Spatial Data Management

- Dr Claire Ellul
- c.ellul@ucl.ac.uk

Overview

- **Database Design**
- Conceptual ER Diagrams
  - Entities, Relationships, Attributes
  - Cardinality of Relationships, Cardinality of Attributes, Identifiers
- Creating an ER Diagram
- Logical ER Diagrams
- Introducing the assignment

Database Design

- What is Database Design?
  - This is the process of defining the structure, characteristics, rules-base and contents of a database
  - Important to document this process so that any decisions made as part of the design process can be traced at a later date

Database Design

- Database Design Tasks
  - Conceptual Design – a diagrammatic and text description of user requirements, documented as an Entity-Relationship diagram
  - Logical Design – takes the information gathered at Conceptual Design stage, and transforms it to take into account system performance and expected operational conditions.
  - Normalisation part of Logical Design, used to remove redundancies (duplicate data) from the model
  - Physical Design – takes the normalised logical design and converts it into actual build scripts for the database.

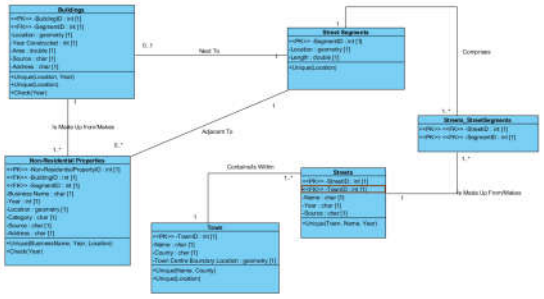
Database Design

- Database not designed in isolation from other system software
- Each step results in the production of a standardised design document
  - This includes 2 * Entity Relationship Diagrams (one conceptual, one logical) and other information

E-R Diagrams

- The Entity Relationship Diagram
  - Presents data requirements of a system in a manner that is easily understood by management
  - Does not take into account expected usage or other operational requirements of the system
  - Used to define what data should be held in the system, and what values typically represent this data
  - Includes actual diagram and SUPPORTING DOCUMENTATION

ERD – An Example



E-R Diagrams - Notation

- UML
  - Was developed for programming, to model classes, properties and methods
  - Diagram Elements include:
    - Class diagrams – attributes, methods and relationships – WE WILL USE THIS
    - Use Case diagram – actors in a system and their goals – not used in this module

E-R Diagrams - Notation

- Learning UML notation is also useful because:
  - For pre-existing databases, you can also reverse engineer the E-R Diagram using tools provided by the DBMS – usually the result is presented as a logical UML diagram
    - This is useful when you are given a database and are not familiar with the tables and data it contains

Overview

- Database Design
- Conceptual ER Diagrams
  - **Entities, Relationships, Attributes**
  - Cardinality of Relationships, Cardinality of Attributes, Identifiers
- Creating an ER Diagram
- Logical ER Diagrams
- Introducing the assignment

E-R Diagrams

- Key Components (Constructs) of an E-R Diagram
  - Entities
  - Relationships
  - Attributes

Entities

- Entity
  - Represents classes of objects that have properties in common and an autonomous existence
  - Each entity must have a unique name
  - Graphically represented in the diagram by means of a UML CLASS

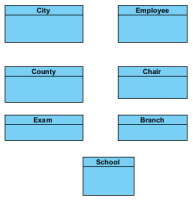
Entities

- An Entity:
  - Examples include: person, product, project assignment
  - Not: technological objects – files, PCs, screens, windows, "Project history"

3. Barker, Richard. 1990. *CASE*Method: Entity Relationship Modeling*. (Wokingham, England: Addison-Wesley). (with thanks to David Hay)

E-R Diagrams

- Some Examples of Entities



E-R Diagrams

- An Entity:
  - … is *limited* to what Richard Barker calls things or objects "*of significance*, whether real or imagined, *about which an organization* needs information."[3]

3. Barker, Richard. 1990. *CASE*Method: Entity Relationship Modeling*. (Wokingham, England: Addison-Wesley). (with thanks to David Hay)

Entities

- An entity is a database object that represents a thing in the real world. Entities are expressed as nouns.
- Entities can be concrete, like buildings and employees or they can be more abstract things like departments and accounts.

https://info.teradata.com/htmlpubs/DB_TTU_16_00/index.html#page/Database_Management/B035-1094-160K/xuq1472240596754.html

Entities

- An entity is any singular, identifiable and separate object.
  - It refers to individuals, organizations, systems or even distinct system components that are considered significant in and of themselves.
- An entity's common denominator is that it can be considered a separate whole and possesses a unique set of characteristics.

Adapted from https://www.techopedia.com/definition/14360/entity-computing

Entities

- Perhaps one 'test' for whether something is an entity is whether you can pick it up and put it somewhere else ?
  - Maybe not valid 100% of the time, but could be a good starting point ?

E-R Diagrams

- Entity Names
  - The name of an entity is in the singular, and refers to an **instance of that class.**
  - Needs to be in natural English
    - Hence, Order and Line Item are acceptable.
    - The name "Project history" is not (as this is composed of many details).
  - Database table names are not allowed, nor are abbreviations or acronyms.

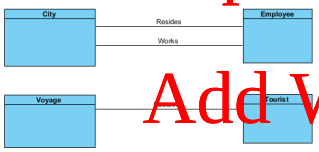
Source: https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0CGcQFjAG&url=http%3A%2F%2Fwww.ir.mac.ca%2F121%2F1.%2520UML%2520and%2520Data%2520Modeling%2520-%2520A%2520Reconciliation.ppt&ei=yhMBUvKfEuOy0QWU84GoAQ&usg=AFQjCNGhvkmoKeIhJC2JSJBrDMgZY_L6Xw&sig2=lhXouLcPSrRDdch TOtpLg&bvm=bv.52434380,d.d2k

E-R Diagrams

- Relationship
  - These represent the logical links and natural associations between two or more entities
  - Each relationship has a unique name
  - There can be more than one relationship between the same two entities
  - Graphically represented by means of an 'association'

E-R Diagrams

- Some Examples of Relationships



E-R Diagrams

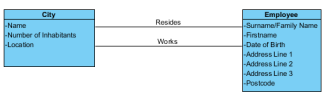
- Attributes
  - Describe the elementary properties of entities or relationships
  - Can be grouped for simplicity into composites
  - Graphically represented by means of a list within the CLASS

E-R Diagrams

- Attributes
  - *An Attribute is a characteristic of an entity type.*
  - *It "serves to qualify, identify, classify, quantify, or express the state of an entity"*

E-R Diagrams

- Some Examples of Attributes



E-R Diagrams

- Some Rules
  - Entities and Relationships should have different names - no duplicate names
  - Make sure each entity appears only once
  - Each entity, relationship and attribute should have a name
  - Don't connect relationships to each other
  - Only connect entities where the connection makes sense - it is not necessary to connect every entity to every other entity

E-R Diagrams – Some Practice

- Read the *UCL Facilities Management* project information sheet
  - Identify and diagram the entities
  - Identify and diagram the relationships between the entities
  - Identify and diagram the attributes
  - Do you need any additional information? Are you making any assumptions?

Overview

- Database Design
- Conceptual ER Diagrams
  - Entities, Relationships, Attributes
  - **Cardinality of Relationships, Cardinality of Attributes, Identifiers**
- Creating an ER Diagram
- Logical ER Diagrams
- Introducing the assignment

E-R Diagram Constructs

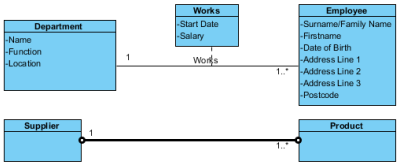
- Cardinality of Relationships
  - These describe the minimum and maximum number of relationship occurrences in which an entity can participate
  - Can be
    - One-one
    - One-many
    - Many-many (although these should be resolved for the logical ER diagram)
  - Also
    - Mandatory (minimum value of 1)
    - Optional (minimum value of 0)

E-R Diagram Constructs

- Cardinality of Relationships
  - Mandatory
    - 1:1
    - 1:N
    - M:N (OK in conceptual, should be eliminated in Logical)
  - Optional
    - 0:1
    - 0:N
    - M:N (OK in conceptual, should be eliminated in Logical)

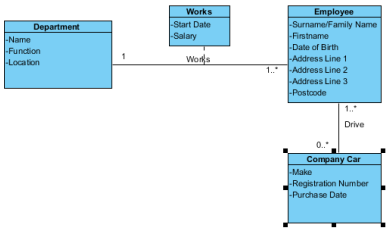
E-R Diagram Constructs

- Examples of Cardinality of Relationships



E-R Diagram Constructs

- Examples of Cardinality of Relationships

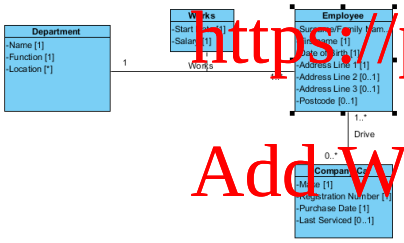


E-R Diagram Constructs

- Cardinality of Attributes
  - Specify the minimum and maximum number of values of the attribute associated with the entity or relationship
  - In most cases, this is (1,1), which is not shown on the diagram
  - However, (0,1) used when the attribute can be null
  - Attribute mandatory when the minimum cardinality is equal to one
  - Attribute optional when the minimum cardinality is equal to zero

E-R Diagram Constructs

- Cardinality of Attributes



E-R Diagram Constructs

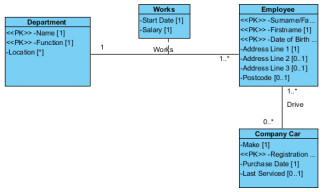
- Identifiers
  - Allow unambiguous identification of entity occurrences
  - Formed from one or more attributes of the entity itself
  - Eventually will form the primary key or for the entity

E-R Diagram Constructs

- Identifiers
  - Those of you who have worked with databases before may be familiar with the concept of a number column as a primary key in a table
  - Be careful – as the E-R diagram is at conceptual level there is no concept of substituting a numerical ID value – your identifiers should be the REAL identifiers for the entities.
    - More about ID values in later weeks

E-R Diagram Constructs

- Examples of Identifiers

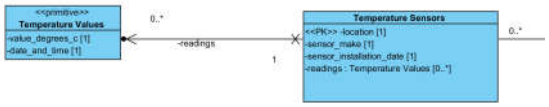


Primitive Types

- These are a special type of entity that is used in a conceptual diagram to represent information that doesn't stand on its own but needs to be modelled separately
  - e.g. the readings from temperature sensors where you have many readings for one sensor
  - The readings don't exist on their own – you can't pick them up – but you have a 1:sensor to many:readings relationship

Primitives

- These are represented in the same way as an entity, but don't have any identifier
  - The relationship is also shown slightly differently (i.e. one way)



E-R Diagrams : Some Practice

- Complete the Conceptual ERD for the UCL Facilities Management System
  - Identify and diagram the cardinality of relationships
  - Identify and diagram the cardinality of attributes
  - Identify and diagram the identifiers for each entity

Overview

- Database Design
- Conceptual ER Diagrams
  - Entities, Relationships, Attributes
  - Cardinality of Relationships, Cardinality of Attributes, Identifiers
- Creating an ER Diagram
- Logical ER Diagrams
- Introducing the assignment

E-R Diagrams

- Creating an E-R Diagram
  - Identify the entities – roles, events, locations, tangible things or concepts about which the end-user wishes to store data
  - Identify the relationships, by finding the natural associations between these entities
  - Identify the attributes for each entity
  - Draw a draft E-R Diagram
  - Identify the cardinality of the relationships (one-one, one-many, many-many)
  - Remove many-many relationships by adding additional entities
  - Define the primary keys for each entity
  - Draw the finalised diagram
  - Write associated documentation!!

E-R Diagram Documentation

- E-R Diagram Documentation
  - Diagram must be accompanied by associated text-based documentation. It is not sufficient to have the diagram without this
- Documentation should detail the business rules that form the basis of the E-R diagram
- This is usually done using text and mathematical formulae

E-R Diagram Documentation

- Business Rules
  - The precise definition of an entity, attribute or relationship
  - An integrity constraint on the data of the application
  - A derivation detailing an arithmetic calculation that can be performed on the data

E-R Diagram Documentation

- Business Rules – Entity Definition
  - An employee is defined as someone who is permanently employed with the company. This does not include temporary employees. Employees are deemed to have signed an employment contract for a minimum of six months.
  - A sale is defined as removing an item from stock and delivering it to the purchaser. A sale is not complete until an invoice has been issued.

E-R Diagram Documentation

- Business Rules – Integrity Constraints
  - Student Name, Surname, Date of Birth and Address provide the unique identifier for the student entity
  - Students must be over 18 years old to register with this university.

E-R Diagram Documentation

- Business Rules – Derivations
  - The cost of the sale can be calculated by adding the direct and indirect costs together
  - The age of the student can be derived from their date of birth

Overview

- Database Design
- Conceptual ER Diagrams
  - Entities, Relationships, Attributes
  - Cardinality of Relationships, Cardinality of Attributes, Identifiers
- Creating an ER Diagram
- **Logical ER Diagrams**
- Introducing the assignment

Database Design

- Database Design Tasks

Conceptual Design – a diagrammatic and text description of user requirements, documented as an Entity-Relationship diagram

Logical Design – takes the information gathered at Conceptual Design stage, and transforms it to take into account system performance and expected operational conditions.

Normalisation part of Logical Design, used to remove redundancies (duplicate data) from the model

Physical Design – takes the normalised logical design and converts it into actual build scripts for the database.

Conceptual versus Logical

- Conceptual – decide WHAT to represent in your system
  - What items and information in the real world are important / needed so that your system can answer the required questions
- Logical decide HOW
  - Which specific DBMS software to use
  - Exactly how the items you identified above should be created

Conceptual to Logical

- Translation into the Logical Model
  - NB: As we are using UML notation there is not much difference in notation – we still use classes, attributes, associations and so forth!

Conceptual to Logical

- Translation into the Logical Model
  - Translate
    - Entities
    - Identifiers
    - Many:many relationships
  - Into
    - Tables
    - 1:many relationships
    - Primary and Foreign Keys
    - IDs

Database Vocabulary

- Some General Terminology
  - Domain/Data Type
    - Strings
    - Dates
    - Numbers
    - Spatial Data
  - Each column stores information using one DataType so you can't put strings (free text) into a date or number column or spatial data into a string column

Database Vocabulary

- Some PostGIS Specific Terminology

General Domain Type	PostGIS Terminology
String	character varying (length)
Date	date
Number	integer numeric (precision, scale)
Spatial Data	geometry
(automatically increasing number used for ID values)	serial

Conceptual to Logical

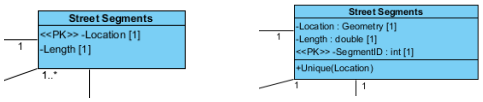
- Decide the data types for each field
- Replace identifiers by ID columns + unique constraints

Conceptual to Logical

- Entities
  - Become tables in the logical model
  - Each table has the same name as the entity
  - Each table has the same attributes but now we add constraints, data types and IDs as primary keys

Conceptual to Logical

- Entities



Conceptual to Logical

- One-to-Many Relationships
  - One table formed for each entity
  - Any attributes of the relationship assigned to the child entity

Create Relational Model

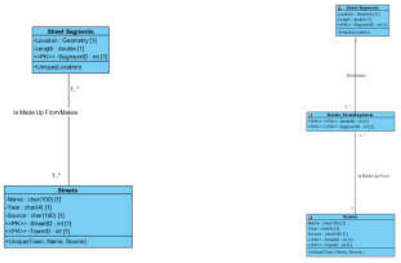
- One-to-One Relationships
  - Each entity becomes a relation (table)
  - The attributes from the relationship between the entities are assigned to one or the other of the new tables

E-R Diagrams

- Resolving Many:Many Relationships
  - As we will see in future weeks relational database cannot handle many:many relationships
  - However, at conceptual level you may find these in your E-R diagram
  - You should go over your diagram and add additional entities to resolve these.

E-R Diagrams

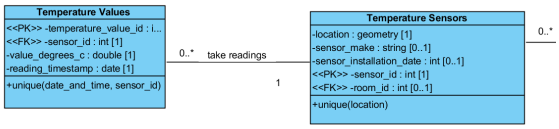
- Resolving Many:Many relationships



Handling << primitive>> types

- Entity type changed from << primitive>> to normal entity
  - 1:many relationship created between parent and primitive
  - Attribute linking to the primitive type removed from the parent

Handling << primitive>> types



Conceptual to Logical

- Key Differences
  - Full Primary keys (identifiers) versus numerical IDs and unique constraints
  - Many:many relationships versus 1:many relationships
  - Non-normalised versus normalised
  - Column names versus column names + data types

Overview

- Database Design
- Conceptual ER Diagrams
  - Entities, Relationships, Attributes
  - Cardinality of Relationships, Cardinality of Attributes, Identifiers
- Creating an ER Diagram
- Logical ER Diagrams
- **Introducing the assignment**

Introducing The Assignment

- See assignment handout in Moodle
- https://moodle-1819.ucl.ac.uk/course/view.php?id=1339&section=16

E-R Diagrams

- Software Used for ERD creation
  - Visual Paradigm community edition (free to download)
    - NB: Use the CLASS DIAGRAM not the Entity Relationship Diagram (which uses non-UML notation)
    - Useful instructions can be found here:
      - http://www.visual-paradigm.com/support/documents/vpuserguide/94/2576/7190_creatingclas.html
    - Also some hints in the document on the Moodle assignment page

UCL

Further Work

- Reading
  - Before next week, read the three worksheets about SQL – you can find these under the 'Week 3' tab in Moodle.
    - DDL
    - DML
    - The Select Statement

You should also start work on your system specification, conceptual and logical diagrams for your assignment!

UCL

E-R Diagrams

- Reference Paper E-R Diagrams
  - P. P.-S. **Chen**. *The **entity-relationship** model-toward a unified view of data*. ACM Transactions on Database Systems, 1(1):9-36, 1976.
- Useful book
  - UML and Data Modelling: A reconciliation, by David Hay, Published 1 July 2012,