

1. Example of developing an ERD

The document provides an example of developing an ERD from a scenario using the process described in the ERD modelling section.

The following scenario was given:

2. Local community radio station - ERD exercise

A local community college has secured funds to run a local radio station called WEB (West End Broadcasting) for approximately one month a year. The administrator in charge of the venture wishes to collect information for a variety of reasons:

- Contact list of various types of people for the next season
- Billing and expenditure
- Activities including listings for each program
- Studio booking
- Equipment booking

The radio station has a core staff of four: a producer /director, manager, recording engineer (part time) and a secretary. Each of these people can play a number of different roles. For example the secretary may be a presenter for a program. Besides these staff a large number of volunteers take part in the day to day running of the station. Some are purely voluntary while others are paid an 'honorarium'. Examples of such staff are sound recording technicians / editors (mostly students from HND courses).

The actual programs themselves are focused within the community, being presented by local people themselves. For example a person with a particular interest in Victorian music might present a program about it.

The local schools and hospitals are also closely involved in the project with slots for various features.

A presenter (many of these are voluntary) can present several programs each of which will consist of several items. The items may be recorded in any sequence, for example a story about a local business success story may be recorded on location before other items in the program. Each program will be of a particular type (music, local news, lifestyle etc.). Similarly a sound technician may record the initial item and then may, or may not, pass it on to an editor. Editors always work at the program level, however if the program is going to be 'live', that is one or more items in it are live, the editor will produce the 'pre-recorded' component to the program and may hand that over to another editor for integration into the live component. A single item may be used by several programs. The final list of items that are broadcast for a program become the playlist for the program.

There is also an information feedback mechanism. This consists of an e-mail and answer phone facility. Both forms of communication are logged keeping details such as (program name, comment code (good, bad, other etc). and a summary of the response send back. Again volunteers staff this.

There are six recording studios - all identical, except that studio 1 is the one that is used for live broadcasting. Any equipment can be shared between the various studios. Equipment movement is very common as additional microphones etc. might be needed at any time in a particular studio. All equipment has a unique code.

3. Developing an initial list of entity types

After reading through the above scenario the following list of entity types suggested themselves (not in any particular order):

Contact list	Program
Billings and Expenditures	Play list
Activities	Item
Listings	Live session
Program types (Hospital, local news, lifestyle)	Pre-recorded session
Studio (no. 1 used for live recording)	Sound technician
Equipment (can be moved around each item has a unique code)	Program Editor
Staff (full time, part-time)	Pre-recorded Item Editor
Volunteers (some with honorarium)	E-mail Feedback
Roles	Answerphone Feedback

4. Consider Rumbaugh's criteria

Redundant entity types In the above list I have "Program Editor" and "Pre-recorded Item Editor" both these could be considered to be synonymous to "Editor". Similarly the "Live Session" or a "Pre-recorded Session" entity types, on first inspection, might be seen to be identical, however reading through the scenario again it mentions that one requires a Sound Technician while the other requires the additional help of an Editor. In contrast both E-mail Feedback and Answerphone Feedback are associated with the same information I therefore decide to replace them with the generic entity Feedback. Listings and Play list are also probably synonymous.

Irrelevant entity types The scenario provides a list of five reasons why the administrator wishes to collect information. Looking at the entities above and those suggested in the other subheadings of this section, all of them seem to be relevant.

Vague entity types In a narrative description often words are used indiscriminately. The scenario uses the word "activities" but does not expand or describe it, anywhere other than describing the production of a Program. I will therefore remove it for the time being. The description describes how Editors are used at the Program level and at the Pre-recorded Item level, it does not state if these can be the same person i.e the same Role. The difference between Staff, Volunteers and Roles is not clear. Obviously in the real world we would investigate this further to obtain further clarification. Unfortunately because this is not possible we will instead state a number of assumptions. I think it would be a good idea to combine the Staff and Volunteers entity types into a new one called Person (which would contain an attribute indicating which type of person they were). The Activities entity may be something that is related to Person but if we have the Role entity how does this differ from it to me at the present they both seem to be the same – I therefore think it is still sensible to ignore the Activities entity.

The Contact list entity type is unclear it could mean a list of people who wish to be involved in helping run the station are a list of potential listeners etc. again clarification is needed.

Entity types that are really attributes Billings and Expenditures could possibly be only attributes related to each piece of Equipment, Staff and some other Entities not mentioned in the scenario. As the Radio Station is a College venture in all probability they may obtain grants and / or donations and charge for some of their services / facilities. To allow such financial input / output to be recorded I will create a new entity type called Account to replace Billings and Expenditure. Incidentally we may have put the various Program types (e.g. Hospital etc) each as a separate entity type at the start and then only latter realised that the information collected about each one was identical, the only difference being an attribute indicating which type of program each was. In fact that is what the proposed entity Program Type is so it can be removed.

Implementation information The ERD is concerned with defining the data that needs to be stored. It is not concerned with the hardware or queries that can be generated from the data, such as listings of Expenditure for the last year etc.

Form the above discussion we now have the following list of entity types:

Contact list	Program
Account	Play list
	Item
	Live session
	Pre-recorded session
Studio (no. 1 used for live recording)	Sound technician
Equipment (can be moved around as each item has a unique code)	Editor
Person	
	Feedback
Roles	

5. Consider entity naming conventions

Once you have a possible list of entity types the module material suggests that you consider various naming conventions

Entity type name should:

1. **be unique for the particular model (This does not apply to attribute names which only need to be unique to a particular entity type)**

Looking at all the above-proposed entity types there do not appear to be any duplicate names.

2. **be a singular Noun e.g. PATIENT not PATIENTS**

Roles should be changed to Role

3. **be self-explanatory to those reading the ERD**

I think they are! However the term item is used in the above to describe both a single piece of Equipment and a part of a program, luckily we understand the difference?

4. **follow any naming conventions locally defined by the modellers. For example two common constraints are that:**
 1. **They should be written in UPPER CASE**
 2. **Possible spaces should be represented by '_'s for example DRUG_TREATMENT not 'DRUG TREATMENT'**

See below

5. **not be a name of an individual object (e.g. Freeman Hospital Newcastle upon tyne)**

No problems

6. **not express more than one concept (e.g EQUIPMENT/BED)**

No problems that I'm aware of

Revised list of entity types:

CONTACT_LIST	PROGRAM
ACCOUNT	PLAY_LIST
	ITEM
	LIVE_SESSION
	PRE_RECORDED_SESSION
STUDIO	SOUND_TECHNICIAN
EQUIPMENT	EDITOR
PERSON	
	FEEDBACK
ROLE	

We are now ready to develop a first draft ERD. It often helps to just have an ERD without any relationship lines to start with.

CONTACT_LIST

ACCOUNT

PERSON

ROLE

FEEDBACK

PRE_RECORDED_SESSION

PLAY_LIST

ITEM

LIVE_SESSION

PROGRAM

SOUND_TECHNICIAN

STUDIO

EDITOR

EQUIPMENT

6. Consider the relationships

This would normally be done in parallel with the above stages, however to make things clearer I have considered it separately.

Here is a list of possible relationships I have drawn up; some are explicitly defined in the scenario while others are assumptions that I have introduced. As a general guide remember that you need to consider each relationship in both directions:

A PERSON (voluntary or paid) has one or more ROLES

A PERSON buys zero or more pieces of EQUIPMENT

A piece of EQUIPMENT can only be brought by one PERSON

A ROLE is related to one or more PERSONS

A ROLE is related to zero or more PROGRAMS

A PROGRAM is related to one or more ROLES

A PROGRAM has one PLAY_LIST

A PLAY_LIST is for one PROGRAM

A PROGRAM consists of one or more ITEMS

A ITEM is related to zero or more PROGRAMS

A ITEM is related to zero or one LIVE_SESSION

A ITEM is related to zero or one PRE_RECORDED_SESSION

A ITEM is recorded by a SOUND_TECHNICIAN

A SOUND_TECHNICIAN records zero or more ITEMS

A PROGRAM has one EDITOR

A PROGRAM takes place in one STUDIO

A STUDIO can record zero or more PROGRAMS

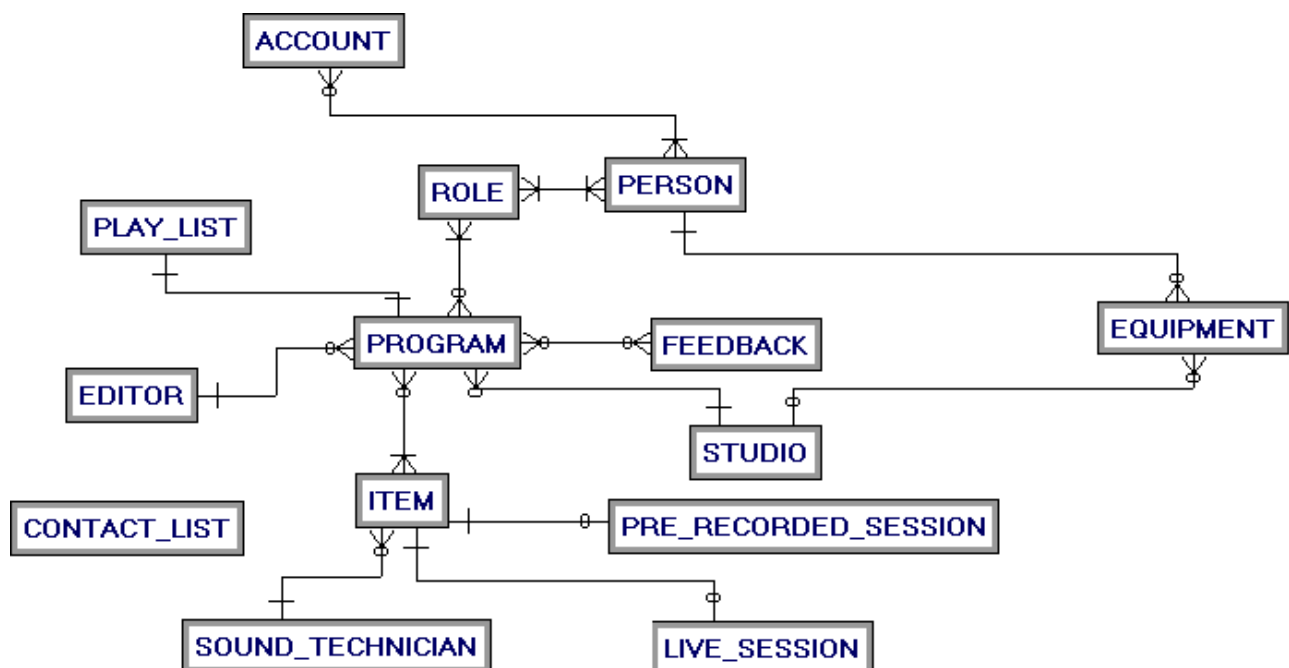
A STUDIO possesses zero or more pieces of EQUIPMENT

A piece of EQUIPMENT is in zero or one STUDIO

A item of FEEDBACK is related to zero or more PROGRAMS

A PROGRAM is related to zero or more items of FEEDBACK

PERSON receives zero or more payments from the ACCOUNT



6.1 Things to note about the above ERD diagram

1. Branches - Besides considering a relationship between just two entities it is important to realise that you can consider whole 'branch'. For example PROGRAM -> ITEM -> SOUND_TECHNICIAN. Remember that relationships are basically dependencies. There in the above branch we must have a instance of a SOUND_TECHNICIAN before we can create a ITEM instance. Similarly we must have a ITEM instance before creating a PROGRAM instance (this may not be necessarily a reflection of reality – but it is what the ERD specifies).

2. Number of relationships between entities. Notice that the majority of entities have only one relationship between them. This is because you can find out information by moving along branches, for example you could find out easily which particular SOUND_TECNICIAN(s) worked on a particular PLAY_LIST by carrying out a thing called a database Query.

7. Assumptions of the ERD

It is assumed that a program is recorded in a single studio, this may not be the case, the question is does the manager want to know if programs are recorded in more than one studio?

In the model each piece of equipment is related to a single buyer is this adequate for large purchases, the answer is probably yes for such a small organisation.

The CONTACT_LIST is not related to any other entity - if it were to be a list of people who were interested in helping with producing programs it would be related to the PERSON entity in fact it may end up being equivalent to it.

A ITEM in the current model may belong to zero or more PROGRAMs. It might be more sensible to always associate a ITEM with a PROGRAM and have the PROGRAM considered to be either one in preparation or having been broadcast. The manager may be interested in finding out how many ITEMS were not broadcast for particular programs or not be bothered to have these identified?

In addition to the relationships defined in the above ERD some others could have been included:

"An ACCOUNT is related to zero or more PROGRAMs" however this can be derived from the relationships between PERSON -> ROLE -> PROGRAM.

EDITOR is related to PERSON and / or ROLE but may also be redundant as an entity as in the current situation a EDITOR may well exist and produce a program but not exist in the PERSON entity. This would need further investigation in the real world.

The relationship between PLAY_LIST, ITEM and PROGRAM may be alternatively defined by:



The problem with both solutions is that the exact relationship (in terms of dependency) between PROGRAM and PLAY_LIST is unclear; can a PLAY_LIST really not exist without a PROGRAM or visa versa. Probably the best solution is that given in the above ERD but with an additional description suggesting that PLAY_LISTs can exist independently of PROGRAMs in contrast to what the scenario suggests.