Lecture 1.2. Types of database models. Advantages and disadvantages.

Hierarchical Database Model

Definition:

> Data stored as records in a tree-like structure.
> Parent–child structure.
> One parent node can have many child nodes (1:N).
> Links explicitly show relationships.

Think of it as a root with branches; each node can have multiple children, but usually one parent.

Advantages:

- Easy data retrieval due to explicit links.
- Referential integrity automatically maintained:
  - Changes in parent propagate to children.
- Promotes data sharing.
- Conceptually simple thanks to parent–child relationship.
- Database security enforced.
- Efficient for 1:N relationships.
- Clear chain of command/authority.
- Increases specialization.
- High performance, especially with simple hierarchical queries.
- Clear results.

Disadvantages:

- If parent and child tables are unrelated, adding a new child entry is hard (you must add parent entry too).
- Complex relationships not supported well.
- Redundancy → may cause inaccurate information.
- Changing the structure forces changes in all application programs.
- M:N relationships not supported.
- No proper data manipulation language or data definition language (historical context).
- Lack of standards.
- Poor flexibility.
- Organizational disunity and rigid structure.

Network Database Model

Definition:

> Developed as progression from hierarchical model.
> Solves lack of flexibility:
>   Each child can have multiple parents.
> Data organized in sets and connections forming a graph, not just a tree.

Advantages:

- Fast data access.
- Allows more complex queries than hierarchical.
- Supports a variety of queries.

Disadvantages:

- User must be very familiar with database structure to navigate.
- Updating is tedious.

- Changing a set structure affects all application programs that use it.
- If you change set structure, you must update all references in code.

Relational Database Model

Definition:

> Organizes data into one or more tables (“relations”).
> Each row is uniquely identified by a key.
> Rows = records / tuples.
> Columns = attributes.

Advantages:

- Ease of use
- Language support (SQL)
- Network access (multi-user)
- Often good performance
- Prevents data redundancy
- Privileges and data security (GRANT/REVOKE, roles, etc.)

Disadvantages:

- Cost (commercial systems like Oracle, SQL Server can be expensive).
- Possible lack of speed for certain tasks (compared to specialized models).
- Uses memory space (overhead for metadata, indexes, logs, etc.).

Object-Oriented Database Model (OOD)

Definition:

> Database system that works with complex data objects.
> Objects mirror those in OOP languages.
> In OOP: “everything is an object”—OOD follows same philosophy.

Elements of Object-Oriented Data Model:

- Object
- Attributes and Methods
- Class
- Inheritance

Advantages:

- Reusability:
  - Generic objects can be defined and reused in many applications.
- Handles complex data types:
  - Documents, graphics, images, voice messages, etc.
- Supports distributed databases more easily due to communication between objects.

Disadvantages:

- Competition with relational model (market).
- Complexity.
- Lack of support for views.
- Lack of support for security (relative to mature RDBMS).
- Lack of standards.
- Lack of experience (fewer experts).

Entity-Relationship (ER) Model

Definition:

> High-level conceptual data model diagram.
> Helps analyze data requirements systematically to produce a well-designed database.
> Represents real-world entities and relationships between them.

ERD displays the relationships of entity sets stored in a database.
Explains the logical structure of the database.

Why use ER diagrams?

- Help define terms related to entity-relationship modeling.
- Provide a preview of:
    - How tables should connect.
    - What fields are needed.
- Help describe:
    - Entities
    - Attributes
    - Relationships
- ER diagrams can be translated into relational tables → allows quick database building.
- ERD helps communicate logical structure of the database to users.

Entity

Definition:

> A real-world thing (living or non-living), recognizable or conceptual.
> Can be: Place, person, object, event, or concept.
> An entity stores data in the database.

In diagrams, entities are usually shown as rectangles.

Attributes

Entities are described by their properties = attributes.

In ERD, attributes are typically shown as ovals connected to the entity.

Relationships

1. One-to-many (1:M)
    a. One department – many students.
2. Many-to-many (M:N)
    a. Students and courses (one student takes many courses; one course has many students).
3. One-to-one (1:1)
    a. One person – one passport.

Typical exam-style questions

MCQ: “Which model organizes data as parent–child tree structure?”
MCQ: “Which model allows a child to have multiple parents?”
MCQ: “In relational model, rows are also called: ...”
Open: “List two advantages and two disadvantages of hierarchical databases.”
Open: “Explain why object-oriented databases are useful for complex data such as images and documents.”

MCQ: “Which of the following is NOT an example of an entity?”
MCQ: “A relationship where each instance of A can be associated with multiple instances of B and vice versa is called: ...”
Open: “Define entity, attribute, and relationship, and give one example for each.”