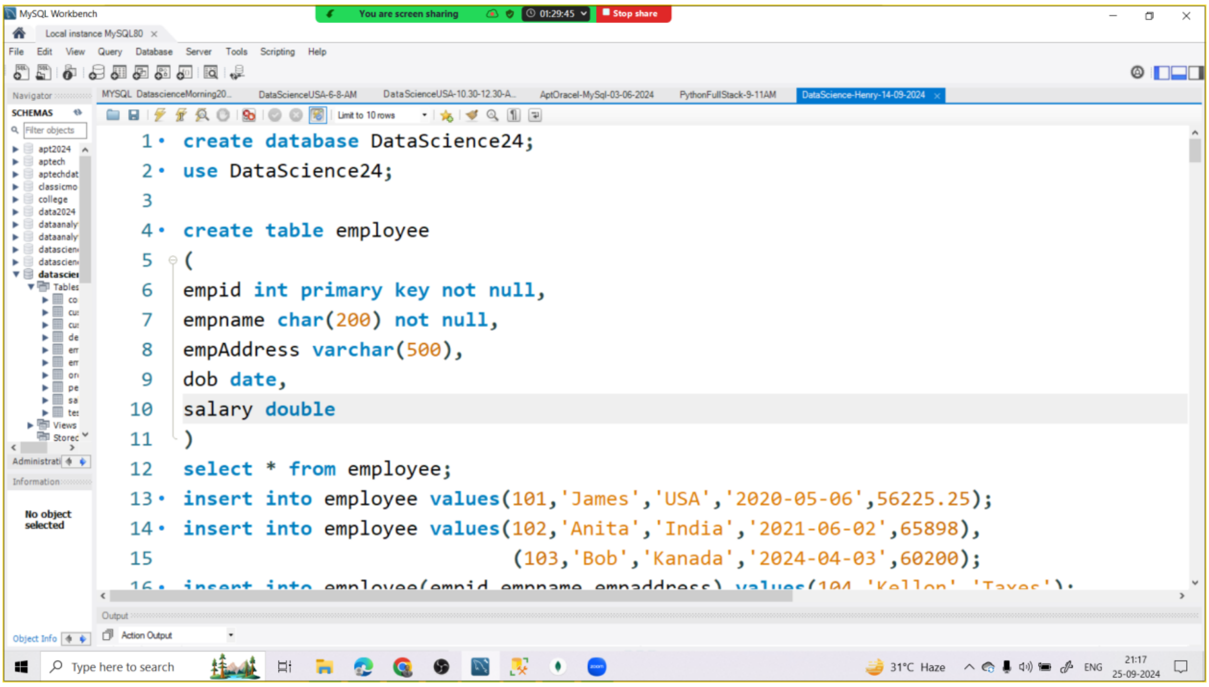
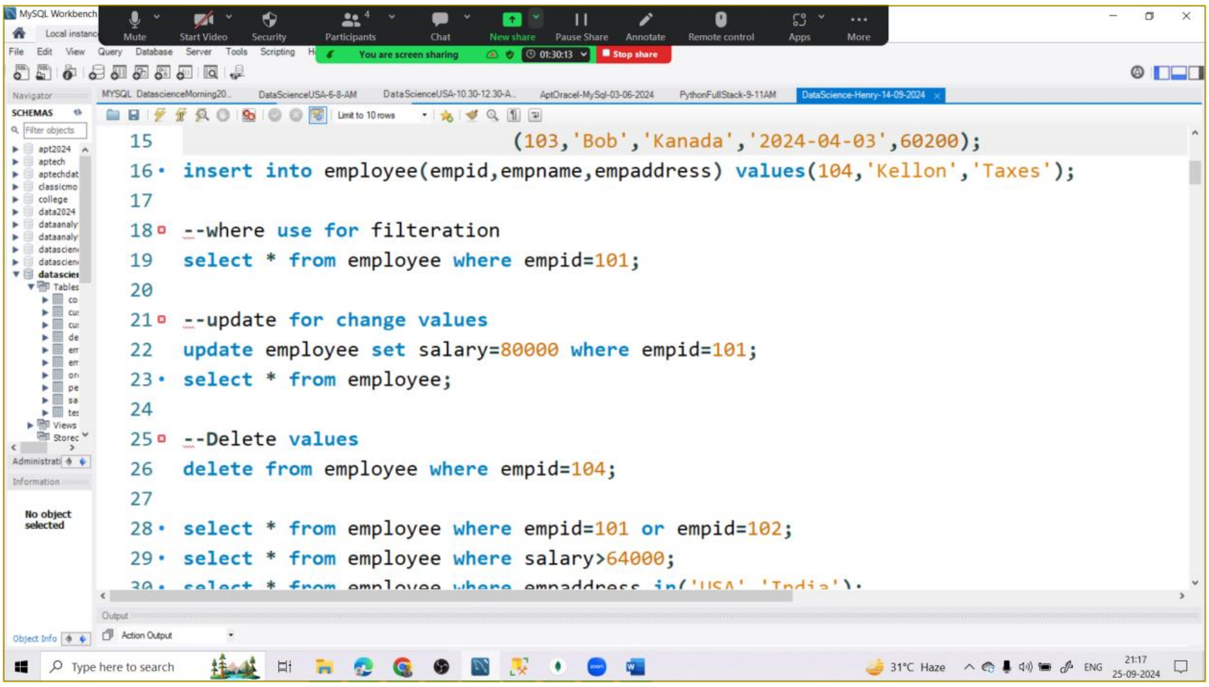
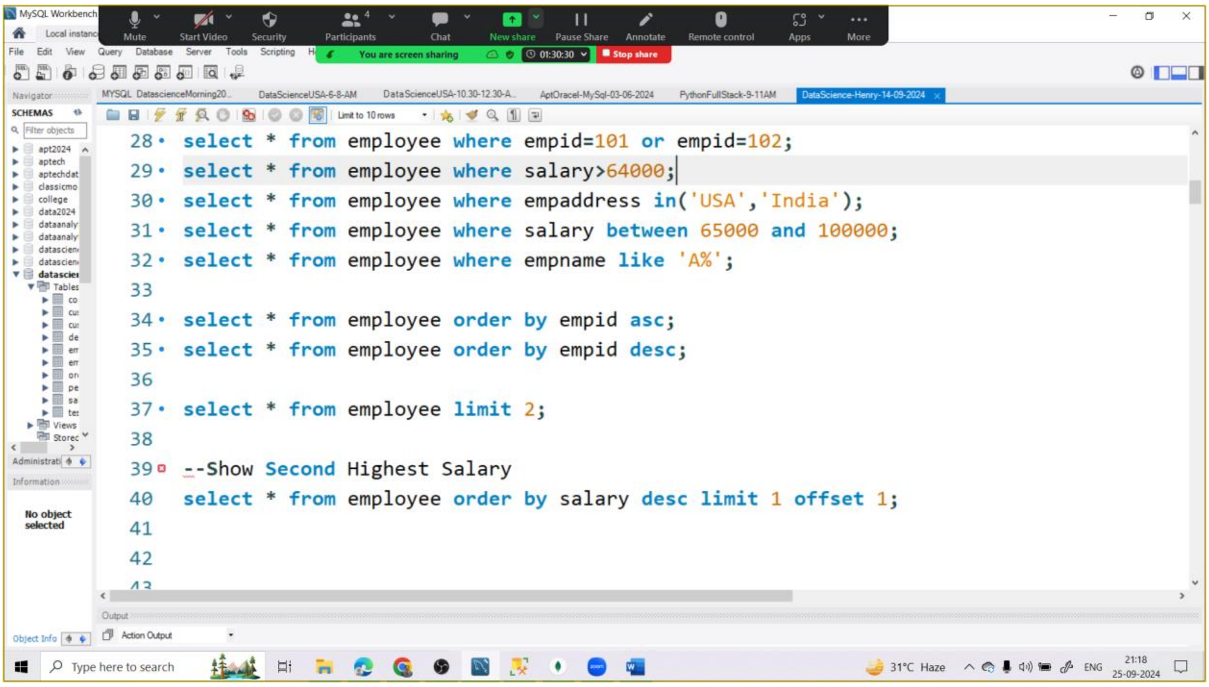


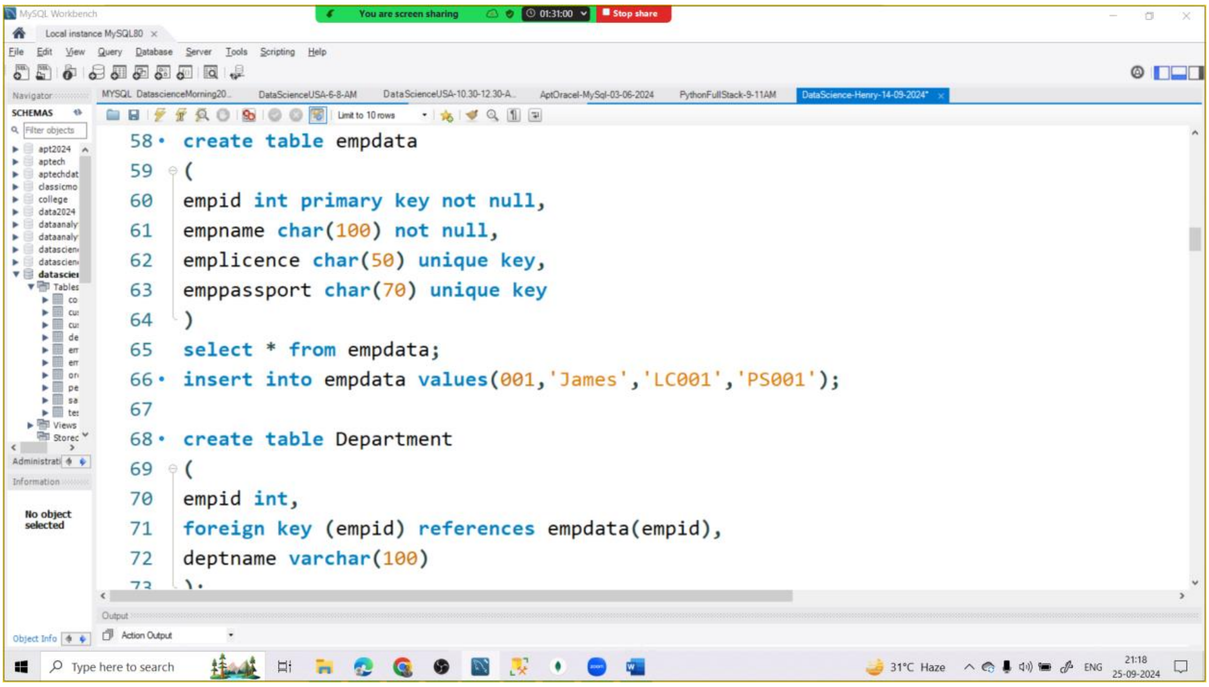
```
create database DataScience24;
use DataScience24;

create table employee
(
empid int primary key not null,
empname char(200) not null,
empAddress varchar(500),
dob date,
salary double
)
select * from employee;
insert into employee values(101,'James','USA','2020-05-06',56225.25);
insert into employee values(102,'Anita','India','2021-06-02',65898),
                            (103,'Bob','Kanada','2024-04-03',60200);
insert into employee(empid,empname,empaddress) values(104,'Kellon','Taxes');
```



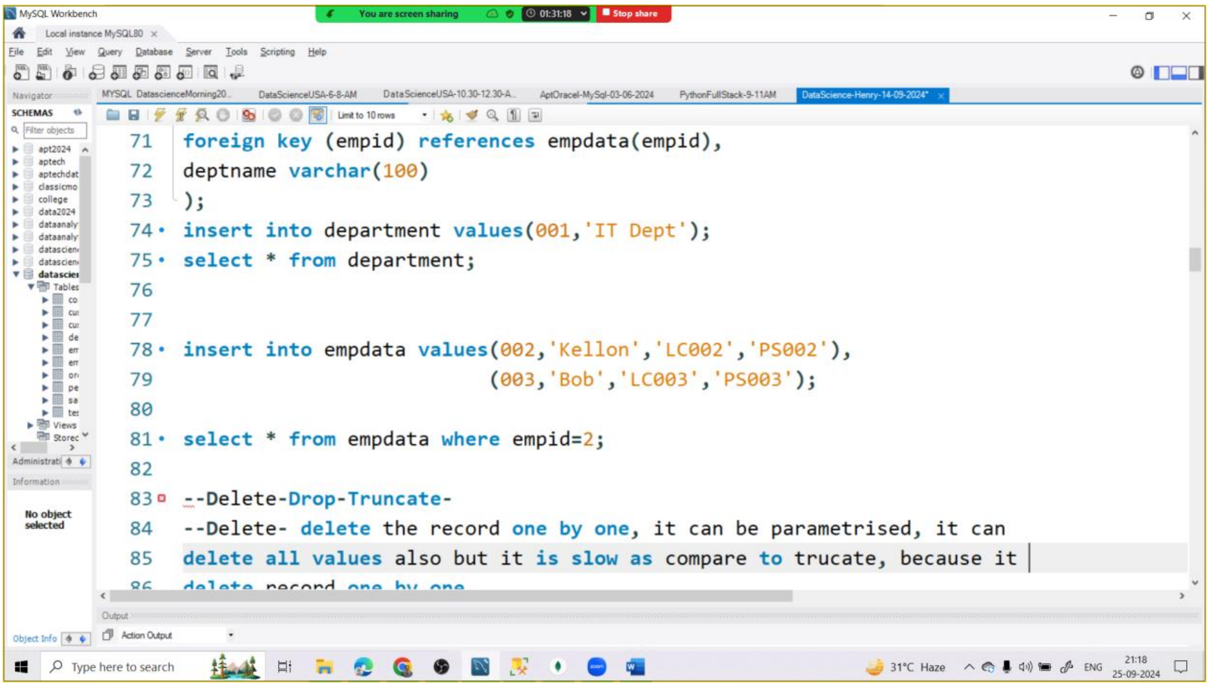
```
                            (103,'Bob','Kanada','2024-04-03',60200);
insert into employee(empid,empname,empaddress) values(104,'Kellon','Taxes');

--where use for filteration
select * from employee where empid=101;

--update for change values
update employee set salary=80000 where empid=101;
select * from employee;

--Delete values
delete from employee where empid=104;

select * from employee where empid=101 or empid=102;
select * from employee where salary>64000;
select * from employee where empaddress in('USA','India');
```



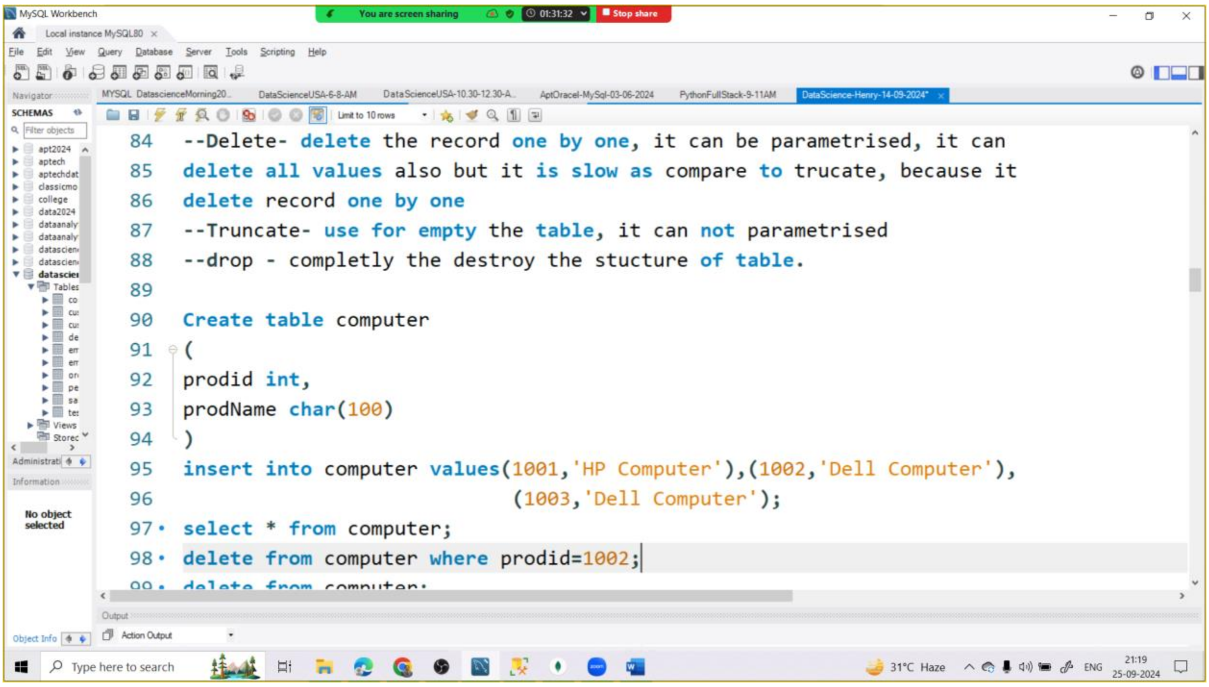
```sql
select * from employee where empid=101 or empid=102;
select * from employee where salary>64000;
select * from employee where empaddress in('USA','India');
select * from employee where salary between 65000 and 100000;
select * from employee where empname like 'A%';

select * from employee order by empid asc;
select * from employee order by empid desc;

select * from employee limit 2;

--Show Second Highest Salary
select * from employee order by salary desc limit 1 offset 1;


```



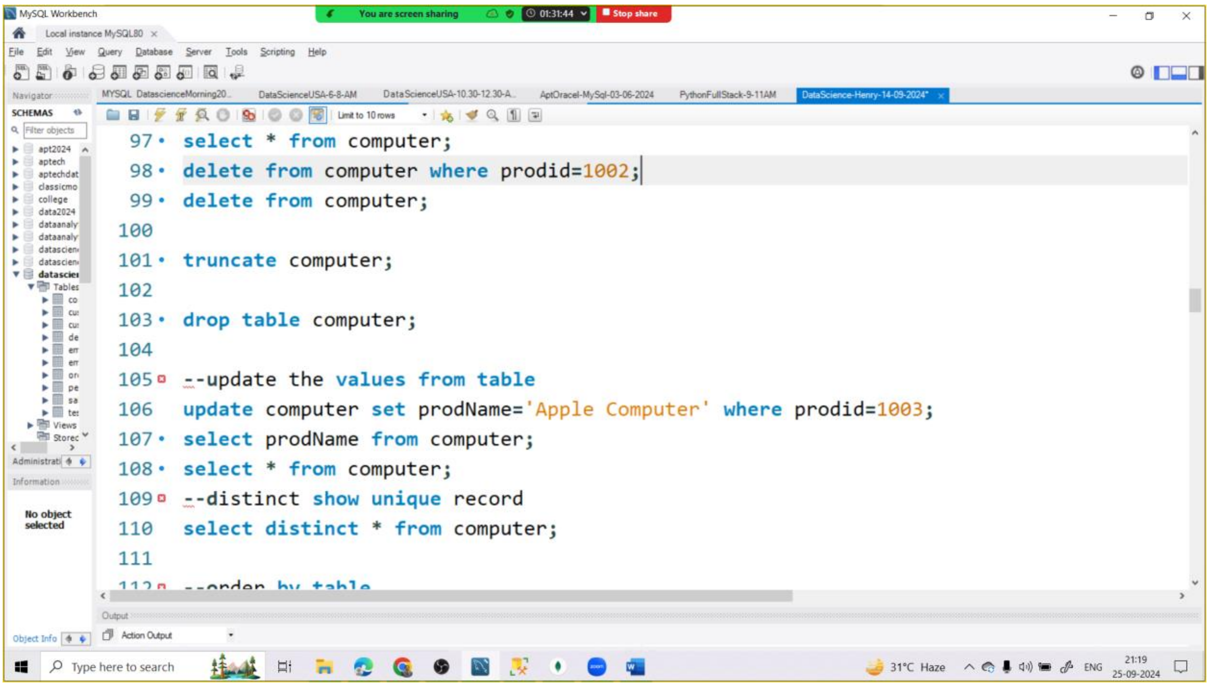
```sql
create table empdata
(
empid int primary key not null,
empname char(100) not null,
emplicence char(50) unique key,
emppassport char(70) unique key
)
select * from empdata;
insert into empdata values(001,'James','LC001','PS001');

create table Department
(
empid int,
foreign key (empid) references empdata(empid),
deptname varchar(100)
);
```



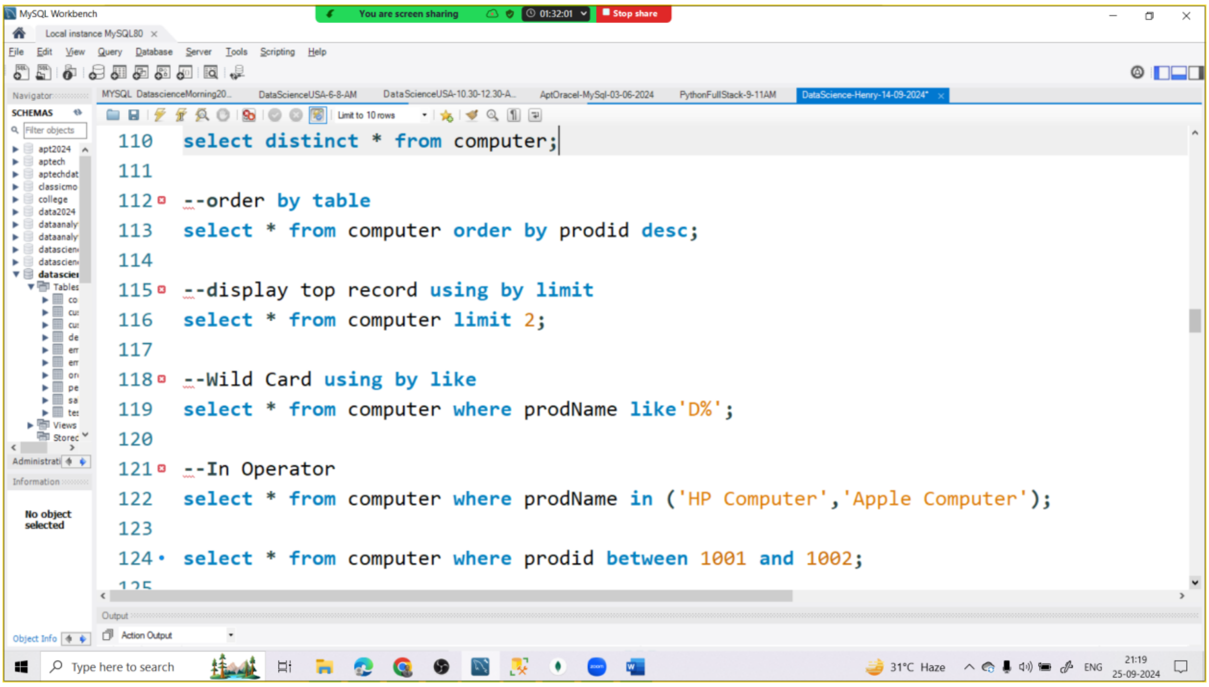
```sql
foreign key (empid) references empdata(empid),
deptname varchar(100)
);
insert into department values(001,'IT Dept');
select * from department;


insert into empdata values(002,'Kellon','LC002','PS002'),
                          (003,'Bob','LC003','PS003');

select * from empdata where empid=2;

--Delete-Drop-Truncate-
--Delete- delete the record one by one, it can be parametrised, it can
delete all values also but it is slow as compare to trucate, because it
delete record one by one
```



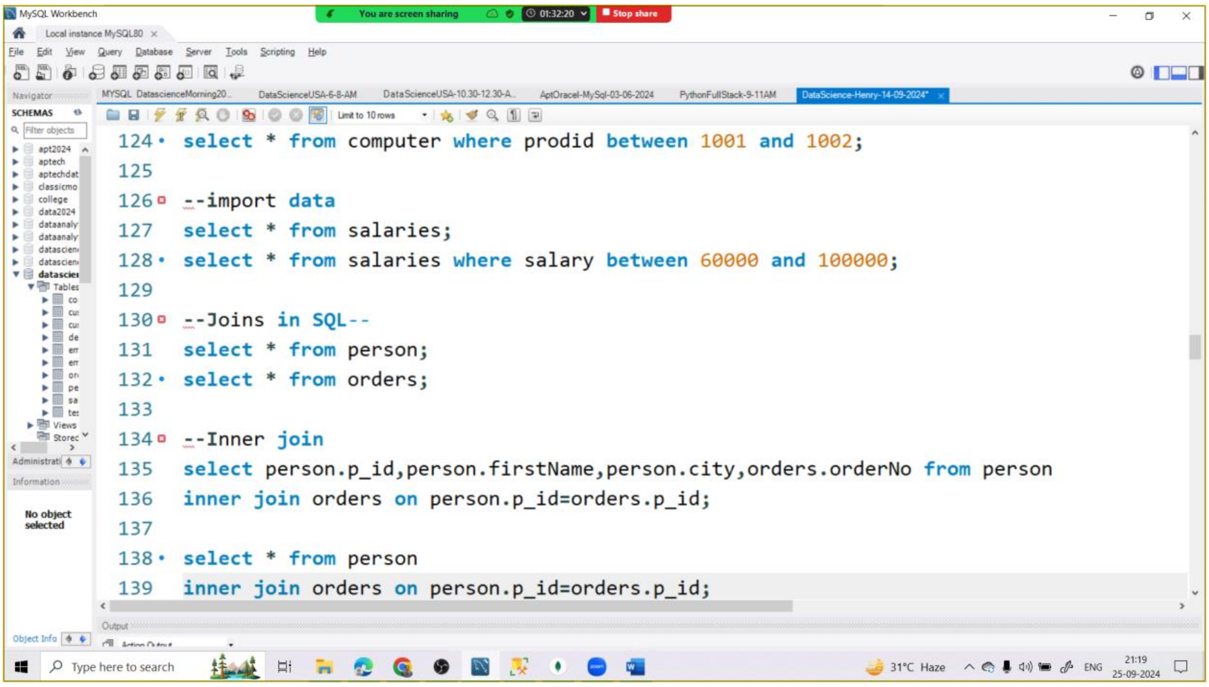
```sql
--Delete- delete the record one by one, it can be parametrised, it can
delete all values also but it is slow as compare to trucate, because it
delete record one by one
--Truncate- use for empty the table, it can not parametrised
--drop - completly the destroy the stucture of table.

Create table computer
(
prodid int,
prodName char(100)
)
insert into computer values(1001,'HP Computer'),(1002,'Dell Computer'),
                            (1003,'Dell Computer');
select * from computer;
delete from computer where prodid=1002;
delete from computer;
```



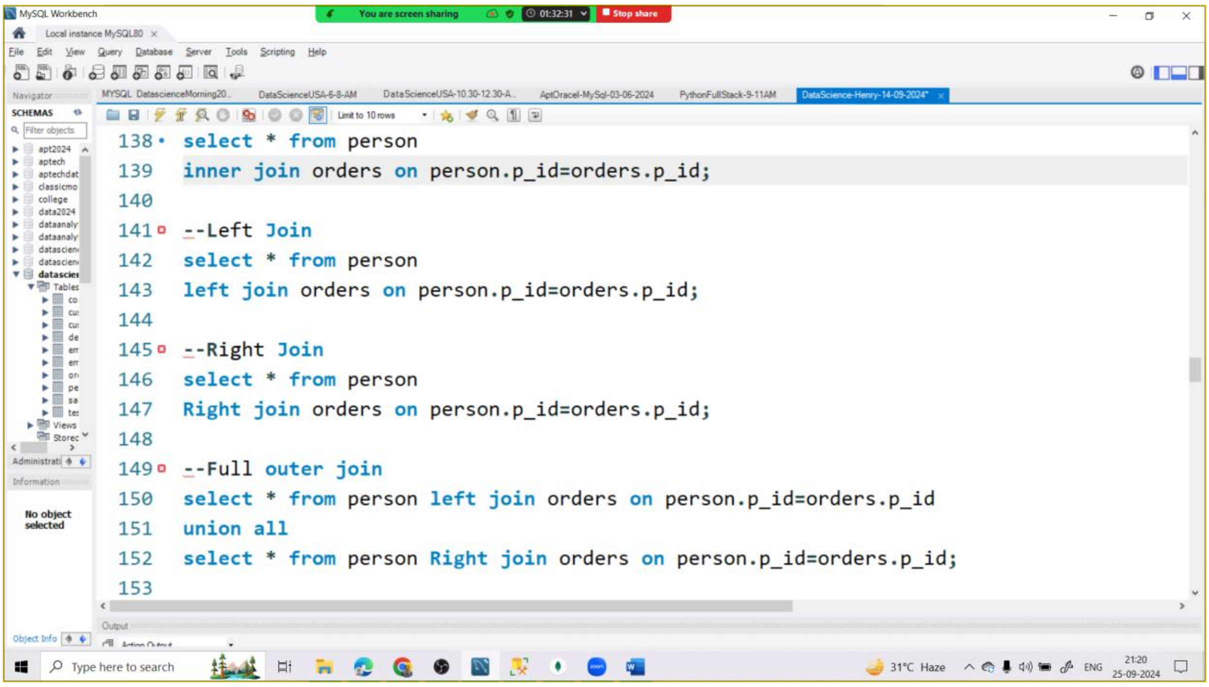
```sql
select * from computer;
delete from computer where prodid=1002;
delete from computer;

truncate computer;

drop table computer;

--update the values from table
update computer set prodName='Apple Computer' where prodid=1003;
select prodName from computer;
select * from computer;
--distinct show unique record
select distinct * from computer;

--order by table
```



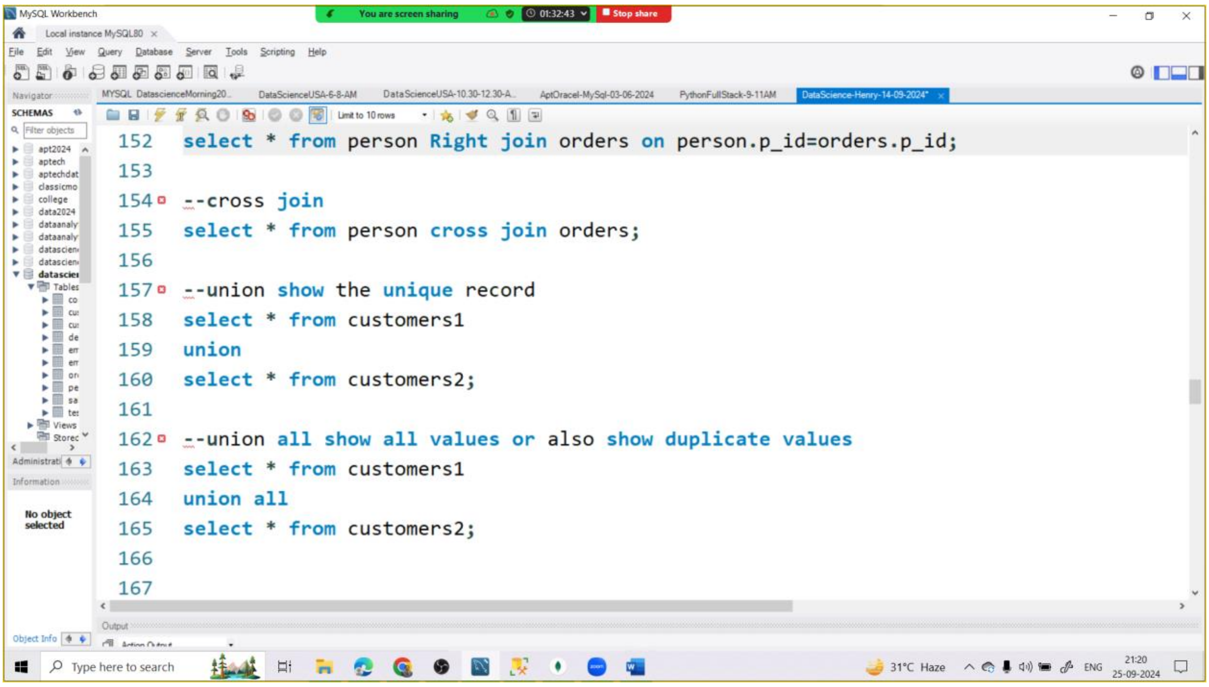
```sql
select distinct * from computer;

--order by table
select * from computer order by prodid desc;

--display top record using by limit
select * from computer limit 2;

--Wild Card using by like
select * from computer where prodName like'D%';

--In Operator
select * from computer where prodName in ('HP Computer','Apple Computer');

select * from computer where prodid between 1001 and 1002;
```



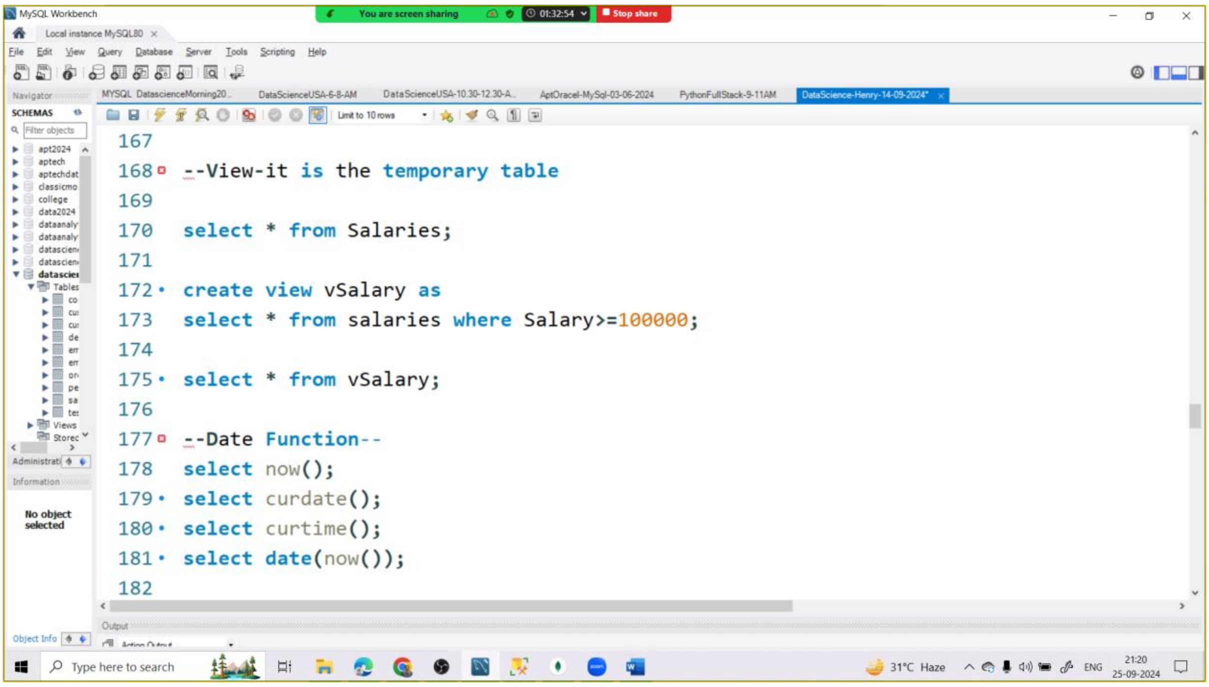
```
select * from computer where prodid between 1001 and 1002;

--import data
select * from salaries;
select * from salaries where salary between 60000 and 100000;

--Joins in SQL--
select * from person;
select * from orders;

--Inner join
select person.p_id,person.firstName,person.city,orders.orderNo from person
inner join orders on person.p_id=orders.p_id;

select * from person
inner join orders on person.p_id=orders.p_id;
```



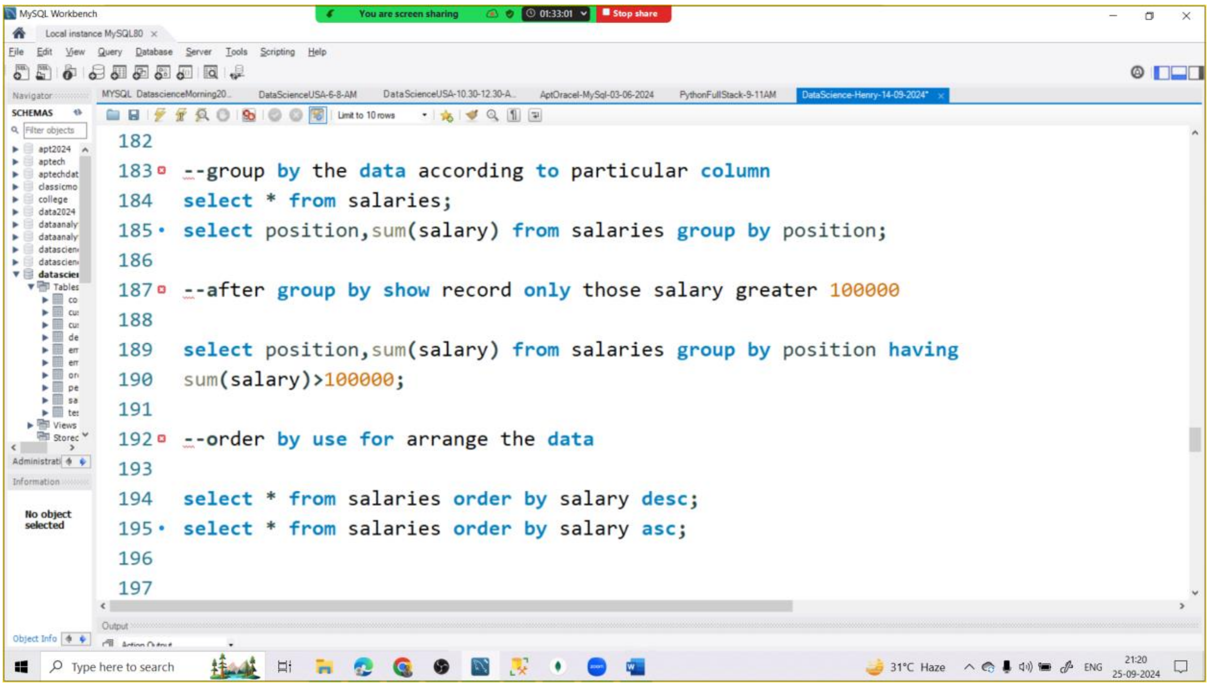
```
select * from person
inner join orders on person.p_id=orders.p_id;

--Left Join
select * from person
left join orders on person.p_id=orders.p_id;

--Right Join
select * from person
Right join orders on person.p_id=orders.p_id;

--Full outer join
select * from person left join orders on person.p_id=orders.p_id
union all
select * from person Right join orders on person.p_id=orders.p_id;
```



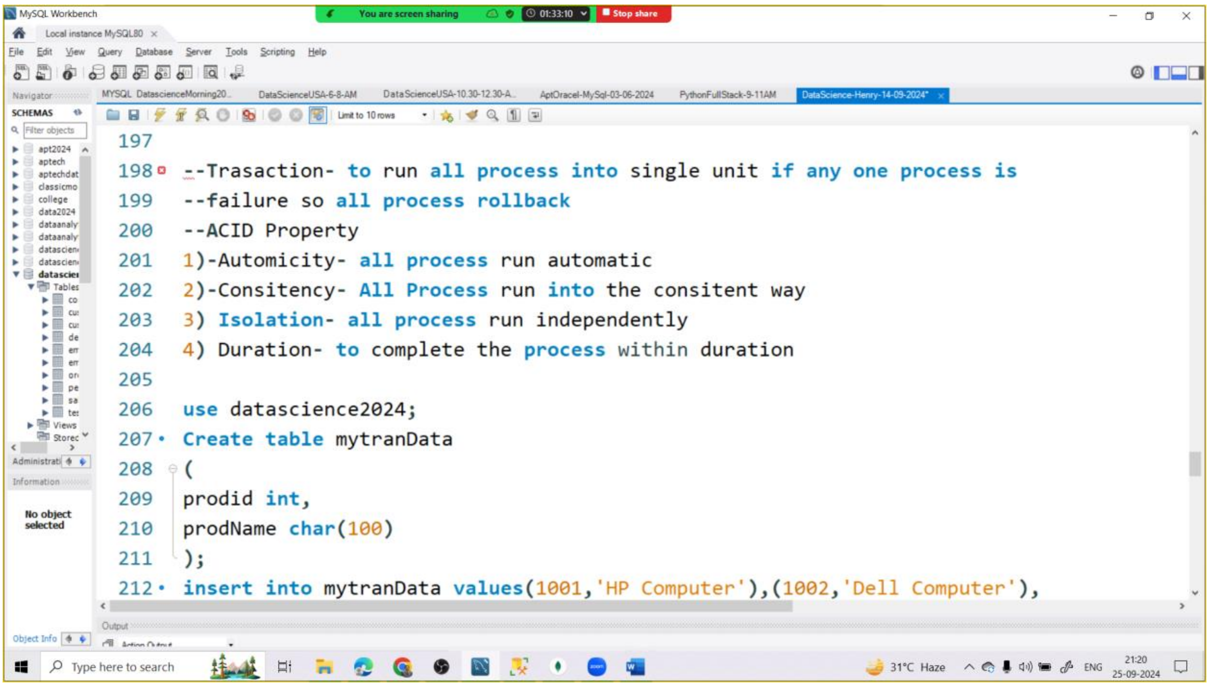
```
select * from person Right join orders on person.p_id=orders.p_id;

--cross join
select * from person cross join orders;

--union show the unique record
select * from customers1
union
select * from customers2;

--union all show all values or also show duplicate values
select * from customers1
union all
select * from customers2;


```



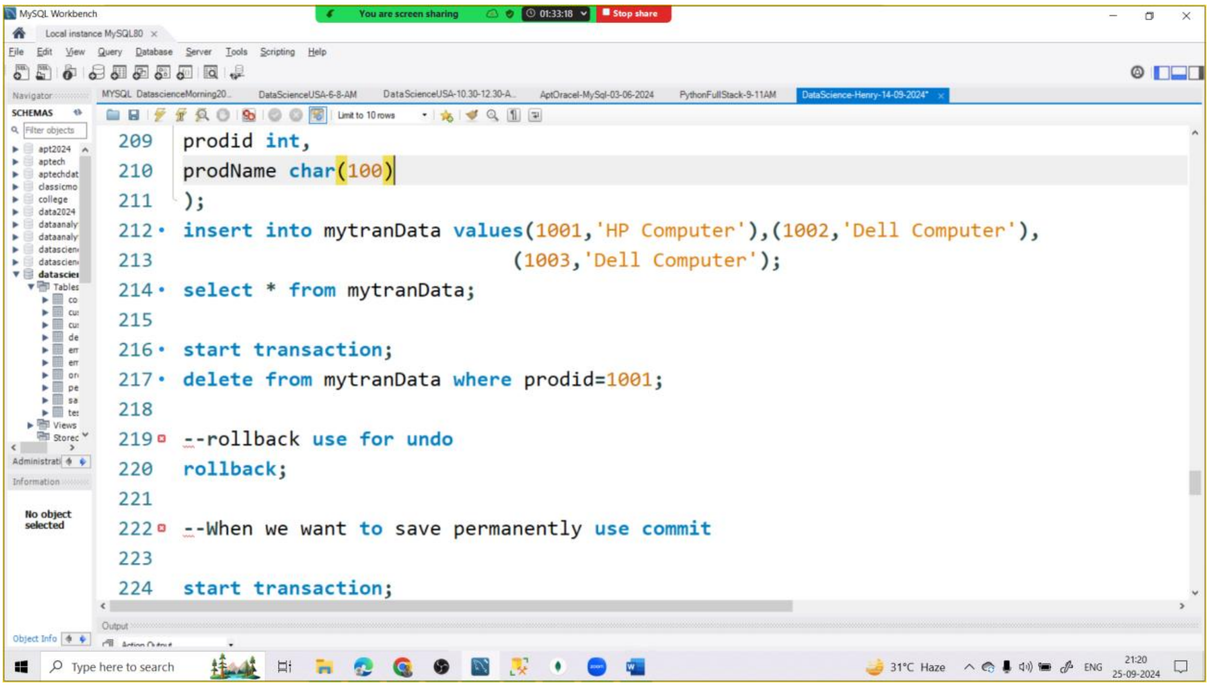
```

--View-it is the temporary table

select * from Salaries;

create view vSalary as
select * from salaries where Salary>=100000;

select * from vSalary;

--Date Function--
select now();
select curdate();
select curtime();
select date(now());

```



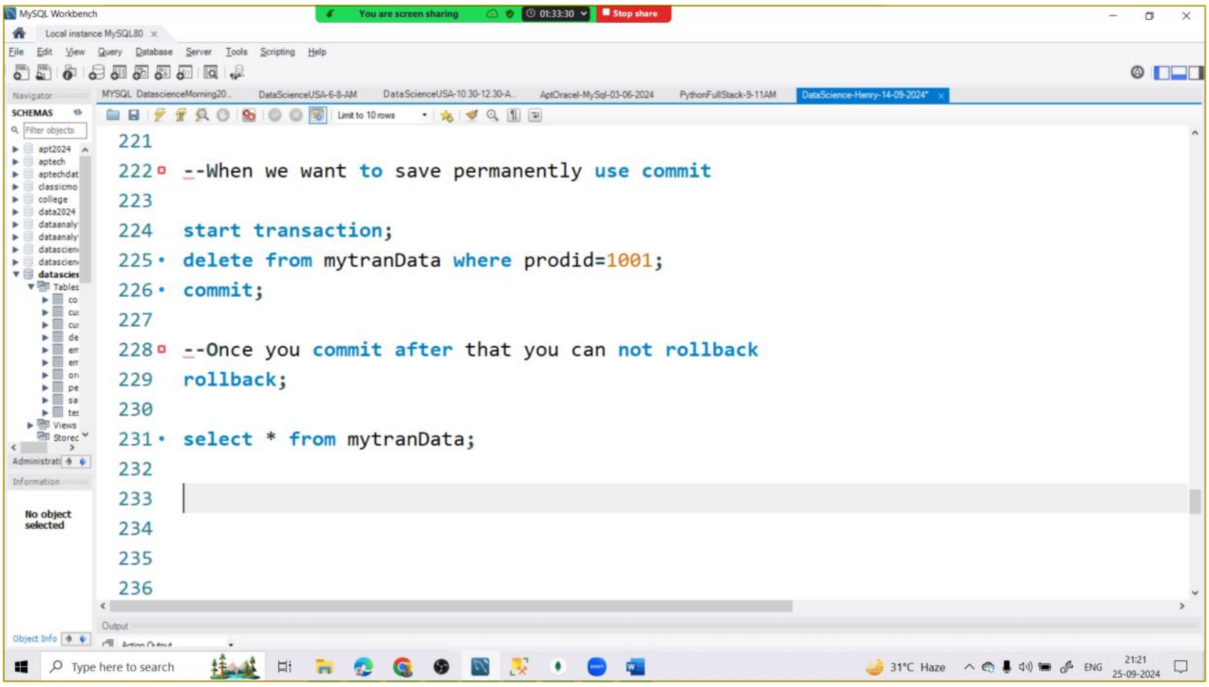
```

--group by the data according to particular column
select * from salaries;
select position,sum(salary) from salaries group by position;

--after group by show record only those salary greater 100000

select position,sum(salary) from salaries group by position having
sum(salary)>100000;

--order by use for arrange the data

select * from salaries order by salary desc;
select * from salaries order by salary asc;


```



```

--Trasaction- to run all process into single unit if any one process is
--failure so all process rollback
--ACID Property
1)-Automicity- all process run automatic
2)-Consitency- All Process run into the consitent way
3) Isolation- all process run independently
4) Duration- to complete the process within duration

use datascience2024;
Create table mytranData
(
prodid int,
prodName char(100)
);
insert into mytranData values(1001,'HP Computer'),(1002,'Dell Computer'),
```



```
prodid int,
prodName char(100)
);
insert into mytranData values(1001,'HP Computer'),(1002,'Dell Computer'),
                            (1003,'Dell Computer');
select * from mytranData;

start transaction;
delete from mytranData where prodid=1001;

--rollback use for undo
rollback;

--When we want to save permanently use commit

start transaction;
```



```

--When we want to save permanently use commit

start transaction;
delete from mytranData where prodid=1001;
commit;

--Once you commit after that you can not rollback
rollback;

select * from mytranData;
```