Missed the lecture on Wednesday, need to go back over that.

ER Diagram

Framework for sketching design ideas before actually creating any tables.

ER = entity relationship.

- Rectangles for entity sets
- Ovals for attached info (which will be columns)
    - Keys are underlined
- Diamond shapes for relationships between entity sets

More on ER Modelling

For our movies DB, castings was a relationship and movies and actors were entity sets.

DB1106 Consultants (our example)

You need to establish first what you want your database to do and keep track of.

Entity set or relationship?

Roughly:

- Entity sets often conveniently describable using nouns, e.g. employees, departments, etc.
- Relationships often conveniently describable using verbs, e.g. supervises, works on, etc.

Entities

- Employees
- Departments
- Projects
- Locations

- Dependents

Relationships

- Manages: employees -> departments
- WorksIn: employees -> departments
- WorksOn: employees -> projects
- Supervises: employees -> employees
- HasLocation: departments -> locations
- IsDependentOf: dependents -> employees

Relationships with attributes

Often nice to associate attributes with relationships rather than entity sets, e.g. WorksOn can have the attribute "hours" which tracks how many hours an employee works on each project they work on.

This needs to be associated with employees, but each employee may work on multiple projects, so it makes sense to have it in WorksOn.

Implicit vs explicit relationships

Could use just attributes to convey them, e.g. employees has an attribute "manages" and departments has an attribute "manager". This is an implicit relationship.

Implicit relationships are redundant and error-prone.

In this case, we would have no way of representing that an employee is managing two departments. Also, if we want to make a change to who's managing a department, we have to make two changes rather than just one.

Reflexive relationships

Supervises is a relationship connected twice to employees, since it's an employee supervising another.

General design principles

- Faithfulness – the DB should reflect the real-world situation as well as possible

- Simplicity – the DB should be as simple as possible
- Avoiding Redundancy – each piece of info should only be represented once. Duplicate info is wasteful of space and makes errors when updating info more likely.

Exams

Since ER diagrams take a lot of time to develop, only quite simple ones could be required in an exam.

The forthcoming exam, on Friday 13th November, will be mostly SQL-based.