TRINITY COLLEGE FOR WOMEN NAMAKKAL

Department of Computer Science

ADVANCED DATABASE MANAGEMENT SYSTEM

21PCS -EVEN Semester

Presented by

Mrs.S.USHARANI

Assistant Professor

Department of Computer Science

http://www.trinitycollegenkl.edu.in/

Basic Definitions

- **Database:** A collection of related data.
- **Data**: Known facts that can be recorded and have an implicit meaning.
- **Mini-world**: Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS)**: A software package/ system to facilitate the creation and maintenance of a computerized database.
- **Database System**: The DBMS software together with the data itself. Sometimes, the applications are also included.

Main Characteristics of the Database Approach

Data Abstraction

A **data model** is used to hide storage details and present the users with a conceptual view of the database.

Support of multiple views of the data

Each user may see a different view of the database, which describes only the data of interest to that user.

Sharing of data and multiuser transaction processing **A**llowing a set of concurrent users to retrieve and to update the database. Concurrency control within the DBMS guarantees that each transaction is correctly executed or completely aborted. OLTP (Online Transaction Processing) is a major part of database applications.

Types of Databases and Database Applications

- Numeric and Textual Databases
- Multimedia Databases
- Geographic Information Systems (GIS)
- Data Warehouses
- Real-time and Active Databases

Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenence efforts.
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects (in Object-oriented DBMS)
- Providing Storage Structures for efficient Query Processing
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.

Data Models

Data Model:

A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

Data Model Operations:

Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.

Categories of data models

Conceptual (high-level, semantic) data models:

Provide concepts that are close to the way many users perceive data. (Also called **entity-based** or **object-based** data models.)

Physical (low-level, internal) data models:

Provide concepts that describe details of how data is stored in the computer.

Implementation (representational) data models:

Provide concepts that fall between the above two, balancing user views with some computer storage details.

History of Data Models

Relational Model: Proposed in 1970 by E.F. Codd (IBM), first commercial system in 1981-82. Now in several commercial products (DB2, ORACLE, SQL Server, SYBASE, INFORMIX).

Network Model: The first one to be implemented by Honeywell in 1964-65 (IDS System).Adopted heavily due to the support by CODASYL (CODASYL - DBTG report of 1971). Later implemented in a large variety of systems - IDMS (Cullinet - now CA), DMS 1100 (Unisys), IMAGE (H.P.), VAX -DBMS (Digital Equipment Corp.).

Hierarchical Data Model: Implemented in a joint effort by IBM and North American Rockwell around 1965. Resulted in the IMS family of systems. The most popular model. Other system based on this model: System 2k (SAS inc.)

Object-oriented Data Model(s): Several models have been proposed for implementing in a database system.One set comprises models of persistent O-O Programming Languages such as C++ (e.g., in OBJECTSTORE or VERSANT), and Smalltalk (e.g., in GEMSTONE).

Object-Relational Models: Most Recent Trend. Started with Informix Universal Server. Exemplified in the latest versions of Oracle-10i, DB2, and SQL Server etc. systems.

Schemas versus Instances

Database Schema:

- The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.

Schema Diagram:

- A diagrammatic display of (some aspects of) a database schema.

Schema Construct:

- A component of the schema or an object within the schema, e.g., STUDENT, COURSE.

Database Instance:

- The actual data stored in a database at a particular moment in time. Also called **database state** (or **occurrence**).

DBMS Languages

- **Data Definition Language (DDL)**: Used by the DBA and database designers to specify the *conceptual schema* of a database. In many DBMSs, the DDL is also used to define internal and external schemas (views). In some DBMSs, separate **storage definition language (SDL)** and **view definition language (VDL)** are used to define internal and external schemas.
- **Data Manipulation Language (DML)**: Used to specify database retrievals and updates.
  - DML commands (**data sublanguage**) can be *embedded* in a general-purpose programming language (**host language**), such as COBOL, C or an Assembly Language.
  - Alternatively, *stand-alone* DML commands can be applied directly (**query language**).
- **High Level or Non-procedural Languages:** e.g., SQL, are *set-oriented* and specify what data to retrieve than how to retrieve. Also called *declarative* languages.
- **Low Level** or **Procedural Languages:** record-at-a-time; they specify *how* to retrieve data and include constructs such as looping.

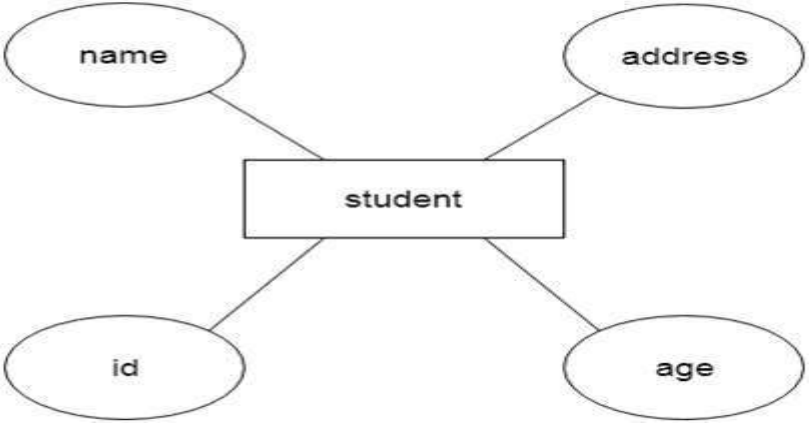
Database System Utilities

- To perform certain functions such as:
  - Loading data stored in files into a database. Includes data conversion tools.
  - Backing up the database periodically on tape.
  - Reorganizing database file structures.
  - Report generation utilities.
  - Performance monitoring utilities.
  - Other functions, such as sorting, user monitoring, data compression, etc.

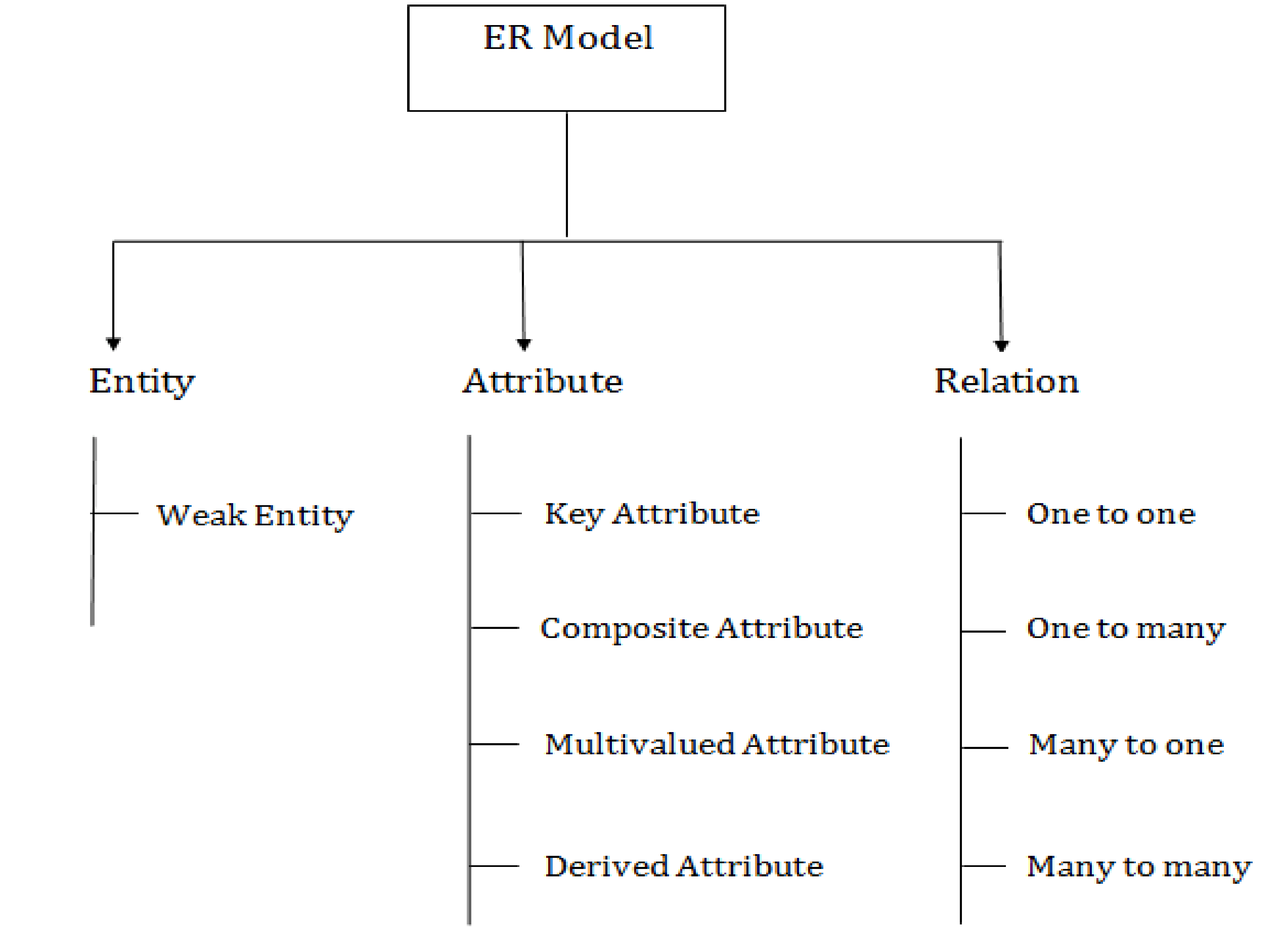
ER MODEL Concepts

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.It develops a conceptual design for the database.
- It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

Example, Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.



Component of ER Diagram



ER Model

Entity

- Weak Entity

Attribute

- Key Attribute
- Composite Attribute
- Multivalued Attribute
- Derived Attribute

Relation

- One to one
- One to many
- Many to one
- Many to many

Relational Integrity Constraints

Constraints are conditions that must hold on all valid relation instances. There are three main types of constraints:

1. **Key** constraints
2. **Entity integrity** constraints
3. **Referential integrity** constraints

Key Constraints

Superkey of R: A set of attributes SK of R such that no two tuples in any valid relation instance r(R) will have the same value for SK. That is, for any distinct tuples t1 and t2 in r(R), t1[SK] □ t2[SK].

Key of R: A "minimal" superkey; that is, a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey.

Example: The CAR relation schema:

CAR(State, Reg#, SerialNo, Make, Model, Year)
has two keys Key1 = {State, Reg#}, Key2 = {SerialNo}, which are also superkeys. {SerialNo, Make} is a superkey but not a key.

If a relation has several **candidate keys**, one is chosen arbitrarily to be the **primary key**. The primary key attributes are underlined.

THANK YOU

http://www.trinitycollegenkl.edu.in/