Databases

Introduction to Databases

João R. Campos

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2024/2025

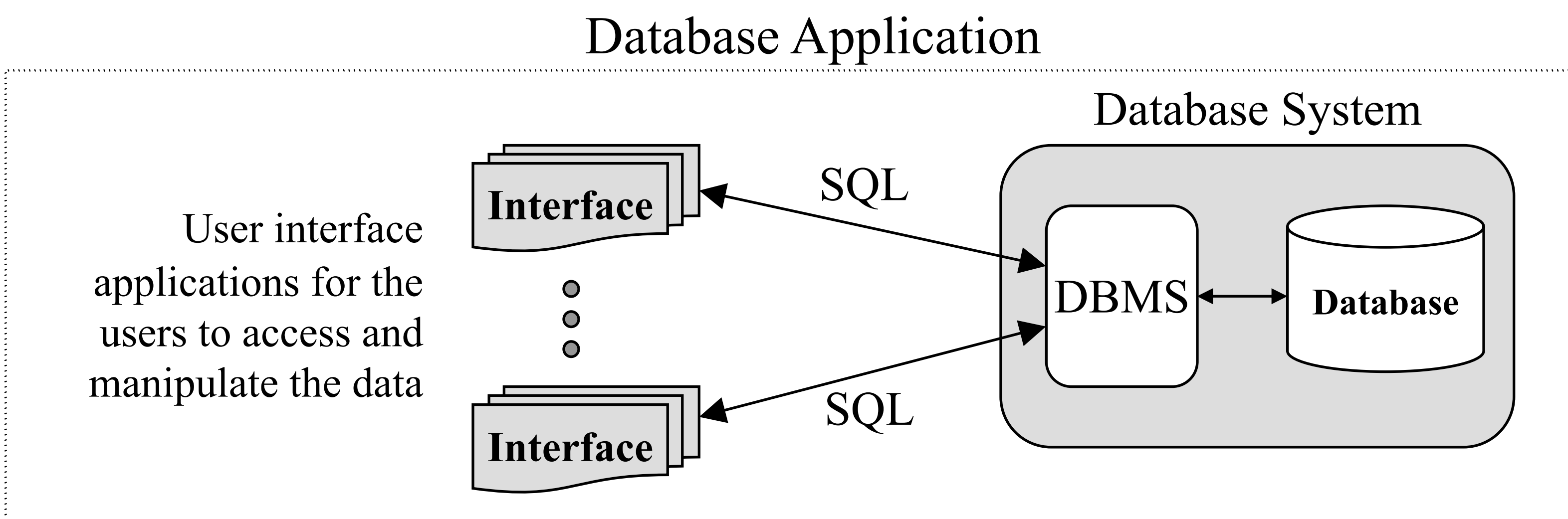
Outline

- Databases, DBMS and Applications
- Why Database Systems?
- Data Models and Data Abstraction
- Database Design
- Database Languages
- Database Engine
- System and Application Architecture
- Users and Administrators

These slides use the following book as reference: Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

Databases, DBMS and Applications

- A database is a collection of interrelated data
- A Database Management System (DBMS) is software package to define, manipulate, retrieve and manage data in a database
  - e.g., Oracle Database, PostgreSQL, MySQL, IBM DB2, Microsoft SQL Server
- The complete system that allow accessing, manipulating and handling the data is a database application



Database Applications

- A database application handles information about a particular enterprise/organization
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both convenient and efficient to use
- Database applications are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple users and applications, often at the same time
- Databases touch all aspects of our lives!

Examples of Database Applications

- Enterprise Information Systems
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.
- Manufacturing
  - Management of production, inventory, orders, supply chain
- Banking
  - Customer information, accounts, loans, and banking transactions
  - Credit card transactions
- Finance
  - Sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data)

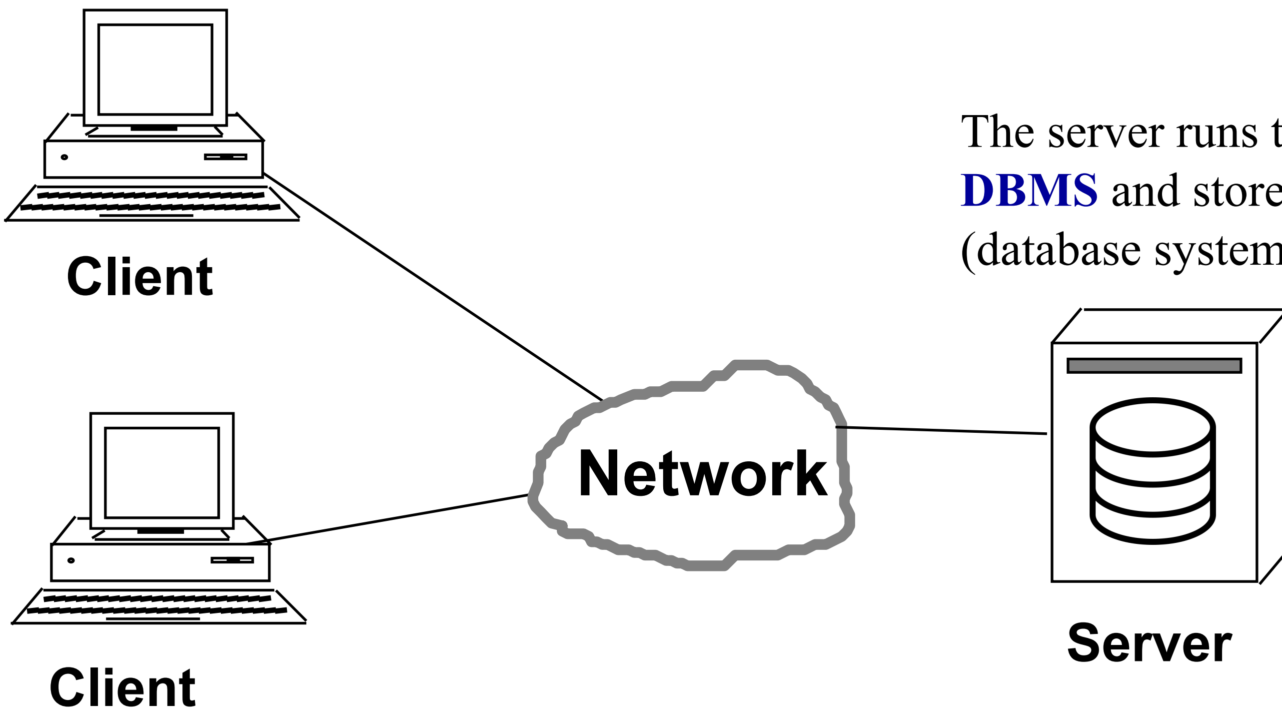
Examples of Database Applications

- Airlines
  - Reservations, schedules
- Telecommunication
  - Records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards
- Web-based services
  - Online retailers: order tracking, customized recommendations
  - Online advertisements
- Document databases
- Navigation systems
  - For maintaining the locations of varies places of interest along with the exact routes of roads, train systems, buses, etc.

Basic Database Application…

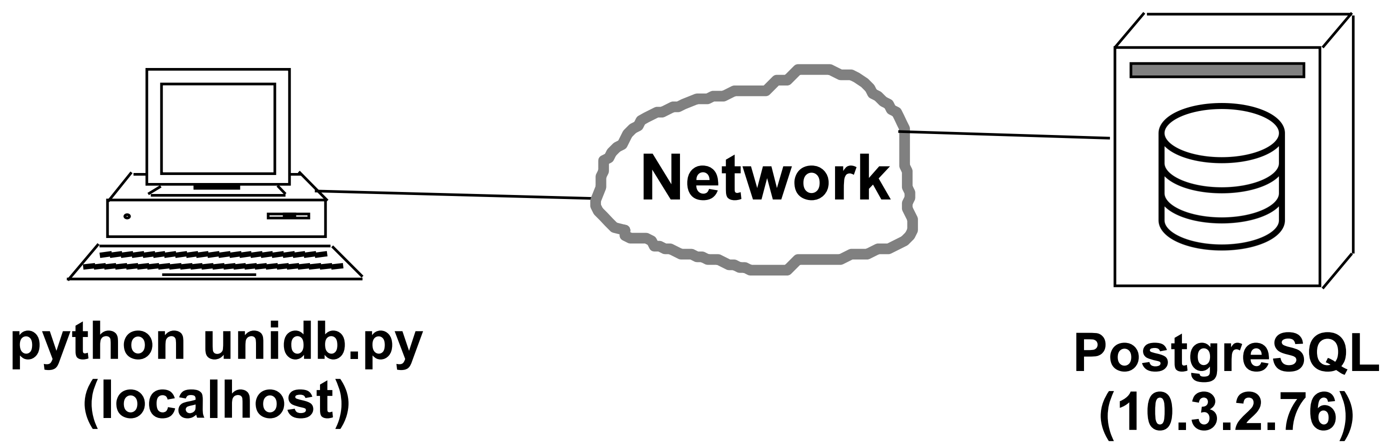
The clients run the user interface applications (standalone, web, mobile, etc.)

The server runs the **DBMS** and stores the data (database system)



DEMO #1

- University database
  - Example that will be used in some classes
- Server: PostgreSQL server running at DEI datacenter
- Client: Simple Python program running in my own machine
- Functionalities: Get the list of instructors and departments



network: DEI / vpn

Why Database Systems?

- In the early days, database applications were built directly on top of file systems, which leads to many problems/difficulties
- Data redundancy and inconsistency
  - Data stored in multiple formats resulting in duplication of information
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation issues
  - Multiple files and formats
- Integrity problems
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones

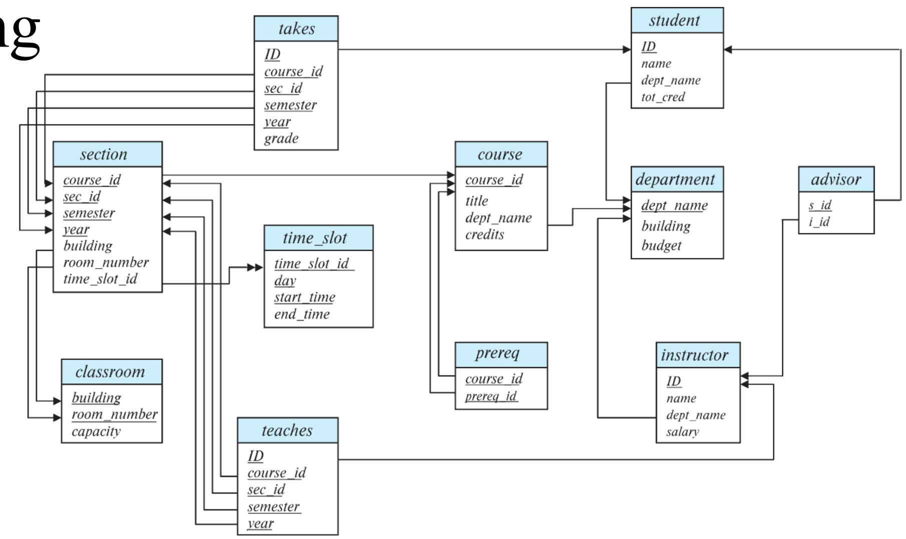
Why Database Systems?

- Atomicity of updates
  - Failures may leave database in an inconsistent state
  - Example: transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - e.g., two people reading a balance and updating it by withdrawing money at the same time
- Security problems
  - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints



- Relational model
- Entity-Relationship (ER) data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)
- Other older models: network model, hierarchical model

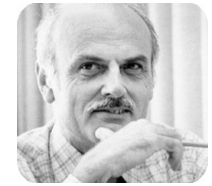
Relational Model

columns

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

rows



Ted Codd
Turing Award 1981

department

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Schemas and Instances

- The logical schema describes the overall logical structure of the database
  - e.g., the database consists of information about a set of customers and accounts in a bank and the relationship between them
- The physical schema describes the overall physical structure of the database
- An instance is the actual content of the database at a particular point in time

Database Design

- How we define the general structure of the database?
- Logical design is the process of deciding about the database schema
  - Database design requires that we find the right collection of relation schemas
  - What attributes should we record in the database? Business decision…
  - What relation schemas should we have and how should the attributes be distributed among the various relation schemas? Technical decision…
- Deciding on the physical layout of the database occurs in the physical design phase
  - Physical design is determined by the output of the logical design phase

Logical Design	$\Rightarrow$	Physical Design
Conceptual Model (entities and relationships)		*Physical Model* (tables)

Database Languages

- A database sytem provides a language to specify the database schema and to express queries and updates to the data:
  - Data Definition Language (DDL): defining relations, deleting relations, and modifying relations
  - Data Manipulation Language (DML): query information from the database and insert, delete and modify tuples
- Structured Query Language (SQL) is the reference language for databases

Data Definition Language (DDL)

- Specification notation for defining the database schema

```
create table instructor(
        ID char(5),
        name varchar(20),
        dept_name varchar(20),
        salary numeric(8,2),
        primary key (ID));
```

- The DDL compiler generates a set of table templates stored in a data dictionary
- The data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization: who can access what?

Data Manipulation Language (DML)

- Language for accessing and updating the data organized by the appropriate data model (aka query language)
- There are basically two types of data-manipulation languages:
  - Procedural DML – require a user to specify what data are needed and how to get those data
  - Declarative DML – require a user to specify what data are needed without specifying how to get those data
- Declarative DMLs are usually easier to learn and use than procedural DMLs
  - Declarative DMLs are also referred to as non-procedural DMLs
- The portion of a DML that involves information retrieval is called a query language

SQL Query Language

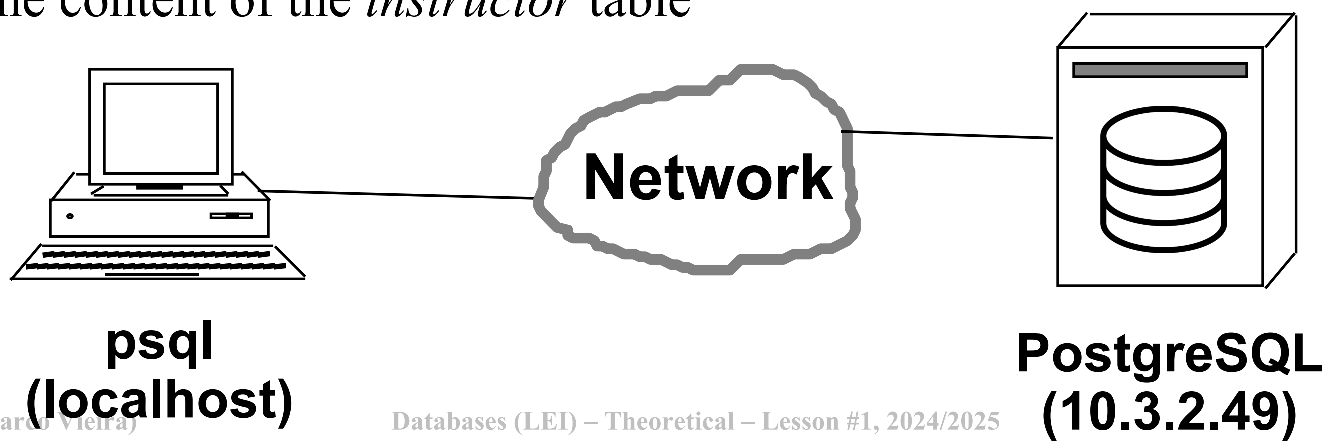
- SQL query language is nonprocedural

```
select name
from instructor
where dept_name = 'Comp. Sci.';
```

- SQL can be embedded in some higher-level language to be able to compute complex functions
- Application programs generally access databases through:
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

DEMO #2

- University database
- Server: PostgreSQL server running at DEI datacenter
- Client: Client program (*psql*) that allows connecting to the database server and execute SQL commands – runs locally
- TODO:
  - Create a simple *courses* table (*ID* and *name*), insert a new line, and then get the content of that table
  - Get the content of the *instructor* table



Access the DB from Application Program

- Non-procedural query languages such as SQL are not as powerful as other programing languages (e.g., JAVA, Python, C/C++)
- SQL does not support actions such as input from users, output to displays, or communication over the network
- Such computations and actions must be written in a host language, with embedded SQL queries that access the data in the database
- An application program is a program that is used to interact with the database, providing some usable interface to the users
  - *unidb.py*, *psql*

Example using Python (*unidb.py*)

```
…
import psycopg2
def list_instructors():
     connection = psycopg2.connect(user = "dblesson1",
         password = "dblesson1",
         host = ” 10.3.2.190",
         port = "5432",
         database = "dbcourse")
    cursor = connection.cursor()
    cursor.execute('select * from instructor')

    print('--- List of Instructors ---')
    for row in cursor:
        print('ID:',row[0])
        print('Name:',row[1])
        print('Department:',row[2])
        print('Salary:',row[3])
        print('---’)
…
```

Database Engine

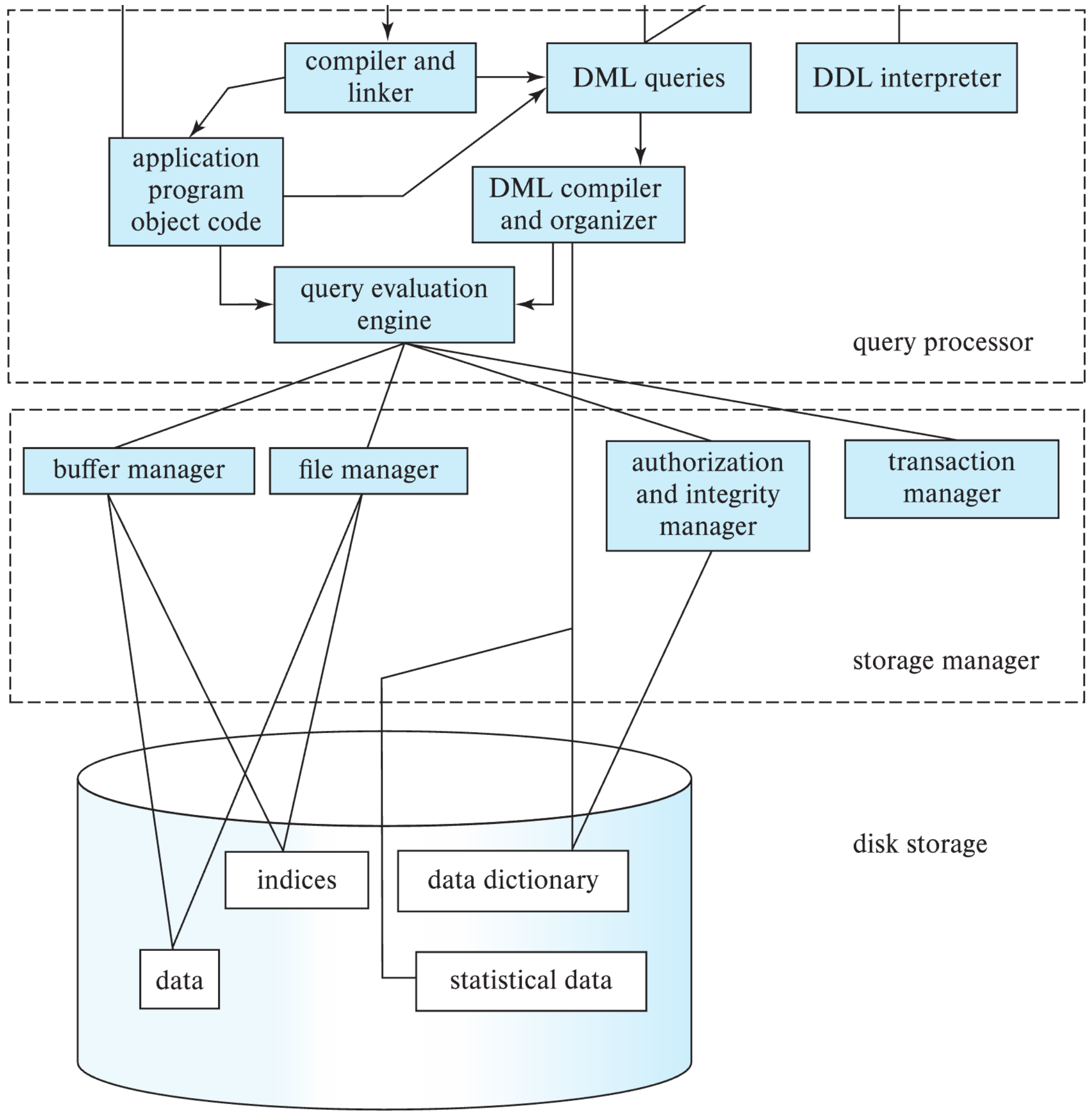
- A database system is partitioned into modules that deal with each of the responsibilities of the overall system
- The functional components of a database system can be divided into storage manager, query processor, and transaction manager
- The storage manager provides the interface between the low-level data stored and the queries submitted to the system
- The query processor is in charge of interpreting, compiling, and evaluating the queries submitted by the application programs
- The transaction manager includes a concurrency control manager and a recovery manager with the goal of assuring ACID transactions

ACID = Atomicity + Consistency + Integrity + Durability

Database Architecture

- Centralized/Shared-Memory
  - Centralized databases, one to a few cores, shared memory
  - The most common
- Parallel databases
  - Many core shared memory
  - Shared disk
  - Shared nothing
- Distributed databases
  - Geographical distribution
  - Schema/data heterogeneity

Centralized Architecture

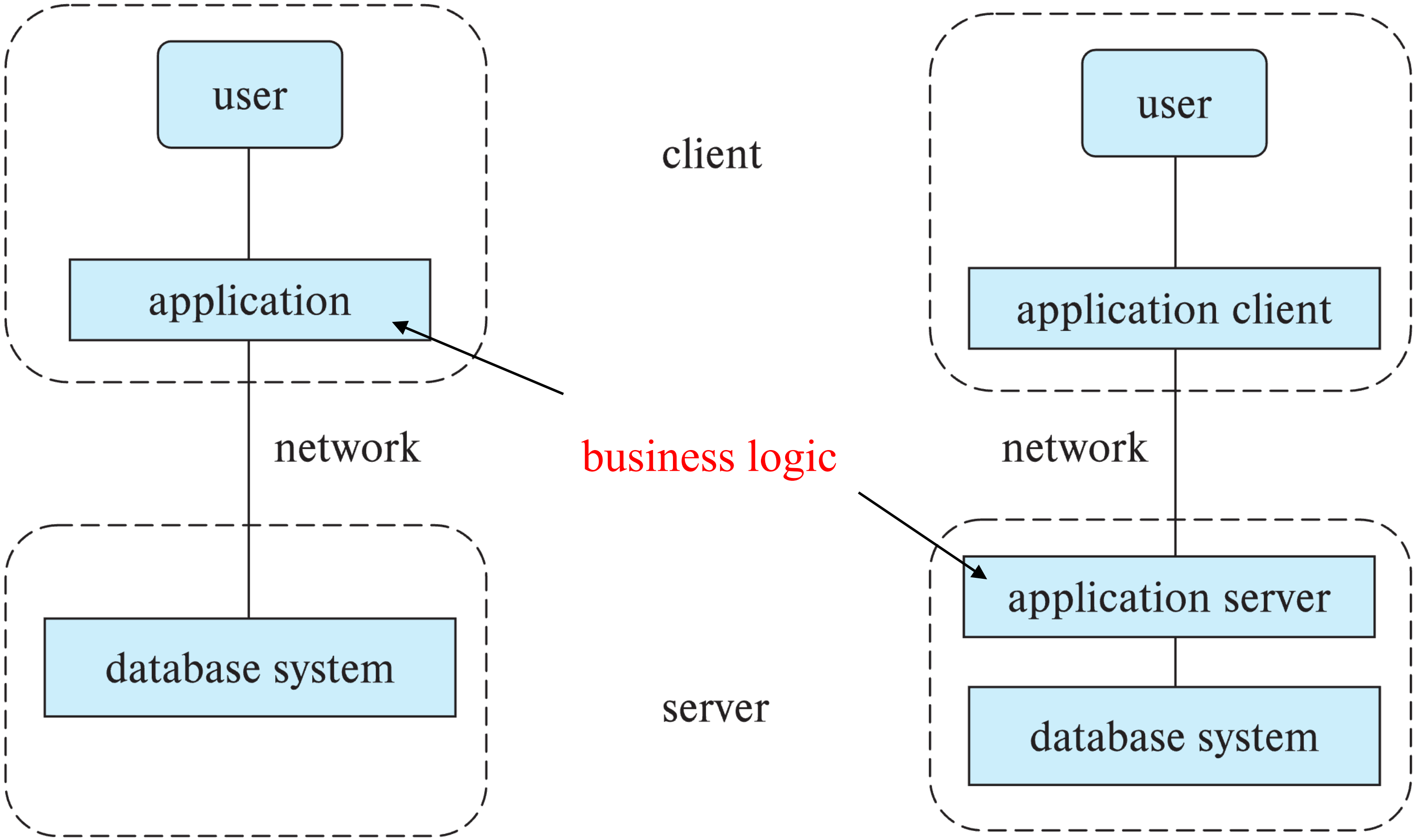


Source: A. Silberschatz, H. F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

Database Applications

- Database applications typically partitioned into two or three parts
  - Other alternatives do exist!
- In a two-tier architecture the application runs in the client machine, where it invokes database system functionality at the server machine
  - Examples we have seen before
- In a three-tier architecture the client machine acts as a front end and does not contain any direct database calls
  - The client communicates with an application server, usually through a forms interface
  - The application server in turn communicates with a database system to access data
- What about n-tier architectures?

Two-tier and Three-tier Architectures



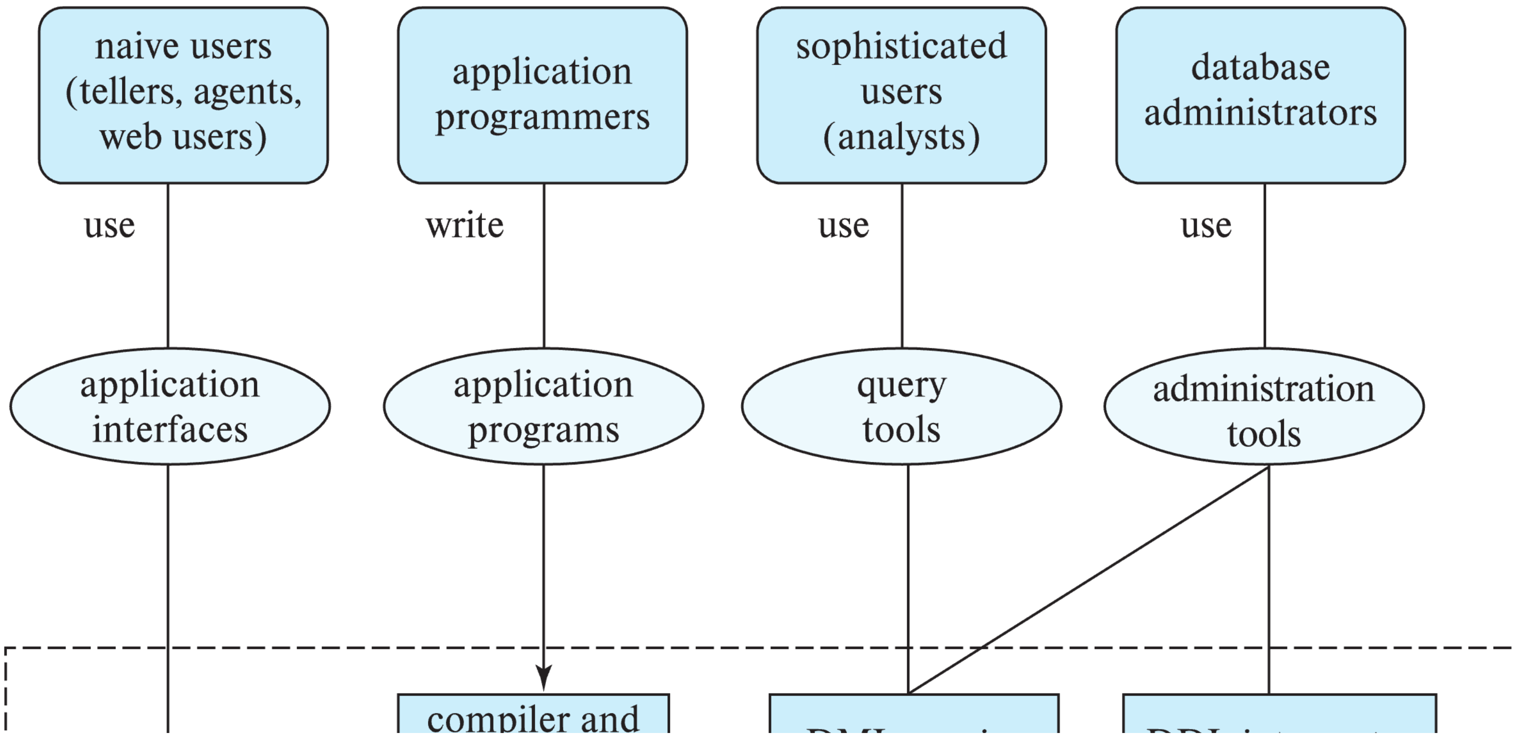
(a) Two-tier architecture

(b) Three-tier architecture

Source: A. Silberschatz, H. F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

Database Users and Administrators

- The people that work with a database can be categorized as database users or database administrators
- Database users: naïve users, application programmers, sophisticated users



Database Administrator

- A person who has central control over the system is called a database administrator (DBA)
- The functions of a DBA include:
  - Schema definition
  - Storage structure and access-method definition
  - Schema and physical-organization modification
  - Granting of authorization for data access
  - Routine maintenance
  - Periodically backing up the database
  - Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
  - Monitoring jobs running on the database

Take-Away(s)

- Databases, DBMS and applications
- Models: relational model
- Design: conceptual model (ER) *vs* physical model (tables)
- Languages: SQL
- Engine: PostgreSQL, Oracle, Microsoft SQL Server
- System Architecture: storage manager, query processor, transaction manager
- Application Architecture: two-tier, three-tier, n-tier
- Users and administrators

Next Lesson(s)

- Introduction to SQL
  - Overview of the SQL Query Language
  - SQL Data Definition Language (DDL)
  - Basic Structure of SQL Queries (select)
  - Modifying the Data (insert, update, delete)
- Relational Model
  - Relational Databases
  - Relation and Database Schema
  - Keys (Primary and Foreign)
  - Schema Diagrams
  - Relational Query Languages

Q&A

Databases

Introduction to Databases

João R. Campos

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2024/2025