational Databases.

Determine the information requirements of the database

- Even if you think you know the procedure you wish to automate always take a step back - do not take bad procedure to the new system.
- Collect data about the requirements
- Get general information about the organisation
- Get specific information about the activities of the department
- Retrieve any existing systems documentation and any other documents used within the organisation.

Produce a logical model of the information requirements of the database

- Before you can begin to implement the requirements of a database you need to make some sense of the complexity of the information you may have gathered.
- A model is an abstract representation of what exists in the real world.
- Entity Relationship diagrams are one of a number of methods for representing the data requirements for an information system.

The Entity-Relationship Diagram

The Entity-Relationship (E-R) Diagram is a design *of* the database which represents all of the information requirements *of* a system. As with all designs, there may be many possible solutions to the design *of* an E-R Diagram, some better than others, but none necessarily either right or wrong. The DBA is responsible for the design of the E-R Diagram, working in collaboration with systems analysts and users.

The components of the E-R Diagram are:

Entity: an entity is something of interest to the organisation about which information needs to be stored. It can be a physical object such as a person or product, an event such as a lecture, or an intangible object such as a course. An entity will eventually be a table in the physical database. When naming entities, names which are meaningful to the user should be chosen, and the singular should always be used. In an E-R Model entities are represented as rectangles.

Relationship

Relationship: a relationship can be defined as the association between two entities.

There are, in fact, two relationships between each two related entities. For example:

CUSTOMER places ORDER the relationship name is 'places'

CUSTOMER books ROOM the relationship name is 'books'

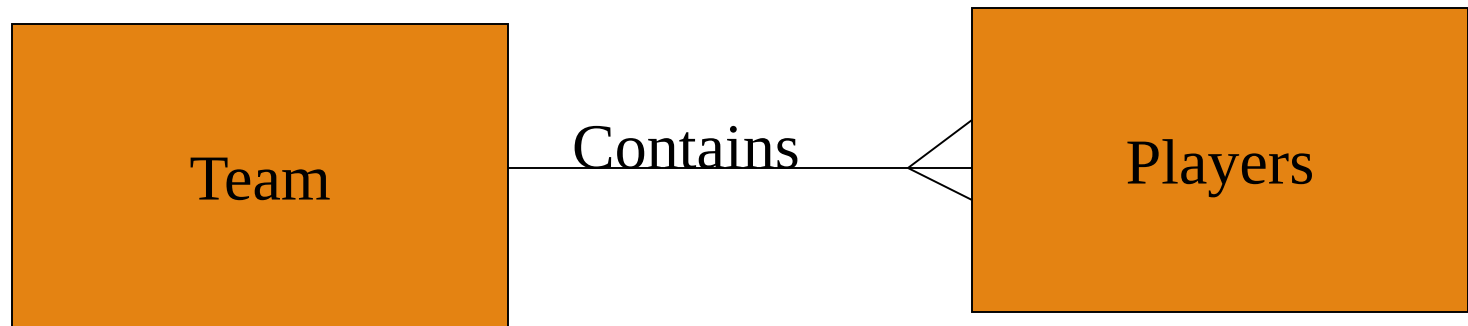
Types of relationship

There are three types of relationships which are:

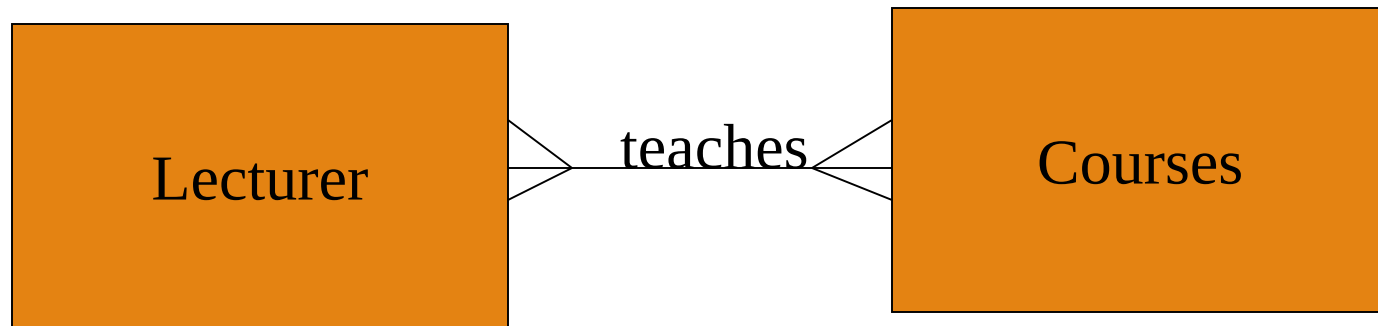
one-to-one



One to many



Many to Many



Convert the data model to a physical data model appropriate to the technology being used

Implement the physical design
using the chosen product

The Relational Model

Data Model

Integrated collection of concepts for describing data, relationships between data, and constraints on the data.

Has three components:

- a structural part;
- a manipulative part;
- a set of integrity rules.

RM Terminology

Relation: table with columns and rows.

Attribute: named column of a relation.

Domain: set of allowable values for one or more attributes.

Tuple: a record of a relation.

Relational Database - collection of normalized relations with distinct relation names.

Alternative Terminology

Relation, attribute, tuple

Table, column, record

File, field, row

Combinations thereof.

Properties of Relations

Table name is distinct from all other table names in the database.

Each cell of table contains exactly one atomic (single) value.

Each column has a distinct name.

Values of a column are all from the same domain.

Properties of Relations

- Each record is distinct; there are no duplicate records.
- Order of columns has no significance.
- Order of records has no significance, theoretically.

Entity Relationship Modeling

Entity Relationship Modeling (ERM) is one of the earliest techniques associated with data analysis and database design.

Basically, an Entity Relationship Model is exactly what it says it is:

- A model of an information system in terms of *Entities* and the *Relationships* between the entities.

Entity & Entity Types

An entity may be defined as:

- A thing which the enterprise recognises as being capable of independent existence and which can be uniquely identified.
- An entity may be an object such as a:
 - *house, car service, video, video rental, video reservation,*
 - *person, car part, order, product, etc.*

Basic Personnel System

Typically in a basic personnel system there will exist the following entities:

- Person;
- Department;
- Grade;
- Position.

Each of the entities could be described by a group of attributes

Basic Personnel System

Person	Department	Grade	Position
Personnel #	Dept. #	Grade Code	Position Code
Name	Dept. Name	Grade Desc.	Desc.
Address	Managers Name	Company Car	Dept #
Grade Code			Educ. Level
Date of Birth			
Dept. #			

Basic Personnel System (Relationships)

The diagram illustrates the relationships between four tables: Person, Department, Grade, and Position. Orange arrows indicate the following connections:

- Person** to **Department**: A line from the top of the Person table connects to the top of the Department table, with an arrow pointing to the **Dept. #** field.
- Person** to **Grade**: An arrow points from the **Grade Code** field in the Person table to the **Grade Code** field in the Grade table.
- Person** to **Position**: An arrow points from the **Dept. #** field in the Person table to the **Dept #** field in the Position table.
- Department** to **Position**: An arrow points from the **Managers Name** field in the Department table to the **Educ. Level** field in the Position table.

Person	Department	Grade	Position
Personnel #	Dept. #	Grade Code	Position Code
Name	Dept. Name	Grade Desc.	Desc.
Address	Managers Name	Company Car	Dept #
Grade Code			Educ. Level
Date of Birth			
Dept. #			

Relationships

A relationship can be defined as:

- an association between two or more entities

A relationship can be said to exist if an attribute is common to both entities,

- a relationship exists between Person and Grade in the Personnel System shown previously, through the attribute ***Grade Code***.
- ***Grade Code*** is common to both entities.

Notation

The end product of ER modeling, is a model of the entities and relationships in a particular application environment.

- We indicate an entity with a rectangular box.
- We indicate a relationship by drawing a labeled line between the relevant boxes.

Example Notation



The Degree of a Relationship

A relationship can be said to be either of the following:

- 1 : 1 (one to one) relationship;
- 1 : M (one to many) relationship;
- M : N (many to many) relationship.

The Degree of a Relationship

One to One Relationship

- The relationship between a salesperson and a customer can be said to be
- 1 : 1 if:-
 - a salesperson has at most one customer;
 - a customer interacts with at most one salesperson.



The Degree of a Relationship

One to Many Relationship

- The relationship between a salesperson and a customer can be said to be 1 : M if:-
 - a salesperson has many customers;
 - a customer interacts with at most one salesperson.

The crows foot is used to represent the many end of the relationship.



The Degree of a Relationship

Many to Many Relationship

- The relationship between a salesperson and customer can be said to be M : N if:-
 - a salesperson has many customers;
 - a customer interacts with many salespersons.

The crows foot is used to represent the many end of the relationship.



Membership Class

Suppose the relationship between a patient and a hospital ward can be detailed as follows:

- every patient must be allocated to a ward;
- a ward may exist without any patients.

The membership of patient in the 'allocated' relationship is **mandatory** (every patient must be allocated to a ward).

The membership of the ward in the allocated relationship is **optional** (a ward may exist without any patients).

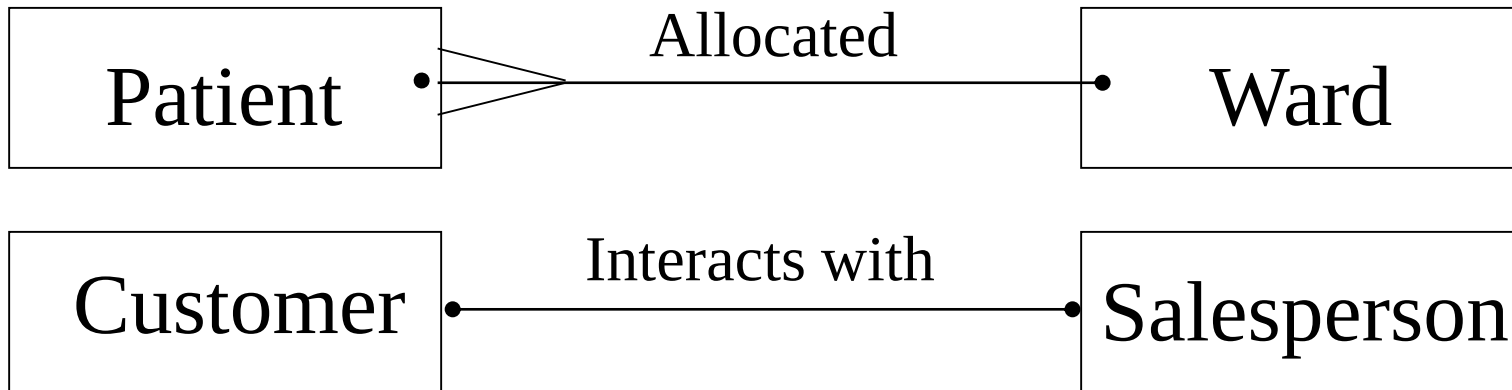
Membership Class Notation

Mandatory membership

- in a mandatory membership class, we draw a dot inside the entity symbol (rectangle).

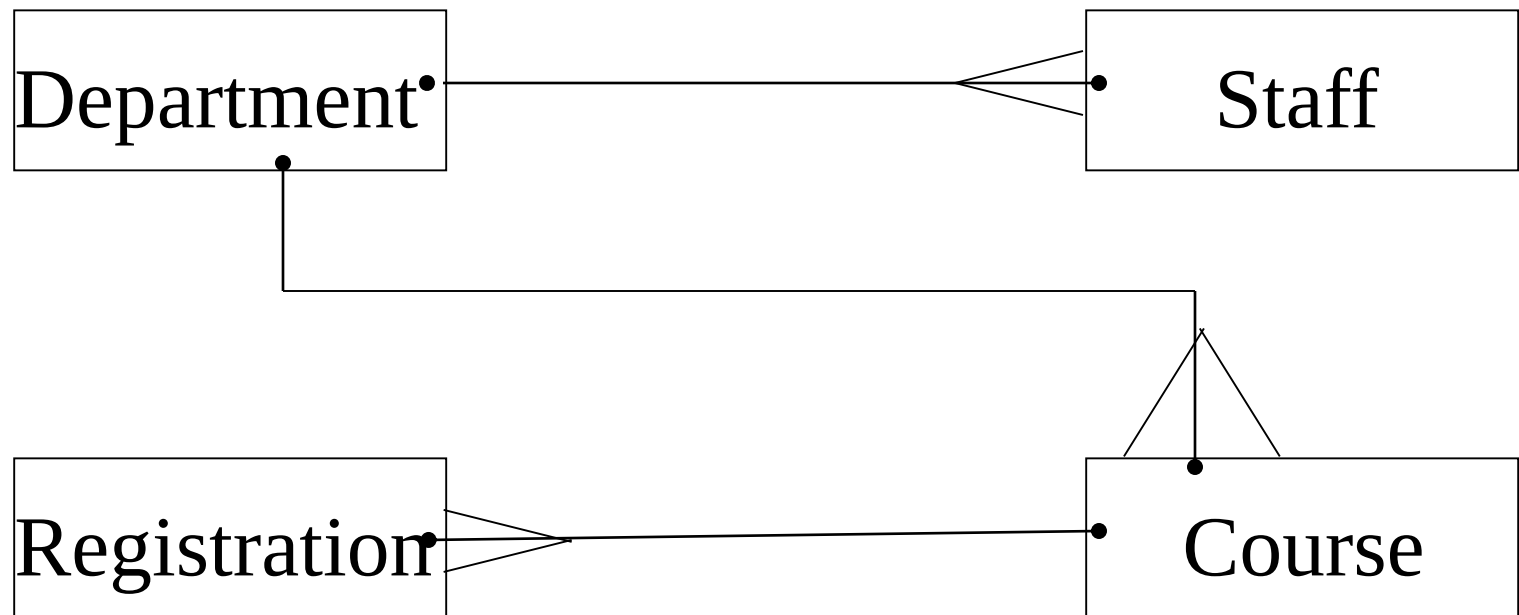
Optional Membership

- in an optional membership class, we draw a dot on the relationship line, prior to any crows foot (used to indicated a many side of a relationship).



Validating an ER Model Against Requirements

Suppose we have the following ER model for an Educational Establishment:



Validating an ER Model Against Requirements

If we wish to validate this model against a system requirement:

- **determine the staff student ratio for a given department**

To perform this analysis we need to access:

- all staff of a department;
- each member of staff associated with a course;
- each course taught by a member of staff;
- each registration of a student for a course in a department.

Validating an ER Model Against Requirements

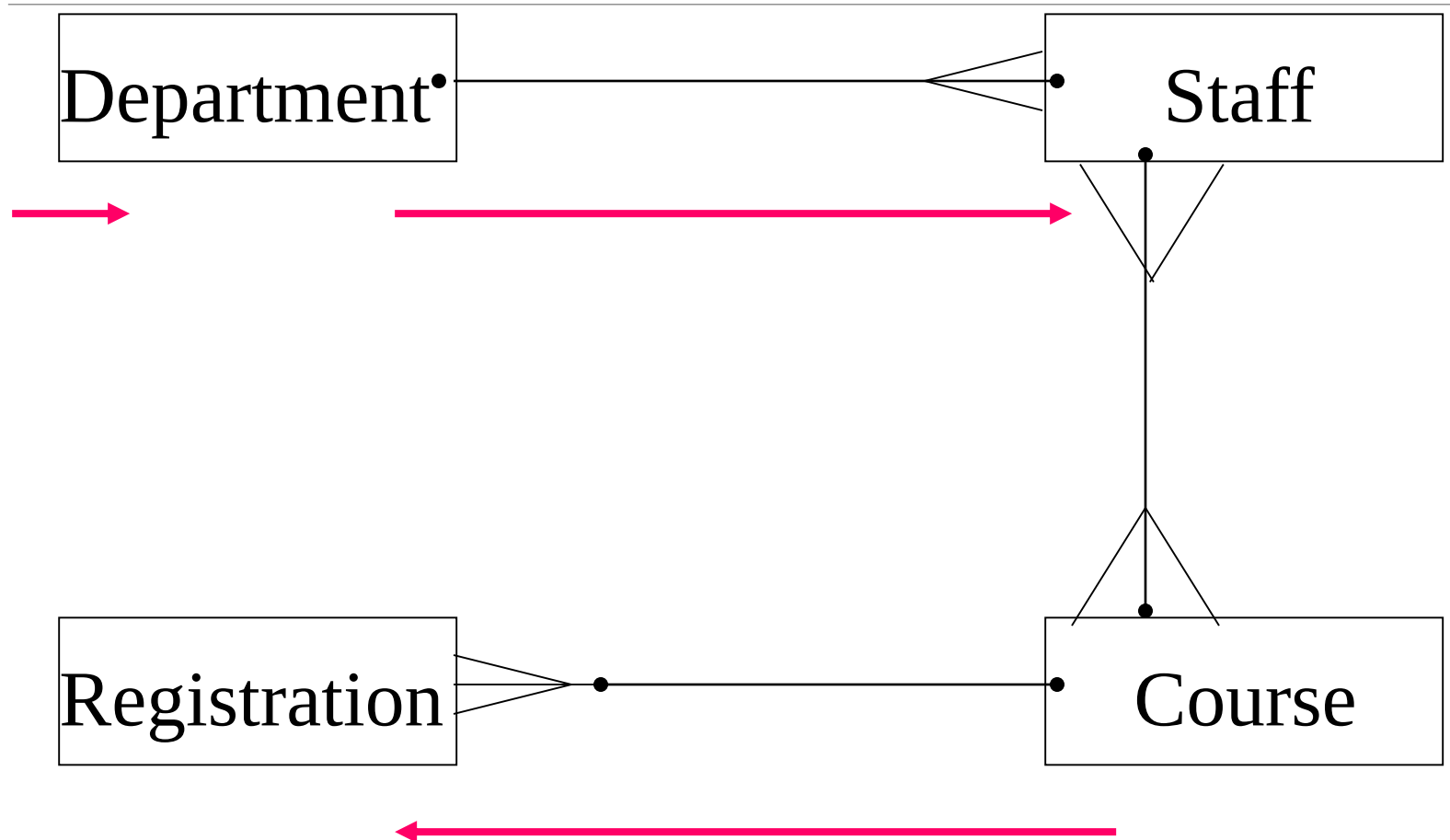
The first ER model is adequate as a data model but is inappropriate for the functions required of it.

We therefore need to take into account not only DATA ANALYSIS but also FUNCTIONAL ANALYSIS.

The ER model on the next slide is the result of both DATA and FUNCTIONAL ANALYSIS.

Validating an ER Model Against Requirements

The FUNCTIONAL ANALYSIS aspects are depicted in the red arrowed lines

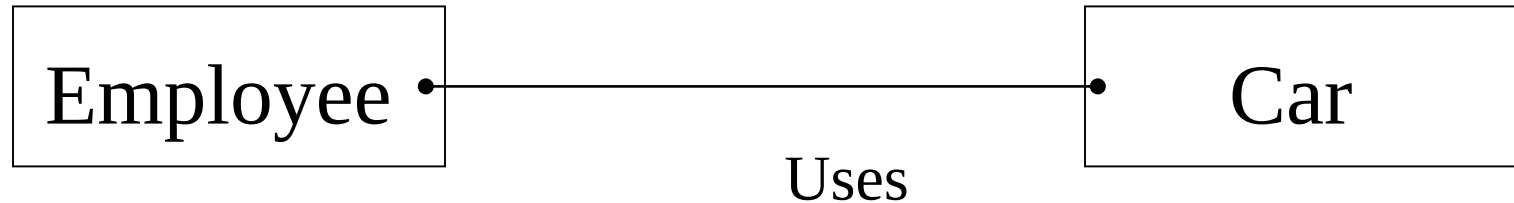


Entity Attribute Assignment

Once we have drawn the ER model we then apply a number of rules for attribute assignment to each entity in our diagram.

1 : 1 Relationship

- Membership mandatory for both entity types.
- Put all attributes in a single table

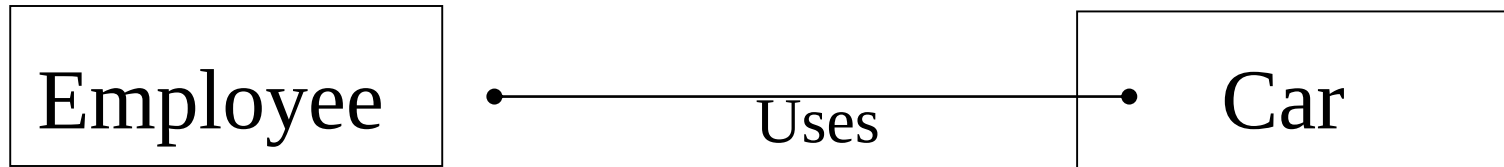


Employee (**Employee#**, ..., ..., ..., Registration #)

Entity Attribute Assignment

1 : 1 Relationship

- Membership mandatory for only one entity type.
- Generate two tables
 - Employee
 - Car
- Post the primary key of the optional table into the mandatory table



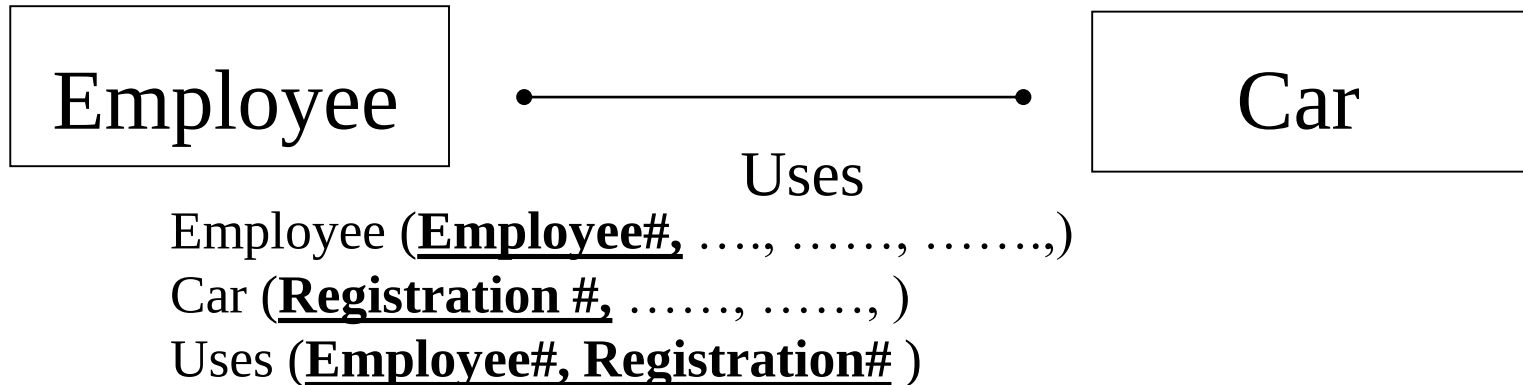
Employee (**Employee#**, ..., ..., ...,)

Car (**Registration #**, ..., ..., Employee#)

Entity Attribute Assignment

1 : 1 Relationship

- Membership optional for both entity types.
 - Generate three tables
 - Employee
 - Car
 - Uses
 - One for each entity and one for the relationship



Entity Attribute Assignment

Note:

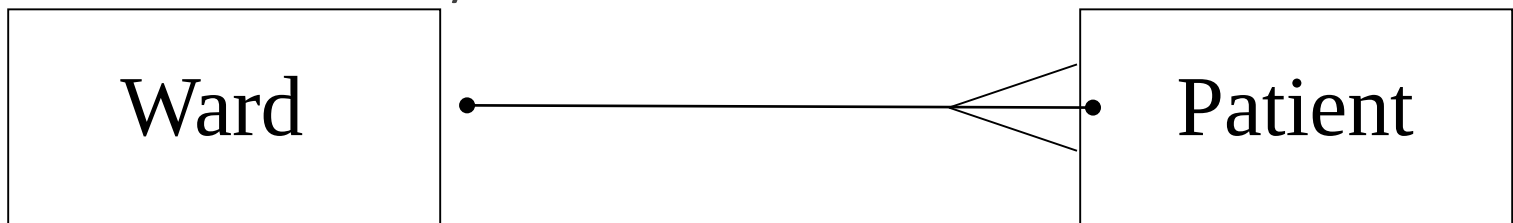
- post primary keys from both
 - Employee and
 - Car
- tables to the relationship table
 - Uses

Employees do not necessarily have cars, and cars, perhaps those newly acquired are not necessarily used by company employees.

Entity Attribute Assignment

1 : M (one to many) Relationship

- Membership of the 'many' entity is mandatory and membership of the 'one' entity is optional.
- Generate two tables
 - Ward
 - Patient
- One for each entity.



Entity Attribute Assignment

Note:

- post the primary key from the 'one' entity to the 'many' entity:
- from Ward
- to Patient

Ward (Ward no, Ward_name, ..., ..., ...,)

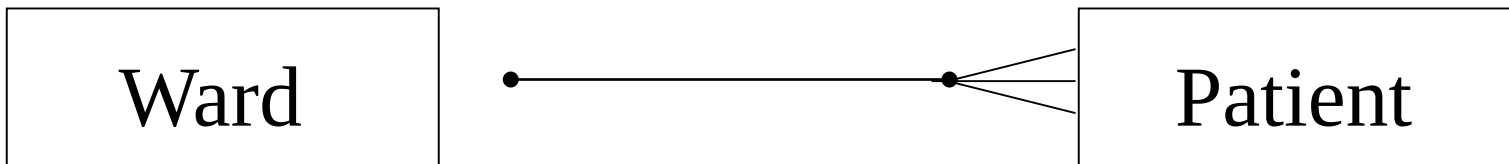
Patient (Patient no, ..., ..., ..., ..., Ward_no)

- A ward may be empty
- A patient must be assigned to a ward
- i.e. there are no out-patients

Entity Attribute Assignment

1 : M (one to many) Relationship

- Membership of both the 'many' and 'one' entity is optional.
- Generate three tables
 - Ward
 - Patient
 - Contains
- One for each entity and one for the relationship



Entity Attribute Assignment

Note:

- post the primary key from both the Ward and Patient entities to the relationship (Contains) table.

Ward (Ward no, Ward_name, ..., ..., ...,)

Patient (Patient no, ..., ..., ..., ...,)

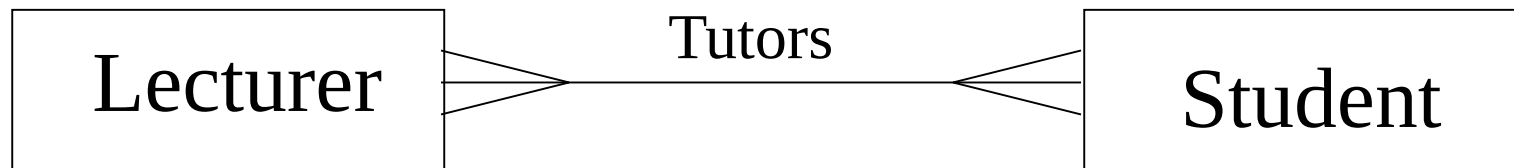
Contains(Ward no, Patient no)

- A ward may be empty
- A patient may be assigned to a ward or
- A patient may be treated as an out-patient

Entity Attribute Assignment

M : N (many to many) Relationship

- Membership of both the entities is irrelevant.
- In all cases generate three tables
 - Lecturer
 - Student
 - Tutors
- One for each entity and one for the relationship



Entity Attribute Assignment

Note:

- post the primary key from both the Lecturer and Student entities to the relationship (Tutors) table.

Lecturer (Lecturer no, Lecturer name, ..., ..., ...)

Student (Student no, ..., ..., ..., ...)

Tutors(Lecturer no, Student no)

- A Lecturer may tutor many Students
- A Student may be tutored by many Lecturers

Conclusion

E-R modelling involves the following steps:

- identify an initial set of entities from the environment;

- investigate and record inter-relationships;

- draw a diagram to represent entities and relationships;

- indicate the degree of the relationship on the diagram;

- indicate the degree of membership class of each of the relationships on the diagram;

- validate the structure against system requirements;

- rationalise the structure;

- revalidate;

- convert the diagram into a set of fully normalised tables.

Conclusion

E - R modelling is primarily a top-down design technique for use in both the analysis and design phase of a systems project and also in the design of databases. An E-R model comprises of a conceptual model of a system or database.

In the design of a database we can derive a set of logical record or table structures, which represent the database in a normalised form. The final step is to convert these logical record structures into physical record structures. This involves two procedures:

- validating the model against system requirements
- convert the logical table structures into a form appropriate for a particular database model (Hierarchical, Network or Relational).

Relational Keys

Superkey

- A column, or a set of columns, that uniquely identifies a record within a table.

Candidate Key

- Superkey (K) such that no proper subset is a superkey within the table.
- In each record, values of K uniquely identify that record (*uniqueness*).
- No proper subset of K has the uniqueness property (*irreducibility*).

Relational Keys

Primary Key

- Candidate key selected to identify records uniquely within table.

Alternate Keys

- Candidate keys that are not selected to be primary key.

Foreign Key

- Column, or set of columns, within one table that matches candidate key of some (possibly same) table.

Relational Integrity

Null

- Represents value for a column that is currently unknown or not applicable for record.
- Deals with incomplete or exceptional data.
- Represents the absence of a value and is not the same as zero or spaces, which are values.

Relational Integrity

Entity Integrity

- In a base table, no column of a primary key can be null.

Referential Integrity

- If FK exists in a table, either FK value must match a candidate key value of some record in its home table or FK value must be wholly null.

Relational Integrity

Business Rules

- Rules that define or constrain some aspect of the organization.