SQL for Software Testing

SQL= Structured Query Language

*Database = Collection of tables (Relative information)
--Types of databases
a. Rational Database (e.g. mySQL, Oracle, Postgres, MariaDB etc.)
b. Non-Rational Database (e.g. JSON, XML,graphs etc.)

*DBMS=Database Management System
--Software application that creates & manages the databases

Operations in SQL: (CRUD)
C=Create
R=Retrive/Read
U-Update
D=Delete

Query= Request to DBMS for retrival of specific information

SQL is hybrid language made of -
1. Data Definition Language (DDL) --> Create DB, tables
2. Data Query Language(DQL) --> Read information in DB, table, row, columns
3. Data Manipulation Language (DML)--> Alter information in DB, table (insert, update, delete)
4. Data Control Language (DCL)--> control access to database (user & permissions)

Important SQL languages for testing:
DDL (10%) , DQL(80%), DML(10%)

Data-types in SQL:
* int =Integer (whole number)
* decimal(m,n)= Decimal number exact value
* varchar(x)=string of length x
* datetime= date & time as YYYY-MM-DD HH:MI:SS
*Boolean

Schema: Database schema is the skeleton of database

Operators in SQL:
1. Arithmatic Operator

Operators	Description	Example
+	Add	x+y will give 90
-	Subtract	x-y will give -30
*	Multiply	x*y will give 1800
/	Divide	y/x will give 2
%	Modulo	y%x will give 0

2. Comparison oprators

Operator	Description	Example
=	Equal to	(x=y) is not true
!=	Equal or not	(x!=y) is true
< >	Not equal to	(x<>y) is true
>	Greater than	(x>y) is not true
<	Less than	(x<="" td="">
>=	Greater than or equal to	(x>=y) is not true
<=	Less than or equal to	(x<=y) is true
!<	Not less than	(x!<="" td="">
!>	Not greater than	(x!>y) is true

3. Logical Operator

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
IN	TRUE if the operand is equal to one of a list of expressions
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
EXISTS	TRUE if the subquery returns one or more records
LIKE	TRUE if the operand matches a pattern

Table : t1

	SR.NO.	NAME	PLACE
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL
4	1004	ABHIJIT CHOLKE	RAHATA

Surrogate Key

Primary Key

Foreign Key

Table: t2

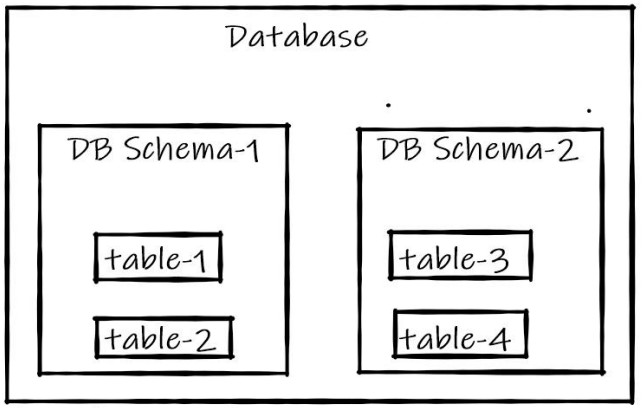
	BRANCH	MARKS	SR.NO.
1	ETC	10	1001
2	COMP	20	1002
3	ELEX	30	1003
4	IT	40	1004

*Surrogate Key --> no relation with data stored

*Primary Key --> important relation with data (defined as first column)

*Foreign Key -->defined in seperate table to refer entity in foreign table

Important SQL queries for testing:



for online compiler use --> www.rextester.com

* Create Database:
Query -->
create database DB1

* Create Schema:
Query-->
create schema dept1

*Create Table:
Query -->
create table dept1.t1 (sr_no int, name varchar(20), place varchar(10))
create table dept1.t2 (branch varchar(10), marks int, sr_no int)

*Using primary key:

Query-->

```
create table dept1.t1 (sr_no int, name varchar(20), place varchar(10),
primary key (sr_no))
```

*Using foreign key:

Query-->

```
create table dept1.t2 (branch varchar(10), marks int, sr_no int, foreign key
(sr_no) references dept1.t1(sr_no))
```

* Insert values in tables:

Syntax--> insert into schema.table_name values(val1,val2,val3)

Query-->

```
insert into dept1.t1 values(1001,'SACHIN TAWARE','PUNE')
insert into dept1.t1 values(1002, 'PRASHANT VIKHE', 'LONI')
insert into dept1.t1 values(1003, 'SACHIN ANAP', 'SATRAL')
insert into dept1.t1 values(1004, 'ABHIJIT CHOLKE', 'RAHATA')
insert into dept1.t1 values(1005, 'SACHIN TAWARE', 'LONI')
go
insert into dept1.t2 values('ETC', 10, 1001)
insert into dept1.t2 values('COMP', 20, 1002)
insert into dept1.t2 values('ELEX', 30, 1003)
insert into dept1.t2 values('IT', 40, 1004)
insert into dept1.t2 values('IT', 35, 1005)
go
insert into dept1.t3 values(2001, 'XYZ', 'LONDON')
insert into dept1.t3 values(2002, 'ABC', 'DELHI')
insert into dept1.t3 values(1004, 'PQR', 'PARIS')
insert into dept1.t3 values(2003, 'PRASHANT VIKHE', 'CALIFORNIA')
insert into dept1.t3 values(1004, 'ABHIJIT CHOLKE', 'RAHATA')
```

* 1. Read full tables:

Syntax--> Select * from table_name

Query-->

```
select * from dept1.t1
select * from dept1.t2
select * from dept1.t3
```

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL
4	1004	ABHIJIT CHOLKE	RAHATA
5	1005	SACHIN TAWARE	LONI

	branch	marks	sr_no
1	ETC	10	1001
2	COMP	20	1002
3	ELEX	30	1003
4	IT	40	1004
5	IT	35	1005

	sr_no	name	place
1	2001	XYZ	LONDON
2	2002	ABC	DELHI
3	1004	PQR	PARIS
4	2003	PRASHANT VIKHE	CALIFORNIA
5	1004	ABHIJIT CHOLKE	RAHATA

Database Schema dept1

t1

sr_no	name	place
1001	SACHIN TAWARE	PUNE
1002	PRASHANT VIKHE	LONI
1003	SACHIN ANAP	SATRAL
1004	ABHIJIT CHOLKE	RAHATA
1005	SACHIN TAWARE	LONI

t2

branch	marks	sr_no
ETC	10	1001
COMP	20	1002
ELEX	30	1003
IT	40	1004
IT	35	1005

t3

sr_no	name	place
2001	XYZ	LONDON
2002	ABC	DELHI
1004	PQR	PARIS
2003	PRASHANT VIKHE	CALIFORNIA
1004	ABHIJIT CHOLKE	RAHATA

2. Read columns from table

Syntax--> select col_namerom schema.table_name

Query:

select name from dept1.t1

	name
1	SACHIN TAWARE
2	PRASHANT VIKHE
3	SACHIN ANAP
4	ABHIJIT CHOLKE
5	SACHIN TAWARE

select marks, branch from dept1.t2

	name	place
1	SACHIN TAWARE	PUNE
2	PRASHANT VIKHE	LONI
3	SACHIN ANAP	SATRAL
4	ABHIJIT CHOLKE	RAHATA
5	SACHIN TAWARE	LONI

select distinct name from dept1.t1

	name
1	ABHIJIT CHOLKE
2	PRASHANT VIKHE
3	SACHIN ANAP
4	SACHIN TAWARE

3.Read using Aggregate function

(max,min,avg,top,count,sum)

Syntax-->select agrfun(col_name) from schema.table_name

Query:

#select max(marks) as max_marks from dept1.t2

	max_marks
1	40

#select min(marks) from dept1.t2

	(No column name)
1	10

#select avg(marks) from dept1.t2

	(No column name)
1	27

#select top 2 * from dept1.t1

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI

#select top 41 percent * from dept1.t1

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL

#select count(marks) from dept1.t2

	(No column name)
1	5

4. Read using the comparison operators

Syntax-->select * from schema.table_name where <condition>

Query:

#select * from dept1.t2 where marks=20

	branch	marks	sr_no
1	COMP	20	1002

#select * from dept1.t2 where marks!=20

	branch	marks	sr_no
1	ETC	10	1001
2	ELEX	30	1003
3	IT	40	1004
4	IT	35	1005

#select * from dept1.t2 where marks>10

	branch	marks	sr_no
1	COMP	20	1002
2	ELEX	30	1003
3	IT	40	1004
4	IT	35	1005

#select * from dept1.t2 where marks<30

	branch	marks	sr_no
1	ETC	10	1001
2	COMP	20	1002

#select * from dept1.t2 where marks>=20

	branch	marks	sr_no
1	COMP	20	1002
2	ELEX	30	1003
3	IT	40	1004
4	IT	35	1005

#select * from dept1.t2 where marks<=20

	branch	marks	sr_no
1	ETC	10	1001
2	COMP	20	1002

#select * from dept1.t2 where marks%3=0

	branch	marks	sr_no
1	ELEX	30	1003

#select * from dept1.t2 where marks/2=20

	branch	marks	sr_no
1	IT	40	1004

5. Read using Arithmatic operators

Syntax-->select * from schema.table_name where <conditions>

Query:

#select * from dept1.t2 where marks+sr_no=1044

	branch	marks	sr_no
1	IT	40	1004

#select * from dept1.t2 where sr_no-marks=973

	branch	marks	sr_no
1	ELEX	30	1003

#select * from dept1.t2 where sr_no*0=0

	branch	marks	sr_no
1	ETC	10	1001
2	COMP	20	1002
3	ELEX	30	1003
4	IT	40	1004
5	IT	35	1005

6. Read using Logical operators (AND, ORD, BETWEEN, IN, LIKE)

Syntax-->select * from schema.table_name where <condition1> operator <condition2>

Query:

#select * from dept1.t2 where branch='ETC' and marks=10

	branch	marks	sr_no
1	ETC	10	1001

#select * from dept1.t2 where branch='ETC' or marks=20

	branch	marks	sr_no
1	ETC	10	1001
2	COMP	20	1002

#select * from dept1.t2 where marks between 30 and 40

	branch	marks	sr_no
1	ELEX	30	1003
2	IT	40	1004
3	IT	35	1005

#select * from dept1.t2 where marks in (35,10,40)

	branch	marks	sr_no
1	ETC	10	1001
2	IT	40	1004
3	IT	35	1005

#select * from dept1.t1 where name like '%KE'

	sr_no	name	place
1	1004	ABHIJIT CHOLKE	RAHATA

#select * from dept1.t1 where name like 'SA%'

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1003	SACHIN ANAP	SATRAL
3	1005	SACHIN TAWARE	LONI

#select * from dept1.t1 where name like '%AN%'

	sr_no	name	place
1	1002	PRASHANT VIKHE	LONI
2	1003	SACHIN ANAP	SATRAL

#select * from dept1.t1 where name like 'P_%'

	sr_no	name	place
1	1002	PRASHANT VIKHE	LONI

7. Read ussing 'Order by' (Asc/Desc)
Syntax--> select * from schema.table_name order by column_name asc/desc

Query:
#select * from dept1.t1 order by name asc

	sr_no	name	place
1	1004	ABHIJIT CHOLKE	RAHATA
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL
4	1005	SACHIN TAWARE	LONI
5	1001	SACHIN TAWARE	PUNE

#select * from dept1.t2 order by marks desc

	branch	marks	sr_no
1	IT	40	1004
2	IT	35	1005
3	ELEX	30	1003
4	COMP	20	1002
5	ETC	10	1001

#select branch from dept1.t2 order by marks desc

	branch
1	IT
2	IT
3	ELEX
4	COMP
5	ETC

#select sr_no,place from dept1.t1 order by name asc

	sr_no	place
1	1004	RAHATA
2	1002	LONI
3	1003	SATRAL
4	1005	LONI
5	1001	PUNE

8. Read using 'Group by' (always used with aggregate function)
Syntax-->select aggr_func(column_name) from schema.table_name group by column_name

#select name,count(name) from dept1.t1 group by name

	name	(No column name)
1	ABHIJIT CHOLKE	1
2	PRASHANT VIKHE	1
3	SACHIN ANAP	1
4	SACHIN TAWARE	2

#select marks, count(marks) from dept1.t2 group by marks

	marks	(No column name)
1	10	1
2	20	1
3	30	1
4	35	1
5	40	1

#select name from dept1.t1 group by name having count(name)=2

	name
1	SACHIN TAWARE

9. --Alise (Nicknaming)
Syntax--> select column_name as alise_name from schema.table_name where <condition>

Query:
select max(marks) as max_marks from dept1.t2

	max_marks
1	40

#select info.name from dept1.t3 info

	name
1	XYZ
2	ABC
3	PQR
4	PRASHANT VIKHE
5	ABHIJIT CHOLKE

10. Copying tables

Syntax: select * into schema.table_name2 from schema.table_name1

Query:

#select * into dept1.t4 from dept1.t1
#select * from dept1.t4

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL
4	1004	ABHIJIT CHOLKE	RAHATA
5	1005	SACHIN TAWARE	LONI

#select * into dept1.t5 from dept1.t2 where 1=2
#select * from dept1.t5
--Note: It copies only layout of the table & not the rows

	sr_no	name	place

11.Union of tables (only for tables with same column names)- Merging of data into one table

Syntax: select * from schema.table_name1 union select * from schema.table_name2

Query:

#select * from dept1.t1 union select * from dept1.t3
#select * into dept1.t6 from (select * from dept1.t1 union select * from dept1.t3) t

	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL
4	1004	ABHIJIT CHOLKE	RAHATA
5	1004	PQR	PARIS
6	1005	SACHIN TAWARE	LONI
7	2001	XYZ	LONDON
8	2002	ABC	DELHI
9	2003	PRASHANT VIKHE	CALIFORNIA

12. Intersection of tables (Only common or associated rows will be selected.

Syntax: select * from schema.table_name1 intersect select * from table_name2

Query:

#select * from dept1.t4 intersect select * from dept1.t3

	sr_no	name	place
1	1004	ABHIJIT CHOLKE	RAHATA

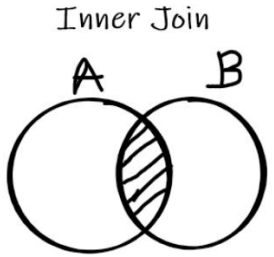
13. Joins between Tables:

Syntax: select * from schema.table_name1 alise1 <join type> schema.table_name2 alise2 on alise1.primary_key =alise2.secondary_key/Foreign key

13.A Inner join: (select common & associated only)

Query:

#select * from dept1.t1 a inner join dept1.t3 b on a.sr_no =b.sr_no
#select * from dept1.t1 a join dept1.t3 b on a.sr_no =b.sr_no



	sr_no	name	place	sr_no	name	place
1	1004	ABHIJIT CHOLKE	RAHATA	1004	PQR	PARIS
2	1004	ABHIJIT CHOLKE	RAHATA	1004	ABHIJIT CHOLKE	RAHATA

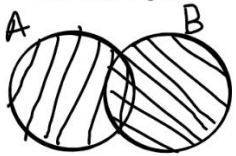
Observe--> row 1 has associated sr.no. & row 2 is common

13.B. Full outer join (Selects all values from both tables + common values+ associated values)

Query:

#select * from dept1.t1 a full outer join dept1.t3 b on a.sr_no =b.sr_no

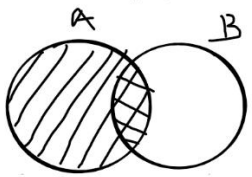
Full outer join



	sr_no	name	place	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE	*NULL*	*NULL*	*NULL*
2	1002	PRASHANT VIKHE	LONI	*NULL*	*NULL*	*NULL*
3	1003	SACHIN ANAP	SATRAL	*NULL*	*NULL*	*NULL*
4	1004	ABHIJIT CHOLKE	RAHATA	1004	PQR	PARIS
5	1004	ABHIJIT CHOLKE	RAHATA	1004	ABHIJIT CHOLKE	RAHATA
6	1005	SACHIN TAWARE	LONI	*NULL*	*NULL*	*NULL*
7	*NULL*	*NULL*	*NULL*	2001	XYZ	LONDON
8	*NULL*	*NULL*	*NULL*	2002	ABC	DELHI
9	*NULL*	*NULL*	*NULL*	2003	PRASHANT VIKHE	CALIFORNIA

13.C. Left Join (select all from table1 & join to only common & associated rows in table2, rest values are NULL)

Left Join



Query:

#select * from dept1.t1 a left join dept1.t3 b on a.sr_no =b.sr_no

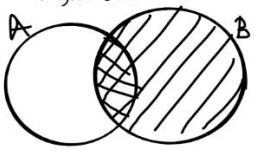
	sr_no	name	place	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE	*NULL*	*NULL*	*NULL*
2	1002	PRASHANT VIKHE	LONI	*NULL*	*NULL*	*NULL*
3	1003	SACHIN ANAP	SATRAL	*NULL*	*NULL*	*NULL*
4	1004	ABHIJIT CHOLKE	RAHATA	1004	PQR	PARIS
5	1004	ABHIJIT CHOLKE	RAHATA	1004	ABHIJIT CHOLKE	RAHATA
6	1005	SACHIN TAWARE	LONI	*NULL*	*NULL*	*NULL*

#select a.name, b.place from dept1.t1 a left join dept1.t3 b on a.sr_no =b.sr_no

	name	place
1	SACHIN TAWARE	*NULL*
2	PRASHANT VIKHE	*NULL*
3	SACHIN ANAP	*NULL*
4	ABHIJIT CHOLKE	PARIS
5	ABHIJIT CHOLKE	RAHATA
6	SACHIN TAWARE	*NULL*

13.D. Right Join (select all from table2 & join to only common & associated rows in table1, rest values are NULL)

Right Join



Query:

#select * from dept1.t1 a right join dept1.t3 b on a.sr_no =b.sr_no

	sr_no	name	place	sr_no	name	place
1	*NULL*	*NULL*	*NULL*	2001	XYZ	LONDON
2	*NULL*	*NULL*	*NULL*	2002	ABC	DELHI
3	1004	ABHIJIT CHOLKE	RAHATA	1004	PQR	PARIS
4	*NULL*	*NULL*	*NULL*	2003	PRASHANT VIKHE	CALIFORNIA
5	1004	ABHIJIT CHOLKE	RAHATA	1004	ABHIJIT CHOLKE	RAHATA

13.E Self Join (joining rows of same table on condition)

Query:

#select * from dept1.t1 a,dept1.t1 b where a.sr_no =b.sr_no

	sr_no	name	place	sr_no	name	place
1	1001	SACHIN TAWARE	PUNE	1001	SACHIN TAWARE	PUNE
2	1002	PRASHANT VIKHE	LONI	1002	PRASHANT VIKHE	LONI
3	1003	SACHIN ANAP	SATRAL	1003	SACHIN ANAP	SATRAL
4	1004	ABHIJIT CHOLKE	RAHATA	1004	ABHIJIT CHOLKE	RAHATA
5	1005	SACHIN TAWARE	LONI	1005	SACHIN TAWARE	LONI

14. Use of 'Case-When-Then' statement

Suntax:

```
select <column_names>
case
when <column_name> <condition> then <new value1>
else <new value2>
end as <name of new  column>
from schema.table_name
```

Query:

```
#select a.name, a.place,
case
when place in ('LONI', 'RAHATA','SATRAL')
then 'AHMEDNAGAR'
else 'PUNE'
end as District
from dept1.t1 a
```

	name	place	District
1	SACHIN TAWARE	PUNE	PUNE
2	PRASHANT VIKHE	LONI	AHMEDNAGAR
3	SACHIN ANAP	SATRAL	AHMEDNAGAR
4	ABHIJIT CHOLKE	RAHATA	AHMEDNAGAR
5	SACHIN TAWARE	LONI	AHMEDNAGAR

15. Alter /Add cloumn into table

--This is used to add columns into the table if required

Syntax: alter table schema.table_name add <column_name> <datatype>

Query:

```
#alter table dept1.t1 add zipcode int
```

sr_no	name	place	zipcode
1,001	SACHIN TAWARE	PUNE	[NULL]
1,002	PRASHANT VIKHE	LONI	[NULL]
1,003	SACHIN ANAP	SATRAL	[NULL]
1,004	ABHIJIT CHOLKE	RAHATA	[NULL]
1,005	SACHIN TAWARE	LONI	[NULL]

```
#insert into dept1.t1(sr_no,name,place,zipcode) values (1006, 'RANI RAO','SOLAPUR',1234)
#insert into dept1.t1 values (1007, 'NILESH RAO','LONI',413713)
#insert into dept1.t1(sr_no,name,place) values (1008, 'PRASHANT PATNKAR','SHRIRAMPUR')
select * from dept1.t1
```

sr_no	name	place	zipcode
1,002	PRASHANT VIKHE	LONI	[NULL]
1,003	SACHIN ANAP	SATRAL	[NULL]
1,004	ABHIJIT CHOLKE	RAHATA	[NULL]
1,005	SACHIN TAWARE	LONI	[NULL]
1,001	SACHIN TAWARE	PUNE	[NULL]
1,006	RANI RAO	SOLAPUR	1,234
1,007	NILESH RAO	LONI	413,713
1,008	PRASHANT PATNKAR	SHRIRAMPUR	[NULL]

16. Update values in table

--This updates /set the value of column on certain condition

Syntax: update schema.table_name set cloumn_name=<value> where <condition>

Query:

```
#update dept1.t1 set zipcode=411011 where sr_no =1006
#update dept1.t1 set zipcode=412307 where sr_no =1001
```

sr_no	name	place	zipcode
1,002	PRASHANT VIKHE	LONI	[NULL]
1,003	SACHIN ANAP	SATRAL	[NULL]
1,004	ABHIJIT CHOLKE	RAHATA	[NULL]
1,005	SACHIN TAWARE	LONI	[NULL]
1,007	NILESH RAO	LONI	413,713
1,008	PRASHANT PATNKAR	SHRIRAMPUR	[NULL]
1,006	RANI RAO	SOLAPUR	411,011
1,001	SACHIN TAWARE	PUNE	412,307

17. Delete Record/Rows

--this Deletes row on meeting certian conidition

Syntax: delete from schema.table_name where <condition>

Query:

```
#delete from dept1.t1 where sr_no=1006
```

sr_no	name	place	zipcode
1,002	PRASHANT VIKHE	LONI	[NULL]
1,003	SACHIN ANAP	SATRAL	[NULL]
1,004	ABHIJIT CHOLKE	RAHATA	[NULL]
1,005	SACHIN TAWARE	LONI	[NULL]
1,007	NILESH RAO	LONI	413,713
1,006	RANI RAO	SOLAPUR	411,011
1,001	SACHIN TAWARE	PUNE	412,307

Unix Shell Commands

--Unix Commands are required to establish communication with remote machine while testing (like thorugh "fileZilla" or "Putty" applications)
--Unix commands are run through shell/commandline directly that addresses Kernal process hence faster than operations performed from UI/GUI

1. To create, read, append files (cat -commands)

```
cat > batch1.txt            #Create a file & save it pressing Cntrl+d
cat batch1.txt              #Read contents of file
cat test1.txt test2.txt     #Read the contents of both files one after another
cat -n test1.txt            #Read contents of file with Line number
cat test1.txt>>test2.txt    #Append/Add (not override) contents of test2.txt into test3.txt keeping original contents of test3.txt as it is
tac test1.txt               #Read contents of file in reverse manner
```

2. To create directories/folders (parent-child) (mkdir-commands)

```
mkdir Lec25                 #Create a Empty directory
mkdir -p Lec26/{1,2,3,4}    #Create multiple directories with Parent-Childs
mkdir -pv Lec27/{a,b,c,d}   #Create multiple directories with Parent-Childs & prompt each while creating
```

3. To list the files & directories (ls commands)

```
ls                  #List files & directories from root directory (Excluding hidden)
ls -l               #List files & directories from root directory with details
ls -a               #List all (including hidden) from root directory with details
ls -d               #List root directories
ls -b               #List all binary files
ls -r               #List all in reverse
ls -R               #List all in recursive way (Parent-Child)
ls -l hello.*       #List all files of any extension but same name
ls -l *.txt         #List all files with specific/same extention
```

4. Remove files or directories (rm commands)

```
rm batch1.txt          #Remove file
rmdir Lec25            #Remove empty directory
rm -r Lec27            #Remove parent & child recursively (Non-Empty)
rm -rv Lec26           #Remove parent & child recursively (Non-Empty) & prompt it
```

5. Navigate to directories/folders (cd commands)

```
cd folder              #Navigate to Parent directory
cd folder/1            #Navigate direct to Child directort
cd..                   #Navigate one step back in hierarchy (child to parent)
```

6. Counting lines, words & characters in file (WC commands)

```
wc test2.c                  #Word count in file (Line-words-characters)
wc test2.c >test3.txt       #Copy outupt of any command in new file
wc test4.txt >>test4.txt    #Append output of any command into existing file
```

7. Change mode (Permissions) of file/directories (u=user, g=group, o=owner)

```
chmod u+rwx test1.txt      #Set-Add user level permission for a file
chmod u-rwx test1.txt      #Set-Remove user level permission of the file
chmod u-wx test1.txt       #Set specific permissions
chmod 777 test1.txt        #Set permissions using binary (111-111-111)
chmod g=u test2.c          #Copy & set permission to one level to another level
chown xxx test2.txt        #Change ownership of the file (username)
```

8. Head,Tail, more,less & top commands

```
head test5.txt          #Read top 10 lines from file
head -2 test5.txt       #Read top 2 lines from file
tail test5.txt          #Read last 10 lines from file
tail -2 test5.txt       #Read last 2 lines from file
less test5.txt          #Read page by page (Space=Next page, b=previous page, g=Last page & Shift+g=First page)
more test5.txt          #Display % wise page by page contents
top                     #Show all processed (Active+ Inactive)
top -i                  #Show only active processes
pidof top               #Extract Process ID (PID) of top active process
kill 812                #Kill the process with PID=812
```

9. Move files or directories (mv commands)

```
mv test1.txt test3.txt     #Move contents of test1.txt into test3.txt and remove test1.txt(original file)
mv -b test3.txt test4.txt  #Move contents of test3.txt into test4.txt and remove test3.txt(original file) keep backup of test4.txt (test4.txt~)
mv test test1              #Move one directory to another & remove original directory
```

10. Copy files or directories (cp commands)

```
cp test4.txt test5.txt     #Copy contents of test4.txt into test5.txt (override test5.txt)
cp test4.txt test5.txt test1 #Copy both files into directory
cp -r test1 test2          #Copy all contents recursively from one directory to another directory
```

11. Auxillary commands

```
clear                   #Clear console (Change console page)
man ls                  #Detail manual of the specified command (how to use)
pwd                     #Shows us present working directory
```

12. Linking file or directories (Hard-Link & Soft-Link)(ln commands)

```
ln test4.txt test4H.txt    #Create a Hardlink of the file (Permenent Backup for all time)
ln -s test6.txt test6S.txt #Crate a Softlink of the file (Temporary Shortcut of the file)
ls -li test4H.txt          #Read the I-node number of the link created (for hardlink i-node number is same & for softink it is different than parent)
```

13. Reading specific contents/string from files (grep commands)

```
grep "Sachin" test1.txt    #Read the line containing string "Sachin" from given file
grep -n "Sachin" test1.txt #Read the line with line number containing string "Sachin" from given file
grep -v "Sachin" test1.txt #Read the line that do not contains string "Sachin" from given file
grep -c "Manjri" test1.txt #Count the number of time certain string repeats in given file
```

14. Edit into file (vi command)

```
vi test4.txt               #Open a file into Editor (Esc+i=Insert ,Shift+:=Save ,wq=Exit)
```