Royal University of Bhutan

DBS101 Database Systems Fundamentals

Lesson 2

Learning Outcomes

1. Describe different types of data models used in database design.
2. Describe database languages and their applications.
3. Explain fundamentals of database system architecture and their components.
4. Differentiate types of database users and their roles.
5. Discuss the key responsibilities of database administrators.

Flat files

Flat files: A type of data storage file in which data is stored as plain text.

Example: Excel spreadsheet, Word documents, CSV files etc.

Let’s take a look at storing information of students in flat files.

Students.xlxs – table 1 and studentdetails.xlxs – table 2

StudentNo,	Name	Programme
12345	Sonam	BESWE
24656	Tandin Sonam	BESWE

StudentNo,	Name	CID
12345	Sonam	1829289822
24656	Tandin Sonam	298392892

Get the CID number of “sonam”

The program would look like:

```
For line in file.readline():
    record = parse(line)
    If record[1] == “sonam”:
        print(record[2])
```

Flat files

Data Integrity

- How do we make sure that the CID number for the sonam is correct?
- What if there are multiple people named sonam?
- What if someone deletes one of the lines?

Implementation

- How do you find the detail of just one student?
- What if we create a new application that uses the same database? What if the application is running on a different machine?
- What if two threads try to access the same resource/database?

Flat files

- What if the machine crashes while our program is updating a record?
- What if we want to replicate the database on multiple machines for high availability?

The solution is as you have guessed a DBMS.

A DBMS is a software that allows applications to store and analyze information in a database.

Flat files VS DBMS

Link to Study Guide

View of Data

A general-purpose DBMS supports the definition, creation, querying, update, and administration of databases in accordance with some data model.

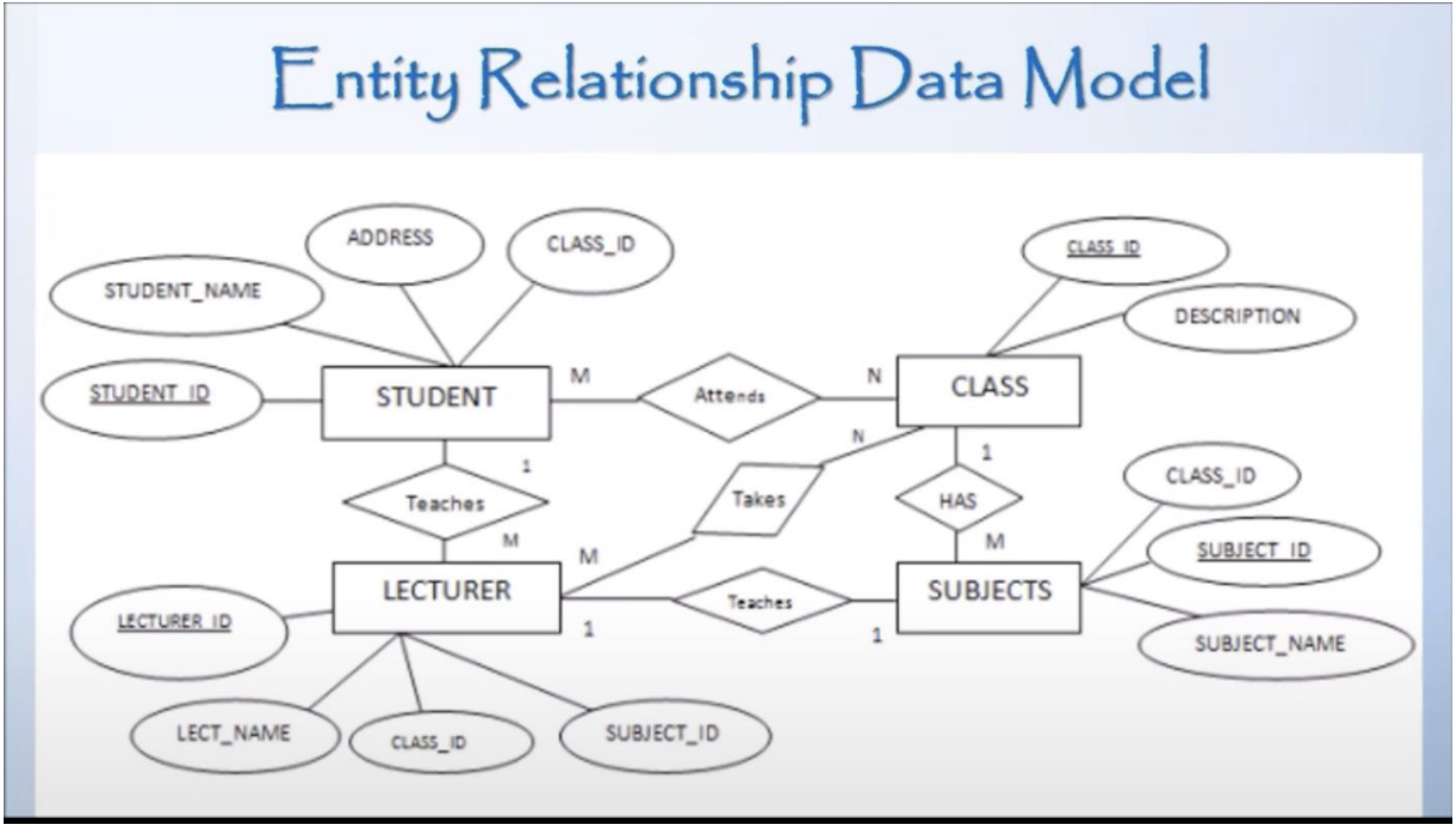
Underlying structure of a database: Data model.

Data Model: a collection of conceptual tools for describing data,data relationships, data semantics, and consistency constraints.

Data Models

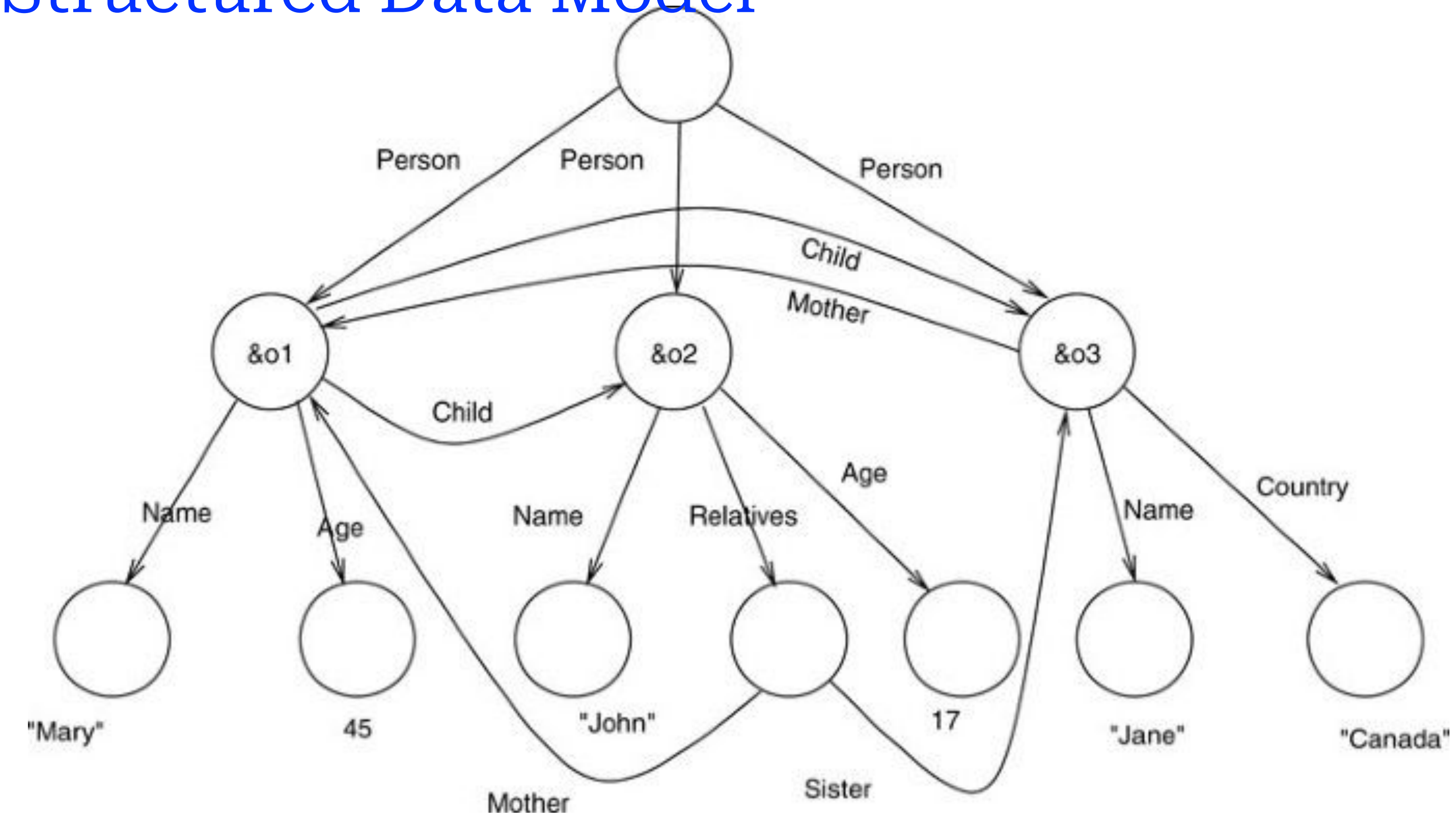
There are many different data models and they can be classified into 4 categories:

1. Entity-Relationship Model
2. Semi-structured Data Model
3. Object Based Data Models
4. Relational Model

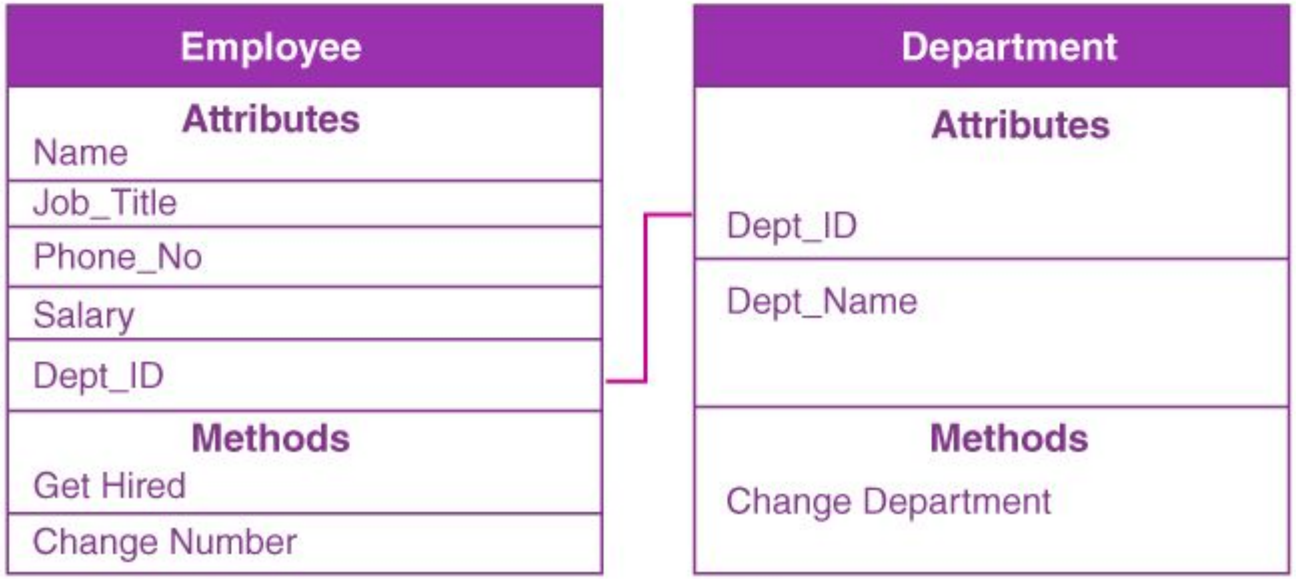


Entity Relationship Data Model

Semi-Structured Data Model



Object Based Data Model



Employee

Attributes

- Name
- Job_Title
- Phone_No
- Salary
- Dept_ID

Methods

- Get Hired
- Change Number

Department

Attributes

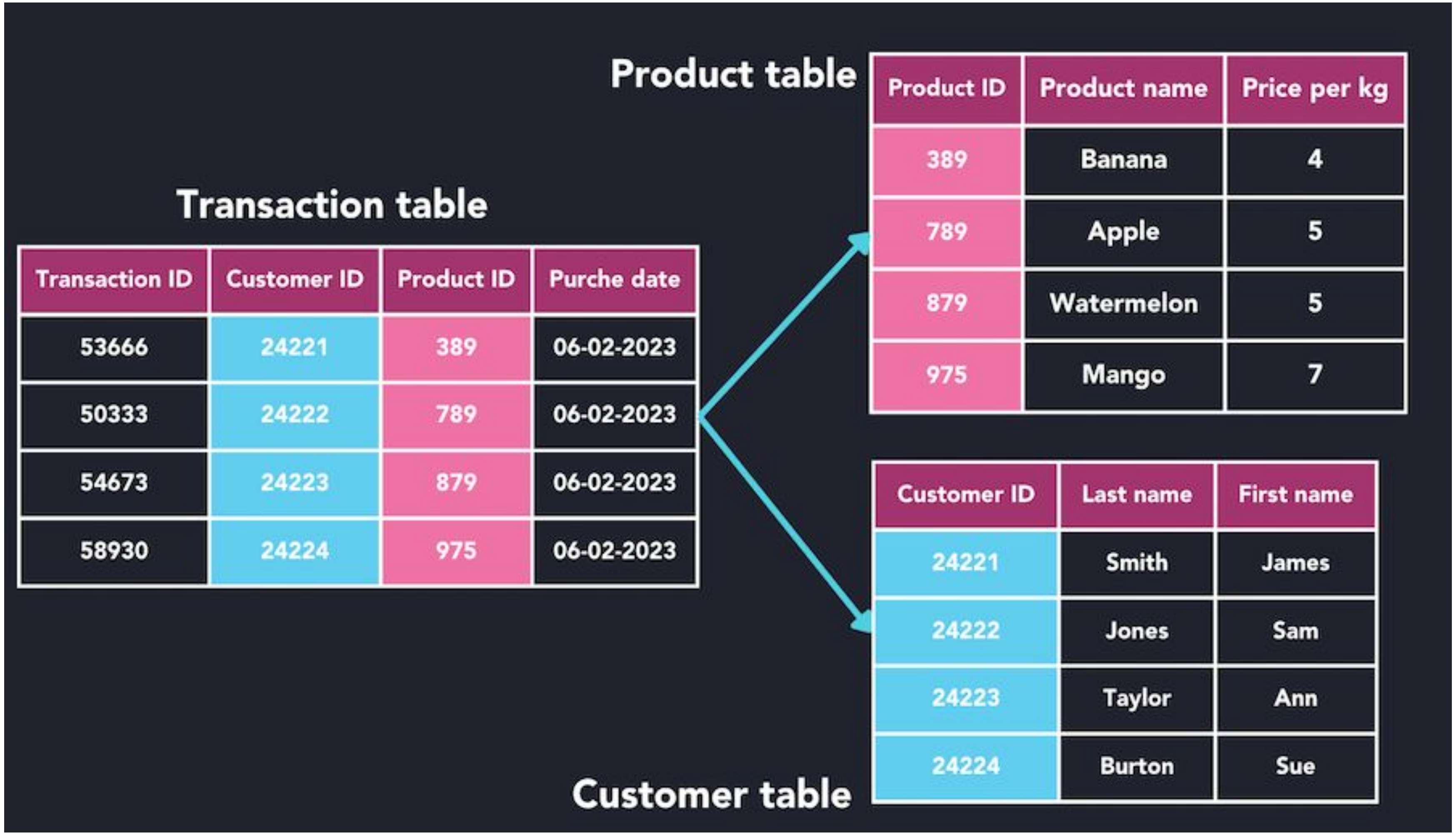
- Dept_ID
- Dept_Name

Methods

- Change Department

Object Oriented Model

Relational Data Model



Transaction table

Transaction ID	Customer ID	Product ID	Purche date
53666	24221	389	06-02-2023
50333	24222	789	06-02-2023
54673	24223	879	06-02-2023
58930	24224	975	06-02-2023

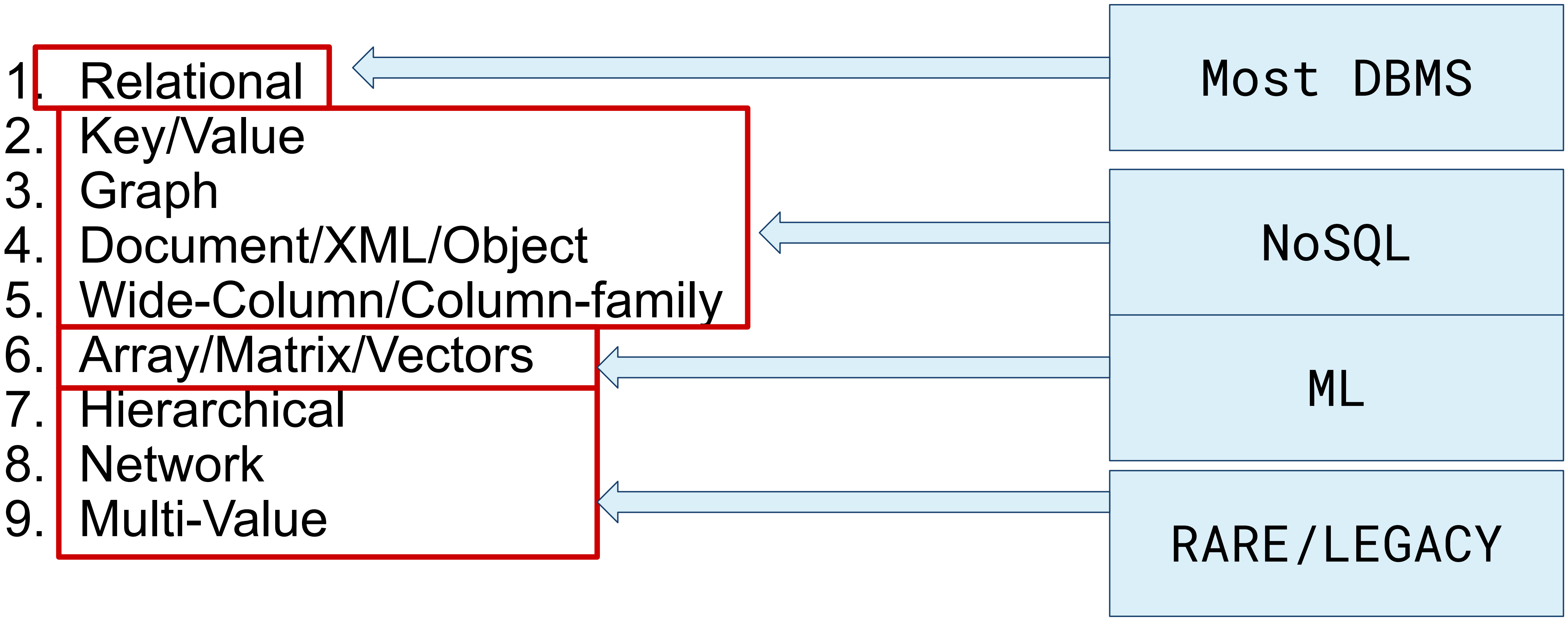
Product table

Product ID	Product name	Price per kg
389	Banana	4
789	Apple	5
879	Watermelon	5
975	Mango	7

Customer table

Customer ID	Last name	First name
24221	Smith	James
24222	Jones	Sam
24223	Taylor	Ann
24224	Burton	Sue

Data Models



1. Relational
2. Key/Value
3. Graph
4. Document/XML/Object
5. Wide-Column/Column-family
6. Array/Matrix/Vectors
7. Hierarchical
8. Network
9. Multi-Value

Early DBMS

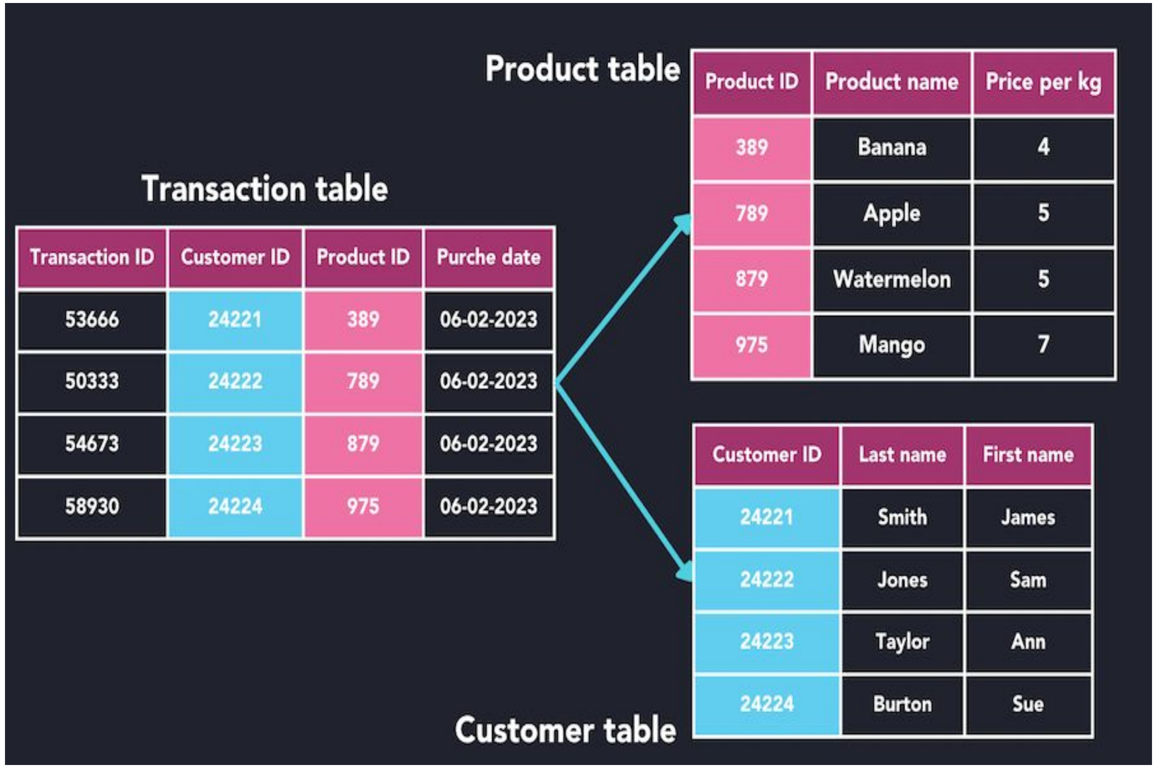
- Difficult to build and maintain
- Tight coupling between logical and physical layers
- Programmers had to know what queries(requests on data) the application would execute before they could deploy the database.

Edgar F. Codd

- Mathematician at IBM Research late 1960s.
- Devised relational model in 1969.
- The relational model separated the logical and physical layers.
- The relational model was put into practice in the late 1970s in IBM’s product System R.

Relational Model

- data are represented in the form of tables.
- Each table has multiple columns, and each column has a unique name
- Each row of the table represents one piece of information.
- SQL (Sequential Query Language) is used to create,read,update and delete(CRUD) tables and entries



Transaction table

Transaction ID	Customer ID	Product ID	Purche date
53666	24221	389	06-02-2023
50333	24222	789	06-02-2023
54673	24223	879	06-02-2023
58930	24224	975	06-02-2023

Product table

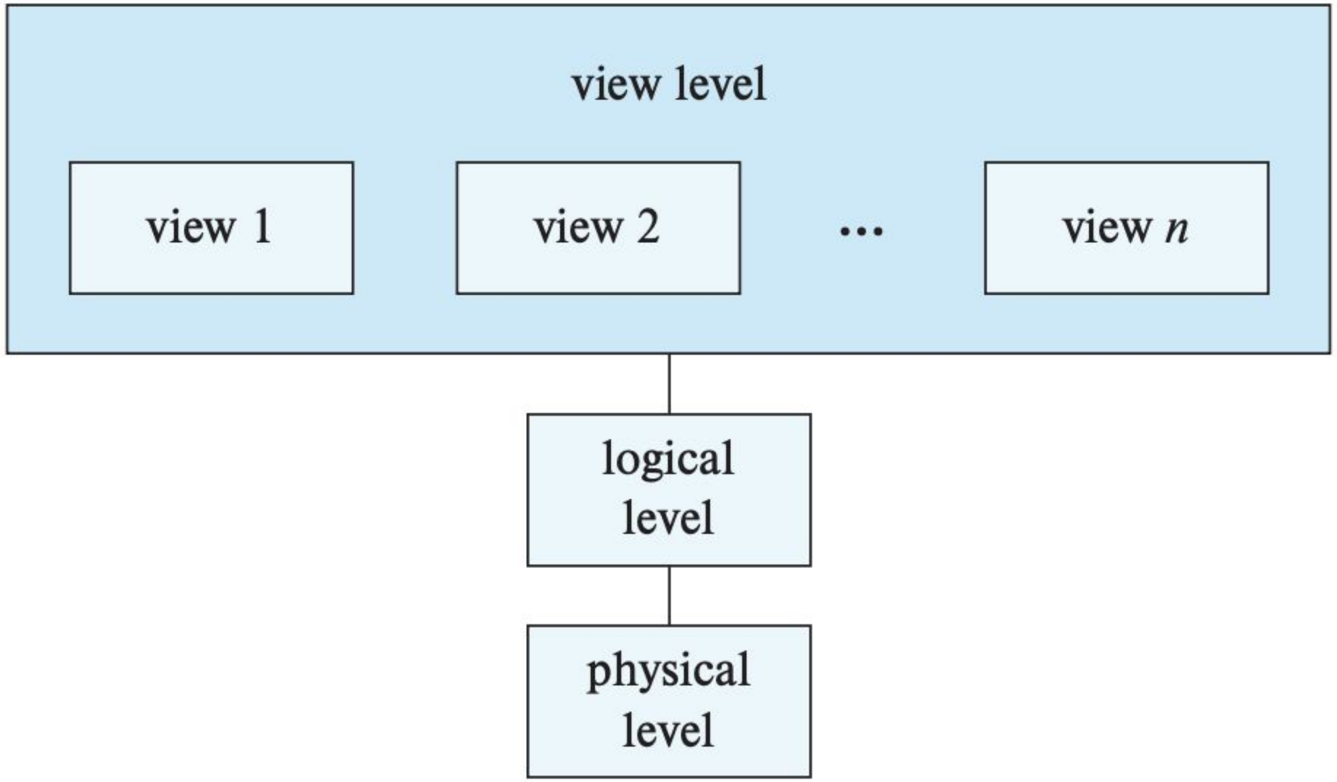
Product ID	Product name	Price per kg
389	Banana	4
789	Apple	5
879	Watermelon	5
975	Mango	7

Customer table

Customer ID	Last name	First name
24221	Smith	James
24222	Jones	Sam
24223	Taylor	Ann
24224	Burton	Sue

Data Abstraction

- Physical level: describes how the data are actually stored.
- Logical level: describes what data are stored in the database, and what relationships exist among those data.
- View level: describes only part of the entire database



Data Abstraction

- Developers hide complexity of data structures in databases from database users.
- Database systems allow developers to use abstractions of data models to manage data.

Take for example: Go to slide 13

Database Instances and Schemas

Database Instance: The collection of information stored in the database at a particular moment is called an instance of the database.

Database schema: Overall design of the database.

Database systems have several schemas partitioned into levels of abstraction like physical and logical schemas.

Which schema do you think affects application development the most?

Database Languages

1. Data-definition language
   - Specify database schema
2. Data-manipulation language
   - Express database queries and updates.

The two languages simply form the parts of a single database language like SQL.

1. Data Definition Language

- Specifies database schema
- Can also help to specify:
  Domain constraints,
  Referential Integrity,
  & Authorization

```
SQL DDL Example:
create table department
   (dept_name char (20) Not Null,
    building char (15),
    budget numeric (12,2));
Check if constraint works:
INSERT INTO Department
(building, budget)
VALUES('Dzong', 40);
```

2. Data Manipulation Language

- A data-manipulation language enables users to access or manipulate data as organized by the appropriate data model. The types of access are:
- Retrieval
- Insertion
- Deletion
- Modification

2 types of DMLs:

- **Procedural DML**: require a user to specify what data are needed and how to get those data.

- **Declarative DMLs** (also referred to as **nonprocedural DMLs**): require a user to specify what data are needed without specifying how to get those data.

2. Data Manipulation Language

- A **query** is a statement requesting the retrieval of information.
- The portion of a DML that involves information retrieval is called a **query language**.

- SQL query language is non procedural.
- A query takes as input several tables (possibly only one) and always returns a single table.
- Example:

```
select Orders.item from Orders where

Orders.customer_id = 1;
```

Database Access from Application Programs

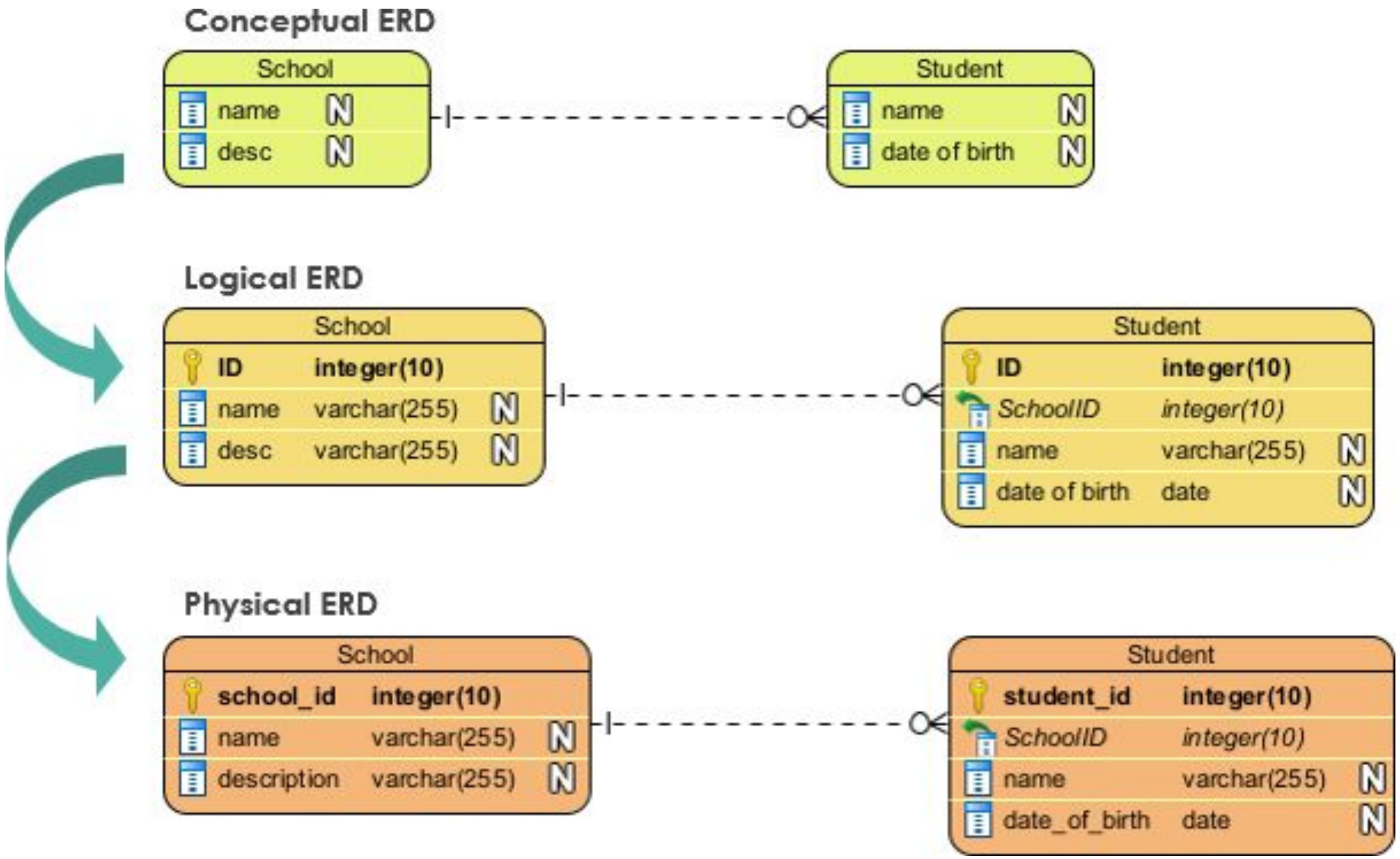
A **host programming language** must be used to send DML statements(SQL statements), usually done with the help of an **Application Programming Interface**(set of procedures).

The **Open Database Connectivity** (ODBC) standard defines application program interfaces for use with C and several other languages.

Database Design

1. Choose a **data model**
2. Develop a **conceptual design** – gives functional requirements of the enterprise
3. Develop **Logical design** – map the high-level conceptual schema onto the implementation data model of the database system
4. Develop **Physical design** – physical features of the database are defined

Example Entity Relationship Diagram



Conceptual ERD

School	
name	N
desc	N

Student	
name	N
date of birth	N

Logical ERD

School		
ID	integer(10)	
name	varchar(255)	N
desc	varchar(255)	N

Student		
ID	integer(10)	
SchoolID	*integer(10)*	
name	varchar(255)	N
date of birth	date	N

Physical ERD

School		
school_id	integer(10)	
name	varchar(255)	N
description	varchar(255)	N

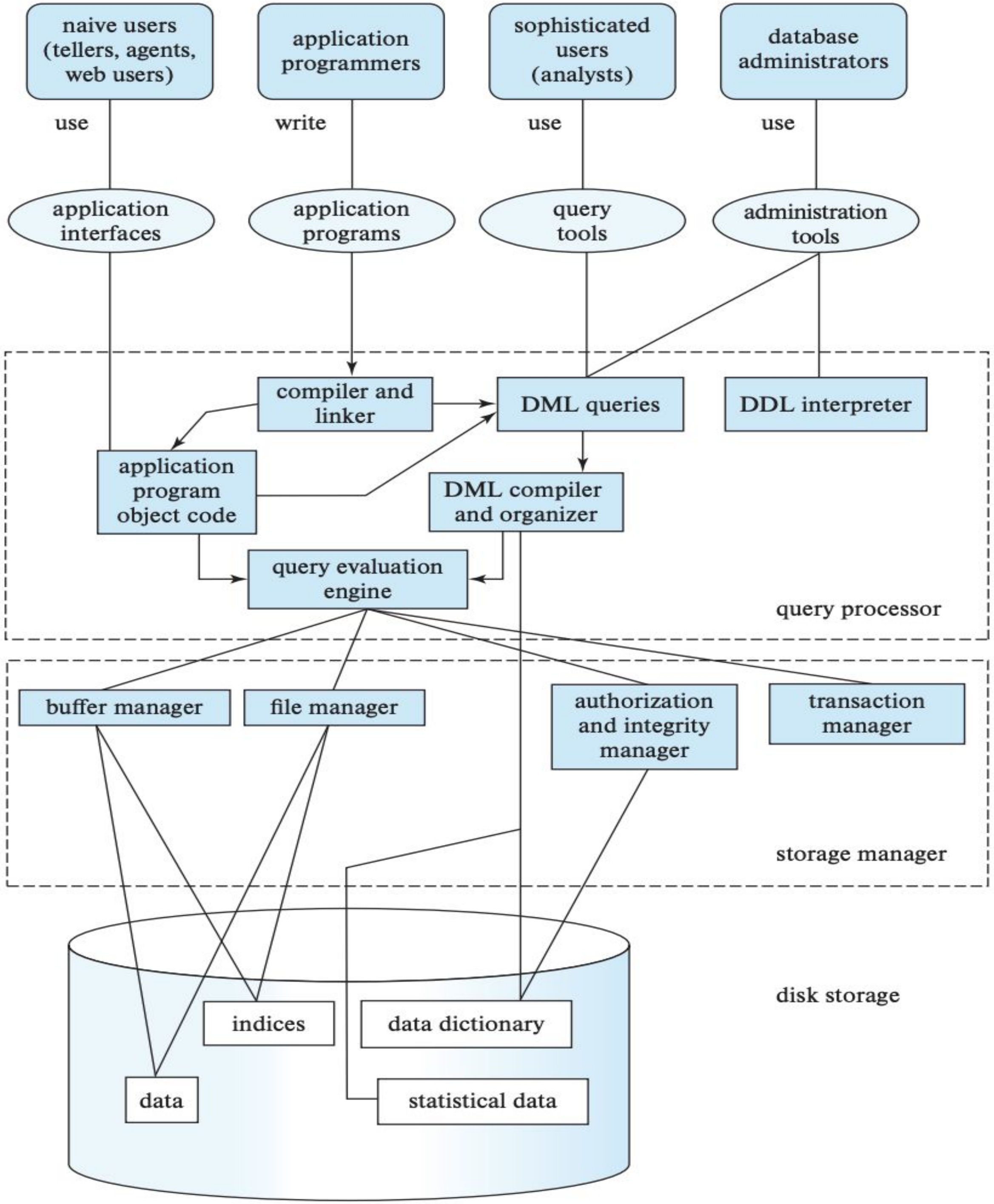
Student		
student_id	integer(10)	
SchoolID	*integer(10)*	
name	varchar(255)	N
date_of_birth	date	N

Database Engine

Functional Components of a database system:

1. **Storage Manager**: provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
2. **Query Processor**: The query processor components include: DDL interpreter, DML compiler and Query evaluation engine.
3. **Transaction Management**: The transaction manager consists of the concurrency-control manager and the recovery manager

Database System Architecture



Architecture of Applications that use Databases as Backend

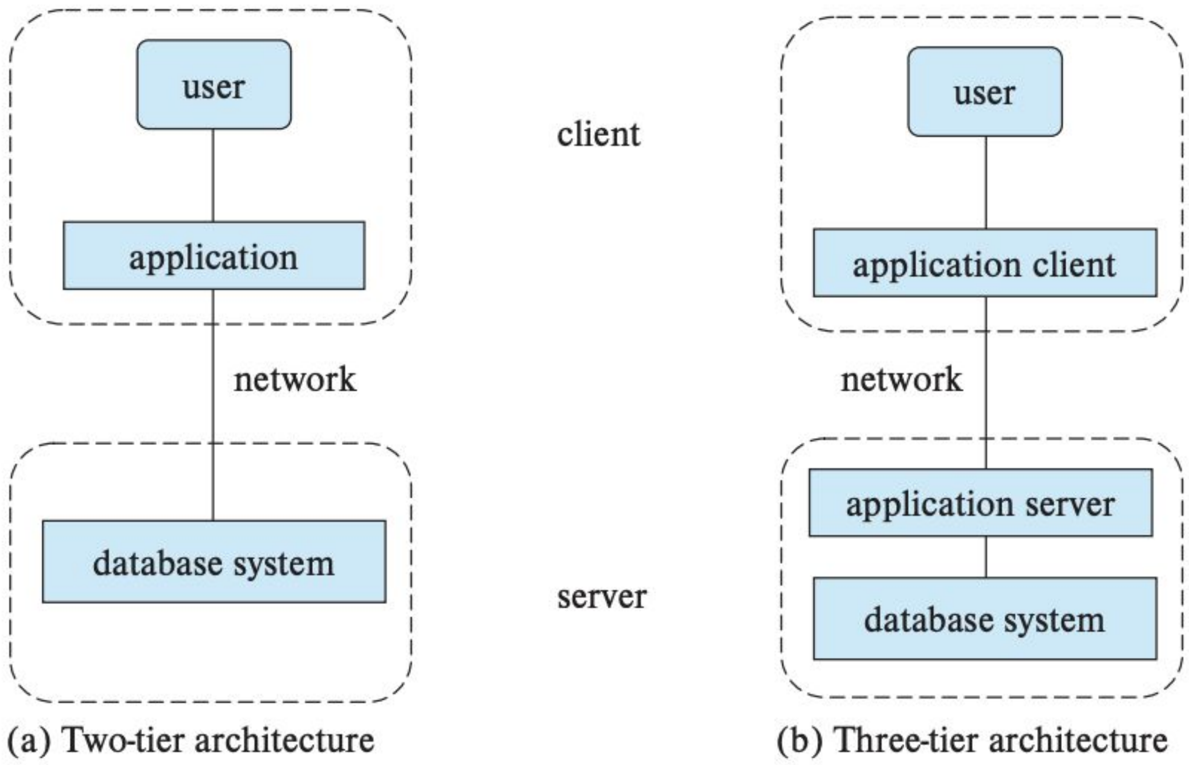


Figure 1.4 Two-tier and three-tier architectures.

Break Time for 10 minutes!!! Afterwards a flipped Class on Database Users and Administrators

References

Codd, F. (2009). Derivability, redundancy and consistency of relations stored in large data banks. *Sigmod Record*, *38*(1), 17–36. https://doi.org/10.1145/1558334.1558336

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377–387. https://doi.org/10.1145/362384.362685

Last Bench Listener. (2017, December 18). *Object based data Model(ER model & object oriented data model) with example (lecture-7)*. YouTube. https://www.youtube.com/watch?app=desktop&v=9MEVHebExhw

Vpadmin. (2023, September 18). Navigating the three levels of database design: conceptual, logical, and physical - visual paradigm guides. Visual Paradigm Guides. https://guides.visual-paradigm.com/navigating-the-three-levels-of-database-design-conceptual-logical-and-physical/#:~:text=Conceptual%20database%20design%20is%20the,getting%20into%20technical%20implementation%20details.

Next Week

Unit II: Database Design

2.1 Overview of the Design Process

2.2 Relational Modeling

- Relational DB Schema
- Relational Query Language
- Relational Algebra

2.3 The Entity-Relationship Model

2.4 Complex Attributes

2.5 Mapping Cardinalities

2.6 Primary Key