Unix filesystem

File types

Vieditor:

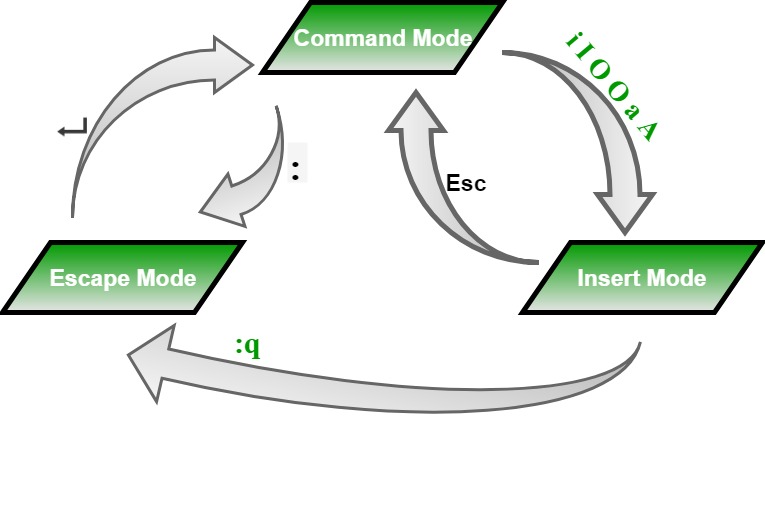
Esc+I will switch to insert mode which allowes us to insert data

To exit we again need to press esc

Esc+x to delete single char

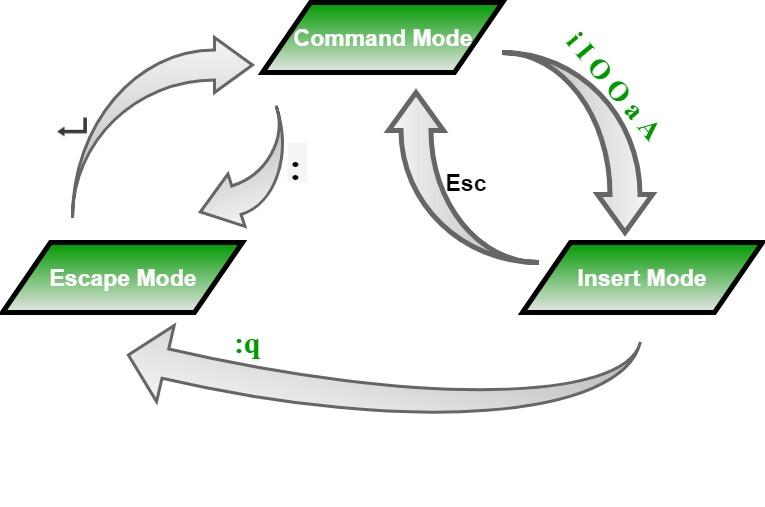
Edc + dd to delete complete line

Esac +a



Modes:command ,escape ,insert

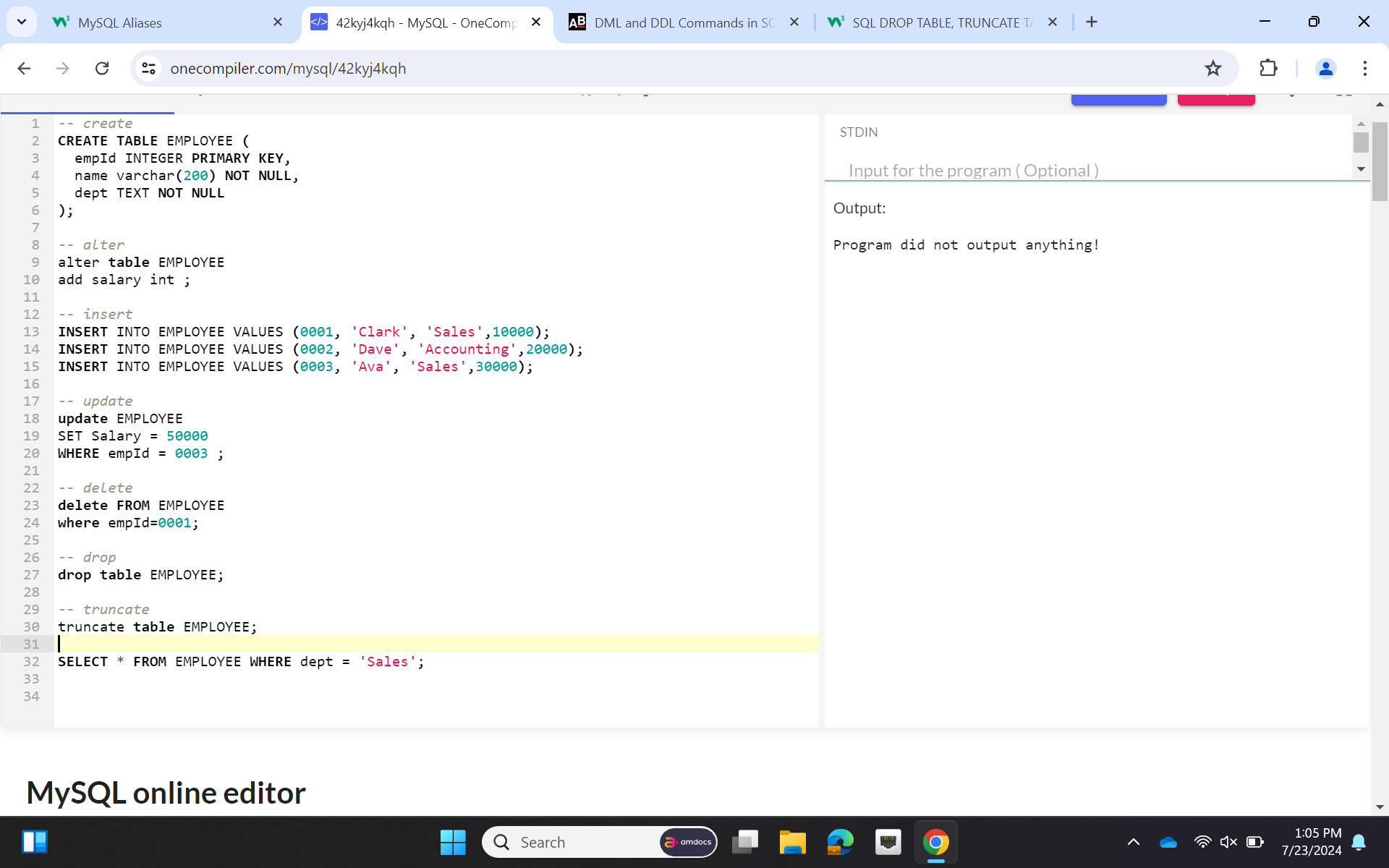
|  |  |  |
| --- | --- | --- |
| **chmod** | Changes the permissions of a file or directory. | chmod 644 file.txt |
| **chown** | Changes the owner and group of a file or directory. | chown user:group file.txt |
| **head** | Displays the first few lines of a file. | head file.txt |
| **tail** | Displays the last few lines of a file. | tail file.txt |
| **more** | Displays the contents of a file page by page. | more file.txt |
| **less** | Displays the contents of a file with advanced navigation features. | less file.txt |
| **diff** | Compares files line by line. | diff file1.txt file2.txt |
| **patch** | Applies a diff file to update a target file. | patch file.txt < changes.diff |



Shell script :acceleration

Grep command

Awk command

SQL:

Select \* from table\_name with (index(index\_name));

CREATE table employee(

empid INT primary key,

empname VARCHAR(400),

department VARCHAR(100),

contactno VARCHAR(12),

emailid VARCHAR(20),

empheadid int

);

CREATE TABLE empdept (

deptid VARCHAR(50) PRIMARY KEY,

deptName VARCHAR(100),

dept\_off VARCHAR(110),

depthead INT,

FOREIGN KEY (depthead) REFERENCES employee(empid)

);

create table empsalary (

empid int ,

salary bigint,

ispermanent varchar(3),

foreign key (empid) references employee(empid)

)

Create table project ( projectid int primary key, duration int)

create table country (cid varchar(50) primary key,cname varchar(100))

create table clienttable(clientid varchar(50) primary key,clientname varchar(50),cid varchar(100) references country(id))

create table empproject(empid int,projectid varchar(50),clientid varchar(50),

startyear int,endyear int,

foreign key(empid) references employee(empid),

foreign key(projectid) references project(projectid),

foreign key(clientid) references clienttable(clientid))

select \* from employee where empname like 'p%';

SELECT COUNT(salary) AS num\_permanent\_high\_salary

FROM empsalary

WHERE ispermanent = 'Yes'

AND salary > 5000;

SELECT \*

FROM employee

WHERE emailid LIKE '%@gmail.com';

SELECT \*

FROM employee

WHERE department LIKE 'E-104' or 'E-102';

PL/SQL:

Declare

a number(4);

b number(4);

c number(4);

Begin

a:= :a;

b:= :b;

if a>b then

c:=a;

else

c:=b;

end if;

dbms\_output.put\_line('Greater number is : ' || c);

End

Declare

a number(4);

b number(4);

Begin

a:= :a;

b:= Mod(a,2);

if b=1 then

dbms\_output.put\_line('Given number is odd');

else

dbms\_output.put\_line('Given number is even');

end if;

End

Declare

a number(4):=100;

Begin

if (a=10) then

dbms\_output.put\_line('value of a is 10');

elsif (a=20) then

dbms\_output.put\_line('value of a is 20');

elsif (a=30) then

dbms\_output.put\_line('value of a is 30');

else

dbms\_output.put\_line('none of the values are matching');

end if;

End

Declare

grade char(1):='A';

Begin

case grade

when 'A' then dbms\_output.put\_line('Very good');

when 'B' then dbms\_output.put\_line('Well Done');

when 'C' then dbms\_output.put\_line('YOu Passed');

when 'D' then dbms\_output.put\_line('Excellent');

else dbms\_output.put\_line('no such grade');

End Case;

End

Declare

x number(4):=10;

Begin

loop

dbms\_output.put\_line(x);

x:=x+10;

if x>50 then

exit;

end if;

End loop;

dbms\_output.put\_line('After exit the value of x i s: '|| x);

End

create table user1(id number(10) primary key,name varchar2(100));

create or replace procedure "INSUSER"

(id in number,

name in varchar2)

is begin

insert into user1 values(id,name);

end;

/

begin

insuser(101,'Rahul');

dbms\_output.put\_line('record inserted successfully');

end;

Declare

Cursor user\_cursor is select name from user1;

unm user1.name %type;

Begin

open user\_cursor;

if user\_cursor%isopen=false then

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

else

loop

fetch user\_cursor into unm;

if user\_cursor%notfound then

Exit;

end if;

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

end loop;

dbms\_output.put\_line('Total records: ' || user\_cursor%rowcount);

end if;

close user\_cursor;

end;

create table customer(

id int not null,

salary int null,

primary key(id)

)

insert into customer values ( 1 , 10000)

insert into customer values ( 2 , 20000)

insert into customer values ( 3 , 30000)

insert into customer values ( 4 , 40000)

select \* from customer

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;

/

update customer set salary=5000 where id=4

Packages in PLSSQL:

create table student(

rollno int primary key,

sname varchar(30),

age int not null,

course varchar(20)

);

insert into student values (1,'dhe',12,'BSC');

insert into student values (2,'josh',21,'BCOm');

insert into student values (3,'bhag',31,'BA');

select \* from student;

-- package specification

create or replace package pkg\_student is

procedure updaterecord(sno student.rollno%type);

function deleteRecord(snm student.sname%type)

return boolean;

End pkg\_student;

create or replace package body pkg\_student is

procedure updateRecord(sno student.rollno%type) is

begin

update student set age=23 where rollno=sno;

if sql%found then

dbms\_output.put\_line('Record updated');

else

dbms\_output.put\_line('recoed not found');

end if;

end updateRecord;

function deleteRecord(snm student.sname%type)

return boolean is

begin

delete from student where sname=snm;

return SQL%found;

end deleteRecord;

end pkg\_student;

Declare

sno student.rollno%type;

s\_age student.age%type;

snm student.sname%type;

Begin

sno:= :sno;

snm:= :snm;

pkg\_Student.updateRecord(sno);

if pkg\_student.deleteRecord(snm) then

dbms\_output.put\_line('Record deleted');

else

dbms\_output.put\_line('record not found');

end if;

END;

Record updated

Record deleted

Statement processed.

Eclipse 20-21 enterprise

Goto jenkin login

Click on new item option

Create new project give name to it click on ok

Select github project

Add git hub project url

Select git

Then add http project url

In repo add github username and password

And then click on add

Mention branch on which we pushed the project

Now select pull scm

In build steps select invoke top level maven target

And write install

And save it