escola d'enginyeria

UAB

ESCI upf. School of International Studies

DATABASE

ARCHITECTURE

Debora Gil, Oriol Ramos, Carles Sanchez

Outline

1. Definition
2. ANSI/SPARC
3. Client-Server

Definition

Structure and description of DBS in modules to guarantee the independence

Most common architectures:

ANSI/SPARC

Client-Server:

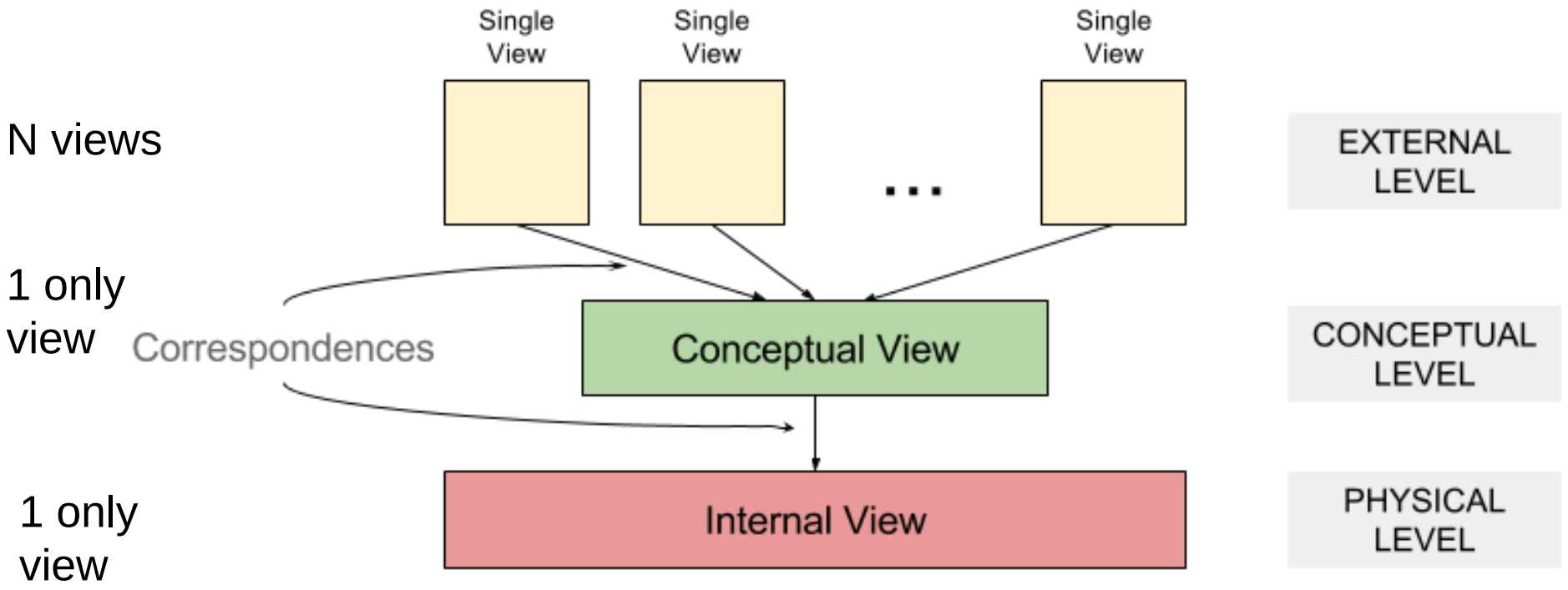
Back-end/Front-end

Distributed systems

2. ANSI/SPARC

ANSI/SPARC

Structure the DB according to the description of the data in 3 levels of abstraction:

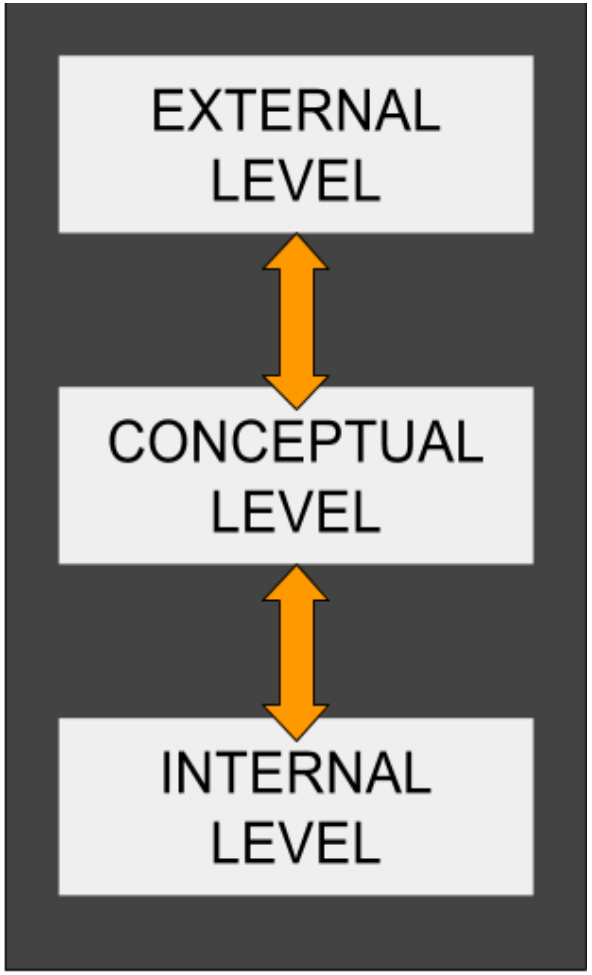


ANSI/SPARC. Abstraction Levels

External Level. Presentation of data to users (applications that use a relational model).

Conceptual Level. Logical description of data: tables (relational), collections (non-relational)

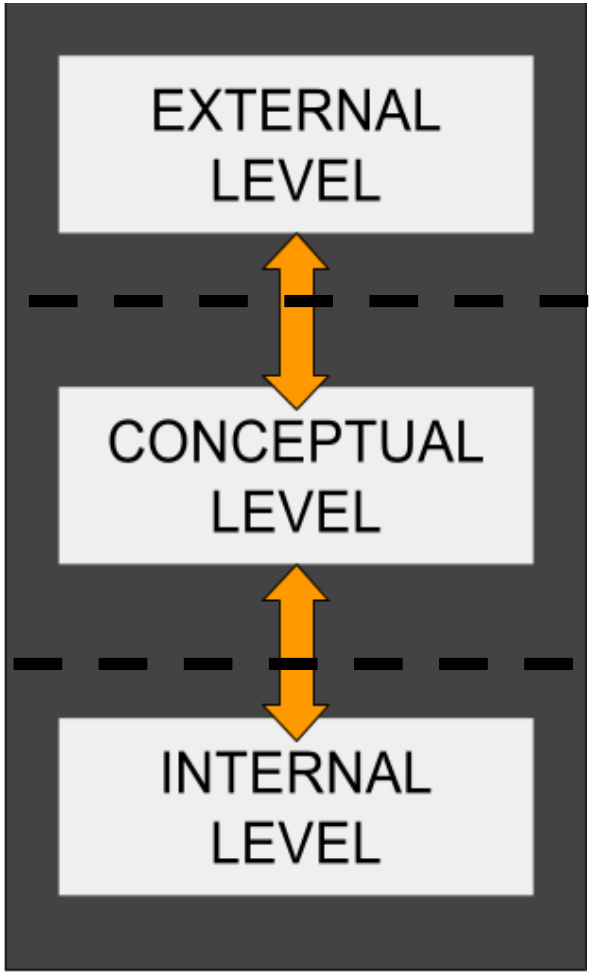
Internal Level. Organization and storage on physical files (low-level, pointers, indexes, …)



Data Independence

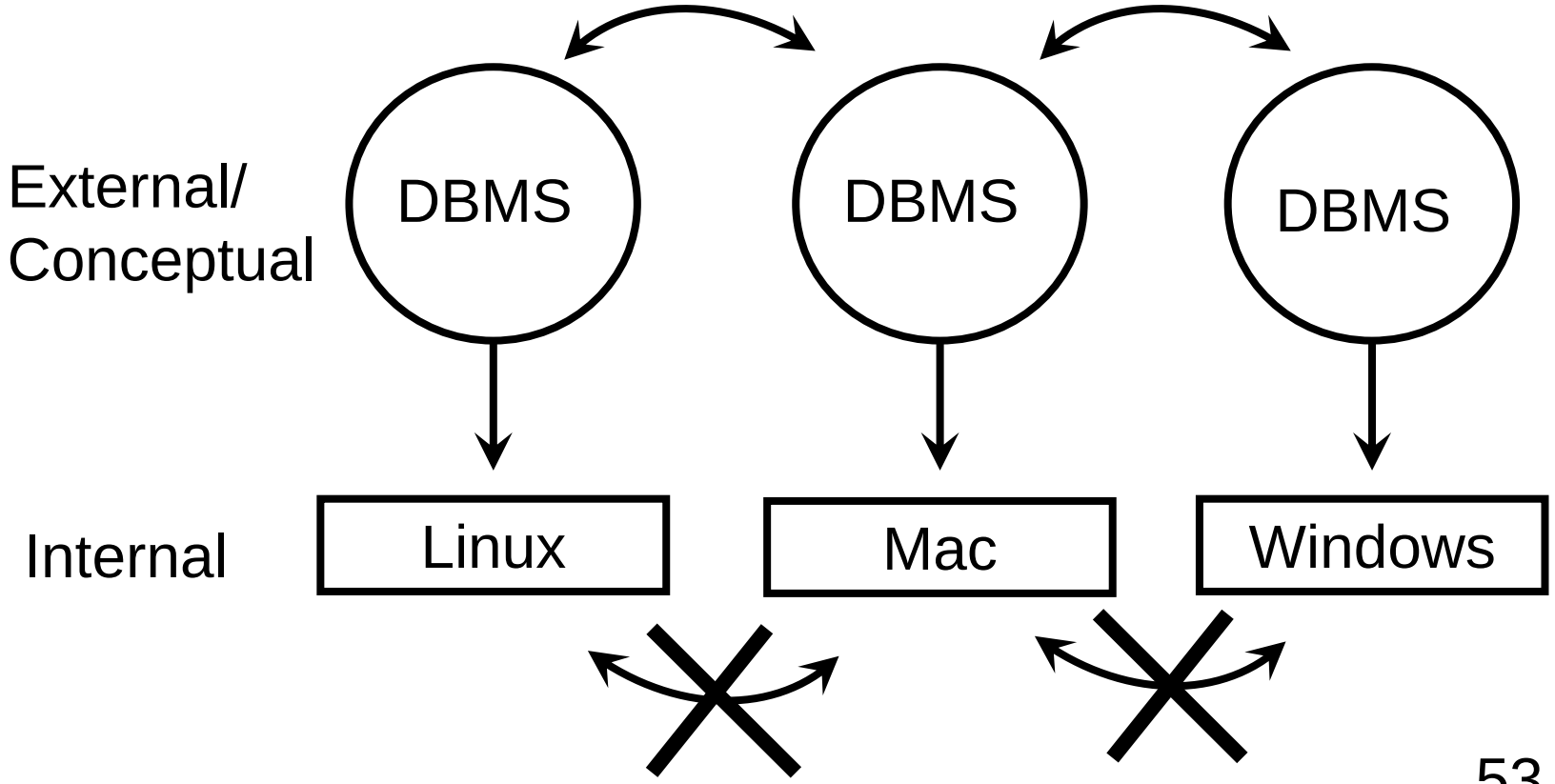
Logic: Modification of the conceptual level without affecting the external level

Physical: Internal distribution does not affect the functionalities of other levels



Data Independence

External and Conceptual Levels **do not** depend of the operating system. Internal level **does** depends of the operating system.

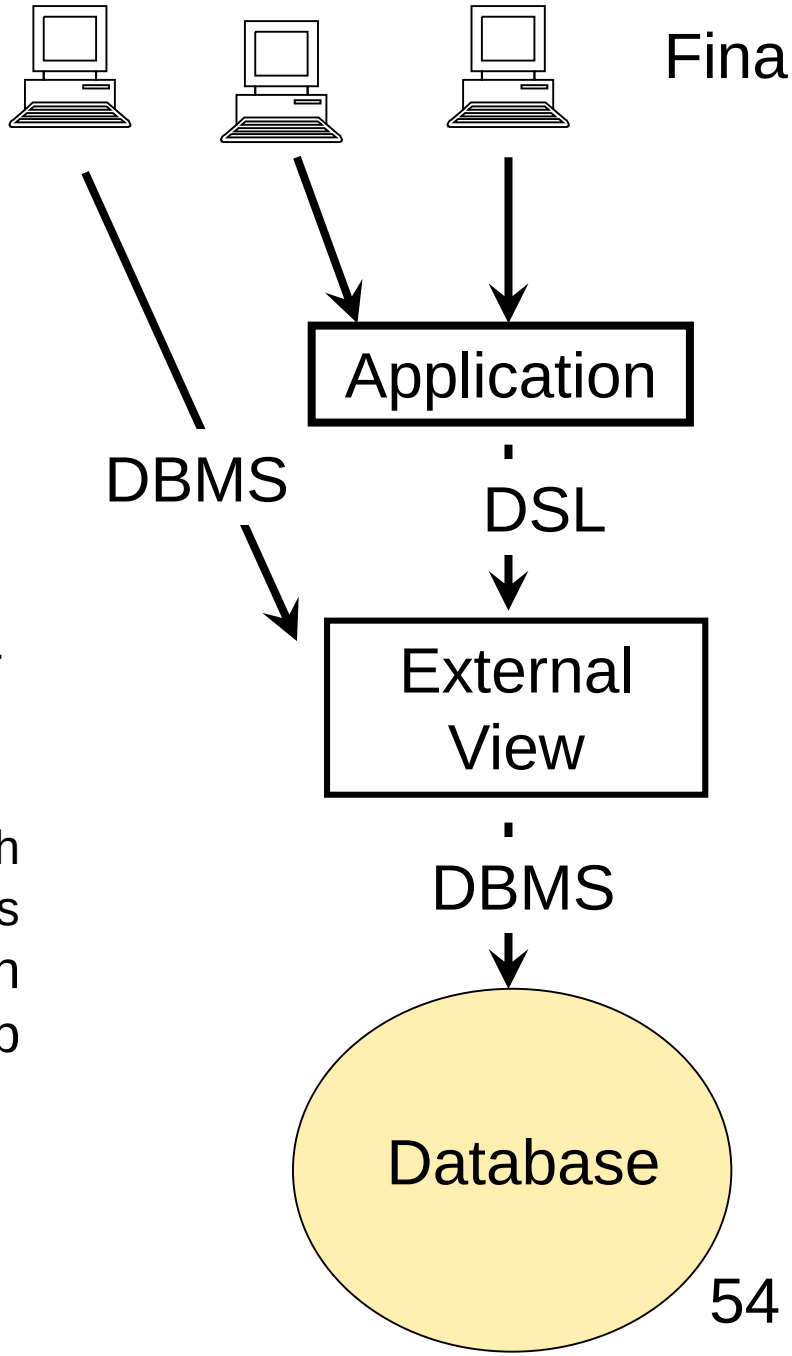


External Level

Presentation of the data to users.

Final user. Accessing via SQL or some application

Application. Developed with high-end programming languages (C++, PASCAL,…) which incorporate a Data Sub Language (DSL).



External Level

View: Content (attributes values) on the part of the DB as seen by a user/application at a given time

Name	Paid €
Pep	300
Joan	1000

DSL sub-language in the high-end language is able for interaction with the DBS: Data Definition Language, Data Manipulation Language

DNI	Name	LastName	Dep.	Paid
333	Pep	Pepito	Compres	300
444	Joan		TIC	1000

Name	Paid $
Pep	395
Joan	1.317

Conceptual Level

View: Content (attribute values) of **ALL** the **DB** at any given time

DNI	Name	LastName	Dep.	Paid
333	Pep	Pepito	Compres	300
444	Joan		TIC	1000

Defined based on DDL conceptual (script SQL: create table, create domain, create foreign key, …)

Specify integrity and security controls

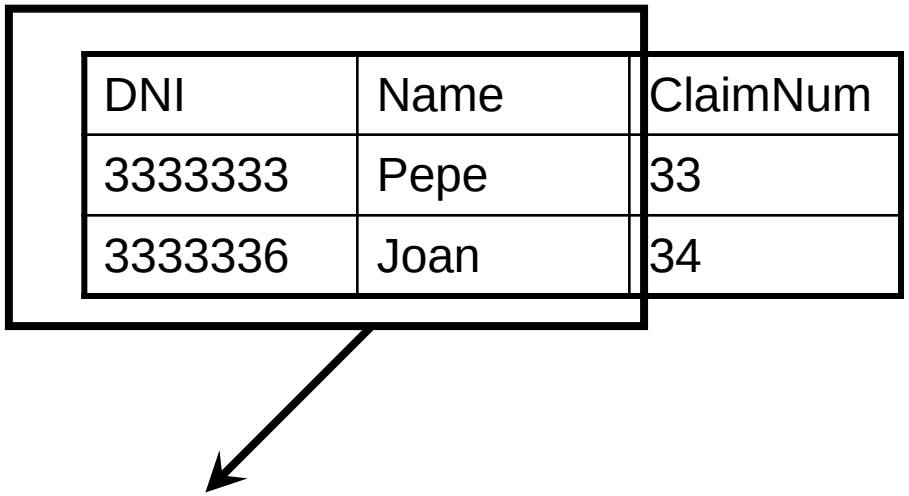
There must be a correspondence between the outer and the conceptual scheme

Internal Level

Content of the files which store the content of tables

Does not match with conceptual view (data distributed in different files)

Files description according the operating system format



Filename, table's attributes, access (indexing)

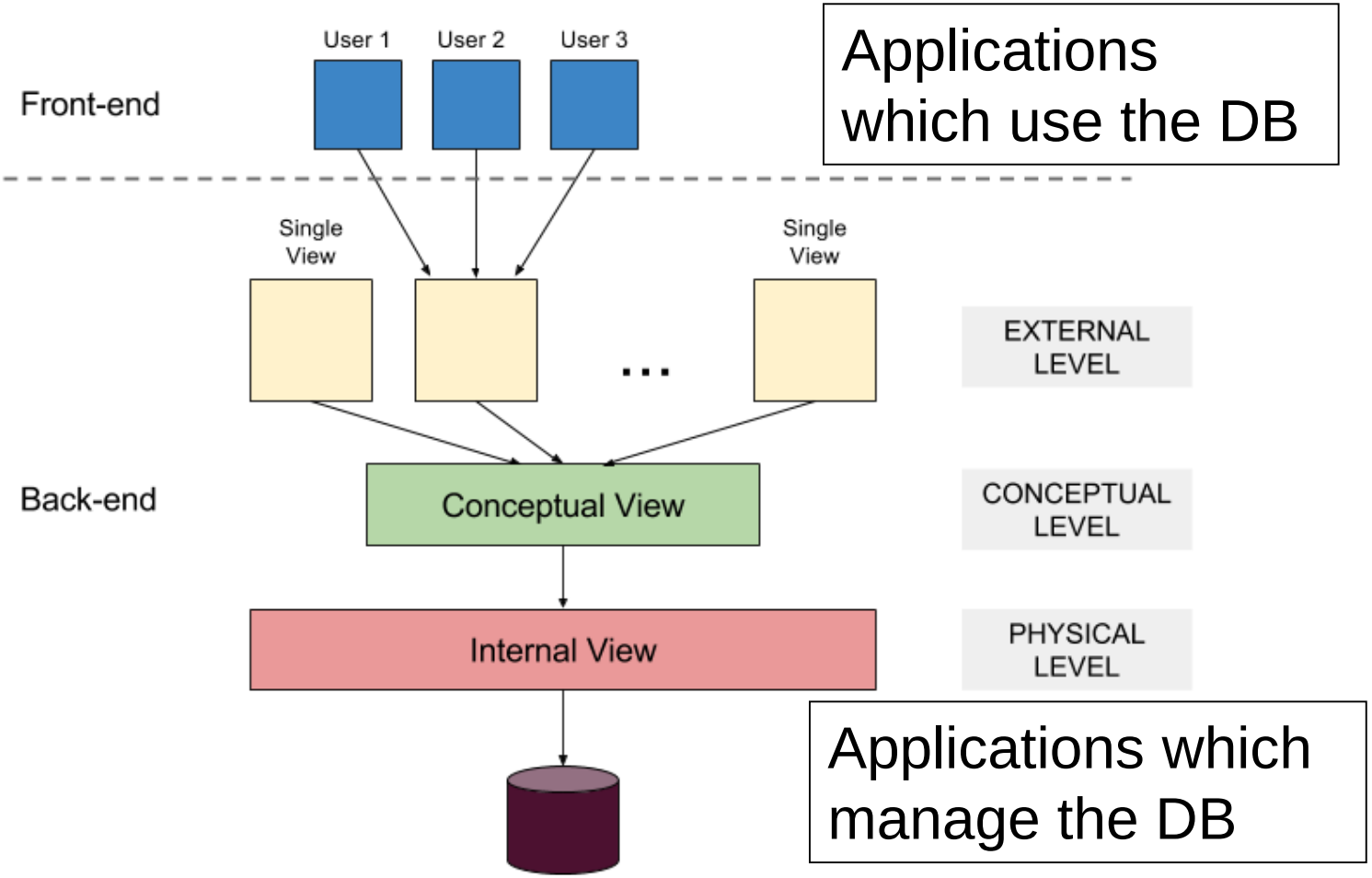
3. Client-Server

1 Back-end/Front-end

2 Distributed Systems

Front-end/Back-end

Structuring the DB according the type of application (software) that are executed on the DB in 2 levels:



Front-end / Back-end: Components

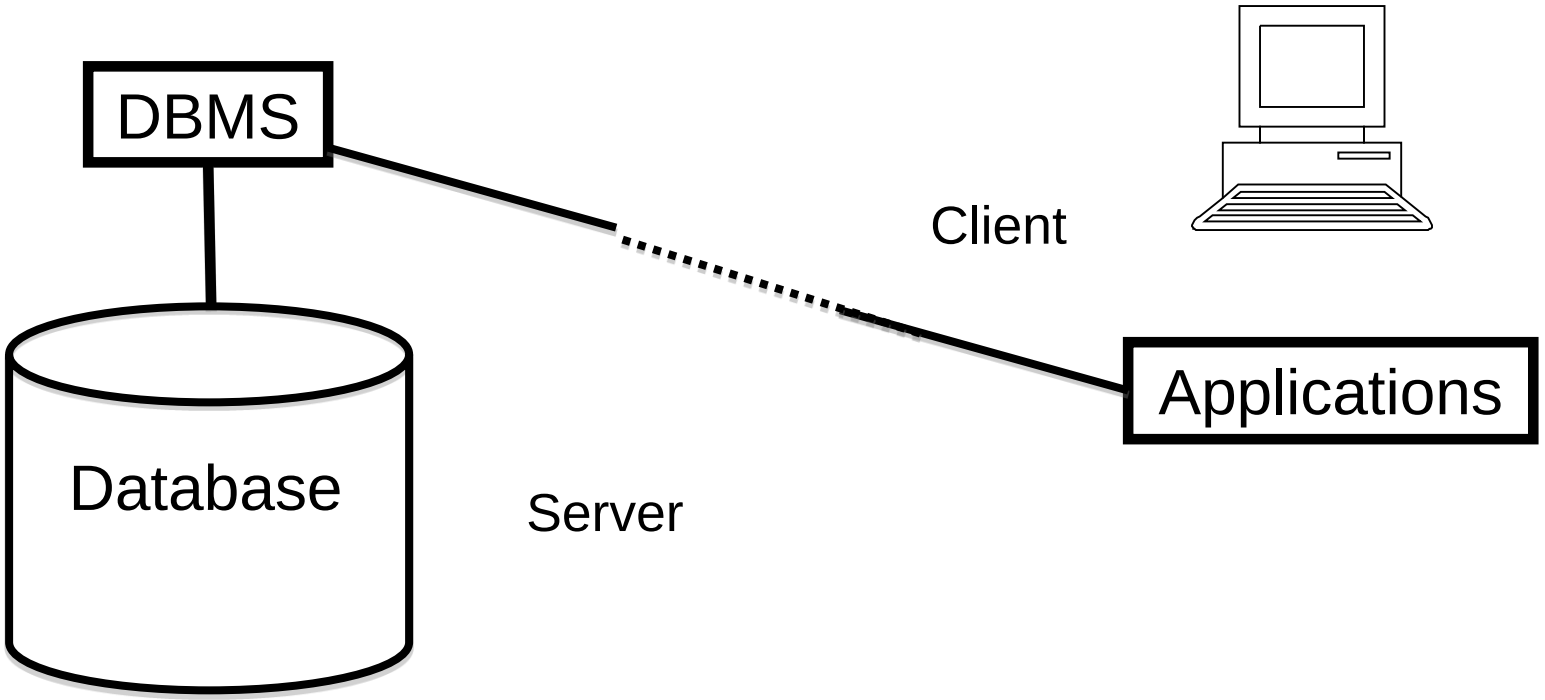
BACK-END (behind). Software which executes all functions specified in a DBMS

FRONT-END (on front). Applications executed over a DBMS (web browser, smartphone app, SQL developer, etc.)

Usually **1** BACK-END and
n FRONT-ENDs

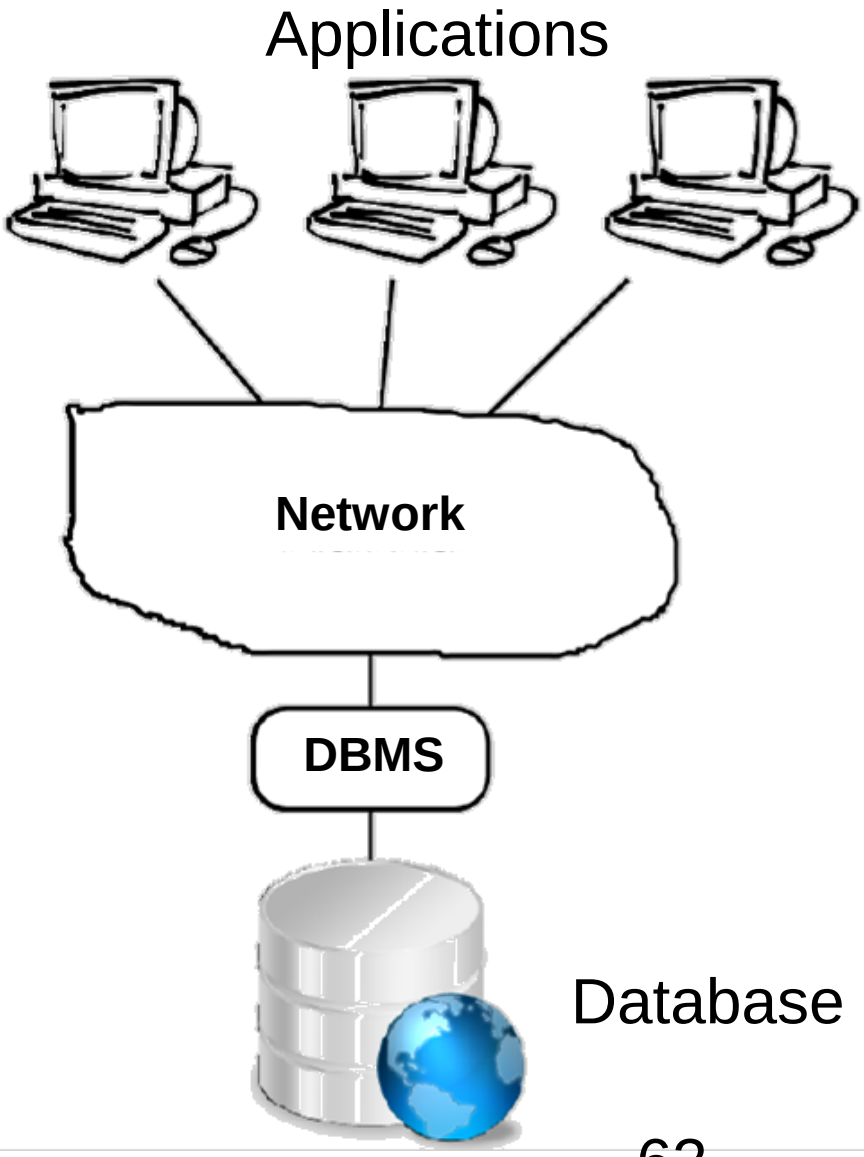
Distributed System

The separation of the DB according to the software allows each part to be in different physical nodes (or computers).



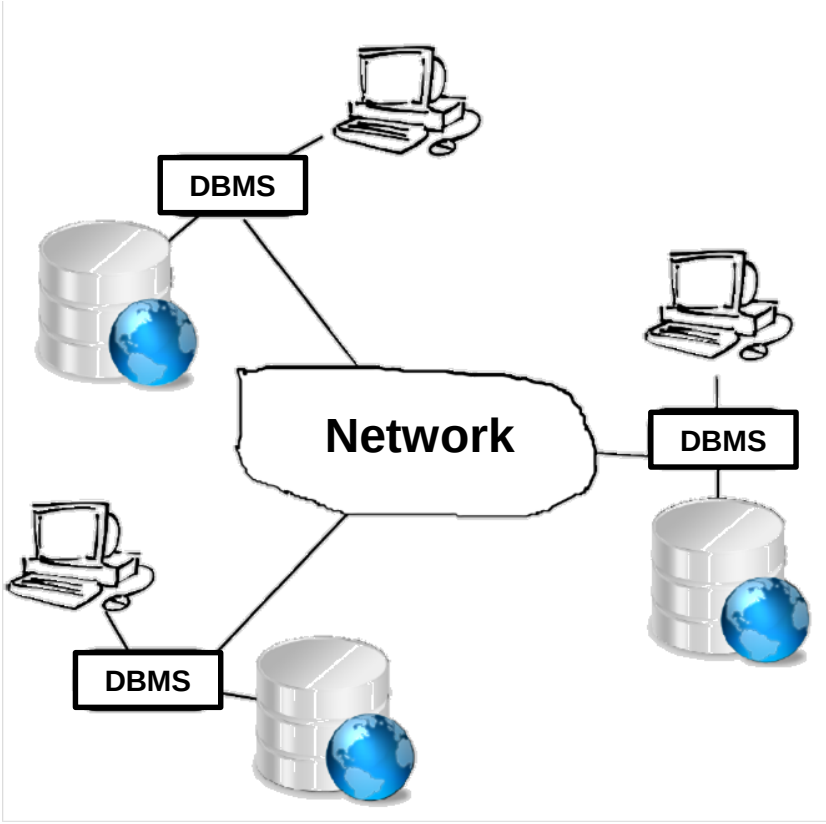
Distributed System: Client-Server

Client (FRONT-END): Machine which executes the application



Server (BACK-END): Machine which executes the DBMS.

Distributed System on Network



Distribution of data on several machines and operating systems.

N client-server locals (nodes) are communicating between them using SQL .

Each machine contains a DBS that works like:

- Server per certain users.
- Client for other users.