Databases – Aaron Connolly

A Database is a shared, integrated computer structure that houses a collection of:

• End users Data (that is raw facts of interest to the end user)

• Metadata (data about data, through which the data are integrated and Managed)

Definitions

Data: stored representations of meaningful objects and events

• Structured: numbers, text, dates

• Unstructured: images, video, documents

Metadata: data that describes the properties and context of user data – Data dictionary

A Database Management System (DBMS) is a collection of programs that manage the database structure and controls access to the data stored in the database.

Multi-file Database are Relational Databases

• An Order entry system for a business, for example, will consist of many tables which maybe related:

- – an orders table to track each order
- – an orders detail table for tracking each item in an order
- – a customer table so you can see who made the order and who to bill
- – an inventory table showing the goods you have on hand
- – a suppliers table, so you can see who you need to re-order your stock from
- – a payments table to track payments for orders
- Problems with the Activities table still are:
  1. Wasted space. Some students don't take a second activity
  2. Addition anomalies. (student wishes to do 3 activities)
  3. Redundant data entry. If the tennis fees go up to $25?
  4. Querying difficulties. It's difficult to find all people doing swimming: 5. Redundant information. If 50 students take swimming?
  6. Inconsistent data.

Disadvantages of File Processing

- Program-Data Dependence

• All programs maintain metadata for each file they use

- Duplication of Data

• Different systems/programs have separate copies of the same

data

- Limited Data Sharing

• No centralized control of data

- Lengthy Development Times

• Programmers must design their own file formats

- Excessive Program Maintenance • 80% of information systems budget

Advantages of the Database Approach

1. Program-data independence -

- Istheseparationofdatadescriptions(metadata)fromtheapplication programs that use the data – with the database approach, data descriptions are stored in a central repository. This property allows an organizations data to change and evolve (within limits) without changing the application programs that process the data

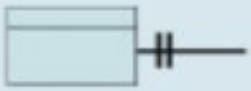
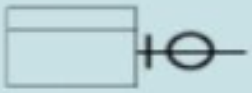
2. Minimal data redundancy

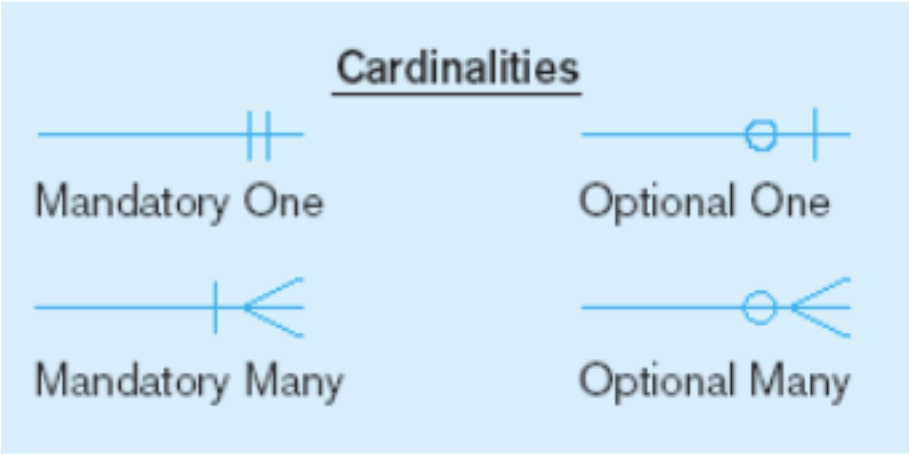
- Data files are integrated into a single, logical structure. Each fact is recorded ideally in only one place in the database. Redundancy cannot be totally eliminated.

3. Improved data consistency

- By eliminating data redundancy, you greatly reduce the chance of data

inconsistency.

Symbol	Meaning
	One and only one
	One or more
	Zero or more
	Zero or one



When should a relationship with attributes instead be an associative entity?

- All relationships for the associative entity should be many-to- many.

A **composite attribute** is one that is composed of smaller parts.
A composite key is a combination of two or more columns in a table that can be used to uniquely identify each row in the table.

A **derived attribute** is an **attribute** whose value is calculated (**derived**) from other **attributes**. The **derived attribute** need not be physically stored within the database; instead, it can be **derived** by using an algorithm. Eg. Studentage – calculated through d.o.b

A **multivalued attribute** can have more than one value at a time for an **attribute**. For ex., skills of a surgeon is a **multivalued attribute** since a surgeon can have more than one skill.]

Translating LDM to PDM:
- Assignment of DBMS data types
- Name abbreviation (if necessary)
- Identifying non-key indexes
- Assignment of storage (e.g., partitioning and tablespace assignment)
- Generation of the data definition language (DDL) to create/update the database

- Transforming entities into tables
- Transforming attributes into columns

Transforming domains into data types and constraints
To support the mapping of attributes to table columns you will need to map each logical domain of the attribute to a physical data type and perhaps additional constraints. In a physical database, each column must be assigned a data type. Certain data types require a maximum length to be specified. For example a character data type could be specified as CHAR(25), indicating that up to 25 characters can be stored for the column. You may need to apply a length to other data types as well, such as graphic, floating point, and decimal (which require a length and scale) types. In addition to a data type and length, you also may need to apply a constraint to the column. Consider a domain of integers between 1 and 10 inclusive. Simply assigning the physical column to an integer data type is insufficient to match the domain. A constraint must be added to restrict the values that can be stored for the column to the specified range, 1 through 10. Without a constraint, negative numbers, zero, and values greater than ten could be stored. Using check constraints you can place limits on the data values that can be stored in a column or set of columns.
Specification of a primary key is an integral part of the physical design of entities and attributes. A primary key should be assigned for every entity in the logical data model. As a first course of action you should try to use the primary key as selected in the logical data model. However, multiple candidate keys often are uncovered during the data modeling process. You may decide to choose a primary key other than the one selected during logical design – either one of the candidate keys or another surrogate key for physical implementation. But even if the DBMS does not mandate a primary key for each table it is a good practice to identify a primary key for each physical table you create. Failure to do so will make processing the data in that table more difficult.

Horizontal partitioning divides a table into multiple tables. Each table then contains the same number of columns, but fewer rows.
The two types of **vertical partitioning** are normalization and row splitting:
Normalization is the standard database process of removing redundant columns from a table and putting them in secondary tables that are linked to the primary table by primary key and foreign key relationships.

Supertype is an entity type that has got relationship (parent to child relationship) with one or more subtypes and it contains attributes that are common to its subtypes.

Subtypes are subgroups of the supertype entity and have unique attributes, but they will be different from each subtype.

Database **normalization**, or simply **normalization**, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce **data** redundancy and improve **data** integrity.

Specialization is the reverse process of Generalization means creating new sub classes from an existing class.
The process of extracting common characteristics from two or more classes and combining them into a generalized superclass, is called Generalization.

EER Diagram
Overlap (circle within circle) – Super class value can be more than one of the sub classes
Disjoint (D within circle) – Super class value cannot be more than one of the sub classes

Sub classes – circle and U

Double line from super class to O – Super class must be a sub class.

A **subtype discriminator** is the attribute in the supertype entity that determines to which **subtype** the supertype occurrence is related.

Strong vs Weak entity
The entity set which does not have sufficient attributes to form a primary key is called as Weak entity set. An entity set that has a primary key is called as Strong entity set.