Lecturer: Sholpan Saimassayeva
s.saimassayeva@kbtu.kz

Lecture 1: Introduction to Databases

Introduction to Databases

PART 1	What is a database? Properties, Characteristics, Database Management System, Application, Example.
PART 2	Types of database models. Advantages and disadvantages.
PART 3	ER Diagram. Entity, Attributes, Relationships.

PART I. What is a database? Properties, Characteristics, Database Management System, Application, Example.

What is a database?

A **database** is a collection of information that is organized so that it can easily be accessed, managed, and updated.

Properties of a database:

- A **database** represents some aspect of the real world.
- A **database** is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A **database** is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

What is a database management system?

A **database management system** (DBMS) is system software for creating and managing databases.

A DBMS makes it possible for end users to create, protect, read, update and delete data in a database.

Database-System Applications

Real-World Applications
Banking: Storing customer records, transactions, loans.
Airlines: Managing reservations, schedules, and passenger details.
Universities: Student records, course registrations, and results.
Telecommunications: Call records, billing, and customer service.
Key Takeaway: Databases are essential in modern information systems across industries.

Example

- The **STUDENT** file stores data of each student
- The **COURSE** file stores contain data on each course
- The **SECTION** stores the information about sections in a particular course
- The **GRADE_REPORT** file stores the grades which students receive in the various sections
- The **PREREQUISITE** file contains information about pre-courses

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Users in a DBMS environment

Component Name	Task
Application Programmers	The Application programmers write programs in various programming languages to interact with databases.
Database Administrators	Database Admin is responsible for managing the entire DBMS system. He/She is called Database admin or DBA.
End-Users	The end users are the people who interact with the database management system. They conduct various operations on database like retrieving, updating, deleting, etc.

Purpose of Database Systems

Data Management:

Efficiently store, retrieve, and manage large amounts of data.

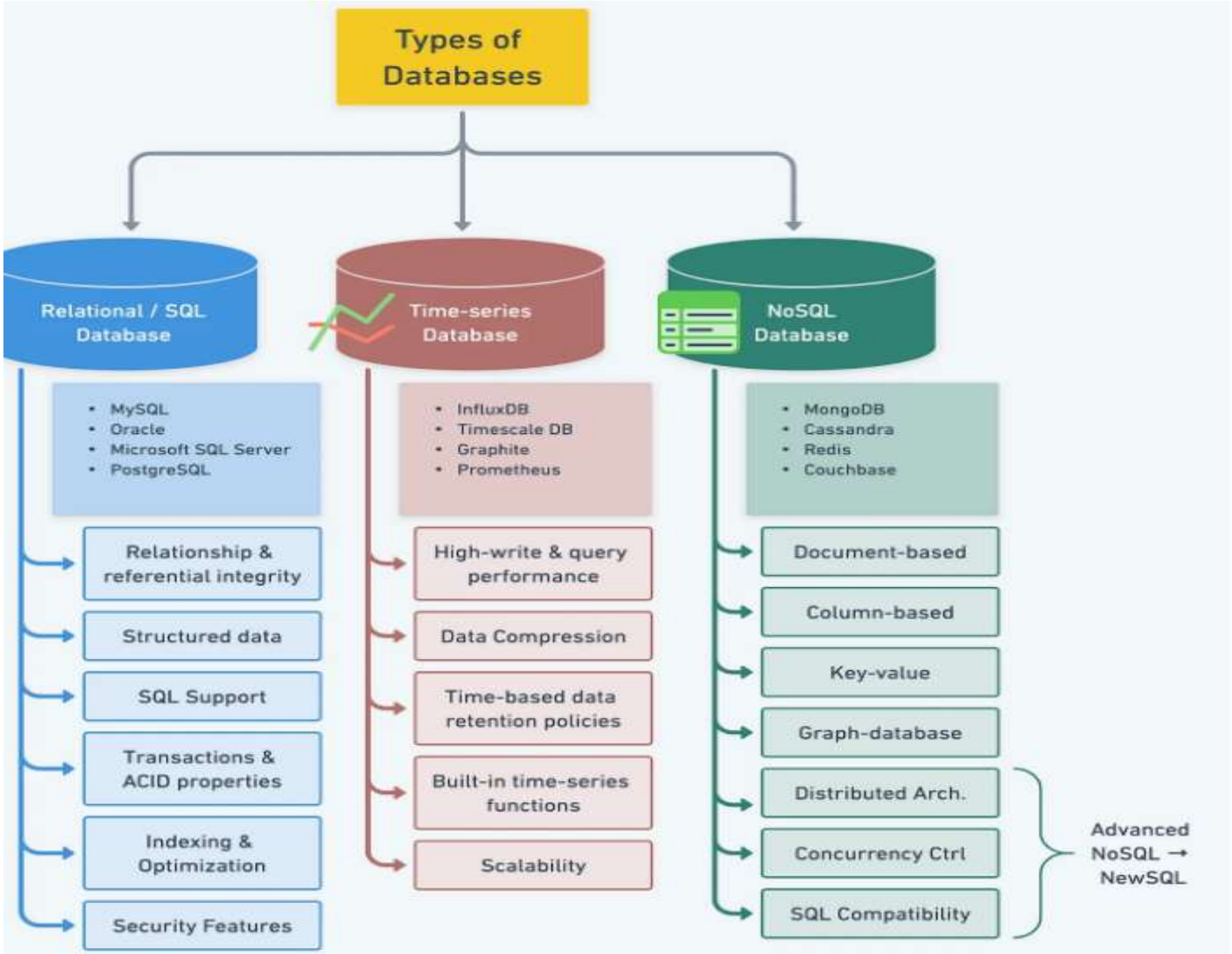
Ensure data integrity and security.

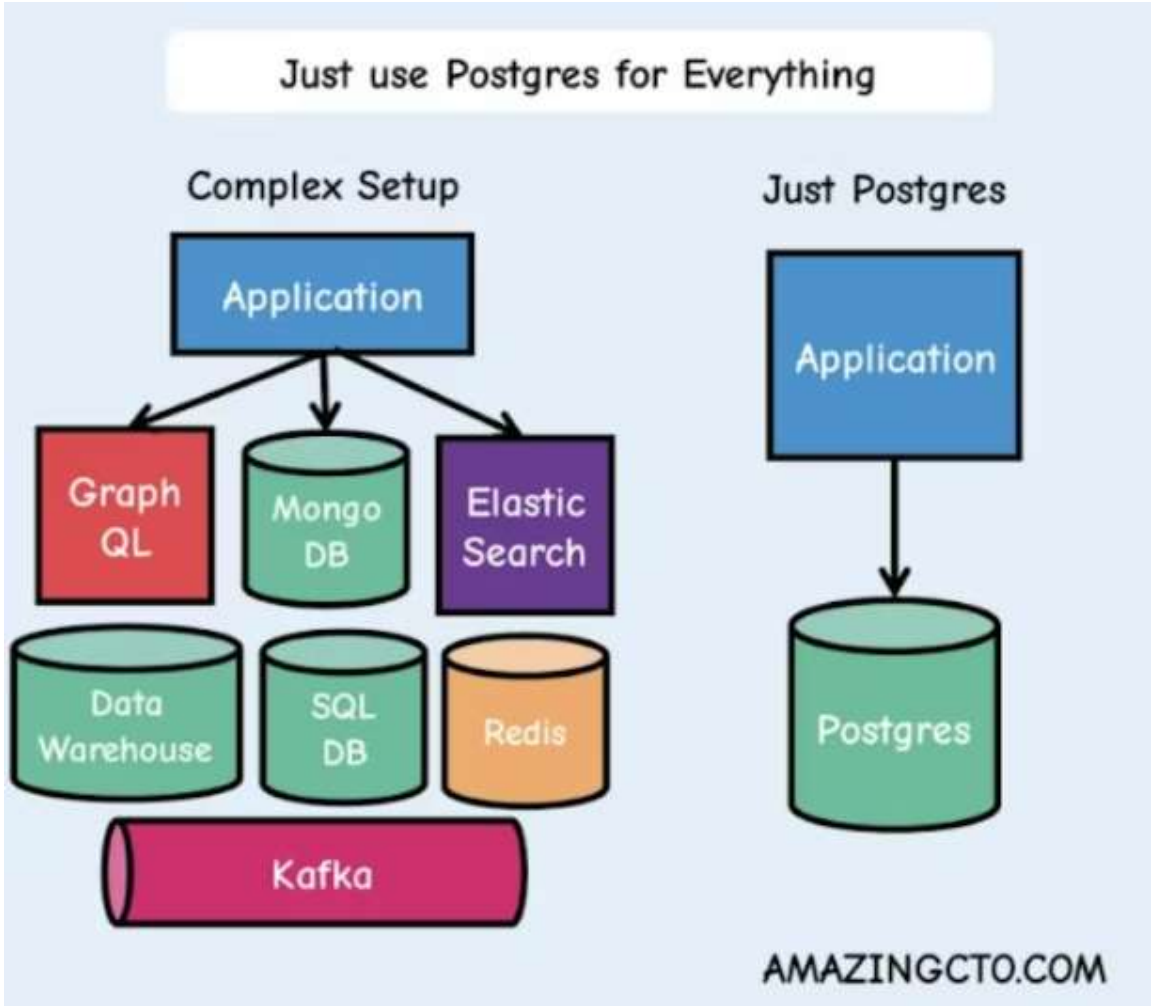
Data Abstraction:

Abstracts the complexity of data storage, making it easier to interact with.

Data Redundancy and Inconsistency:

Reduce data redundancy and ensure consistency through normalization.


Types of Databases
Relational / SQL Database
Time-series Database
NoSQL Database
MySQL
Oracle
Microsoft SQL Server
PostgreSQL
InfluxDB
Timescale DB
Graphite
Prometheus
MongoDB
Cassandra
Redis
Couchbase
Relationship & referential integrity
Structured data
SQL Support
Transactions & ACID properties
Indexing & Optimization
Security Features
High-write & query performance
Data Compression
Time-based data retention policies
Built-in time-series functions
Scalability
Document-based
Column-based
Key-value
Graph-database
Distributed Arch.
Concurrency Ctrl
SQL Compatibility
Advanced NoSQL → NewSQL



Toolkit

Popular relational database management systems:

- Oracle
- SQL Server
- MySQL
- **PostgreSQL**
- SQLite

What is PostgreSQL?

- **PostgreSQL** is a powerful, **open source** object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.
- **PostgreSQL** is highly extensible. For example, you can define your own data types, build out custom functions, even write code from **different programming languages** without recompiling your database!

Useful links

- Link to download the installer:

https://www.postgresql.org/download/

- PostgreSQL Documentation:

https://www.postgresql.org/docs/current/

PART II. Types of database models. Advantages and disadvantages.

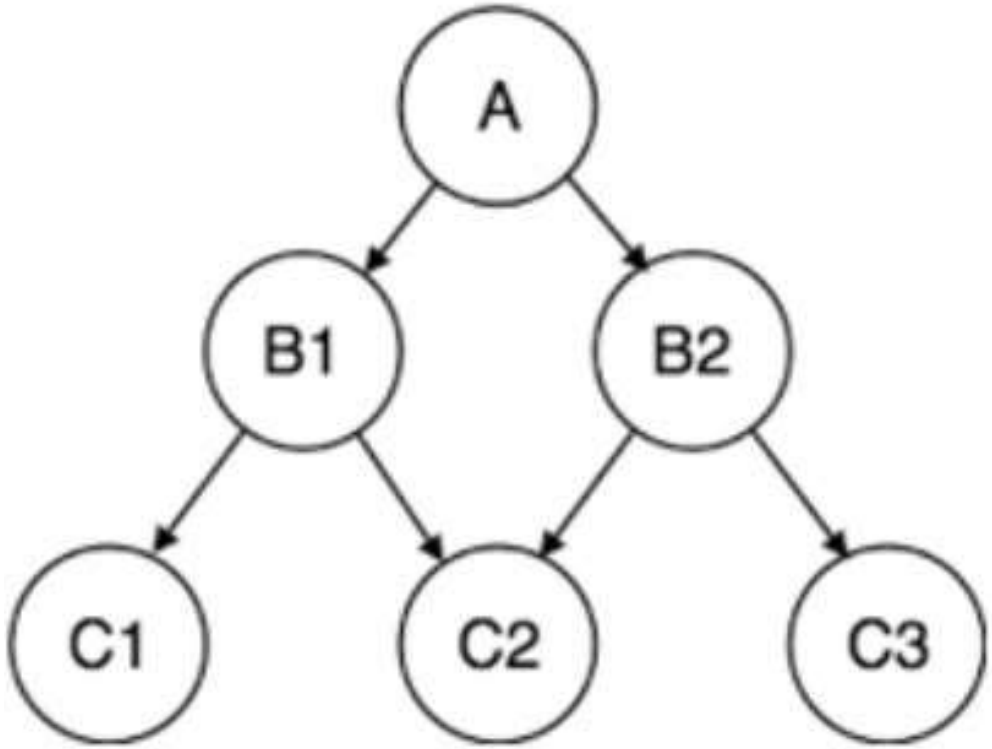
Types of database models:

1. Hierarchical Database
2. Network Database
3. Relational Database
4. Object-Oriented Model

Type 1. Hierarchical Database

- A **hierarchical database** is a data model in which data is stored in the form of records and organized into a tree-like structure, or parent-child structure, in which one parent node can have many child nodes connected through links.

Example of hierarchical database



Advantages of the Hierarchical DBMS :

- Data can be retrieved easily due to the explicit links present between the table structures.
- Referential integrity is always maintained i.e. any changes made in the parent table are automatically updated in a child table.
- Promotes data sharing.
- It is conceptually simple due to the parent-child relationship.
- Database security is enforced.
- Efficient with 1: N relationships.
- A clear chain of command or authority.
- Increases specialization.
- High performance.
- Clear results.

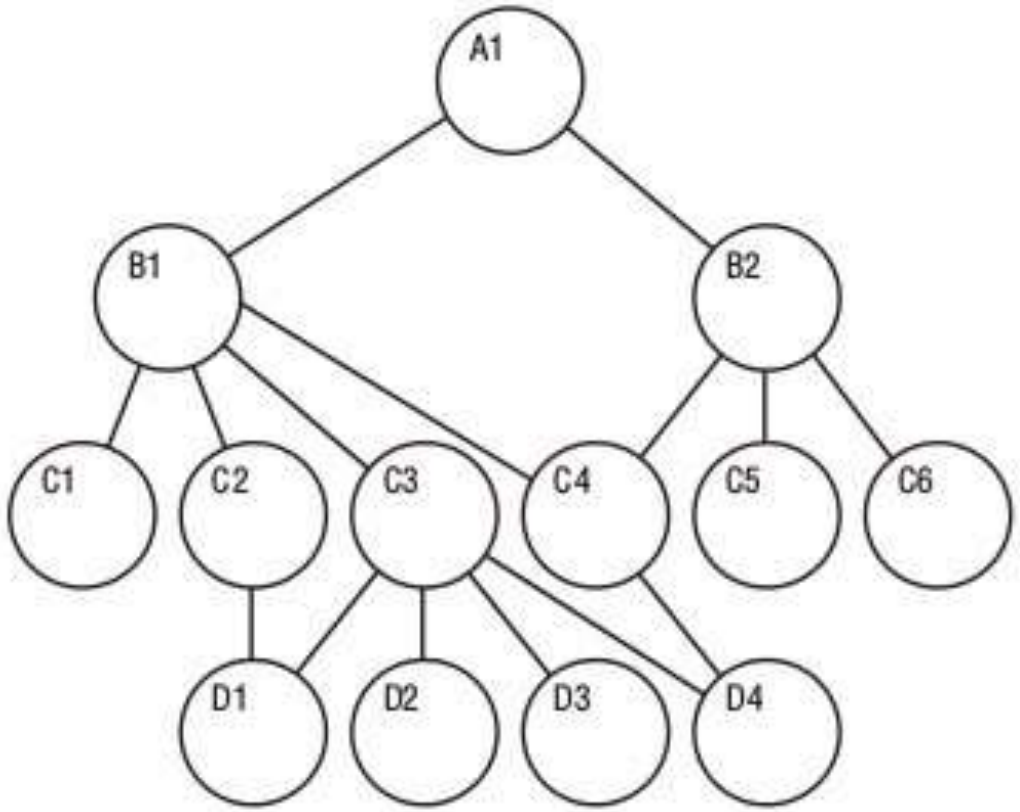
Disadvantages of the Hierarchical DBMS :

- If the parent table and child table are unrelated then adding a new entry in the child table is difficult because additional entry must be added in the parent table.
- Complex relationships are not supported.
- Redundancy which results in inaccurate information.
- Change in structure leads to change in all application programs.
- M: N relationship is not supported.
- No data manipulation or data definition language.
- Lack of standards.
- Poor flexibility
- Organizational Disunity and Rigid structure.

Type 2. Network database

The **network database model** was a progression from the hierarchical database model and was designed to solve some of that model's problems, specifically the lack of flexibility. Instead of only allowing each child to have one parent, this model allows each child to have multiple parents

Example of Network DBMS:



Advantages of the Network DBMS:

- fast data access.
- It also allows users to create queries that are more complex than those they created using a hierarchical database. So, a variety of queries can be run over this model.

Disadvantages of the Network DBMS:

- A user must be very familiar with the structure of the database to work through the set structures.
- Updating inside this database is a tedious task. One cannot change a set structure without affecting the application programs that use this structure to navigate through the data. If you change a set structure, you must also modify all references made from within the application program to that structure.

Type 3. Relational Database

- A **relational model** organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row.
- Rows are also called records or tuples. Columns are also called attributes.

Example of the Relational DBMS:

name	age	country
Natalia	11	Iceland
Ned	6	New York
Zenas	14	Ireland
Laura	8	Kenya

Advantages of using RDBMS:

- Ease of Use
- Network Access
- Language
- Performance
- Prevents Data Redundancy
- Privileges and Data Security

Disadvantages of RDBMS:

- Cost
- Lack of Speed
- Memory Space

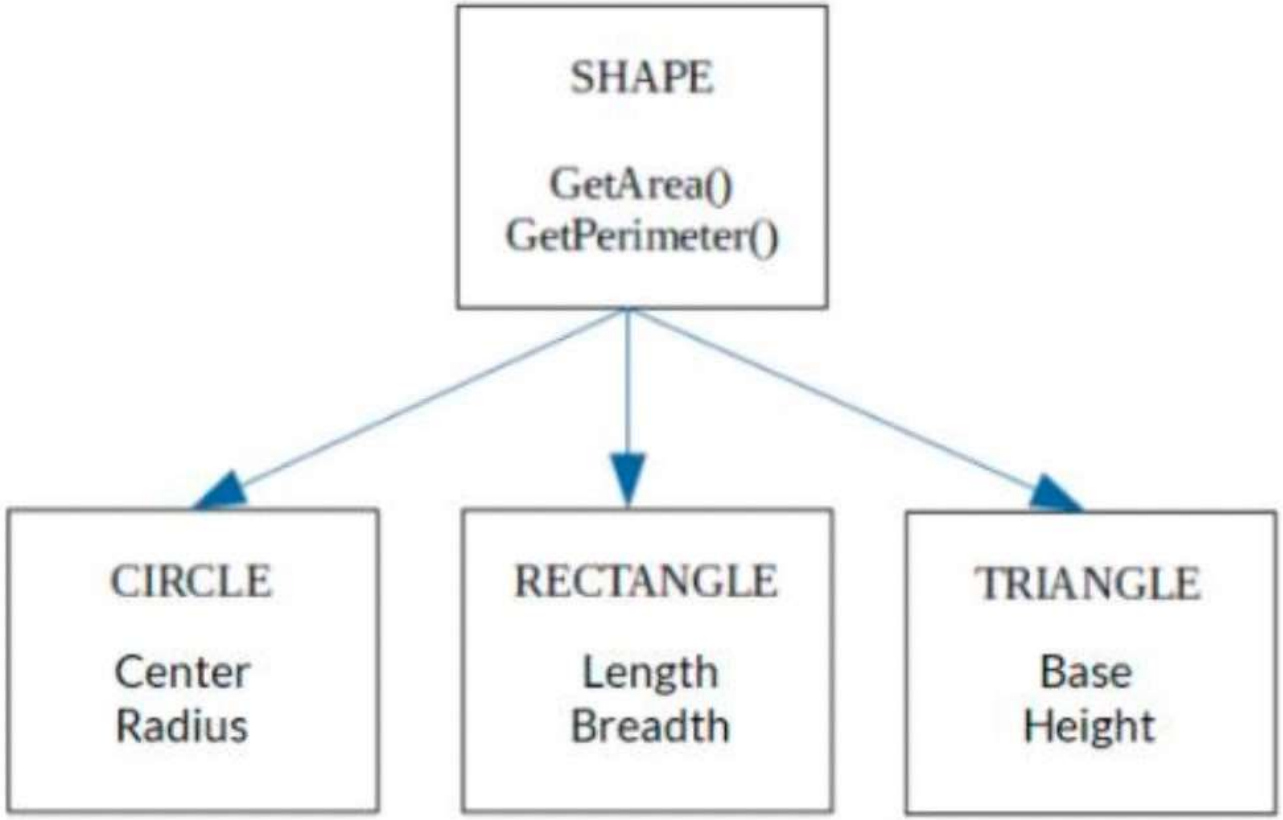
Type 4. Object-Oriented Model

- An **object-oriented database** (OOD) is a database system that can work with complex data objects — that is, objects that mirror those used in object-oriented programming languages. In object-oriented programming (OOP), everything is an object.

Elements of Object-Oriented data model:

- Object
- Attributes and Method
- Class
- Inheritance

Example of Object-Oriented data model:



Advantages of Object-Oriented data model:

- **Reusability:** generic objects can be defined and then reused in numerous application.
- **Complex data types:** Can manage complex data such as document, graphics, images, voice messages, etc.
- **Distributed databases:** Due to mode of communication between objects, OODBMS can support distribution of data across networks more easily.

Disadvantages of Object Oriented Databases:

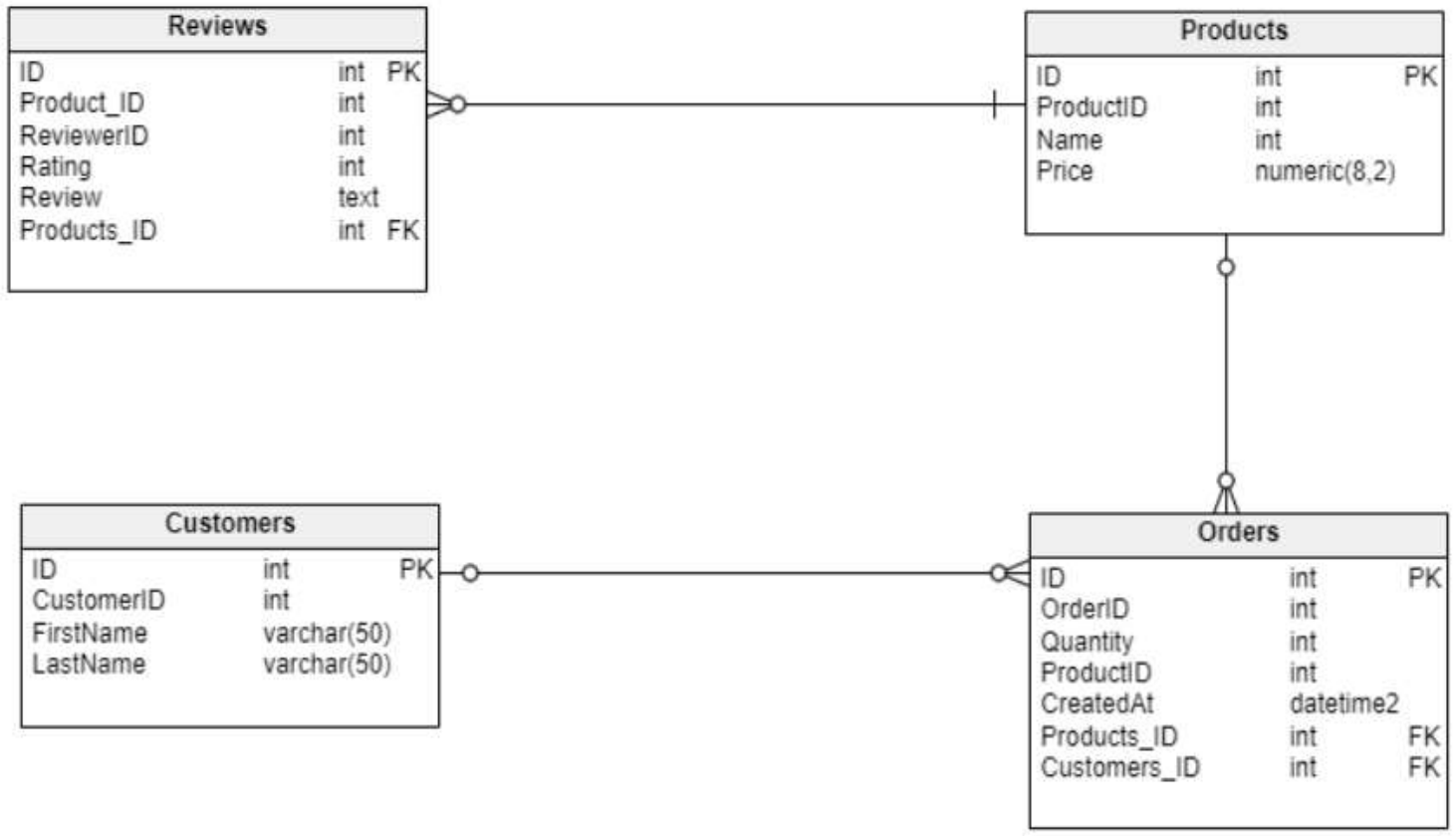
- Competition
- Complexity
- Lack of support for views
- Lack of support for security
- Lack of standards
- Lack of experience
- Lack of universal data model
- Query Optimization compromises encapsulation

ENTITY RELATIONAL (ER) MODEL

- is a high-level conceptual data model diagram.
- ER modeling helps you to analyze data requirements systematically to produce a well-designed database.
- The Entity-Relation model represents real-world entities and the relationship between them.

ER DIAGRAMS

- **ENTITY-RELATIONSHIP DIAGRAM (ERD)** displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases.



Why use ER Diagrams?

- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ERD is allowed you to communicate with the logical structure of the database to users

PART III. ER Diagram. Entity, Attributes, Relationships.

Entity Name

Entity

Person,place,object,event or concept about which data is to be maintained

Example: Car, Student

Jack

Attribute Name

Attribute

Property or characteristic of an entity

Example: Color of car Entity
Name of Student Entity

Relation

Verb Phrase

Association between the instances of one or more entity types

Example: Blue Car Belongs to Student Jack

WHAT IS ENTITY?

- A real-world thing either living or non-living that is easily recognizable and nonrecognizable.
- An entity can be place, person, object, event or a concept, which stores data in the database.
- **Examples of entities:**
- **Person:** Employee, Student, Patient
- **Place:** Store, Building
- **Object:** Machine, product, and Car
- **Event:** Sale, Registration, Renewal
- **Concept:** Account, Course

ATTRIBUTES

- Entities are represented by their properties, which also called attributes.
- For example, a student entity may have a name, age, class, as attributes.

Relationships

- There are three types of relationships between entities (tables) in data modeling:
- One-to-many relationships (also denoted as 1:M).
- Many-to-many relationships (M:N).
- One-to-one relationships (1:1).

Steps to Create an ERD

z2odz40 hu