❖ 2-1 Data Modeling and Data Models

Data modeling- the first step in designing a database, refers to the process of creating a specific data model for a determined problem domain(well-defined scope/boundaries to be addressed systematically).

Data Model (database model)- relatively simple representation, usually graphical, of more complex real-world data structures. (For understanding Real-World complexities) Representing data structures and their characteristics, relations, constraints, transformations, and other constructs with the purpose of supporting a specific problem domain. Final model is a "blueprint" for the comprehensive Database Design, which will include atleast:

- ✓ A description of the data structure that will store the end-user data
- ✓ A set of enforceable rules to guarantee the integrity of the data
- ✓ A data manipulation methodology to support the real-world data transformations

❖ 2-2 The Importance of Data Models

- ✓ A good blueprint is needed to make a successful Database Design.

❖ 2-3 Data Model Basic Building Blocks

✓ The basic building blocks of all data models are entities, attributes, relationships, and constraints.

Entity – Person, place, thing, or event about which data will be collected and stored.

Attribute – Characteristic of an Entity. Equivalent to Fields.

Relationship - describes an association among entities.

3 Types of Relationships – One-to-Many[1:M or 1..*], Many-to-Many[M:N or *..*], & One-to-One[1:1 or 1..1]. and can go both ways, bidirectional.

- **One-to-many relationship** – Ex: Guitarist-to-Song. Expressed as: GUITARIST plays SONG
- **Many-to-many relationship** –Students can take many classes and each class can be taken by many students. Ex: M:N Label / expressed as: STUDENT takes CLASS
- **One-to-one relationship** – If a company gives company cars to Managers. Ex: 1:1 Label / expressed as: MANAGER drives CAR

Constraint – Rules/Restriction placed on the data. Ex: GPA must be between 0.00 & 4.00

❖ 2-4 Business Rules

Business rule - precise, clear description of a policy, procedure, or principle within a specific organization. EX: Pilots can only fly 10 hours out of 24 hours.

- ✓ Properly written business rules are used to define entities, attributes, relationships, and constraints.

○ 2-4a Discovering Business Rules

- ✓ Main Source: company managers, policy makers, department managers, and written documentation such as a company's procedures, standards, and operations manuals.
- ✓ Identifying/Documenting Business Rules is essential to database design to: standardize the company's view of data/ communication tool between users and designers/ allows the designer to understand the nature, role, and scope of the data/ allows the designer to understand business processes/ allows the designer to develop appropriate relationship participation rules and constraints and to create an accurate data model.

○ 2-4b Translating Business Rules into Data Model Components

✓ As a general rule, a noun in a business rule will translate into an entity in the model, and a verb (active or passive) that associates the nouns will translate into a relationship among the entities.

○ 2-4c Naming Conventions

- ✓ Entity names should be descriptive of the objects in the business environment and use terminology that is familiar to the users.

❖ 2-5 The Evolution of Data Models

✓ EVOLUTION FOCUS: what a database is, what it should do, the types of structures that it should employ, and the technology that would be used to implement these structures.

Evolution of Major Data Models

Generation	Time	Data Model	Examples	Comments
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS, ADABAS, IDS-II	Early database systems Navigational access
Third	Mid-1970s	Relational	DB2 Oracle MS SQL Server MySQL	Conceptual simplicity Entity relationship (ER) modeling and support for relational data modeling
Fourth	Mid-1980s	Object-oriented Object/relational (O/R)	Versant Objectivity/DB DB2 UDB Oracle 12c	Object/relational supports object data types Star Schema support for data warehousing Web databases become common
Fifth	Mid-1990s	XML Hybrid DBMS	dbXML Tamino DB2 UDB Oracle 12c MS SQL Server	Unstructured data support O/R model supports XML documents Hybrid DBMS adds object front end to relational databases Support large databases (terabyte size)
Emerging Models: NoSQL	Early 2000s to present	Key-value store Column store	SimpleDB (Amazon) BigTable (Google) Cassandra (Apache) MongoDB Riak	Distributed, highly scalable High performance, fault tolerant Very large storage (petabytes) Suited for sparse data Proprietary application programming interface (API)

- **2-5a**Hierarchical and Network Models

Hierarchical Model-1960's database model whose basic concepts and characteristics formed the basis for subsequent database development. This model is based on an upside-down tree structure in which each record is called a segment. The top record is the root segment. Each segment has a 1:M relationship to the segment directly below it. The higher layer is perceived as the parent of the segment directly beneath it, which is called the child.

Segment- the equivalent of a file system's record type. Also called Levels.

NETWORK MODEL - was created to represent complex data relationships more effectively than the hierarchical model, to improve database performance, and to impose a database standard. ---the network model allows a record to have more than one parent.

- Lacks of ad hoc query capability, which required Extensive Programming for Simple Reports. Limited Data Independence, though change could wreak havoc. Standard database *CONCEPTS* that emerged with the network model:
  - **Schema -** A logical grouping of database objects, such as tables, indexes, views, and queries, that are related to each other.
  - **Subschema-** portion of the database "seen" by the application programs that actually produce the desired information from the data within the database.
  - **Data Manipulation Language(DML) -** The set of commands that allows an end user to manipulate the data in the database, such as SELECT, INSERT, UPDATE, DELETE, COMMIT, and ROLLBACK.
  - **Data Definition Language(DDL) -** The language that allows a database administrator to define the database structure, schema, and subschema.

- **2-5b**The Relational Model

- 1970s, based on mathematical set theory and represents data as independent relations. Each relation (table) is conceptually represented as a two dimensional structure of intersecting rows and columns. The relations are related to each other through the sharing of common entity characteristics (values in columns). Most relational database software uses Structured Query Language (SQL). Ingenious, but Impractical at the time, because Computers were much less powerful and more expensive.
- **Relation** = Table. **Tuple** = a Row in a Table. Columns represent an Attribute.

- **Relational Database Management System (RDBMS) -** A collection of programs that manages a relational database. The RDBMS software translates a user's logical requests (queries) into commands that physically locate and retrieve the requested data.
- **Relational Diagram -** graphical representation of a relational database's entities, the attributes within those entities, and the relationships among the entities.
- Any SQL-based relational database application involves 3 parts: a user interface, a set of tables stored in the database, and the SQL "engine."
  - *End-user interface* – allows the end user to interact with the data (by automatically generating SQL code)
  - *Collection of tables stored in the database* – Presents the data to the end user in a way that is easy to understand. Each table is independent. Rows in different tables are related by common values in common attributes.
  - *SQL Engine* – executes all queries, or data requests. Hidden from End-User.

○ **2-5c**The Entity Relationship Model

- **entity relationship (ER) model (ERM) –** 1976, A data model that describes relationships (1:1, 1:M, and M:N) among entities at the conceptual level with the help of ER diagrams. The model was developed by Peter Chen.
  - **entity relationship diagram (ERD)**A Graphical diagram that depicts an entity relationship model's entities, attributes, and relations.

- The ER model is based on the following components:
  - Entity = An entity is represented in the ERD by a rectangle with the Entity's name in the Center, also known as an Entity Box. *The entity name is generally written in capital letters and in singular form. Usually, when applying the ERD to the relational model, an entity is mapped to a relational table.*
    - **Entity Instance** or **Entity Occurrence**- A row in a relational table.
    - **Entity Set** -collection of like entities.
    - Each entity consists of a set of ***ATTRIBUTES*** that describes particular characteristics of the entity.
    - **Relationships** - describe associations among data, usually 2 entities. **Connectivity -** The type of relationship between entities. Classifications include 1:1, 1:M, and M:N.
- Different types of relationships using three ER notations:
- **original Chen Notation** – EX: PAINTER paints PAINTING 1:M
- **Crow's Foot Notation** - entity relationship diagram that uses a three-pronged symbol to represent the "many" sides of the relationship.
- **Class Diagram Notation** - The set of symbols used in the creation of class diagrams. Part of the Unified Modeling Language (UML) EX: 1..1 or 1..*

○ **2-5d**The Object-Oriented (OO) Model

- **Object-Oriented Data Model (OODM)** data model, basic modeling structure is an object.
- **Object-**entity that has a unique identity, embedded properties, and the ability to interact with other objects and itself.
- **Object-Oriented Database Management System (OODBMS)**Data management software used to manage data in an object-oriented database model.

✓ an object is described by its factual content. But, quite *unlike* an entity, an object includes information about relationships between the facts within the object, as well as information about its relationships with other objects.

✓ object also to contain all *OPERATIONS:* data values, finding a specific data value, and printing data values.

- **Semantic Data Model** - modeling both data and their relationships in a single structure known as an object.

✓ OO Data Model Components: OBJECTS(equivalent to an ER model's entity.), ATTRIBUTES(describe properties of object), **CLASS**(collection of similar objects with shared structure (attributes) and behavior (methods).[Objects with similar attributes are grouped in these][resembles ER model's *entity* set][Contains **METHODS**, Set of Procedures that Define an Objects *BEHAVIOR*), **CLASS HIERARCHY**(organization of classes in hierarchical tree, each parent class is a *superclass* and each child class is

a *subclass*. See also *inheritance*.) **INHERITANCE**(the ability of an object to inherit the data structure and methods of the classes above it in the class hierarchy), **Unified Modeling Language(UML)**(A language based on object-oriented concepts that provides tools such as diagrams and symbols to graphically model a system, uses **class diagrams**[diagram used to represent data and their relationships in UML object notation])

- **2-5e**Object/Relational and XML
  - **Extended Relational Data Model(ERDM)**A model that includes the object-oriented model's best features in an inherently simpler relational database structural environment. See *extended entity relationship model (EERM)*.
  - **object/relational database management system (O/R DBMS)**A DBMS based on the extended relational model (ERDM).
  - **Extensible Markup Language(XML)**A metalanguage used to represent and manipulate data elements. Unlike other markup languages, XML permits the manipulation of a document's data elements. XML facilitates the exchange of structured documents such as orders and invoices over the Internet. de facto standard for the efficient and effective exchange of structured, semistructured, and unstructured data.
- **2-5f**Emerging Data Models: Big Data and NoSQL

- **Big Data**A movement to find new and better ways to manage large amounts of web-generated data and derive business insight from it, while simultaneously providing high performance and scalability at a reasonable cost.
- **3Vs**, Big Data Characteristics: Volume, Velocity, Variety
  - *Volume*=Amount of Data Stored. *Velocity*=speed which data grows & need to process this data quickly in order to generate information and insight. *Variety*=data being collected comes in multiple different data formats.
- most frequently used Big Data technologies: Hadoop, MapReduce, and NoSQL databases.
- **Hadoop-**A Java based, open source, high speed, fault-tolerant distributed storage and computational framework. Hadoop uses low-cost hardware to create clusters of thousands of computer nodes to store and process data.
  - **Hadoop Distributed File System**A highly distributed, fault-tolerant file storage system designed to manage large amounts of data at high speeds. once the data is written, it cannot be modified. HDFS uses three types of nodes: **NAME NODE** -stores all the metadata about file system, **DATA NODE**-stores fixed-size data blocks (that could be replicated to other data nodes), **CLIENT NODE**-acts as interface between user application and the HDFS.
- **MapReduce-**An open-source application programming interface (API) that provides fast data analytics services; one of the main Big Data technologies that allows organizations to process massive data stores. Map function takes a job and divides it into smaller units of work; the Reduce function collects all the output results generated from the nodes and integrates them into a single result set.
- **NoSQL-**A new generation of database management systems that is not based on the traditional relational database model.
  - NoSQL CHARACTERISTICS: They are not based on the relational model and SQL, hence the name NoSQL. They support distributed database architectures. They provide high scalability, high availability, and fault tolerance.They support very large amounts of sparse data. They are geared toward performance rather than transaction consistency.
- **key-value-**A data model based on a structure composed of two data elements: a key and a value, in which every key has a corresponding value or set of values. The keyvalue data model is also called the **associative or attribute-value data model**.
- **sparse data-**A case in which the number of table attributes is very large but the number of actual data instances is low.
- **eventual consistency-**A model for database consistency in which updates to the database will propagate through the system so that all data copies will be consistent eventually.

- **2-5g**Data Models: A Summary

data models, some common characteristics: Easy to Understand, Representation of the real-world transformations (behavior) must be in compliance with the consistency and integrity characteristics required by the intended use of the data model.

ADVANTAGES AND DISADVANTAGES OF VARIOUS DATABASE MODELS

Data Model	Data Independence	Structural Independence	Advantages	Disadvantages
Hierarchical	Yes	No	It promotes data sharing. Parent/child relationship promotes conceptual simplicity. Database security is provided and enforced by DBMS. Parent/child relationship promotes data integrity. It is efficient with 1:M relationships.	Complex implementation requires knowledge of physical data characteristics. Navigational system yields complex application development, and use; requires knowledge of hierarchical path. Changes in structure require changes in all application progra There are implementation limitations (no multiparent or M:N There is no data definition or data manipulation language in t There is a lack of standards.
Network	Yes	No	Conceptual simplicity is at least equal to that of the hierarchical model. It handles more relationship types, such as M:N and multiparent. Data access is more flexible than in hierarchical and file system models. Data owner/member relationship promotes data integrity. There is conformance to standards. It includes data definition language (DDL) and data manipulation language (DML) in DBMS.	System complexity limits efficiency—still a navigational syster Navigational system yields complex implementation, applicati and management. Structural changes require changes in all application program
Relational	Yes	Yes	Structural independence is promoted by the use of independent tables. Changes in a table's structure do not affect data access or application programs. Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use. Ad hoc query capability is based on SQL. Powerful RDBMS isolates the end user from physical-level details and improves implementation and management simplicity.	The RDBMS requires substantial hardware and system softwa Conceptual simplicity gives relatively untrained people the too system poorly, and if unchecked, it may produce the same da found in file systems. It may promote islands of information problems as individuals departments can easily develop their own applications.
Entity relationship	Yes	Yes	Visual modeling yields exceptional conceptual simplicity. Visual representation makes it an effective communication tool. It is integrated with the dominant relational model.	There is limited constraint representation. There is limited relationship representation. There is no data manipulation language. Loss of information content occurs when attributes are remov to avoid crowded displays. (This limitation has been addresse graphical versions.)
Object-oriented	Yes	Yes	Semantic content is added. Visual representation includes semantic content. Inheritance promotes data integrity.	Slow development of standards caused vendors to supply the enhancements, thus eliminating a widely accepted standard. It is a complex navigational system. There is a steep learning curve. High system overhead slows transactions.
NoSQL	Yes	Yes	High scalability, availability, and fault tolerance are provided. It uses low-cost commodity hardware. It supports Big Data. Key-value model improves storage efficiency.	Complex programming is required. There is no relationship support—only by application code. There is no transaction integrity support. In terms of data consistency, it provides an eventually consis

DATA MODEL BASIC TERMINOLOGY COMPARISON

Real World	Example	File Processing	Hierarchical Model	Network Model	Relational Model	ER Model	OO Model
A group of vendors	Vendor file cabinet	File	Segment type	Record type	Table	Entity set	Class
A single vendor	Global supplies	Record	Segment occurrence	Current record	Row (tuple)	Entity occurrence	Object instance
The contact name	Johnny Ventura	Field	Segment field	Record field	Table attribute	Entity attribute	Object attribute
The vendor identifier	G12987	Index	Sequence field	Record key	Key	Entity identifier	Object identifier

---

❖ **2-6**Degrees of Data Abstraction

- **American National Standards Institute (ANSI)**The group that accepted the DBTG recommendations and augmented database standards in 1975 through its SPARC(Standards Planning and Requirements Committee) committee.
- ANSI/SPARC architecture 3 levels of data abstraction: external, conceptual, and internal.

○ **2-6a**The External Model

- **external model-**The application programmer's view of the data environment. Given its business focus, an external model works with a data subset of the global database schema.
- **external schema-**The specific representation of an external view; the end user's view of the data environment.

○ **2-6b**The Conceptual Model

- **conceptual model-**The output of the conceptual design process. The conceptual model provides a global view of an entire database and describes the main data objects, avoiding details. Also known as **Conceptual Schema**.
- **Software independence-**A property of any model or application that does not depend on the software used to implement it. Doesn't rely on DBMS.
- **Hardware independence-**A condition in which a model does not depend on the hardware used in the model's implementation. Therefore, changes in the hardware will have no effect on the database design at the conceptual level.
- **logical design**A stage in the design phase that matches the conceptual design to the requirements of the selected DBMS and is therefore software dependent. Logical design is used to translate the conceptual design into the internal model for a selected database management system, such as DB2, SQL Server, Oracle, IMS, Informix, Access, or Ingress.

○ **2-6c**The Internal Model

- **internal model-**In database modeling, a level of data abstraction that adapts the conceptual model to a specific DBMS model for implementation. The internal model is the representation of a database as "seen" by the DBMS. In other words, the internal model requires a designer to match the conceptual model's characteristics and constraints to those of the selected implementation model.
- **internal schema**A representation of an internal model using the database constructs supported by the chosen database.
- **logical independence**A condition in which the internal model can be changed without affecting the conceptual model. (The internal model is hardware independent because it is unaffected by the computer on which the software is installed. Therefore, a change in storage devices or operating systems will not affect the internal model.)

- **2-6d** The Physical Model

physical model A model in which physical characteristics such as location, path, and format are described for the data. The physical model is both hardware- and software dependent. See also *physical design*. Lowest Level of Abstraction.

physical independence A condition in which the physical model can be changed without affecting the internal model.

Levels of Data Abstraction

Model	Degree of Abstraction	Focus	Independent of
External		End-user views	Hardware and software
Conceptual		Global view of data (database model independent)	Hardware and software
Internal		Specific database model	Hardware
Physical		Storage and access methods	Neither hardware nor software

Summary

- A data model is an abstraction of a complex real-world data environment. Database designers use data models to communicate with programmers and end users. The basic data-modeling components are entities, attributes, relationships, and constraints. Business rules are used to identify and define the basic modeling components within a specific real-world environment.
- The hierarchical and network data models were early models that are no longer used, but some of the concepts are found in current data models.
- The relational model is the current database implementation standard. In the relational model, the end user perceives the data as being stored in tables. Tables are related to each other by means of common values in common attributes. The entity relationship (ER) model is a popular graphical tool for data modeling that complements the relational model. The ER model allows database designers to visually present different views of the data—as seen by database designers, programmers, and end users—and to integrate the data into a common framework.
- The object-oriented data model (OODM) uses objects as the basic modeling structure. Like the relational model's entity, an object is described by its factual content. Unlike an entity, however, the object also includes information about relationships between the facts, as well as relationships with other objects, thus giving its data more meaning.
- The relational model has adopted many object-oriented (OO) extensions to become the extended relational data model (ERDM). Object/relational database management systems (O/R DBMS) were developed to implement the ERDM. At this point, the OODM is largely used in specialized engineering and scientific applications, while the ERDM is primarily geared to business applications.
- Emerging Big Data technologies such as Hadoop, MapReduce, and NoSQL provide distributed, fault-tolerant, and cost-efficient support for Big Data analytics. NoSQL databases are a new generation of databases that do not use the relational model and are geared to support the very specific needs of Big Data organizations. NoSQL databases offer distributed data stores that

provide high scalability, availability, and fault tolerance by sacrificing data consistency and shifting the burden of maintaining relationships and data integrity to the program code.

- Data-modeling requirements are a function of different data views (global versus local) and the level of data abstraction. The American National Standards Institute Standards Planning and Requirements Committee (ANSI/SPARC) describes three levels of data abstraction: external, conceptual, and internal. The fourth and lowest level of data abstraction, called the physical level, is concerned exclusively with physical storage methods.