DATABASE

and
Database Management (DBM)
(Health Informatics)

Colonel Zulfiquer Ahmed Amin
M Phil, MPH, PGD (Health Economics), MBBS
Armed Forces Medical Institute (AFMI)

WHAT IS DATABASE (DB)?

A database is an organized collection of data, which can be used:

- alone, or
- combined /

related to other data

for multiple purposes.

DATABASE MANAGEMENT SYSTEMS

A database management system (DBMS) is a collection of programs that enables to store, modify, and extract information from a database

DB & DBMS

“Database" refers to a set of related data and the way it is structured or organized.

Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data)

KEY CHARACTERISTICS OF DBMS

Key characteristics of DBMS are:

- Performance
- store large volume of database
- share data (access)
- provide security (authorization)
- remove redundancy (Duplication) and
- provide concurrent access (different users at the same time).


Data Center

PURPOSE OF DATABASE MANAGEMENT

Database management systems were developed to handle the following difficulties of typical file-processing systems supported by conventional operating systems:

PURPOSE OF DATABASE MANAGEMENT

- Data redundancy (Duplication) and inconsistency
- Difficulty in accessing data
- Data isolation – multiple files and formats. Isolating data as much as possible can keep malware from spreading and contain it to one unit.
- Integrity problems: **Data integrity** is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle.

- Atomicity of updates: In computer science, **ACID** (*Atomicity*, *Consistency*, *Isolation*, *Durability*) of database transactions intended to guarantee validity even in the event of errors, power failures, etc. Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged.
- Concurrent access by multiple users
- Security problems

ADVANTAGES OF A DBMS

- Data independence: Data independence is the idea that generated and stored data should be kept separate from applications that use the data for computing and presentation.
- Efficient data access: DBMS makes use of queries to access data from the database.
- Data integrity: Data integration means ensuring that the data in the database is accurate. Incorrect information gets rejected and cannot be stored in the database.

- Data administration: *Database administration* is the function of managing and maintaining *database management* systems (*DBMS*) software.
- Concurrent access: The ability to gain admittance to a system or component by more than one user.
- Crash recovery: DBMS is a highly complex system with hundreds of transactions being executed every second. If it fails or crashes amid transactions, it is expected that the system would follow some sort of techniques to recover lost data.

- Controlling redundancy: In a file system, each application has its own data storage system. That means some of the data may be repeated in each of these storage system increasing redundancy and wasting storage space. But when database is centralized, all data is stored only in one place.

- Data Security: With more users being able to access the data, security of data gains prime importance. Corporations spend a lot of time and energy to develop DBMS that enforce better data privacy and security policies. DBMS can enforce access controls to selectively make data visible to certain classes of users only. Access controls may even decide if a user is allowed to only retrieve data or if the user can update it too. Usually, databases are password protected to prevent unauthorized use.

- Reduced application development (RAD) time: In case of a DBMS, development of a new application using the same database takes very little time. Once a database is up and running, generally substantially less time is required to create new applications using DBMS facilities.
- Control data inconsistency: Data inconsistency occurs when the same data appears in different places. If a change is made in the data at one place but the same data value in another place is not updated, then it results in inconsistency. In a DBMS, all applications use the same set of centralized data. So a change in data needs to be done only in one place and the change gets updated in all applications using the database.

WHY NOT USE DBMS ALWAYS?

So why not use them always?

- Expensive/complicated to set up & maintain
- This cost & complexity must be offset by need
- General-purpose, not suited for special-purpose tasks: A common example is an email system that performs many of the functions of a general-purpose DBMS such as the insertion and deletion of messages composed of various items of data or associating messages with a particular email address; but these functions are limited to what is required to handle email and don't provide the user with all of the functionality that would be available using a general-purpose DBMS.

DATABASE MANAGEMENT SYSTEMS

Well-known DBMSs include:

- MySQL
- Microsoft Access
- Oracle
- dBASE
- FoxPro
- IBM DB2
- LibreOffice Base
- MariaDB
- PostgreSQL
- SQLite
- Microsoft SQL Server etc.

TYPES OF DBMS

A database model shows the structure of a database, including the relationships and constraints that determine how data can be stored and accessed.

- Hierarchical DBMS
- Network DBMS
- Relational DBMS
- Object-oriented DBMS

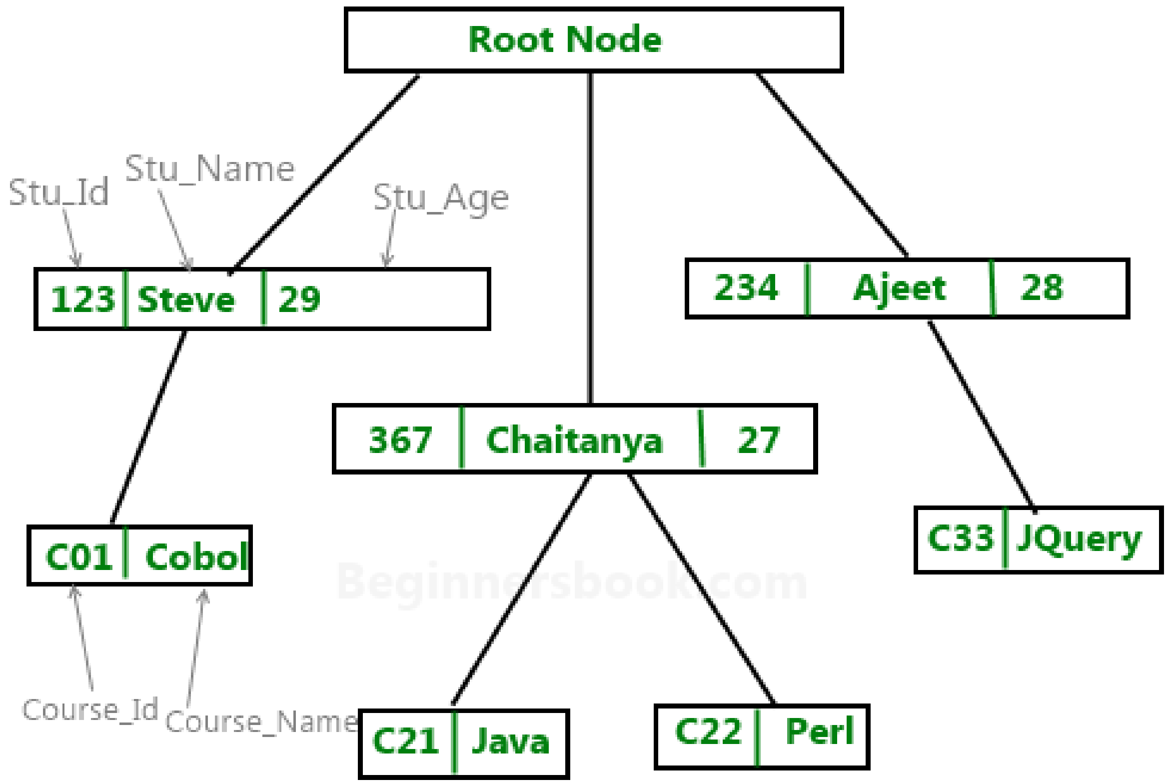
Hierarchical database model

In a **Hierarchical database model** is a data model where the data is organised like a tree. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent. All attributes of a specific record are listed under an entity type.

In a database, an entity type is the equivalent of a table; each individual record is represented as a row and an attribute as a column. Entity types are related to each other using *1: N* mapping, also known as one-to-many relationships.

In order to retrieve data from a hierarchical database the whole tree needs to be traversed starting from the root node. This model is recognized as the first database model created by IBM in the 1960s.

Hierarchical Model

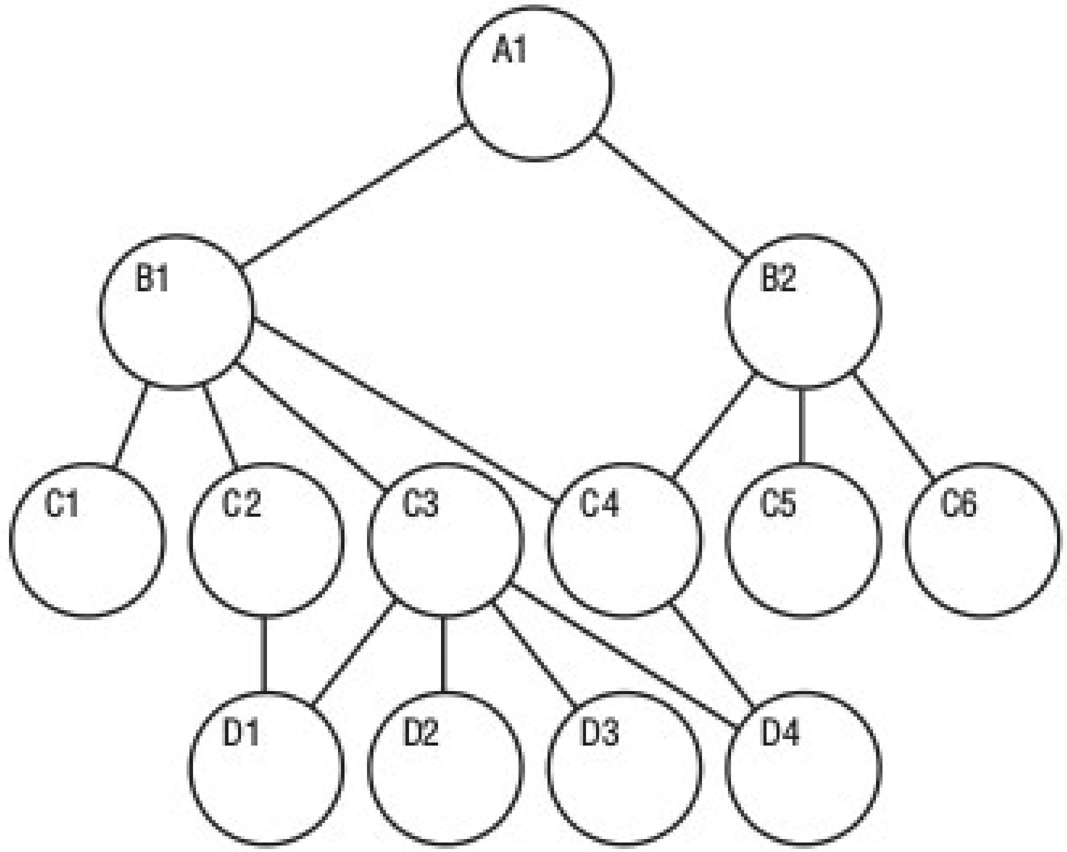


For example, where each student in a school reports to a given department, the department can be used as a parent record and the individual students will represent secondary records, each of which links back to that one parent record in a hierarchical structure.

A record in the hierarchical database model corresponds to a row (or tuple) in the relational database model and an entity type corresponds to a table (or relation).

Network Database Model

The network database model was a progression from the hierarchical database model and was designed to solve some of that model's problems, specifically the lack of flexibility. Instead of only allowing each child to have one parent, this model allows each child to have multiple parents (it calls the children *members* and the parents *owners*). It addresses the need to model more complex relationships such as the orders/parts many-to-many relationship mentioned in the hierarchical article.



As the figure shows, *A1* has two members, *B1* and *B2*. *B1.* is the owner of *C1*, *C2*, *C3* and *C4*. However, in this model, *C4* has two owners, *B1* and *B2*

Relational Database Design

Relational database was proposed by Edgar Codd (of IBM Research) around 1969. It has since become the dominant database model for commercial applications (in comparison with other database models such as hierarchical, network and object models). Today, there are many commercial *Relational Database Management System* (RDBMS), such as Oracle, IBM DB2 and Microsoft SQL Server.

A relational database organizes data in *tables* (or *relations*). A table is made up of rows and columns. A row is also called a *record* (or *tuple*). A column is also called a *field* (or *attribute*). A database table is similar to a spreadsheet. However, the relationships that can be created among the tables enable a relational database to efficiently store huge amount of data, and effectively retrieve selected data.

Student ID	First name	Last name
52-743965	Charles	Peters
48-209689	Anthony	Sondrup
14-204968	Rebecca	Phillips

ProviderID	Provider name
156-983	UnitedHealth
146-823	Blue Shield
447-784	Carefirst Inc.

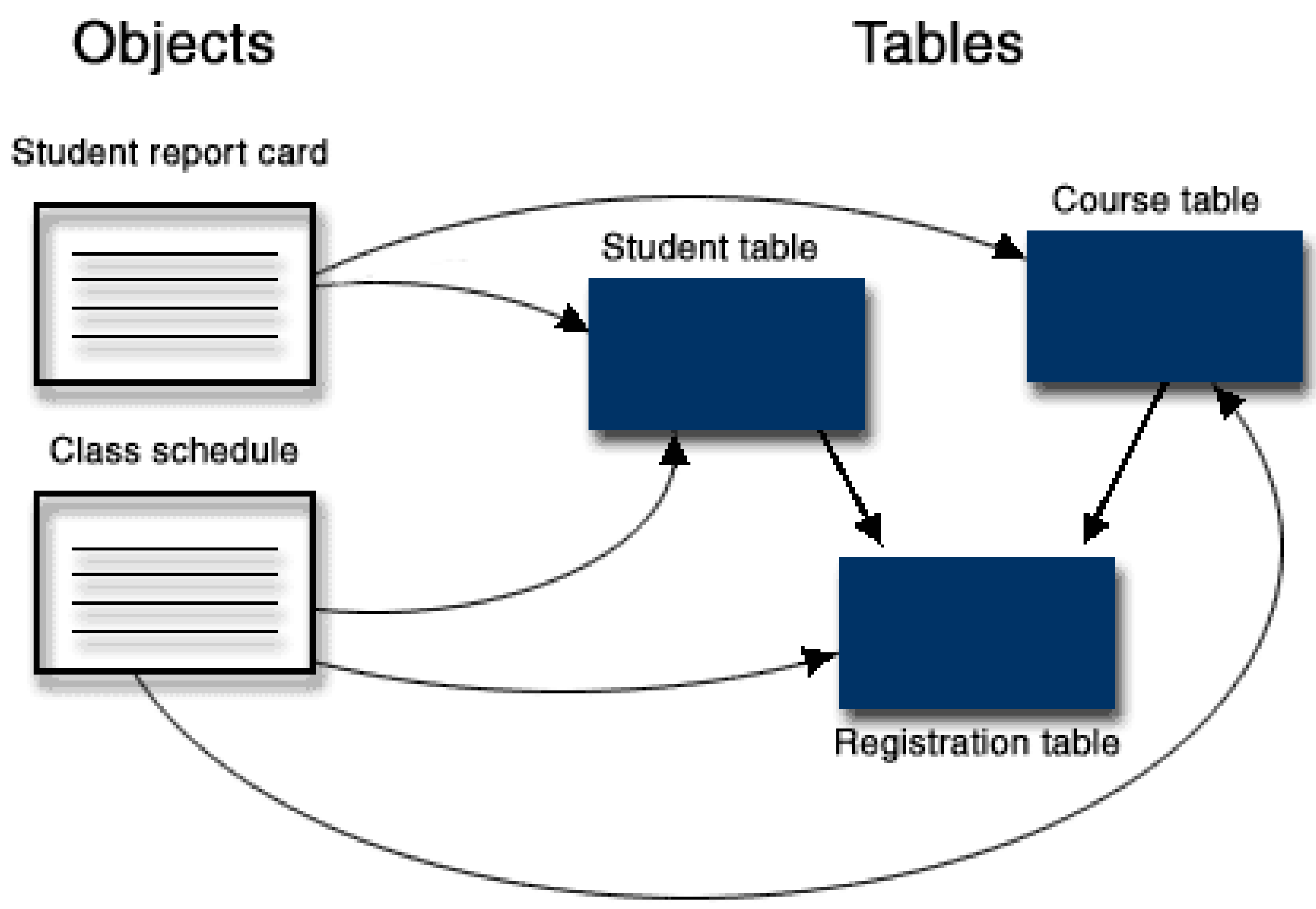
Student ID	ProviderID	Type of plan	Start date
52-743965	156-983	HSA	04/01/2016
48-209689	146-823	HMO	12/01/2015
14-204968	447-784	HSA	03/14/2016

The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships.

Object Oriented Database

An **object database** is a database management system in which information is represented in the form of objects. Object is the physical entity which exist in the universe.

- Engineering Database
- Multimedia Database


Objects
Tables
Student report card
Course table
Student table
Class schedule
Registration table

DATABASE DESIGN

It is important to design the database in such a way that:

- A specific item can be reached easily
- The database can respond to the user's different questions easily
- The database occupies minimum storage space

DATABASE DESIGN

It is important to design the database in such a way that (cont.):

- The database contains no unnecessary data

- Data can be added and updated easily without causing mistakes

STEPS IN DATABASE DESIGN

- Requirement analysis
- Conceptual database design
- Physical database design

STEPS IN DATABASE DESIGN

- **Requirement analysis**

  What does the user want?

- **Conceptual database design**

  Defining the entities and attributes, and the relationships between these --> The Entity-Relation model

STEPS IN DATABASE DESIGN

- **Physical database design**

Implementation of the conceptual design using a Database Management System

DATA ABSTRACTION

Data abstraction is the reduction of a particular body of data to a simplified representation of the whole.

DATA ABSTRACTION

- For the system to be usable, it must retrieve data efficiently.
- Since many database-systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users’ interactions with the system

Levels of Abstraction

The are three levels of abstraction are as follows:

Physical level: The lowest level of abstraction describes how data are stored.

Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.

View level: The highest level of abstraction describes only part of entire database.

LEVELS OF ABSTRACTION

Physical level:

The lowest level of abstraction describes how the data are actually stored. The physical level describes complex data structures in detail.

LEVELS OF ABSTRACTION

Logical level:

- The next-higher level of abstraction describes what data are stored in the database, and what relationships exist among those data.
- The logical level thus describes the entire database in terms of a small number of relatively simple structures.

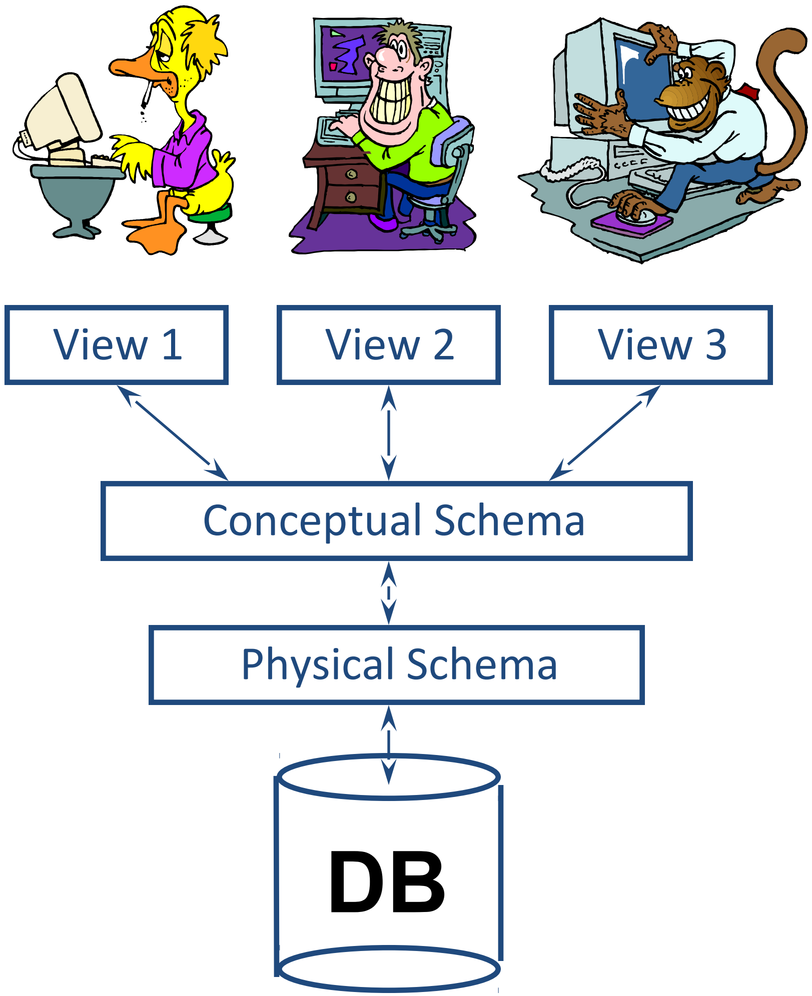
LEVELS OF ABSTRACTION

View level:

- The highest level of abstraction describes only part of the entire database.
- Many users of the database system do not need all this information; instead, they need to access only a part of the database.
- The view level of abstraction exists to simplify their interaction with the system.

LEVELS OF ABSTRACTION

- Views describe how users see the data.
- Conceptual schema defines logical structure
- Physical schema describes the files and indexes used.

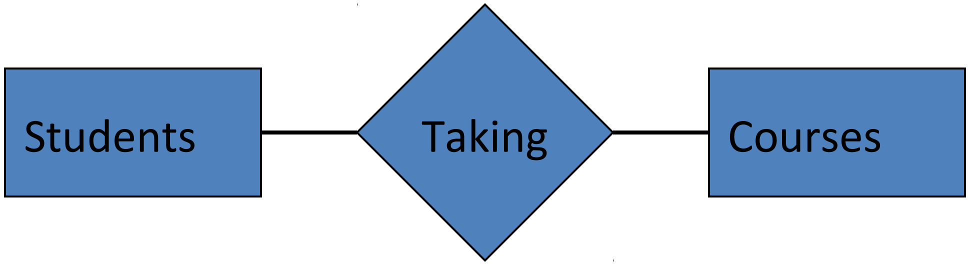


INSTANCE AND SCHEMES

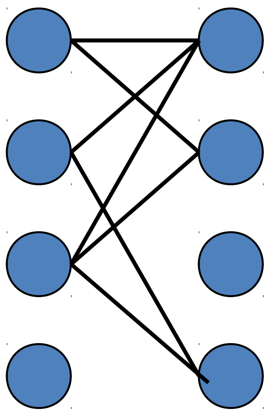
- The collection of database at a particular moment is called the instance of the database
- The overall design of the database is called the database scheme

RELATIONSHIPS

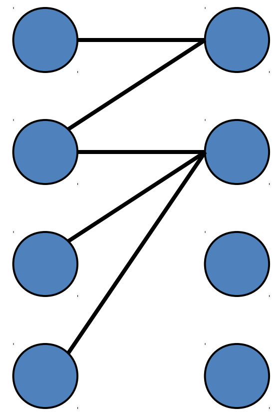
- Connect two or more entity sets.
- Represented by diamonds.



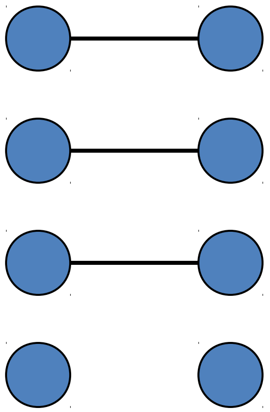
MULTIPLICITY OF RELATIONSHIPS



Many-many

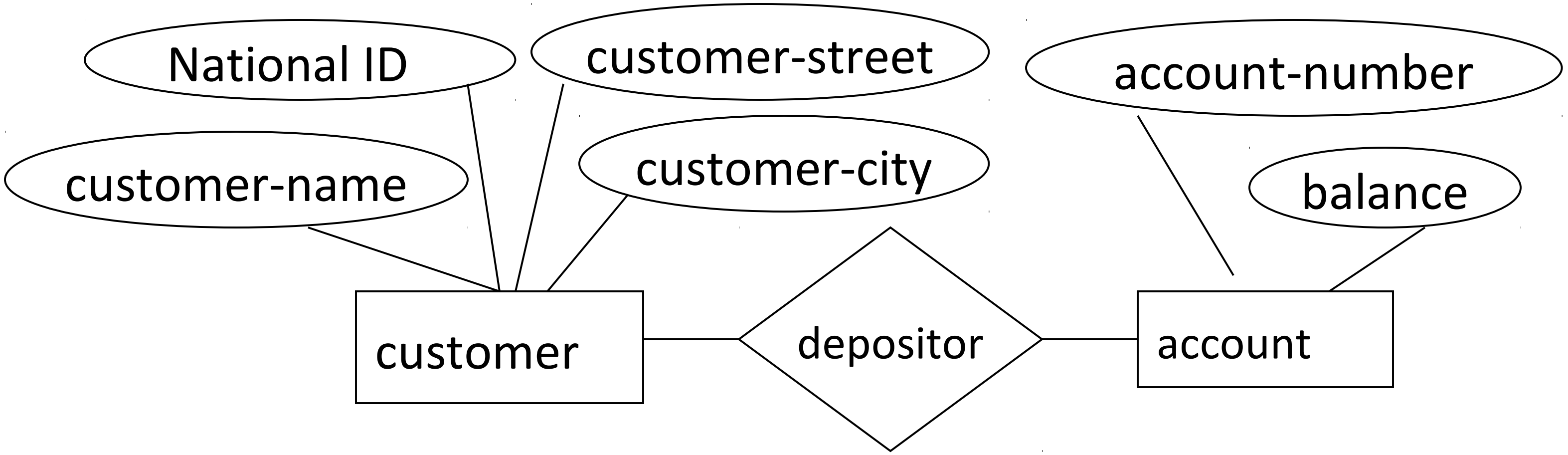


Many-one

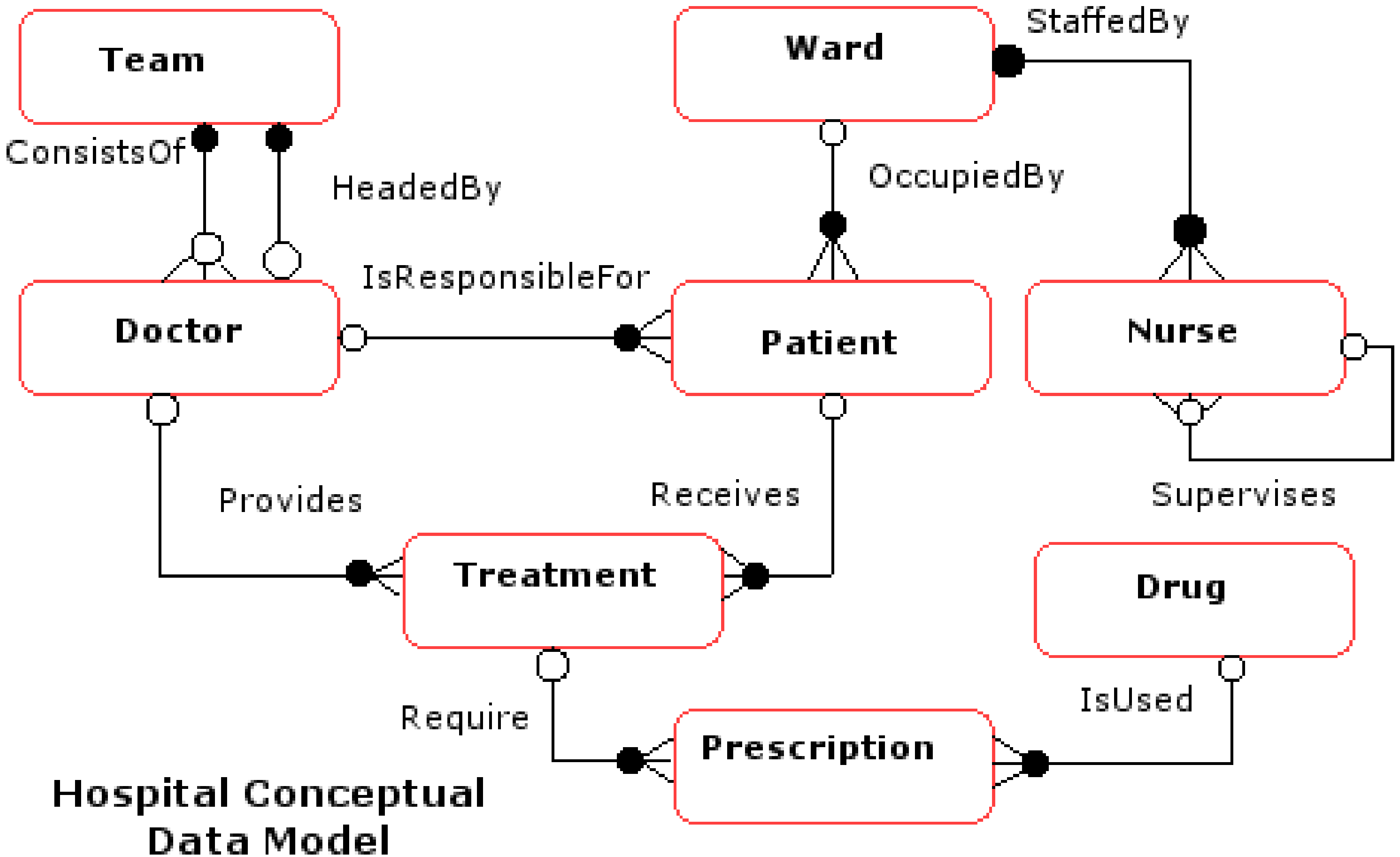


One-one

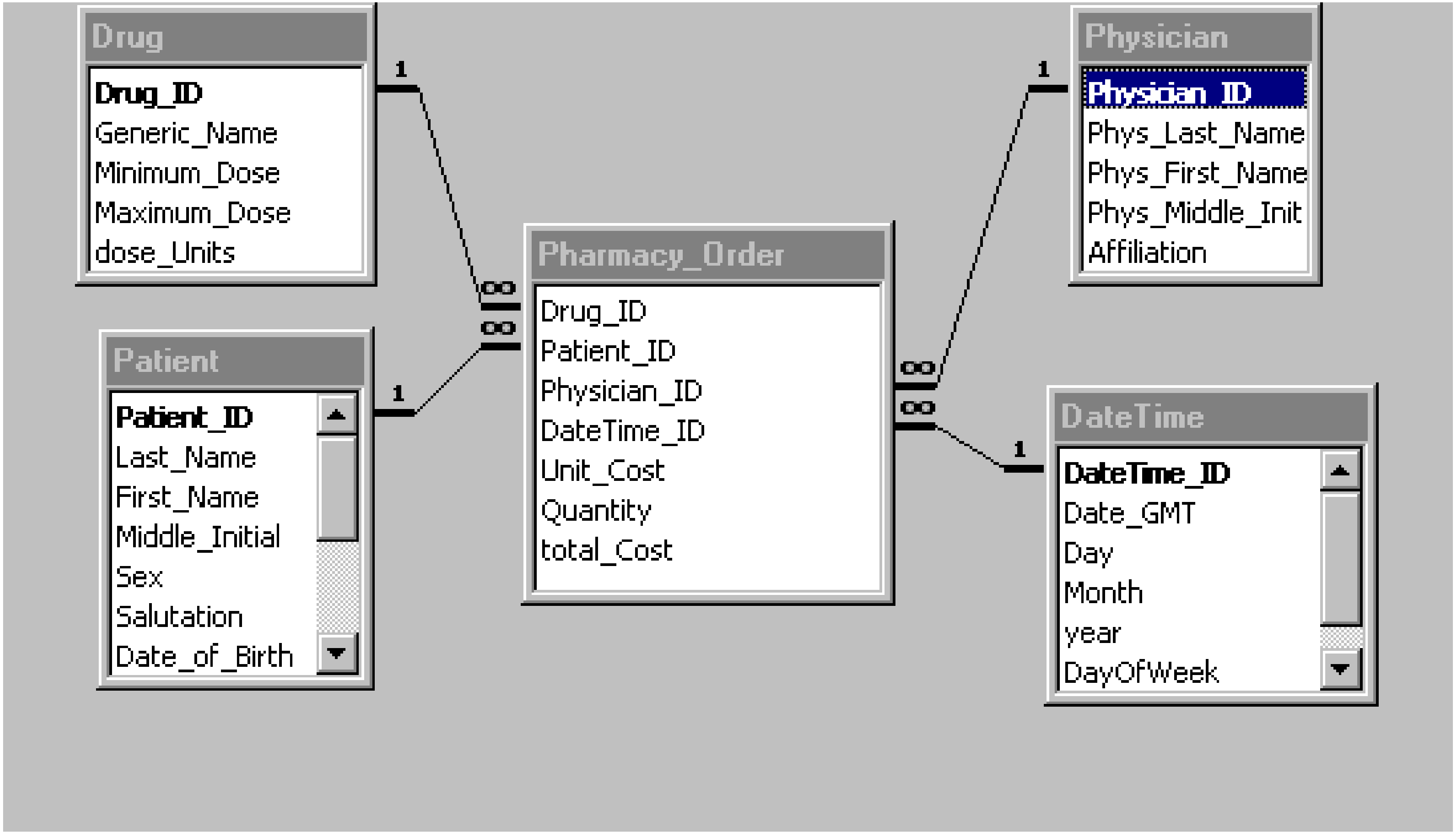
EXAMPLE OF ENTITY-RELATIONSHIP MODEL



HOSPITAL DATABASE E-R RELATIONSHIPS



PHARMACY DATABASE E-R RELATIONSHIP



DATA ADMINISTRATION

Data Administration

A high-level function that is responsible for the overall management of data resources in an organization, including maintaining corporate-wide definitions and standards

DATABASE ADMINISTRATION

Database Administration

A technical function that is responsible for physical database design and for dealing with technical issues such as security enforcement, database performance, and backup and recovery

DATA ADMINISTRATION FUNCTIONS

- Data policies, procedures, standards
- Planning
- Data conflict (ownership) resolution
- Managing the information repository
- Internal marketing of DA concepts

DATABASE ADMINISTRATION FUNCTIONS

- Selection of DBMS and software tools
- Installing/upgrading DBMS
- Tuning database performance
- Improving query processing performance
- Managing data security, privacy, and integrity
- Data backup and recovery

DATA WAREHOUSE

The Data Warehouse is a stable, read-only database that combines information from separate systems into one, easy-to-access location.

DATA WAREHOUSE