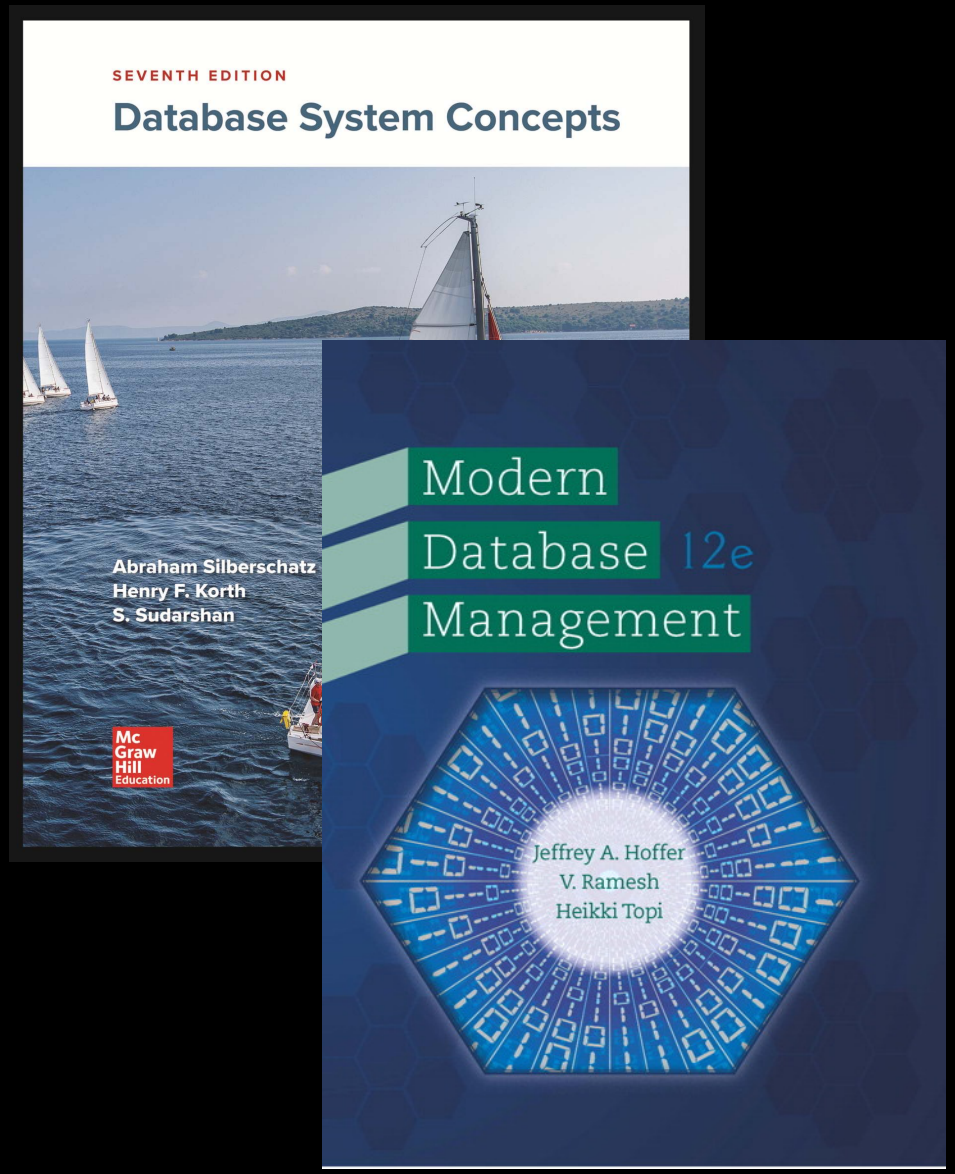
IF2240 – Basis Data

Data Model

KNOWLEDGE & SOFTWARE ENGINEERING



References

Abraham Silberschatz, Henry F. Korth, S. Sudarshan : "**Database System Concepts**", 7th Edition

- Chapter 1: Introduction

Jeffrey A. Hoffer, Mary B. Prescott, Heikki Topi : "**Modern Database Management**", 12th Edition

- Chapter 1: The Database Environment and Development Process

Data Modeling (1/2)

- A technique aimed at optimizing the way that information is stored and used within an organization
  - Begins with the **identification of the main data groups**, continues by **defining the detailed content** of each of these groups.
  - Result: **structured definitions** for all of the information that is stored and used within a given system.
- **An essential precursor** to analysis, design, maintenance & documentation and improving the performance of an existing system.

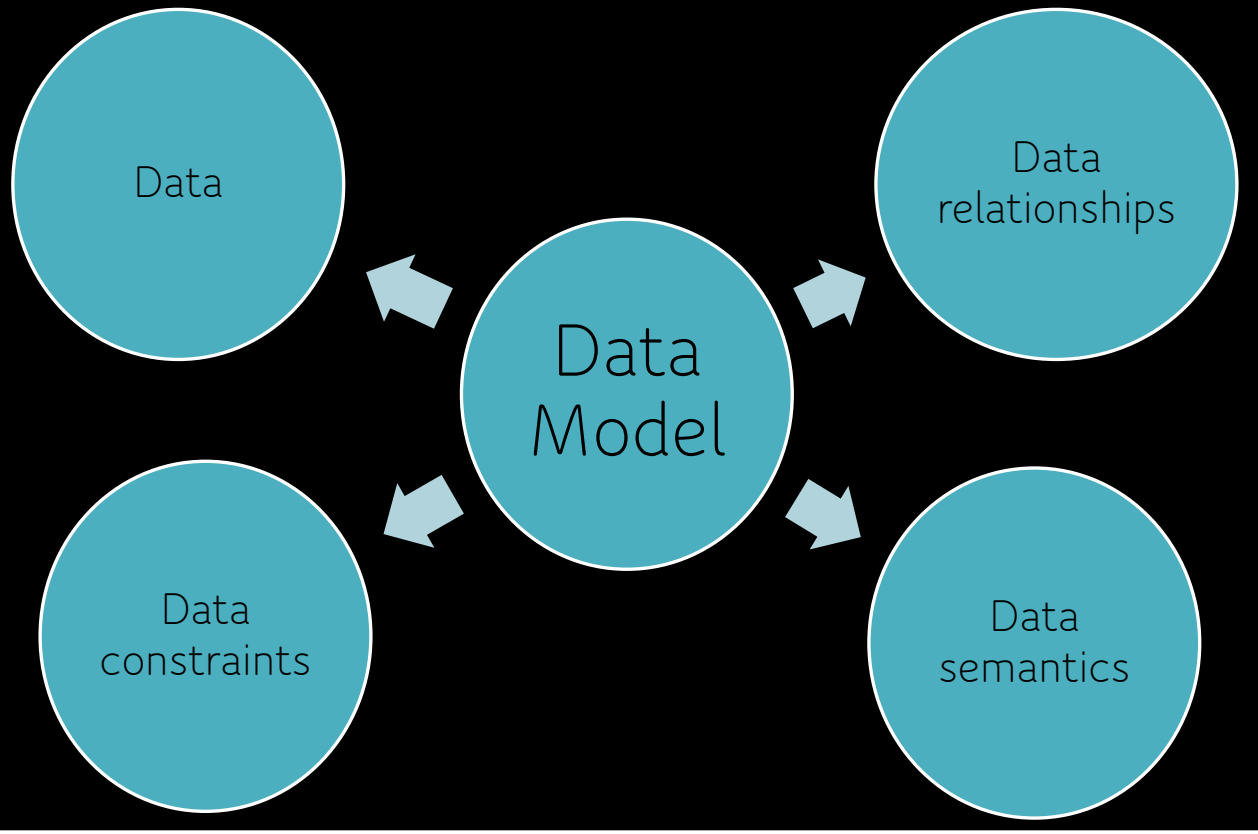
Data Modeling (2/2)

There are three different types of data models produced while progressing from requirements to the actual database.

- Conceptual data model: a set of technology independent specifications about the data and is used to discuss initial requirements with the business stakeholders.
- Logical data model: the structures of the data that can be implemented in databases.
- Physical data model: that organizes the data into tables, and accounts for access, performance and storage details.

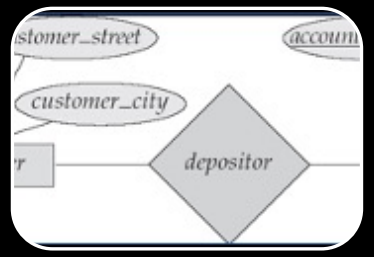
Data Model (1/2)

A collection of tools for describing:



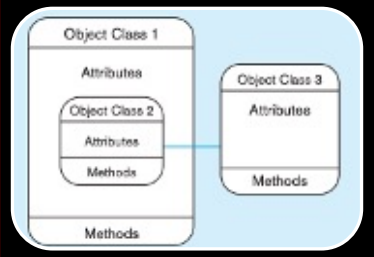
Data Model (2/2)

Types of data model:

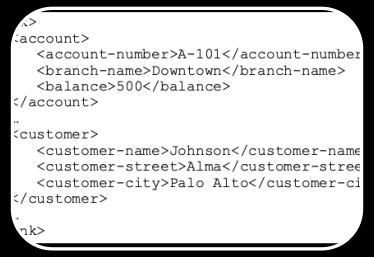


Entity-relationship model

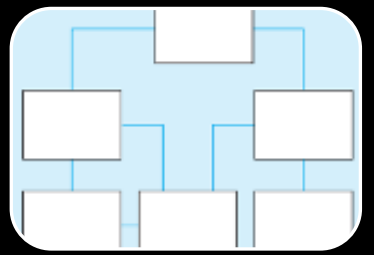
Relational model



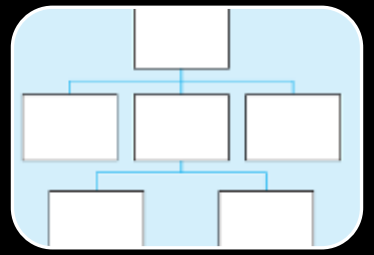
Object-based data model (object-oriented and object relational)



Semi-structured data model (XML)



Network model

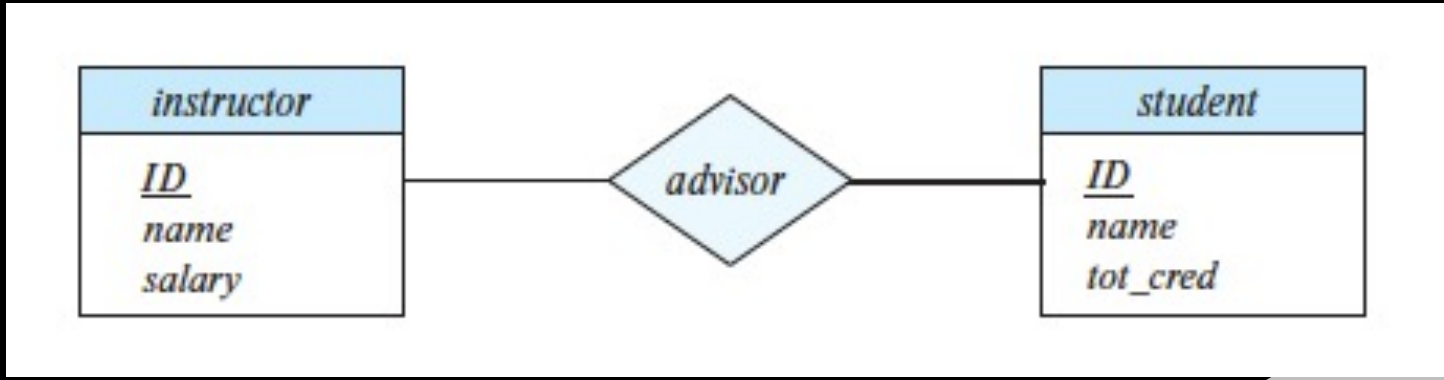


Hierarchical model

Older data models

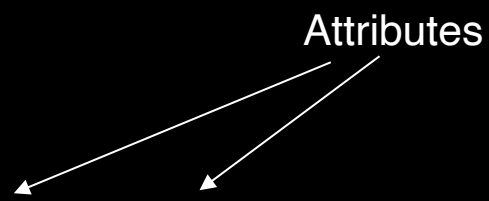
Entity Relationship Model

- Widely used for database design
  - Database design in E-R model usually converted into design in Relational Model
- Models an enterprise as a collection of *entities* and *relationships*
  - **Entity**: a "thing" or "object" in the enterprise that is distinguishable from other objects
  - **Relationship**: an association among several entities
- Represented diagrammatically by an *entity-relationship diagram* (ERD)



Relational Model

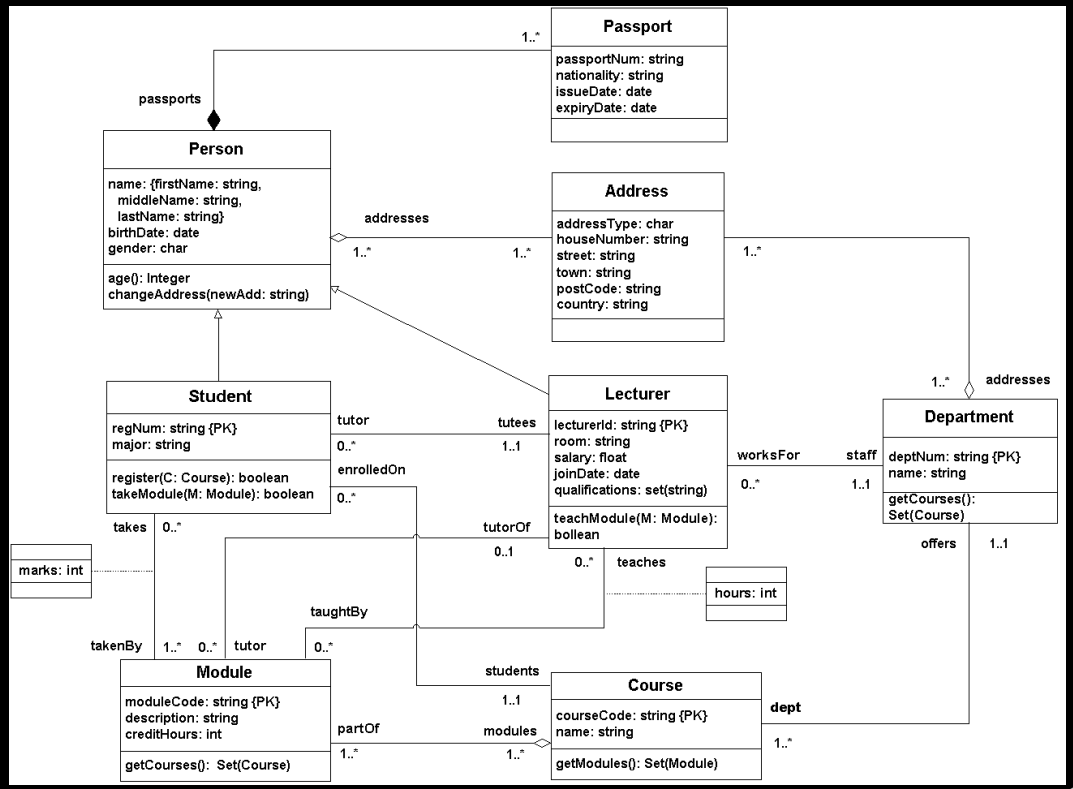
- Consist of collection of tables to represent data and the relationship among those data
- Example of tabular data in the relational model



customer_id	*customer_name*	*customer_street*	*customer_city*	*account_number*
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-101
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-201
677-89-9011	Hayes	3 Main St.	Harrison	A-102
182-73-6091	Turner	123 Putnam St.	Stamford	A-305
321-12-3123	Jones	100 Main St.	Harrison	A-217
336-66-9999	Lindsay	175 Park Ave.	Pittsfield	A-222
019-28-3746	Smith	72 North St.	Rye	A-201

Object-Oriented Data Model

- Adaptation of the object-oriented programming paradigm (e.g. Smalltalk, C++) to database systems
- The *object-oriented paradigm* is based on *encapsulating* code and data related to an object into single unit



Object Structure

An object has associated with it:

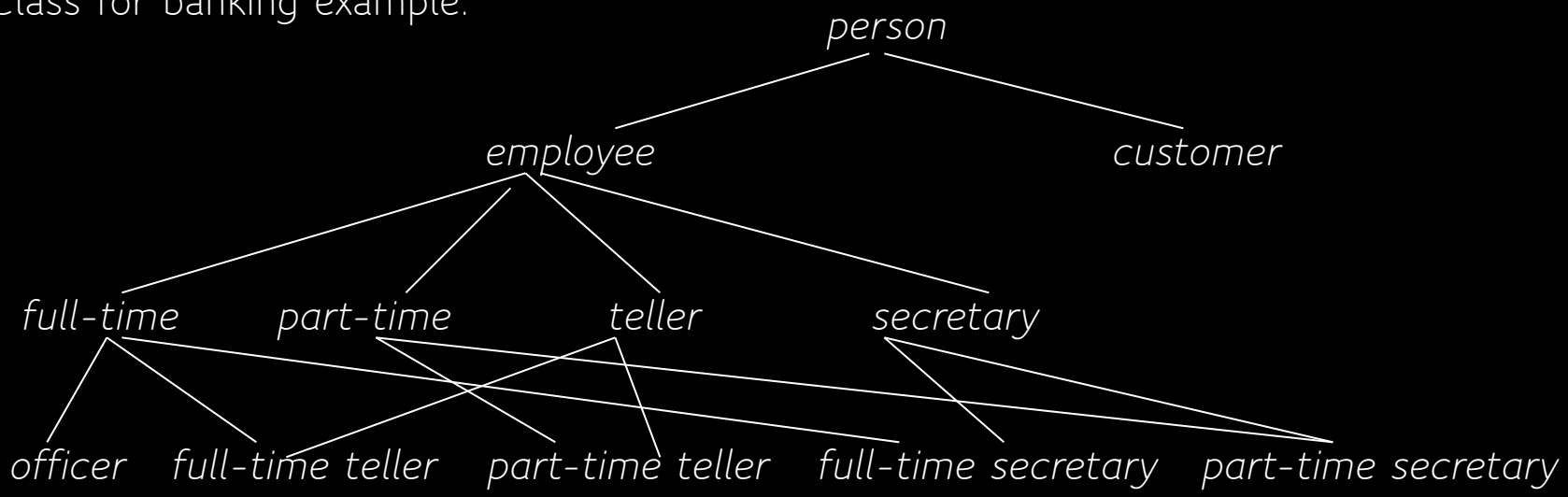
- **A set of variables** that contain the data for the object. The value of each variable is itself an object.
- **A set of messages** to which the object responds; each message may have zero, one, or more *parameters*.
- **A set of methods,** each of which is a body of code to implement a message; a method returns a value as the *response* to the message

```
class Person {
   /* variable */
      Name        name;
      date        birthDate;
      string      address;
      char        gender;
   /* messages */
      int         age();
      int         changeAddress(string newAdd);
      Name        getName();
      date        getBirthDate();
      string      getAddress();
      char        getGender();

};

string get-address() {
   return address;
}
```

Object-Relational Data Model

- Extend the relational data model by including object orientation and a richer type system including collection types
  - Object orientation provides inheritance with subtypes and sub tables
  - Collection types include nested relations, sets, multisets, and arrays
- Preserve relational foundations, in particular the declarative access to data, while extending modeling power

Class for banking example:



Semi Structured Data Model (ex: XML)

- Characteristics:
  - organized in semantic entities
  - similar entities are grouped together
  - entities in same group may not have same attributes
  - order of attributes not necessarily important
  - not all attributes may be required
  - size of same attributes in a group may differ
  - type of same attributes in a group may differ

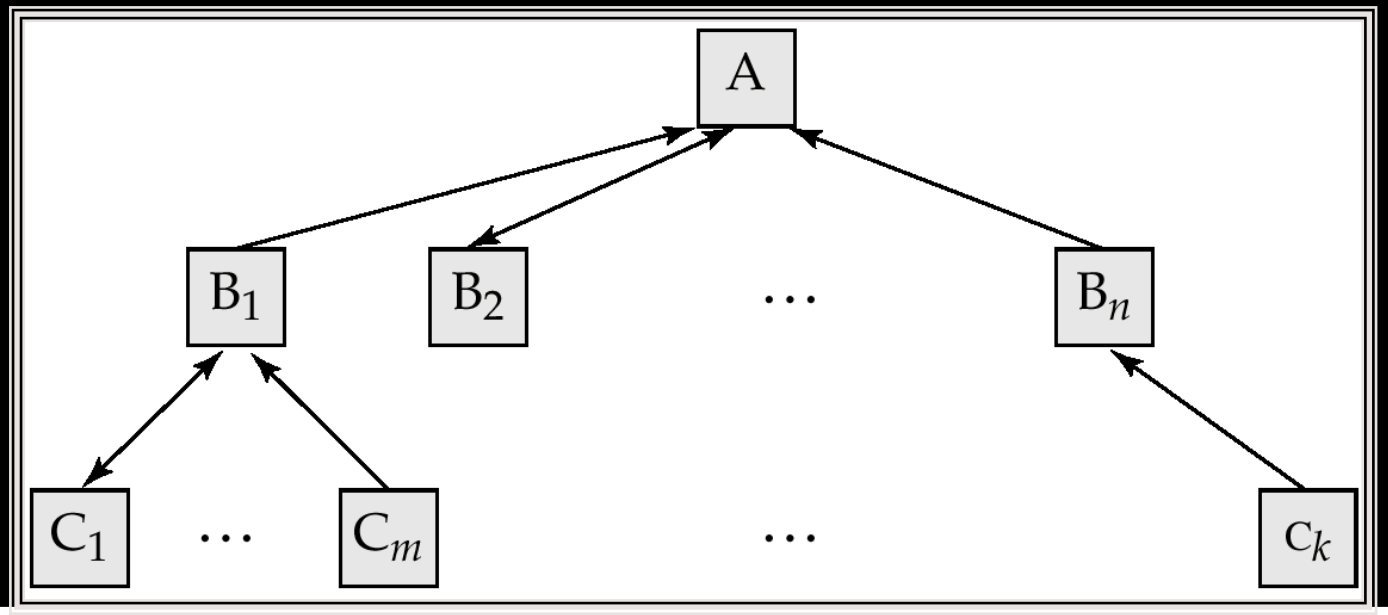
→Self-describing, irregular data, no a priori structure

- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**, not just documents
- XML has become the basis for all new generation data interchange formats.
- A wide variety of tools is available for parsing, browsing and querying XML documents/data

```
<bank>
   <account>
      <account-number>A-101</account-number>
      <branch-name>Downtown</branch-name>
      <balance>500</balance>
   </account>
   …
   <customer>
      <customer-name>Johnson</customer-name>
      <customer-street>Alma</customer-street>
      <customer-city>Palo Alto</customer-city>
   </customer>
   …
</bank>
```

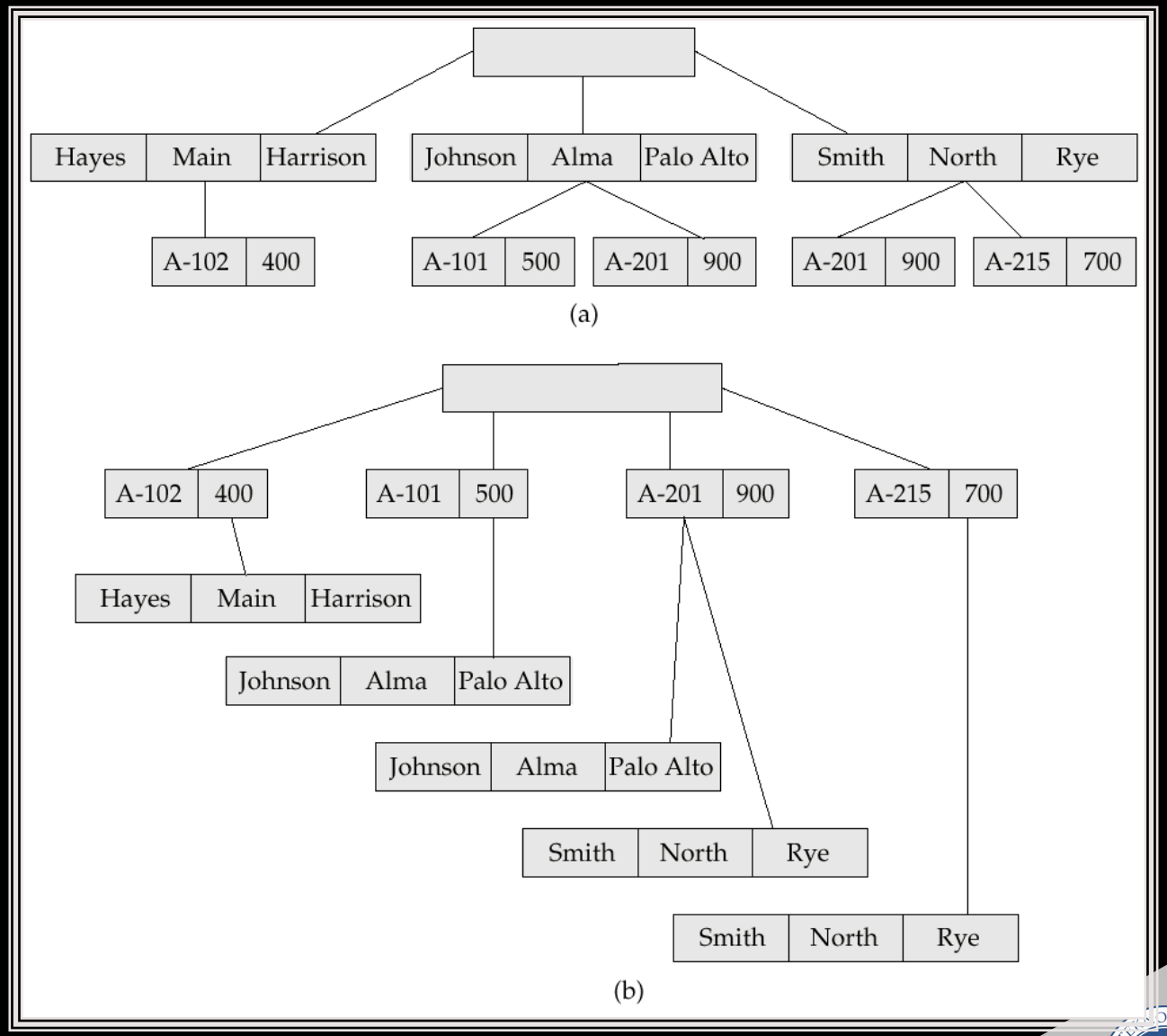
Hierarchical Model (1/2)

- World's leading mainframe hierarchical database system in the 1970s and early 1980s
- Database schema is represented as a collection of tree-structure diagrams
  - *Single* instance of a database tree
  - The root of this tree is a dummy node
  - The children of that node are actual instances of the appropriate record type
- The schema for a hierarchical database consists of
  - *boxes*, which correspond to record types
  - *lines*, which correspond to links
- Record types are organized in the form of a *rooted tree*



A parent *may* have an arrow pointing to a child, but a child *must* have an arrow pointing to its parent

Hierarchical Model (2/2)



Network Model (1/2)

- Data are represented by collections of *records*.
  - similar to an entity in the E-R model
  - Records and their fields are represented as *record type*
- Relationships among data are represented by *links*
  - similar to a restricted (binary) form of an E-R relationship
  - restrictions on links depend on whether the relationship is many-many, many-to-one, or one-to-one.

```
type customer = record
     customer-name: string;
     customer-street: string;
     customer-city: string;
end;
```

```
type account = record
     account-number: integer;
     balance: integer;
end;
```

Network Model (2/2)

- Graph like structure
  - Child may have multiple parent
- A data-structure diagram consists of two basic components:
  - **Boxes**, which correspond to record types
  - **Lines**, which correspond to links
- Specifies the overall logical structure of the database