

RELATIONAL DATABASES TOOLS

DataBase Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.
carlos.andres.sierra.v@gmail.com

Lecturer
Computer Engineer
School of Engineering
Universidad Distrital Francisco José de Caldas

2024-I



Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



Outline

1 DataBase Management Systems — DBMS

2 UI: GUI vs. CLI

3 Agnostic: MultiDB Tools

4 DataBases in the Cloud

5 DataBases: SaaS

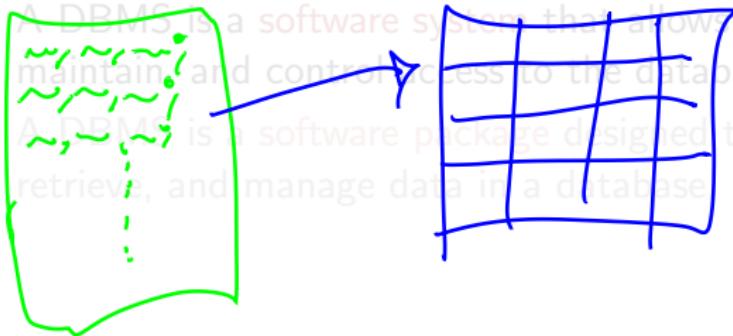
6 Local Environment

7 DevOps



What is a DBMS?

- A Database Management System (DBMS) is a software system that uses a standard method to store and organize data.
- The data can be stored in tables, which are related to each other.
- A DBMS is a software system that allows users to define, create, maintain, and control access to the database.
- A DBMS is a software package designed to define, manipulate, retrieve, and manage data in a database.



What is a DBMS?

- A Database Management System (DBMS) is a software system that uses a standard method to store and organize data.
- The data can be stored in tables, which are related to each other.
- A DBMS is a software system that allows users to define, create, maintain, and control access to the database.
- A DBMS is a software package designed to define, manipulate, retrieve, and manage data in a database.



What is a DBMS?

content
titles

comes

- A **Database Management System** (DBMS) is a software system that uses a standard method to store and organize data.
- The data can be stored in **tables**, which are related to each other.
- A **DBMS** is a **software system** that allows **users** to define, create, **maintain**, and **control access** to the database.
- A DBMS is a software package designed to define, manipulate, retrieve, and manage data.

DBA => Database Administrator



What is a DBMS?

- A **Database Management System** (DBMS) is a software system that uses a standard method to store and organize data.
- The data can be stored in **tables**, which are related to each other.
- A DBMS is a **software system** that allows users to define, create, maintain, and control access to the database.
- A **DBMS** is a **software package** designed to define, manipulate, retrieve, and manage data in a database.

Pandas



Pros & Cons of DBMS

● Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

● Cons:

• Complexity of initial setup and maintenance.

• High cost of implementation and maintenance.

• Limited flexibility in data storage and retrieval.

• Inability to handle unstructured data effectively.

• Dependence on hardware and software infrastructure.

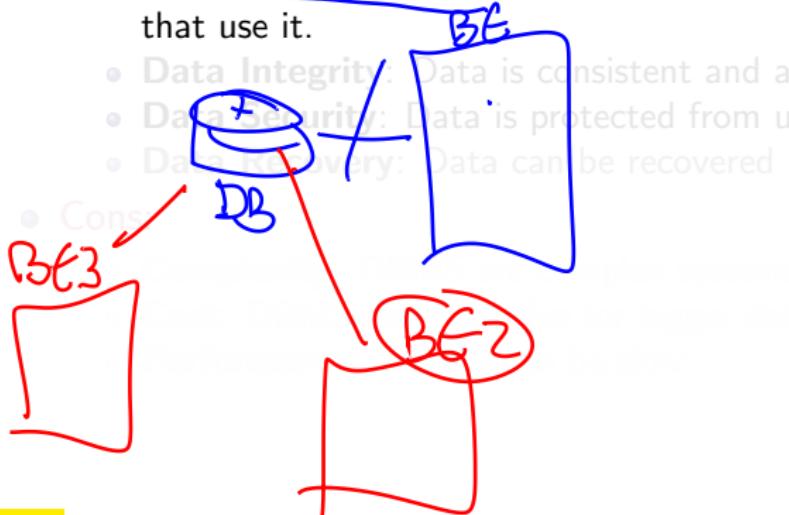
• Potential security risks associated with centralized data storage.



Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.



Pros & Cons of DBMS

- HTML Inj.
- 
- ~~SELECT * FROM Users;~~
- SQL Injection OLM
- Pros:
 - **Data Independence:** Data is stored independently of the applications that use it.
 - **Data Integrity:** Data is consistent and accurate.
 - **Data Security:** Data is protected from unauthorized access.
 - **Data Recovery:** Data can be recovered in case of failure.
 - Cons:



Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:



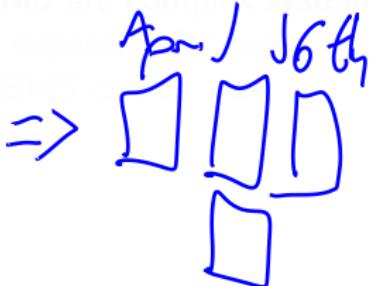
Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:

Backup



• Log
DevOps



Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:

- **Complexity:** DBMS are complex systems.
- **Cost:** DBMS are expensive for bigger data volumes.
- **Performance:** DBMS can be slow.



Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:

- **Complexity:** DBMS are complex systems.
- **Cost:** DBMS are expensive for bigger data volumes.
- **Performance:** DBMS can be slow.



Pros & Cons of DBMS

- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:

- **Complexity:** DBMS are complex systems.
- **Cost:** DBMS are expensive for bigger data volumes.
- **Performance:** DBMS can be slow.



Pros & Cons of DBMS



- Pros:

- **Data Independence:** Data is stored independently of the applications that use it.
- **Data Integrity:** Data is consistent and accurate.
- **Data Security:** Data is protected from unauthorized access.
- **Data Recovery:** Data can be recovered in case of failure.

- Cons:

- **Complexity:** DBMS are complex systems.
- **Cost:** DBMS are expensive for bigger data volumes.
- **Performance:** DBMS can be slow.

Index => Sort



Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



GUI Assistants

- A Graphical User Interface (GUI) is a type of user interface that allows users to interact with electronic devices using graphical icons and visual indicators.
- GUIs are easier to use than Command Line Interfaces (CLI) because they allow users to interact with the system using visual elements such as windows, buttons, and menus.
- GUIs are more intuitive and user-friendly than CLIs, which makes them ideal for users who are not familiar with the system.



GUI Assistants

- A **Graphical User Interface** (GUI) is a type of user interface that allows users to interact with electronic devices using graphical icons and visual indicators.
- GUIs are easier to use than **Command Line Interfaces** (CLI) because they allow users to interact with the system using visual elements such as windows, buttons, and menus.
- GUIs are more intuitive and user-friendly than CLIs, which makes them ideal for users who are not familiar with the system.



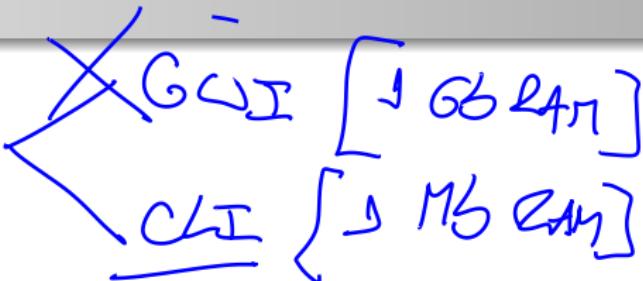
GUI Assistants

- A **Graphical User Interface** (GUI) is a type of user interface that allows users to interact with electronic devices using graphical icons and visual indicators.
- GUIs are easier to use than **Command Line Interfaces** (CLI) because they allow users to interact with the system using visual elements such as windows, buttons, and menus.
- GUIs are more intuitive and user-friendly than CLIs, which makes them ideal for users who are not familiar with the system.

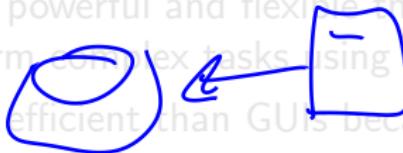


Command Line

5.0. → Kernel



- A Command Line Interface (CLI) is a type of user interface that allows users to interact with electronic devices using text-based commands.
- CLIs are more powerful and flexible than GUIs because they allow users to perform complex tasks using simple commands.
- CLIs are more efficient than GUIs because they do not require users to navigate through menus and windows to perform tasks.



Command Line

- A **Command Line Interface (CLI)** is a type of user interface that allows users to interact with electronic devices using text-based commands.
- CLIs are more powerful and flexible than GUIs because they allow users to perform complex tasks using simple commands.
- CLIs are more efficient than GUIs because they do not require users to navigate through menus and windows to perform tasks.



Command Line

- A **Command Line Interface** (CLI) is a type of user interface that allows users to interact with electronic devices using text-based commands.
- CLIs are more powerful and flexible than GUIs because they allow users to perform complex tasks using simple commands.
- CLIs are more efficient than GUIs because they do not require users to navigate through menus and windows to perform tasks.

SQL



Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



Why use an agnostic tool?

DBeaver

TablePlus

- An agnostic tool is a tool that is not tied to a specific technology or platform.
- Agnostic tools are useful because they allow users to work with multiple databases without having to learn different tools.
- Agnostic tools are also useful because they allow users to work with multiple databases without having to switch between different tools.



Why use an agnostic tool?

- An **agnostic tool** is a tool that is not tied to a specific technology or platform.
- Agnostic tools are useful because they allow users to work with multiple databases without having to learn different tools.
- Agnostic tools are also useful because they allow users to work with multiple databases without having to switch between different tools.



Why use an agnostic tool?

- An **agnostic tool** is a tool that is not tied to a specific technology or platform.
- Agnostic tools are useful because they allow users to work with multiple databases without having to learn different tools.
- Agnostic tools are also useful because they allow users to work with multiple databases without **having to switch** between different tools.



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility**: Agnostic tools can work with multiple databases.
- **Efficiency**: Agnostic tools can save time and effort.
- **Cost-Effective**: Agnostic tools can reduce costs.

- Cons:



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:

FoSS tools can be complex
Free Open Source Software



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:

- **Complexity:** Agnostic tools can be complex.
- **Performance:** Agnostic tools can be slow.
- **Compatibility:** Agnostic tools may not be compatible with all databases.



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:

- **Complexity:** Agnostic tools can be complex.
- **Performance:** Agnostic tools can be slow.
- **Compatibility:** Agnostic tools may not be compatible with all databases.



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:

- **Complexity:** Agnostic tools can be complex.
- **Performance:** Agnostic tools can be slow.
- **Compatibility:** Agnostic tools may not be compatible with all databases.



Pros & Cons of Agnostic Tools

- Pros:

- **Flexibility:** Agnostic tools can work with multiple databases.
- **Efficiency:** Agnostic tools can save time and effort.
- **Cost-Effective:** Agnostic tools can reduce costs.

- Cons:

- **Complexity:** Agnostic tools can be complex.
- **Performance:** Agnostic tools can be slow.
- **Compatibility:** Agnostic tools may not be compatible with all databases.

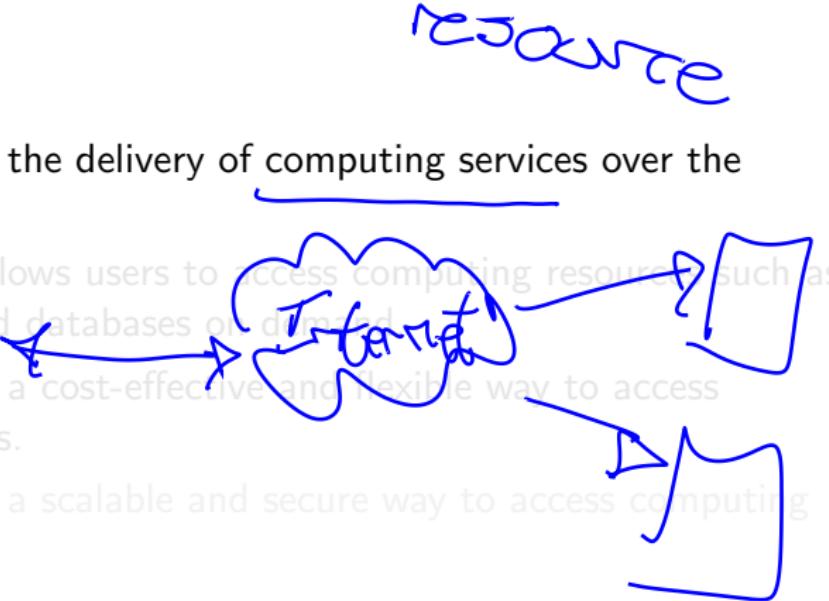


Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



What is the cloud computing?

- **Cloud computing** is the delivery of computing services over the internet.
 - Cloud computing allows users to access computing resources such as servers, storage, and databases on demand.
 - Cloud computing is a cost-effective and flexible way to access computing resources.
 - Cloud computing is a scalable and secure way to access computing resources.
- 



What is the cloud computing?

- Cloud computing is the delivery of computing services over the internet.
- Cloud computing allows users to access computing resources such as servers, storage, and databases on demand.
- Cloud computing is a cost-effective and flexible way to access computing resources.
- Cloud computing is a scalable and secure way to access computing resources.

$DB \Rightarrow$

query \Rightarrow

\$0.0001 usd

server \Rightarrow \$4 usd / hour

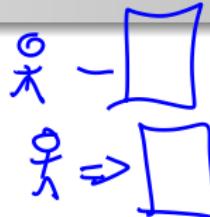
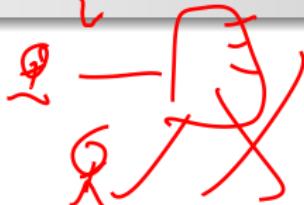


What is the cloud computing?

- **Cloud computing** is the delivery of computing services over the internet.
- Cloud computing allows users to access computing resources such as servers, storage, and databases on demand.
- Cloud computing is a **cost-effective** and **flexible** way to access computing resources.
- Cloud computing is a scalable and secure way to access computing resources.



What is the cloud computing?



- **Cloud computing** is the delivery of computing services over the internet.
- Cloud computing allows users to access computing resources such as servers, storage, and databases on demand.
- Cloud computing is a cost-effective and flexible way to access computing resources.
- **Cloud computing** is a **scalable and secure way** to access computing resources.



Cloud Computing Vs. On-Premises

- **Cloud computing** is the delivery of computing services over the internet, while **on-premises** computing is the delivery of computing services on the premises of an organization.
- Cloud computing is a cost-effective and flexible way to access computing resources, while on-premises computing is a more traditional and expensive way to access computing resources.
- Cloud computing is a more scalable than on-premises computing, while on-premises computing is expensive to make scalable.

Internet Internet



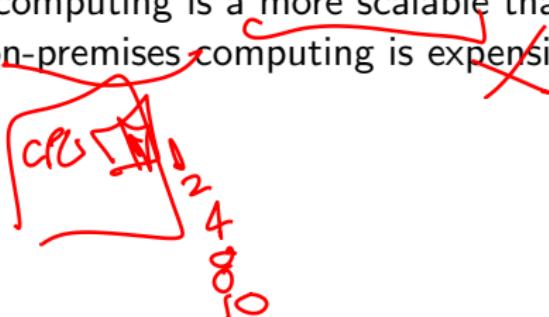
Cloud Computing Vs. On-Premises

- Cloud computing is the delivery of computing services over the internet, while on-premises computing is the delivery of computing services on the premises of an organization.
- Cloud computing is a cost-effective and flexible way to access computing resources, while on-premises computing is a more traditional and expensive way to access computing resources.
- Cloud computing is a more scalable than on-premises computing, while on-premises computing is expensive to make scalable.



Cloud Computing Vs. On-Premises

- **Cloud computing** is the delivery of computing services over the internet, while **on-premises** computing is the delivery of computing services on the premises of an organization.
- Cloud computing is a cost-effective and flexible way to access computing resources, while on-premises computing is a more traditional and expensive way to access computing resources.
- Cloud computing is a more scalable than on-premises computing, while on-premises computing is expensive to make scalable.



Pros & Cons of Cloud Computing

- **Pros:**

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- **Cons:**

- **Dependency on the cloud provider:** Cloud computing requires a dependency on the cloud provider's infrastructure, which can lead to potential downtime or service interruptions if the provider experiences issues.
- **Data security concerns:** Storing data in the cloud can raise concerns about data security, particularly if the provider does not have strong security measures in place to protect sensitive information.
- **Cost fluctuations:** While cloud computing is generally cost-effective, there can be fluctuations in costs depending on the provider's pricing models and usage patterns.
- **Loss of control:** By using cloud computing, users may lose some level of control over their data and how it is stored and managed.



Pros & Cons of Cloud Computing

• Pros:

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

• Cons:

- **Dependency on Internet:** Cloud computing requires a stable internet connection to access resources.
- **Data Privacy:** Storing data in the cloud can raise concerns about data privacy and security.
- **Cost:** While it can be cost-effective, the initial setup and ongoing costs of using cloud services can be significant.
- **Loss of Control:** You may lose some control over your data and infrastructure when using a public cloud provider.
- **Service Disruption:** If the cloud provider experiences downtime or technical issues, it can affect your access to services.



Pros & Cons of Cloud Computing

- Pros:

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- Cons:



Pros & Cons of Cloud Computing

- Pros:

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- Cons:

- Security: Cloud computing can be less secure than on-premises computing.



Pros & Cons of Cloud Computing

- **Pros:**

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- **Cons:**

- **Security:** Cloud computing can be less secure than on-premises computing.
- **Performance:** Cloud computing can be slower than on-premises computing.
- **Reliability:** Cloud computing can be less reliable than on-premises computing.



Pros & Cons of Cloud Computing

- Pros:

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- Cons:

- **Security:** Cloud computing can be less secure than on-premises computing.
- **Performance:** Cloud computing can be slower than on-premises computing.
- **Reliability:** Cloud computing can be less reliable than on-premises computing.



Pros & Cons of Cloud Computing

- Pros:

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- Cons:

- **Security:** Cloud computing can be less secure than on-premises computing.
- **Performance:** Cloud computing can be slower than on-premises computing.
- **Reliability:** Cloud computing can be less reliable than on-premises computing.



Pros & Cons of Cloud Computing

- **Pros:**

- **Cost-Effective:** Cloud computing is a cost-effective way to access computing resources.
- **Scalable:** Cloud computing is a scalable way to access computing resources.
- **Flexible:** Cloud computing is a flexible way to access computing resources.

- **Cons:**

- **Security:** Cloud computing can be less secure than on-premises computing.
- **Performance:** Cloud computing can be slower than on-premises computing.
- **Reliability:** Cloud computing can be less reliable than on-premises computing.



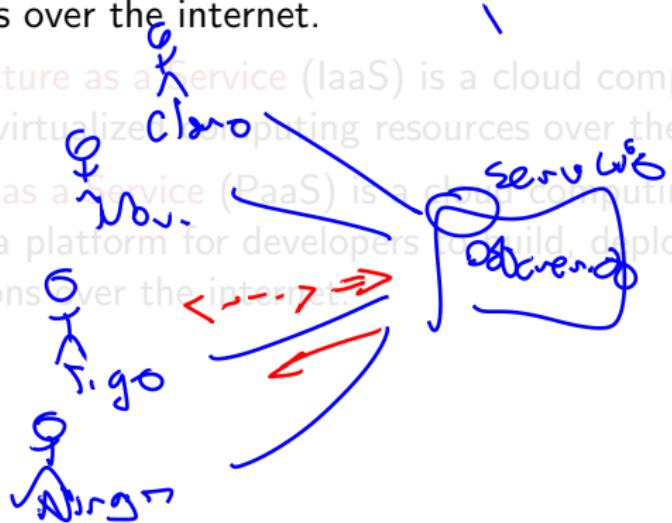
Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



SaaS Vs. IaaS Vs. PaaS

- **Software as a Service (SaaS)** is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the internet.
- **Infrastructure as a Service (IaaS)** is a cloud computing model that provides virtualized computing resources over the internet.
- **Platform as a Service (PaaS)** is a cloud computing model that provides a platform for developers to build, deploy, and manage applications over the internet.



SaaS Vs. IaaS Vs. PaaS

- **Software as a Service** (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the internet.
- **Infrastructure as a Service** (IaaS) is a cloud computing model that provides virtualized computing resources over the internet.
- **Platform as a Service** (PaaS) is a cloud computing model that provides a platform for developers to build, deploy, and manage applications over the internet.
*(Servers
Storage)*



SaaS Vs. IaaS Vs. PaaS

- **Software as a Service (SaaS)** is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the internet.
- **Infrastructure as a Service (IaaS)** is a cloud computing model that provides virtualized computing resources over the internet.
- **Platform as a Service (PaaS)** is a cloud computing model that provides a platform for developers to ~~build, deploy, and manage~~ applications over the internet.



DataBases as a Service

- Database as a Service (DBaaS) is a cloud computing model that provides database services over the internet.
- DBaaS is a cost-effective and flexible way to access database services. *Firestore*
- DBaaS is a reliable and secure way to access database services. *BigQuery*
- DBaaS is a reliable and efficient way to access database services.

At home => SQL Server

AWS => Relational Database



DataBases as a Service

- Database as a Service (DBaaS) is a cloud computing model that provides database services over the internet.
- DBaaS is a cost-effective and flexible way to access database services.
- DBaaS is a scalable and secure way to access database services.
- DBaaS is a reliable and efficient way to access database services.



DataBases as a Service

- **Database as a Service** (DBaaS) is a cloud computing model that provides database services over the internet.
- **DBaaS** is a cost-effective and flexible way to access database services.
- **DBaaS** is a scalable and secure way to access database services.
- **DBaaS** is a reliable and efficient way to access database services.



DataBases as a Service

- **Database as a Service** (DBaaS) is a cloud computing model that provides database services over the internet.
- **DBaaS** is a cost-effective and flexible way to access database services.
- **DBaaS** is a scalable and secure way to access database services.
- **DBaaS** is a reliable and efficient way to access database services.



Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps



localhost

Apache Server

Tomcat
Parquet / Iceberg
NetBeans

- **localhost** is a hostname that refers to the local computer that a program is running on.

- **localhost** is used to access the services that are running on the local computer.

- **localhost** is used to access the services that are running on the local computer.



Stream => Kafka

Pipelines => Airflow

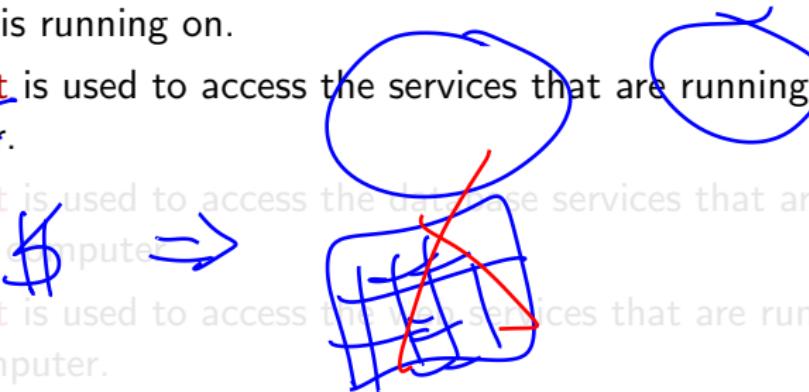
cache => Redis

queue => RabbitMQ



localhost

- **localhost** is a hostname that refers to the local computer that a program is running on.
- **localhost** is used to access the services that are running on the local computer.
- **localhost** is used to access the database services that are running on the local computer.
- **localhost** is used to access the Web services that are running on the local computer.



Localhost

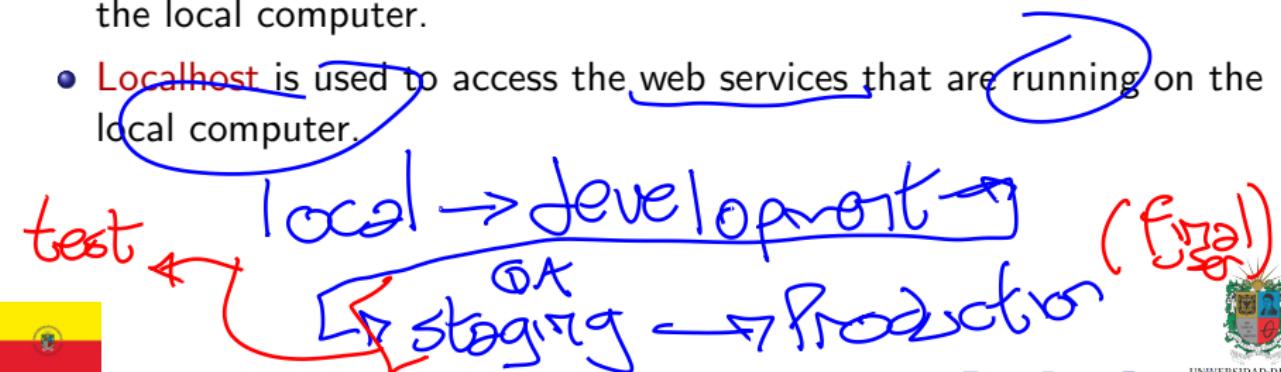
- **Localhost** is a hostname that refers to the local computer that a program is running on.
- **Localhost** is used to access the services that are running on the local computer.
- **Localhost** is used to access the database services that are running on the local computer.
- **Localhost** is used to access the web services that are running on the local computer.



Localhost



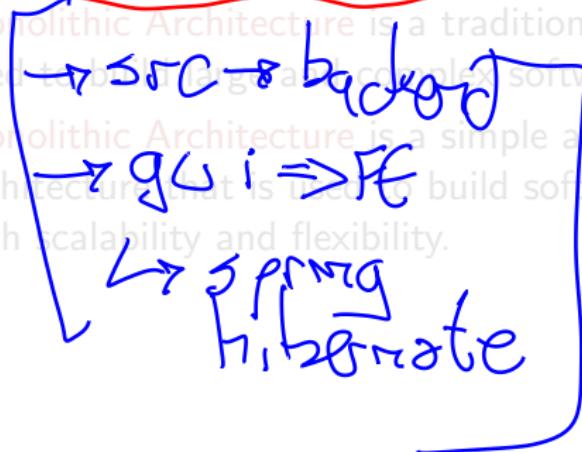
- **localhost** is a hostname that refers to the local computer that a program is running on.
- **localhost** is used to access the services that are running on the local computer.
- **localhost** is used to access the database services that are running on the local computer.
- **localhost** is used to access the web services that are running on the local computer.



Monolithic Architecture

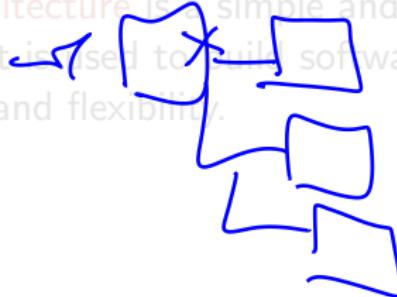


- **Monolithic Architecture** is a software architecture in which all the components of the software are combined into a single program.
- Monolithic Architecture is a traditional software architecture that is used to build large and complex software systems.
- Monolithic Architecture is a simple and easy-to-understand software architecture that is used to build software systems that do not require high scalability and flexibility.



Monolithic Architecture

- Monolithic Architecture is a software architecture in which all the components of the software are combined into a single program.
- Monolithic Architecture is a traditional software architecture that is used to build large and complex software systems.
- Monolithic Architecture is a simple and easy-to-understand software architecture that is used to build software systems that do not require high scalability and flexibility.



Monolithic Architecture

- Monolithic Architecture is a software architecture in which all the components of the software are combined into a single program.
- Monolithic Architecture is a traditional software architecture that is used to build large and complex software systems.
- Monolithic Architecture is a simple and easy-to-understand software architecture that is used to build software systems that do not require high scalability and flexibility.

POC \Rightarrow Proof-of-Concept



Outline

- 1 DataBase Management Systems — DBMS
- 2 UI: GUI vs. CLI
- 3 Agnostic: MultiDB Tools
- 4 DataBases in the Cloud
- 5 DataBases: SaaS
- 6 Local Environment
- 7 DevOps

Developnot Operations



Continuous Integration

$(R \subset A, P) \Rightarrow IBM$

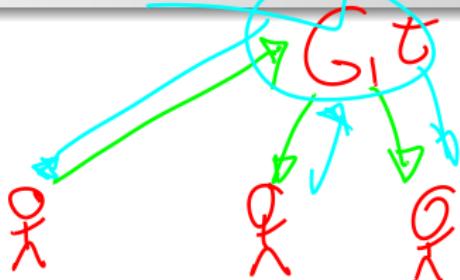
Agile Methodologies \Rightarrow standup (2-5 persons)

SCRUM
↳ sprint (time window) \Rightarrow 2 weeks

- Continuous Integration is a software development practice in which developers integrate code into a shared repository frequently.



Continuous Integration



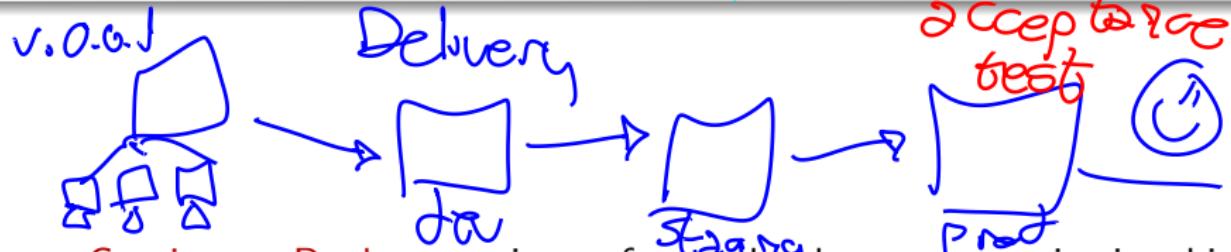
Push (git push)
 ↪ each developer
 ↓ daily

- **Continuous Integration** is a software development practice in which developers integrate code into a shared repository frequently.
- **Continuous Integration** is a software development practice that helps developers to detect and fix integration errors early.

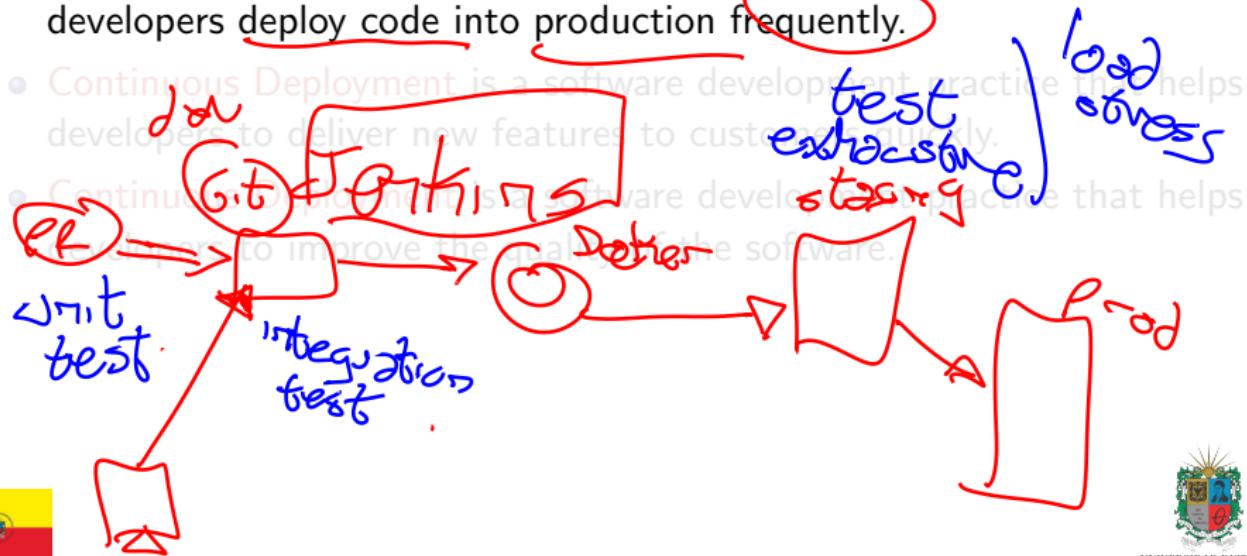


Continuous Deployment

CI/CD



- **Continuous Deployment** is a software development practice in which developers deploy code into production frequently.



Continuous Deployment

HOT FIX

- Continuous Deployment is a software development practice in which developers deploy code into production frequently.
- Continuous Deployment is a software development practice that helps developers to deliver new features to customers quickly.
- Continuous Deployment is a software development practice that helps developers to improve the quality of the software.

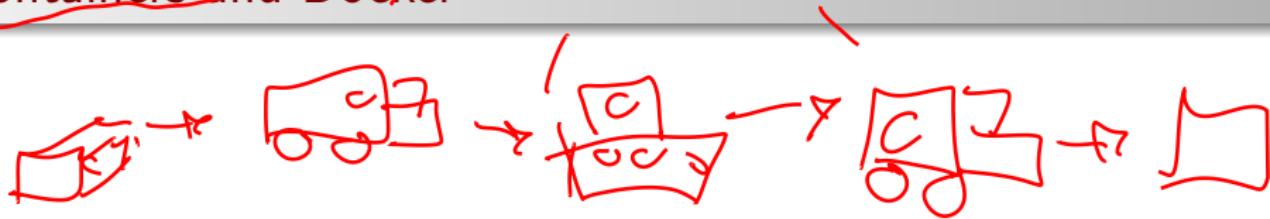


Continuous Deployment

- **Continuous Deployment** is a software development practice in which developers deploy code into production frequently.
- **Continuous Deployment** is a software development practice that helps developers to deliver new features to customers quickly.
- **Continuous Deployment** is a software development practice that helps developers to improve the quality of the software.



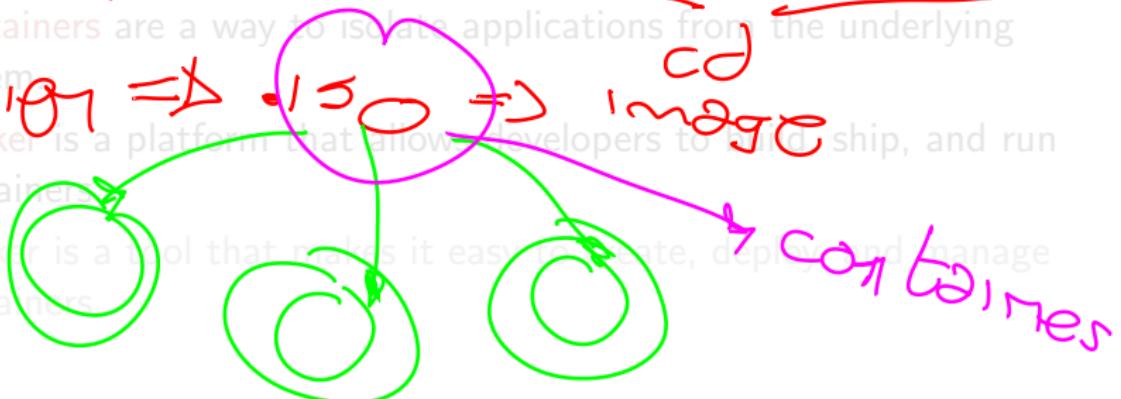
~~Containers and Docker~~



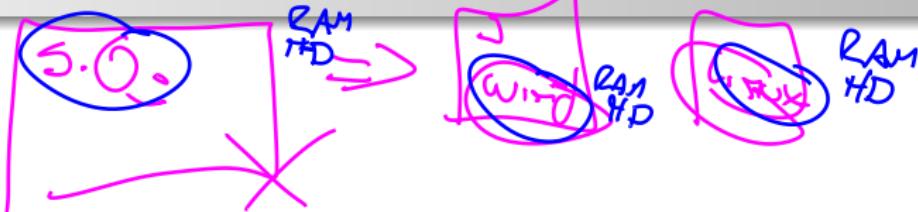
- Containers are a lightweight and portable way to package software.
 - Containers are a way to isolate applications from the underlying system.
 - Docker is a platform that allows developers to build, ship, and run containers.
 - Docker is a tool that makes it easy to create, deploy, and manage containers.

Algori => IS => Image

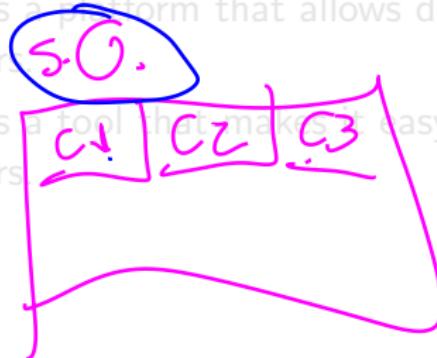
```
graph LR; A((Algori)) --> B((IS)); B --> C((Image)); C --> D((Container)); C --> E((Container))
```



Containers and Docker



- Containers are a lightweight and portable way to package software.
- Containers are a way to isolate applications from the underlying system.
- Docker is a platform that allows developers to build, ship, and run containerized applications.
- Docker is a tool that makes it easy to create, deploy, and manage containers.



Containers and Docker

- **Containers** are a lightweight and portable way to package software.
- **Containers** are a way to isolate applications from the underlying system.
- **Docker** is a platform that allows developers to build, ship, and run containers.
- **Docker** is a tool that makes it easy to create, deploy, and manage containers.



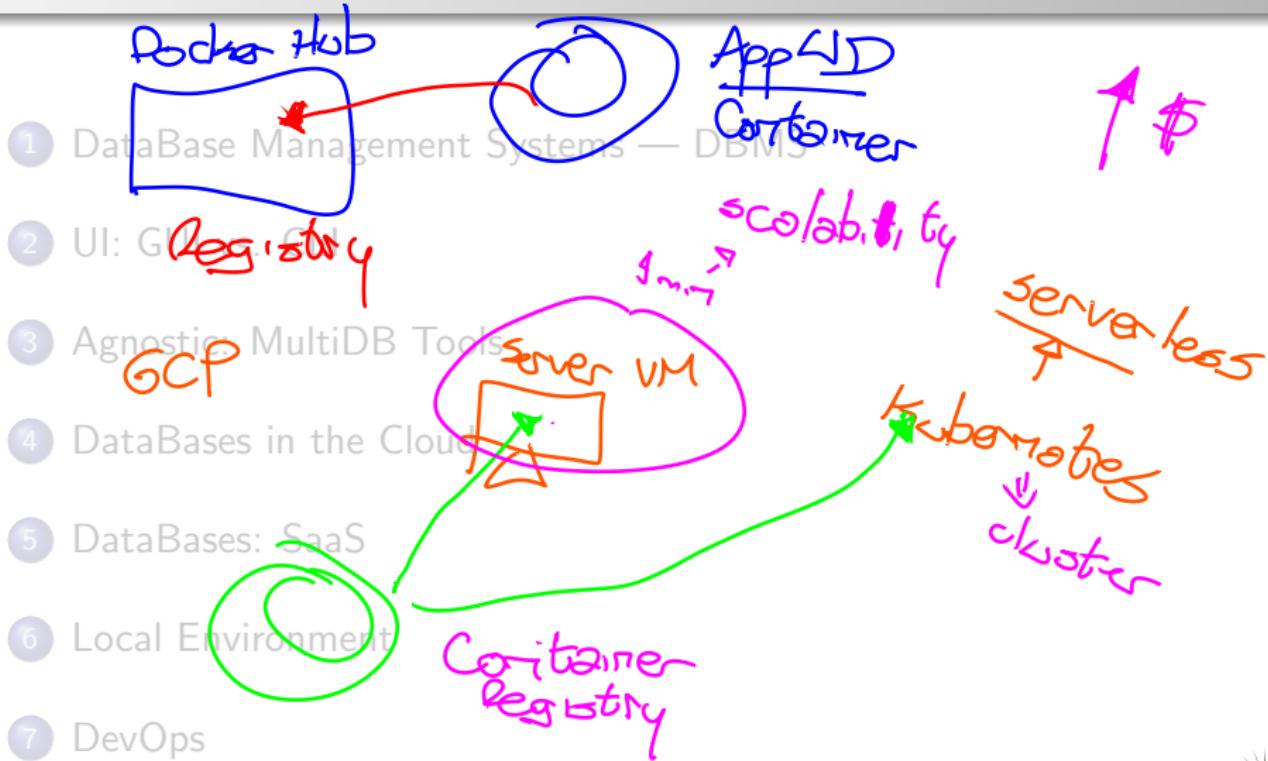
Containers and Docker

- **Containers** are a lightweight and portable way to package software.
- **Containers** are a way to isolate applications from the underlying system.
- **Docker** is a platform that allows developers to build, ship, and run containers.
- **Docker** is a tool that makes it easy to **create, deploy** and manage containers.

*docker cli
docker-compose*



Outline



Thanks!

Questions?



Repo:

 github.com/EngAndres/ud-public/tree/main/courses/databases-foundations