جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

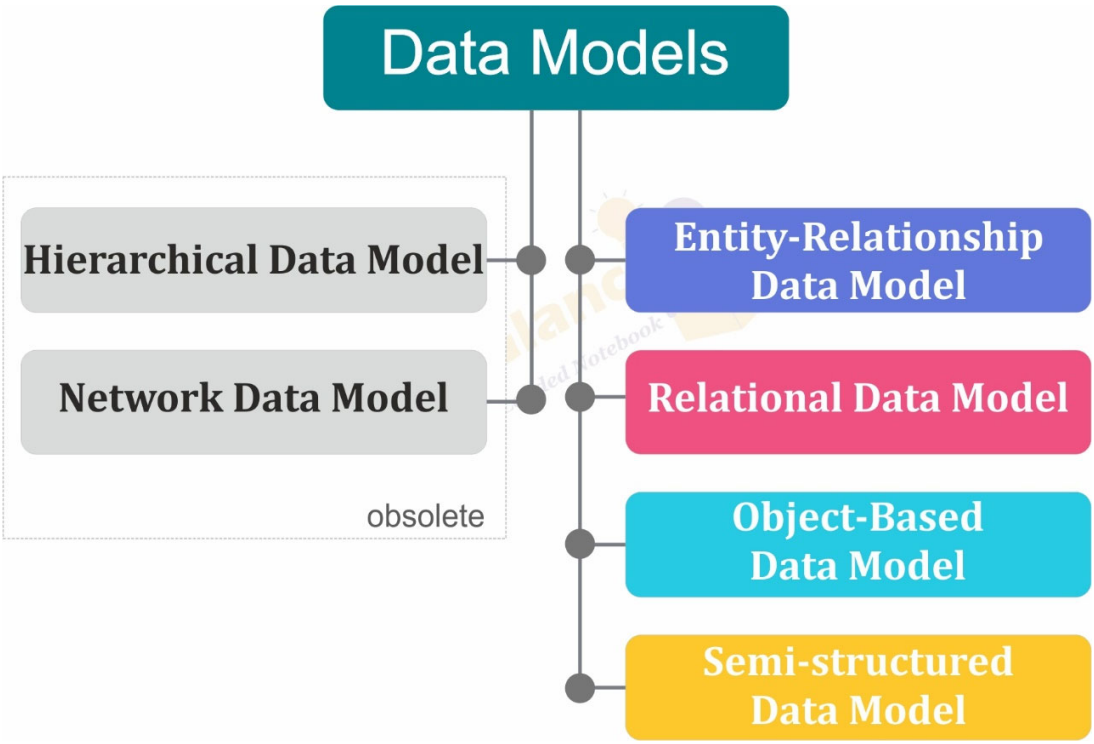
قسم الامن السيبراني

Lecture: (3)

Subject: Database Systems
Level: Second
Lecturer: Asst. Lecturer Qusai AL-Durrah

Data Models Type

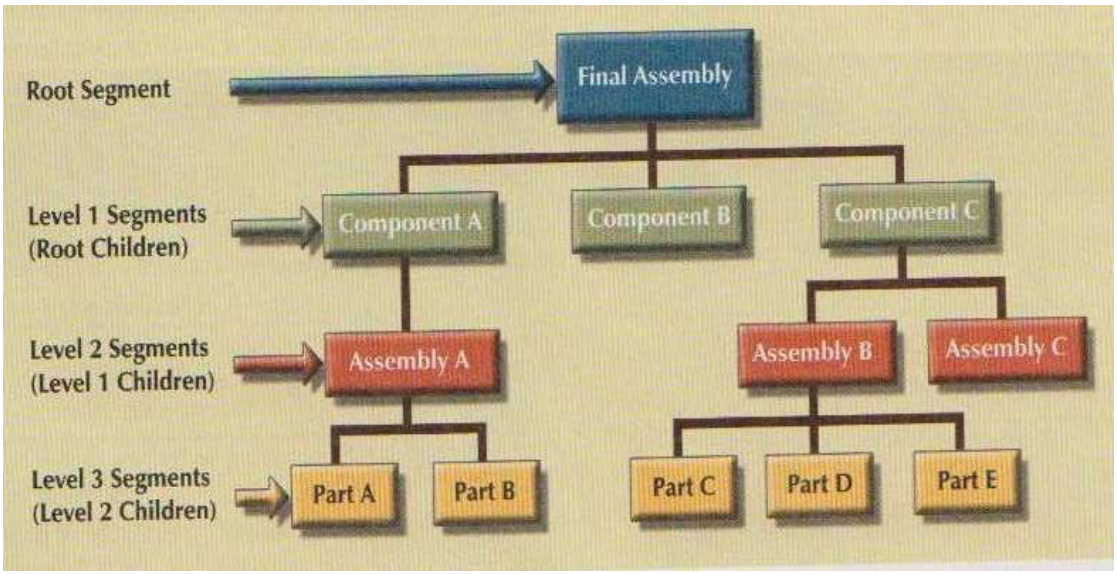
The quest for better data management has led to several different models that attempt resolve the file systems critical shortcomings.



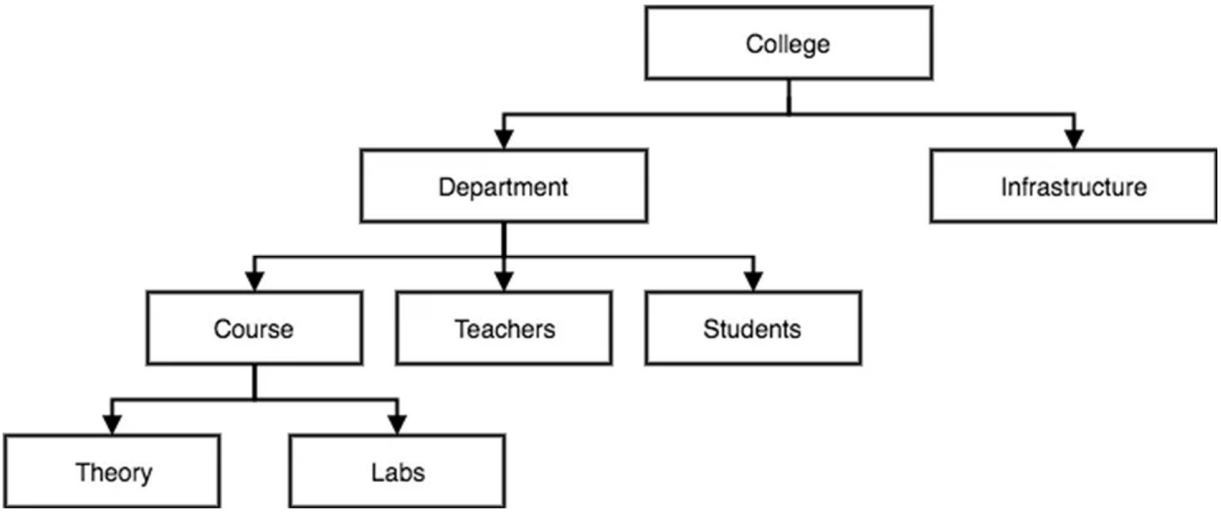
THE HIERARCHICAL MODEL

It was developed in 1960s to manage large amounts of data for complex manufacturing projects Apollo rocket that landed on the moon in 1969. its basic logical structure is represented by an upside-down tree- its structure contains levels, or segments. Segment is the equivalent of a file systems record type. The top layer (the root) is perceived as the parent of the segment (each parent can have many children, but each child has only one parent)

The hierarchical model had limitations: it was complex to implement, it was difficult to manage, and it lacked structural independence. Also, many common data relationships do not conform to the 1: M from, and there were no standards for how to implement the model.



In a hierarchical data model, data is organized in a tree-like structure with one **one-to-many relationship** between two different types of data. For example- one department can have many courses, teachers and students

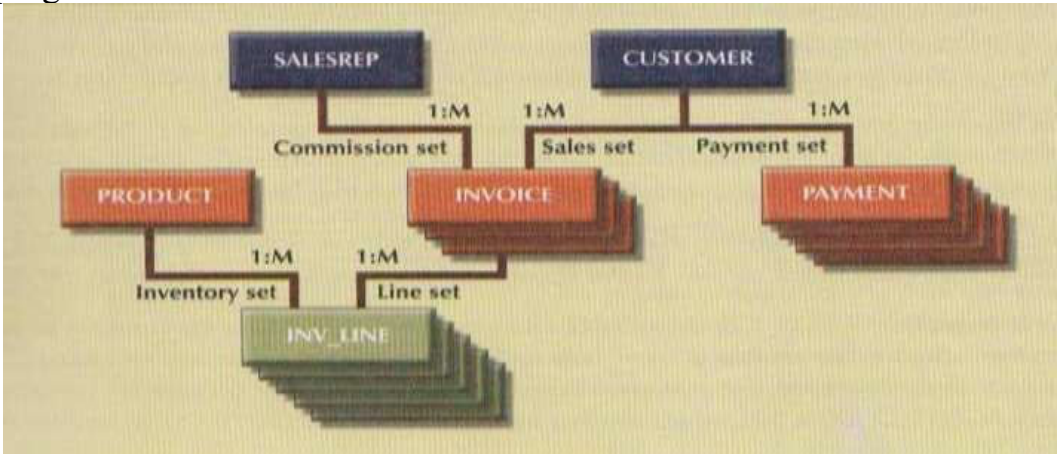


The Network Model

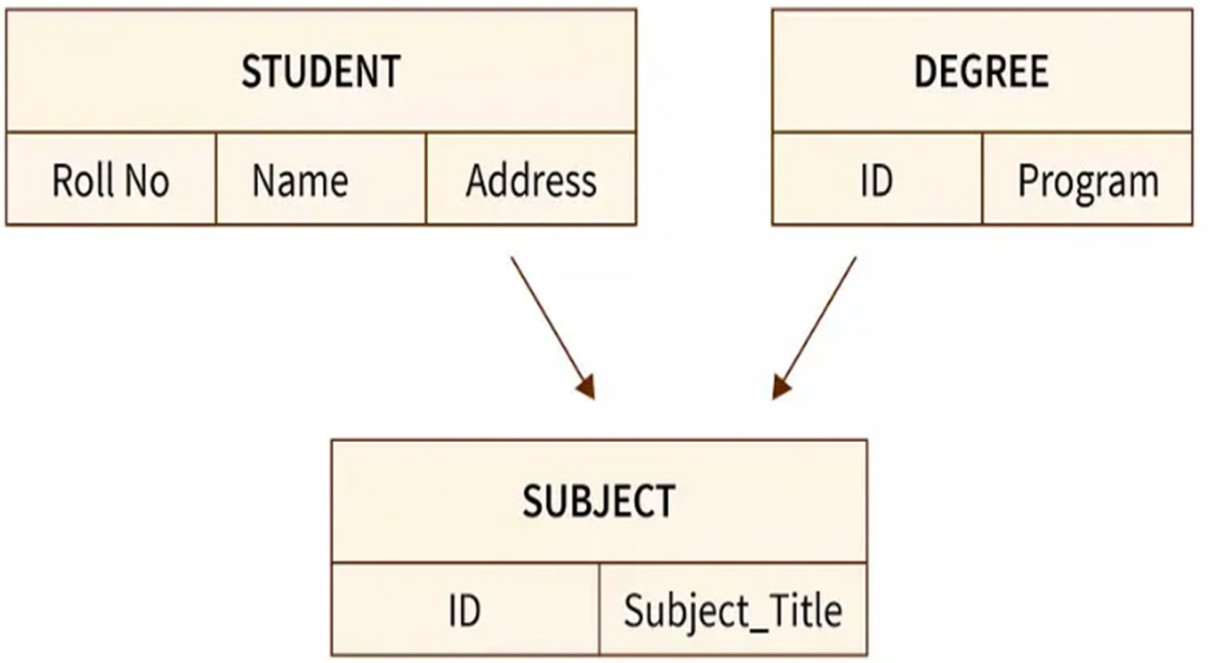
The network model (1970) was created to represent complex data relationships more effectively than the hierarchical model, in the network model, the user perceives the network database as a collection of records in M:N

relationship. However, unlike the hierarchical model, the network model allows a record to have more than one parent. In network db terminology, relationship is called a set. Each set is composed of at least two record types: an owner record and a member record. a set represents a 1:M relationship between the owner and the member.

Disadvantages is the lack of ad hoc query capability put heavy pressure on programmers to generate the code required to produce even the simplest report, and although the existing databases provided limited data independence, any structural change in the database still could produce have in all application programs that drew data from the database.



Suppose we are designing the network model for the Students database. As we can see that the Subject entity has a relationship with both the Student entity and Degree entity. So there is an edge connecting the Subject entity with both Student and Degree.
The Subject entity has two parents and the other two entities have one child entity.



The Relational Model (1970)

The relational model for database management is an approach to logically represent and manage the data stored in a database. In this model, the data is organized into a collection of two-dimensional inter-related tables, also known as relations. Each relation is a collection of columns and rows, where the column represents the attributes of an entity and the rows (or tuples) represents the records.

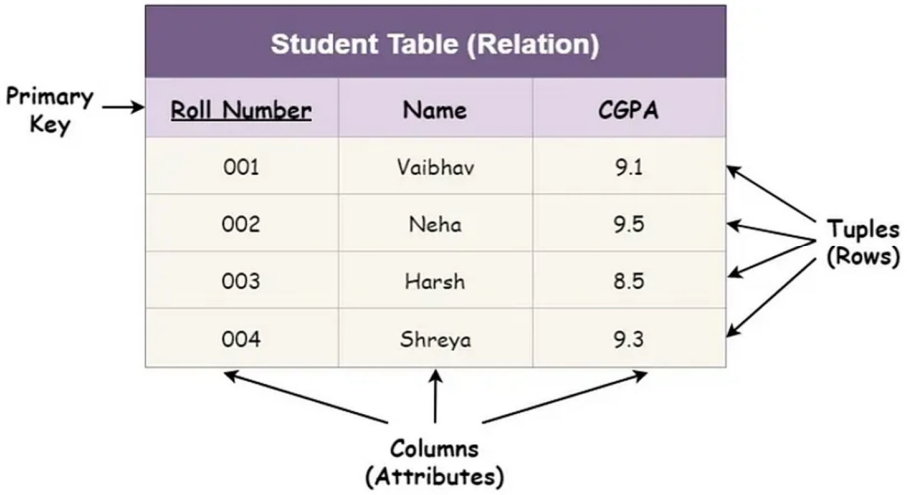
The relational model consists of:

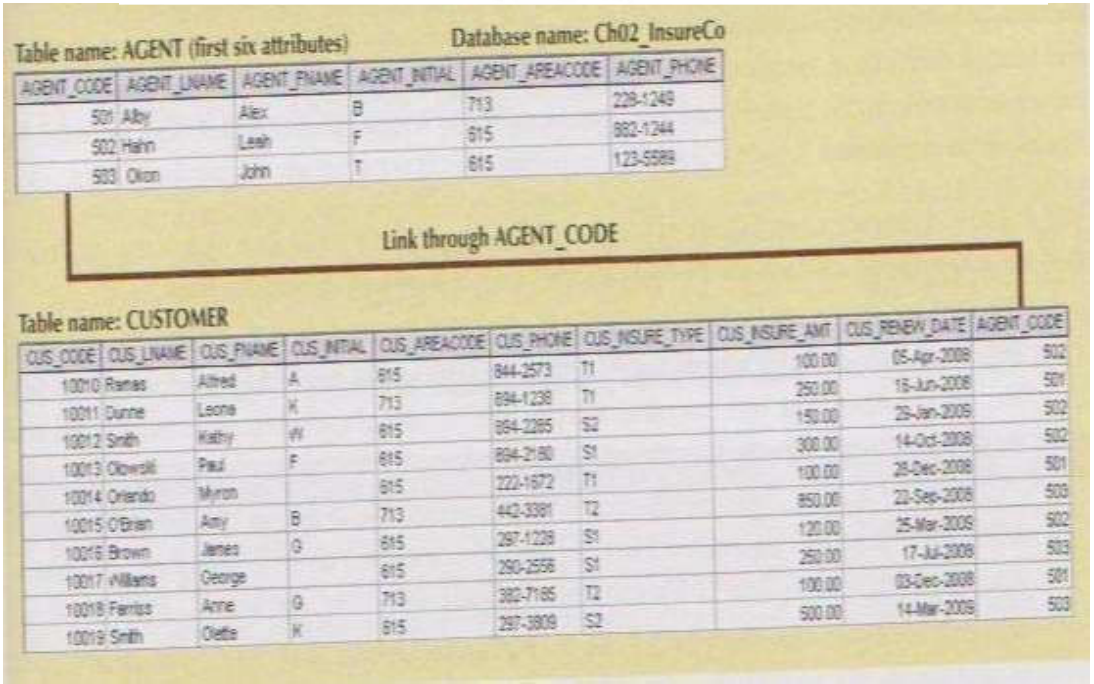
- Relation: table as a matrix.
- Tuple: raw in relation.
- Attribute: column in relation.

Advantages

- Tables are related to each other through the sharing of common attribute.

- The tables are independent of another;
- Provides a minimum level of controlled redundancy.
- The relationship type(1:1, 1:M or M:N) is often show in relational model.
- The relational model is powerful and flexible in query language.





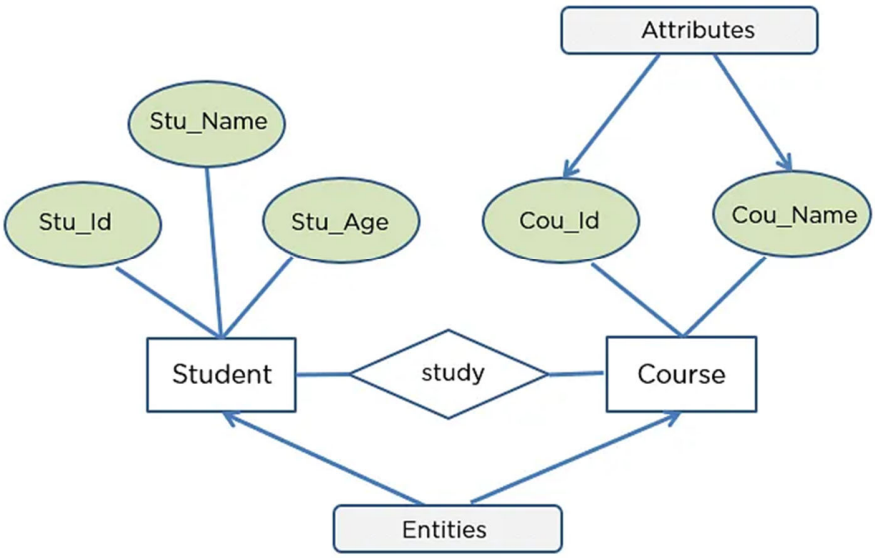
The Entity-Relationship Model

Peter Chen first introduced the ER data model in 1976: it was the graphical representation of entities and their relationships in a database structure the quickly become popular because it complemented the relational data model complex.

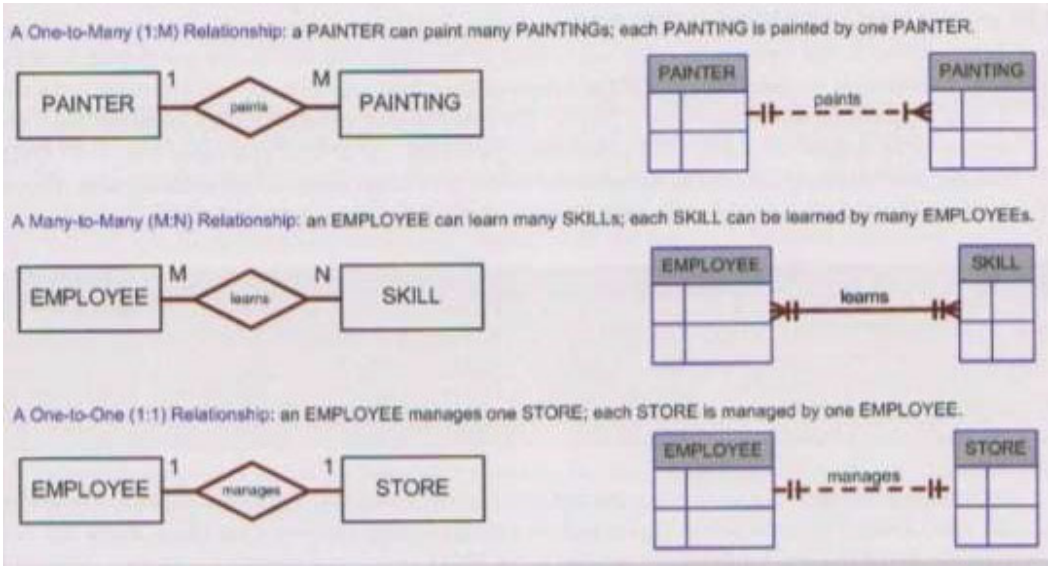
An Entity Relationship Diagram (ER Diagram) pictorially explains the relationship between entities to be stored in a database.

- entity is represented by rectangle

- Relationships: describe associations among data. The ER model uses the term connectivity it may be 1:1 or 1:M or M:N



The diagram showcases two entities — Student and Course, and their relationship. The relationship described between student and course is **many-to-many**, as a course can be opted by several students, and a student can opt for more than one course. Student entity possesses attributes — Stu_Id, Stu_Name & Stu_Age. The course entity has attributes such as Cou_ID & Cou_Name.

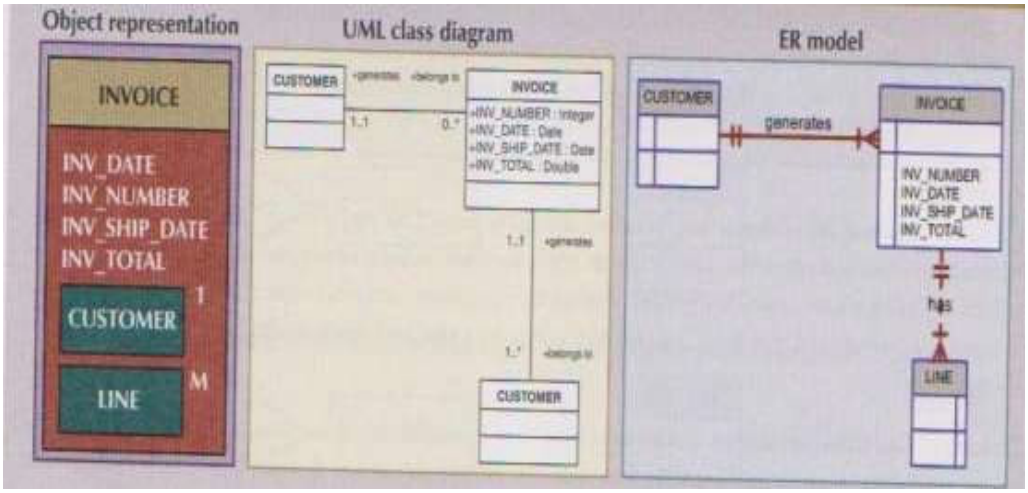


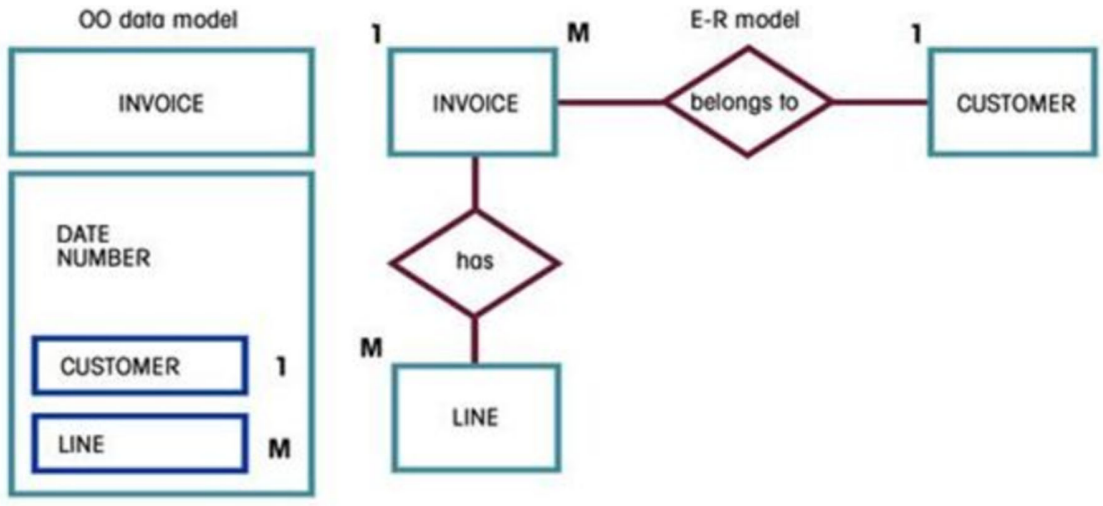
The Object-oriented (OO) Model

Increasingly complex real-world problems demonstrated a need for a data model that more closely represented the real world. In the object-oriented data model (OODM), both data and their relationships are contained in a single structure known as an object. The OO data model is based on the following components:

- Object: may be considered equivalent to an ER models entity. The objects semantic content is defined through several of the items in this list
- Attributes: describe the properties of an object.
- Class is a collection of similar objects with shared structure (attributes) and behavior(methods).
- Classes are organized in a class hierarchy are upside-down tree in which each class has only one parent.

- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of the classes above it.



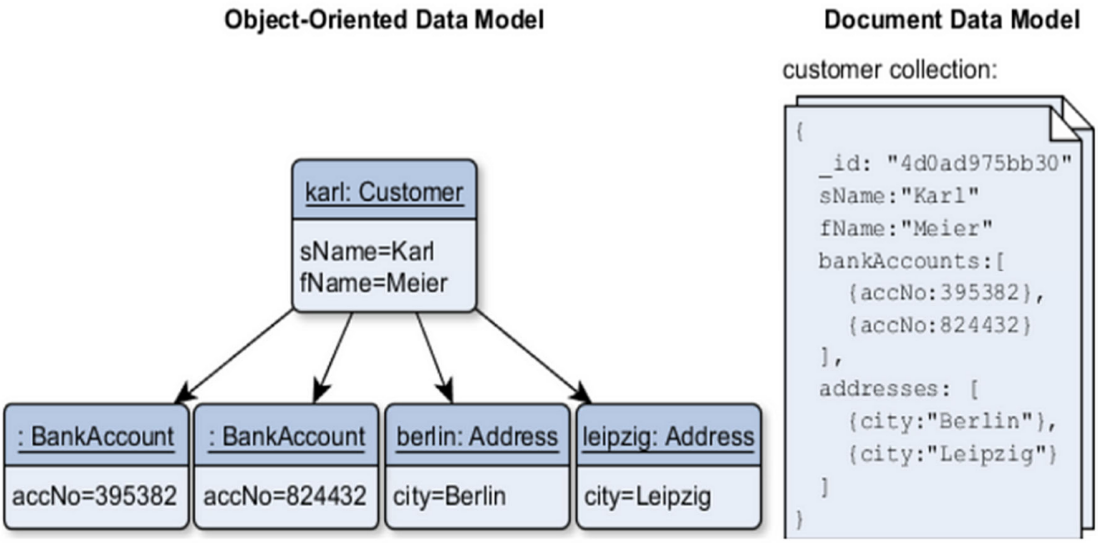


A comparison of OO data model and ER model

Document Model:

Documents are the most common way for storing, retrieving, and managing semi-structured data. Unlike the traditional relational data model, the document data model is not restricted to a rigid schema of rows and columns.

- **Description:** Stores data in flexible, semi-structured documents (e.g., JSON, XML) rather than rigidly structured tables.



Document data models are best fit for use cases requiring a flexible schema and fast data access. E.g. nested documents enable applications to store related pieces of information in the same database record in a denormalized manner. As a result, applications can issue fewer queries and updates to complete

common operations.

The development of data models

Each of these models serves different purposes and fits different types of data and relationships within a database system, catering to various needs from structured data to complex relationships and flexibility in data representation.