Module - I

Lecture-06

DATA MODELS

- Relational Model
- Semi structured Model
- ER Model
- Other Models

Data Models

- Data Model is a collection of concepts that can be used to describe structure of the database, provide the necessary means to achieve abstraction.
- Structure of a database means data types, relationships and constraints that should hold for the data.
- Data Model also includes the set of basic operations for specifying retrievals and updates for the database.
- A data model is a logic organization of the real world objects (entities), constraints on them, and the relationships among objects.
- A database language is a concrete syntax for a data model.
- A database system implements a data model.

Data Models

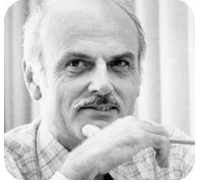
- Data Model is a collection of conceptual tools for describing
  - ✓ Data
  - ✓ Data relationships
  - ✓ Data semantics
  - ✓ Data constraints

Data Models

Categories of Data Models

- Relational Model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)
- Other older models:
  - ✓ Network model
  - ✓ Hierarchical model

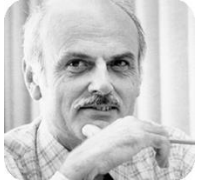
Relational Model



Edgar Frank "Ted" Codd
Turing Award 1981

- The relational data model was introduced by E. F. Codd in 1970.
- Currently, it is the most widely used data model.
- This model represents data and relationships among data by a collection of tables known as relations, each of which has a number of columns with unique names.
- Relational model relates records by the values they contain.

Relational Model

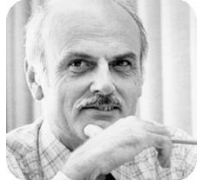


Edgar Frank "Ted" Codd
Turing Award
1981

- One of the main advantages of the relational model is that it is conceptually simple and more importantly based on mathematical theory of relation.
- It is also frees the users from details of storage structure and access models.

Relational Model

- Example of tabular data in the relational model



Edgar Frank "Ted" Codd
Turing Award
1981

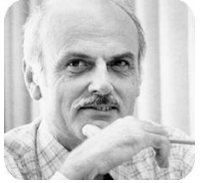
Columns

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Rows

(a) The *instructor* table

A Sample Relational Database



Edgar Frank "Ted" Codd
Turing Award
1981

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

The Entity-Relationship Model

- A database can be modeled as:
  - ✓ a collection of *entities*
  - ✓ *relationships* among entities
- Technique called Entity-Relationship Modeling (ER 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
    - Described by a set of attributes
  - ✓ Relationship: an association among several entities

The Entity-Relationship Model

- An entity is an object that exists and is distinguishable from other objects
  - ✓ Example: specific person, company, event, plant
  - ✓ Entities are usually expressed by nouns
- Entities have properties denotes as attributes
  - ✓ Example: people have *names* and *addresses*
  - ✓ An entity set is a set of entities of the same type that share the same properties.

The Entity-Relationship Model

- An entity set is a set of entities of the same type that share the same properties.
  - ✓ Example: set of persons, companies, trees, loans

customer_id	customer_name	customer_street	customer_city
12-345	Johnson	12 Alma St.	Palo Alto
12-346	Hayes	22 Main St.	Bromfield
12-358	Smith	45 Park Ave.	Berkeley
25-836	Brown	33 High St.	Auckland
35-795	Jones	26 Almond St.	Oakwood
45-678	Turner	123 Putnam St.	Stanford

customer

loan_id	amount
L-101-A	1500
L-201-A	2700
L-102-C	1450
L-118-D	3800
L-249-B	2550
L-157-A	6350

loan

The Entity-Relationship Model

- Attributes: An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

  ✓ Example:

  customer = (customer_id, customer_name, customer_street, customer_city)

  loan = (loan_number, amount)

The Entity-Relationship Model

- Represented diagrammatically by an Entity-Relationship Diagram:

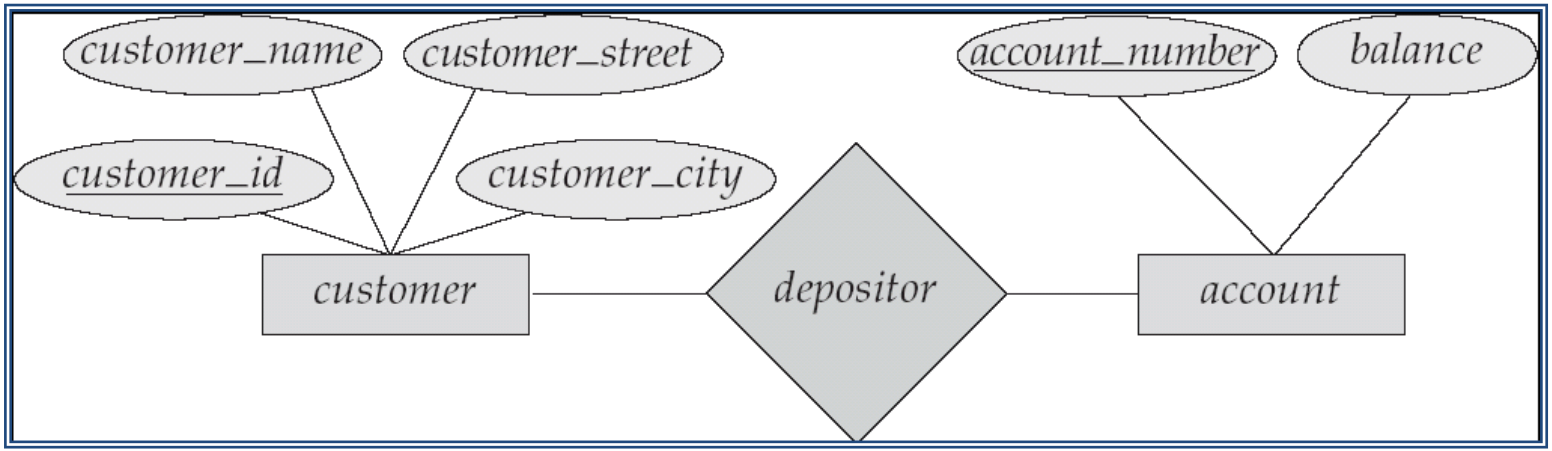
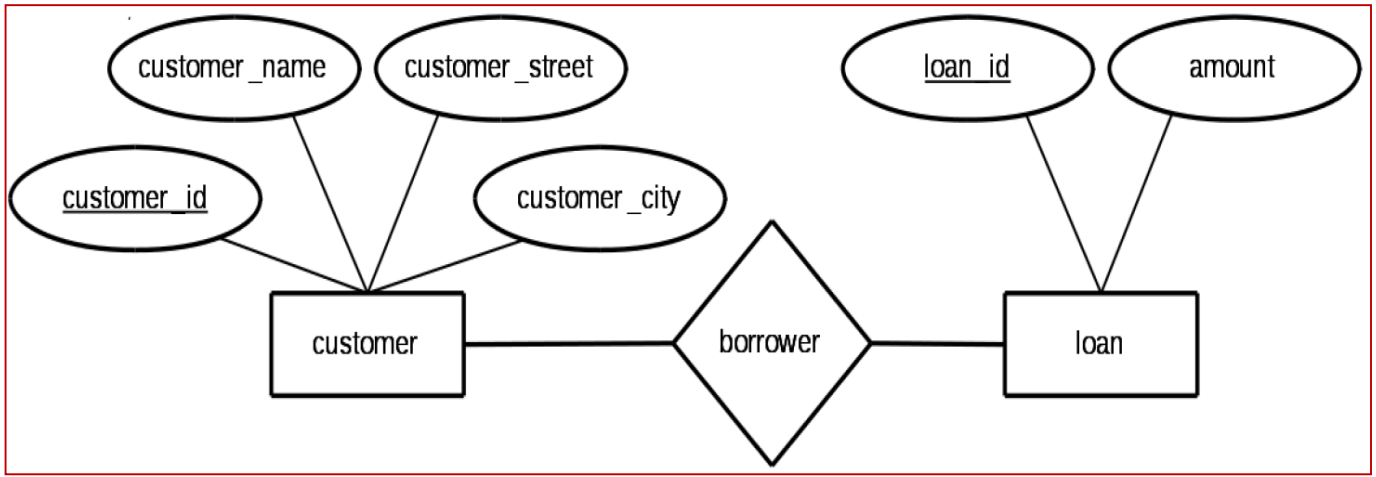


Fig. **Entity-Relationship Diagram (ERD)**

The Entity-Relationship Model - Example



- Rectangles represent entity sets
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
  - ✓ Double ellipses represent multivalued attributes.
  - ✓ Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes

Object-based Data Models (Object-Oriented and Object-Relational)

- An Object-Oriented data model is a logic organization of the real world objects (entities), constraints on them, and the relationships among objects.
- Extend the relational data model by including object orientation and constructs to deal with added data types.
- Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.

Object-based Data Models (Object-Oriented and Object-Relational)

- Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
- Provide upward compatibility with existing relational languages.

Object-based Data Models (Object-Oriented and Object-Relational)

A core Object-Oriented data model consists of the following basic Object-Oriented concepts:

i. Object and Object Identifier: Any real world entity is uniformly modeled as an object. For example, person, car, place etc.

ii. Attributes and Methods: Every object has a state (the set of values for the attributes of the object) and a behavior (the set of methods - program code-which operate on the state of the object). The state and behavior *encapsulated* in an object are accessed or invoked from outside the object only through explicit message passing.

Object-based Data Models (Object-Oriented and Object-Relational)

iii. Class: A means of grouping all the objects which share the same set of attributes and methods. An object must belong to only one class as an instance of that class. A class is similar to an abstract data type. A class may also be primitive (no attributes), ex., integer, string, Boolean etc.

iv. Class Hierarchy and Inheritance: Derive a new class (sub class) from an existing class (super class). The sub class inherits all the attributes and methods of the existing class and may have additional attributes and methods.

Semi-structured data model (XML: Extensible Markup Language)

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
- 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

Network model

- This model represents data by collection of records and relationships among data.
- This is represented by links, which can be viewed as pointers.
- Network model consisting of the following three components:

  i. Record type: A record type represents a finite number of similar type entities.

  ii. Data elements: Entities are distinguished by the values of the data elements with which the corresponding record type is associated.

Network model

iii. Links: In the network data model, all relationships between the same or different record types are restricted to binary, many-to-one relationships. These many-to-one relationships are called links.

Network model

Example: Network model for departmental store

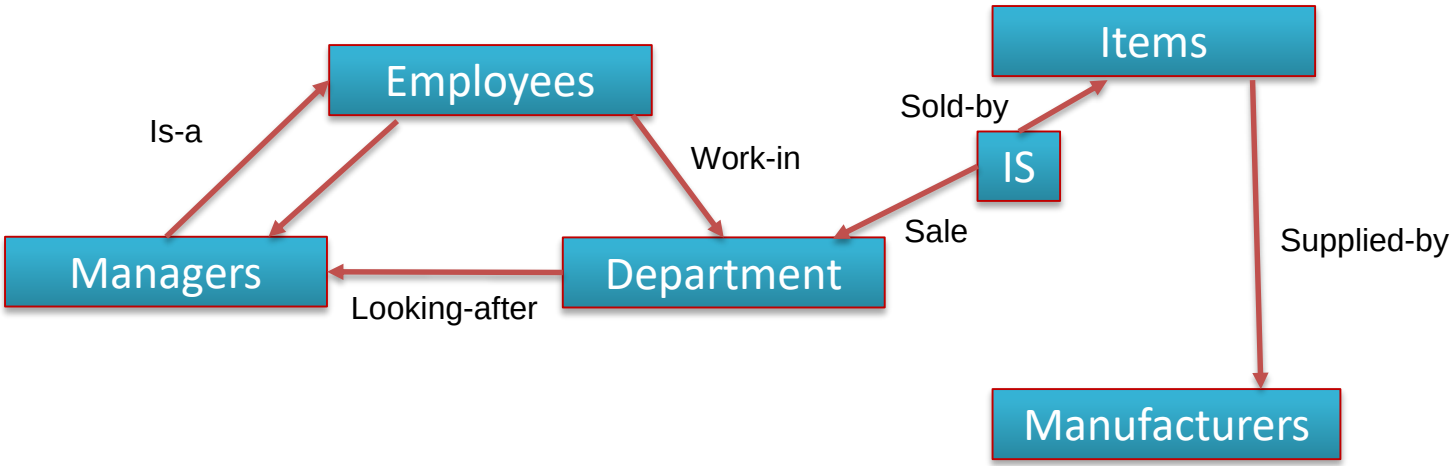


Fig. **Network Model for Department Store**

Hierarchical Model

- This model is similar to network model in the sense that data and relationships among data are represented by records and links respectively.
- It differs from network model in that the records are organized as collection of trees rather than arbitrary graphs.
- A Database Management System (DBMS) belonging to the hierarchical data model uses tree structures to represent relationship among records.

Hierarchical Model

- A hierarchical database therefore consists of a collection of records, which are connected with each other through links.
- Each record is a collection of fields (attributes), each of which contains one data value.
- A link is an association between precisely two records.

Hierarchical Model

Example: Consider the following *Employee* hierarchy.

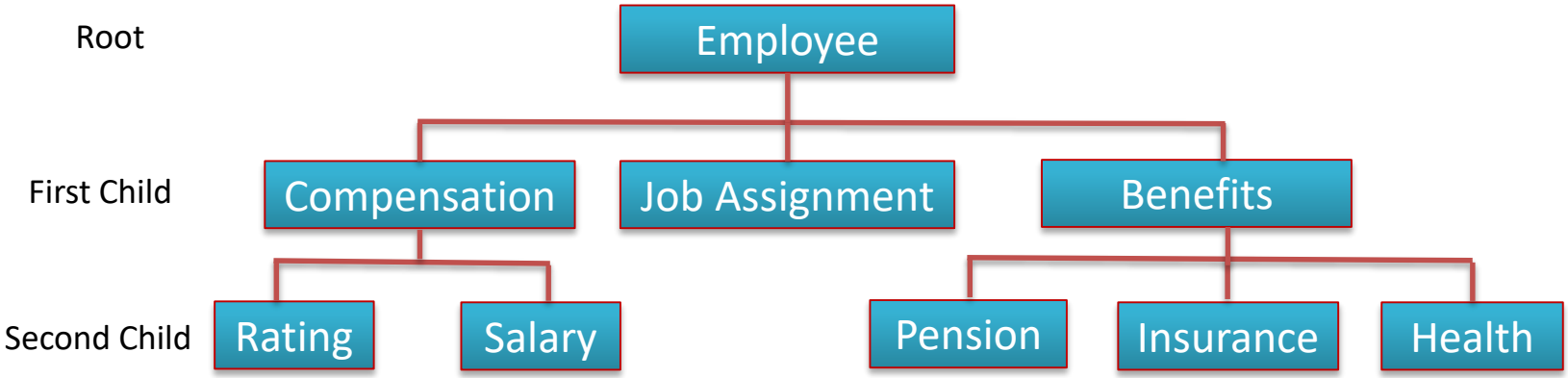


Fig. **Hierarchical Model**