|  |
| --- |
| Police  Station  Database |

|  |
| --- |
| DONE BY:  B.Sai Kumar Reddy  IT-B  1602-19-737-094 |

ABSTRACT:-

**Police Station Database** is platform which enables the police officers to store the important records of criminal activities. This Software stores all the information from Thieves to court which includes all the details of police officers ,thieves ,FIR ,jails and lot more.

So that a police officer can easily track the records of criminals within seconds and it is user friendly to insert all the records And it also establish many relations among the tables with relevant and meaning full information

**REQUIREMENT ANALYSIS:**

List Of Tables:

* UserDetails
* Officer
* FIR
* Thieves
* Jail
* Handle

**List of attributes with their domain types :-**

UserDetails:

1. UserId number
2. UserName varchar2(20)
3. PassWord varchar2(20)
4. OfficerId integer

Officer:

1. OfficerId number
2. Name varchar2(20)
3. phoneno number
4. UserId number

FIR:

1. FirId number
2. NameOfInformer varchar2(10)
3. NameOfCriminal varchar2(10)
4. DescriptionOfOffence varchar2(100)

Thieves:

1. ThievesId number
2. Name varchar2(10)
3. Reason varchar2(50)
4. JailId number not null

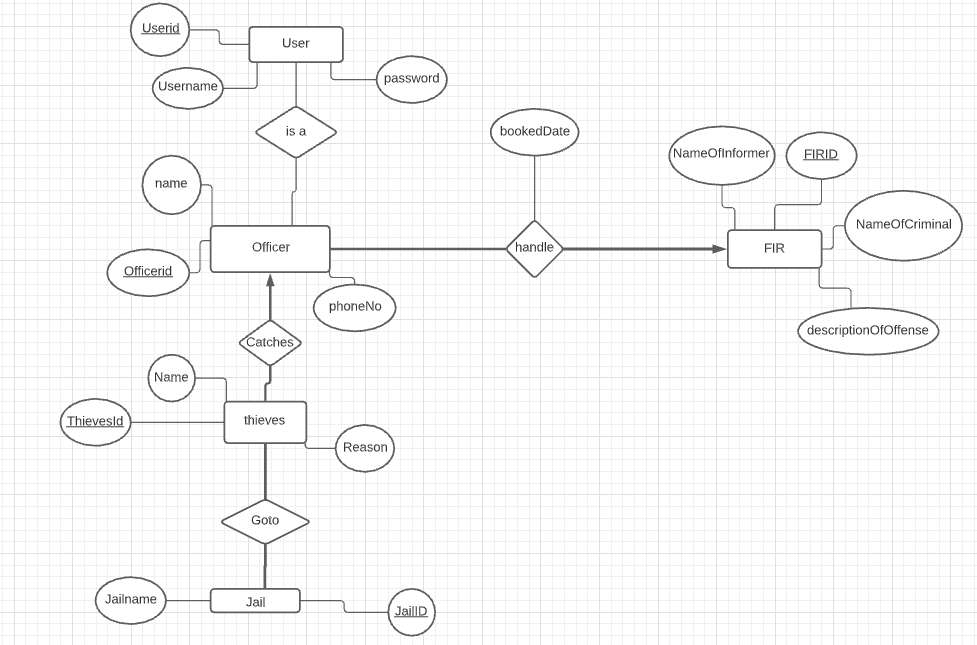
Jail:

1. JailId number
2. jailname varchar2(10)
3. ThievesId number not null

Handle:

1. OfficerId
2. FIRID
3. BookedDate

**ER DIAGRAM:**



**MAPPING CARDINALITY AND PARTICIATION CONSTRAINTS**

Many Officers can handle any of the FIR but One Fir can be handled by only one officer which is many to one relationship

Many thieves can go to jail and a jail can have many thieves which is many to many relationship

One officer can catch many thieves i.e one to many relationship

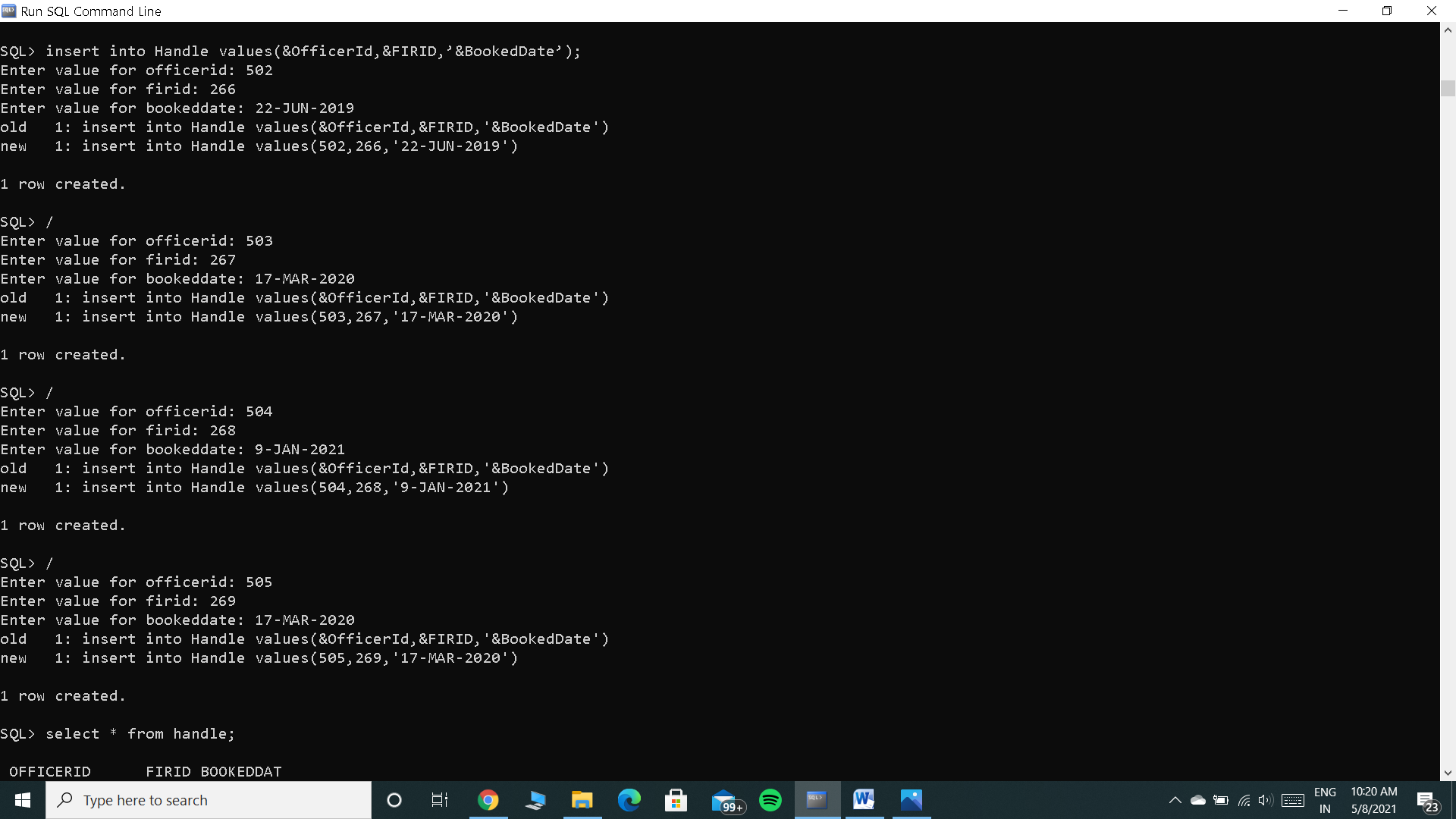
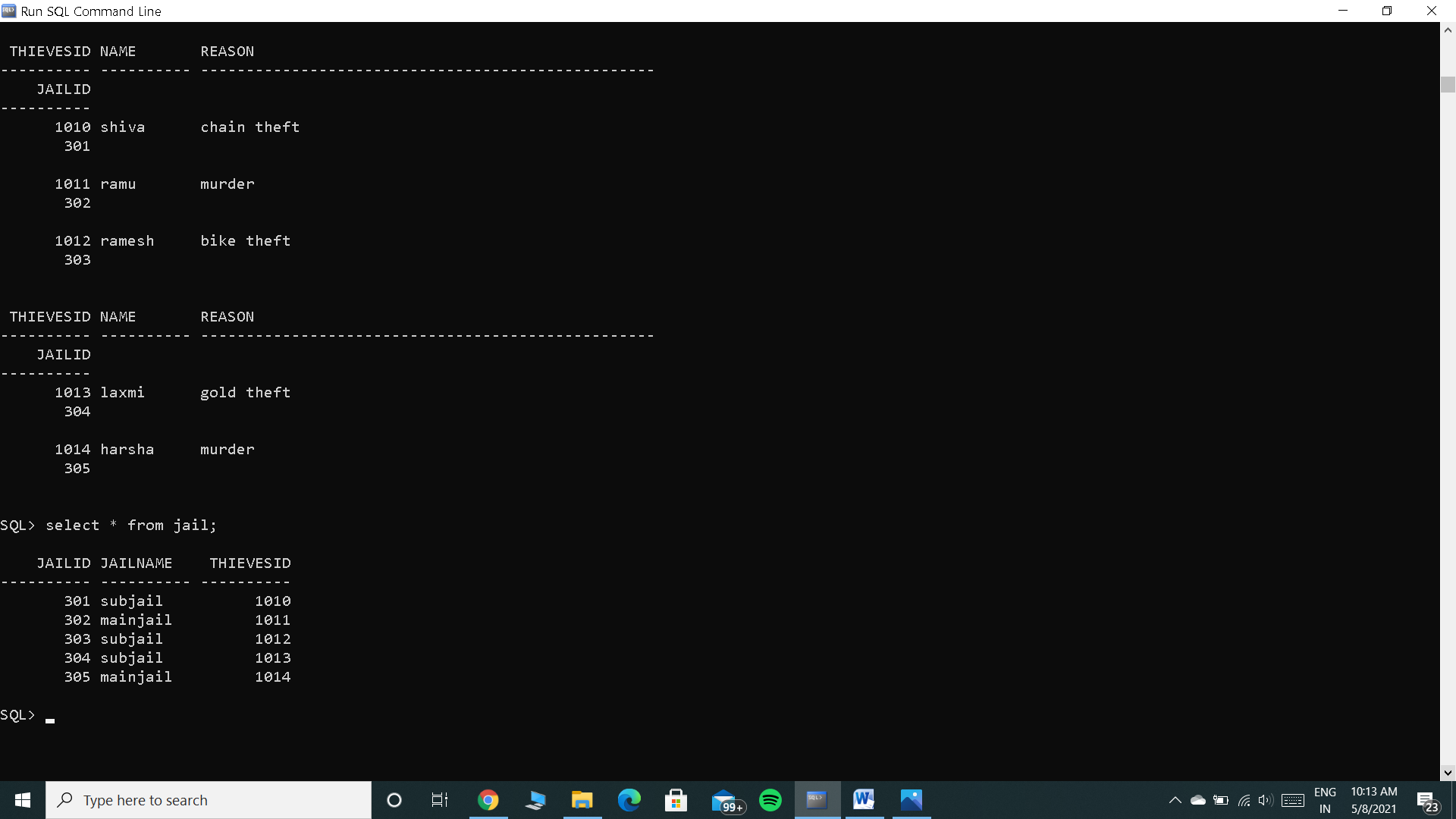
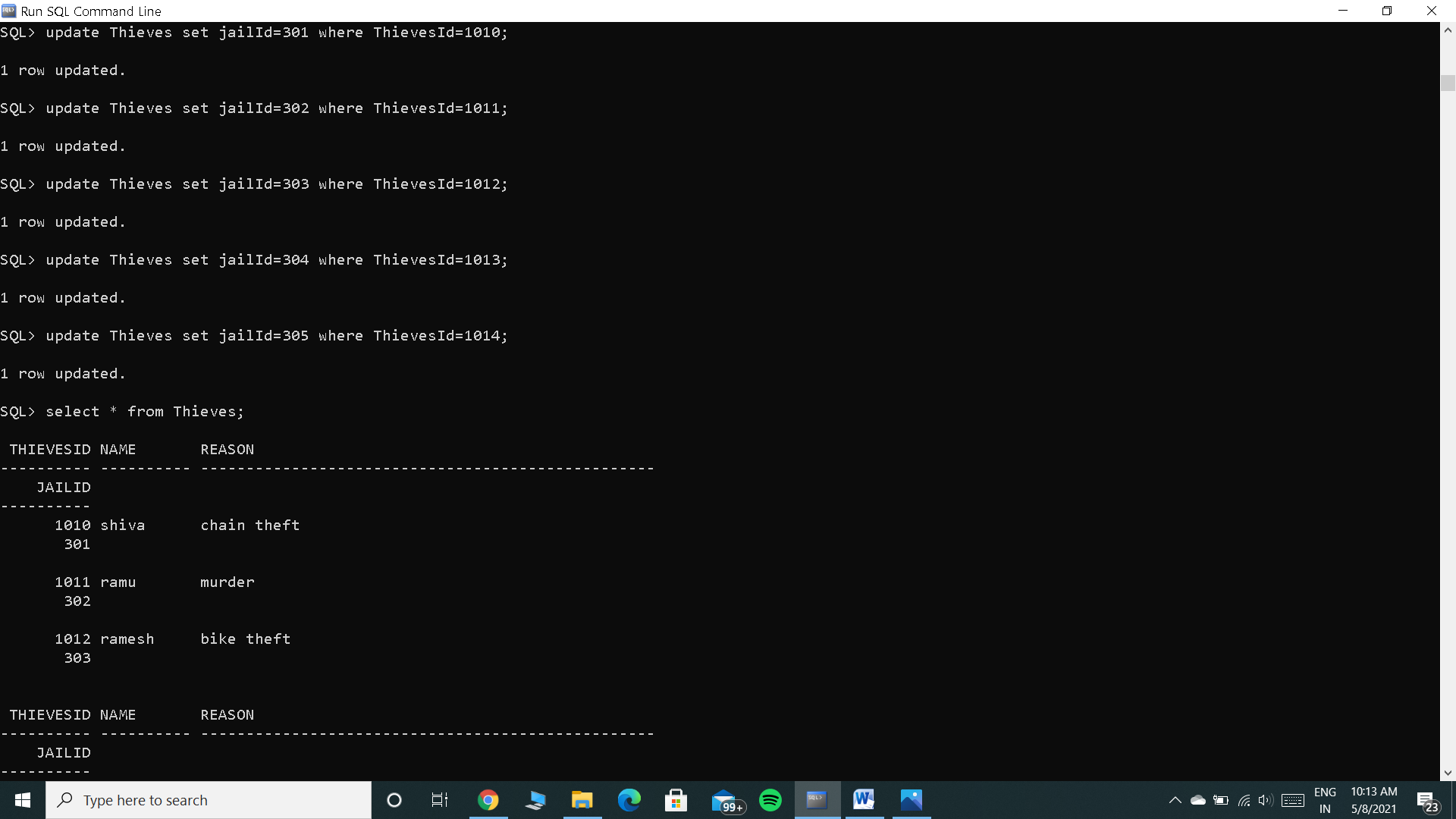
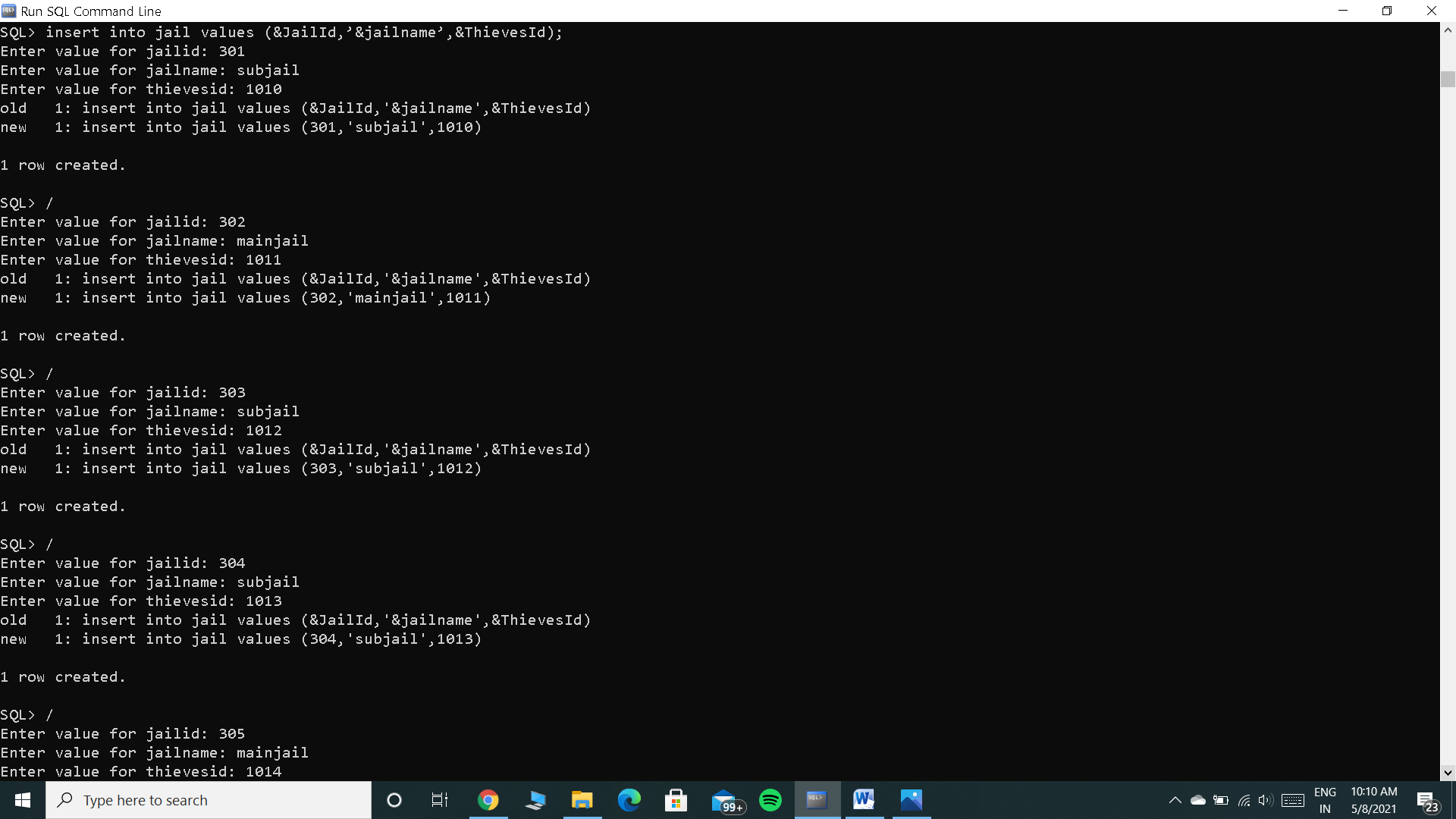
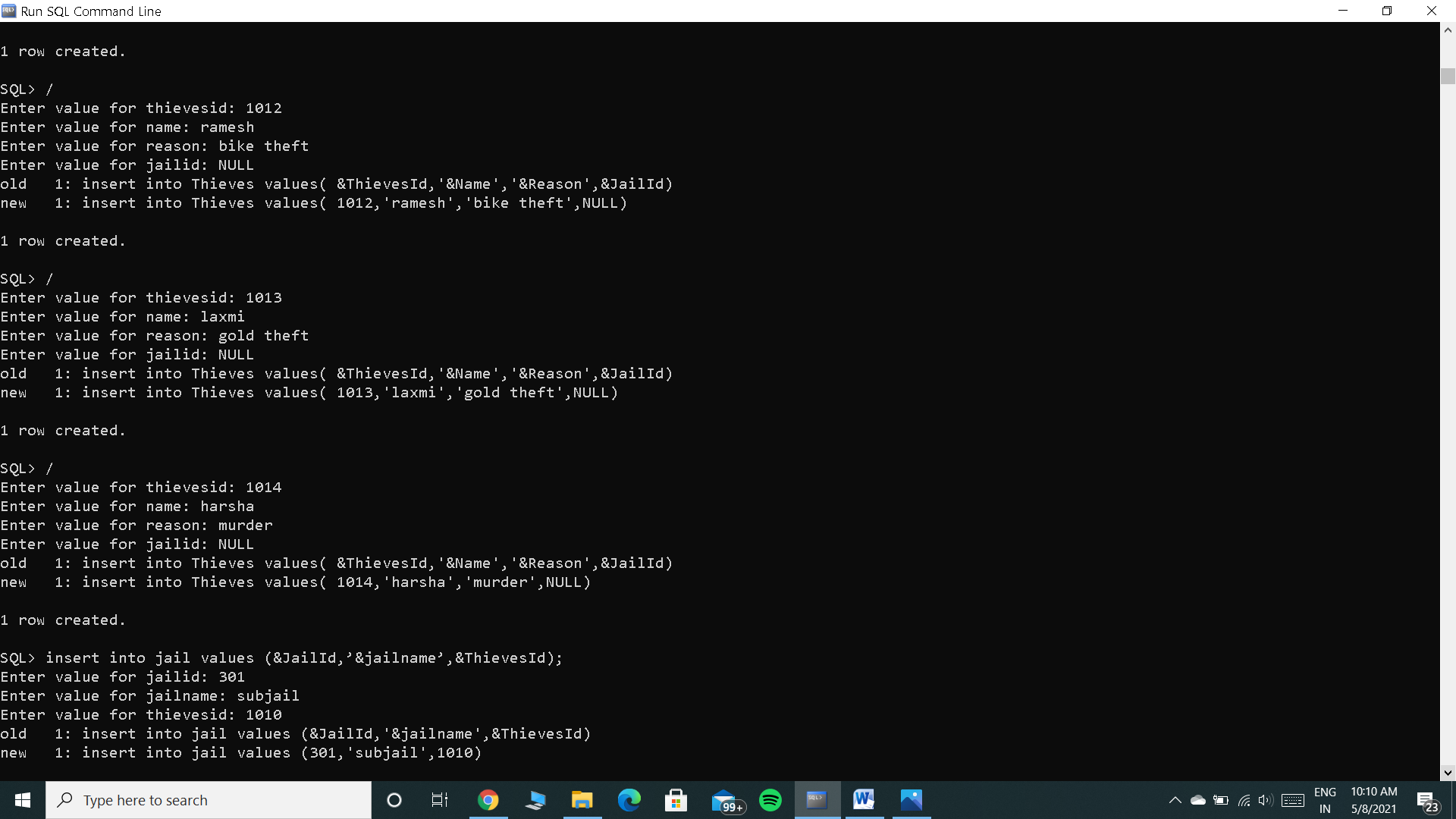
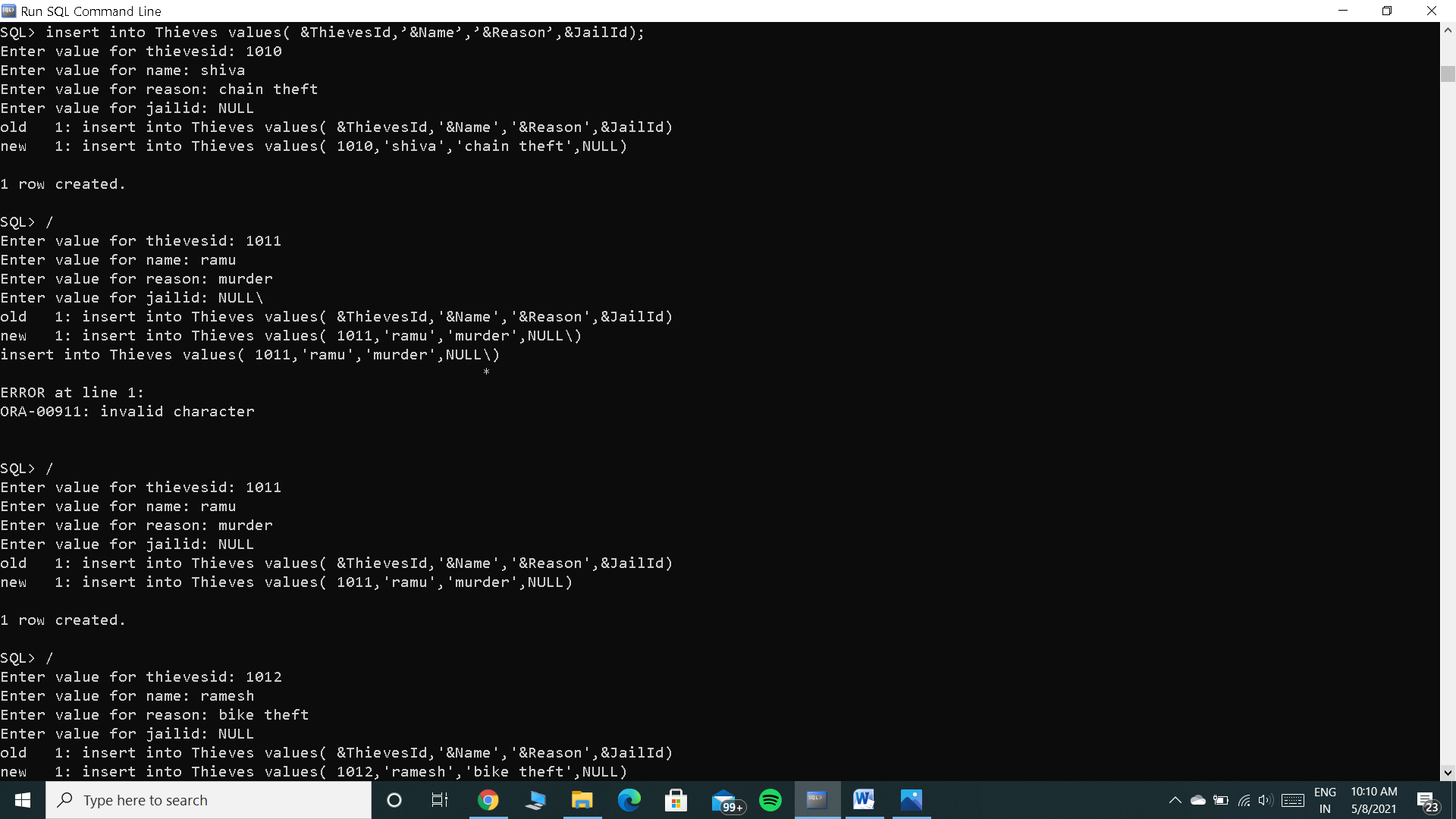
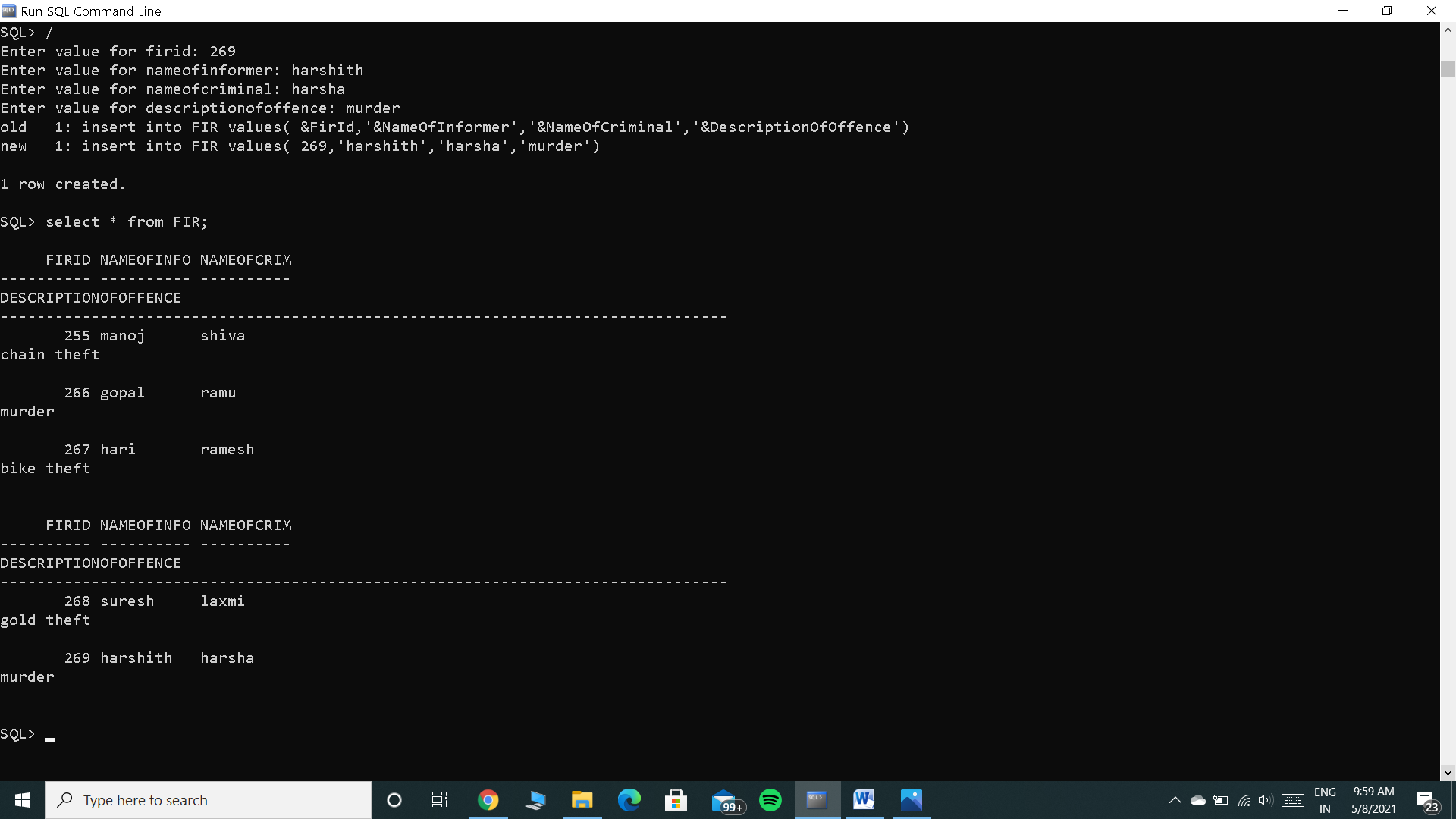
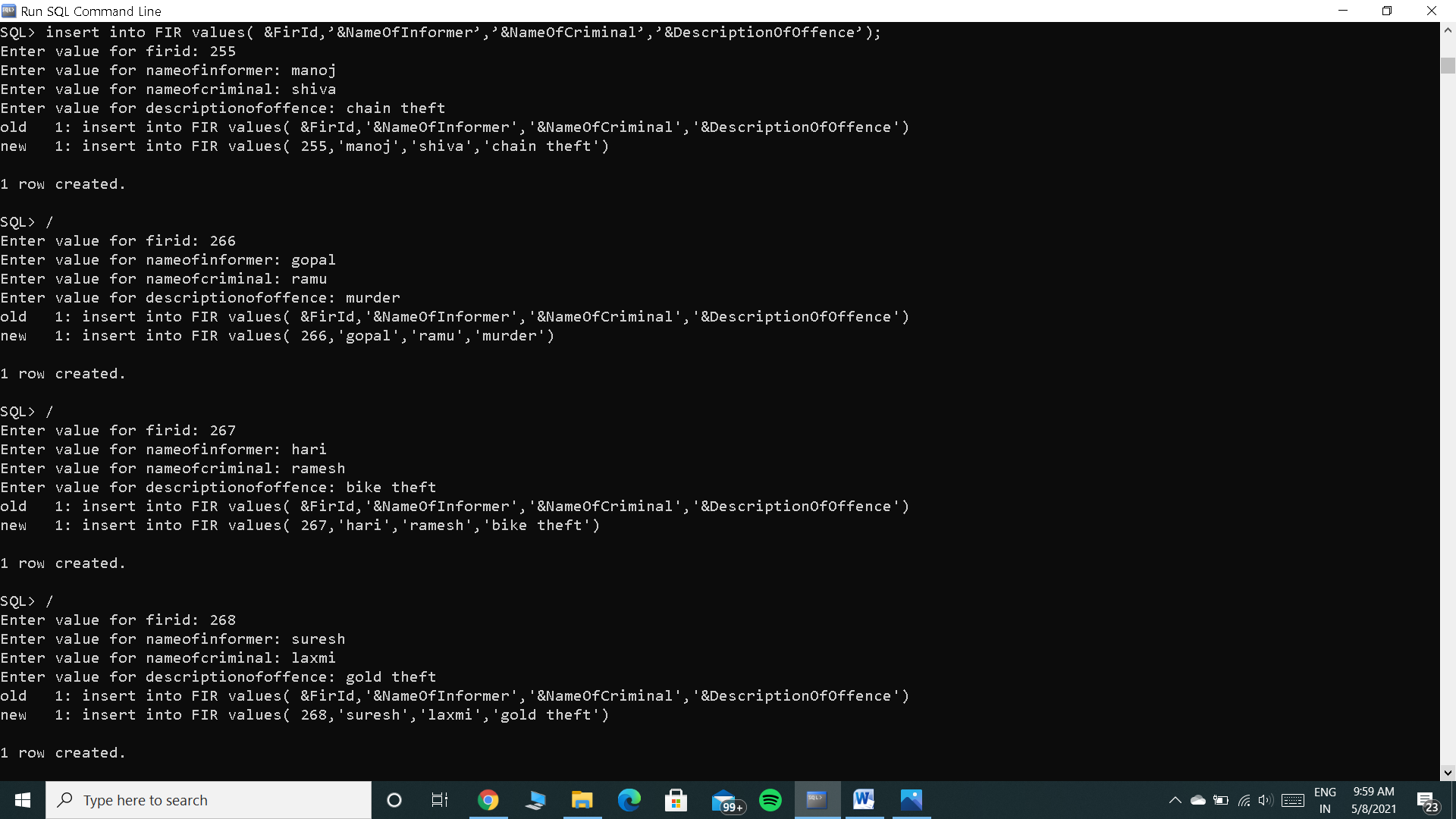
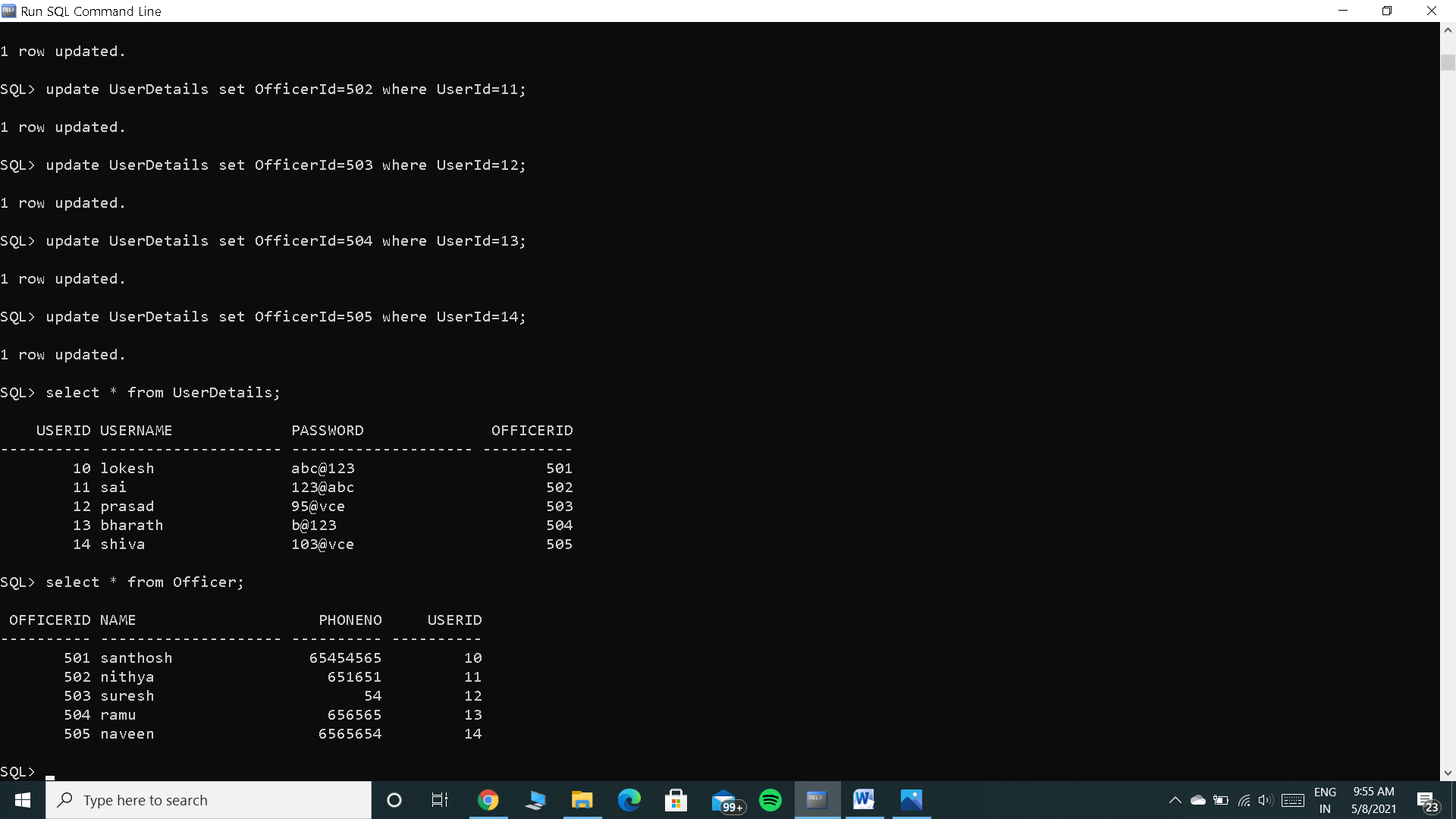
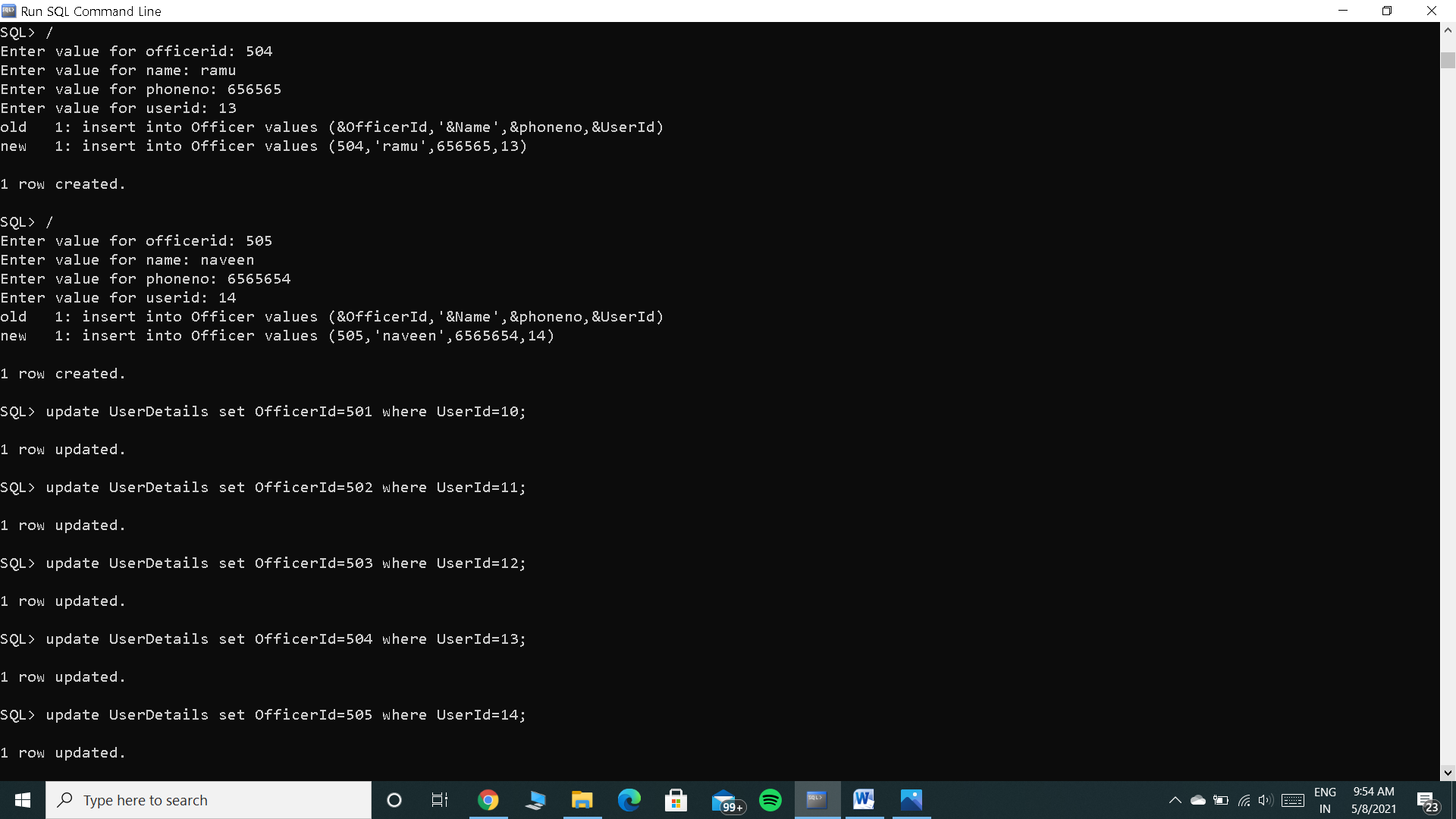
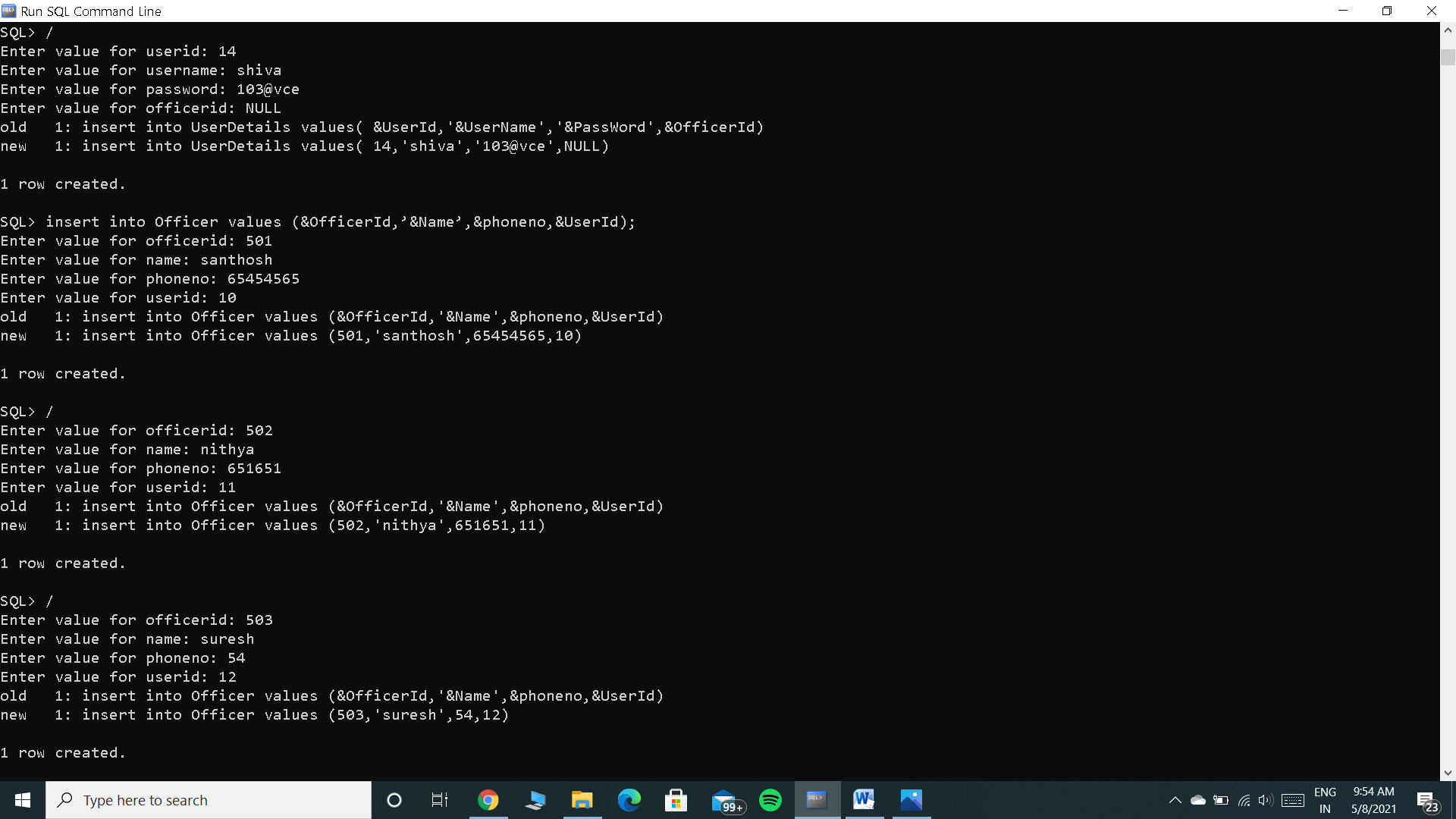
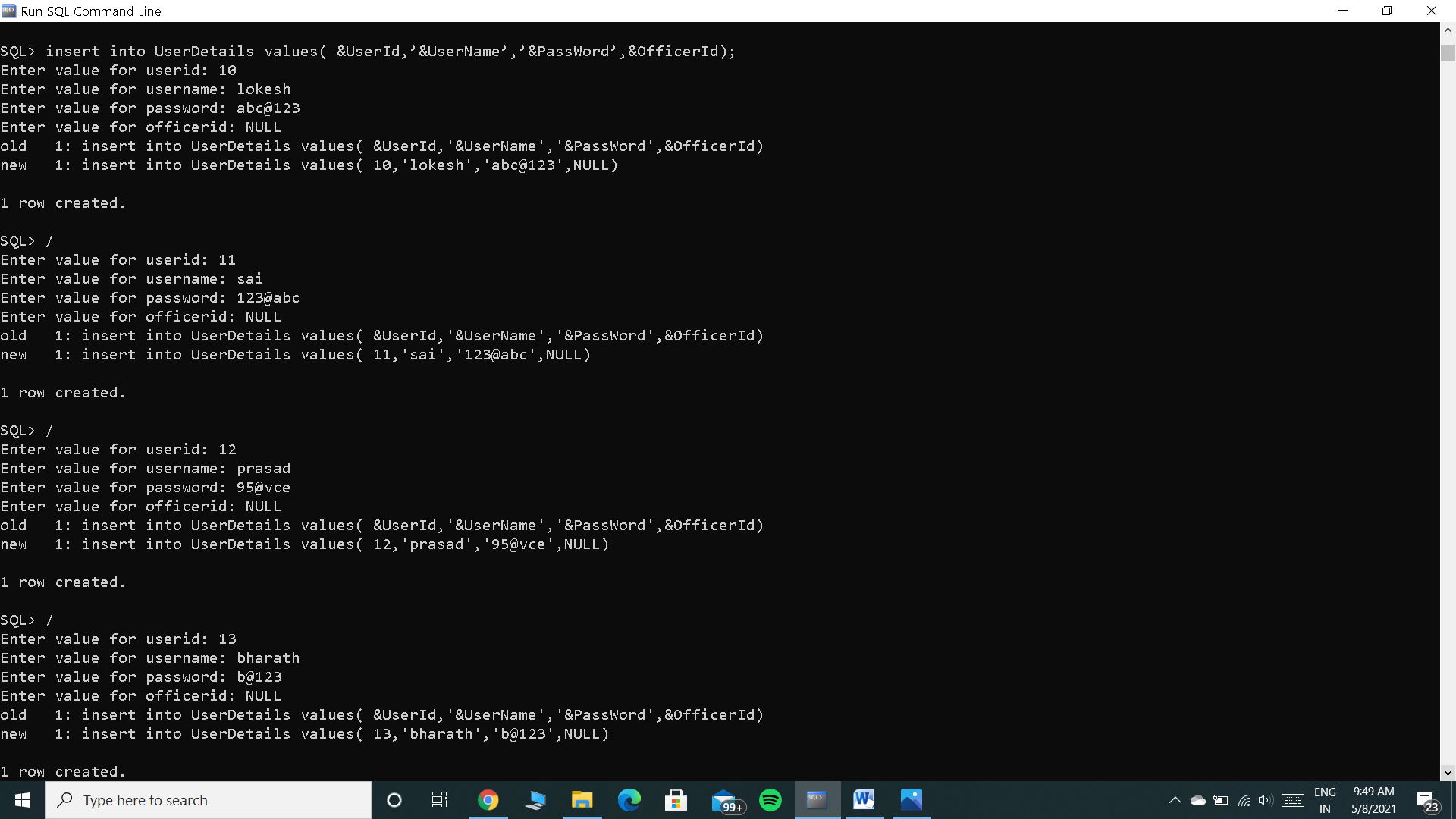
**DDL COMMANDS**

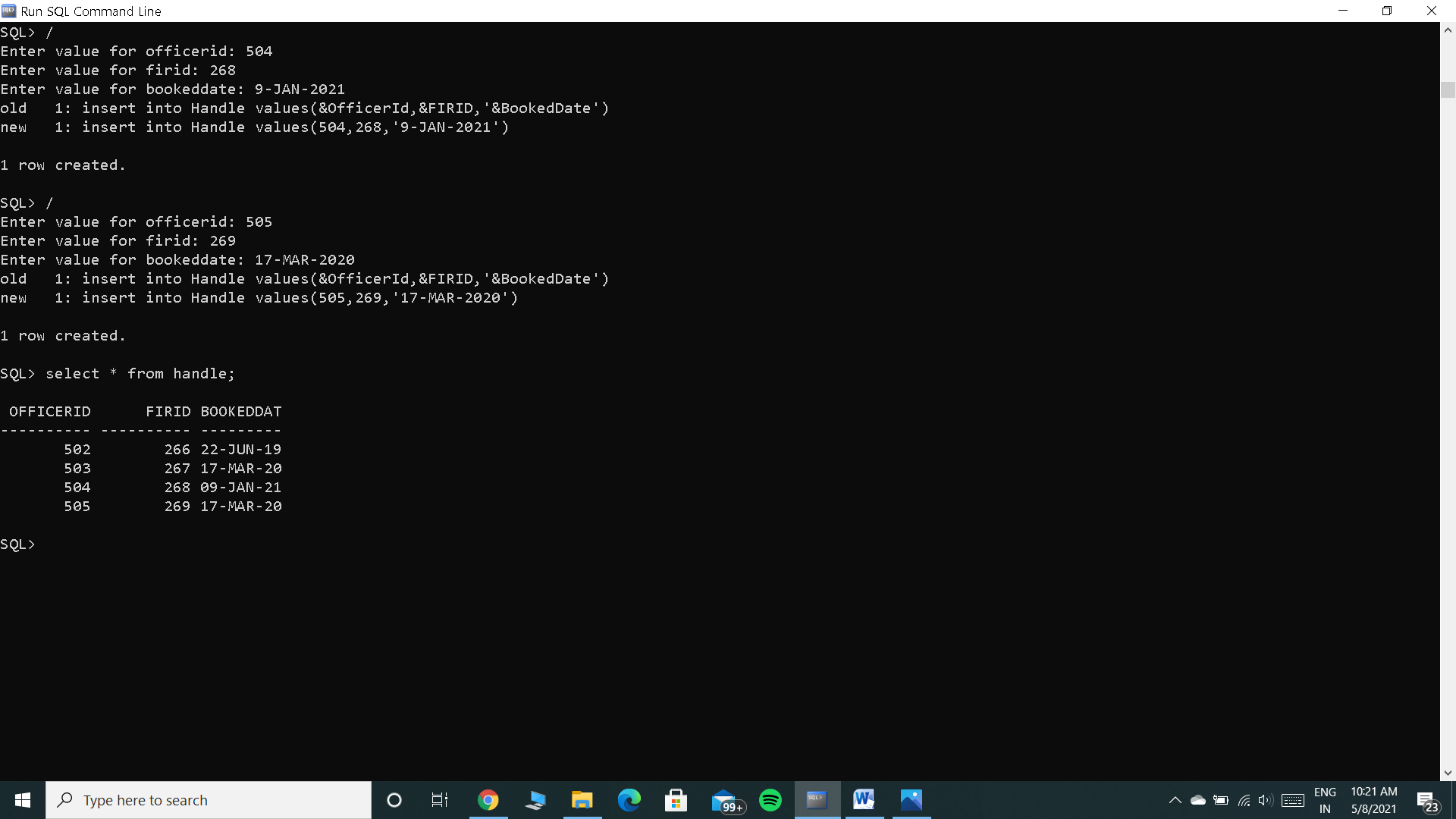
1. create table UserDetails( UserId number(3) not null primary key,UserName varchar2(20),PassWord varchar2(20) ,OfficerId integer null );
2. create table Officer( OfficerId number not null primary key,Name varchar2(20,phoneno number ,UserId number null);
3. alter table UserDetails add constraint fk\_user foreign key(OfficerId) references Officer;
4. alter table Officer add constraint fk\_Officer foreign key(UserId) references UserDetails;
5. create table FIR ( FirId number not null primary key,NameOfInformer varchar2(10) ,NameOfCriminal varchar2(10), DescriptionOfOffence varchar2(100));
6. create table Thieves( ThievesId number not null primary key,Name varchar2(10), Reason varchar2(50),JailId number null);
7. create table jail(JailId number not null primary key,jailname varchar2(10),ThievesId number not null);
8. alter table jail add constraint fk\_jail foreign key(ThievesId) references Thieves;
9. alter table Thieves add constraint fk\_Thieves1 foreign key(JailId) references jail;
10. create table Handle(OfficerId number not null,FIRID number not null,BookedDate date,foreign key(OfficerId) references Officer(OfficerId),foreign key(FIRID) references FIR(FirId));

**DML COMMANDS**

1. insert into UserDetails values( &UserId,’&UserName’,’&PassWord’,&OfficerId);
2. insert into Officer values (&OfficerId,’&Name’,&phoneno,&UserId);
3. insert into FIR values( &FirId,’&NameOfInformer ,’&NameOfCriminal’,’&DescriptionOfOffence’);
4. insert into Thieves values( &ThievesId,’&Name’ ,’&Reason’,&JailId);
5. insert into jail values (&JailId,’&jailname’,&ThievesId);
6. insert into Handle values(&OfficerId,&FIRID,’&BookedDate’);

**Outputs:**



****