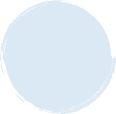
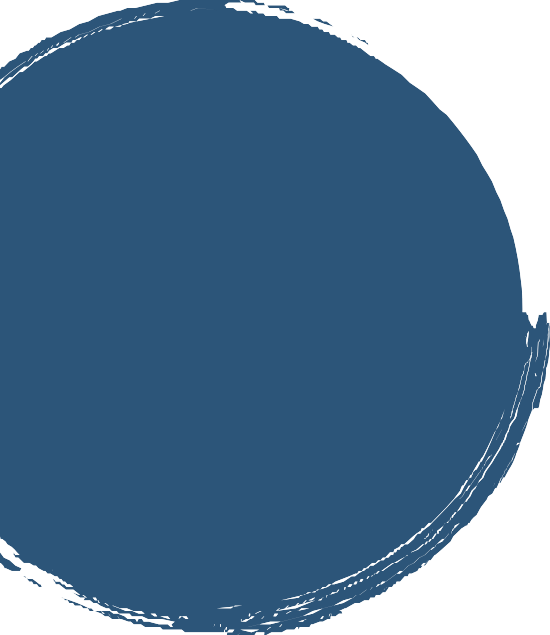
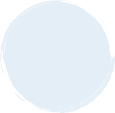
**Razib Saraff**

**ID :- 20-42949-1**



**Jailhouse Management System**



**Course Details:**

Course: Advance Database Management System.

Section: B.

Course Teacher: Juena Ahmed Noshin

**CONTENTS**

**Introduction 3**

**Scenario Description 4**

**Er Diagram 5**

**Use Case Diagram 6**

**Class Diagram 7**

**Activity Diagram 8**

**User Interface 9 - 10**

**Normalization 11 - 16**

**Schema Diagram 17**

**Table Creation 18 - 28**

**User Creation 29 - 31**

**Data Insertion 32 - 39**

**Query Writing 40 – 51**

**PL/SQL 52 - 71**

**Conclusion 72**

## Introduction:

Prison, is an institution for the confinement of persons who have been remanded (held) in custody by a judicial authority or who have been deprived of their liberty following conviction for a crime. A person found guilty of a felony or a misdemeanor may be required to serve a prison sentence. As there is crime everywhere in this world, the place prison is vastly used. Almost every city has prisons and prisoners. In today's world, managing the whole prison manually is tough as there are so much data and information. It takes a huge time to find out any criminal history, or match any data, or find out any prisoner's personal details. So, to simplify this problem our team is designing a Jailhouse Management System using a database where every prisoner all personal details such as name, age, blood group, and date of in, everything will be stored and every prisoner will also be uniquely identified by an id. Moreover, every single piece of information of visitors, constables, and officers will be stored in the management system and the officers can log in using their own id and the password to which they will have all the data access. We are very optimistic that our Jailhouse Management System will work properly to solve the difficulties.

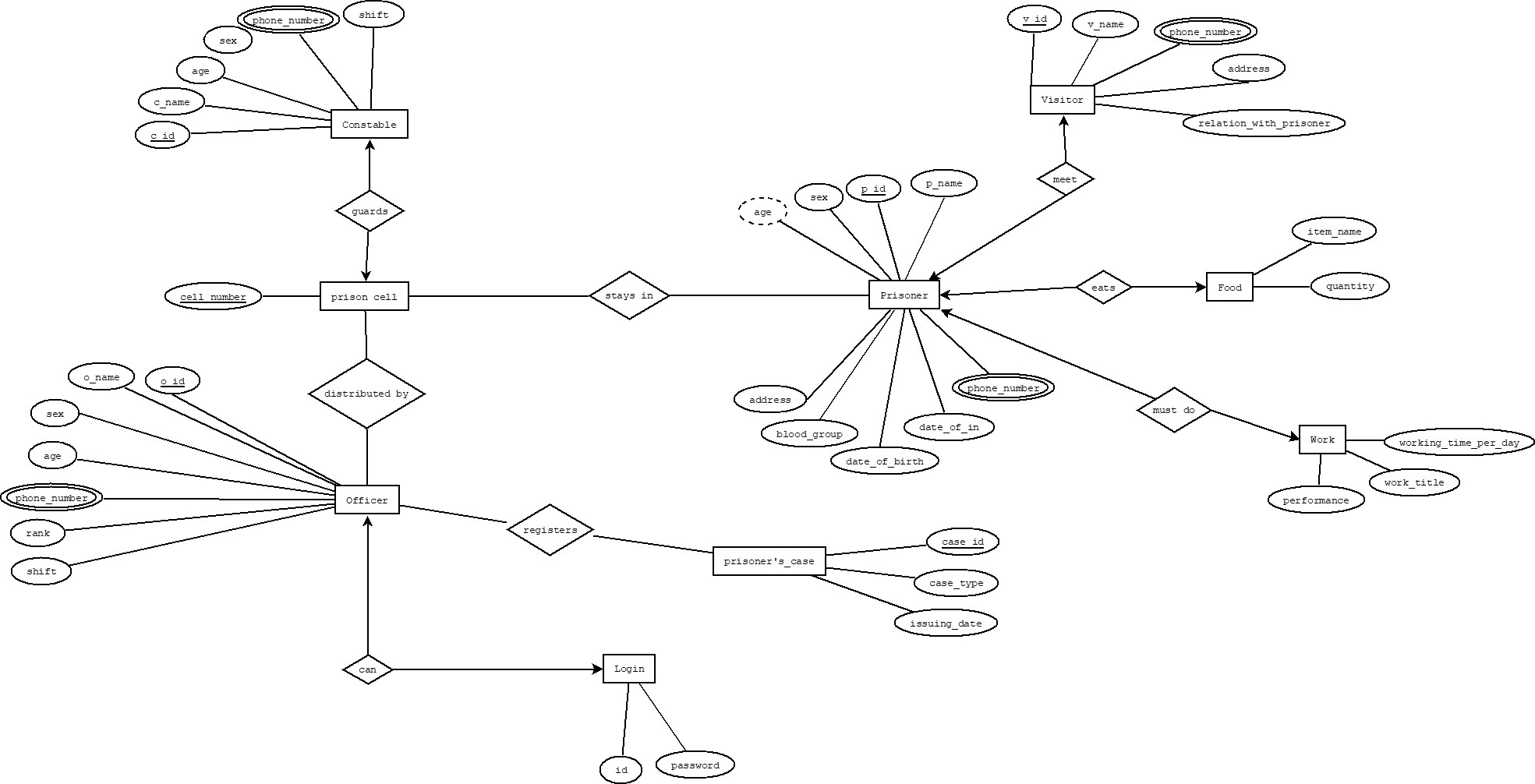
## Scenario Description:

In this Jailhouse Management System, every prisoner will be uniquely identified by the prisoner's id. The prisoner’s name, age,sex, address, blood group, phone number, date of birth, and date of in will be stored as well. The age depends on the date of birth and the phone number can be multiple. Prisoners will stay in prison cells. There is single-seated cell and multi-seated cell so one prisoner can stay in one room and multiple prisoners also can stay in one room. The database will store the cell number of prison cells. The prison cell will be distributed by an officer

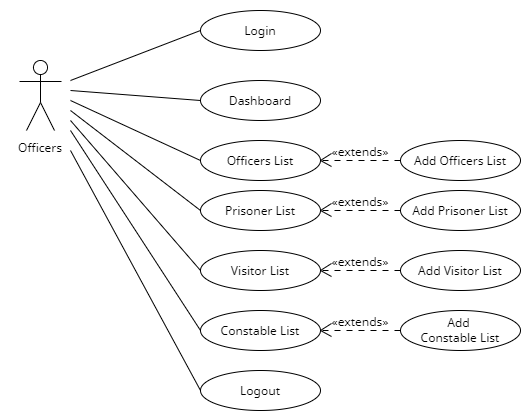
where an officer will distribute one or multiple cells between prisoners.

An officer will be identified by the officer's id. Moreover, the officer's name, sex, age, phone number (multiple), rank, and shift will be stored in the Jailhouse Management System. Each officer can register one or more prisoner's cases where he/she will identify cases using a unique case id. Case type and issuing date will be also stored in the prisoners' case document. Officers can log in to the system using their id and password where he/she will have every single access to the system. Prison cells need security so, cells will be guarded by the constable. One constable will guard one cell. Each constable will be identified by the constable id, also the constable’s name, age, sex, phone number, and shift will be stored. Phone numbers can be multiple. Every prisoner's food is well defined by the item name and quantity. Prisoners must do work. Work title, per day working time, and performance will be stored. There are always visitors who come to meet with prisoners so, for security purposes, every visitor will have one unique visitor id and the management system will store the visitor's name, address, relationship with the prisoner, and one or more phone numbers. Each prisoner can meet only one visitor and one visitor can meet one prisoner at a time.

## ER Diagram:

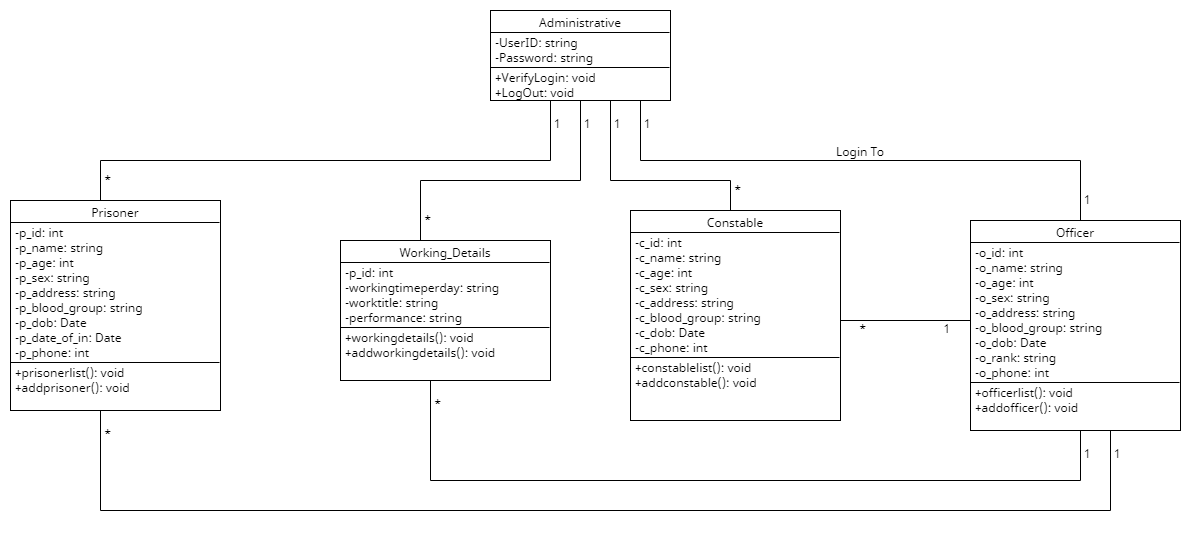


## Use Case Diagram:

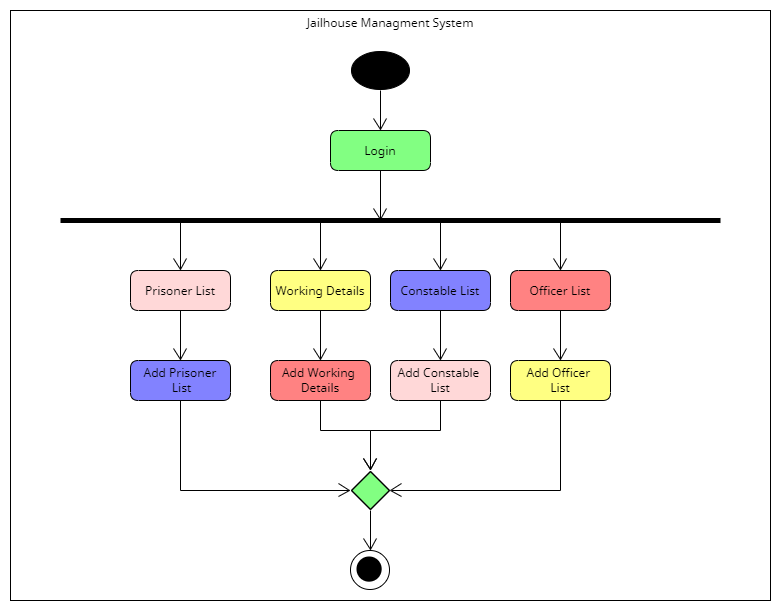


## Class Diagram:

## 

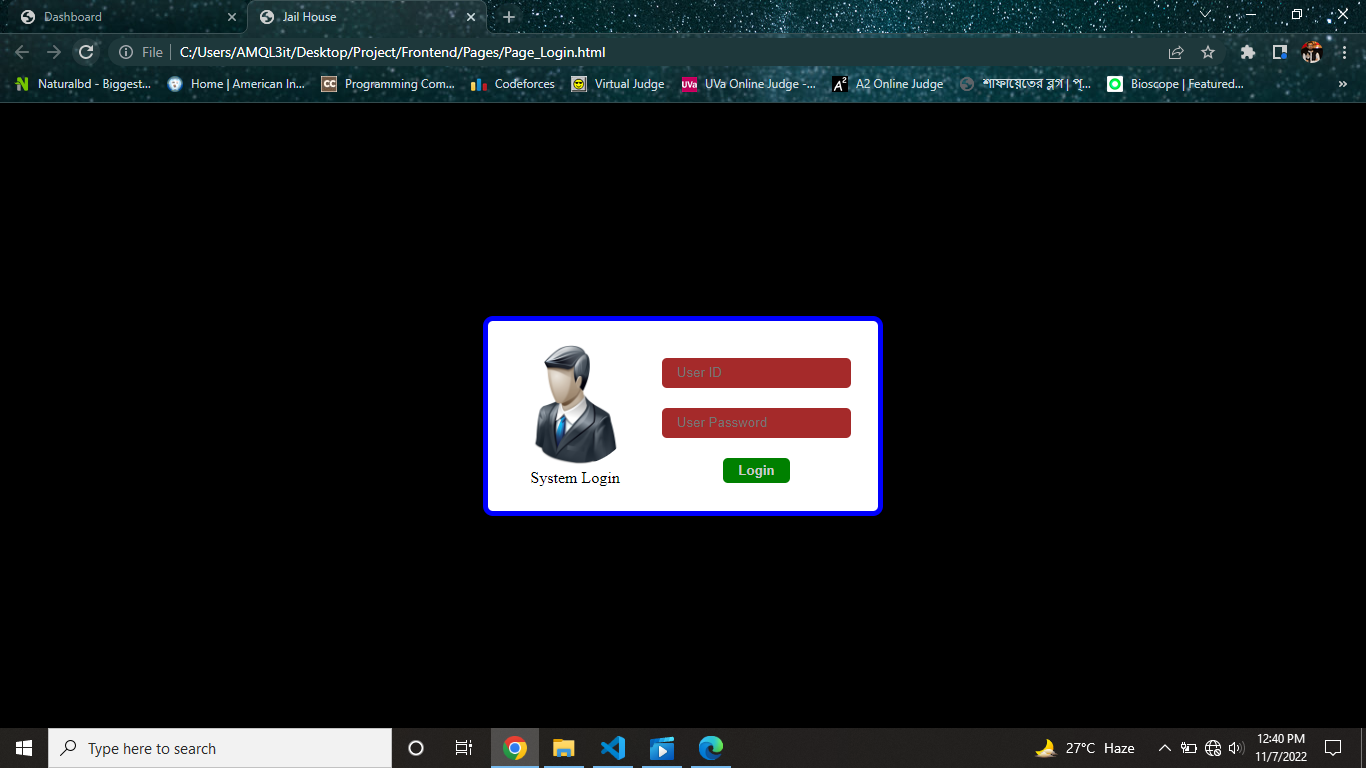


## Activity Diagram:

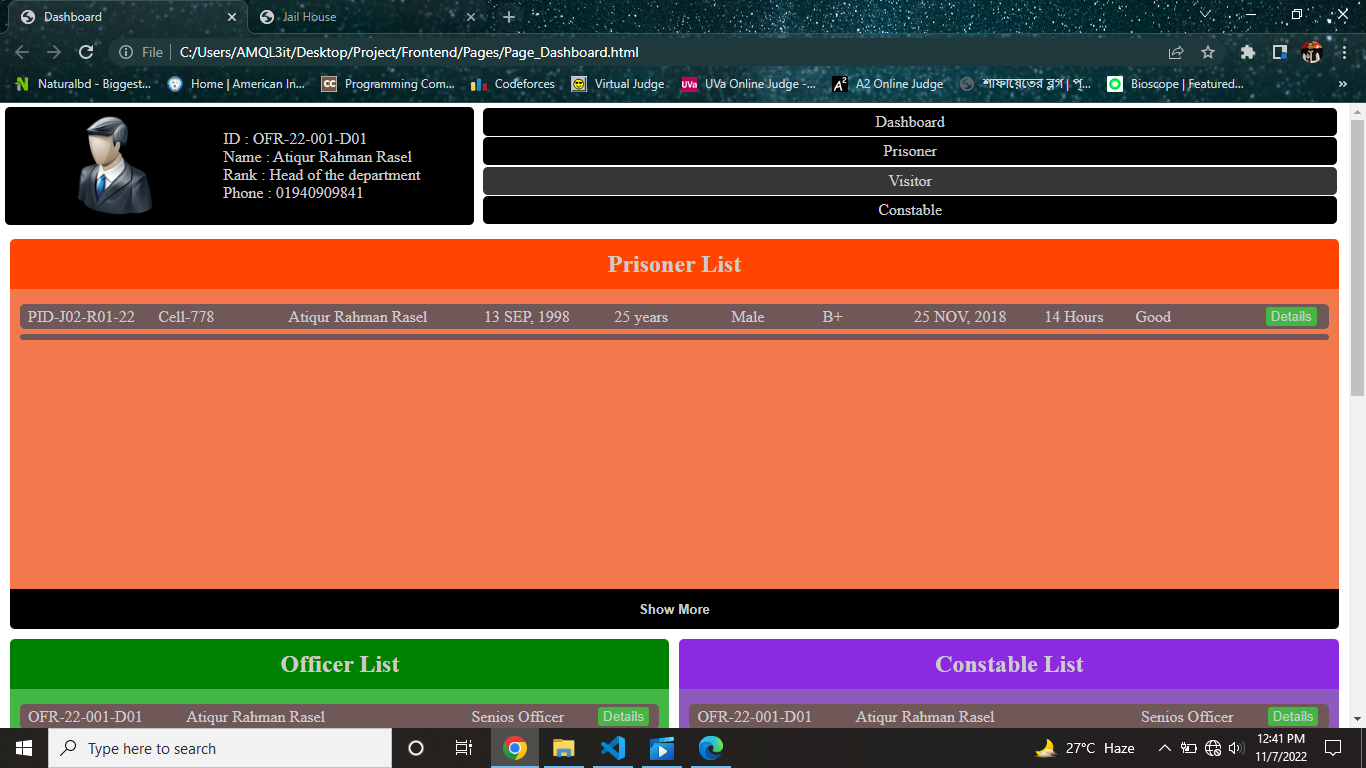


## User Interface:

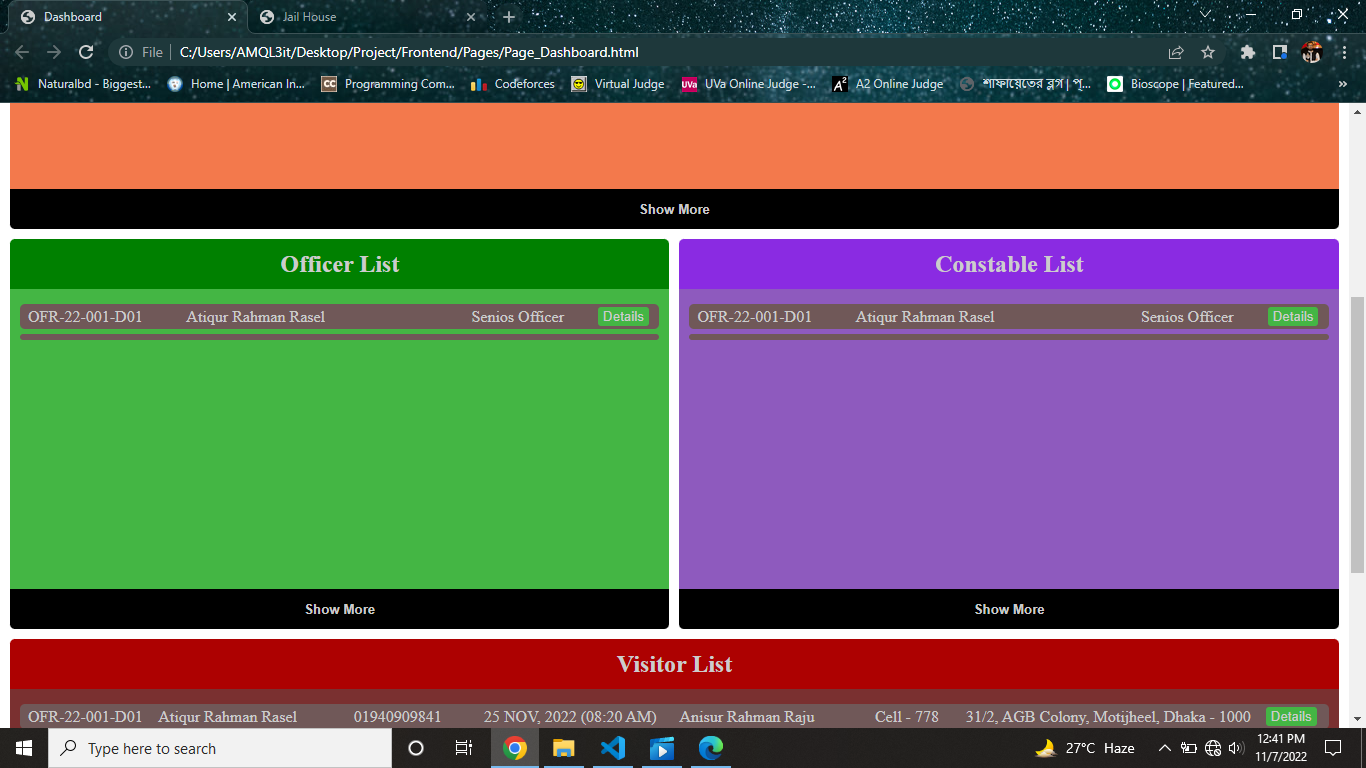
**Login Page:**



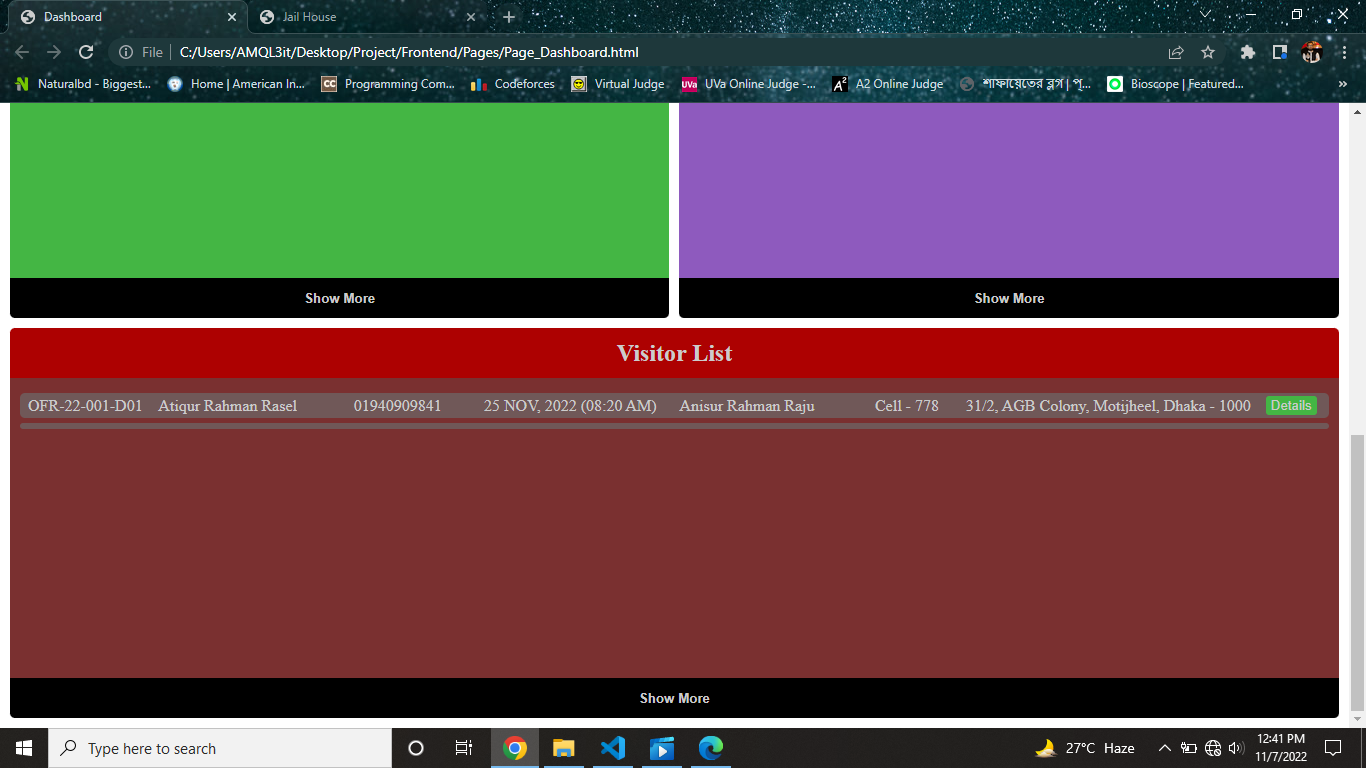
**Dashboard Page:**



**Officer and Constable List:**



**Visitor List:**



**NORMALIZATION**

**Eats**

**UNF:**

Eats( item\_name, quantity, age, sex, **p\_id**, p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number)

**1NF:**

phone\_number is a multi valued attribute.

1.item\_name, quantity, age, sex, **p\_id**, p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**2NF:**

1.item\_name, quantity

2. age, sex, **p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**3NF:**

There is no transitive dependency. Relation already in 3NF.

1.item\_name, quantity

2. age, sex, **p\_id**, p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**Table Creation:**

1.item\_name, quantity , **a\_id**

2. age, sex, **p\_id**, p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number, **a\_id**

**Meet**

**UNF:**

Meet(age, sex, **p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number, **v\_id,** v\_name, address, relation\_with\_prisoner)

**1NF:**

phone number is a multi valued attribute.

1. age, sex,**p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number, **v\_id,** v\_name, address, relation\_with\_prisoner

**2NF:**

1. age, sex, **p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number
2. phone\_number, **v\_id,** v\_name, address, relation\_with\_prisoner

**3NF:**

There is no transitive dependency, Relation already in 3NF

1. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number,

1. phone\_number, **v\_id,** v\_name, address, relation\_with\_prisoner,

**Table Creation:**

1. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number,
2. phone\_number, **v\_id,** v\_name, address, relation\_with\_prisoner,aid

**Must do** **UNF:**

must do(working\_time\_per\_day, work\_title, performance, age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number )

**1NF:**

Phone number is a multi valued attribute .

1. working\_time\_per\_day, work\_title, performance, age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**2NF:**

1. working\_time\_per\_day, work\_title, performance
2. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**3NF:**

There is no transitive dependency. Relation already in 3NF

1. working\_time\_per\_day, work\_title, performance
2. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**Table Creation:**

1. working\_time\_per\_day, work\_title, performance, **a\_id**

**Stays In**

**UNF:**

1. Stays in(cell\_number, age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number )

**1NF:**

Phone number is a multi valued attribute.

1. cell\_number, age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**2NF:**

1.cell\_number

1. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**3NF:**

There is no transitive dependency. Relation already in 3NF.

1. cell\_number
2. age, sex**, p\_id,** p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number

**Table Creation:**

1. cell\_number, **p\_id**

**Guards**

**UNF:**

Guards(cell\_number, **c\_id,** c\_name, age, sex, phone\_number, shift)

**1NF:**

Phone number is a multi-valued attribute.

1. cell\_number, **c\_id,** c\_name, age, sex, phone\_number, shift

**2NF:**

1.cell\_number

1. **c\_id,** c\_name, age, sex, phone\_number, shift

**3NF:**

There is no transitive dependency. Relation already in 3NF.

1.cell\_number

2.**c\_id,** c\_name, age, sex, phone\_number, shift

**Table Creation:**

1.cell\_number, **c\_id**

2.**c\_id**, c\_name, age, sex, phone\_number, shift

**Distributed by**

**UNF:**

Distributed by(cell\_number, shift, rank, phone\_number, age, sex, o\_name, **o\_id** )

**1NF:**

Phone number is a multi valued attribute.

1. cell\_number, shift, rank, phone\_number, age, sex, o\_name, **o\_id**

**2NF:**

1. cell\_number
2. shift, rank, phone\_number, age, sex, o\_name, **o\_id**

**3NF:**

There is no transitive dependency. Relation already in 3NF.

**Table Creation:**

1. cell\_number, **o\_id**
2. shift, rank, phone\_number, age, sex, o\_name, **o\_id**

**Can**

**UNF:**

Can(shift, rank, phone\_number, age, sex, o\_name, **o\_id,** id, password)

**1NF:**

Phone number is a multi valued attribute.

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id,** id, password.

**2NF:**

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
2. id, password

**3NF:**

There is no transitive dependency. Relation already in 3NF.

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
2. id, password

**Table Creation**

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
2. id, password, **o\_id**

**Registers**

**UNF:**

1. Registers(shift, rank, phone\_number, age, sex, o\_name, **o\_id**, case\_id, case\_type, issuing\_date)

**1NF:**

Phone number is a multi-valued attribute.

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id,** case\_id, case\_type, issuing\_date

**2NF:**

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
2. **case\_id,** case\_type, issuing\_date

**3NF:**

There is no transitive dependency. Relation already in 3NF

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
2. **case\_id**, case\_type, issuing\_date

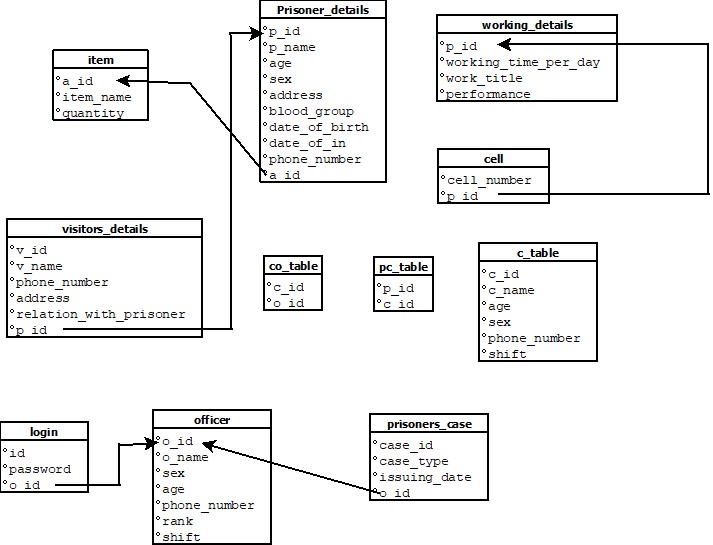
**Table Creation:**

1. shift, rank, phone\_number, age, sex, o\_name, **o\_id.**
2. **case\_id,** case\_type, issuing\_date, **o\_id**

**Final Tables:**

1. item\_name, quantity , **a\_id**
2. age, sex, **p\_id**, p\_name, address, blood\_group, date\_of\_birth, date\_of\_in, phone\_number , **a\_id**
3. phone\_number, **v\_id,** v\_name, address, phone number, relation\_with\_prisoner,aid
4. working\_time\_per\_day, work\_title, performance, **p\_id**
5. cell\_number, **p\_id**
6. **p\_id**, **c\_id**
7. **c\_id,** c\_name, age, sex, phone\_number, shift
8. **c\_id**, o\_id
9. shift, rank, phone\_number, age, sex, o\_name, **o\_id**
10. id, password, **o\_id**
11. **case\_id,** case\_type, issuing\_date, **o\_id**

## Schema Diagram:



**Table Creation**

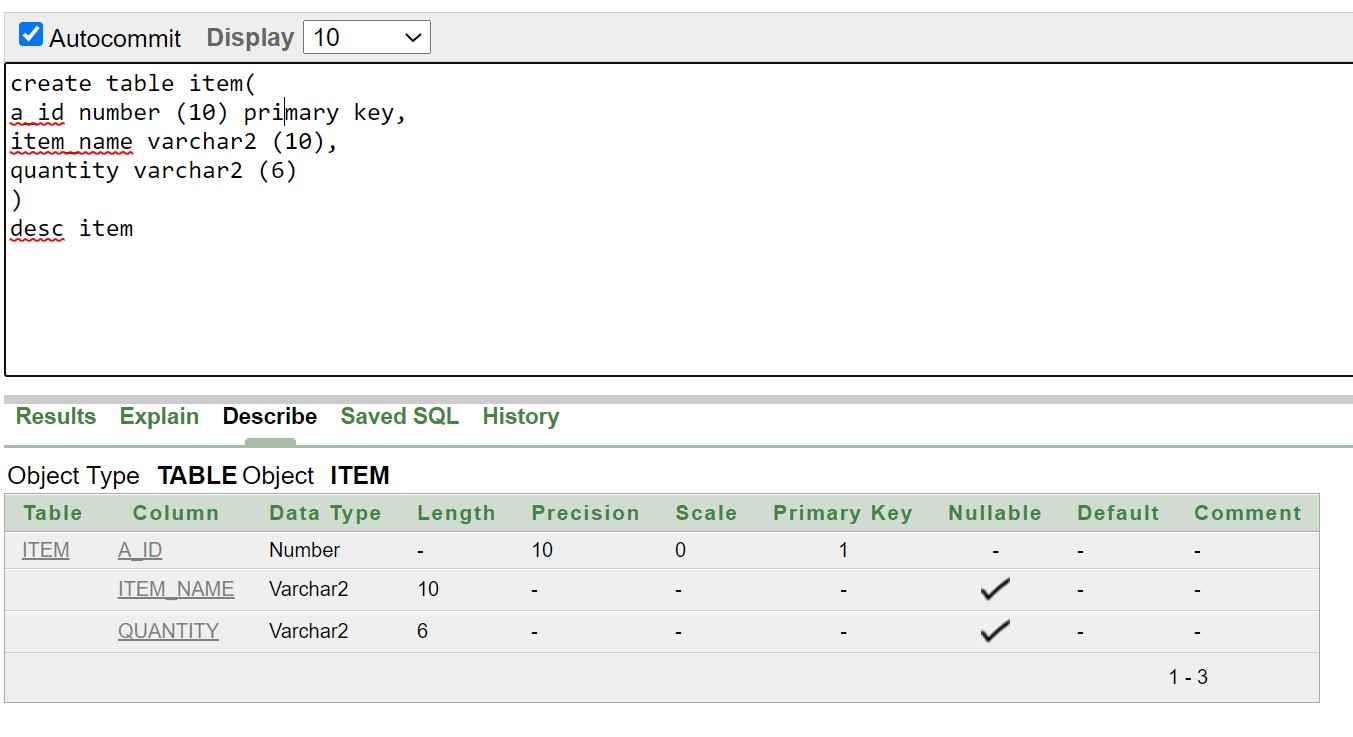
**1.**

create table item( a\_id number (10) primary key, item\_name varchar2 (10), quantity varchar2 (6)

)

Alter table item modify(item\_name varchar2 (20))

Alter table item modify(quantity varchar2 (20))



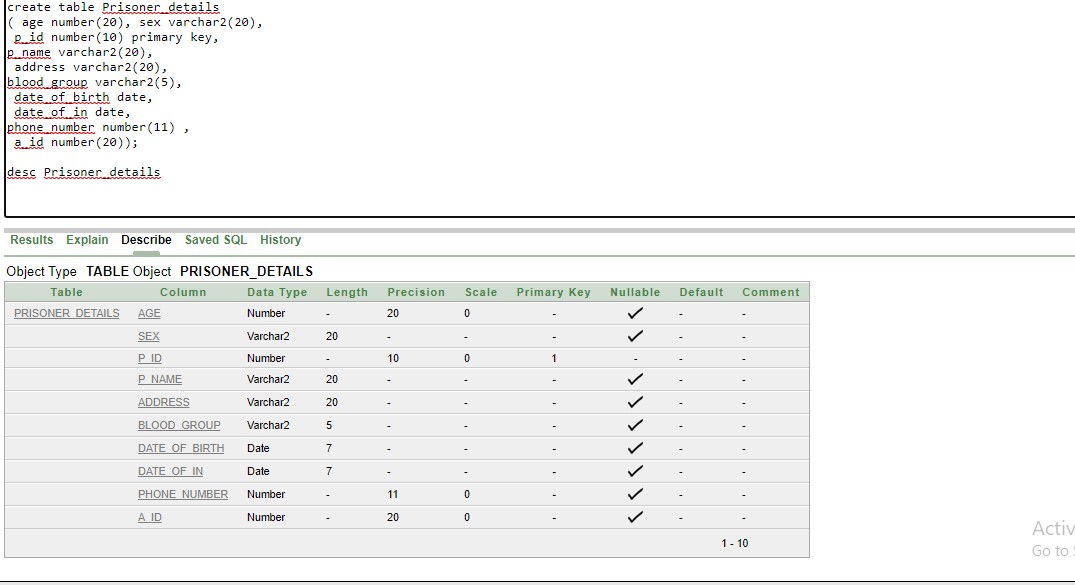
**2.**

create table Prisoner\_details (age number(20), sex varchar2(20), p\_id number(10) primary key, p\_name varchar2(20), address varchar2(20), blood\_group varchar2(5), date\_of\_birth date, date\_of\_in date,

phone\_number number(11) ,

a\_id number(20));

alter table Prisoner\_details add constraint a\_id\_fk foreign key(a\_id) references item (a\_id)

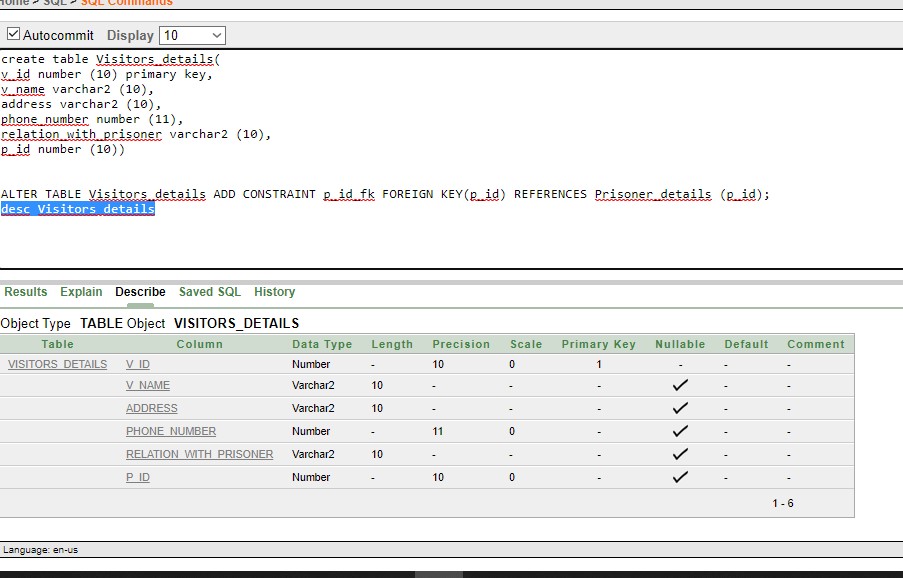


**3.**

create table Visitors\_details( v\_id number (10) primary key, v\_name varchar2 (10), address varchar2 (10),

phone\_number number (11), relation\_with\_prisoner varchar2 (10), p\_id number (10))

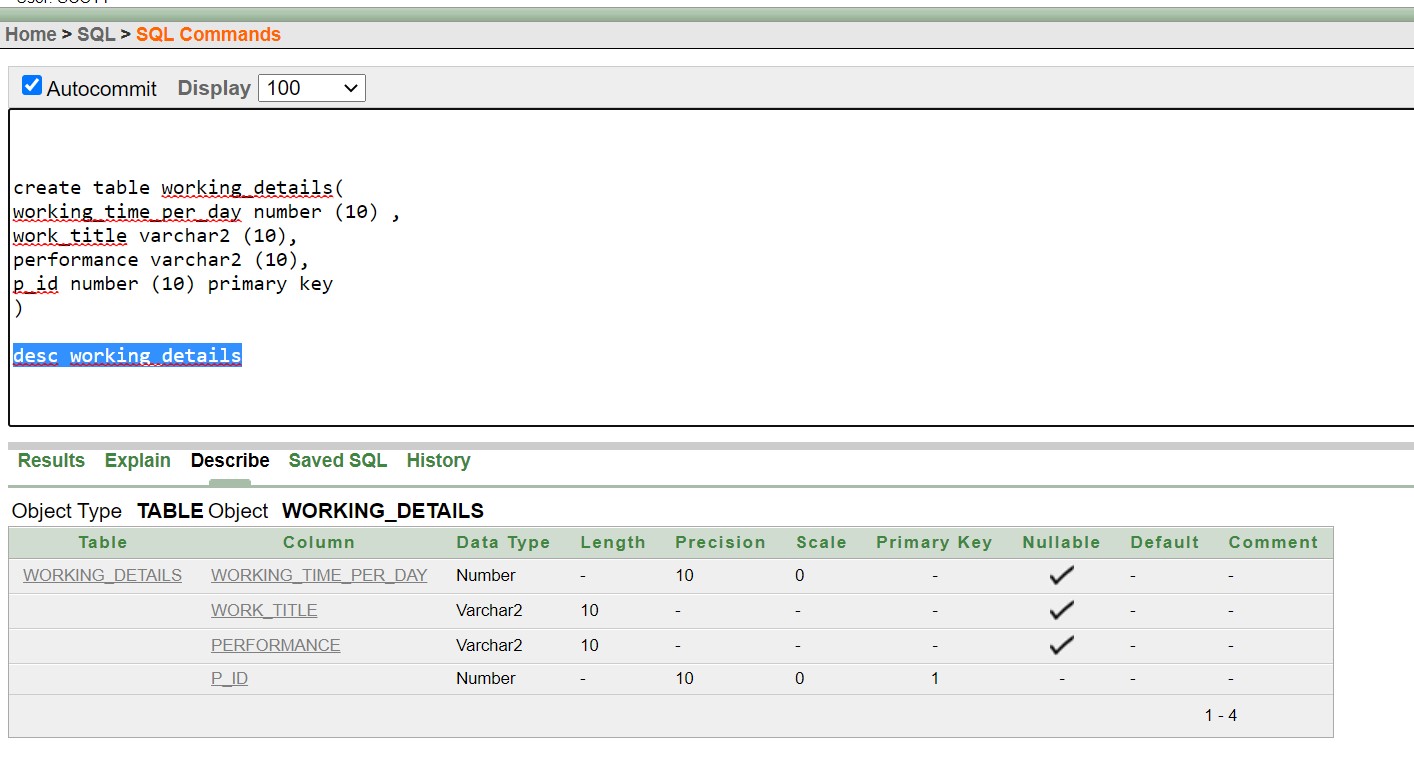
ALTER TABLE Visitors\_details ADD CONSTRAINT p\_id\_fk FOREIGN KEY(p\_id) REFERENCES Prisoner\_details (p\_id);



**4.**

create table working\_details( working\_time\_per\_day number (10) , work\_title varchar2 (10), performance varchar2 (10), p\_id number (10) primary key

)



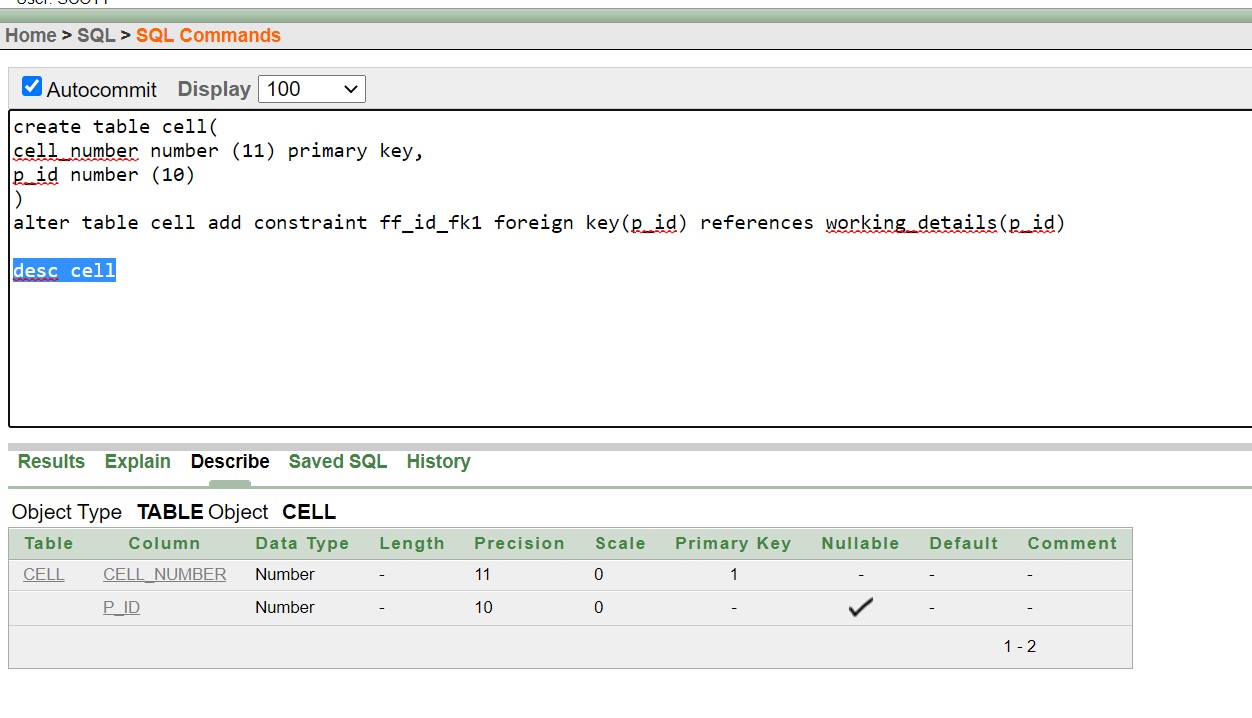
**5.**

create table cell(

cell\_number number (11) primary key, p\_id number (10)

)

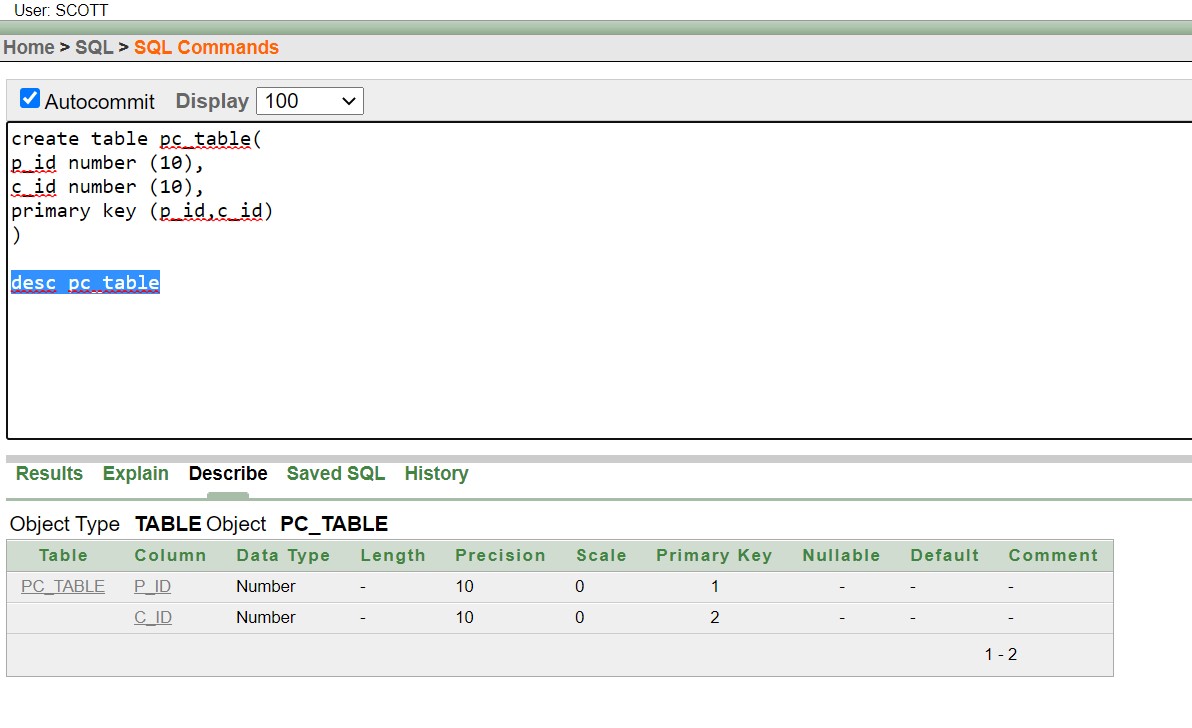
alter table cell add constraint ff\_id\_fk1 foreign key(p\_id) references working\_details(p\_id)



**6.**

create table pc\_table( p\_id number (10), c\_id number (10), primary key (p\_id,c\_id)

)



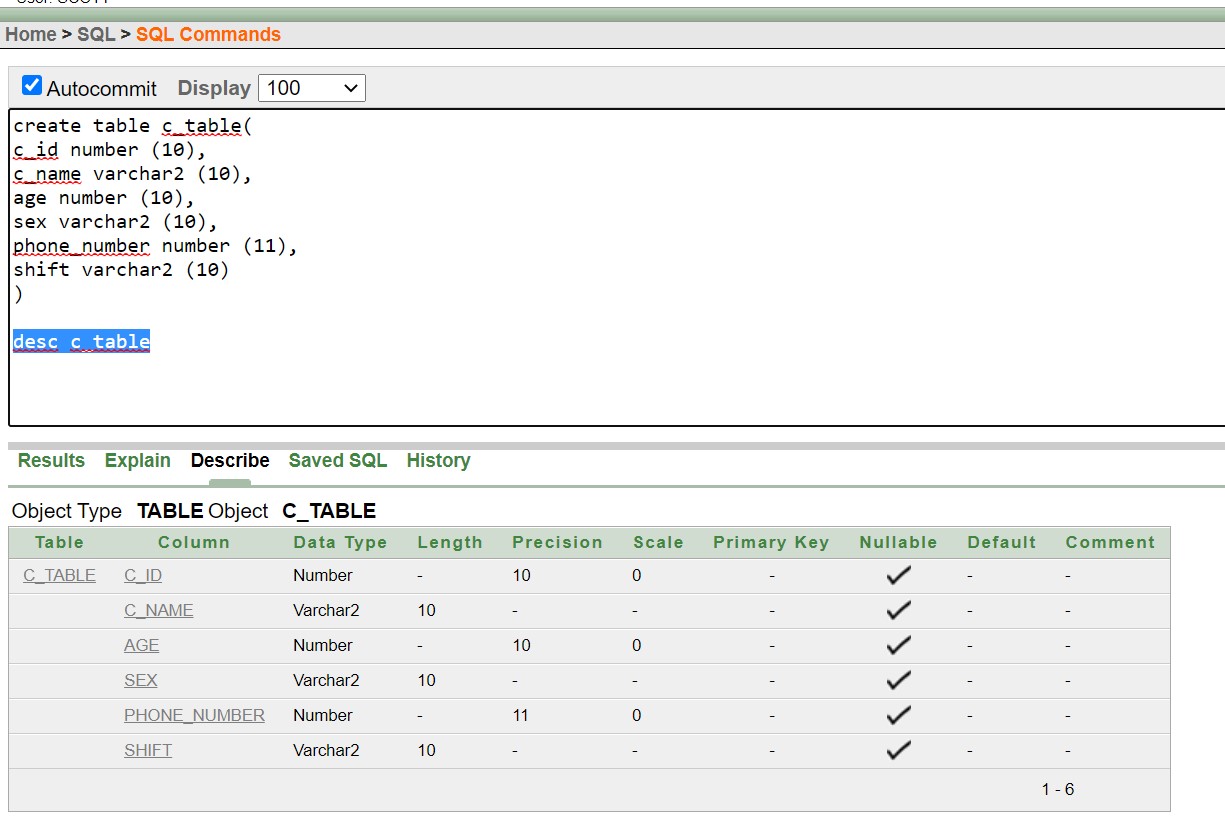
**7.**

create table c\_table( c\_id number (10) primary key, c\_name varchar2 (10), age number (10), sex varchar2 (10),

phone\_number number (11),

shift varchar2 (10)

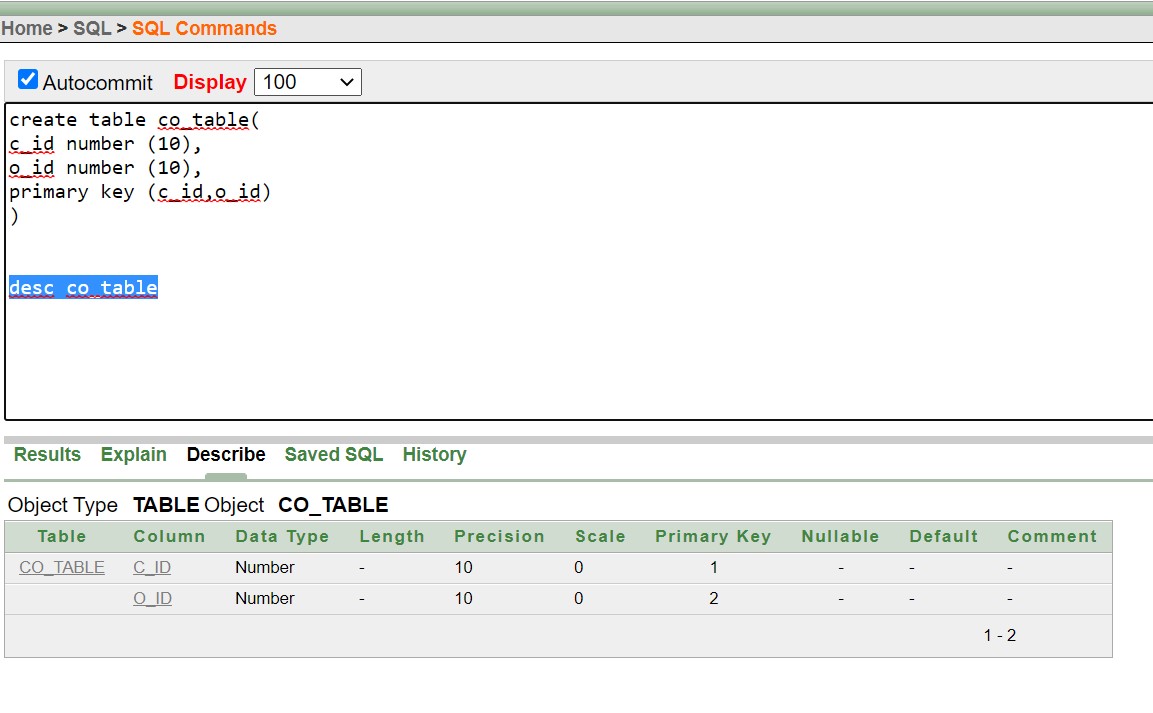
)



**8.**

create table co\_table( c\_id number (10), o\_id number (10), primary key (c\_id,o\_id)

)

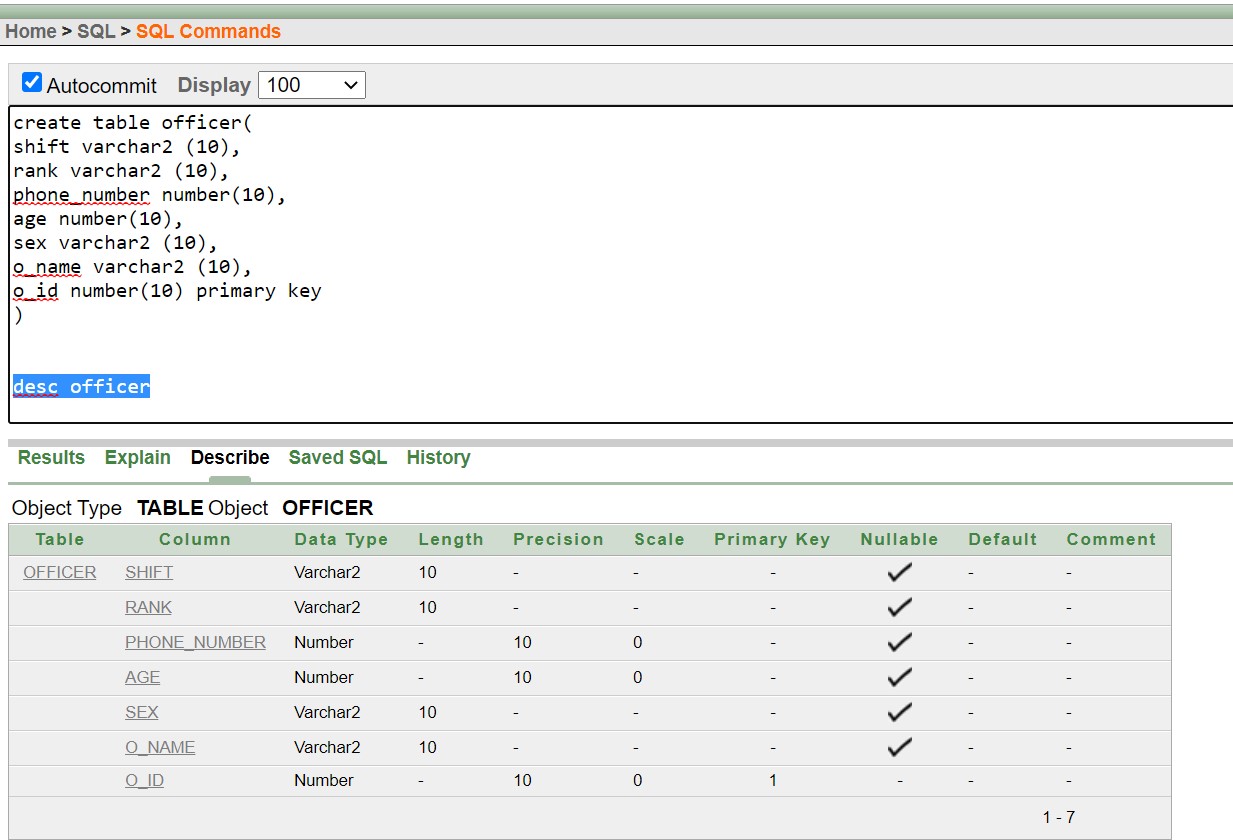


**9.**

create table officer( shift varchar2 (10), rank varchar2 (10),

phone\_number number(10), age number(10), sex varchar2 (10), o\_name varchar2 (10), o\_id number(10) primary key

)

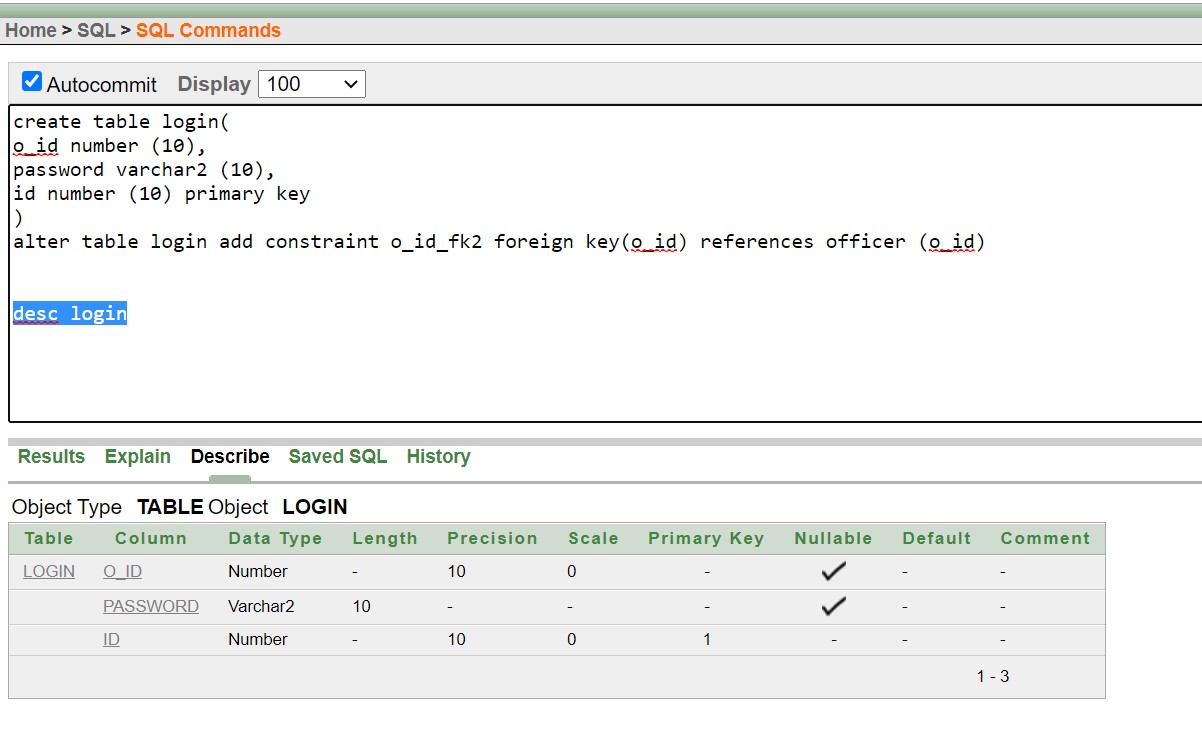


**10.**

create table login( o\_id number (10), password varchar2 (10), id number (10) primary key

)

alter table login add constraint o\_id\_fk2 foreign key(o\_id) references officer (o\_id)

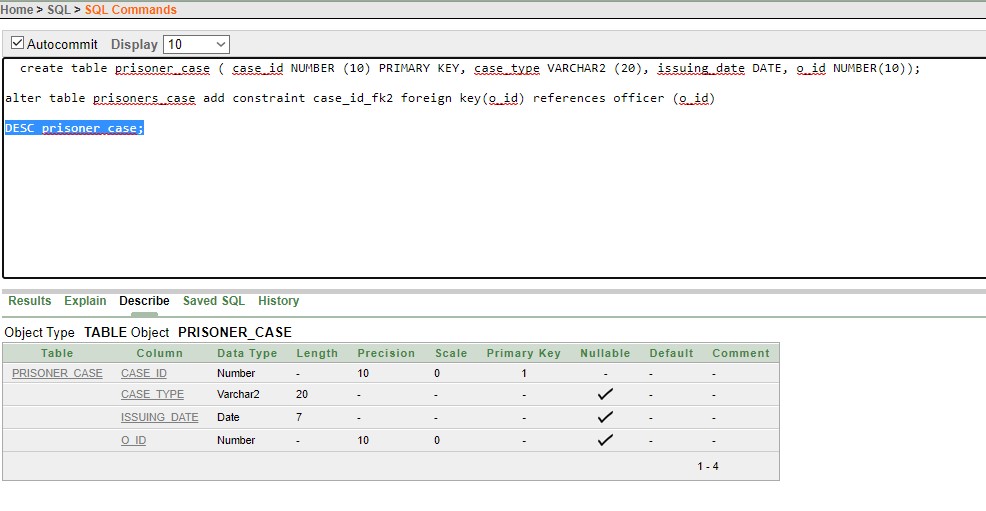


**11.**

create table prisoner\_case (

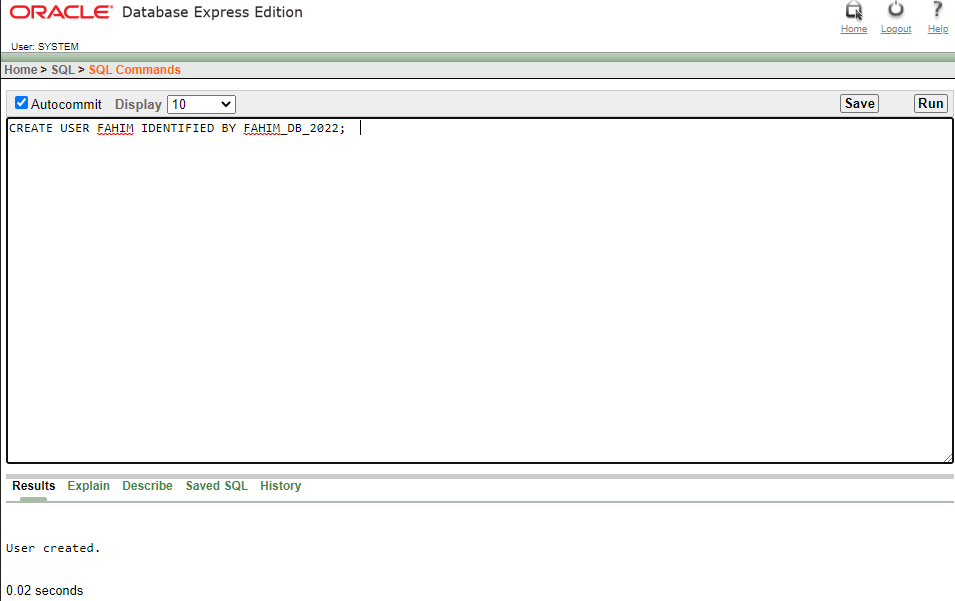
case\_id NUMBER (10) PRIMARY KEY, case\_type VARCHAR2 (20), issuing\_date DATE, o\_id NUMBER(10));

alter table prisoner\_case add constraint case\_id\_fk2 foreign key(o\_id) references officer (o\_id)

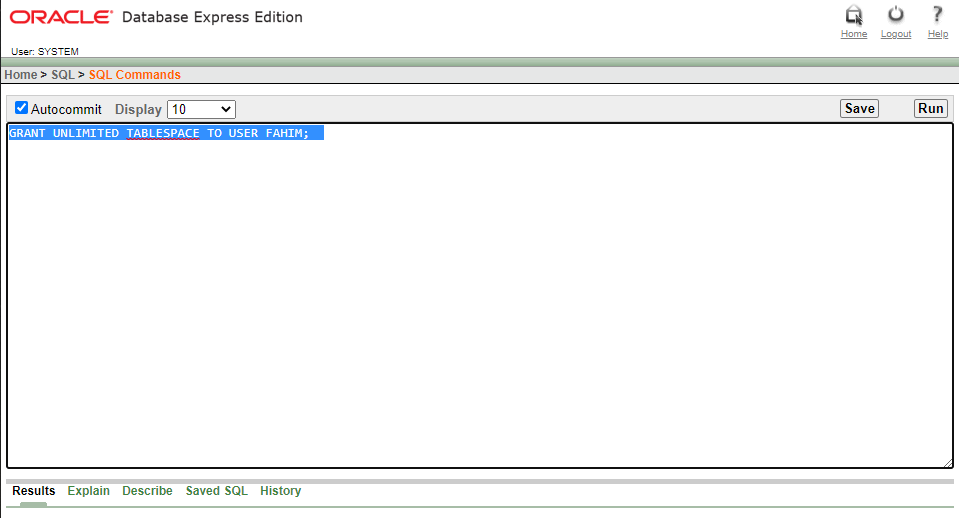


**User Creation and Role Assigning:**

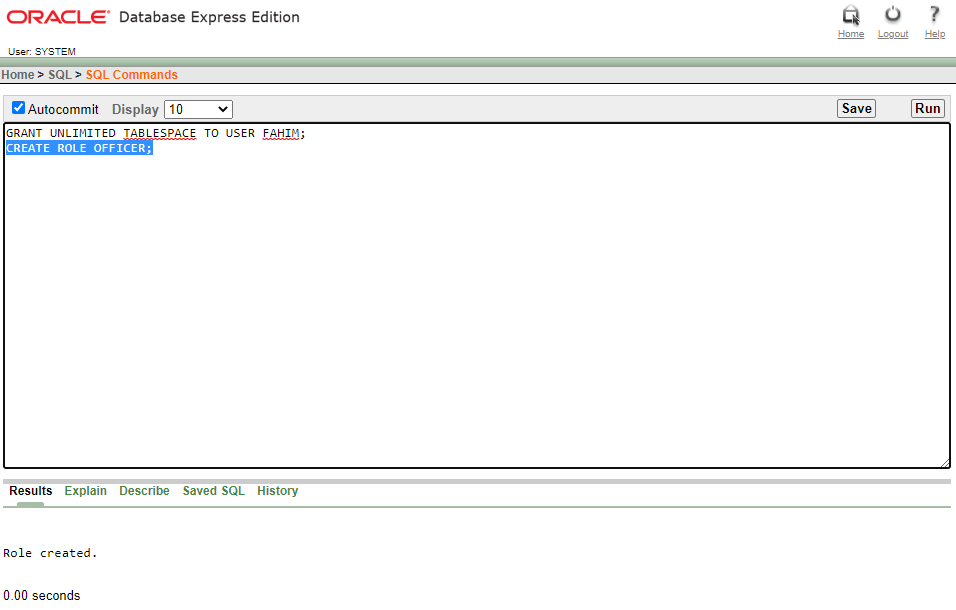
**Step 1:** CREATE USER FAHIM IDENTIFIED BY FAHIM\_DB\_2022;



**Step 2:** GRANT UNLIMITED TABLESPACE TO USER FAHIM;



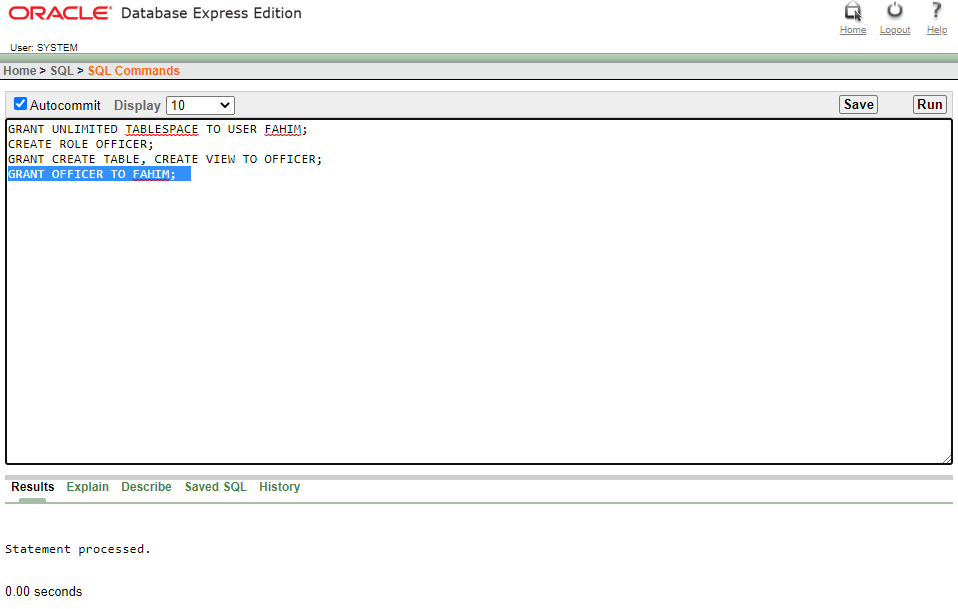
**Step 3:** CREATE ROLE OFFICER;



**Step 4:** GRANT CREATE TABLE, CREATE VIEW TO OFFICER;



**Step 5:** GRANT OFFICER TO FAHIM;



**Data Insertion**

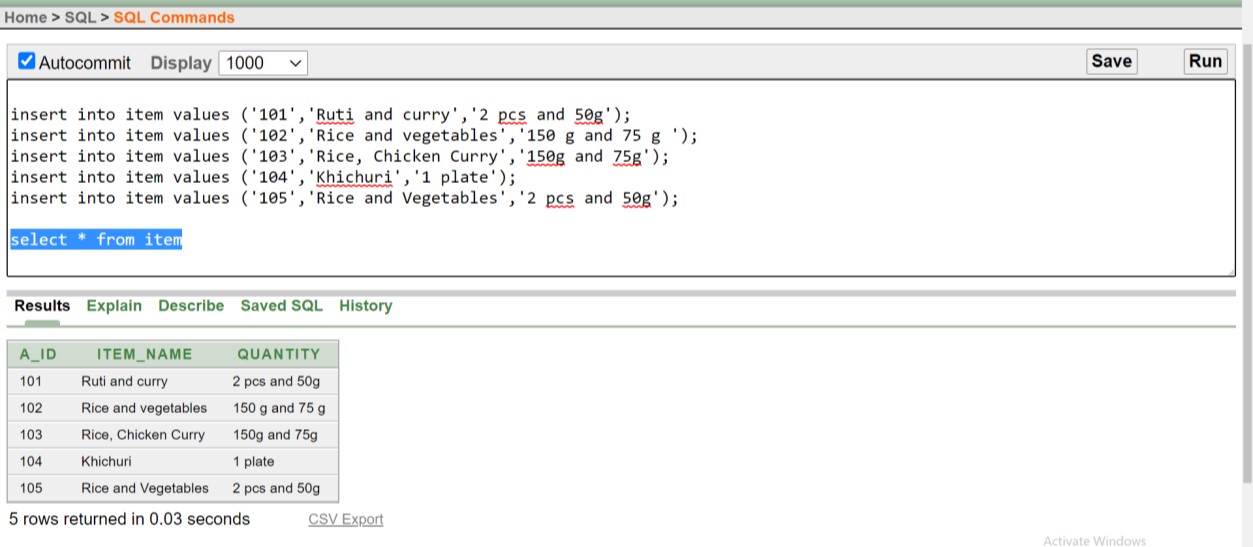
**1.**

Alter table item modify(item\_name varchar2 (20))

Alter table item modify(quantity varchar2 (20))

insert into item values ('101','Ruti and curry','2 pcs and 50g'); insert into item values ('102','Rice and vegetables','150 g and 75 g '); insert into item values ('103','Rice, Chicken Curry','150g and 75g'); insert into item values ('104','Khichuri','1 plate');

insert into item values ('105','Rice and Vegetables','2 pcs and 50g');



**2.**

insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id)

values(50 , 'Male',10195,'Rafiqul Islam','Khulna','b+','15 Feb 1970', '15 Jan

2012',0196548735,101);

insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id)

values(55 , 'Male',10112,'Shafik Islam','Rajshahi','a+','13 Feb 1965', '15 Jan

2019',0196557985,102);

insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id)

values(30 , 'Female',10185,'Habiba Islam','Munshiganj','ab+','19 May 1990', '11 Jan

2017',0174127985,103);

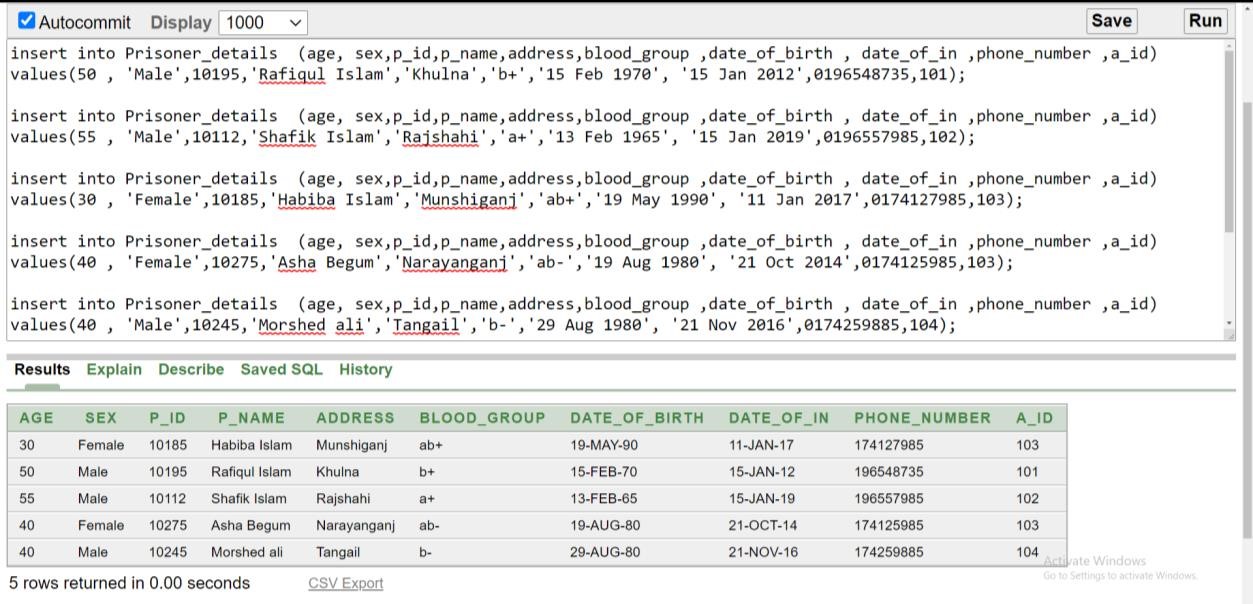
insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id)

values(40 , 'Female',10275,'Asha Begum','Narayanganj','ab-','19 Aug 1980', '21 Oct

2014',0174125985,103);

insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id)

values(40 , 'Male',10245,'Morshed ali','Tangail','b-','29 Aug 1980', '21 Nov 2016',0174259885,104);



**3.**

Alter table Visitors\_details modify(v\_name varchar2 (20))

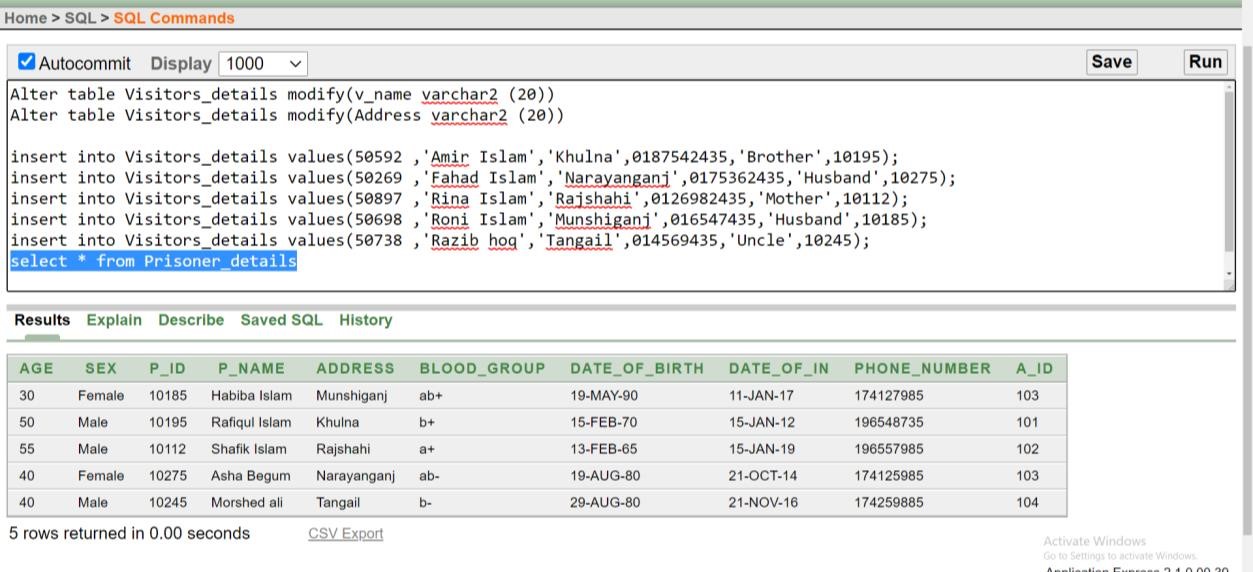
Alter table Visitors\_details modify(Address varchar2 (20))

insert into Visitors\_details values(50592,'Amir Islam', 'Khulna',0187542435,' Brother',10195); insert into Visitors\_details values(50269,'Fahad Islam','Narayanganj',0175362435,' Husband',10275);

insert into Visitors\_details values(50897 ,' Rina Islam','Rajshahi',0126982435,' Mother',10112);

insert into Visitors\_details values(50698 ,' Roni Islam', 'Munshiganj',016547435,' Husband',10185);

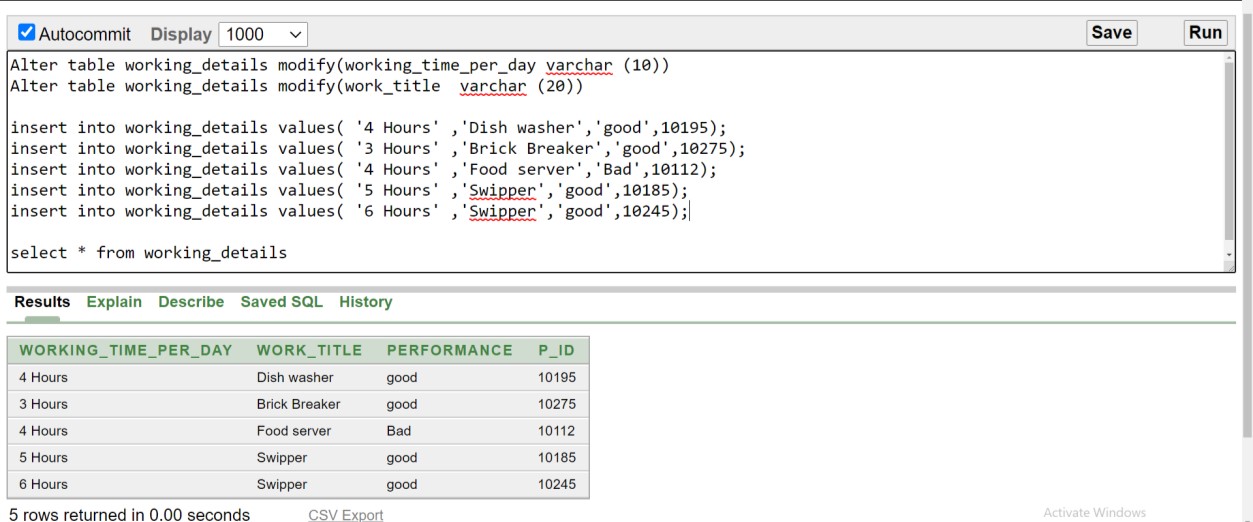
insert into Visitors\_details values(50738 ,'Razib hoq','Tangail',014569435,'Uncle',10245);



**4.**

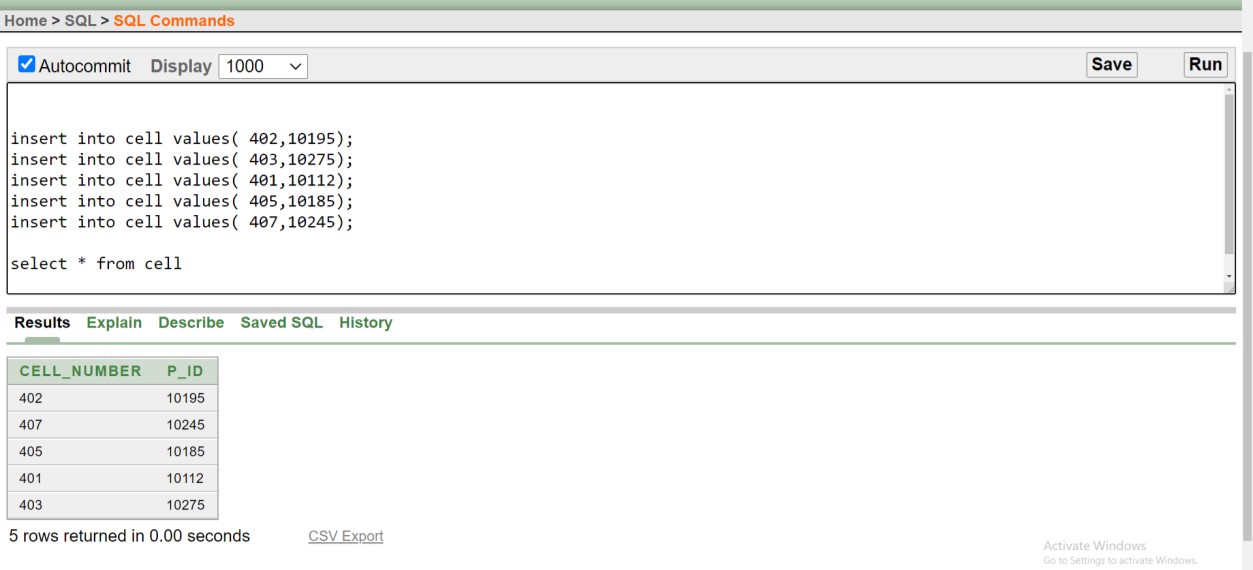
Alter table working\_details modify(working\_time\_per\_day varchar (10)) Alter table working\_details modify(work\_title varchar (20))

insert into working\_details values( '4 Hours' ,'Dish washer','good',10195); insert into working\_details values( '3 Hours' ,'Brick Breaker','good',10275); insert into working\_details values( '4 Hours' ,'Food server','Bad',10112); insert into working\_details values( '5 Hours' ,'Swipper','good',10185); insert into working\_details values( '6 Hours' ,'Swipper','good',10245);



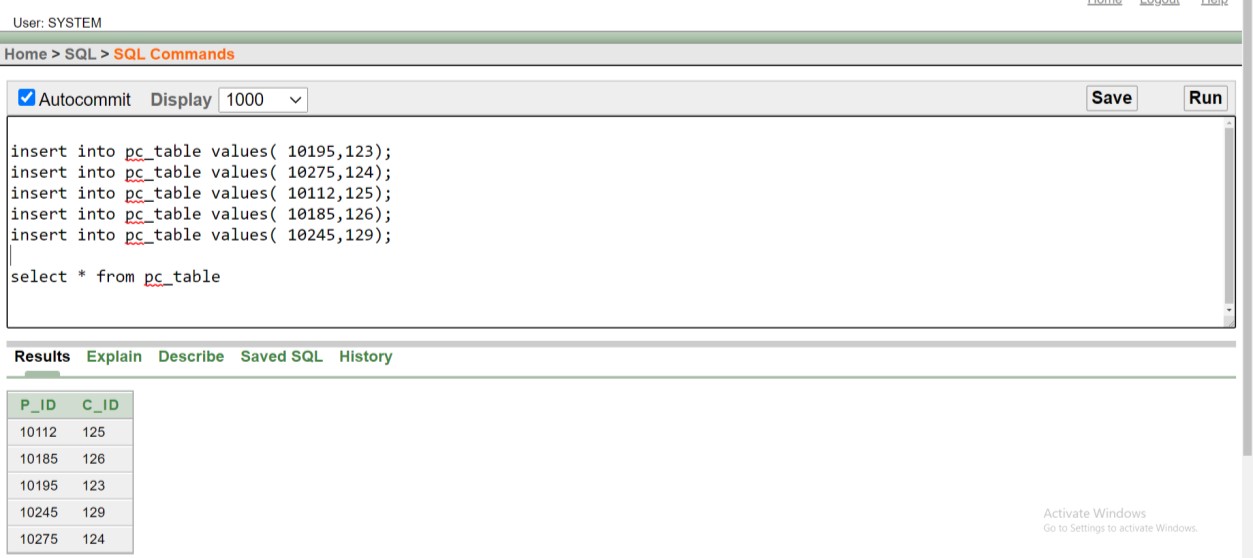
**5.**

insert into cell values( 402,10195); insert into cell values( 403,10275); insert into cell values( 401,10112); insert into cell values( 405,10185); insert into cell values( 407,10245);



**6.**

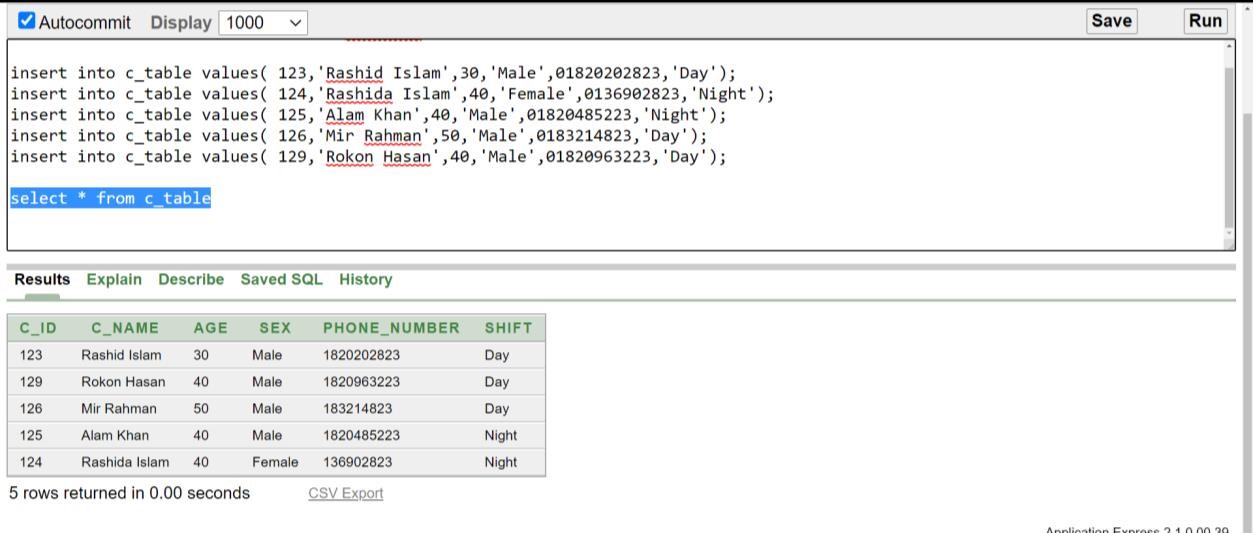
insert into pc\_table values( 10195,123); insert into pc\_table values( 10275,124); insert into pc\_table values( 10112,125); insert into pc\_table values( 10185,126); insert into pc\_table values( 10245,129);



**7.**

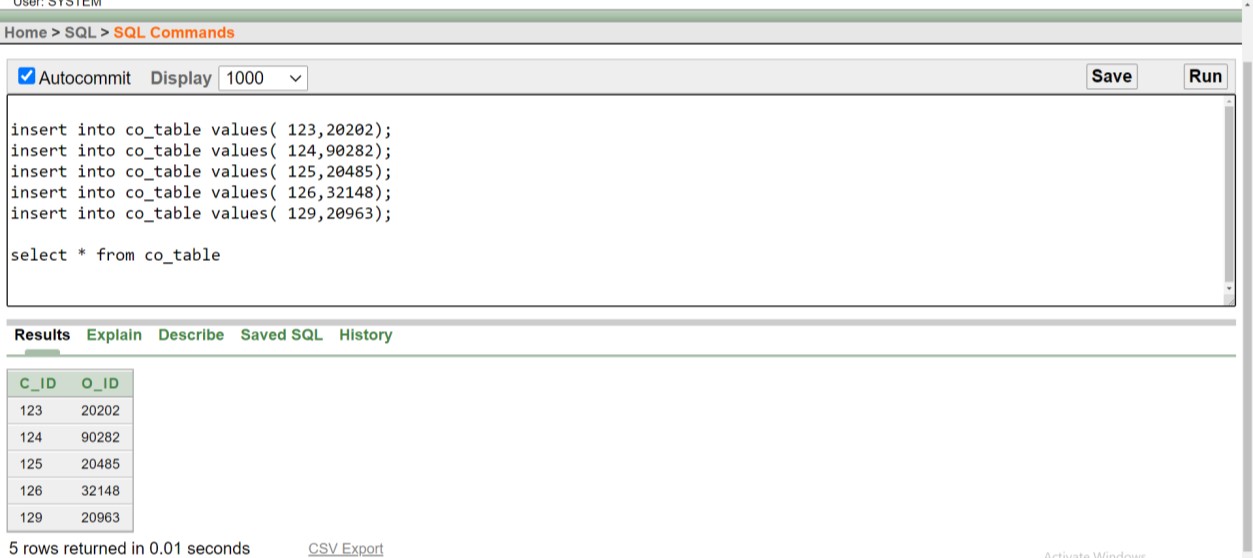
Alter table c\_table modify( c\_name varchar2 (20))

insert into c\_table values( 123,'Rashid Islam',30,'Male',01820202823,'Day'); insert into c\_table values( 124,'Rashida Islam',40,'Female',0136902823,'Night'); insert into c\_table values( 125,'Alam Khan',40,'Male',01820485223,'Night'); insert into c\_table values( 126,'Mir Rahman',50,'Male',0183214823,'Day'); insert into c\_table values( 129,'Rokon Hasan',40,'Male',01820963223,'Day');



**8.**

insert into co\_table values( 123,20202); insert into co\_table values( 124,90282); insert into co\_table values( 125,20485); insert into co\_table values( 126,32148); insert into co\_table values( 129,20963);



**9.**

Alter table officer modify(rank varchar2 (25))

Alter table officer modify(o\_name varchar2 (20))

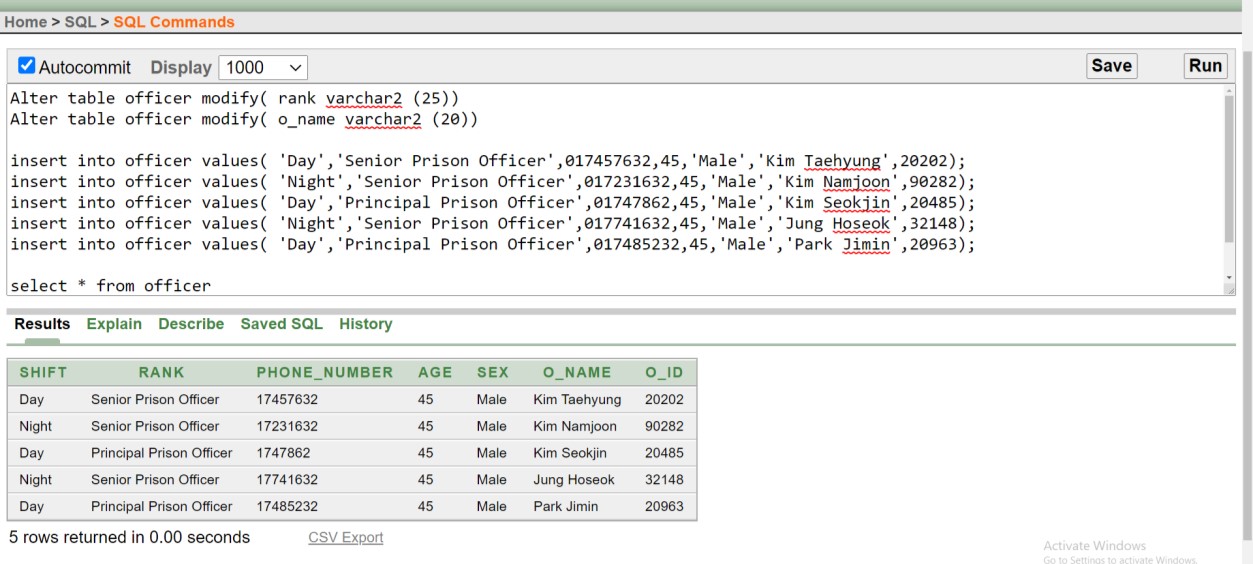
insert into officer values( 'Day','Senior Prison Officer',017457632,45,'Male','Kim Taehyung',20202);

insert into officer values( 'Night','Senior Prison Officer',017231632,45,'Male','Kim Namjoon',90282);

insert into officer values( 'Day','Principal Prison Officer',01747862,45,'Male','Kim Seokjin',20485);

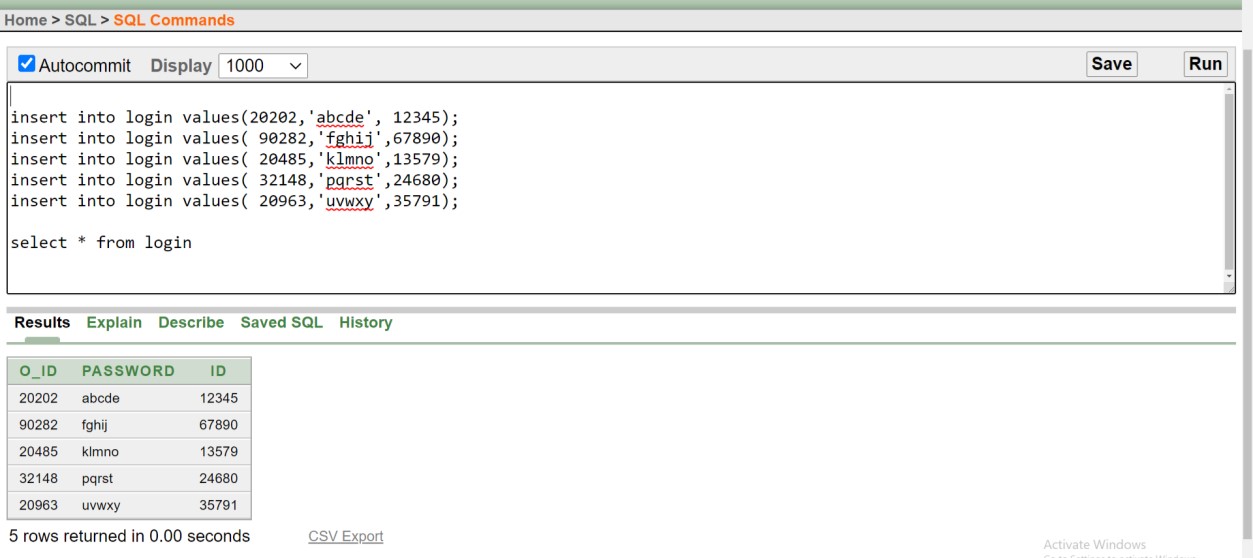
insert into officer values( 'Night','Senior Prison Officer',017741632,45,'Male','Jung Hoseok',32148);

insert into officer values( 'Day','Principal Prison Officer',017485232,45,'Male','Park Jimin',20963);

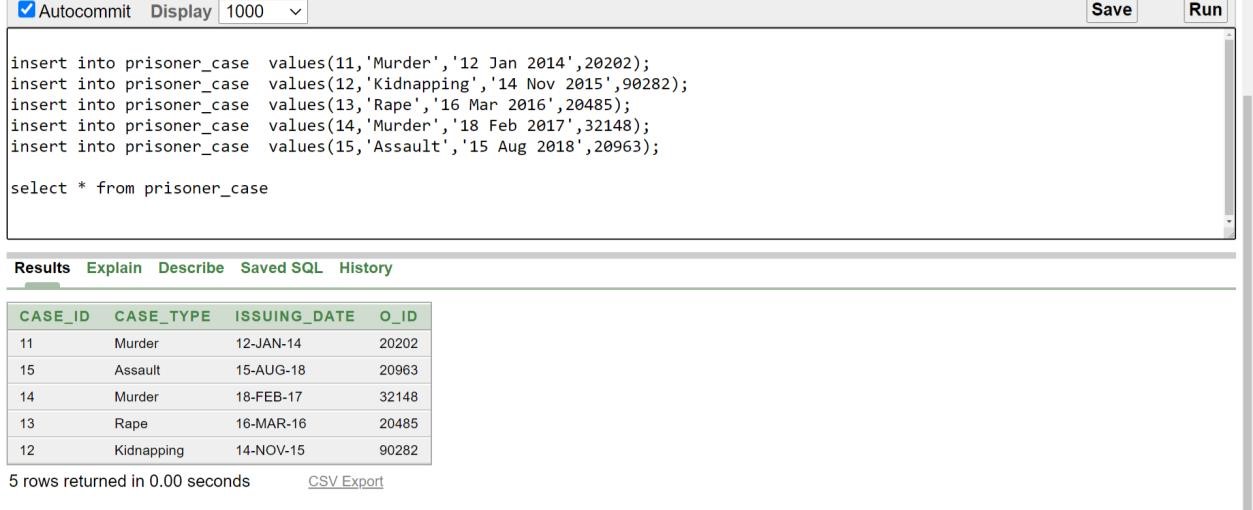


**10.**

insert into login values(20202,'abcde', 12345); insert into login values( 90282,'fghij',67890); insert into login values( 20485,'klmno',13579); insert into login values( 32148,'pqrst',24680); insert into login values( 20963,'uvwxy',35791);



**11.** insert into prisoner\_case values(11,'Murder','12 Jan 2014',20202); insert into prisoner\_case values(12,'Kidnapping','14 Nov 2015',90282); insert into prisoner\_case values(13,'Rape','16 Mar 2016',20485); insert into prisoner\_case values(14,'Murder','18 Feb 2017',32148); insert into prisoner\_case values(15,'Assault','15 Aug 2018',20963);

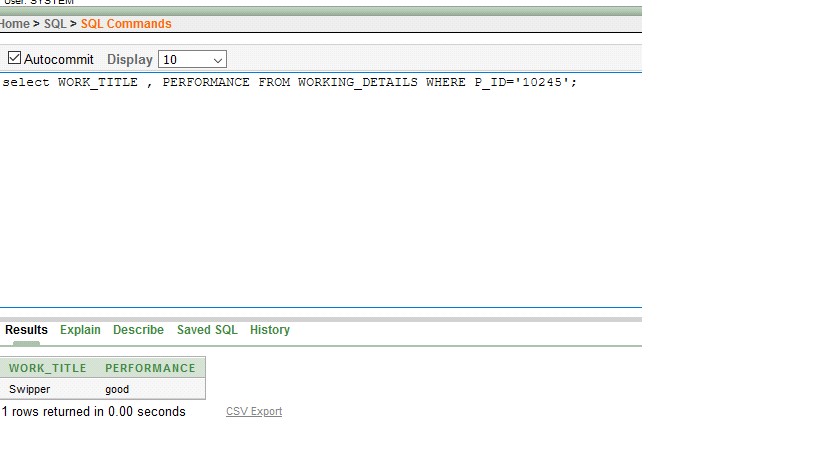


**QUERY WRITING:**

**Single row:**

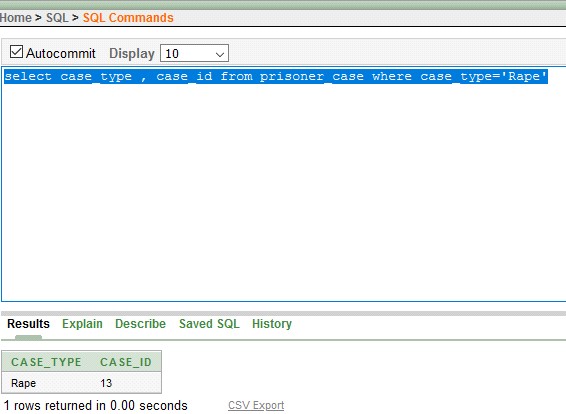
**1**. display the work title and performance for p\_id '10245'

- **select WORK\_TITLE, PERFORMANCE FROM WORKING\_DETAILS WHERE P\_ID='10245';**



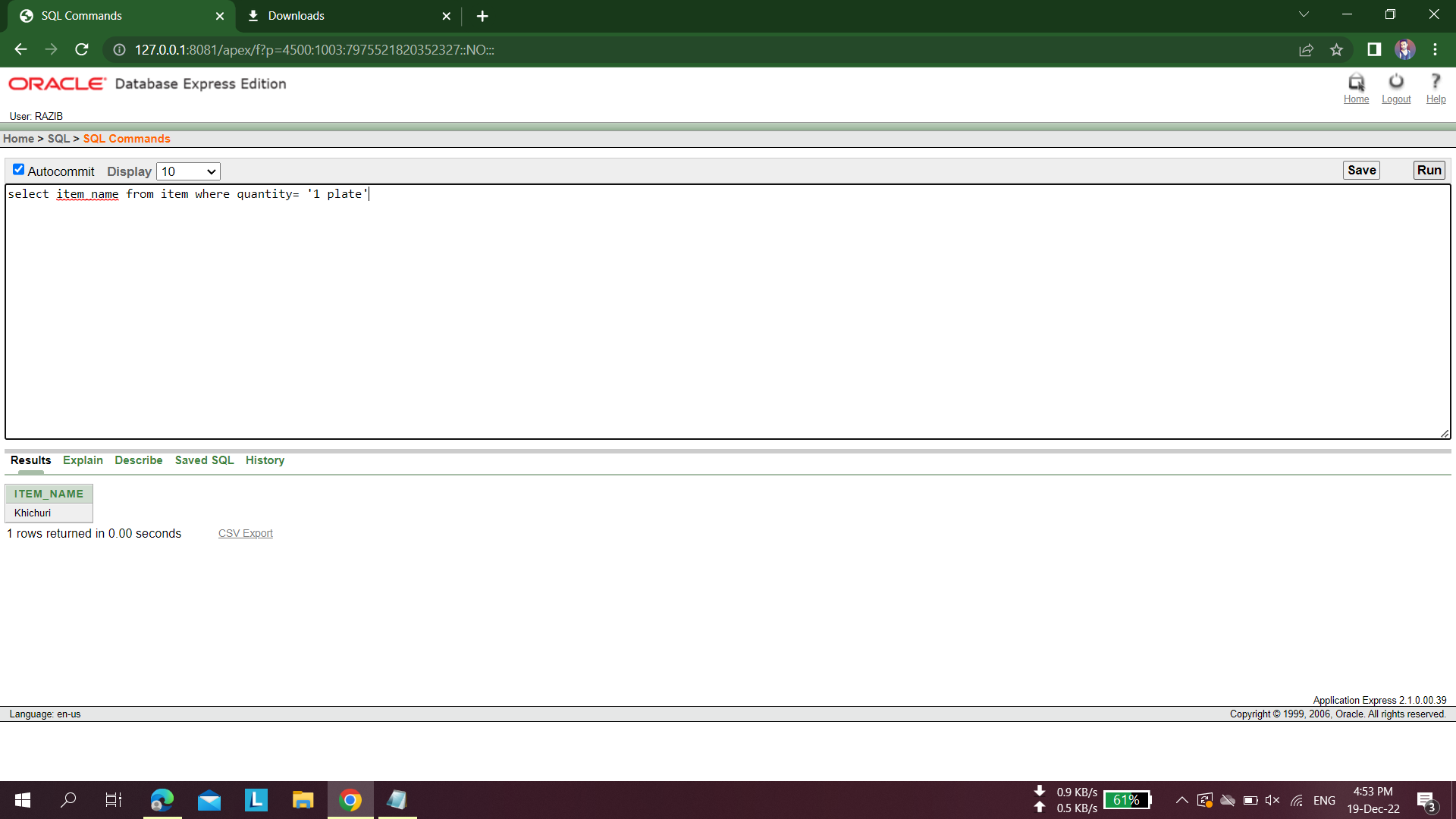
**2.** display case id from prisoner case details where case type is RAPE

- **select case\_type , case\_id from prisoner\_case where case\_type='Rape';**



1. Display Item name where quantity is 1 plate

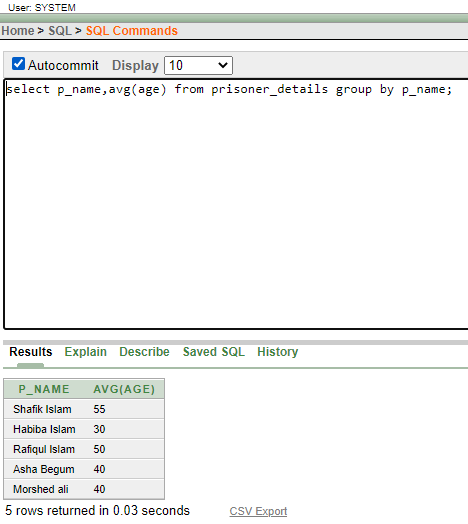
* **select item\_name from item where quantity = ‘1 plate’;**



**Group Function:**

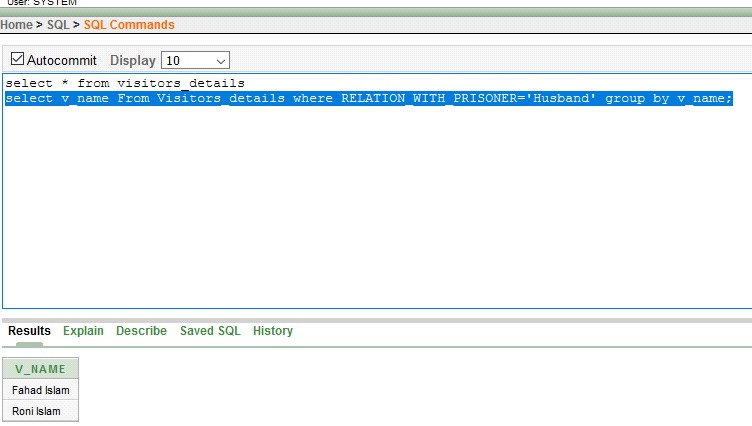
1. Display prisoner’s name and average age and group them by names

* **select p\_name,avg(age) from prisoner\_details group by p\_name;**



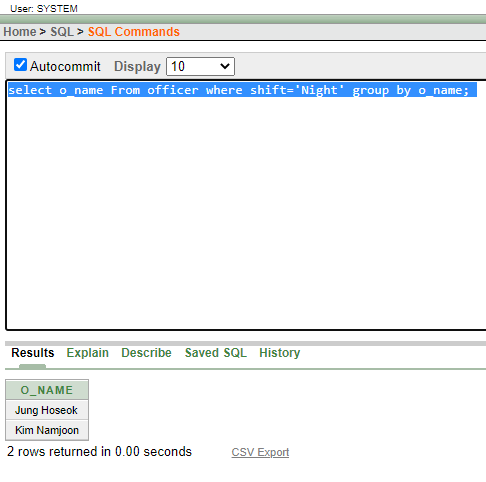
1. Display visitor’s name by a group of those who have a relation of husband with prisoners

* **select v\_name From Visitors\_details where RELATION\_WITH\_PRISONER='Husband' group by v\_name;**



1. Display officer names and group them by officer name those who have shift at night

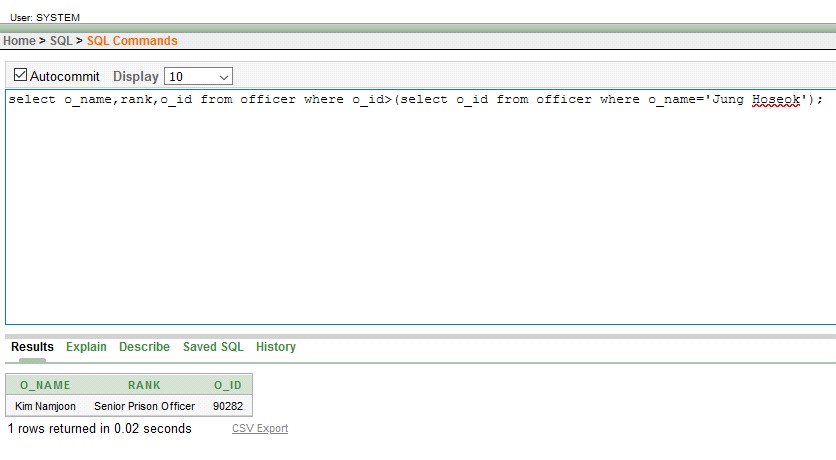
* **select o\_name from officer where shift='night' group by o\_name;**



**Subquery:**

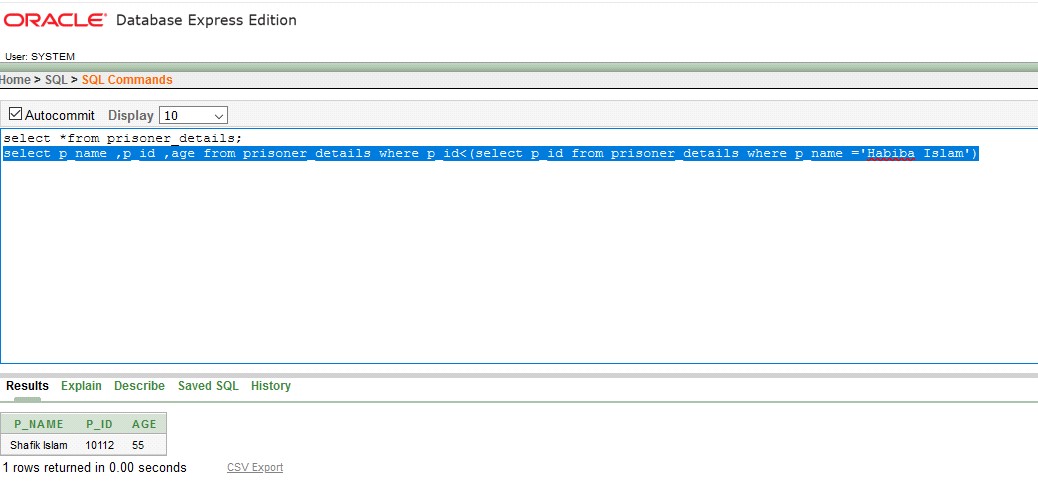
1. display officer name , rank and ID and whose officer id is bigger than Jung Hoseok's Id from officer table

* **select o\_name,rank,o\_id from officer where o\_id>(select o\_id from officer where o\_name='Jung Hoseok');**



1. display prisoner name, id and age and whose prisoner id are smaller than Habiba Islam's id from prisoner \_details table

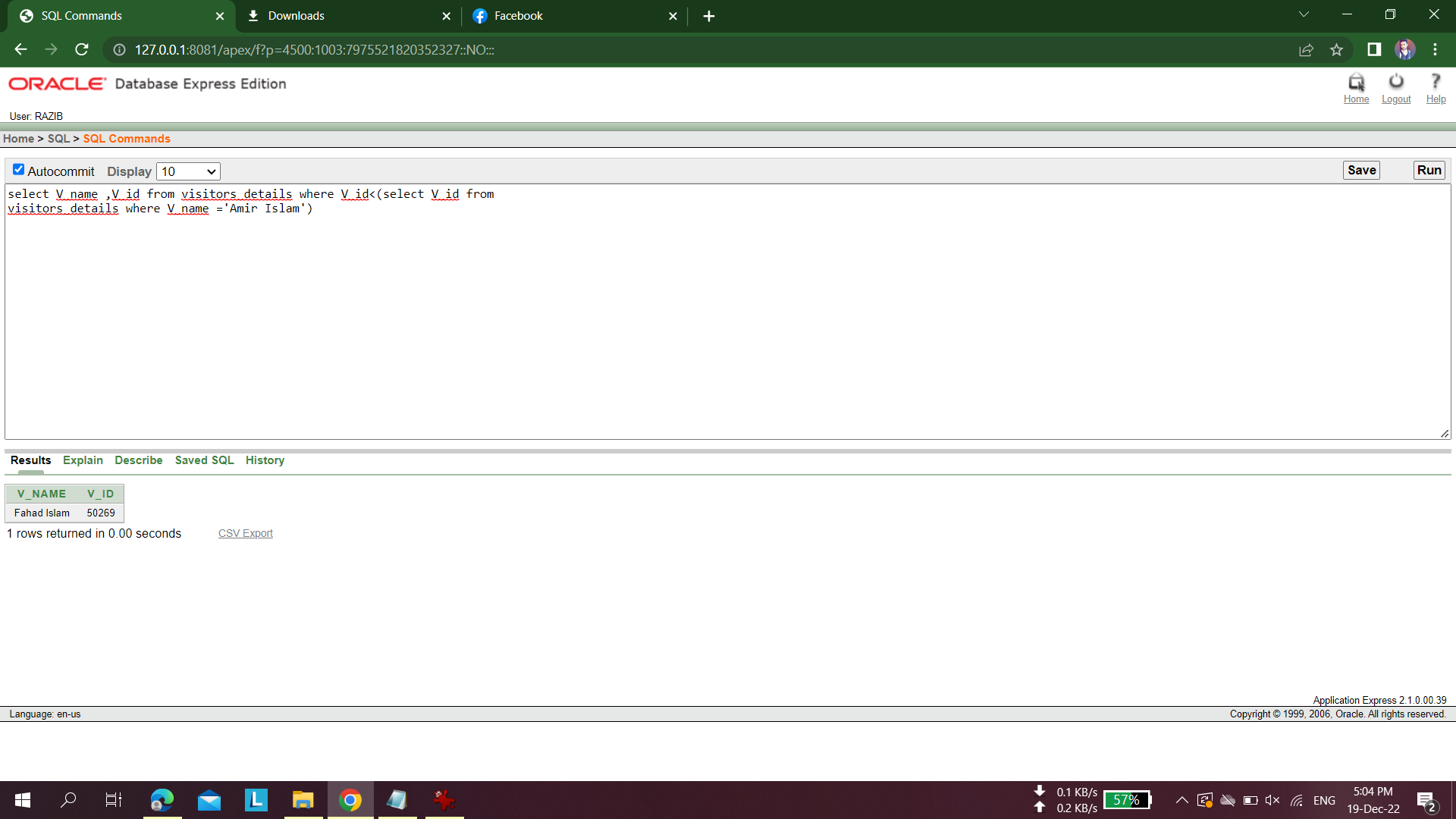
* **select p\_name ,p\_id ,age from prisoner\_details where p\_id<(select p\_id from prisoner\_details where p\_name ='Habiba Islam')**



1. display visitors name, id whose visitor id is greater than Amir Islam's id from visitors \_details table

* **select V\_name ,V\_id from visitors\_details where V\_id<(select V\_id from**

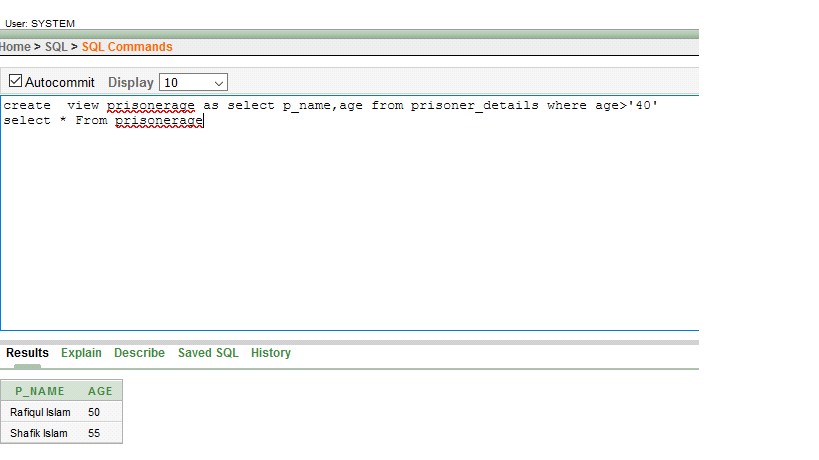
**visitors\_details where V\_name ='Amir Islam');**



**View:**

1. create a view prisonerage that contains the details of prisoners with age greater than 40

* **create view prisonerage as select p\_name, age from prisoner\_Details where age>40**



1. create a view WORKLOAD , that contains the id of prisoners who works less than 5 hours

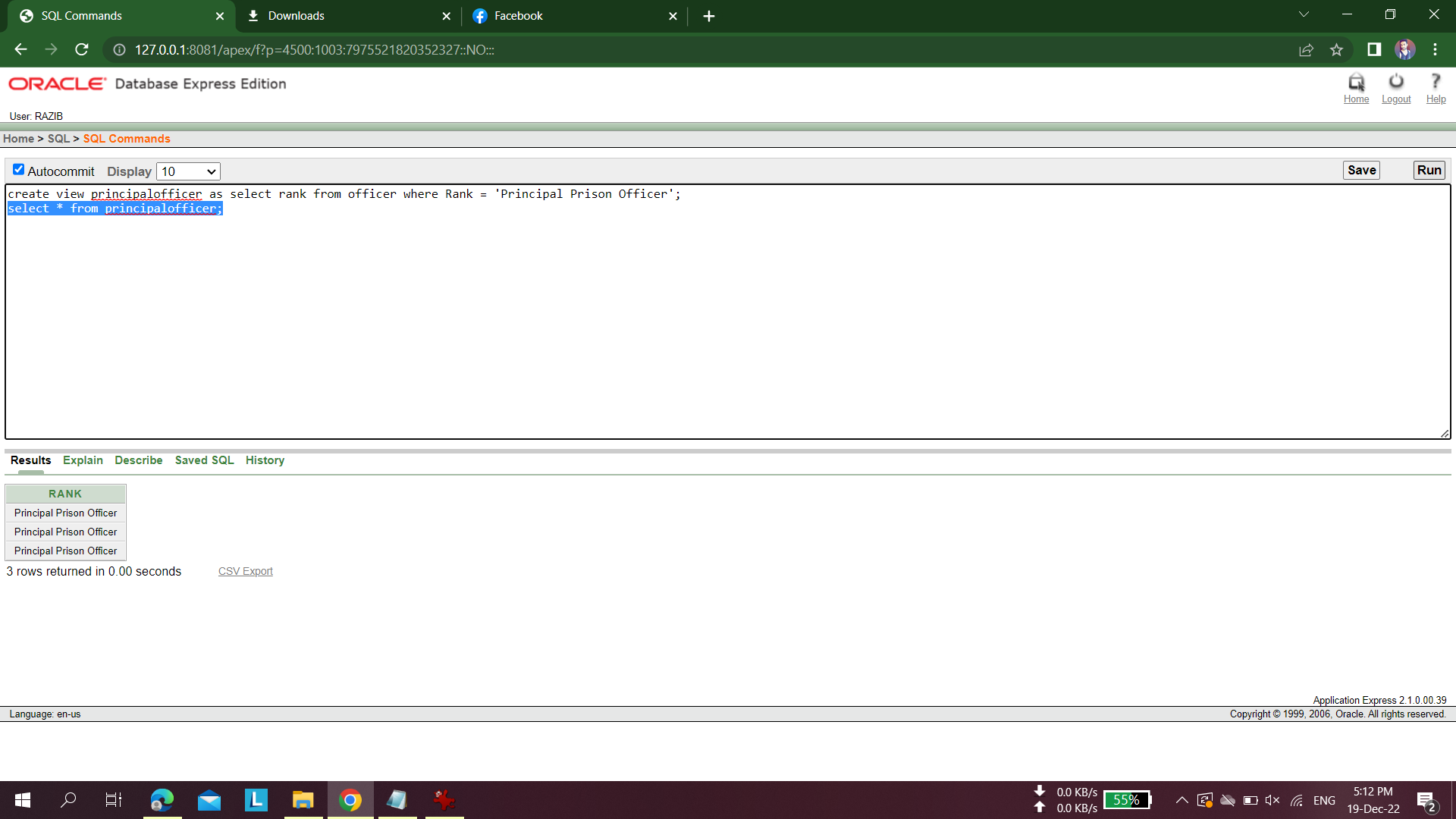
* **create view WORKLOAD as select p\_id, WORKING\_TIME\_PER\_DAY FROM**

**WORKING\_DETAILS WHERE WORKING\_TIME\_PER\_DAY <'5 Hours';**



1. Create a view PrincipleOfficer that contains the details of Officers with Rank = PrinciplePrisonOffice

* **Create view principalofficer as select rank from officer where Rank = 'Principal Prison Officer';**



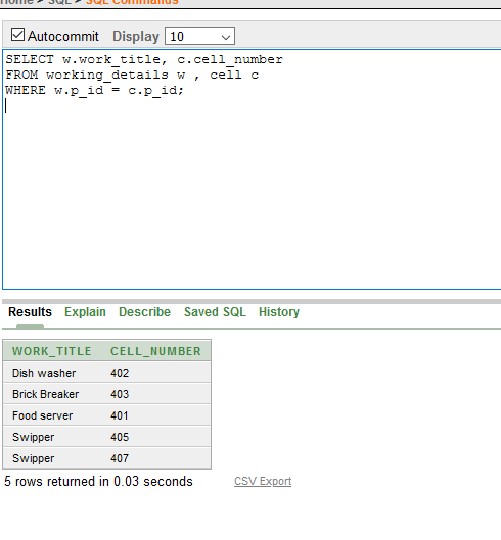
**Join:**

1. Joining the work title from working\_details table and cell number from cell table using EQUIJOIN as working\_details and cell table has direct relation.

**SELECT w.work\_title, c.cell\_number**

**FROM working\_details w , cell c**

**WHERE w.p\_id = c.p\_id;**

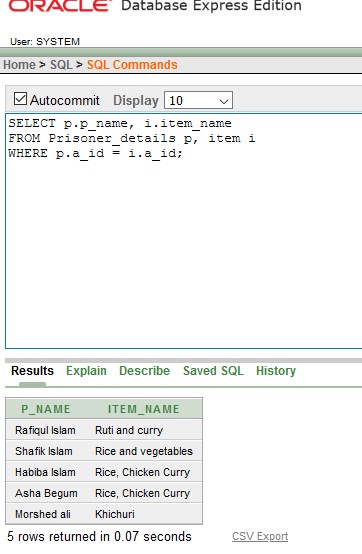


1. Joining the prisoner name from Prisoner\_details table and item name of allocated food from item table using EQUIJOIN as Prisoner\_details and item table has direct relation.

**SELECT p.p\_name, i.item\_name**

**FROM Prisoner\_details p, item i**

**WHERE p.a\_id = i.a\_id;**

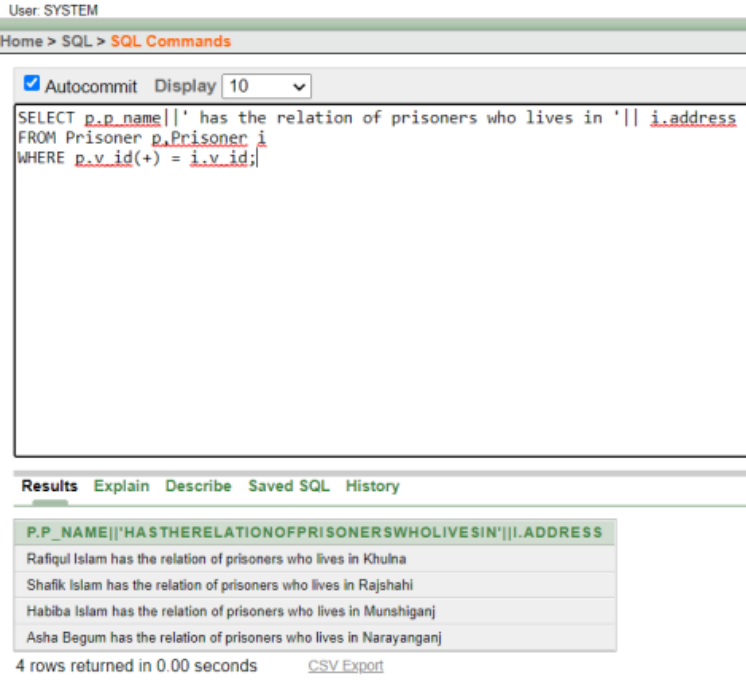


1. Joining the prisoners’s name from visitors table and relation of prisoners where he/she lived from prisoner’s details table using SELFJOIN has Visitor and prisoner has direct relation.

**SELECT p.p\_name||' has the relation of prisoners who lives in '|| i.address**

**FROM Prisoner p,Prisoner i**

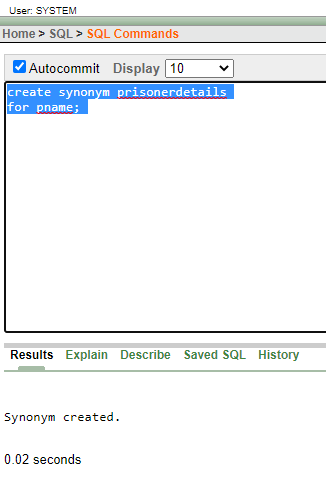
**WHERE p.v\_id(+) = i.v\_id;**



**Synonym:**

1. Create a synonym that contains the details of prisoner’s name for the view of prisoner details.

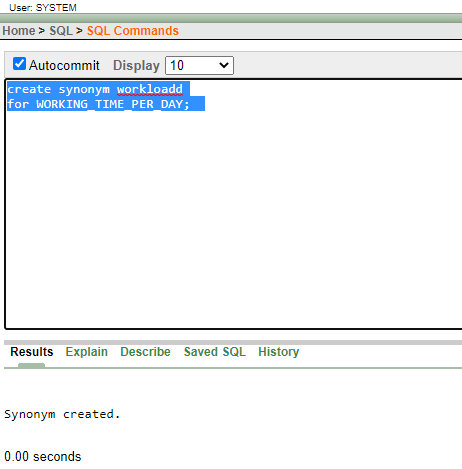
* **create synonym prisonerdetails for p\_name;**



1. Create a synonym that contains the details of Working Time Per Day for the view of workload.

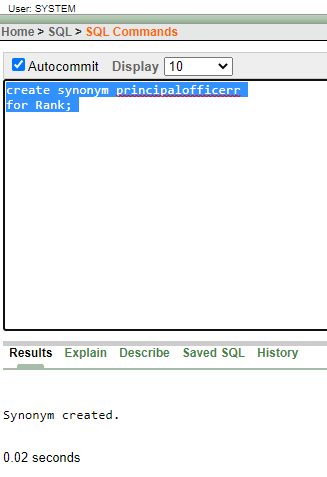
* **create synonym workloadd**

**for WORKING\_TIME\_PER\_DAY;**



1. Create a synonym that contains the details of Officers with Rank for the view of PrincipleOfficer.

* **create synonym principalofficerr for Rank;**



# **PL/SQL:**

# **Function:**

1. Create a function that returns the total number of officers in the officer table.

**CREATE FUNCTION totalOfficers**

**RETURN number AS**

**total number(2) := 0;**

**BEGIN**

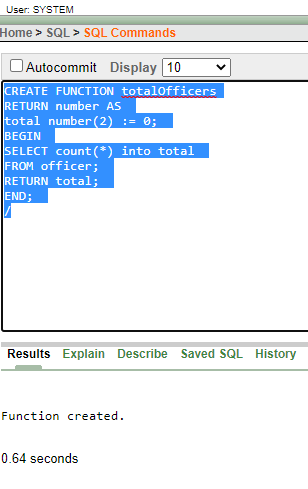
**SELECT count(\*) into total**

**FROM officer;**

**RETURN total;**

**END;**

**/.**



**DECLARE**

**c number(2);**

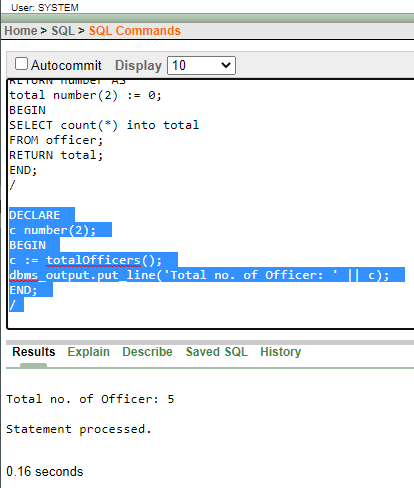
**BEGIN**

**c := totalOfficers();**

**dbms\_output.put\_line('Total no. of Officer: ' || c);**

**END;**

**/**



1. Create a function that returns the number of prisoners who works less than 4 hours in the working details table.

**CREATE FUNCTION worker**

**RETURN number AS**

**total number(2) := 0;**

**BEGIN**

**select count(\*) into total FROM**

**WORKING\_DETAILS WHERE WORKING\_TIME\_PER\_DAY <'4 Hours';**

**RETURN total;**

**END;**

**/**

**Calling a Function:**

**DECLARE**

**c number(2);**

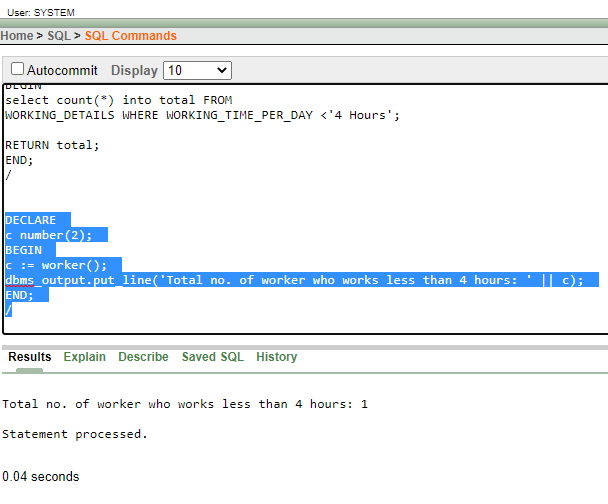
**BEGIN**

**c := worker();**

**dbms\_output.put\_line('Total no. of worker who works less than 4 hours: ' || c);**

**END;**

**/**



1. Create a function that returns the number of prisoners where the case type is Murder in the working details table.

**CREATE FUNCTION NumOfMurdererr**

**RETURN number AS**

**total number(2) := 0;**

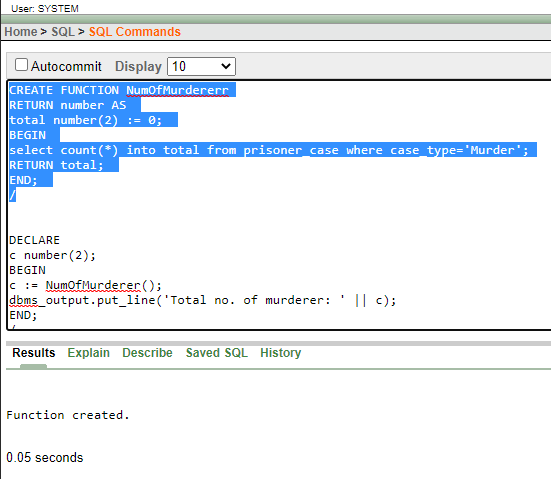
**BEGIN**

**select count(\*) into total from prisoner\_case where case\_type='Murder';**

**RETURN total;**

**END;**

**/**



**Calling a Function:**

**DECLARE**

**c number(2);**

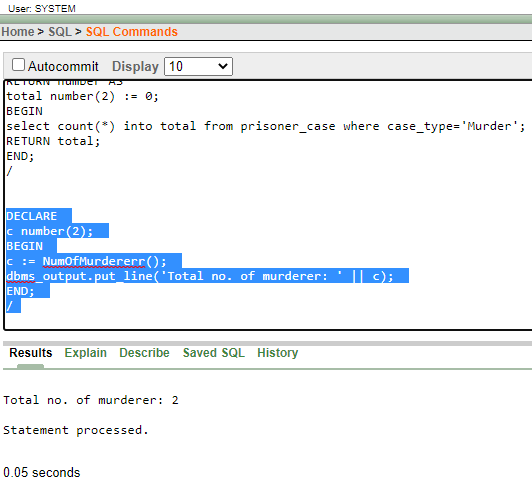
**BEGIN**

**c := NumOfMurdererr();**

**dbms\_output.put\_line('Total no. of murderer: ' || c);**

**END;**

**/**



# **Procedure:**

1. create a procedure to update the value of the column of performance from good to Bad in the working table.

**CREATE PROCEDURE adjust\_performance(**

**in\_performance IN working\_details.performance%TYPE)**

**IS**

**BEGIN**

**UPDATE working\_details**

**SET performance ='Bad'**

**WHERE performance= in\_performance;**

**END;**

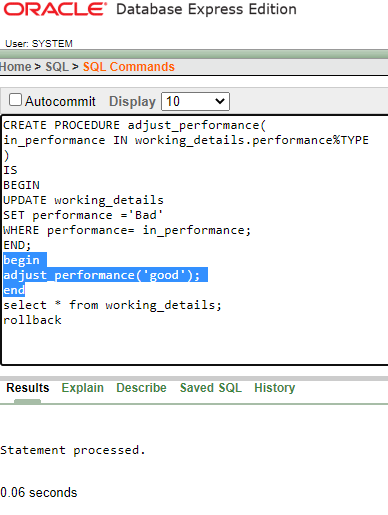
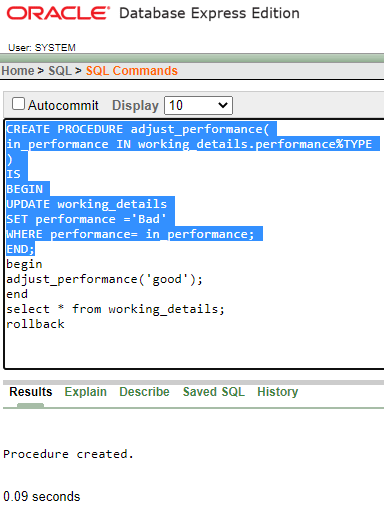
**begin**

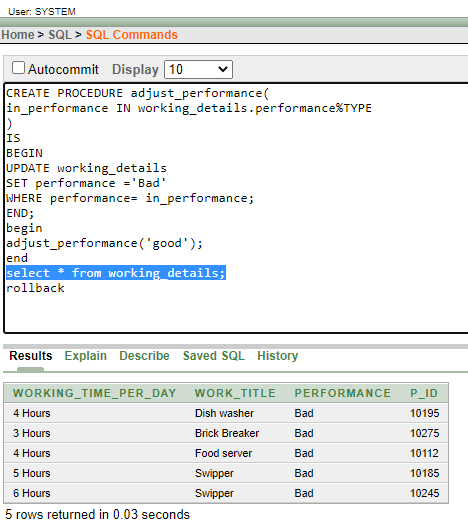
**adjust\_performance('good');**

**end**

**select \* from working\_details;**

**rollback**





1. create a procedure to update the value of the column of working time per day from 4 Hours to 2 Hours in the working table.

**CREATE PROCEDURE adjust\_working\_time\_per\_day(**

**in\_working\_time\_per\_day IN working\_details.** **working\_time\_per\_day%TYPE)**

**IS**

**BEGIN**

**UPDATE working\_details**

**SET performance ='2 Hours'**

**WHERE working\_time\_per\_day= in\_working\_time\_per\_day;**

**END;**

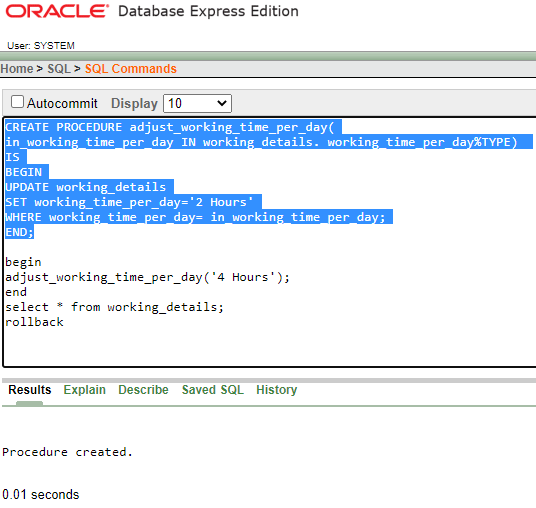
**begin**

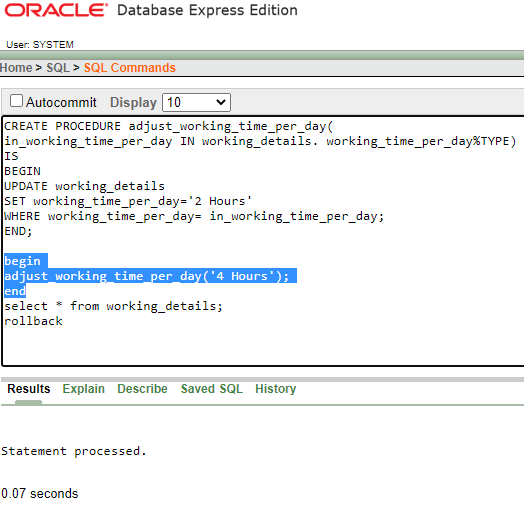
**adjust\_working\_time\_per\_day('4 Hours’);**

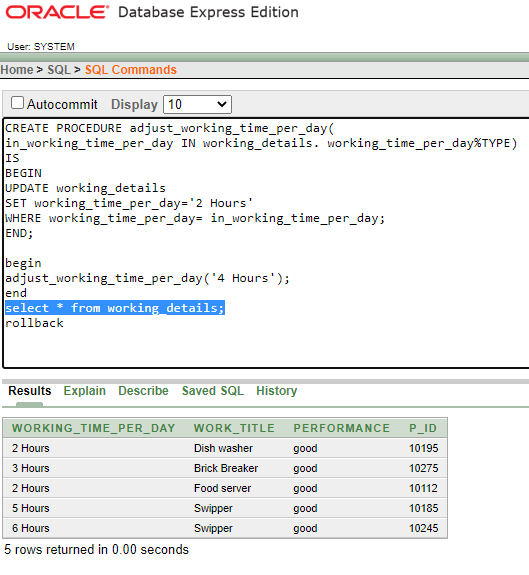
**end**

**select \* from working\_details;**

**rollback**







1. create a procedure to update the value of the column of shift from Day to Night in officer table.

**CREATE PROCEDURE adjust\_shift(**

**in\_shift IN officer.** **shift%TYPE)**

**IS**

**BEGIN**

**UPDATE officer**

**SET shift ='Night'**

**WHERE shift = in\_shift;**

**END;**

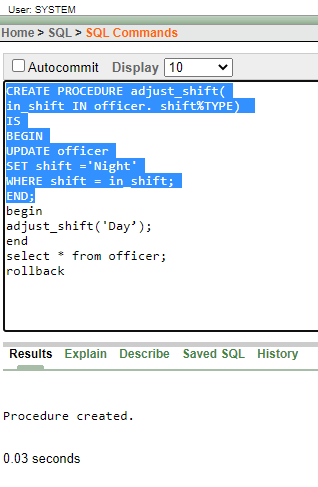
**begin**

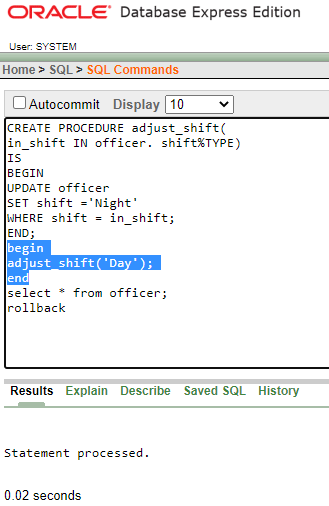
**adjust\_shift('Day’);**

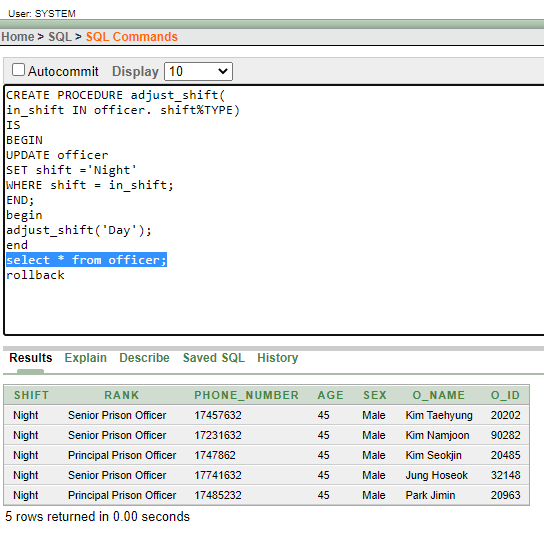
**end**

**select \* from officer;**

**rollback**







# **Record:**

1. Create a record that can output the name, rank and shift of the officer whose id is 20202 in the officer table.

**declare**

**dept\_officer officer%rowtype;**

**begin**

**select \* into dept\_officer from officer**

**where o\_id=20202;**

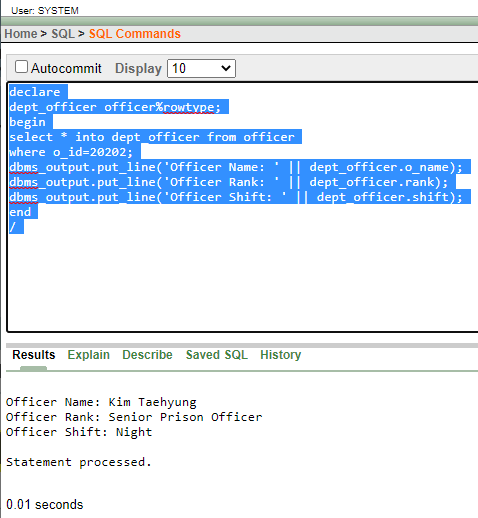
**dbms\_output.put\_line('Officer Name: ' || dept\_officer.o\_name);**

**dbms\_output.put\_line('Officer Rank: ' || dept\_officer.rank);**

**dbms\_output.put\_line('Officer Shift: ' || dept\_officer.shift);**

**end**

**/**



1. Create a record that can output the name of prisoners.

**declare**

**dept\_prisoner Prisoner\_details%rowtype;**

**begin**

**for dept\_prisoner**

**in(select \* from Prisoner\_details)**

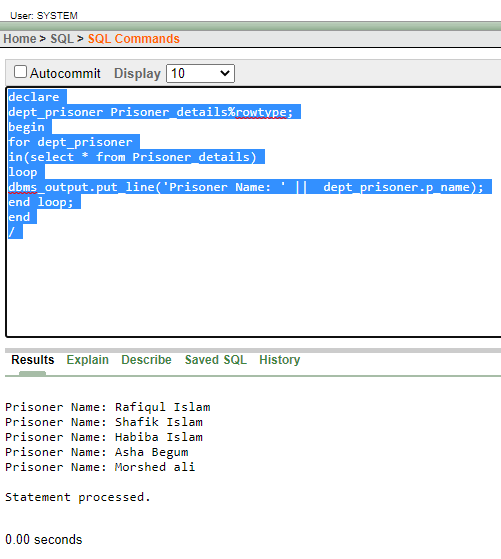
**loop**

**dbms\_output.put\_line('Prisoner Name: ' || dept\_prisoner.p\_name);**

**end loop;**

**end**

**/**



1. Create a record that can output the name of the item and quantity.

**declare**

**dept\_item item%rowtype;**

**begin**

**for dept\_item**

**in(select \* from item)**

**loop**

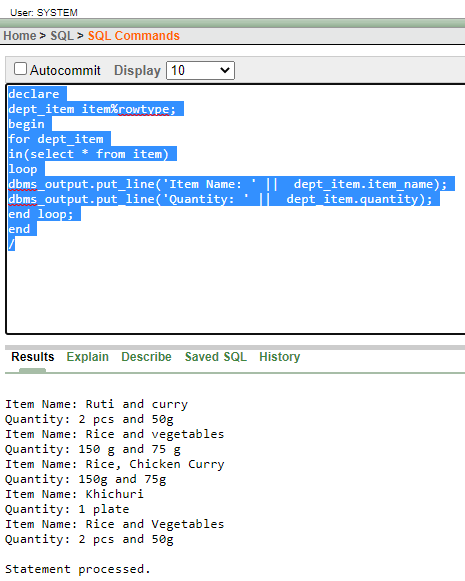
**dbms\_output.put\_line('Item Name: ' || dept\_item.item\_name);**

**dbms\_output.put\_line('Quantity: ' || dept\_item.quantity);**

**end loop;**

**end**

**/**



# **Cursor:**

1. Create a cursor that can output 1st prisoner’s name on the prisoner table.

**declare**

**pname Prisoner\_details.p\_name%type;**

**cursor p\_prisoner is**

**select p\_name from Prisoner\_details;**

**begin**

**open p\_prisoner;**

**fetch p\_prisoner into pname;**

**dbms\_output.put\_line('1st Prisoner Name: ' || pname);**

**close p\_prisoner;**

**end**

**/**



1. Create a cursor that can output the id and name of all the officers.

# **declare**

# **oid officer.o\_id%type;**

# **oname officer.o\_name%type;**

# **cursor o\_officer is**

# **select o\_id, o\_name from officer;**

# **begin**

# **open o\_officer;**

# **loop**

# **fetch o\_officer into oid,oname;**

# **exit when o\_officer%notfound;**

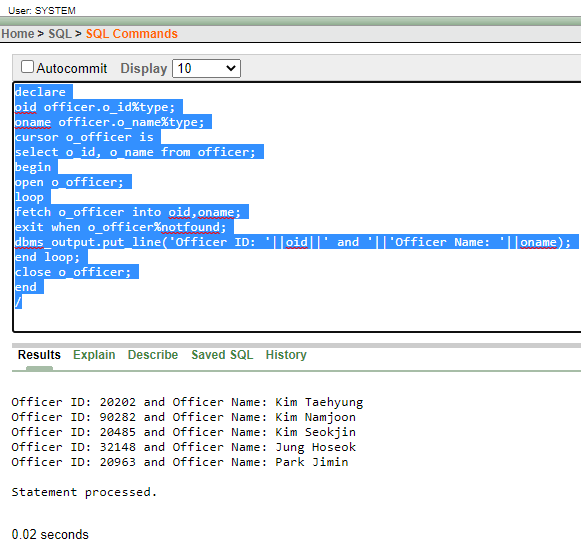
# **dbms\_output.put\_line('Officer ID: '||oid||' and '||'Officer Name: '||oname);**

# **end loop;**

# **close o\_officer;**

# **end**

# **/**



1. Create a cursor that can output the id and work title of all the prisoners.

**declare**

**pid working\_details.p\_id%type;**

**worktitle working\_details.work\_title%type;**

**cursor workingdetails is**

**select p\_id, work\_title from working\_details;**

**begin**

**open workingdetails;**

**loop**

**fetch workingdetails into pid,worktitle;**

**exit when workingdetails%notfound;**

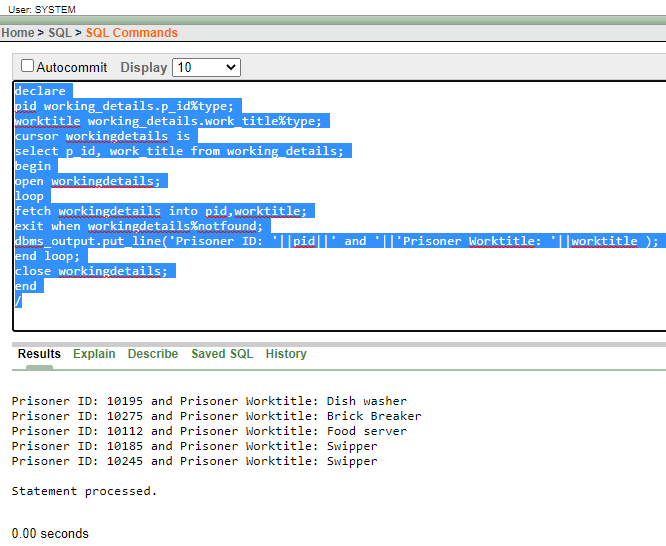
**dbms\_output.put\_line('Prisoner ID: '||pid||' and '||'Prisoner Worktitle: '||worktitle );**

**end loop;**

**close workingdetails;**

**end**

**/**



# **Trigger:**

1. Create a trigger in such a way that whenever a new row is inserted into the prisoner table an output ‘New Prisoner Added’ is generated.

**CREATE TRIGGER prisoner\_added**

**after INSERT ON Prisoner\_details**

**FOR EACH ROW**

**BEGIN**

**dbms\_output.put\_line('New Prisoner Added');**

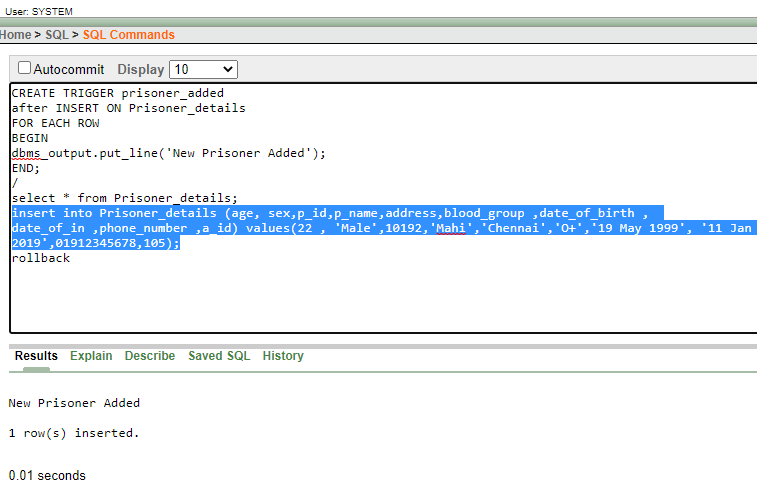
**END;**

**/**

**select \* from Prisoner\_details;**

**insert into Prisoner\_details (age, sex,p\_id,p\_name,address,blood\_group ,date\_of\_birth , date\_of\_in ,phone\_number ,a\_id) values(22 , 'Male',10192,'Mahi','Chennai','O+','19 May 1999', '11 Jan 2019',01912345678,105);**

**rollback**



1. Create a trigger in such a way that whenever a row is deleted from the prisoner table an output ‘A Prisoner Deleted’ is generated.

**CREATE TRIGGER prisoner\_d**

**after delete ON Prisoner\_details**

**FOR EACH ROW**

**BEGIN**

**dbms\_output.put\_line('A Prisoner Deleted');**

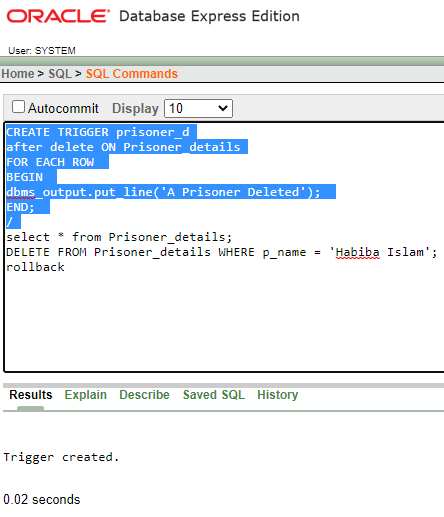
**END;**

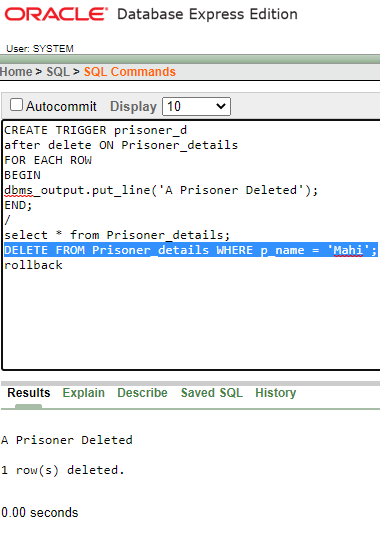
**/**

**select \* from Prisoner\_details;**

**DELETE FROM Prisoner\_details WHERE p\_name = 'Mahi';**

**rollback**





1. Create a trigger in such a way that whenever a row is updated from the prisoner table an output ‘A Prisoner Information Updated’ is generated.

**CREATE TRIGGER prisoner\_up**

**after update ON Prisoner\_details**

**FOR EACH ROW**

**BEGIN**

**dbms\_output.put\_line('A Prisoner Information Updated');**

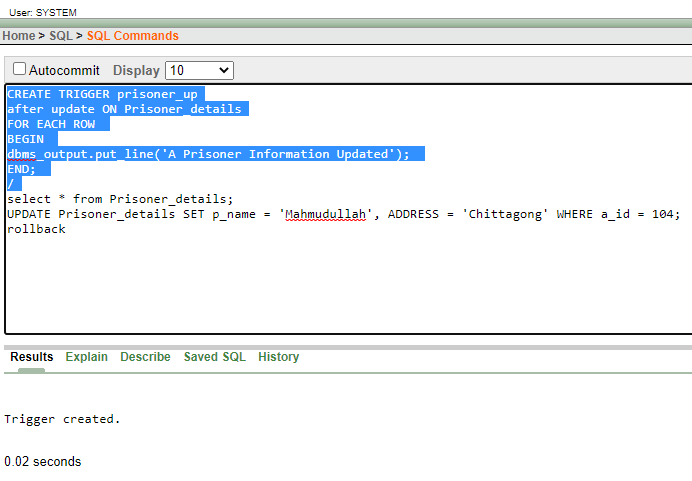
**END;**

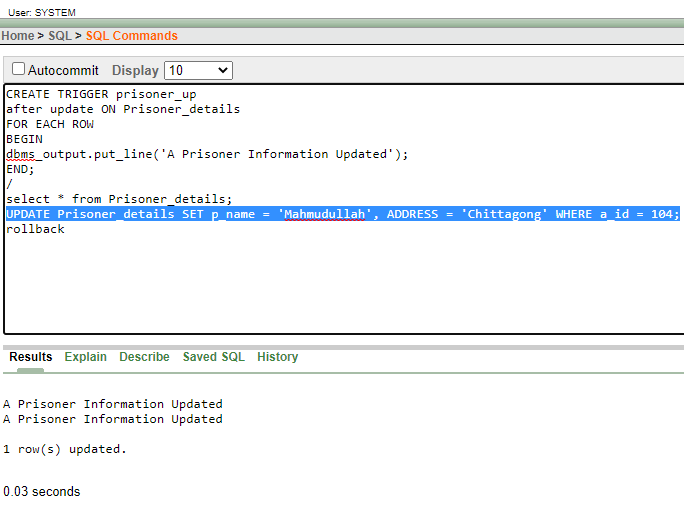
**/**

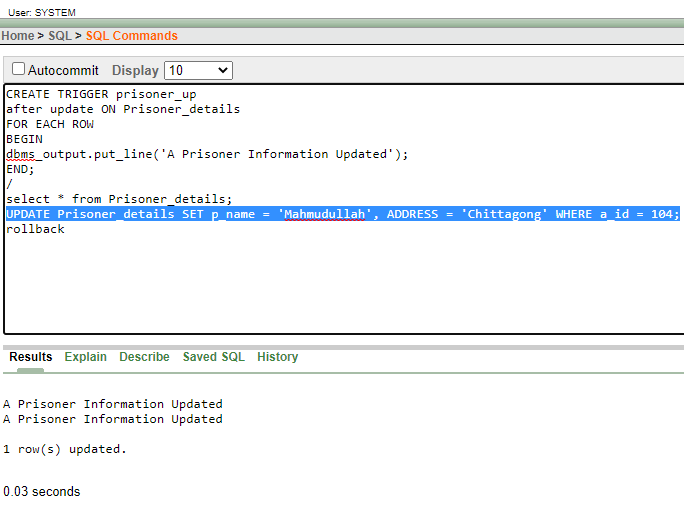
**select \* from Prisoner\_details;**

**UPDATE Prisoner\_details SET p\_name = 'Mahmudullah', ADDRESS = 'Chittagong' WHERE a\_id = 104;**

**rollback**







# **Conclusion:**

During the Jailhouse Management System project work, we learned the way to implement our theoretical knowledge in a real-life structure. We made a mind map where we structured our scenario and following that we drew the Entity-relationship model. After that, we normalized and got our final table to implement it on a database. We drew the schema diagram to visualize the tables and relations in a pictorial format. Then we created tables, manipulated the attributes where needed, and inserted data. Additionally, we learned important topics such as single-row function, group function, synonym subqueries, joining, view, user creation, role creation, relational algebra, and most importantly, PL/SQL which is a Procedural Language extension of SQL.

We tried to make the Jailhouse Management System in the best possible way so that this management system can be used at most of the prisons whereas there are some lackings such as could include entities such as medical centers, entertainment, sports, and maybe more. We faced several problems during the work period but we solved them together in a group discussion where we got a very clear view of every topic and the foundation of the database. We hope in the future we will perform more efficiently while doing this type of project using our learning and experience. If we can get the biometric data of prisoners, it would be so helpful for crime investigators to search for any criminal fingerprints and check if it actually matches in our records also, we can use 3D password security to admin login system for more authentication.