### **Navigation and File Operations:**

1. **cd (Change Directory):**
   * Changes the current working directory.
   * Example: cd /path/to/directory
2. **ls (List):**
   * Displays a list of files and directories.
   * Example: ls -l
3. **pwd (Print Working Directory):**
   * Prints the current working directory.
   * Example: pwd
4. **cp (Copy):**
   * Copies files or directories.
   * Example: cp file.txt /path/to/destination
5. **mv (Move):**
   * Moves or renames files and directories.
   * Example: mv file.txt newname.txt
6. **rm (Remove):**
   * Removes files or directories.
   * Example: rm file.txt
7. **mkdir (Make Directory):**
   * Creates a new directory.
   * Example: mkdir new\_directory
8. **rmdir (Remove Directory):**
   * Removes an empty directory.
   * Example: rmdir empty\_directory
9. **touch:**
   * Creates an empty file or updates file timestamps.
   * Example: touch newfile.txt
10. **cat:**
    * Concatenates and displays the content of files.
    * Example: cat file.txt
11. **nano/vi:**
    * Text editors for file editing.
    * Example: nano filename
12. **find:**
    * Searches for files in a directory hierarchy.
    * Example: find /path/to/search -name "filename"
13. **locate:**
    * Quickly finds the location of files.
    * Example: locate filename
14. **diff:**
    * Compares two files line by line.
    * Example: diff file1.txt file2.txt

### **User and Permission Management:**

1. **whoami:**
   * Displays the current user.
   * Example: whoami
2. **sudo (Superuser Do):**
   * Executes a command with superuser privileges.
   * Example: sudo command
3. **su (Switch User):**
   * Switches user or becomes another user.
   * Example: su username
4. **chmod (Change Mode):**
   * Changes file permissions.
   * Example: chmod 755 filename
5. **chown (Change Owner):**
   * Changes file owner and group.
   * Example: chown user:group filename
6. **passwd:**
   * Changes user password.
   * Example: passwd username
7. **useradd:**
   * Adds a new user account.
   * Example: useradd newuser
8. **userdel:**
   * Deletes a user account.
   * Example: userdel username
9. **usermod:**
   * Modifies a user account.
   * Example: usermod -aG groupname username

### **System Information and Monitoring:**

1. **uname:**
   * Displays system information.
   * Example: uname -a
2. **lscpu:**
   * Displays information about the CPU.
   * Example: lscpu
3. **lsblk:**
   * Lists information about block devices.
   * Example: lsblk
4. **df (Disk Free):**
   * Displays disk space usage.
   * Example: df -h
5. **du (Disk Usage):**
   * Displays directory space usage.
   * Example: du -h
6. **free:**
   * Displays system memory usage.
   * Example: free -m
7. **top/htop:**
   * Displays real-time system statistics.
   * Example: top
8. **ps (Process Status):**
   * Displays information about running processes.
   * Example: ps aux
9. **pgrep:**
   * Searches and displays the process IDs based on names.
   * Example: pgrep process\_name
10. **pkill:**
    * Kills or signals processes based on their names.
    * Example: pkill process\_name

### **Package Management:**

1. **yum (Yellowdog Updater Modified):**
   * Manages packages on Red Hat-based systems.
   * Example: yum install package\_name
2. **rpm (RPM Package Manager):**
   * Manages RPM packages.
   * Example: rpm -ivh package.rpm

### **Networking:**

1. **ifconfig (Interface Configuration):**
   * Displays and configures network interfaces.
   * Example: ifconfig
2. **ip:**
   * Configures and displays network interfaces.
   * Example: ip address show
3. **ping:**
   * Tests network connectivity.
   * Example: ping google.com
4. **traceroute:**
   * Displays the route to a destination.
   * Example: traceroute google.com
5. **netstat (Network Statistics):**
   * Displays network statistics.
   * Example: netstat -an
6. **nslookup:**
   * Performs DNS queries.
   * Example: nslookup example.com
7. **hostnamectl:**
   * Controls system hostname and settings.
   * Example: hostnamectl status

### **System Control:**

1. **systemctl:**
   * Controls the systemd system and service manager.
   * Example: systemctl start service\_name
2. **journalctl:**
   * Queries and displays messages from the journal.
   * Example: journalctl
3. **reboot:**
   * Restarts the system.
   * Example: reboot
4. **shutdown:**
   * Shuts down the system.
   * Example: shutdown -h now

### **File Compression and Archiving:**

1. **tar (Tape Archive):**
   * Creates and manipulates archive files.
   * Example: tar -czvf archive.tar.gz /path/to/files
2. **gzip (GNU zip):**
   * Compresses or decompresses files.
   * Example: gzip filename
3. **gunzip:**
   * Decompresses compressed files.
   * Example: gunzip filename.gz

### **Text Processing:**

1. **grep (Global Regular Expression Print):**
   * Searches for patterns in files.
   * Example: grep pattern file.txt
2. **sed (Stream Editor):**
   * Edits text using scripts.
   * Example: `sed

‘s/old/new/’ file.txt`

1. **awk:**
   * A versatile programming language for text processing.
   * Example: awk '{print $1}' file.txt

### **System Logs:**

1. **tail:**
   * Displays the last part of a file.
   * Example: tail -n 10 filename
2. **head:**
   * Displays the first part of a file.
   * Example: head -n 10 filename
3. **journalctl:**
   * Queries and displays messages from the journal.
   * Example: journalctl -xe

### **SSH:**

1. **ssh (Secure Shell):**
   * Connects to a remote server securely.
   * Example: ssh user@remote\_server
2. **scp (Secure Copy):**
   * Copies files securely between hosts.
   * Example: scp file.txt user@remote\_server:/path/to/destination

### **System Backup and Restore:**

1. **rsync (Remote Sync):**
   * Efficiently copies and synchronizes files between hosts.
   * Example: rsync -av source/ destination/
2. **tar:**
   * Creates and manipulates archive files.
   * Example: tar -czvf backup.tar.gz /path/to/files
3. **dd (Data Description):**
   * Copies and converts files, creating disk images.
   * Example: dd if=/dev/sdX of=/path/to/image.img

### **Miscellaneous:**

1. **echo:**
   * Displays messages or variables.
   * Example: echo "Hello, World!"
2. **uptime:**
   * Shows how long the system has been running.
   * Example: uptime

### **Database and Table Operations:**

1. **CREATE DATABASE:**
   * Creates a new database.

CREATE DATABASE mydatabase;

1.   
   **USE:**
   * Selects a database to work with.

USE mydatabase;

1.   
   **SHOW DATABASES:**
   * Displays a list of available databases.

SHOW DATABASES;

1.   
   **DROP DATABASE:**
   * Deletes an existing database.

DROP DATABASE mydatabase;

1.   
   **CREATE TABLE:**
   * Creates a new table.

CREATE TABLE employees (

id INT PRIMARY KEY,

name VARCHAR(50),

salary DECIMAL(10, 2)

);

1.   
   **DESCRIBE/DESC:**
   * Displays the structure of a table.

DESCRIBE employees;

1.   
   **INSERT INTO:**
   * Adds new records into a table.

INSERT INTO employees (id, name, salary)

VALUES (1, 'John Doe', 50000.00);

1.   
   **SELECT:**
   * Retrieves data from one or more tables.

SELECT id, name, salary

FROM employees

WHERE salary > 50000;

1.   
   **UPDATE:**
   * Modifies existing records in a table.

UPDATE employees

SET salary = 55000.00

WHERE id = 1;

1.   
   **DELETE:**
   * Removes records from a table.

DELETE FROM employees

WHERE id = 1;

### **Data Filtering and Sorting:**

1. **WHERE:**
   * Filters data based on a specified condition.

SELECT \*

FROM employees

WHERE department = 'IT';

1.   
   **ORDER BY:**
   * Sorts the result set based on one or more columns.

SELECT \*

FROM employees

ORDER BY salary DESC;

1.   
   **LIMIT:**
   * Restricts the number of rows in the result set.

SELECT \*

FROM employees

LIMIT 10;

1.   
   **GROUP BY:**
   * Groups rows based on a column’s values.

SELECT department, AVG(salary) as avg\_salary

FROM employees

GROUP BY department;

1.   
   **HAVING:**
   * Filters the result set based on a GROUP BY condition.

SELECT department, AVG(salary) as avg\_salary

FROM employees

GROUP BY department

HAVING avg\_salary > 60000;

### **Joins:**

1. **INNER JOIN:**
   * Retrieves records that have matching values in both tables.

SELECT employees.id, employees.name, departments.department\_name

FROM employees

INNER JOIN departments ON employees.department\_id = departments.id;

1.   
   **LEFT JOIN (or LEFT OUTER JOIN):**
   * Retrieves all records from the left table and matching records from the right table.

SELECT employees.id, employees.name, departments.department\_name

FROM employees

LEFT JOIN departments ON employees.department\_id = departments.id;

1.   
   **RIGHT JOIN (or RIGHT OUTER JOIN):**
   * Retrieves all records from the right table and matching records from the left table.

SELECT employees.id, employees.name, departments.department\_name

FROM employees

RIGHT JOIN departments ON employees.department\_id = departments.id;

### **Aggregation Functions:**

1. **COUNT:**
   * Counts the number of rows in a table.

SELECT COUNT(id) AS total\_employees

FROM employees;

1.   
   **SUM:**
   * Calculates the sum of values in a column.

SELECT department, SUM(salary) AS total\_salary

FROM employees

GROUP BY department;

1.   
   **AVG:**
   * Calculates the average value of a numeric column.

SELECT AVG(salary) AS avg\_salary

FROM employees;

1.   
   **MIN/MAX:**
   * Retrieves the minimum or maximum value in a column.

SELECT MIN(salary) AS min\_salary

FROM employees;

### **Data Modification and Transaction Control:**

1. **BEGIN TRANSACTION:**
   * Starts a transaction block.

BEGIN;

1.   
   **COMMIT:**
   * Saves the changes made during the current transaction.

COMMIT;

25

. **ROLLBACK:**- Undoes the changes made during the current transaction.  
sql ROLLBACK;

### **Indexing:**

1. **CREATE INDEX:**
   * Creates an index on one or more columns.

CREATE INDEX idx\_department

ON employees (department);

1.   
   **DROP INDEX:**
   * Removes an existing index.

DROP INDEX idx\_department;

### **SQL Server-Specific Commands:**

1. **TOP:**
   * Limits the number of rows returned in a result set (SQL Server).

SELECT TOP 5 \*

FROM employees;

1.   
   **ROW\_NUMBER():**
   * Assigns a unique number to each row in a result set (SQL Server).

SELECT name, salary, ROW\_NUMBER() OVER (ORDER BY salary DESC) AS row\_num

* + FROM employees;

