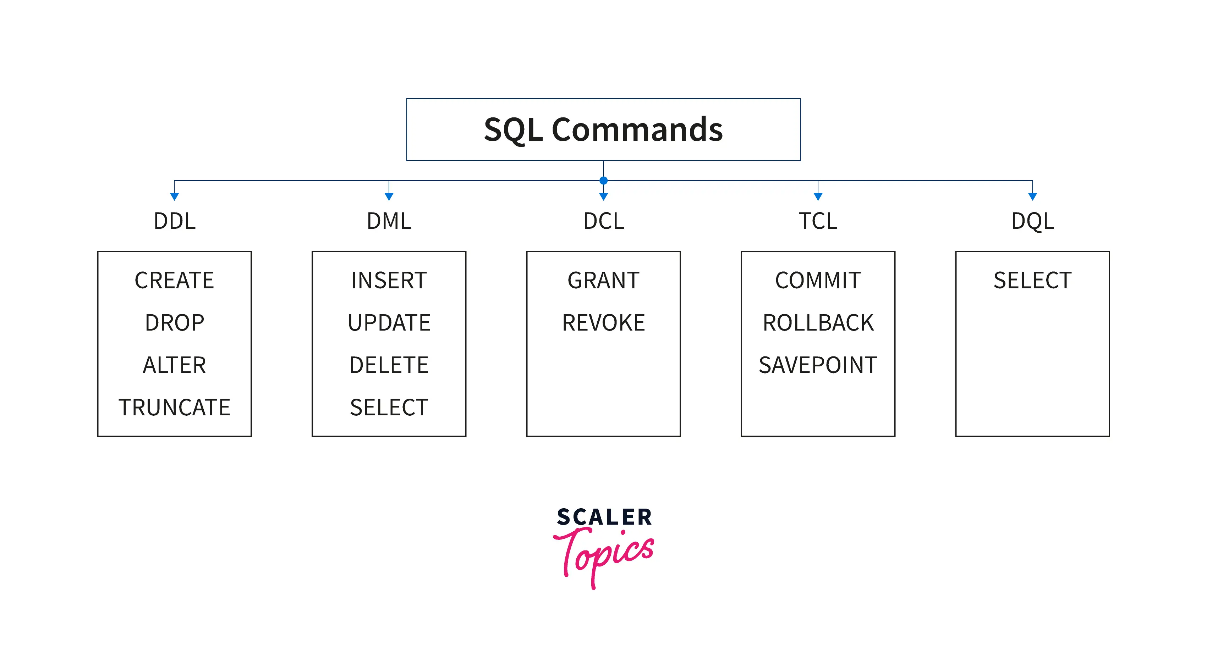
Commands :



**JOINS :**

**1.INNER JOIN**

**2.OUTER JOIN**

**3.RIGHT JOIN**

**4.LEFT JOIN**

**5.SELF JOIN**

CREATE TABLE Commerce (

order\_id int,

customer\_id int,

city varchar(20),

order\_dt date,

status varchar(15),

amount int

);

-- insert

INSERT INTO Commerce(order\_id,customer\_id,city,order\_dt,status,amount) VALUES (1, 101,'Pune','2025-08-01','PAID',1200);

INSERT INTO Commerce (order\_id,customer\_id,city,order\_dt,status,amount) VALUES (2, 102,'Pune','2025-08-01','PAID',1500);

INSERT INTO Commerce(order\_id,customer\_id,city,order\_dt,status,amount) VALUES (3, 102,'hyd','2025-08-01','CANCELLED',800);

SELECT \* FROM Commerce;

SELECT avg(amount) AS avg\_amt FROM Commerce;

SELECT SUM(amount) AS SMT FROM Commerce;

SELECT city,SUM(amount) AS revenue FROM Commerce WHERE status='PAID' GROUP BY city HAVING SUM(amount)>2500;

SELECT order\_id,city,amount FROM Commerce ORDER BY amount DESC LIMIT 3;

SELECT customer\_id FROM Commerce

WHERE amount between 1000 and 1500;

SELECT status FROM Commerce WHERE city IN ('hyd');

SELECT \* FROM Commerce WHERE city LIKE '%d';

CREATE TABLE Customer (

customer\_id int,

id\_name varchar(20),

mailid varchar(200)

);

INSERT INTO Customer(customer\_id,id\_name,mailid) VALUES (1, 'Bunny','bunny@gmail.com');

INSERT INTO Customer (customer\_id,id\_name,mailid) VALUES (2, 'Ruchitha','ruchitha@gmail.com');

INSERT INTO Customer(customer\_id,id\_name,mailid) VALUES (3, 'Pandu','pandu@gmail.com');

SELECT \* FROM Customer;

SELECT c.id\_name,o.order\_id

FROM Customer c

CROSS JOIN Commerce o;

**INDEX :**

**WITHOUT INDEX**

Select \* FROM emoployees WHERE name=’SARAH’;

**CREATING AN INDEX**

CREATE INDEX idx\_name ON employees(name);

SELECT \* FROM employees WHERE name=’SARAH’;

**DECLARE**

**a1 number;**

**b number;**

**c1 number;**

**BEGIN**

**a1 := :a1;**

**b :=:b;**

**IF a1 > b THEN**

**c1:=a1;**

**ELSE**

**c1:=b;**

**END IF ;**

**dbms\_output.put\_line('Maximum number' ||c1);**

**END**

**/**

**BY USING CONSTANT :**

School\_name constant varchar(20) := “NRHS”;

dbms\_output.put\_line(‘I study in ' || school\_name);

**ARITHMETIC VALUES :**

DECLARE

a1 number;

b number;

c1 number;

d number;

e1 number;

f1 number;

g1 number;

BEGIN

a1 := 10;

b := 5;

c1:=a1+b;

d:=a1-b;

e1:=a1\*b;

f1:=a1/b;

g1:=MOD(a1,b);

dbms\_output.put\_line('add' ||c1);

dbms\_output.put\_line('sub' ||d);

dbms\_output.put\_line('mul' ||e1);

dbms\_output.put\_line('div' ||f1);

dbms\_output.put\_line('fdiv' ||g1);

END

/

**EVEN OR ODD :**

DECLARE

a1 number;

BEGIN

a1 := 11;

IF MOD(a1,2)=0 THEN

dbms\_output.put\_line(a1|| 'is even');

ELSE

dbms\_output.put\_line(a1|| 'is odd');

END IF;

END

/

**GREATER VALUE :**

DECLARE

a1 int;

b int;

BEGIN

a1:= :a1;

b:= :b;

IF(a1>b) THEN

dbms\_output.put\_line('a1 is greater than b');

elsif(b>a1) then

dbms\_output.put\_line('b is greater than a1');

ELSE

dbms\_output.put\_line('a1 and b both are equal');

END IF;

END;

/

**USING SWITCH CASE :**

DECLARE

a1 int;

b int;

BEGIN

a1:= :a1;

b:= mod(a1,2);

case b

when 0 then dbms\_output.put\_line('Even number');

when 1 then dbms\_output.put\_line('Odd number');

else dbms\_output.put\_line('User did not give any input');

END case;

END;

/

**EXAMPLE FOR SWITCH CASE :**

DECLARE

c\_grade CHAR(1):=:c\_grade;

c\_rank varchar(29);

BEGIN

case c\_grade

when 'A' then c\_rank :='Excellent';

when 'B' then c\_rank :='GOOD';

END case;

dbms\_output.put\_line(c\_grade);

dbms\_output.put\_line(c\_rank);

END;

/

**LOOP :**

DECLARE

i int;

BEGIN

i := 1;

LOOP

if i>10 then

exit;

END IF;

dbms\_output.put\_line(i);

i:= i+1;

END LOOP;

END

/

LOOP :

DECLARE

i int;

b int;

BEGIN

i := :i;

b :=1;

LOOP

if b>10 then

exit;

END IF;

dbms\_output.put\_line(i||'\*'||b||'='||(i\*b));

b:=b+1;

END LOOP;

END;

/

**EXMP FOR WHILE LOOP :**

DECLARE

num int:=1;

BEGIN

while(num<=10) LOOP

dbms\_output.put\_line(''|| num);

num :=num+2;

END LOOP;

END;

/

**FOR LOOP:**

DECLARE

i number;

BEGIN

FOR i IN 1..10 LOOP

dbms\_output.put\_line(i);

END LOOP;

END;

/

**EVENODD FUNCTION :**

DECLARE

a1 number;

c1 number;

FUNCTION EvenOdd(x IN number)

RETURN number

IS

z number;

BEGIN

if MOD(x,2)=0 THEN

dbms\_output.put\_line('Even ' ||x);

ELSE

dbms\_output.put\_line('odd ' ||x);

END IF;

RETURN z;

END;

BEGIN

a1:=10;

c1:=EvenOdd(a1);

END;

/

**CURSOR**

**IMPLICIT** INSERT UPDATE DELETE

**EXPLICIT** SELECT

**TO INSERT**

INSERT INTO Student(rollno,sname,course) VALUES (2,'Bunny','It');

INSERT INTO Student(rollno,sname,course) VALUES (3,'pandu','CSM');

**CURSOR**

declare

d number;

cursor showRec(sno student.rollno%type) IS

SELECT sname,course FROM student WHERE rollno=sno;

a1 student.sname%type;

b student.course%type;

c1 student.rollno%type;

BEGIN

d:=:rollno;

OPEN showRec(d);

IF showRec%ISopen = False then

dbms\_output.put\_line('Cannot open cursor');

ELSE

LOOP

FETCH ShowRec into a1,b;

EXIT WHEN showRec%NOTFOUND;

dbms\_output.put\_line(a1|| ' ' ||b);

END LOOP;

END IF;

CLOSE showRec;

END;

/

**CREATE AND INSERT**

create table customer(id int not null,name varchar2(40) NULL,Age int Not Null,address varchar2(40) nULL, salary int null,primary key(id));

INSERT INTO customer(id,name,Age,address,salary) values (1,'ruchitha',21,'mbnr',40000);

INSERT INTO customer(id,name,Age,address,salary) values (2,'bunny',22,'mbnr1',50000);

INSERT INTO customer(id,name,Age,address,salary) values (3,'pandu',23,'mbnr2',60000);

INSERT INTO customer(id,name,Age,address,salary) values (4,'manasa',24,'mbnr3',70000);

INSERT INTO customer(id,name,Age,address,salary) values (5,'sudhakar',25,'mbnr4',80000);

INSERT INTO customer(id,name,Age,address,salary) values (6,'tanvi',26,'mbnr5',90000);

INSERT INTO customer(id,name,Age,address,salary) values (7,'purvi',27,'mbnr6',100000);

INSERT INTO customer(id,name,Age,address,salary) values (8,'sony',28,'mbnr7',10000);

**FOR UPDATE IN A TABLE :**

UPDATE customer

set salary = 50000 where id =8;

**TRIGGER CREATED :**

CREATE OR Replace TRIGGER display\_salary\_changes

BEFORE DELETE OR INSERT OR UPDATE ON customer

for each row

when (new.id > 0)

declare

sal\_diff number;

begin

sal\_diff := :NEW.salary - :OLD.salary;

dbms\_output.put\_line('Old salary: ' || :OLD.salary);

dbms\_output.put\_line('New salary: ' || :NEW.salary);

dbms\_output.put\_line('Salary difference: ' || sal\_diff);

END;

/

create or replace package c\_package AS

procedure addcustomer(c\_id customer.id%type,

c\_name customer.name%type,

c\_age customer.Age%type,

c\_addr customer.address%type,

c\_sal customer.salary%type);

procedure delcustomer(c\_id customer.id%type);

procedure listcustomer;

end c\_package;

create or replace package c\_package AS

procedure addcustomer(c\_id customer.id%type,

c\_name customer.name%type,

c\_age customer.Age%type,

c\_addr customer.address%type,

c\_sal customer.salary%type);

is

begin

insert into customer (id,name,address,salary) values(c\_id,c\_name,c\_age,c\_sal);

end add customer

procedure delcustomer(c\_id customer.id%type) IS;

begin

delete from customer

where id=c\_id;

end delcustomer;

end c\_package;

**create or replace package body c\_package AS**

**procedure addcustomer(c\_id customer.id%type,**

**c\_name customer.name%type,**

**c\_age customer.Age%type,**

**c\_addr customer.address%type,**

**c\_sal customer.salary%type)**

**is**

**begin**

**insert into customer (id,name,address,salary) values(c\_id,c\_name,c\_age,c\_sal);**

**end addcustomer;**

**procedure delcustomer(c\_id customer.id%type) IS;**

**begin**

**delete from customer**

**where id=c\_id;**

**end delcustomer;**

**end c\_package;**

**/**

**Agile :**

Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints.

The Agile framework is an [iterative methodology](https://asana.com/resources/iterative-process). After every sprint, teams reflect and look back to see if there was anything that could be improved so they can adjust their strategy for the next sprint.



**1.Scrum**

**2.Product owner**

**Scrum :**

Scrum is an agile framework used for managing work and developing products, particularly in software development. It's designed to help teams self-organize, adapt to change, and deliver value in short, iterative cycles called [sprints](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&biw=1280&bih=665&q=sprints&sa=X&ved=2ahUKEwiqsvOUprmPAxXESGcHHYjUIuEQxccNegQIKhAB&mstk=AUtExfBWGHHXJYFQFFTlo6e-QvG_jYnQfnzmaMgaHkyAhDu1nK8wH-Yez9S-xVCM510SyzqApeKFclxapw5fHrmFluPX7tipVeXwKSktZ7O2hJFIbmHm5CCCz7gWAbDoAeaxAWU&csui=3). Think of it as a structured approach to teamwork that emphasizes collaboration, flexibility, and continuous improvement.

**Product Owner :**

A Product Owner (PO) is an individual on an agile development team, particularly within the Scrum framework, responsible for maximizing the value of the product. They define the product vision, manage the [Product Backlog](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&cs=0&q=Product+Backlog&sa=X&ved=2ahUKEwj04bLPprmPAxXhWHADHbl3EcMQxccNegQIBRAB&mstk=AUtExfAbID25pv4FwfC72xvol-0TY9bKdD7zHWNZs2iCLbUulBakOfpDf1Vf6kX9zrEMVUbc5-Zl9XEPe2N9-BzFji-bN3c3DaxqxZ0HUt1A0EyV-NVGZoWr95buBp3vmw4aLds&csui=3) (a list of work for the development team), and act as a liaison between business stakeholders, customers, and the development team to ensure the product meets market and customer needs.

**How an Agile Team plan their work ?**

Agile teams plan their work continuously in iterative cycles called sprints, rather than with a single, upfront plan. They use a prioritized [product backlog](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&cs=0&q=product+backlog&sa=X&ved=2ahUKEwjC2OzvprmPAxXZ1jgGHbNTGy8QxccNegQIAxAB&mstk=AUtExfDr0NGw4KFY_VpOmjDfaCsoIsiwPgTsdSDOQGLBNYQ6CbnXqHI2ccz8Hlbr7Rcvmmq7vnJGtzQFCofs9lQhNW-JRiDxa2hPWRyj0PB-w5CSUjSLmJy8UUwG3VUGb9DFFfE&csui=3) of tasks, often as user stories, to guide development. Planning includes [sprint planning meetings](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&cs=0&q=sprint+planning+meetings&sa=X&ved=2ahUKEwjC2OzvprmPAxXZ1jgGHbNTGy8QxccNegQIBxAB&mstk=AUtExfDr0NGw4KFY_VpOmjDfaCsoIsiwPgTsdSDOQGLBNYQ6CbnXqHI2ccz8Hlbr7Rcvmmq7vnJGtzQFCofs9lQhNW-JRiDxa2hPWRyj0PB-w5CSUjSLmJy8UUwG3VUGb9DFFfE&csui=3) to choose work, daily stand-ups to review progress and obstacles, and regular [sprint reviews and retrospectives](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&cs=0&q=sprint+reviews+and+retrospectives&sa=X&ved=2ahUKEwjC2OzvprmPAxXZ1jgGHbNTGy8QxccNegQIBxAC&mstk=AUtExfDr0NGw4KFY_VpOmjDfaCsoIsiwPgTsdSDOQGLBNYQ6CbnXqHI2ccz8Hlbr7Rcvmmq7vnJGtzQFCofs9lQhNW-JRiDxa2hPWRyj0PB-w5CSUjSLmJy8UUwG3VUGb9DFFfE&csui=3) to gather feedback and adjust for future iterations. This adaptive, collaborative approach allows teams to stay aligned to goals, remain flexible, and deliver value incrementally.

**User Requirement :**

User requirements are the expectations, needs, and desires of users for a product or service

**Relation between user and task :**

The relationship between a user and a task is hierarchical, where users initiate and benefit from tasks, while tasks are the activities and steps required to achieve a user's goal. A user performs or initiates tasks to get a specific outcome, which is the value they receive, whereas tasks are the work done, often broken down from a larger user story, to deliver that value.

**What is Software Acceptance Criteria ?**

Software acceptance criteria are predefined, testable conditions that a software product, feature, or user story must meet to be considered complete and accepted by the end-user, customer, or stakeholders. They act as a clear checklist from the end-user's perspective, ensuring everyone has a common understanding of the requirements, defining the scope, and providing a basis for testing the functionality to confirm it works as intended.

**Advantages of Agile Methodology :**

Flexibility & Adaptability

* **Adapts to Change:**

Agile projects are iterative, allowing for changes in project requirements and priorities throughout the development cycle, which is a significant advantage over rigid traditional models.

* **Responds to Evolving Needs:**

Teams can adjust their approach to meet real-time working conditions and changing client demands, ensuring the final product remains relevant.

Speed & Efficiency

* **Faster Time to Market:**

Agile delivers working software in short, frequent cycles (sprints), allowing for quicker product releases and a competitive edge.

* **Continuous Delivery:**

Projects are broken into small, manageable increments, which leads to ongoing delivery of value and continuous improvement.

Quality & Customer Focus

* **Higher Product Quality:**

Regular testing and feedback loops within each sprint help in the early detection and quick resolution of defects, leading to a more robust product.

* **Increased Customer Satisfaction:**

Continuous customer involvement and feedback ensure the product closely aligns with user requirements and expectations, boosting satisfaction.

Control & Transparency

* **Improved Project Visibility:**

Regular demonstrations and daily progress reports provide stakeholders with a clear, day-to-day view of project status.

* **Better Control:**

The iterative nature and constant communication give stakeholders more control over the project's direction and progress.

Risk Reduction

* **Early Risk Detection:**

By delivering functional software in small increments, any major issues or flaws can be identified and addressed early, minimizing costly rework later in the project.

* **Salvageable Progress:**

Even if one approach doesn't work, the project is broken into small, manageable parts, making it easier to salvage usable components and adapt.

**Disadvantages of Agile methodology :**

* **Lack of Predictability:**

The high flexibility of Agile makes it difficult to predict the final outcome, timeline, or total cost of a project from the outset.

* **Limited Documentation:**

Comprehensive documentation is often deferred to the end of the project, or sometimes not done at all, which can make maintenance, onboarding new team members, and long-term knowledge transfer challenging.

* **Intensive Collaboration:**

Agile requires continuous interaction between developers, clients, and other stakeholders, demanding significant time, effort, and commitment from all parties.

* **Scope Creep:**

While flexibility is a strength, it can also make it easier for project scope to creep, as requirements can change and expand with each iteration.

* **Difficulty in Progress Measurement:**

Since progress is incremental and delivered in small iterations, measuring the overall project's health and final completion can be more complex than with traditional methods.

* **Client Involvement:**

Active client participation and constant feedback are crucial for Agile's success, but this can be demanding and may not be feasible for all clients.

* **Fragmented or Infinite Projects:**

If not managed carefully, Agile projects can become "endless" or deliver fragmented outputs if there isn't a clear overarching vision to steer the team toward a unified objective.

**CI/CD :**

CI/CD (Continuous Integration/Continuous Delivery or Deployment) is a DevOps practice that automates the software development lifecycle, from code changes to deployment, using a "CI/CD pipeline". Its main purpose is to enable faster, more frequent, and reliable delivery of high-quality software by integrating code changes, running automated tests, and deploying them to production environments. The use of CI/CD helps organizations catch bugs early, reduce manual errors, decrease the risk of deployments, and improve overall efficiency and development speed.

**Uses of Scrum :**

* **Delivering Value Quickly:**

Scrum enables teams to deliver working, valuable product increments at the end of each short sprint, getting faster end-to-end workflows from concept to completion.

* **Managing Complex Projects:**

For intricate projects where requirements are constantly changing, Scrum helps teams by breaking down complexity into smaller, more manageable pieces.

* **Adapting to Change:**

The framework's iterative nature and frequent feedback loops allow teams to respond to new information, market demands, or stakeholder input without disrupting the entire project.

* **Enhancing Collaboration & Transparency:**

Scrum emphasizes regular communication through daily meetings and involves stakeholders in sprint reviews, increasing transparency and building trust among team members.

* **Improving Product Quality:**

By incorporating testing and quality checks into each short sprint, Scrum helps identify and resolve issues early, resulting in a higher-quality product.

* **Fostering Continuous Improvement:**

Sprint retrospectives provide a structured opportunity for teams to reflect on their process and identify ways to become more efficient and effective.

**When Scrum is Particularly Useful**

* **Software Development:**

Scrum is widely used in software engineering and development to build complex products by embracing change.

* **Dynamic Environments:**

Startups and other businesses operating in fast-moving markets benefit from Scrum's ability to adapt to evolving needs and customer feedback.

* **Any Complex Project:**

While popular in tech, Scrum can be applied to any project involving complex problems, allowing teams to self-organize and deliver sustainable results.

**What is scrum framework ?**

Scrum is an agile framework for project management, primarily used in software development, that emphasizes iterative development and collaboration to deliver valuable products or services. It focuses on breaking down work into short, time-boxed iterations called [sprints](https://www.google.com/search?sca_esv=eceb7d72bb7e7d31&rlz=1C1GCEA_enIN1177IN1177&biw=1280&bih=665&q=sprints&sa=X&ved=2ahUKEwiS_eXTtbmPAxXMS3ADHSkyMfcQxccNegUIoAEQAQ), typically lasting one to four weeks. Scrum relies on a set of roles, events, artifacts, and rules to help teams self-organize and adapt to changing requirements.

**Advantages and Disadvantages of Scrum Project Management**

**Advantages of Scrum**

Here's why the framework is so popular today:

[Scrum](https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-scrum) can help teams complete project deliverables quickly and efficiently.

* Scrum can help teams complete project deliverables quickly and efficiently
* Scrum ensures effective use of time and money
* Large projects are divided into easily manageable sprints
* Developments are coded and tested during the sprint review
* Works well for fast-moving development projects
* The team gets clear visibility through scrum meetings
* Scrum, being agile, adopts feedback from customers and stakeholders
* Short sprints enable changes based on feedback a lot more easily
* The individual effort of each team member is visible during daily scrum meetings

**Disadvantages of Scrum**

But like every framework, scrum also has few disadvantages.

Nothing is perfect, and the Scrum methodology is no exception. In some cases, Scrum is combined with other project management techniques that can help resolve some of these drawbacks:

* Scrum often leads to scope creep, due to the lack of a definite end-date
* The chances of project failure are high if individuals aren't very committed or cooperative
* Adopting the Scrum framework in large teams is challenging
* The framework can be successful only with experienced team members
* Daily meetings sometimes frustrate team members
* If any team member leaves in the middle of a project, it can have a huge negative impact on the project
* Quality is hard to implement until the team goes through an aggressive testing process

**1.Docker :**

Docker is an open-source platform that automates the deployment, scaling, and management of applications within lightweight, portable containers. It packages an application and its dependencies into an isolated unit, ensuring consistent execution across different environments.

**Key Features of Docker:**

* **Containerization:**

Docker's core feature, enabling the packaging of applications and their dependencies into isolated, portable containers. This ensures consistency from development to production environments.

* **Portability:**

Docker containers can run on any system that supports Docker, regardless of the underlying infrastructure, promoting easy migration and deployment.

* **Isolation:**

Each container operates in an isolated environment, preventing conflicts between applications and ensuring secure execution.

* **Faster and Easier Configuration:**

Docker simplifies environment setup and application deployment, reducing configuration time and effort.

* **Increased Productivity:**

By streamlining development workflows and providing consistent environments, Docker enhances developer productivity.

* **Scalability:**

Docker facilitates rapid scaling of applications by easily replicating and deploying containers across multiple hosts.

* **Image Management:**

Docker utilizes images, which are read-only templates for creating containers. Docker Hub provides a vast repository for sharing and managing these images.

* **Networking and Volume Management:**

Docker provides robust features for managing network communication between containers and persistent data storage through volumes.

* **Docker Compose:**

A tool for defining and running multi-container Docker applications, simplifying the orchestration of complex services.

* **Docker Swarm:**

A native clustering and orchestration solution for Docker containers, enabling the management of a cluster of Docker hosts as a single virtual host.

* **Security:**

Docker incorporates security features like image scanning (Docker Scout) and content trust to ensure the integrity and security of containerized applications.

* **Integration with Development Tools:**

Docker seamlessly integrates with various development tools and platforms, enhancing the overall development experience.

**2.Jenkins :**

Jenkins is an open-source automation server primarily used for continuous integration (CI) and continuous delivery (CD) in software development. It automates various stages of the software development lifecycle, including building, testing, and deploying applications.

**Key Features of Jenkins:**

* **Continuous Integration and Delivery (CI/CD):**

Jenkins facilitates the automation of the entire CI/CD pipeline, enabling frequent integration of code changes and automated releases. This minimizes manual intervention and ensures a consistently deployable software state.

* **Extensive Plugin Ecosystem:**

Jenkins boasts a vast ecosystem of plugins, allowing integration with a wide array of tools and technologies. This includes version control systems (e.g., Git), build tools (e.g., Maven, Gradle), container platforms (e.g., Docker, Kubernetes), cloud services (e.g., AWS), and more.

* **Pipeline as Code:**

Jenkins allows defining build, test, and deployment pipelines as code, typically using Groovy scripts. This provides version control and programmatic control over the entire workflow.

* **Distributed Builds:**

Jenkins supports distributed builds by allowing tasks to be executed on multiple machines or nodes, enhancing scalability and performance.

* **Ease of Use and Configuration:**

Jenkins is relatively easy to install and configure through its web interface, which includes error checks and built-in help.

* **Open-Source and Community Support:**

As an open-source project, Jenkins benefits from a large and active community of users and contributors, providing extensive support and continuous development.

* **Automation Capabilities:**

Jenkins can automate virtually any task, not limited to code-related processes. It can execute scripts (e.g., Bash, Python) and integrate with automation tools like Ansible.

**3.Git :**

Git is a free and open-source distributed version control system designed to handle projects of all sizes with speed and efficiency. It allows multiple developers to work collaboratively on the same project without interfering with each other's work and provides a robust mechanism for tracking and managing changes to files over time.

Key features of Git include:

* **Distributed Architecture:**

Each developer has a complete local copy of the entire repository history, allowing for offline work and independent development. Changes are synchronized between local and remote repositories.

* **Branching and Merging:**

Git's powerful branching model enables developers to create isolated environments (branches) for developing new features, fixing bugs, or experimenting without affecting the main codebase. These branches can then be efficiently merged back into the main line of development.

* **Version Control and History Tracking:**

Git meticulously tracks every change made to files, providing a detailed history of modifications. This allows users to easily review past versions, compare changes, and revert to previous states if necessary.

* **Speed and Efficiency:**

Written in C, Git is known for its speed in handling large repositories and performing common operations like committing, branching, and merging.

* **Data Integrity:**

Git uses cryptographic hashing (SHA-1) to ensure the integrity of the stored data and prevent accidental corruption or malicious tampering.

* **Staging Area (Index):**

Git introduces a staging area, or index, which acts as an intermediate layer between the working directory and the repository. This allows developers to selectively stage changes for the next commit, providing fine-grained control over what gets committed.

* **Collaboration Support:**

Git facilitates seamless collaboration among development teams, enabling multiple individuals to work on the same project concurrently and efficiently manage code contributions.

* **Open Source and Community Support:**

As an open-source project, Git benefits from a large and active community, providing extensive documentation, tools, and support.

**4.Selenium :**

Selenium is an open-source framework designed for automating web browsers, primarily used for testing web applications.

**Key Features of Selenium:**

* **Cross-Browser Compatibility:**

Supports testing across various web browsers, including Chrome, Firefox, Safari, Edge, and Internet Explorer.

* **Multi-Language Support:**

Allows test script creation in multiple programming languages such as Java, Python, C#, Ruby, JavaScript, and more.

* **Open Source and Portable:**

Freely available for use and can be easily implemented across different operating systems like Windows, macOS, and Linux.

* **WebDriver API:**

Provides a robust API for interacting directly with web elements and controlling browser behavior.

* **Selenium Grid:**

Enables parallel test execution across multiple machines and browsers, significantly reducing test execution time.

* **Selenium IDE:**

Offers a record and playback tool for creating basic test scripts without extensive coding knowledge.

* **Relative Locators:**

(Introduced in Selenium 4) Allows locating web elements relative to other elements on a page, enhancing script robustness.

* **Integration with Testing Frameworks:**

Seamlessly integrates with popular testing frameworks like TestNG, JUnit, and NUnit for enhanced test management and reporting.

* **Handling Dynamic Elements:**

Provides mechanisms to interact with dynamic web elements like dropdowns, pop-ups, and alerts.

* **Cloud Integration:**

Can be integrated with cloud-based testing platforms for scalable and efficient testing.

* **Active Community Support:**

Benefits from a large and active community, providing extensive resources and support for users.

**Scrum Ceromonics :**

Scrum ceremonies, officially known as [Scrum events](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Scrum+events&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQIKRAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3), are structured meetings within a Scrum sprint to facilitate communication, planning, and reflection, ensuring transparency and alignment toward the Sprint Goal. The main ceremonies are Sprint Planning, the [Daily Scrum](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Daily+Scrum&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQIKhAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3), the Sprint Review, and the Sprint Retrospective. A fifth, continuous event, [Backlog Refinement](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Backlog+Refinement&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQILBAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3) (or grooming), helps prepare work for future sprints.

The Five Scrum Events

* [**Sprint Planning**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Planning&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIhwIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Held at the start of the sprint, this event is where the Development Team plans the work they will do to achieve the Sprint Goal.

* **Daily Scrum:**

A short, daily meeting (typically 15 minutes) for the Developers to inspect progress toward the Sprint Goal and plan their work for the next 24 hours.

* [**Sprint Review**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Review&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIlQIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

At the end of the sprint, the Scrum Team demonstrates the work they have completed to stakeholders and gathers feedback to adapt the [Product Backlog](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Product+Backlog&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIngIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3).

* [**Sprint Retrospective**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Retrospective&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIlwIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Also at the end of the sprint, the Scrum Team reflects on the past sprint to identify opportunities for improvement in their processes, tools, interactions, and individuals.

* [**Backlog Refinement**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Backlog+Refinement&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIjQIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Also known as backlog grooming, this is an ongoing activity where the Scrum Team clarifies and orders items in the Product Backlog, ensuring it remains up-to-date and ready for future Sprints.

Why They Matter

* **Transparency:** Ceremonies provide clear visibility into project progress and challenges.
* **Alignment:** They ensure the team is focused on the same goals and priorities.
* **Adaptability:** Regular feedback loops allow the team to adjust and improve their product and processes continuously.
* **Collaboration:** They create opportunities for team members to interact, discuss impediments, and find solutions together.

**Scrum ceramics :**

Scrum ceremonies, officially known as [Scrum events](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Scrum+events&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQIKRAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3), are structured meetings within a Scrum sprint to facilitate communication, planning, and reflection, ensuring transparency and alignment toward the Sprint Goal. The main ceremonies are Sprint Planning, the [Daily Scrum](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Daily+Scrum&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQIKhAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3), the Sprint Review, and the Sprint Retrospective. A fifth, continuous event, [Backlog Refinement](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Backlog+Refinement&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegQILBAB&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3) (or grooming), helps prepare work for future sprints.

The Five Scrum Events

* [**Sprint Planning**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Planning&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIhwIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Held at the start of the sprint, this event is where the Development Team plans the work they will do to achieve the Sprint Goal.

* **Daily Scrum:**

A short, daily meeting (typically 15 minutes) for the Developers to inspect progress toward the Sprint Goal and plan their work for the next 24 hours.

* [**Sprint Review**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Review&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIlQIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

At the end of the sprint, the Scrum Team demonstrates the work they have completed to stakeholders and gathers feedback to adapt the [Product Backlog](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Product+Backlog&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIngIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3).

* [**Sprint Retrospective**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Sprint+Retrospective&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIlwIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Also at the end of the sprint, the Scrum Team reflects on the past sprint to identify opportunities for improvement in their processes, tools, interactions, and individuals.

* [**Backlog Refinement**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Backlog+Refinement&sa=X&ved=2ahUKEwjvl7zHxLmPAxUWn2MGHf-OBDYQxccNegUIjQIQAQ&mstk=AUtExfCt_gWqukwNFg5JoXQXiFsLOeZ3rb6cBOjkYFOGHbayQRKxu1hNmcLZrQcGIcUgkTq2e4nVr7oqCV5eMcQprRcUwMlVI52iQgAfPZMgAxnKVwsdah3ReB0iGe74oJFkg0Y&csui=3)**:**

Also known as backlog grooming, this is an ongoing activity where the Scrum Team clarifies and orders items in the Product Backlog, ensuring it remains up-to-date and ready for future Sprints.

**Why They Matter**

* **Transparency:**Ceremonies provide clear visibility into project progress and challenges.
* **Alignment:**They ensure the team is focused on the same goals and priorities.
* **Adaptability:**Regular feedback loops allow the team to adjust and improve their product and processes continuously.
* **Collaboration:**They create opportunities for team members to interact, discuss impediments, and find solutions together.

**Kanban :**

Kanban is a visual work management method, originating from Toyota's lean manufacturing, that uses a visual board to track tasks through a process, like "To Do," "Doing," and "Done". It emphasizes visualizing work, limiting work in progress (WIP) to prevent bottlenecks, and fostering continuous improvement by managing the flow of tasks and proactively addressing obstacles. The term "kanban" means "visual signal" or "card" in Japanese, referring to the cards that represent tasks on the board.

Key Aspects of the Kanban Method

* **Visualization:**

Kanban boards provide a clear, visual representation of the entire workflow and the status of each task.

* **Work in Progress (WIP) Limits:**

This principle limits the number of tasks that can be in progress at any given time, preventing overloading and improving flow.

* **Continuous Improvement:**

The method is designed for incremental and evolutionary change, allowing teams to improve their processes by identifying and resolving bottlenecks.

* **Pull System:**

Work is "pulled" into the process as capacity becomes available, rather than being pushed onto the team.

* **Respect for People:**

Kanban values respect for people and encourages leadership at all levels.

How it Works

1. **1.**[**Kanban Board**](https://www.google.com/search?sca_esv=7eeccd2ba4d781b2&rlz=1C1GCEA_enIN1177IN1177&q=Kanban+Board&sa=X&ved=2ahUKEwjPtJeu27mPAxW9zzgGHamFBNQQxccNegUIrgMQAQ&mstk=AUtExfDStFacmfXqH3QsWJcItR2KWfQU0Dgb9WdWjM1owKhz-cAzN7Ph1SPbn6ulx5-BZcAZFPlLNIB4MRbLGjI2Z462l3ozovR_ARJuTAD4RojLL9qRPrqvHSfeGQ5KQrNSzMo&csui=3)**:**

Tasks are represented as cards on a board divided into columns that show the stages of a workflow (e.g., To Do, Doing, Done).

1. **2. Tracking Progress:**

Cards are moved across the columns as work progresses from one stage to the next.

1. **3. Identifying Bottlenecks:**

The visual nature of the board helps identify areas where tasks are piling up, indicating a bottleneck in the workflow.

1. **4. Balancing Work and Capacity:**

By managing WIP limits, teams can balance the demand for work with their available capacity, leading to a more efficient and predictable flow.

Benefits

* **Increased Efficiency:**

By highlighting bottlenecks and promoting continuous improvement, Kanban helps to make processes more efficient.

* **Better Transparency:**

All team members and stakeholders can see the status of work in real-time, improving communication and alignment.

* **Reduced Waste:**

Kanban helps to minimize excess inventory and context switching, reducing waste in the system.

* **Flexibility:**

It can be adapted to almost any repeatable process, not just manufacturing, including software development, IT services, and other commercial sectors.

**Devops lifecycle :**

**Key Stages of the DevOps Lifecycle:**

1. **Plan:**

Define project goals, strategies, and requirements by gathering feedback from users to align the project with business objectives.

1. **Code:**

Developers write the actual software code, with tools like Git used for version control and to ensure organization and good coding practices.

1. **Build:**

The code is compiled and integrated, often using tools like Jenkins, to ensure all components work together smoothly.

1. **Test:**

Automated tests verify the quality and functionality of the software changes at each step of the development pipeline, identifying bugs early.

1. **Release:**

The tested code is prepared for deployment to a production environment.

1. **Deploy:**

The prepared code is deployed, often using techniques like containerization with tools such as Docker and Kubernetes for efficient and stable releases.

1. **Operate:**

After deployment, software is run and managed in the production environment.