

# Informatics 1 Data & Analysis

## Tutorial 1

Week 3, Semester 2, 2013–2014

This worksheet has three parts: tutorial Questions, followed by some Examples and their Solutions.

- Before your tutorial, work through and attempt all of the Questions in the first section.
- The Examples are there for additional preparation, practice, and revision.
- Use the Solutions to check your answers, and read about possible alternatives.

You must bring your answers to the main questions along to your tutorial. You will need to be able to show these to your tutor, and may be exchanging them with other students, so it is best to have them printed out on paper.

If you cannot do some questions, write down what it is that you find challenging and use this to ask your tutor in the meeting.

Tutorials will not usually cover the Examples, but if you have any questions about those then write them down and ask your tutor, or go along to InfBASE during the week.

It's important both for your learning and other students in the group that you come to tutorials properly prepared. If you have not attempted the main tutorial questions, then you may be sent away from the tutorial to do them elsewhere.

Data & Analysis tutorials are not formally assessed, but they are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.

Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your tutor, and if possible attend another tutorial group in the same week.

*Please send any corrections and suggestions to [Ian.Stark@ed.ac.uk](mailto:Ian.Stark@ed.ac.uk)*

## Introduction

In this tutorial you are required to design an Entity Relationship Model for a database to be used by the organisers of a poster exhibition in order to keep track of three phases in the exhibition: submission, selection and presentation. You should read the scenario carefully, making a note of all the candidate entities you think are involved, and of their attributes. Think about the relationships between the entities as you do this. The tutorial proceeds with a series of questions about your design, finishing with an ER diagram.

## Scenario: A Poster Exhibition

The setting is that you are one organiser of a poster exhibition on “Global Problems of the 21st Century”, and you must design a database to keep track of the administration of the exhibition. Three main phases are recognised for the exhibition: the submission, the selection and the presentation phase. These are further explained below.

*Submission Phase:* Graphic designers create posters for the exhibition to illustrate one of the chosen global problems. Relevant information on designers includes their name and their affiliation, i.e. the organization they work for. A poster has a title and is assigned an identification number, and it may be created by several graphic designers; although each individual designer may only be involved with one poster. Where a group of graphic designers create a poster, we distinguish between the main designer and the co-designers. In case of a single graphic designer, that person is considered to be the main designer of the poster. The main designer is always the point of contact, so should provide an email address.

*Selection Phase:* All posters created for this exhibition are judged by members of a jury. A judge is a graphic design expert with experience in communication for raising public awareness and for public benefit. Judge information that is of relevance to the organising committee includes the judge's name, their affiliation and email. Each poster is judged by three different judges. When judging a poster, a judge gives a decision: accept or reject. A poster is selected for the exhibition only if all three judges give an "accept" decision. Note that judges are not allowed to compete in "Global Problems of the 21st Century" themselves.

*Presentation Phase:* All selected posters are then presented in the exhibition by their main graphic designers. The poster presentation is allocated a stand and an exhibition session. Each exhibition session takes place at a specific date, and 4 session topics have been announced: human rights, environmental pollution, poverty, and war.

### **Question 1: Determining possible entity-sets**

What are the candidate entity sets you would consider in order to build your ER model? Choose your final set of entity sets from these candidates, making sure to include 'Judge', 'Poster' and 'Graphic Designer'.

*Things to think about:* Is it possible to map entity sets directly and unambiguously from the information given, or is there more than one possibility for modelling these entity sets?

### **Question 2: Defining attributes for a given entity-set**

By now you will have decided your sets of entities that can be used to build the ER diagram. For this question, focus on just the two entity sets below:

- Judge
- Poster

What are the attributes you can assign to each of these given the available information? What are the attribute domains?

Note: In order to draw your diagram you will need to decide the attributes for *all* your ER objects. You do not need to list them all here, but they must appear on your final diagram.

*Things to think about:* According to the scenario given, a judge gives a decision on a poster: accept or reject. Is this an attribute? If not, what is it? If it is an attribute, does it belong to Judge or to Poster or to neither of them? If it belongs to neither, where does it feature in the ER design?

### **Question 3: Describing relationships**

Define relationships between the entities you defined in the first question. What kinds of relationships are these? *one-to-one*? *one-to-many*? *many-to-many*? Are there any weak entities in your model? Are there any ISA hierarchies? Are there any key constraints? Do any of these relationships involve total participation?

### **Question 4: Defining primary keys**

Choose primary keys for each entity in your model. Which ones did you choose, and why?

## Question 5: Drawing the diagram

Draw the diagram representing your ER model by hand on paper.

## Question 6: Using the dia tool

Draw the ER diagram you created in Question 5 using the `dia` tool and following the instructions below. DICE machines have `dia` installed, and the package is also freely available for a range of platforms. Print out the result and bring it to your tutorial.

**Q.** Can I use a different tool?

**A.** Sure, but first do it with `dia`. Then, if you like, try another tool (like yEd, Calligra Flow, Lucid-Chart, Creately, ...). How does it differ from `dia`? Can you identify any advantages or disadvantages for drawing ER diagrams?

- (a) Type `dia` at the command line in a terminal window. (You can run it in the background by typing "`dia &`" — ask a Lab demonstrator if you don't know what this means.)
- (b) You will see a diagram window and a **Diagram Editor** window with tool buttons. In the middle of the tool buttons window there is a pull-down menu (which may be labelled **Misc** or **Assorted** initially). Click on that menu and select the option **ER**, either from the list if it's present, or from the **Other Sheets** sub-menu. Once you've done this, you will see a set of buttons for drawing entities, relationships and so on, just below the pull-down menu.
- (c) The blue lines on the diagram window are page-delimiters. By default they are set as size A4 with portrait orientation. For landscape use the **File - Page Setup...** menu option.
- (d) To draw an Entity, select the **Entity** drawing tool (first one on the left, denoted with a square labelled "E") and click anywhere on the diagram window. Double click on the Entity you just created and explore the different options you can use for customising entities. Once you have played with these options, set the final name of this entity to be **Poster**.
- (e) In the same way, but using the **Attribute** tool (a circle with the letter "A") define the attributes for this entity. When you double click on a defined attribute you also have a number of options for customisation. Spend some time exploring these as well.
- (f) In order to connect objects, select the icon with two parallel lines and then click on the screen. Then click and drag one of the green ends of the line and connect it to one of the objects (when you see a red line around the object it means the connection is made and you can release the mouse). Do the same for the other end of the line. The small square in the middle of the line allows you to change the shape of the line. Notice that the connection lines will follow if you move the objects around — this makes it easy to lay your diagram out neatly.
- (g) If you prefer a straight line connection to a zig-zag, use the **Line** button from the top part of the tools window. You can also add arrows and change the width of lines using the options at the bottom of the window. These are useful for showing key dependencies and total participation relationships. Experiment with these options.
- (h) There is also a tool for drawing relationships (the button icon contains a letter "R"). Explore how it works and define the relationships between entities.
- (i) Build up your poster exhibition ER diagram by adding and connecting all the entities, attributes and relationships from the model you designed in the tutorial.
- (j) Editing tips:

*Deleting:* If you want to delete an object, click on it and press the `delete` key. You can also undo the last action performed, by clicking **Edit - Undo**.

*Scaling:* Objects created in `dia` by default are quite large; you will want to make them smaller for printing. The simplest way to achieve this is to lay out your diagram on the screen without worrying about the size, and then — once it is complete — use the `File - Page Setup...` menu, `Scaling` option, and scale the diagram to fit in a single page (i.e. 1 x 1).

(k) To print your diagram, use the `File - Print Diagram...` menu.

### **Tutorial Discussion: Thinking around the model**

- According to the scenario: ‘A poster ... may be created by several graphic designers. However, each individual designer may only be involved with one poster.’ Can you model this in your ER diagram? If yes, how?
- Can we model that every poster has a single main designer?
- According to the scenario: ‘Each poster is judged by three different judges.’ Can you model this in your ER diagram? If yes, how?
- According to the scenario: ‘A poster is selected for the exhibition only if all three judges give an “accept” decision.’ Can you model this in your ER diagram? If yes, how?
- According to the scenario: ‘...judges are not allowed to compete in “Global Problems of the 21st Century” themselves.’ Can you model this in your ER diagram? If yes, how?
- Based on your answers for the above questions, what do you think about the expressivity of the ER model?
- Based on your answers (and your classmates’ answers) for this tutorial, is there a natural “best” solution for this ER model?
- If there is more than one correct answer, how can you be sure that your answer is correct? Any suggestions on how to check?
- So far in Informatics 1 you’ve seen several different modelling formalisms, such as the ER model, Propositional Logic and Finite State Machines. Based on your experience, for what kinds of situations/scenarios would you use one over the other? What aspects do these modelling systems have in common? How do they differ?

## Examples

This section contains further exercises on entity-relationship modelling. Examples 1 to 6 run in a sequence very similar to the main tutorial questions, designing an ER model for a given scenario. Examples 7 to 10 are taken from past exam papers.

Following these there is a section presenting solutions and notes on all the examples.

### An Inter-University Gliding Competition

For these examples you are required to design an entity-relationship model for a database based on a textual description of a particular scenario. You should read the description carefully, making a note of all the entities you think are involved, and of their attributes. Think about the relationships between the entities as you do this. After this there are a series of questions to help you develop your design and then produce a final ER diagram for it.

The scenario is that you are organising an inter-university gliding competition, and you have decided to design a database to keep track of the administration of the competition.<sup>1</sup>

A number of universities have each entered a team in the competition (known as a gliding *Task Week*), and one of the things you need to keep track of is whether or not they have paid the entry fee. Each university team consists of a variable number of people who will take part in the competition; everybody who competes must be a member of one of the teams.

The pilots will have different levels of experience. Some will be *pre-solo*, which means they can only fly as second pilot (known as *crew capacity* “P2”) in a two-seater glider. They can still compete for their team in this capacity, as long as there is an instructor flying with them as pilot-in-charge (crew capacity “P1”). Pilots who are of *cross-country* standard can fly as P2 just like pre-solo pilots, but may also fly on their own (flying “solo”) in any kind of aircraft. A pilot flying solo is always P1. Pilots who are *instructors* can fly solo in single- or two-seater gliders, or as P1 in a two-seater with a less experienced pilot. If two instructors are flying together they will simply decide between them who is P1 and who is P2.

There are a number of different types of glider involved in the competition. Some are two-seaters, such as K7, K13, K21 and DG505. The rest are single-seaters; their types include K8, Pirat, DG300, Discus and LS4. There may be more than one glider of a particular type, but every glider can be distinguished by its *callsign* — a short string which is used to identify it in radio communications. Typical callsigns include “MF”, “P19”, “FNS” and “CPG”.

The competition is organised around *tasks*, which are routes that each competing glider must attempt to fly around. On each competition day a task is set for the pilots to fly in their gliders. The task is defined by choosing a set of *turning points* taken from a list available from the BGA (British Gliding Association). There are almost a thousand such turning points defined for the UK, and each has a unique *trigraph* or three-letter acronym to identify it. For example, “STI” is for Stirling and “LOM” is the Lake of Menteith. The competition is to be held at Portmoak Airfield (about 30 miles north of Edinburgh), which is “POR”. The task-setter will decide on a suitable task for each competition day, which will involve trying to glide from the starting point at the airfield around one, two or more turning points. For example, a set like “POR, STI, MVN” would define a triangle of just over 100km, with the corners at Portmoak, Stirling and Methven (which is near Perth). For the purposes of this example we will assume that competitors are allowed to fly around the turning points in any order they choose. As well as specifying the trigraph, the BGA list of turning points gives the latitude and longitude of each turning point, so their positions can be precisely identified on a map.

Sometimes competitors can gain an unfair advantage by starting off much higher than other gliders, or by happening to pick a better time of day. The task-setter can therefore attach conditions to each task, specifying the maximum starting height allowed and the earliest time at which a glider can start.

---

<sup>1</sup>Gliders are aeroplanes without engines, designed for maximal efficiency in extracting the energy freely available in the air, and (on a good day) using it to fly hundreds of kilometres.

### Example 1: Determining possible entity-sets

What are the entity sets you would consider in order to build your ER model? Is it possible to map these entity sets directly and unambiguously from the information given, or is there more than one possibility for modelling these entity sets? Choose your final set of entity sets, making sure to include both ‘Person’ and ‘Glider’, and justify your answer in as much detail as possible.

### Example 2: Defining attributes for a given entity-set

At this point you will have decided your sets of entities that can be used to design the ER diagram. For this question, focus on just these two entity sets:

- Person
- Glider

What are the attributes you can assign to each of these given the available information? What are the domains of these attributes?

The scenario mentioned that a pilot can fly a glider in crew capacity either P1 or P2. Is this an attribute? If not, what is it? If it is an attribute, does it belong to Person or to Glider or to neither of them? If it belongs to neither, where does it feature in the ER design? Explain your reasoning.

Note: In order to draw your diagram you will need to decide the attributes for all of your ER objects. You do not need to list them all here, but they must appear on your final diagram.

### Example 3: Describing relationships

Describe the relationships between the entities you defined in the first question and give them names. What kinds of relationships are there? *one-to-one*? *one-to-many*? *many-to-many*? Are there any weak entities in your model? Are there any key constraints? Do any of the relationships involve full participation?

### Example 4: Defining primary keys

Choose primary keys for each entity in your model. Explain briefly why you picked those.

### Example 5: Drawing the diagram

Draw the diagram representing your ER model.

### Example 6: Using dia

Draw the ER diagram you created in Question 5 using the `dia` application.

### Discussion: Thinking around the model

Do not alter your diagram for this question, but describe in outline changes that you might make (such as adding certain attributes to particular entities or relationships).

Imagine you really are running a gliding competition. Is there anything you think is missing from the scenario — any other information that you would like to collect and add to your model? Explain what you would add and why. Could your model be easily altered to include new information or would you need a major redesign?

## Example 7

The craft trading website *Itsy! Bitsy!* is setting up a database to record sellers and their products. This requires recording the following information:

- For each seller, their name, contact email, and postal address.
- For each product, its name, price, and number available.
- Which product is from which seller.
- A unique id number for each product.

Draw an entity-relationship (ER) diagram that represents this information. Make sure to capture the constraints on the relationships involved, and designate appropriate primary keys for the entities.

## Example 8

The organisers of the *EXAM 2011* international multi-conference need to keep track of a large collection of workshops associated with the event. Initial requirements analysis brings out the following information about what needs to be recorded.

- Each workshop has a name, and happens on a particular date — or dates, as some workshops last more than one day.
  - There are several participants, each of which may sign up to one or more workshops.
  - For each participant, it is important to record their name, email address, and the workshops which they wish to attend.
  - There are a number of meeting rooms at the conference venue, each of a fixed capacity. Meetings rooms are identified by a floor and room number.
  - Every workshop needs an allocated meeting room; where a workshop lasts for two days, it will use the same room on both days.
- (a) Draw an entity-relationship diagram suitable for representing this information, in particular the connections between participants, workshops, rooms, and dates.
- (b) For each of the following concepts give a brief description of what it means, and give an example from your ER diagram for the previous part.
- (i) Key
  - (ii) Composite key
  - (iii) Total participation
  - (iv) Key constraint

How is total participation shown in an ER diagram? How is a key constraint shown?

- (c) Further analysis reveals additional requirements. However, not all of these can be captured easily in an ER diagram.
- Each workshop must have an identified organiser among the conference participants.
  - No participant may register for two workshops on the same day.
  - Every participant must register for at least one workshop.
- (i) Identify two of these which can be captured in an ER diagram.
- (ii) For those two, show the additions required to your diagram.

### Example 9

A database is to be set up to record details of experts on different subjects. The database will be used to contact experts to be called as witnesses in court cases. The following information needs to be recorded.

- For each expert, their witness identity number, name, affiliation (e.g., “Edinburgh University”), and email address.
  - For each field of expertise, the name of the field (e.g., “DNA forensics”).
  - For each expert, all the fields of expertise that they are expert in.
  - For each field of expertise, at most one identified leading expert.
- (a) Draw an ER diagram that expresses the requirements for the database. Make sure that you capture all the constraints on the data mentioned above.
- (b) Are there any other natural constraints one might impose on the data that are not captured by the requirements above? For each such constraint, say whether it would be possible to modify your ER diagram to include the constraint, and, if so, explain how this would be done.

### Example 10

A university wants to set up a database to record details about its staff, and the departments they belong to. They intend to record the following information.

- For each member of staff, their staff identity number, name, job title, and salary.
  - For each department, its name and address.
  - For each member of staff, all departments that they belong to. It is required that every member of staff belongs to at least one department.
  - For each department, the head of department. It is required that each department has exactly one head of department.
- (a) Draw an ER diagram that expresses the requirements for the database. Make sure that you capture all the constraints on the data mentioned above.
- (b) Are there any other natural constraints one might impose on the data that are not captured by the requirements above? For each such constraint, say whether it would be possible to modify your ER diagram to include the constraint, and, if so, explain how this would be done.



## Solutions to Examples

These are not entirely “model” answers; instead, they indicate a possible solution, together with some comments on what features are relevant and where there might be trade-offs between different alternatives.

Remember that not all of these questions will have a single “right” answer. There can be multiple appropriate ways to design an ER diagram, each with particular advantages or disadvantages. Even where these notes include more than one solution, it still cannot cover every possible correct alternative.

If you have difficulties with a particular example, or have trouble following through the solution, please raise this as a question in your tutorial.

### Solution 1

Figure 1 on page 10 shows a possible ER diagram. That uses the following entities: Team, Person, Glider, Task and TurningPoint. Using different names for the same things — such as University instead of Team, or Pilot instead of Person — would give an equivalent solution.

### Solution 2

In the solution of Fig. 1, the Person entity has attributes `personId` (an integer assigned to each person), `name` (string) and `experienceLevel` (string). Other possibilities might be to split the name into first name and last name, or to add further information like an email address. There needs to be enough information to build a key, which is why this solution includes an integer identifier — name alone is probably not enough. There is no need to add a person’s university as an attribute, because that follows from their team membership.

For each Glider, Fig. 1 gives attributes for `callSign` (string), `type` (string), and `numSeats` (integer).

The crew capacity in which someone is flying is an attribute of a particular flight: in this diagram, that is an instance of the “Flies” relationship between a Person, the Glider, and the Task being flown. Capacity is not an attribute of the Person because someone may fly as P1 on one day and P2 on another, possibly even in the same Glider.

### Solution 3

The relationship between Team and Person is one-to-many: each team can have any number of pilots, but no pilot can be in more than one team. In fact, the constraints on the relationship are slightly stronger, with a key constraint on Person as well as total participation: every Person must be in exactly one team. This should be shown on the diagram as a bold line (total participation) with an arrow-head (key constraint) between Person and the MemberOf relationship.

The other relationships are many-to-many, and there are no weak entities in the model shown here.

### Solution 4

In the solution of Figure 1, `{university}` is the primary key for Team. Alternatively, one could choose to generate a unique `teamId` and use that instead. This is what has been done here for Person, with the `{personId}` key. For Glider and TurningPoint, the scenario makes it clear that `{callsign}` and `{trigraph}` are appropriate choices. For Task the model uses `{compDayNo}`, the day number in the competition, as the scenario describes setting one task for each day of the competition. Alternatives might be to use the calendar date of the task, or give each task a unique name.

### Solutions 5 and 6

See the ER diagram in Figure 1.

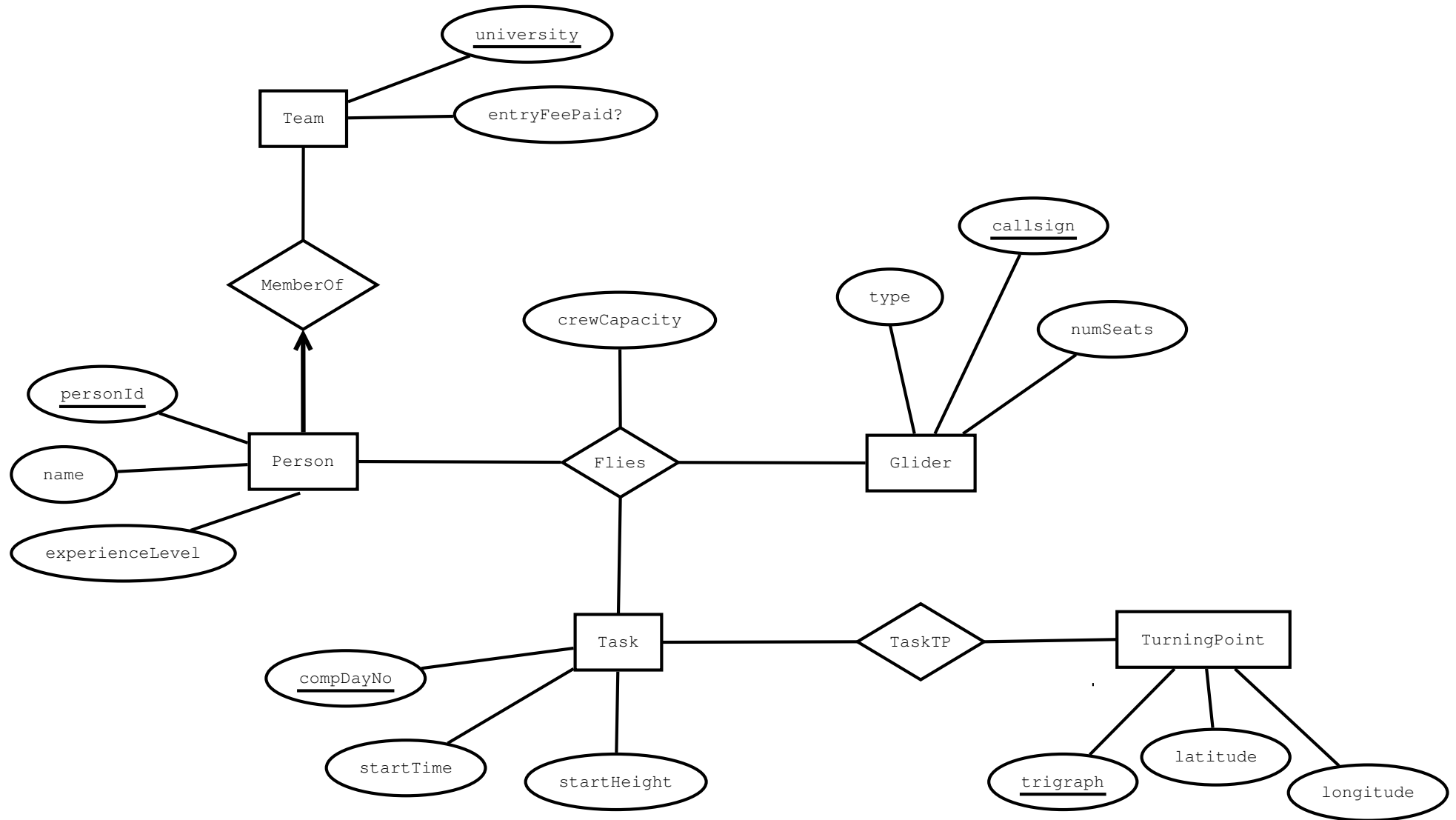


Figure 1: A proposed Entity-Relationship model for the gliding competition database.

## Discussion

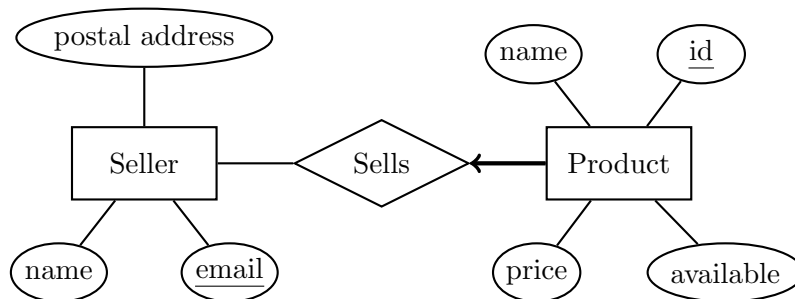
One rather serious omission in the model proposed is that there is no provision whatever for keeping score in the competition and working out who won. This was not, however, in the scenario; so in this situation it might be appropriate to go back to the author of that and ask what is required there.

Adding a score to each flight is not, however, completely straightforward: the obvious approach of adding a `score` attribute to the `Flies` relationship doesn't work satisfactorily, as there will be two entries in the corresponding relation for each two-seater aircraft. One possibility would be to introduce a new, separate relationship between `Glider` and `Task`, and add the `score` attribute to that. This would give one score per glider on the task, which is the desired result.

You may be able to suggest other solutions; or identify other data which could be useful to add. These should still be guided by an appropriate scenario: simply adding data without relevant motivation can cause new problems without solving any existing ones.

## Solution 7

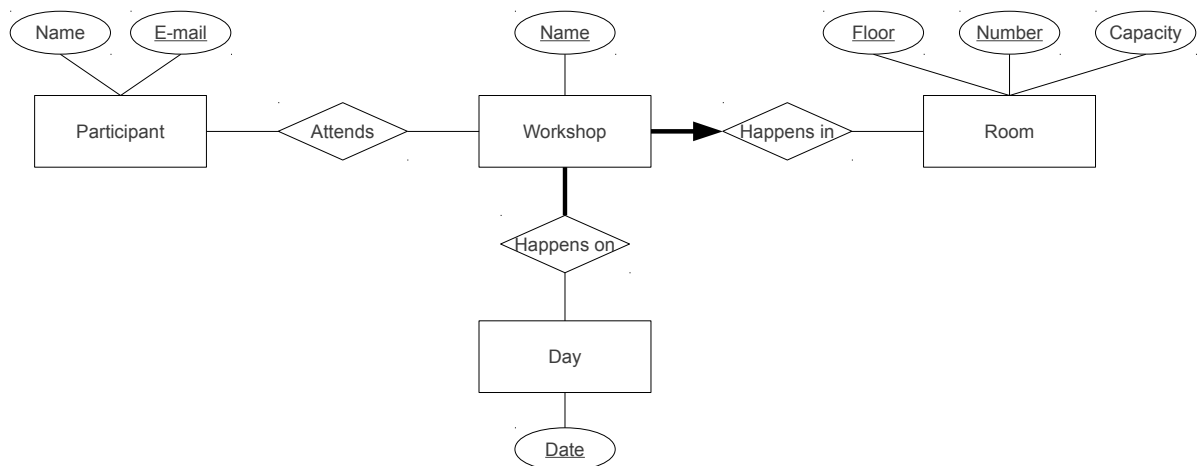
This ER diagram captures the information listed.



The thick arrow from *Product* to *Sells* indicates the *key constraint* and *total participation constraint* that every product must have exactly one seller.

## Solution 8

(a) Here is a suitable entity-relationship diagram.



One incorrect alternative is to assign a single “date” attribute to workshops; notice the explicit mention of multiple-day workshops in the scenario description.

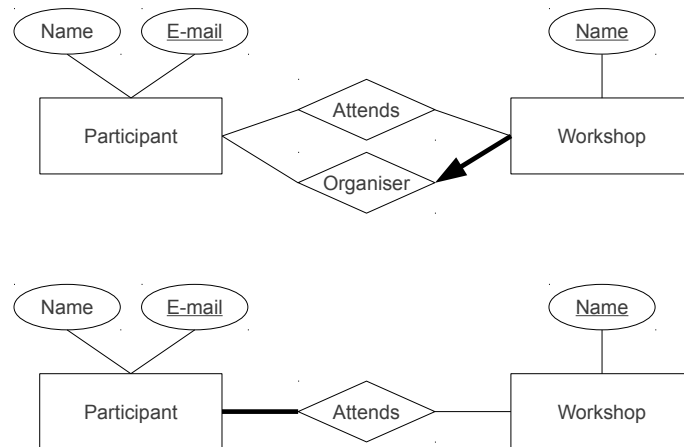
As well as the entities and relationships, it's important to include the primary keys, the participation constraints, and the key constraint.

- (b) (i) A *key* is a minimal set of attributes whose values uniquely identify an item in an entity set. For example, the e-mail address of a participant is a key for the corresponding entity set.

- (ii) A *composite key* is a key that includes more than one attribute. For example, the floor and room number of a meeting room.
- (iii) *Total participation* is a requirement that every element of an entity set must appear at least once in a particular relationship. For example, the requirements that every workshop must happen on at least one date, and in some meeting room.
- (iv) A *key constraint* is a requirement that each element of an entity set may appear at most once in a particular relationship. For example, the requirement that every workshop must be allocated no more than one room, even if it runs for more than one day.

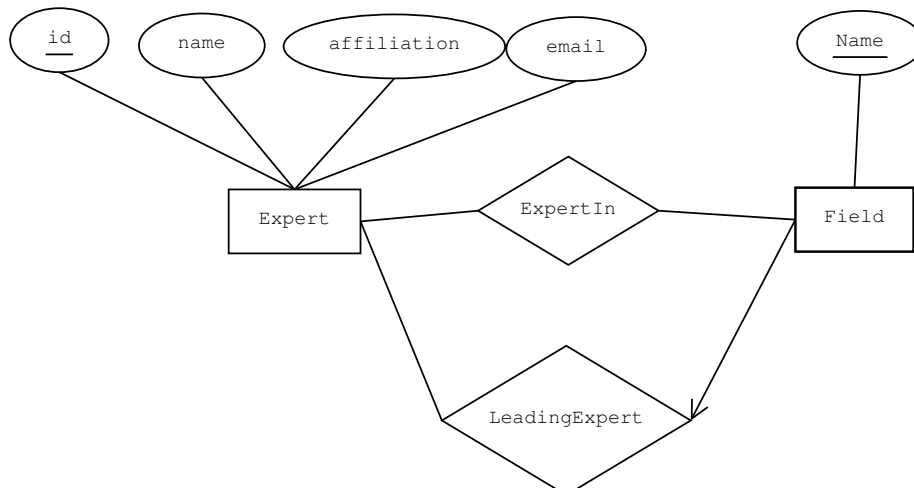
A total participation constraint is shown by a thick line joining the participating entity to the relationship. A key constraint is shown by an arrowhead on the line joining the entity to the relationship.

- (c) (i) The requirement for workshop organisers, and for every participant to register for at least one workshop, are both readily presented in an ER diagram.
- (ii) Recording organisers requires an additional relationship, with both a participation and a key constraint, as shown in the first diagram below. Participant registration for at least one workshop is represented as a participation constraint on the existing *Attends* relationship, as shown in the second diagram below.



### Solution 9

- (a) Here is a suitable ER diagram:



Notice that the line from entity *Field* to relationship *LeadingExpert* has an arrow indicating a key constraint. This captures the requirement that each field has at most one leading expert recorded.

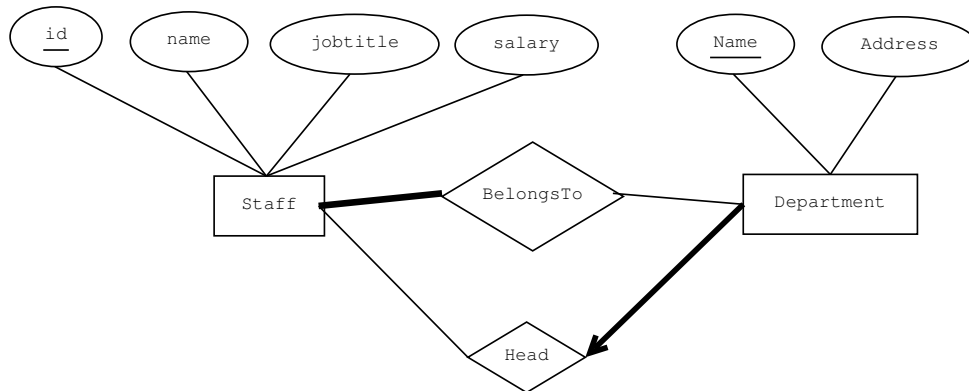
(b) Here are three possible additional constraints:

- Every field should have at least one expert. This could be incorporated by a participation constraint on *Field* in the *ExpertIn* relationship, denoted by a thick or double line for the existing arrow.
- Leading experts should always be listed as being experts in the appropriate field. This cannot be incorporated in the ER diagram.
- Every member of the *Expert* entity set should have at least one field of expertise. This can be incorporated by a participation constraint for *Expert* in the *ExpertIn* relationship, shown by a thick or double line.

You may be able to identify others.

### Solution 10

(a) Here is one possible ER diagram:



This includes these constraints, as required by the specification:

- A thick line of total participation from entity **Staff** to relationship **BelongsTo**, recording the requirement that every member of staff must belong to some department (although still possibly more than one).
- A thick line with an arrowhead from **Department** to **Head**, capturing the constraint that every department has exactly one head of department.

(b) These are some possible further constraints:

- Every Department should have at least one member of staff. This would be incorporated by a participation constraint on **Departments** in the **BelongsTo** relationship.
- Every head of department should belong to the department of which they are head. This cannot be readily incorporated in this ER diagram.
- No member of staff should be head of more than one department. This can be incorporated using a key constraint on **Staff** in the **Head** relationship.

Again, you may be able to identify other reasonable constraints: some of which are represented easily in an ER diagram, some that are not.