COM201: System Analysis and Design

Lecture 7: Conceptual Database Design

Dr. Fayza A. Nada

1. Title
2. Overview
3. Databases

Purpose: Revise the introduction to databases from lecture 1

Notes:

- Organized collection of facts and information.
- A database is like an electronic filing cabinet. Databases store (known as persistent data) and organize data
- Need software: Database Management Systems (DBMS) to interact with database
    - Create, Maintain, and manipulate data – to provide useful outputs

What are they good for:

- Store large quantities of information
- Retrieve information quickly
- Organise and reorganise information
- Print and distribute information in a variety of ways

Key elements

- Tables
- Row (records)
- Columns (fields)
- Relationships

4. Levels of Information Design

Purpose: Revise the levels of abstraction in database design introduced in the previous lecture

Notes:

- **Levels of abstraction:**

- Taking essential characteristics of the real-world and incorporating them in a some kind of graphical representation
- High → closer to real world
- Low → closer to computer representation (binary)

- The design abstraction layers can become more intuitive when we replace the existing terms with business, system model, and technology models
- On right-hand side are design artefacts or representations for data (or database) related designs
- Conceptual design:
  - As close to the real world as possible, i.e., documenting the real world (ERDS, class diagrams etc.)
  - Attempts to abstract or represent real-world information and relationships from domain
  - Entirely independent of implementation concerns (i.e., No concern for the type of database being used to store resulting data)
- Logical design:
  - More closely associated with documenting the real world in terms of corresponding to database model, typically relational database.
  - Preparing database schema
  - Specification:
    - Applied to generic database solution
    - Data model but not technology dependent to a specific implementation of the relationship model
  - Implementation:
    - Technology chosen and database schema transformed to meet requirements of technology
- Physical design:
  - Physical storage within the database
  - Records, pointers, tracks, sectors, etc.
  - As we will see handled by the DBMS and invisible to users.
  - Performance, response time, etc.

5. Entity Relationship Diagrams

Purpose: Introduce Entity Relationship Diagrams (ERDs) and their key elements

Notes:

- **There are many different ERD notations**
- **Entities:**
  - **Usually nouns**
  - **Often described with the singular**
- **Relationships:**
  - **Links or associations**
- **Attributes:**
  - **Are atomic (represent one thing) – cannot be decomposed further**
  - **Must be relevant to the problem domain, appropriate for the context (e.g., when we describe a member of a video store, we don't require height, weight, eye colour etc. that might appropriate for a hospital patient system)**
  - **Often described with the singular**
  - **May or may be shown on a diagram but will be defined**

6. How is Data Modeling supported by CASE?

Purpose: Revise how data modeling (ERDs) are supported by CASE

Notes:

- **Has repository so designs are shared amongst developers**
- **Automatic transformation and generation of databases from models**
- **Can view High-level (conceptual, business) vs. lower-level (logical, system) representations**

7. Entities

Purpose: Describe ERD element of entities in more detail

Notes:

Entity representation

- **Shape – Diagrammatically: normally rectangle (i.e. most notations) → Oracle designer soft box (i.e. a rectangle with rounded corners). Any size**

Name:

- **Usually name as singular (i.e. no "s")**

- Should be unambiguous names (also good practice not to use abbreviations)
- Synonyms: often in a business context there is more than one name for something. In these cases one name should be used and the other record to create a clear definition of the entity.

Identifying entities

- Entities are things that people talk about, write about, record information about and do work on.
- Often the nouns in sentences
- Can often be actors (roles people play) – that you need to store information about
- Nouns – "not all nouns are entities" (on next slide)
- Once you have identified nouns then you need to desire what is an entity:
  - Is it a unique thing the system needs to know about?
  - Is it inside the scope of the system you are working on?
  - Does the system need to remember more than one of these items?
  - Is it really a synonym for some other thing you have identified? – it could be a different role of the same thing (i.e. still an entity)
  - Does it describe a specific piece of information (attribute) about some other thing you have identified?
  - Can be difficult to identify entities from people talking – because people often use examples (instances or occurrences) and synonyms to describe things.

Categories:

- Tangible things: e.g., airplane, books, vehicle, document, worksheet
  - Often the most obvious – easily identified (compared to intangible)
- Roles played: e.g., employee, customer, doctor, patient, end user
  - Other is the roles people play – you record different information based on the roles they play. Need to be important to the organisation.
- Organisational unit: e.g., division, department, task force
- Devices: e.g., sensor, printer, display, hard disk
- Incidents, events or interactions: e.g., flight, service call, purchases, order, payment
  - Something these are thought of as relationship between things – but if you are storing something about them – then they are entities.
- Sites or locations: e.g., warehouse, factory, retail store

8. Is 'it' really an Entity?

Purpose: Describe this list of questions to help check whether something is an entity or not.

9. Entity Type versus Entity Instance

Purpose: Describe the difference between entity types and instances

Notes:

Entity type

- **Definition of the thing or object.**
- **General representation of the object or thing**
- **Template**
    - **Include type of information (attributes) that describe the entity**

Entity instance

- **A particular example (occurrence) of an entity type.**
- **Values assigned to each attribute of the entity.**
- **Each instance (occurrence) of an entity must be separate and distinct.**

10. Attributes

Purpose: Describe ERD element of attributes in more detail

Notes:

Attributes:

- **"A piece of information that describes an entity" (Date 2004)**
- **One piece of specific information about a thing (atomic).**
- **Attributes could be text, numbers, a picture, sound, etc.**

Attribute representation:

- **Diagrammatically (optional): Not a requirement of most notation but most offer you a means to show some or all of the attributes on the diagram**

Attribute roles include:

- **Descriptive: individual atomic properties that describes the entity of interest.**
- **Identifying: used to uniquely identify occurrences of entity types.**

Attributes assigned values from domains:

- Values (entity instance) from a pool of values known as a domain
- In practical terms data types are mainly used.
- e.g. attribute: colour → domain: {red, blue, white, yellow}

Attribute optionality:

Optional: e.g., alternate phone number, email, …

Logical represented using "null"

Mandatory: e.g., customer name, postal address, …

A value is always required when instance is created

Different ways to represent identifiers in different notations:

- Usually underline
- In the case of the CASE tool you will use (PowerDesigner) it also has a PI.

- Why might birthdate or name be a bad idea for identification?
- Using what you know of databases from the MS Access experience – what will this unique identifier eventually become in the actual database?
  - Student ID, VIN, Account number, Video code, Serial number

11.Relationships

Purpose: Describe ERD element of relationships in more detail

Notes:

- An association (relationship) that exists between entities reflects something from the real world.
- A naturally occurring association among specific things, e.g. an order is placed by a customer and an employee works in a department

Cardinality:

- Represents how many instances of one entity type can be associated with an instance of the other entity type.
- Also known as relationship degree.
- 1:1 - not that common (could be the same entity) (e.g. driver 1:1 licence)
- 1:M - most common (1 to many)

- **M:M - are common (conceptual – initial design)**
- **Optionality or participation:**
- **Represents whether or not instances of one entity type must participate with instances of a related entity type.**

Labels:

- **Some ER notations use only one label on the relationship – but relationships can be read from both directions**
- **Reason for using two relationships (labels)**
    - **Forces you to think more closely about all aspects of the relationship (i.e. meaning in both directions)**
    - **Translates to business rules in conjunction with other elements (describes the business)**
- **Describes in sentence from the nature of the association.**
- **Adds meaning, purpose and clarity to relationships**
- **Without labels they can be meaningless, ambiguous, incomplete**
- **Also, potential exists for multiple relationships between entities therefore need labels to distract**

Degree:

- **The entity types are call participants**
- **The number of participants in a relationship is known as the degree.**
- **Usually binary (two participants) - what we will deal with in this course**
- **Can be unary (one participant between an entity and itself), ternary (three participant cannot be implement in most CASE tools including PowerDesigner), …**

12.Relationship Example I

Purpose: Illustrate relationships with Author-Book example

Notes:

- **Point out the cardinality, labels, direction, etc.**

13.Relationship Example II

Purpose: Continue to illustrate relationships with Author-Book example by introducing optionality

14.Video Store Database Example

Purpose: Describe the Video Store ERD example – covering all that the elements introduced in the previous slides

15.Categories of Entity

Purpose: Describe the different types of entities

Notes:

Strong

- **Exist independently of other entities**
- **Synonyms: kernel, major, regular, independent.**
- **e.g., EMPLOYEE**

Weak

- **Existence dependent on another entity (strong)**
- **Synonyms: dependent, minor.**
- **e.g., ORDER**

Associative

- **Describes a particular association**
- **Requires at least one attribute**

16.Associative Entities

Purpose: Introduce associative entities

Notes:

- **Use the Student/Course M:M relationship example to illustrate the need for associative entities**
  - **A student can be enrolled in one or many courses**
  - **A course can have one or many students enrolled**
- **When you want to store information about the relationship (e.g. the date started) need to create an associative entity**
- **This is done by breaking the associative entity into two 1:M relationships with an entity in-between**
- **Remember only done at the conceptual data modelling level if you required information to describe the relationship**
  - **Student: Student_number**

- Course: Course_ID
- Enrolment:
    - Combined Student_number and Course_ID (more in the next lecture)

- ***Laboratory 5 is a good example for building an associative entity.***

17.Associative Entities – PowerDesigner

Purpose: Describe how associative entities are represented in PowerDesigner

Powerdesigner is not freeware; you can download it free for 15 days only (www.sybase.com)

Notes:

18.Form Analysis

Purpose: Introduce form analysis, which can be used to determine the data requirement (modeled in an ERD) from source documents

Notes:

- **Use the order form to illustrate the kind of information we can obtain and how we can use this information:**
    - **Identify things of interest which eventually form entities.**
- **We can take this analysis and construct a model that represents this information and associated relationships**

19.Resulting Conceptual Design

Purpose: Describe the resulting ERD from the form analysis on the previous slide

Notes:

- **Could all this information be gained from source document?**

20.Suggested Steps for ERD creation

Purpose: Introduce a suggest approach for creating ERDs from whatever source (interview, source documents, etc.)

Notes:

Identify entities:

- **Does it have at least one descriptive attribute?**
- **Are there multiple occurrences (define identifier)?**

- Is it a strong, weak, associative, ...?
- Are there any synonyms (e.g. customer → client)?

Identify relationships between entities:

- What is the optionality, cardinality, degree (business rules)?
- Place the labels on the relationship.

Document the attributes for each entity:

- Attach attribute to each entity.
- Define suitable domain (data types).
- Investigate possibilities for further decomposition (e.g. are the atomic or more than one attribute?)

21.UML: Class Diagram

Purpose: Describe how UML: Class Diagrams can also be used to model data requirement.

Notes

- Discuss the differences in notation.
- This one shows entity classes (domain model) analogous with ERD
  - Classes are equivalent to entity types
  - Class diagrams will also exist for interface (boundary) and logic (control)
  - Models inheritance and aggregation
  - Multiplicity: both cardinality and optionality.
  - Behaviour is also included for modelling functionality