VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DATABASE MANAGEMENT SYSTEMS-ITE1003

J-COMPONENT

Review-1,2,3

GROUP MEMBERS-

19BIT0166 - AMAN SOMANI

19BIT0211 - ROHAN PAL

19BIT0411 - RITWIK SUKHIJA

TOPIC-

WOODEN FURNITURE PRODUCTION MANAGEMENT SYSTEM

CONTENTS-

ABSTRACT –

Our project basically deals with how WOODEN FURNITURE production occurs in an industry and how we can manage the whole factory with DBMS. It basically provides information regarding the basic FURNITURE manufacturing process such as raw material handling, types of people involved, departments handling different tasks. The database contains details of all the Engineers, workers, departments, their relationships among themselves etc. Through this PROJECT one may get a basic overview of the general aspects of a FURNITURE industry.

DATA REQUIREMENTS:

Entities and Attributes:

1) ENGINEER:

a) The database should be able to add details of the various details of the engineers who works in thecompany.

b) The database should include the Name of the engineer. It should be a string oflength
less than 10 characters and it can't be null.
c) The data must include a unique Engineer_ID for all the employees. Itshould be a 10-digitnumber.
d) The data should include the Address of the engineer that should be a string of
length less than 100 characters.
e) The data should include Monthly Salary of the employee which will be a number data type with 10digits.
f) The data should include the Gender of the engineer which will be a string of less than 10characters.

2) DEPARTMENTS:

a) There must be a unique Department Number for all the employees in the database. It should be a 10-digitnumber.
b) The database should include the Department Name and it should be a string of length
less than 50 characters and it can't be null.
c) The data must include Department Location that should be a string of less than 50words

3.) PROJECT:

a)Thedatabaseshouldbecontainingthevariousdetailsaboutprojectthatare going tohappen.
b) ThisshouldcontainattributelikeauniqueProjectNumberwhichmustbea 10-digitnumber.
c) The database should include the Name of the project. It should be a stringof length

less than 50 characters and it can’t be null.
d) The database must include the details of the project location which will must be a string of length less than 50characters.

4.)**WORKERS:**
a)The database should include the Worker Name and it should be a string of length less than 50 characters and it can’t benull.
b) The data must include a unique Worker_ID for all the employees. It should be a 10-digitnumber.
c) The data should include the Gender of the worker which will be a string of less than 10characters.
d) The data should include the Address of the worker that should be a string of length less than 100characters.
e) The data should include the Salary of the worker which will be a number data type with 10digits.

5.)**RAW WOOD:**
a) The database maintainer must be able to add data associated with the Woods.
b) The data must include the LOT_NO which should be a 5-digit unique number. Itcan't be leftnull.
c) The data must include the TYPE of wood. It must be a string of lengthless than 50.
d) The data should include the DATE OF CUTTING of the wood.It should bein date format. It can’t be null.
e) The data must include the QUANTITY of the woods which will be in tone unit. The data should be a Number of 10digits.
f) The data must include the PRICE of the woods, which should be a Number of less than 7digits.

6.) **CEO:**
a) The database maintainer should be able to add details of the various details of theCEO.
b) The data must include the NAME of the CEO. It should be a string oflength less than 50 characters. It can’t be left null.
c) The data should include a unique CEO_ID for all the employees. It should be a10-digitnumber.
d) The data should include the phone number of the CEO, which should be a 10 digitnumber.

e) The data should include Monthly Salary of the CEO which will be a number data type with 10digits.

7.)**MACHINES:**

a) The data must store the Machine Name which will be a string of less than 50 characters.
b) The database must include Machine Number which will be a 10-digit number.
c) The database should contain the data of Manufacturing Company which will be a string of less than 50characters.
d) The database should include the details of machine department which should be a string of less than 50characters.

8.)**FURNITURE:**

a) There should be Furniture Name which is made from the wood and it should be a string of length less than 50characters.
b) There should be Finishing House data which will be a string of length less than 50characters.
c) There must be a weight of that furniture from where we can calculate the final quantity of the woods. This data must be stored in number of less than 10 digits.
d) The data must include the Cost of the furniture and it should be a number that should be within 10digits.

Relationship among Entities:

1) Engineer **Handles** Project. One Engineer can handle many projects at a time and One project also can be worked upon by many engineers. As a result, the relationship here is M-N. (Many to Many). Since each Engineer have to take part in at least one project so, there is total participation of both theentities.

2) Engineer **Manages** Departments. One Engineer can manage only one engineering department at a particular time whereas a single department also can be managed by one Engineer at a time. So, the relationship is 1-1. (One to One) and Total participation of department whereas partial participation of Engineer.

3)Engineer **Works** inDepartment.
One Engineer can work for only one department at a time but a single department can have many number of Engineers so the relationship here is 1-N. (One to Many). There is also a total participation of both Engineers and departments

4) Engineer **Controls** workers. A single Engineer can control multiple workers and multiple number of workers are controlled by one Engineer. Therefore, the relationship here is 1-N. (One to Many). Since only few Engineers are allotted to control the workers so there will be a total participation of workers and partial participation ofEngineers.

5) Engineer **Reports** the CEO. Multiple Engineers report to only one CEO whereas a single CEO is reported to by more than one Engineers. Thus, the relationship here is N-1. (Many to One). Since only few Engineers report to the CEO there is partial participation of the Engineers and total participation of the CEO.

6) CEO **Sells** Furniture. One CEO can sell multiple Furniture whereas one furniture can be sold by the only the same CEO. Therefore, the relationship here is 1-N. (One to Many) Since, every furniture has to be sold through CEO only so there will be a total participation of CEO and total participation of Furniture

7) CEO **Supervises** Departments. A single CEO can supervise all the Engineering Departments and whereas a single department can be managed by only one CEO. So, the relationship here is clear 1-N. (One – Many) and there will be total participation of both theentities.

8) Workers **Operate** Machines. Many workers use a single machine whereas a single machine is being used by multiple workers. Thus, Cardinality constant is M-1. (Many to One). Since all the workers cannot operate a particular machine so there will be total participation of machines and partial participation of workers.

9) Workers **Worked Upon** raw woods. Since one worker can works on multiple Raw Woods and one raw wood can be worked upon by multiple workers so the relationship is M-N. (Many to many). There is total participation workers and partial participation of rawmaterials.
10) Workers **Aid in** Project. Many workers can aid in one Project only. So, the cardinality constant here is M-1. (Many to one). It isn't mandatory to work on a project every single worker at a time thus, partial participation of both workers andproject.

11) Machine **Produces** Furniture. A machine can make many Furniture whereas a single (Chairs, Beds, Doors, Wardrobes) furniture can be produced by only one machine. Hence the relationship is 1-N. (One to Many). Each and every machine does not produce furniture and every furniture is also not produced by only machines. As a result, there will be a partial participation of machines andFurniture.

12) Worker **Makes** Furniture. A worker can make one Furniture at a time whereas a single (Chairs, Beds, Doors, Wardrobes) furniture can be made by many workers. Hence the relationship is N-1. (Many to one). Each and every worker does not produce furniture and every furniture is also not produced by a worker only. As a result, there will be a partial participation for bothentities.

FUNCTIONAL REQUIREMENTS-

1.)**DATA RETRIEVAL:**

- To View the machinery(Name, Number, Manufacturing Company, Department)
- To View the details of the projects (Name, Number,Location)
- To View Manufactured Furniture(Name, Finishing house, Weight,Cost)
- To View the Departments (Number, Name,Location)
- To View the details of the raw wood used(Lot no., Type, Date of Cutting, Quantity,Price)
- To View the details of the workers(Name, Gender, Address,Salary)
- To View the details of the engineer(Name, ID, Address, Monthly Salary, Gender)
- To View the CEO's info(Name, ID, Phone Number, MonthlySalary)

2.) **DATA MODIFICATION:**

- Updating the Information on Wood and the Machinery- The user is able to modify and add information about the machinery if any new machinery or any new quality of wood is being used for manufacturing purposes.
- Updating the Information about the Projects- The user has the authority to modify and update the status of the current project and also the upcomingprojects.

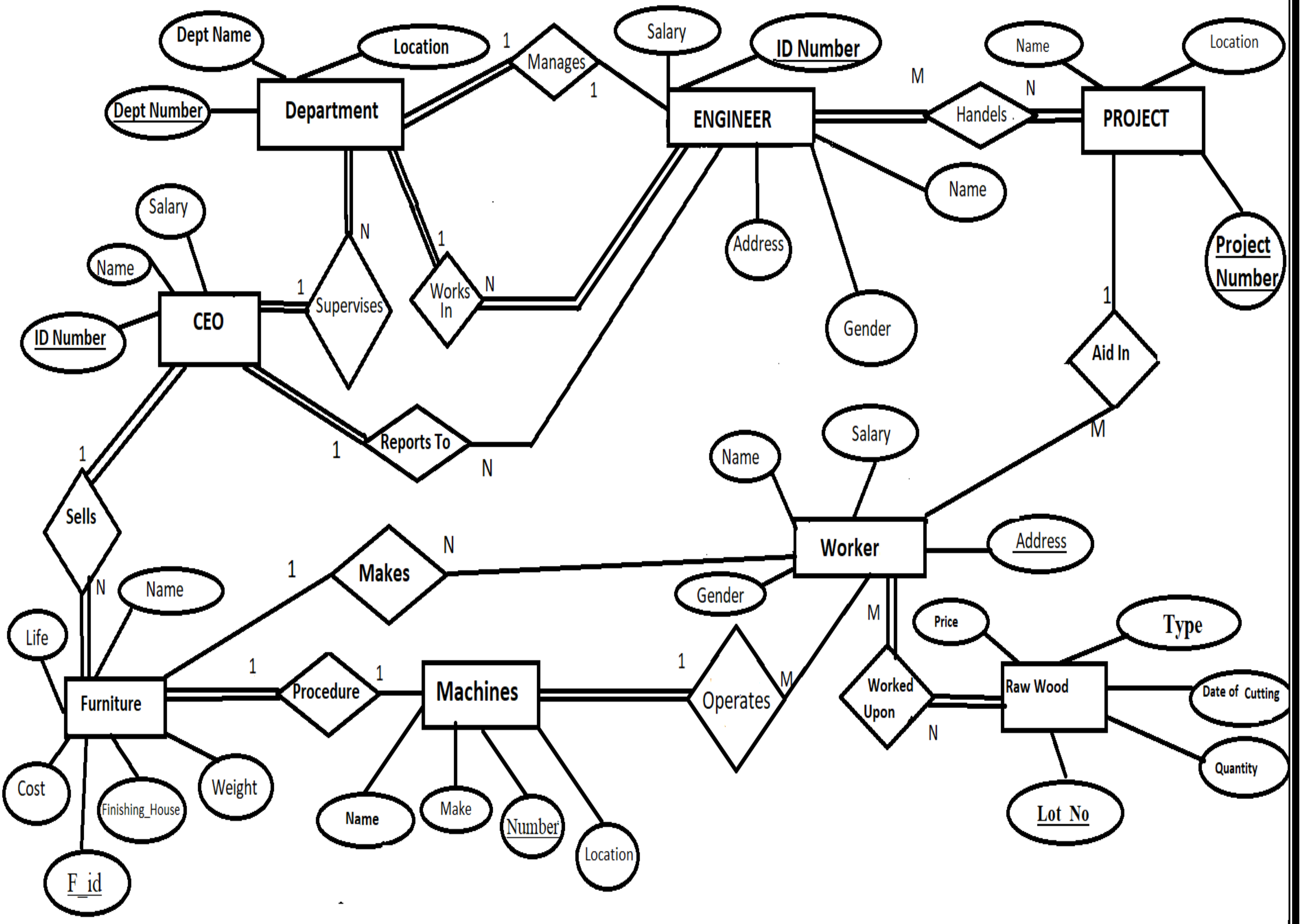
- Updating the Information Regarding the furniture manufactured- The user can update the information regarding the furniture manufactured if there are some changes and modification in the details of the furniture. For example, if the cost of the furniture is to be changed, the user can add the new cost in thedatabase.
- Updating the information about the raw wood- The user can also update the status of the raw wood if the date of cutting the wood is changedor

the quantity of wood required changes or if there is shoot or a dip in the price of the raw wood.

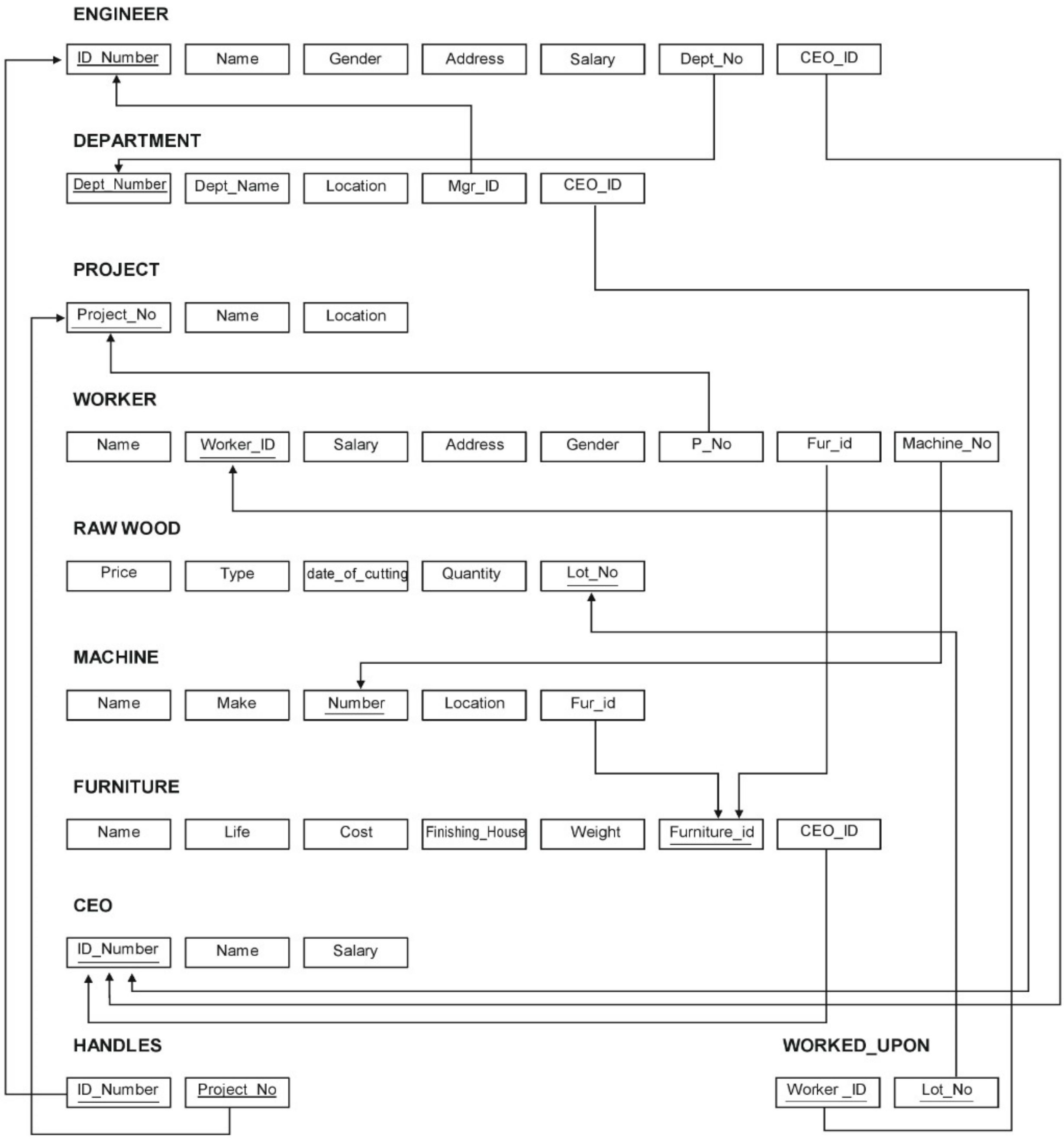
3.) **DATA DELETION:**

- Deleting the information regarding the machinery- The machinery information can be deleted or updated in the database if a certain machine fails to function properly or is under maintenance and out of use because it might result in wrong and undesired results and inadequate amount of energy might be used and other mishaps also might takeplace.
- Deleting the Engineer’s information- If a CEO leaves the job or if a new CEO is appointed then the information of the old CEO must be deleted from thedatabase.
- Deleting the Project Details- In case a project is completed or is abandoned due to lack or raw wood or other resourced, the information of that project should be removed from the database to avoidconfusion.
- Deleting the data of the Raw wood which got cancelled- Ifthe supply of some raw wood gets cancelled then, the related details such as lot no., quantity, price etc. must be removed from the database to avoid confusion.

E-R DIAGRAM:

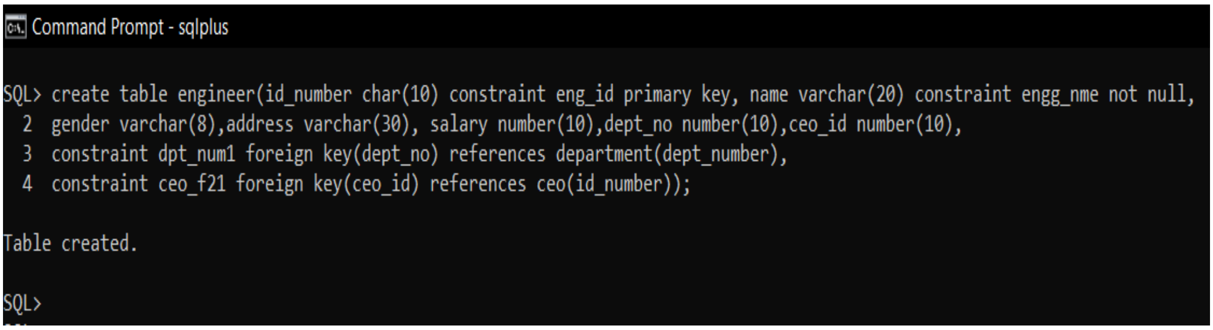

Dept Name
Location
1
Manages
1
Salary
ID Number
Name
Location
Dept Number
Department
ENGINEER
M
Handels
N
PROJECT
Salary
Name
Name
Project Number
N
1
Address
ID Number
CEO
1
Supervises
Works In
N
Gender
1
Aid In
M
1
Reports To
N
Name
Salary
1
Sells
N
Makes
1
Worker
Address
N
Name
Gender
M
Price
Type
Life
1
Procedure
1
Machines
1
Operates
M
Worked Upon
Raw Wood
Date of Cutting
N
Furniture
Quantity
Cost
Finishing_House
Weight
Name
Make
Number
Location
Lot No
F_id

- ***RELATIONAL DATABASE SCHEMA-***



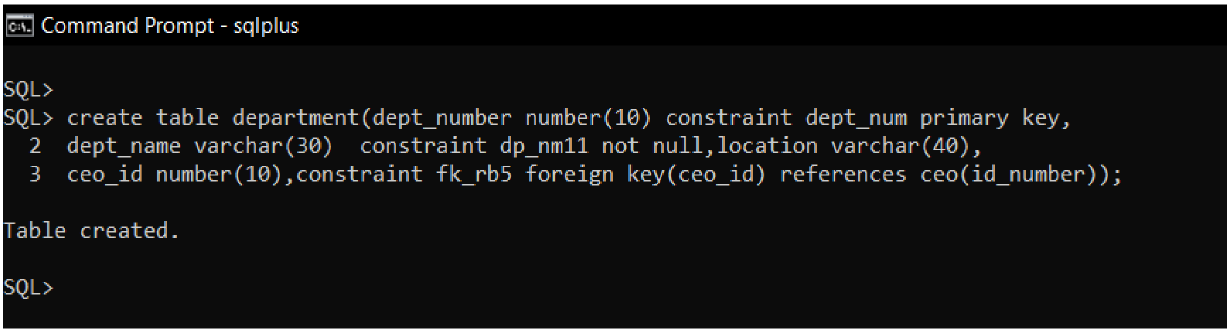
• CREATION OF TABLES-

1) ENGINEER-



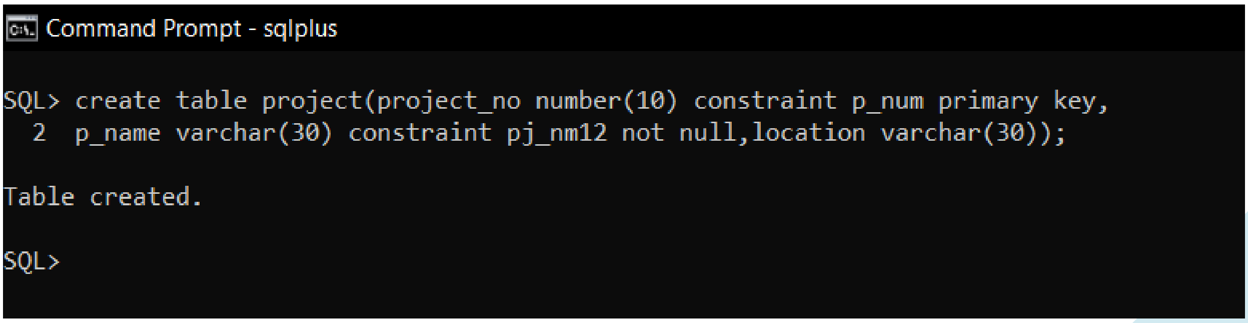
```
SQL> create table engineer(id_number char(10) constraint eng_id primary key, name varchar(20) constraint engg_nme not null,
  2  gender varchar(8),address varchar(30), salary number(10),dept_no number(10),ceo_id number(10),
  3  constraint dpt_num1 foreign key(dept_no) references department(dept_number),
  4  constraint ceo_f21 foreign key(ceo_id) references ceo(id_number));

Table created.

SQL>
```

2) DEPARTMENT-



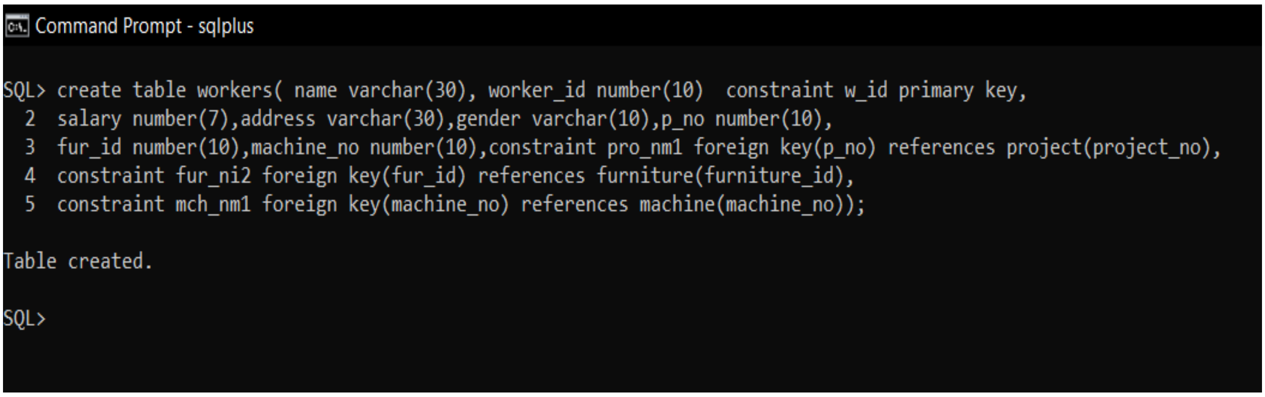
```
SQL>
SQL> create table department(dept_number number(10) constraint dept_num primary key,
  2  dept_name varchar(30)  constraint dp_nm11 not null,location varchar(40),
  3  ceo_id number(10),constraint fk_rb5 foreign key(ceo_id) references ceo(id_number));

Table created.

SQL>
```

3) PROJECT-



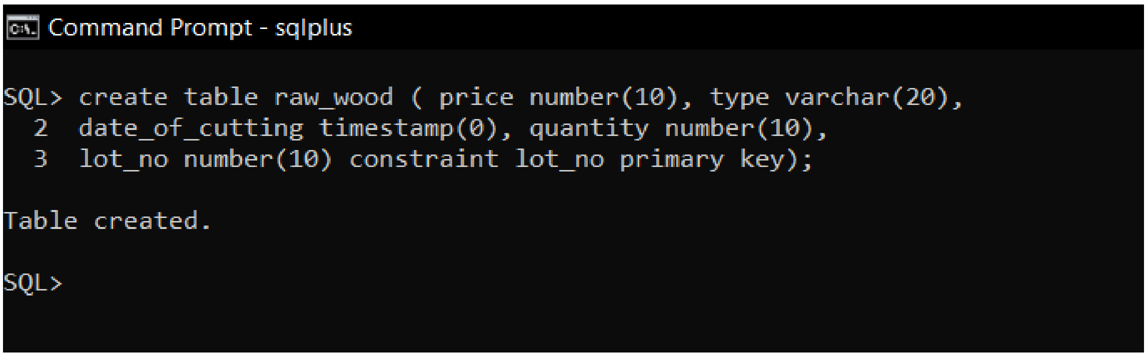
```
SQL> create table project(project_no number(10) constraint p_num primary key,
  2  p_name varchar(30) constraint pj_nm12 not null,location varchar(30));

Table created.

SQL>
```

4) **WORKER-**



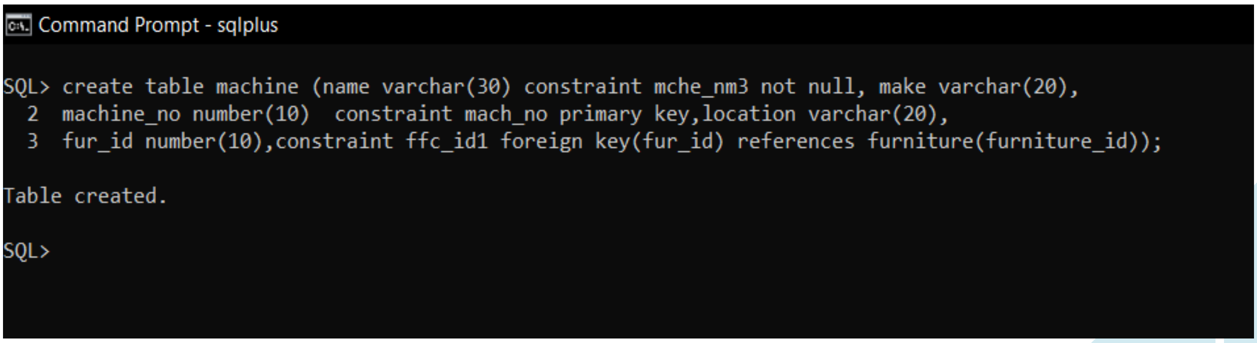
```
SQL> create table workers( name varchar(30), worker_id number(10)  constraint w_id primary key,
     salary number(7),address varchar(30),gender varchar(10),p_no number(10),
     fur_id number(10),machine_no number(10),constraint pro_nm1 foreign key(p_no) references project(project_no),
     constraint fur_ni2 foreign key(fur_id) references furniture(furniture_id),
     constraint mch_nm1 foreign key(machine_no) references machine(machine_no));

Table created.

SQL>
```

5) **RAW WOOD-**



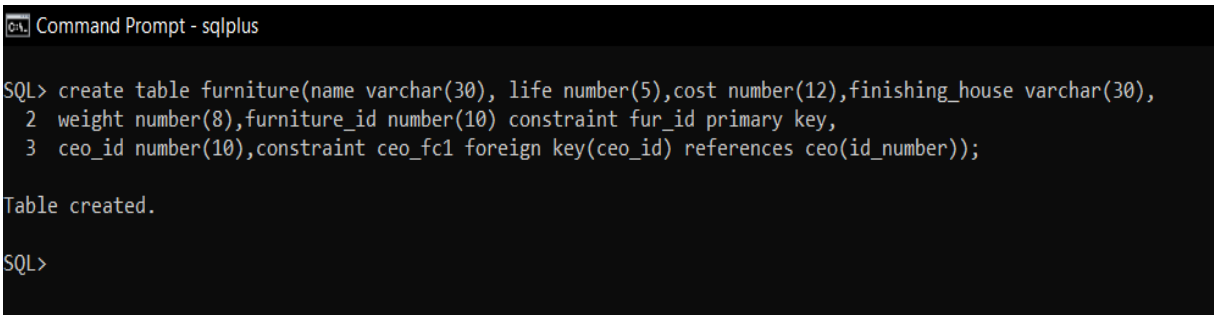
```
SQL> create table raw_wood ( price number(10), type varchar(20),
     date_of_cutting timestamp(0), quantity number(10),
     lot_no number(10) constraint lot_no primary key);

Table created.

SQL>
```

6) **MACHINE-**



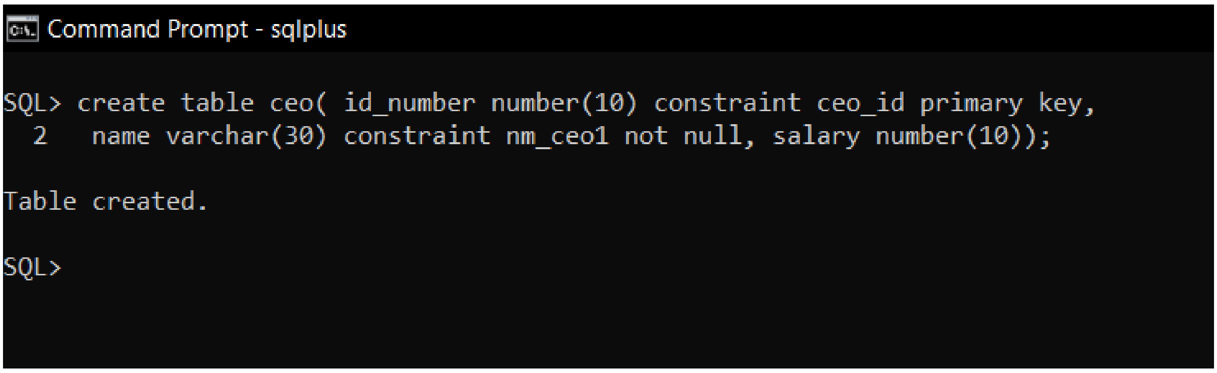
```
SQL> create table machine (name varchar(30) constraint mche_nm3 not null, make varchar(20),
     machine_no number(10)  constraint mach_no primary key,location varchar(20),
     fur_id number(10),constraint ffc_id1 foreign key(fur_id) references furniture(furniture_id));

Table created.

SQL>
```

7) FURNITURE-



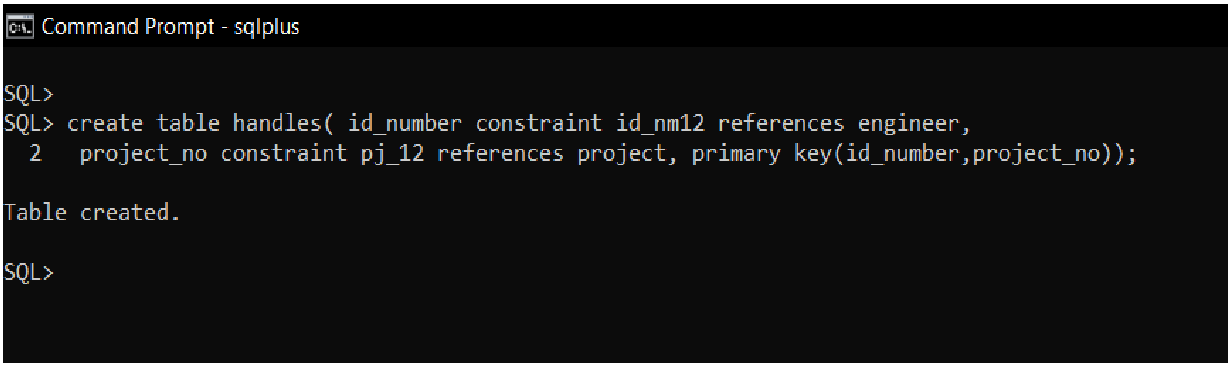
```
SQL> create table furniture(name varchar(30), life number(5),cost number(12),finishing_house varchar(30),
  2  weight number(8),furniture_id number(10) constraint fur_id primary key,
  3  ceo_id number(10),constraint ceo_fc1 foreign key(ceo_id) references ceo(id_number));

Table created.

SQL>
```

8) CEO-



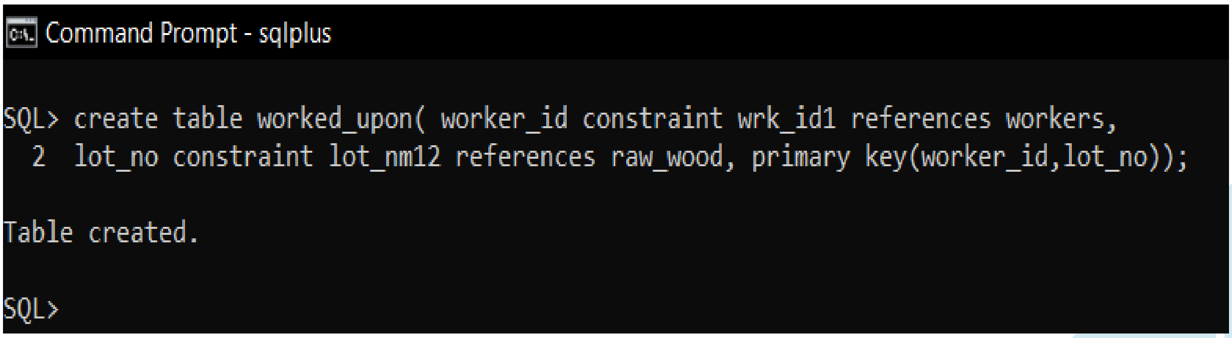
```
SQL> create table ceo( id_number number(10) constraint ceo_id primary key,
  2   name varchar(30) constraint nm_ceo1 not null, salary number(10));

Table created.

SQL>
```

9) HANDLES-



```
SQL>
SQL> create table handles( id_number constraint id_nm12 references engineer,
  2   project_no constraint pj_12 references project, primary key(id_number,project_no));

Table created.

SQL>
```

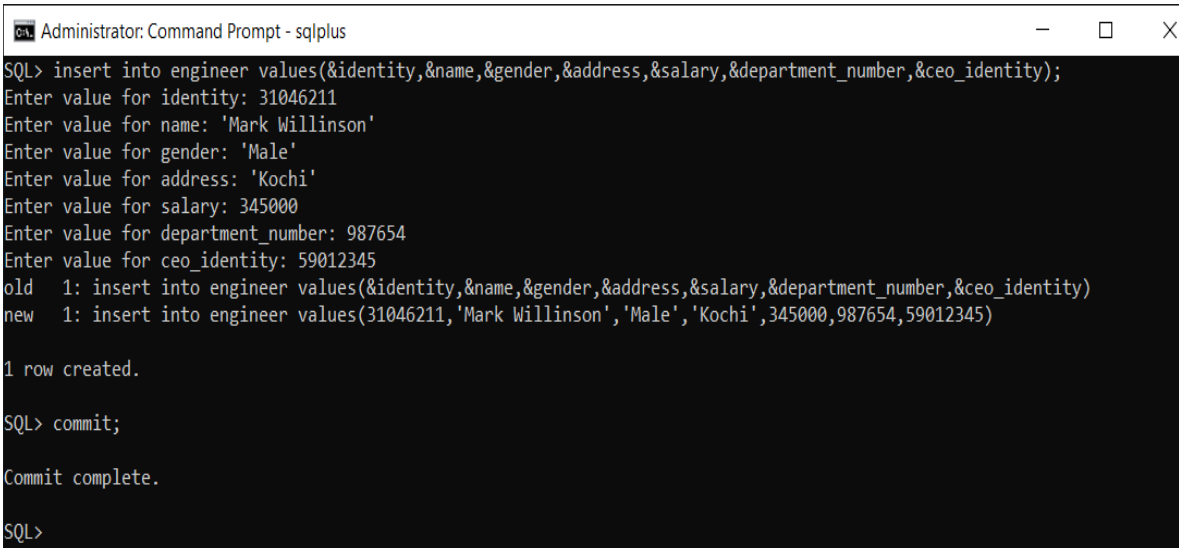
10) WORKED UPON-



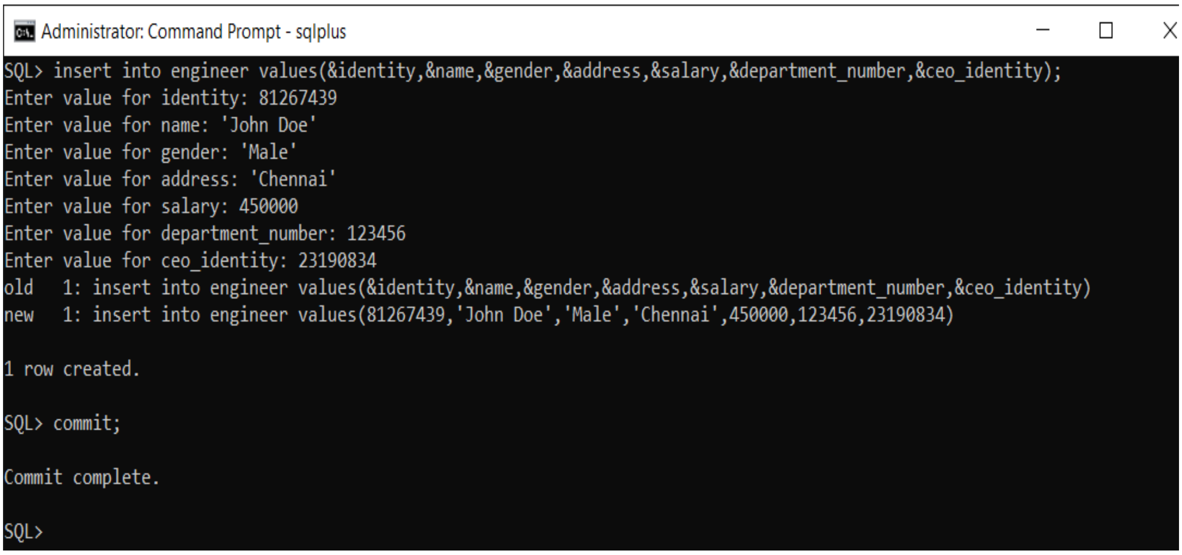
```
SQL> create table worked_upon( worker_id constraint wrk_id1 references workers,
  2  lot_no constraint lot_nm12 references raw_wood, primary key(worker_id,lot_no));

Table created.

SQL>
```

- **INSERTING ROWS IN THE ABOVE CREATED TABLES-**

1) **ENGINEER-**

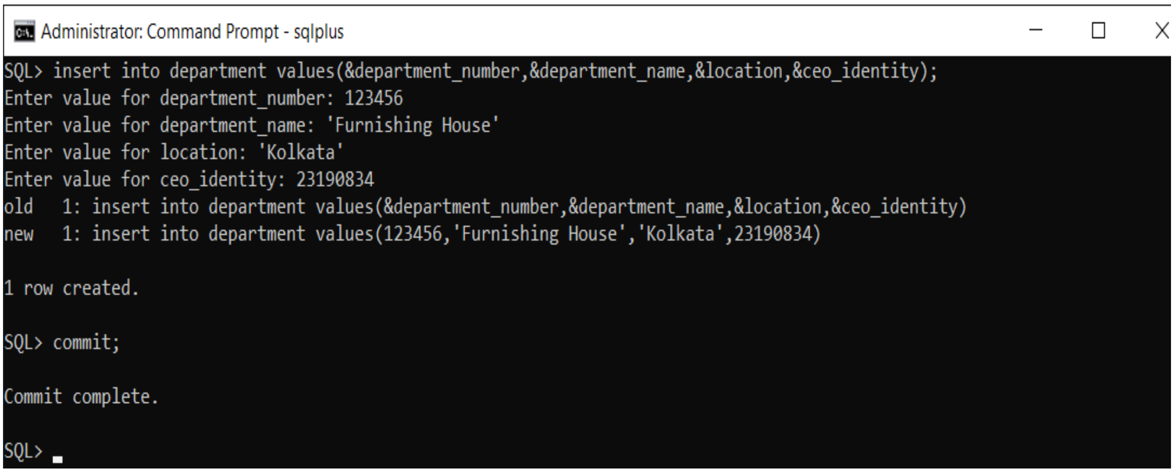


```
SQL> insert into engineer values(&identity,&name,&gender,&address,&salary,&department_number,&ceo_identity);
Enter value for identity: 31046211
Enter value for name: 'Mark Willinson'
Enter value for gender: 'Male'
Enter value for address: 'Kochi'
Enter value for salary: 345000
Enter value for department_number: 987654
Enter value for ceo_identity: 59012345
old   1: insert into engineer values(&identity,&name,&gender,&address,&salary,&department_number,&ceo_identity)
new   1: insert into engineer values(31046211,'Mark Willinson','Male','Kochi',345000,987654,59012345)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

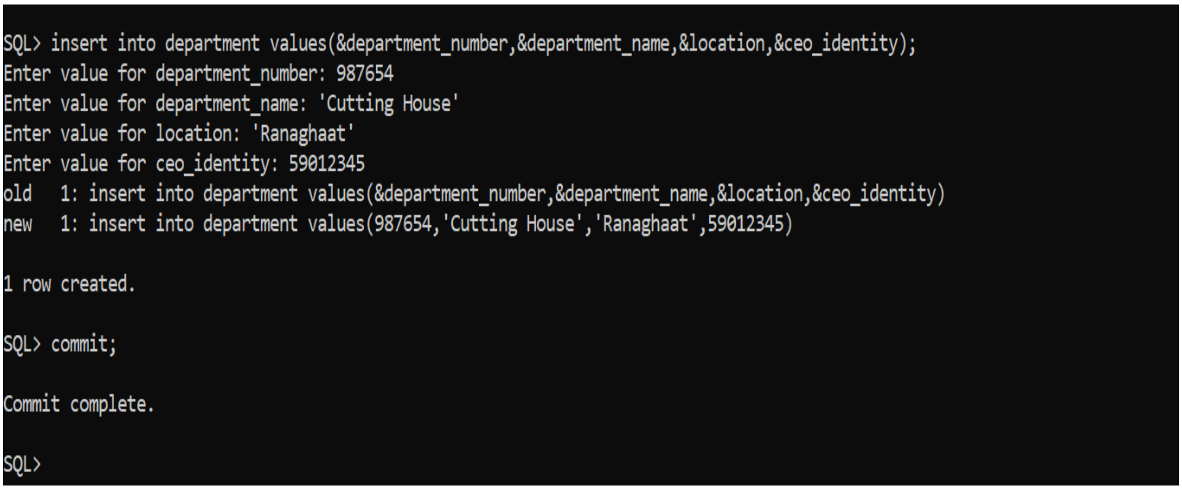


```
SQL> insert into engineer values(&identity,&name,&gender,&address,&salary,&department_number,&ceo_identity);
Enter value for identity: 81267439
Enter value for name: 'John Doe'
Enter value for gender: 'Male'
Enter value for address: 'Chennai'
Enter value for salary: 450000
Enter value for department_number: 123456
Enter value for ceo_identity: 23190834
old   1: insert into engineer values(&identity,&name,&gender,&address,&salary,&department_number,&ceo_identity)
new   1: insert into engineer values(81267439,'John Doe','Male','Chennai',450000,123456,23190834)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

2) DEPARTMENT-

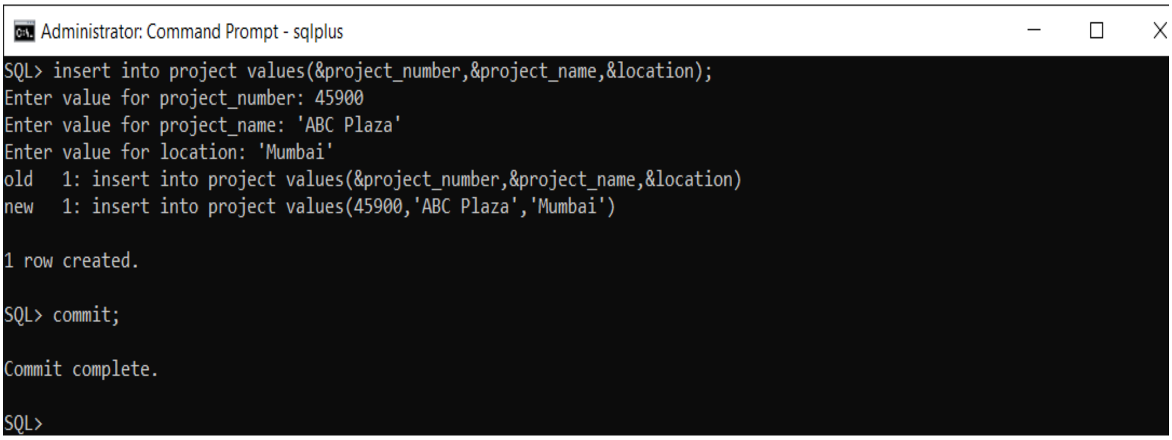


```
SQL> insert into department values(&department_number,&department_name,&location,&ceo_identity);
Enter value for department_number: 123456
Enter value for department_name: 'Furnishing House'
Enter value for location: 'Kolkata'
Enter value for ceo_identity: 23190834
old   1: insert into department values(&department_number,&department_name,&location,&ceo_identity)
new   1: insert into department values(123456,'Furnishing House','Kolkata',23190834)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

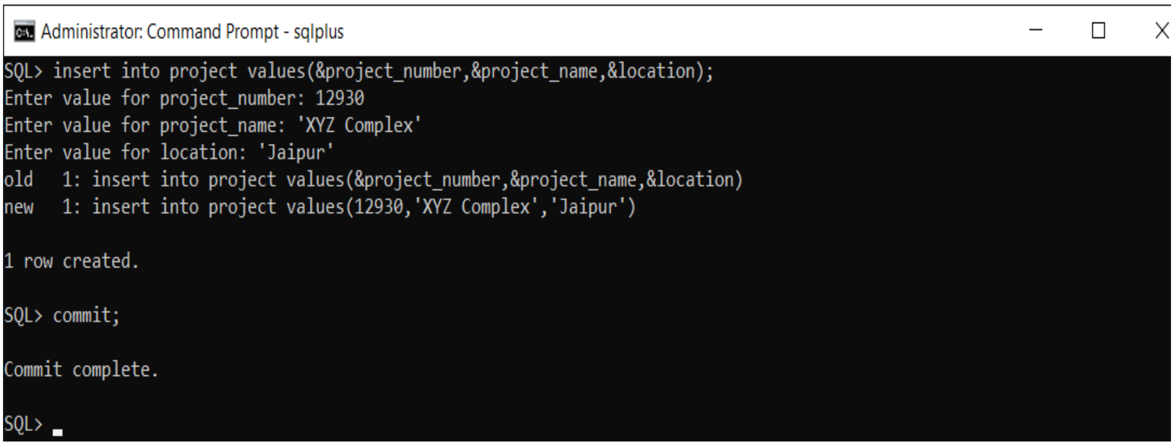


```
SQL> insert into department values(&department_number,&department_name,&location,&ceo_identity);
Enter value for department_number: 987654
Enter value for department_name: 'Cutting House'
Enter value for location: 'Ranaghaat'
Enter value for ceo_identity: 59012345
old   1: insert into department values(&department_number,&department_name,&location,&ceo_identity)
new   1: insert into department values(987654,'Cutting House','Ranaghaat',59012345)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

3) PROJECT-

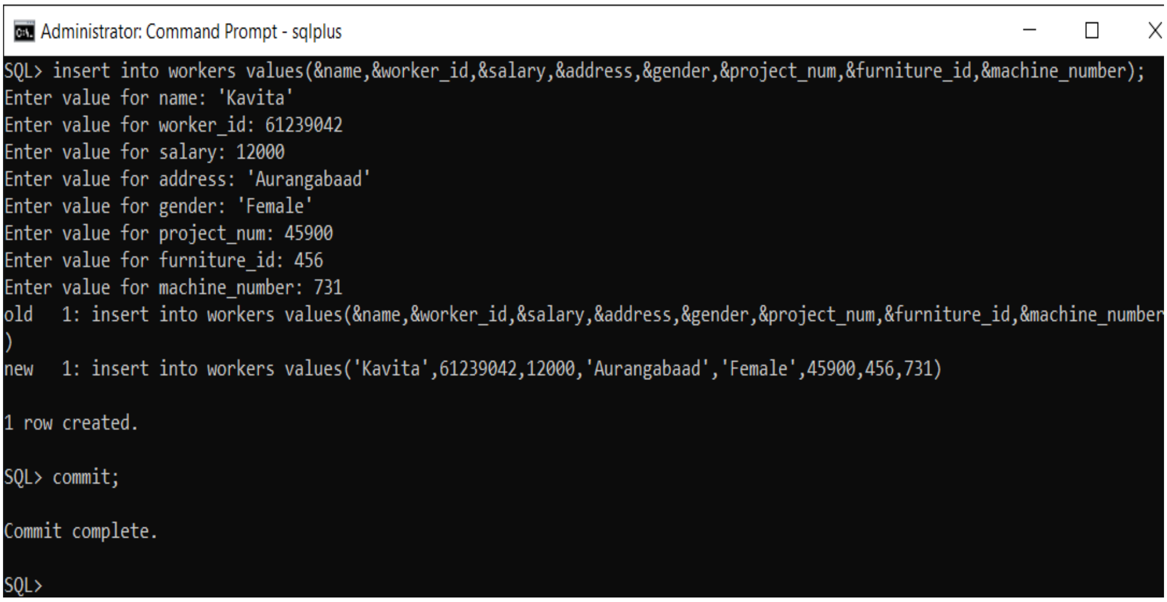


```
SQL> insert into project values(&project_number,&project_name,&location);
Enter value for project_number: 45900
Enter value for project_name: 'ABC Plaza'
Enter value for location: 'Mumbai'
old   1: insert into project values(&project_number,&project_name,&location)
new   1: insert into project values(45900,'ABC Plaza','Mumbai')

1 row created.

SQL> commit;

Commit complete.

SQL>
```

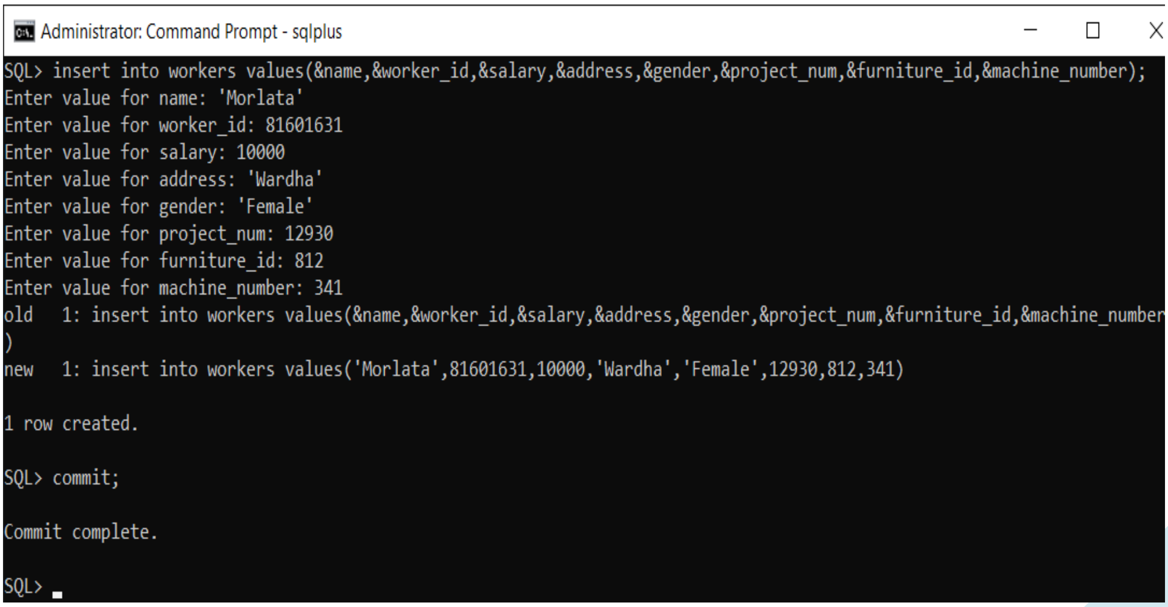


```
SQL> insert into project values(&project_number,&project_name,&location);
Enter value for project_number: 12930
Enter value for project_name: 'XYZ Complex'
Enter value for location: 'Jaipur'
old   1: insert into project values(&project_number,&project_name,&location)
new   1: insert into project values(12930,'XYZ Complex','Jaipur')

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

4) WORKERS-

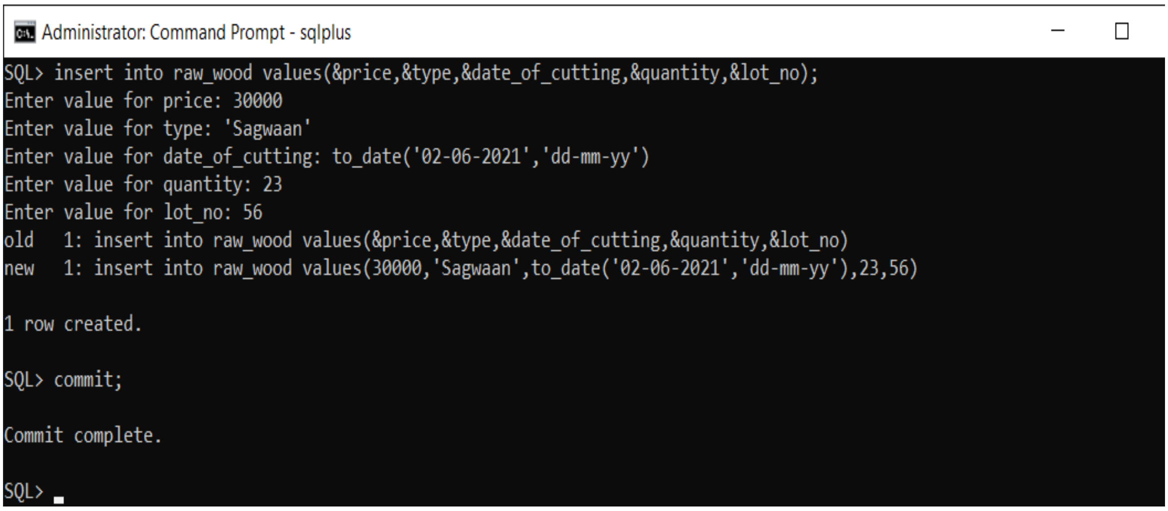


```
SQL> insert into workers values(&name,&worker_id,&salary,&address,&gender,&project_num,&furniture_id,&machine_number);
Enter value for name: 'Kavita'
Enter value for worker_id: 61239042
Enter value for salary: 12000
Enter value for address: 'Aurangabaad'
Enter value for gender: 'Female'
Enter value for project_num: 45900
Enter value for furniture_id: 456
Enter value for machine_number: 731
old   1: insert into workers values(&name,&worker_id,&salary,&address,&gender,&project_num,&furniture_id,&machine_number
)
new   1: insert into workers values('Kavita',61239042,12000,'Aurangabaad','Female',45900,456,731)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

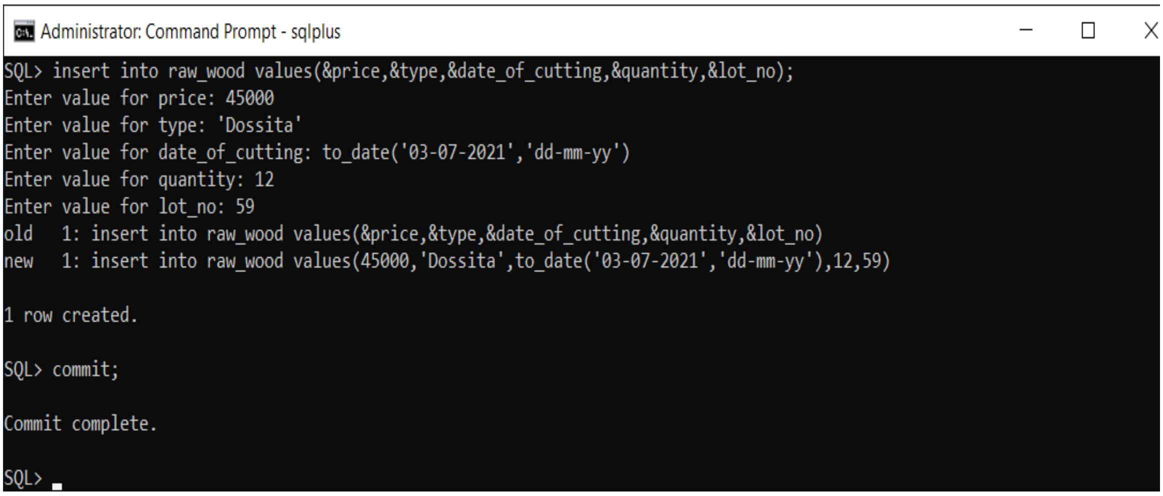


```
SQL> insert into workers values(&name,&worker_id,&salary,&address,&gender,&project_num,&furniture_id,&machine_number);
Enter value for name: 'Morlata'
Enter value for worker_id: 81601631
Enter value for salary: 10000
Enter value for address: 'Wardha'
Enter value for gender: 'Female'
Enter value for project_num: 12930
Enter value for furniture_id: 812
Enter value for machine_number: 341
old   1: insert into workers values(&name,&worker_id,&salary,&address,&gender,&project_num,&furniture_id,&machine_number
)
new   1: insert into workers values('Morlata',81601631,10000,'Wardha','Female',12930,812,341)

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

5) RAW WOOD-

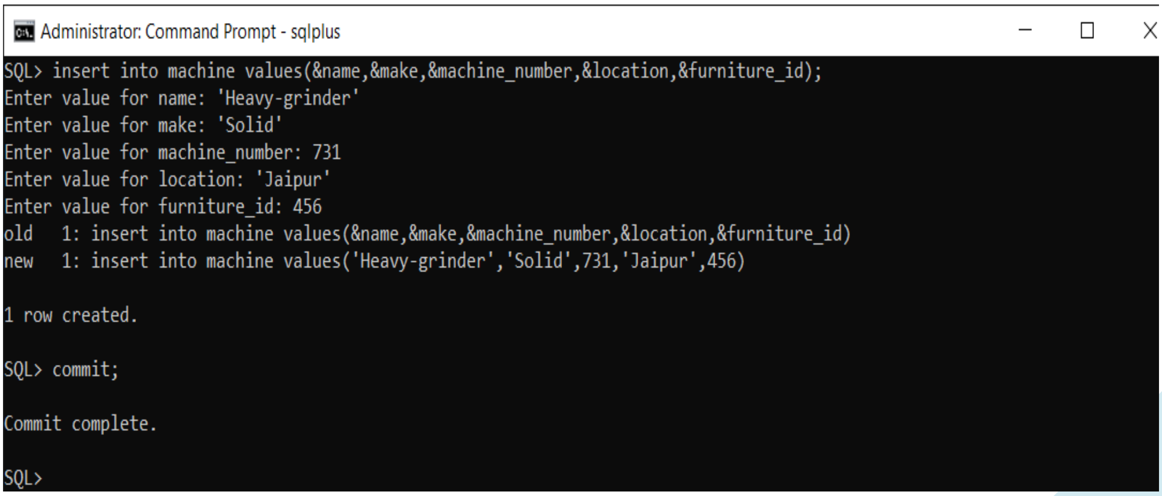


```
SQL> insert into raw_wood values(&price,&type,&date_of_cutting,&quantity,&lot_no);
Enter value for price: 30000
Enter value for type: 'Sagwaan'
Enter value for date_of_cutting: to_date('02-06-2021','dd-mm-yy')
Enter value for quantity: 23
Enter value for lot_no: 56
old   1: insert into raw_wood values(&price,&type,&date_of_cutting,&quantity,&lot_no)
new   1: insert into raw_wood values(30000,'Sagwaan',to_date('02-06-2021','dd-mm-yy'),23,56)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

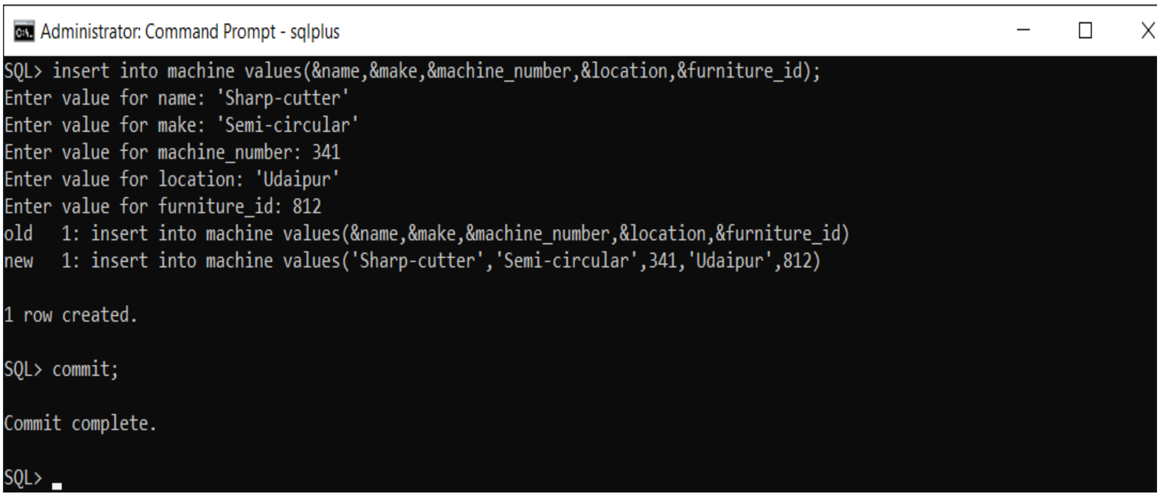


```
SQL> insert into raw_wood values(&price,&type,&date_of_cutting,&quantity,&lot_no);
Enter value for price: 45000
Enter value for type: 'Dossita'
Enter value for date_of_cutting: to_date('03-07-2021','dd-mm-yy')
Enter value for quantity: 12
Enter value for lot_no: 59
old   1: insert into raw_wood values(&price,&type,&date_of_cutting,&quantity,&lot_no)
new   1: insert into raw_wood values(45000,'Dossita',to_date('03-07-2021','dd-mm-yy'),12,59)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

6) MACHINE-

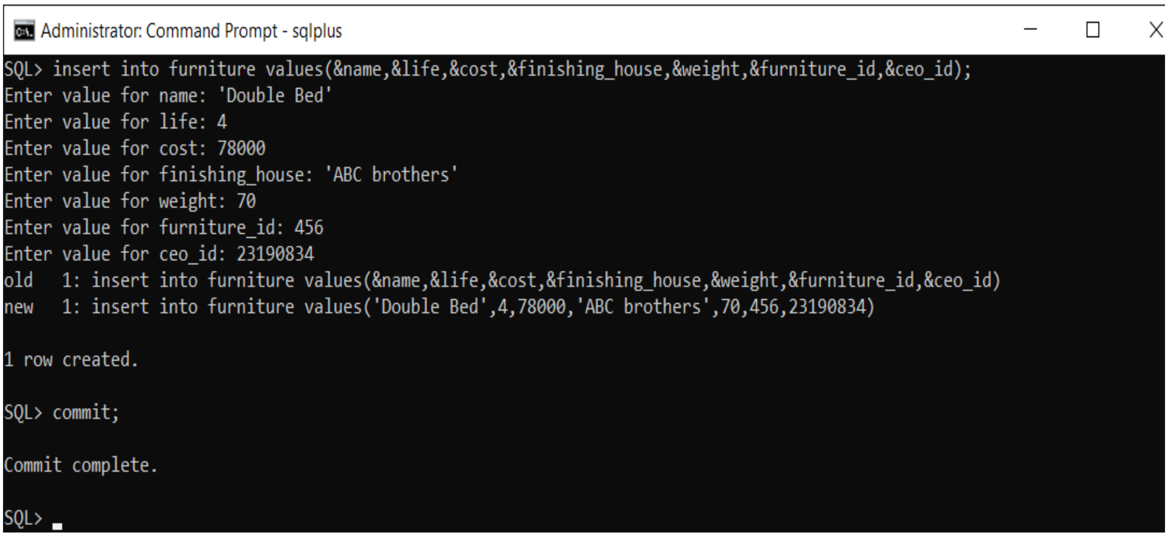


```
SQL> insert into machine values(&name,&make,&machine_number,&location,&furniture_id);
Enter value for name: 'Heavy-grinder'
Enter value for make: 'Solid'
Enter value for machine_number: 731
Enter value for location: 'Jaipur'
Enter value for furniture_id: 456
old   1: insert into machine values(&name,&make,&machine_number,&location,&furniture_id)
new   1: insert into machine values('Heavy-grinder','Solid',731,'Jaipur',456)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

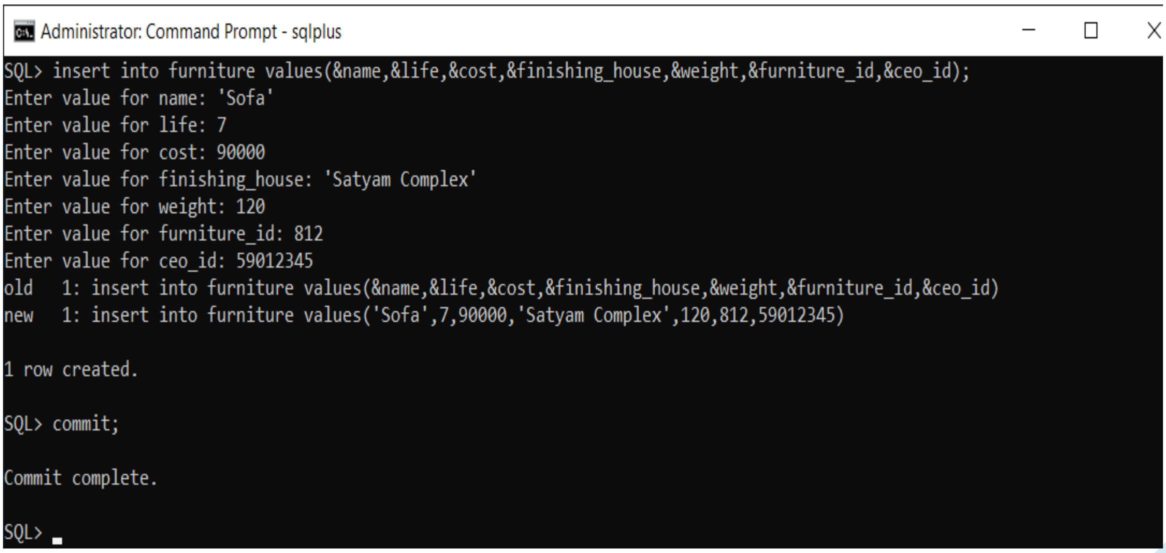


```
SQL> insert into machine values(&name,&make,&machine_number,&location,&furniture_id);
Enter value for name: 'Sharp-cutter'
Enter value for make: 'Semi-circular'
Enter value for machine_number: 341
Enter value for location: 'Udaipur'
Enter value for furniture_id: 812
old   1: insert into machine values(&name,&make,&machine_number,&location,&furniture_id)
new   1: insert into machine values('Sharp-cutter','Semi-circular',341,'Udaipur',812)

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

7) FURNITURE-

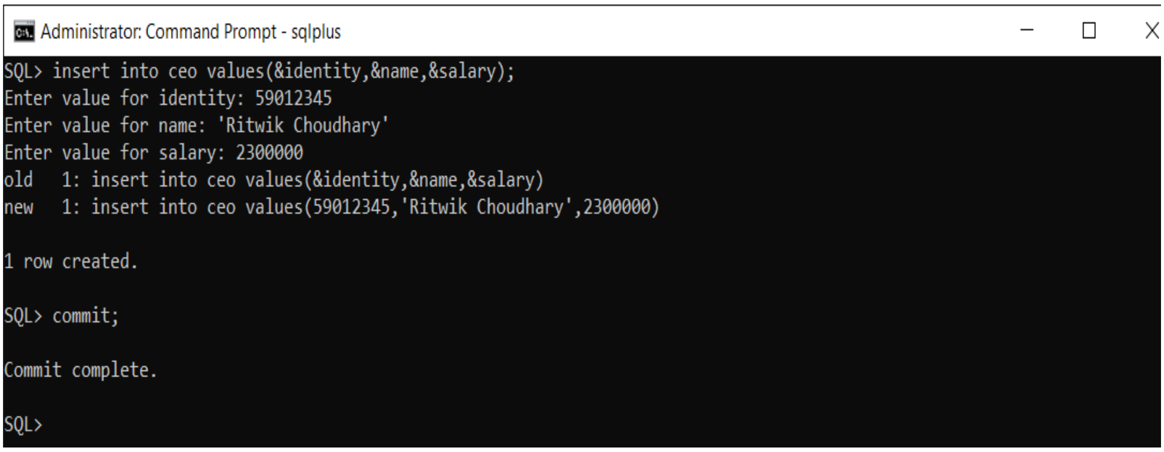


```
SQL> insert into furniture values(&name,&life,&cost,&finishing_house,&weight,&furniture_id,&ceo_id);
Enter value for name: 'Double Bed'
Enter value for life: 4
Enter value for cost: 78000
Enter value for finishing_house: 'ABC brothers'
Enter value for weight: 70
Enter value for furniture_id: 456
Enter value for ceo_id: 23190834
old   1: insert into furniture values(&name,&life,&cost,&finishing_house,&weight,&furniture_id,&ceo_id)
new   1: insert into furniture values('Double Bed',4,78000,'ABC brothers',70,456,23190834)

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

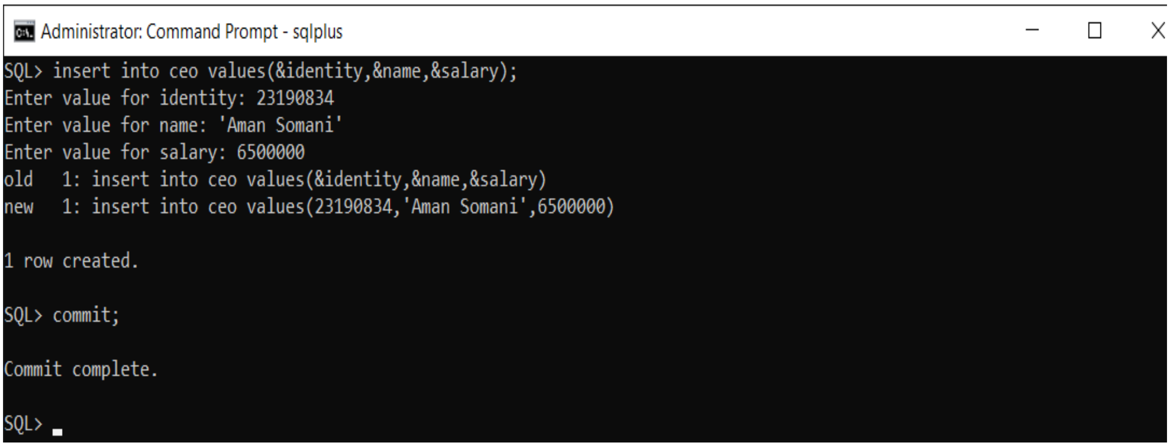


```
SQL> insert into furniture values(&name,&life,&cost,&finishing_house,&weight,&furniture_id,&ceo_id);
Enter value for name: 'Sofa'
Enter value for life: 7
Enter value for cost: 90000
Enter value for finishing_house: 'Satyam Complex'
Enter value for weight: 120
Enter value for furniture_id: 812
Enter value for ceo_id: 59012345
old   1: insert into furniture values(&name,&life,&cost,&finishing_house,&weight,&furniture_id,&ceo_id)
new   1: insert into furniture values('Sofa',7,90000,'Satyam Complex',120,812,59012345)

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

8) CEO-

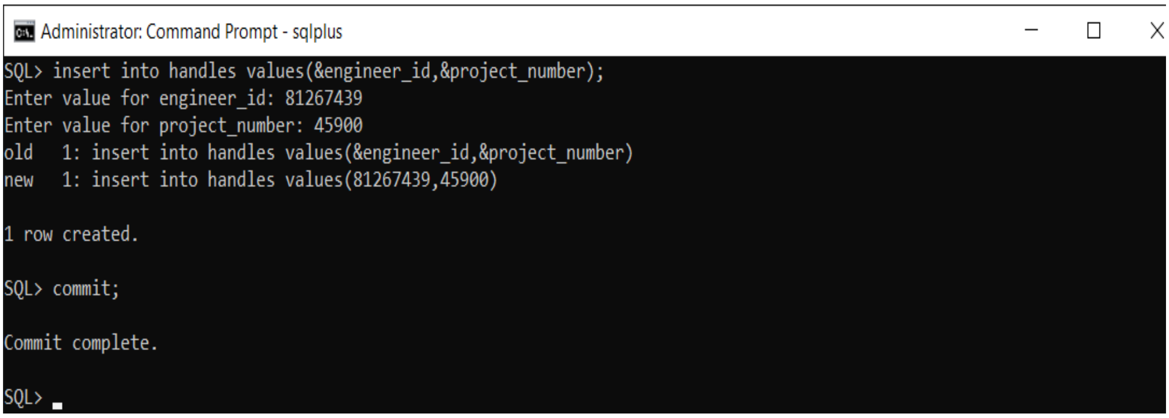


```
SQL> insert into ceo values(&identity,&name,&salary);
Enter value for identity: 59012345
Enter value for name: 'Ritwik Choudhary'
Enter value for salary: 2300000
old   1: insert into ceo values(&identity,&name,&salary)
new   1: insert into ceo values(59012345,'Ritwik Choudhary',2300000)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

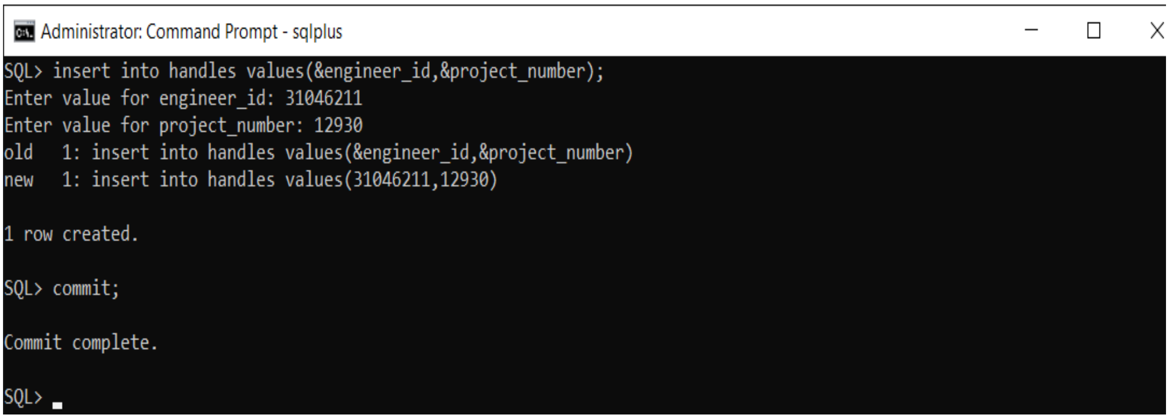


```
SQL> insert into ceo values(&identity,&name,&salary);
Enter value for identity: 23190834
Enter value for name: 'Aman Somani'
Enter value for salary: 6500000
old   1: insert into ceo values(&identity,&name,&salary)
new   1: insert into ceo values(23190834,'Aman Somani',6500000)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

9) HANDELS-

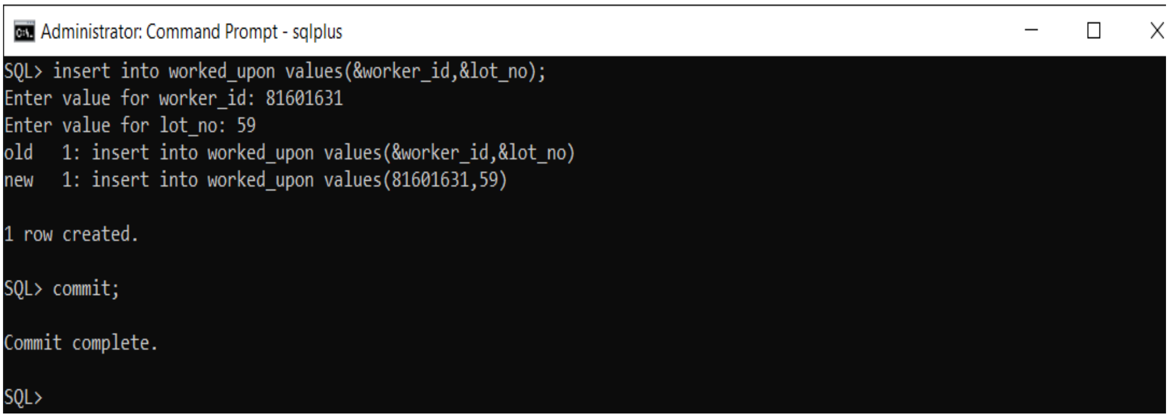


```
SQL> insert into handles values(&engineer_id,&project_number);
Enter value for engineer_id: 81267439
Enter value for project_number: 45900
old   1: insert into handles values(&engineer_id,&project_number)
new   1: insert into handles values(81267439,45900)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

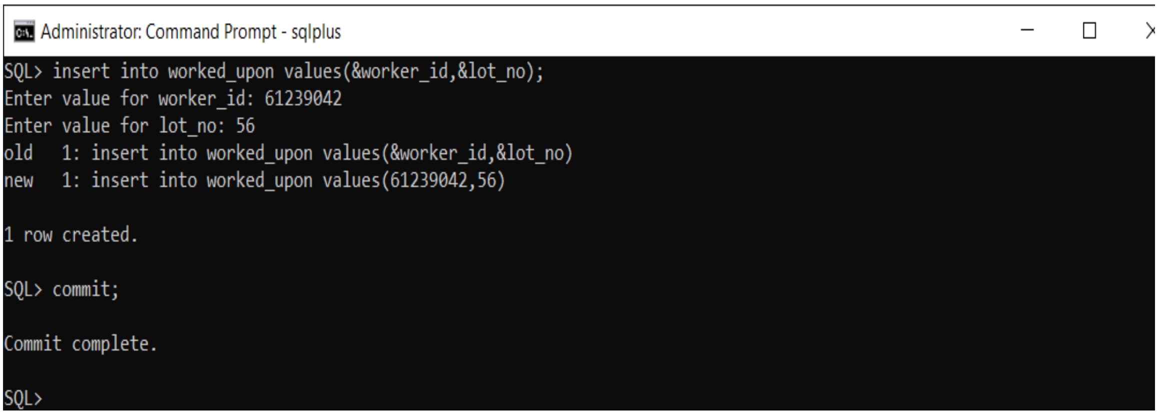


```
SQL> insert into handles values(&engineer_id,&project_number);
Enter value for engineer_id: 31046211
Enter value for project_number: 12930
old   1: insert into handles values(&engineer_id,&project_number)
new   1: insert into handles values(31046211,12930)

1 row created.

SQL> commit;

Commit complete.

SQL> _
```

10) WORKED UPON-



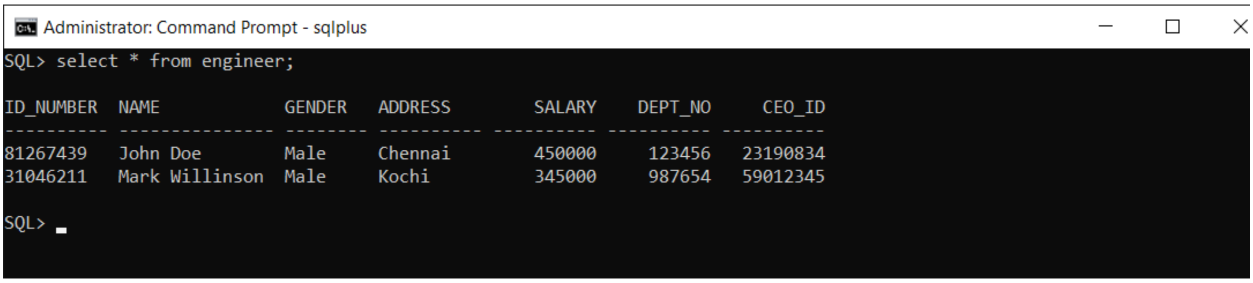
```
SQL> insert into worked_upon values(&worker_id,&lot_no);
Enter value for worker_id: 81601631
Enter value for lot_no: 59
old   1: insert into worked_upon values(&worker_id,&lot_no)
new   1: insert into worked_upon values(81601631,59)

1 row created.

SQL> commit;

Commit complete.

SQL>
```



```
SQL> insert into worked_upon values(&worker_id,&lot_no);
Enter value for worker_id: 61239042
Enter value for lot_no: 56
old   1: insert into worked_upon values(&worker_id,&lot_no)
new   1: insert into worked_upon values(61239042,56)

1 row created.

SQL> commit;

Commit complete.

SQL>
```

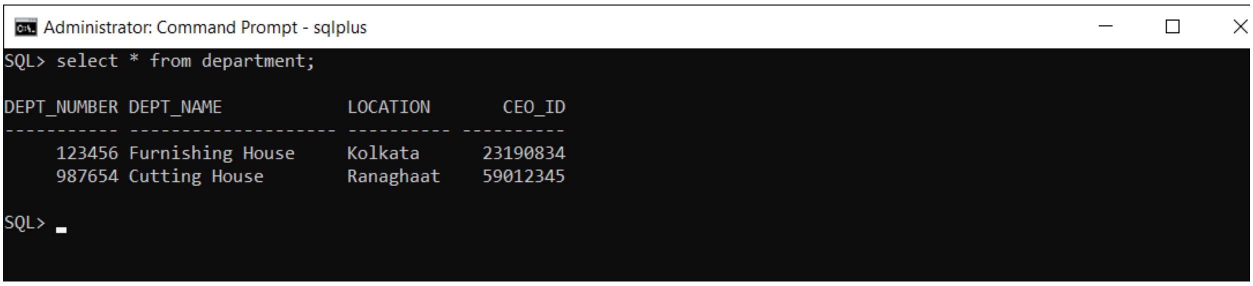
• Displaying entered rows in tables:

1.) ENGINEER-



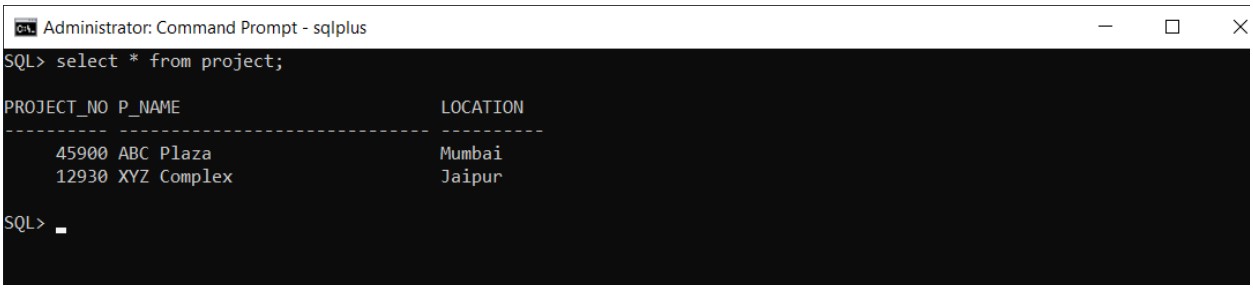
```
SQL> select * from engineer;

ID_NUMBER  NAME            GENDER   ADDRESS        SALARY    DEPT_NO     CEO_ID
---------- --------------- -------- ---------- ---------- ---------- ----------
81267439   John Doe        Male     Chennai        450000     123456   23190834
31046211   Mark Willinson  Male     Kochi          345000     987654   59012345

SQL>
```

2.) DEPARTMENT-



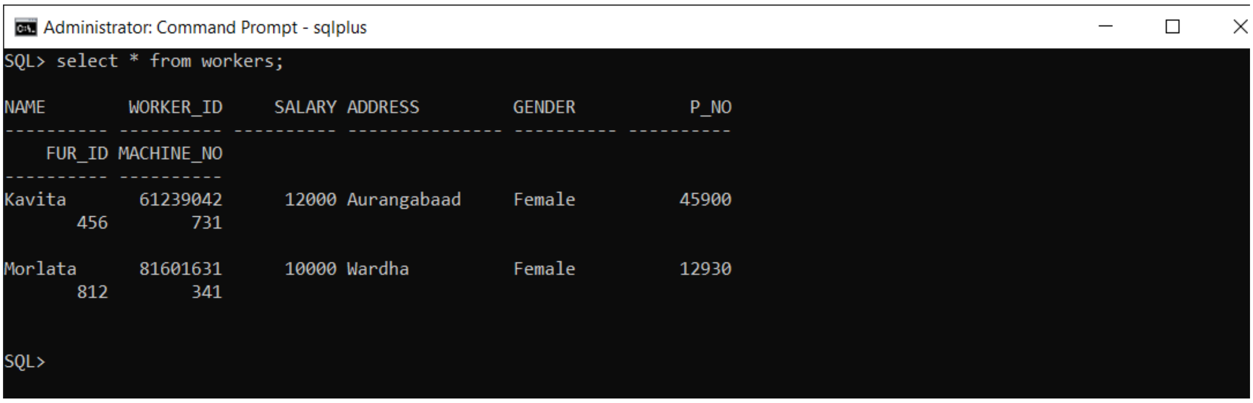
```
SQL> select * from department;

DEPT_NUMBER DEPT_NAME            LOCATION       CEO_ID
----------- -------------------- ---------- ----------
     123456 Furnishing House     Kolkata      23190834
     987654 Cutting House        Ranaghaat    59012345

SQL>
```

3.) Project-



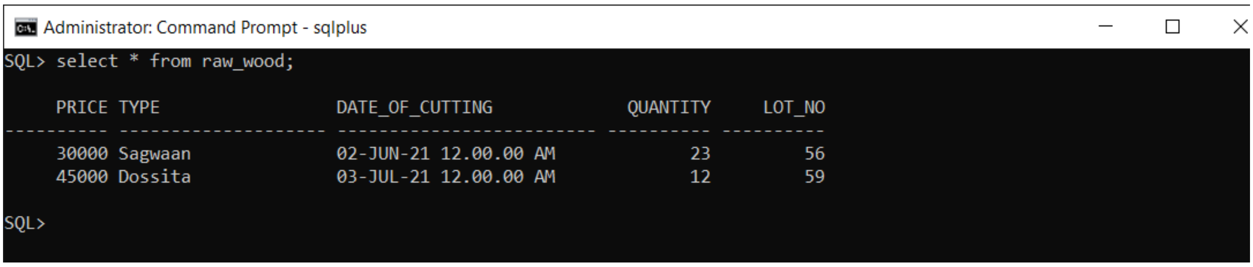
```
SQL> select * from project;

PROJECT_NO P_NAME                         LOCATION
---------- ------------------------------ ----------
     45900 ABC Plaza                      Mumbai
     12930 XYZ Complex                    Jaipur

SQL>
```

4.) Workers-



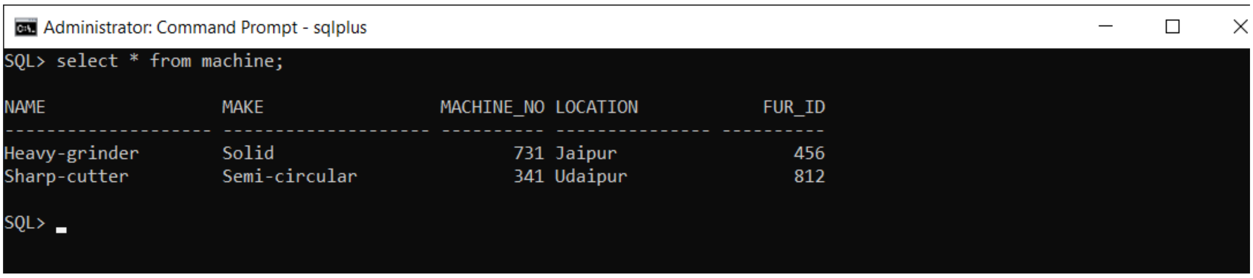
```
SQL> select * from workers;

NAME        WORKER_ID     SALARY ADDRESS        GENDER          P_NO
---------- ---------- ---------- -------------- ---------- ----------
    FUR_ID MACHINE_NO
---------- ----------
Kavita       61239042      12000 Aurangabaad    Female         45900
       456        731

Morlata      81601631      10000 Wardha         Female         12930
       812        341


SQL>
```

5.) Raw_wood-



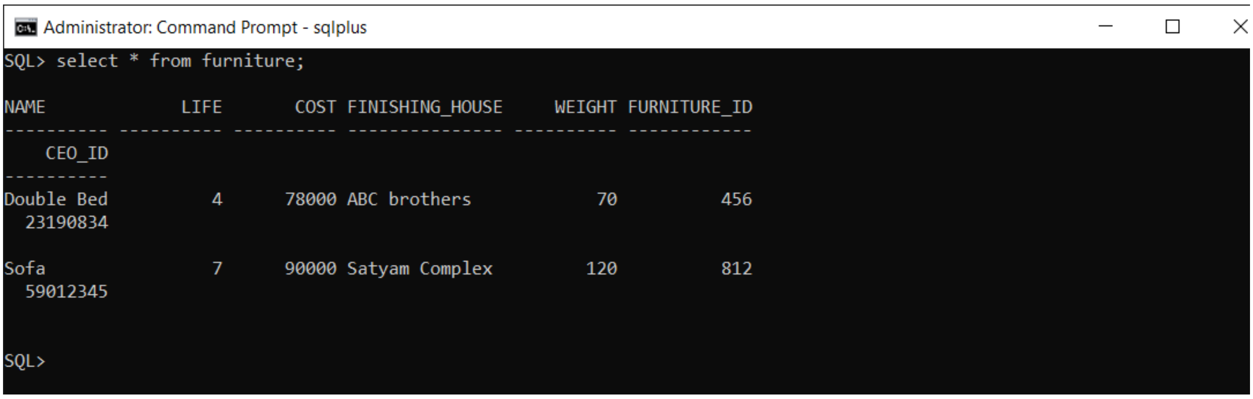
```
SQL> select * from raw_wood;

     PRICE TYPE                 DATE_OF_CUTTING             QUANTITY     LOT_NO
---------- -------------------- ------------------------- ---------- ----------
     30000 Sagwaan              02-JUN-21 12.00.00 AM             23         56
     45000 Dossita              03-JUL-21 12.00.00 AM             12         59

SQL>
```

6.) Machine-



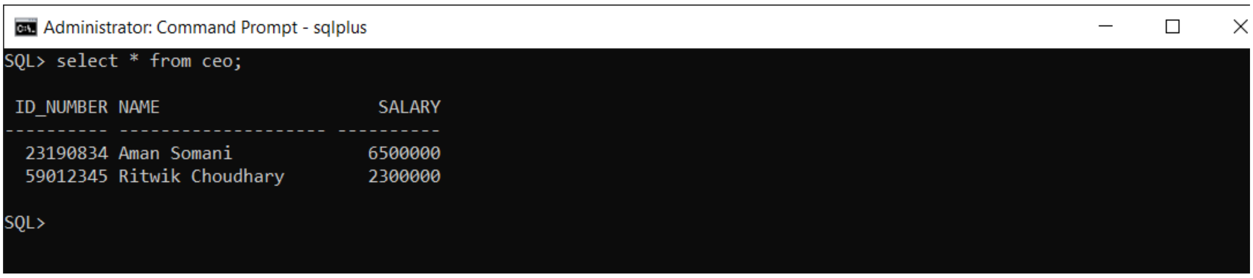
```
SQL> select * from machine;

NAME                 MAKE                 MACHINE_NO LOCATION           FUR_ID
-------------------- -------------------- ---------- -------------- ----------
Heavy-grinder        Solid                       731 Jaipur                456
Sharp-cutter         Semi-circular               341 Udaipur               812

SQL> _
```

7.) Furniture-



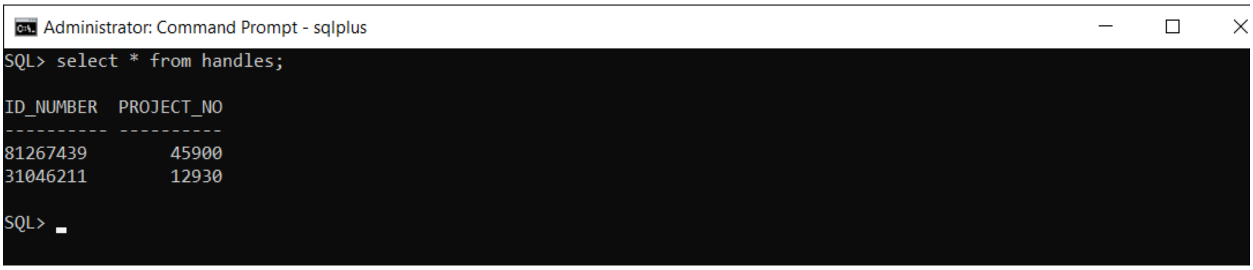
```
SQL> select * from furniture;

NAME             LIFE       COST FINISHING_HOUSE     WEIGHT FURNITURE_ID
---------- ---------- ---------- --------------- ---------- ------------
    CEO_ID
----------
Double Bed          4      78000 ABC brothers            70          456
  23190834

Sofa                7      90000 Satyam Complex         120          812
  59012345


SQL>
```

8.) Ceo-



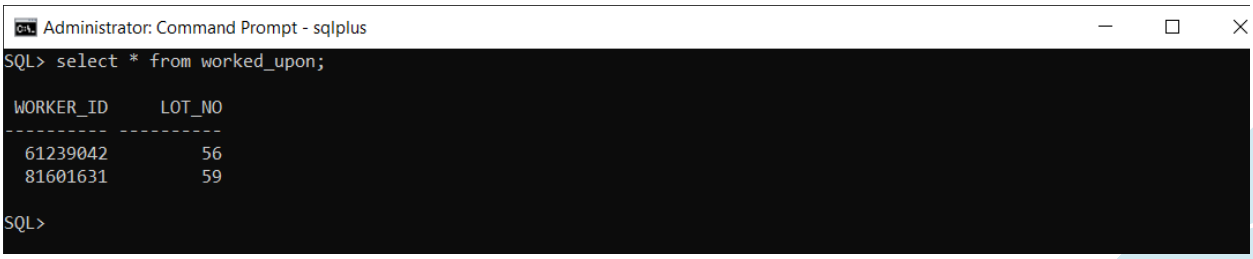
```
SQL> select * from ceo;

 ID_NUMBER NAME                          SALARY
---------- -------------------- ----------
  23190834 Aman Somani                  6500000
  59012345 Ritwik Choudhary             2300000

SQL>
```

9.) Handles-



```
SQL> select * from handles;

ID_NUMBER  PROJECT_NO
---------- ----------
81267439        45900
31046211        12930

SQL> _
```

10.) Worked_upon-

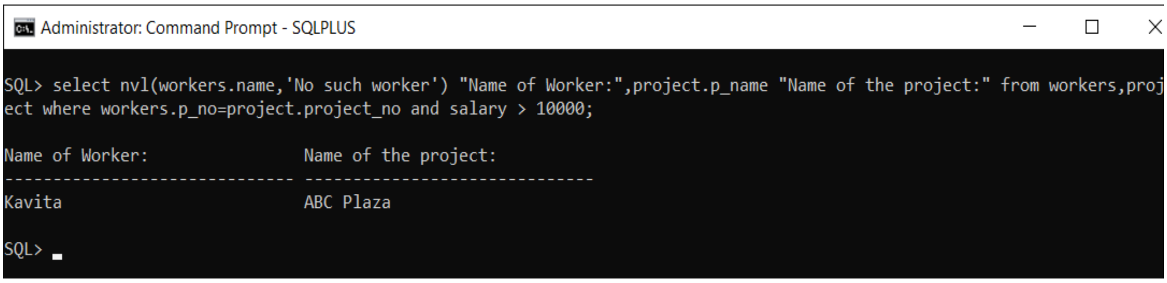


```
SQL> select * from worked_upon;

 WORKER_ID     LOT_NO
---------- ----------
  61239042         56
  81601631         59

SQL>
```

- ## SQL Statements:

1) Nvl-

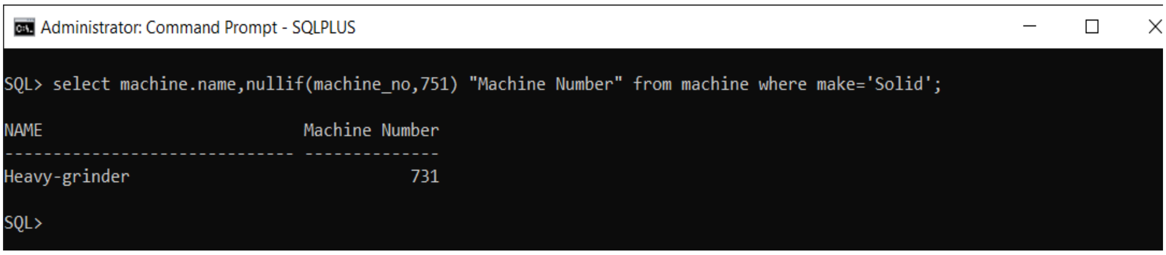
Display name of workers and project name on which the are working and getting salary of above 10000 and if not available display ‘No such worker’.



```
SQL> select nvl(workers.name,'No such worker') "Name of Worker:",project.p_name "Name of the project:" from workers,proj
ect where workers.p_no=project.project_no and salary > 10000;

Name of Worker:                Name of the project:
------------------------------ ------------------------------
Kavita                         ABC Plaza

SQL> _
```

2) Nullif-

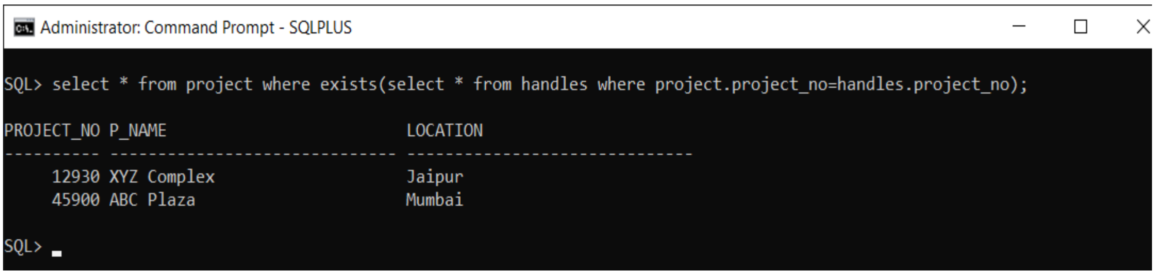
Display name of the machine and machine number whose make is solid and display null if the entered machine number and machine number is same.



```
SQL> select machine.name,nullif(machine_no,751) "Machine Number" from machine where make='Solid';

NAME                           Machine Number
------------------------------ --------------
Heavy-grinder                             731

SQL>
```

3) Correlated-

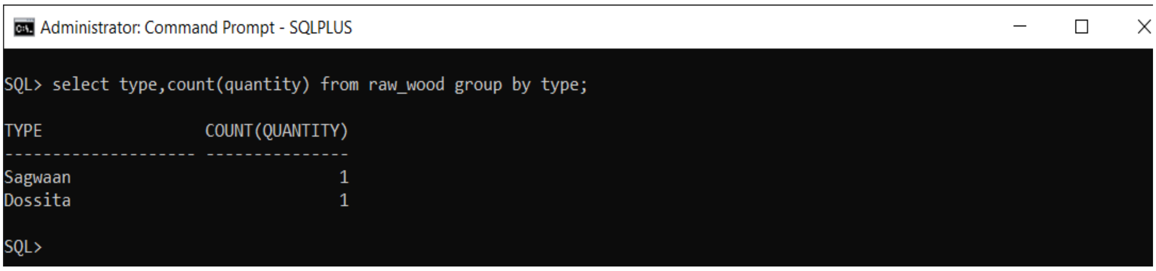
Display project number,name and location of all the projects.



```
SQL> select * from project where exists(select * from handles where project.project_no=handles.project_no);

PROJECT_NO P_NAME                         LOCATION
---------- ------------------------------ ------------------------------
     12930 XYZ Complex                    Jaipur
     45900 ABC Plaza                      Mumbai

SQL> _
```

4) Group by-

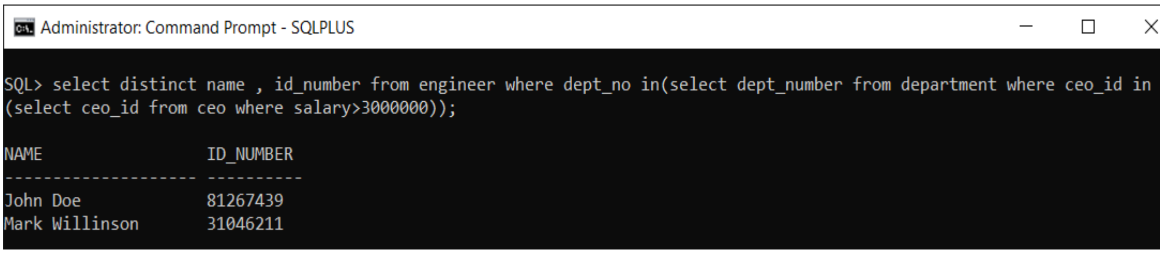
Count the quantity of raw woods based on their type.



```
SQL> select type,count(quantity) from raw_wood group by type;

TYPE                 COUNT(QUANTITY)
-------------------- ---------------
Sagwaan                            1
Dossita                            1

SQL>
```

4) Un-correlated-

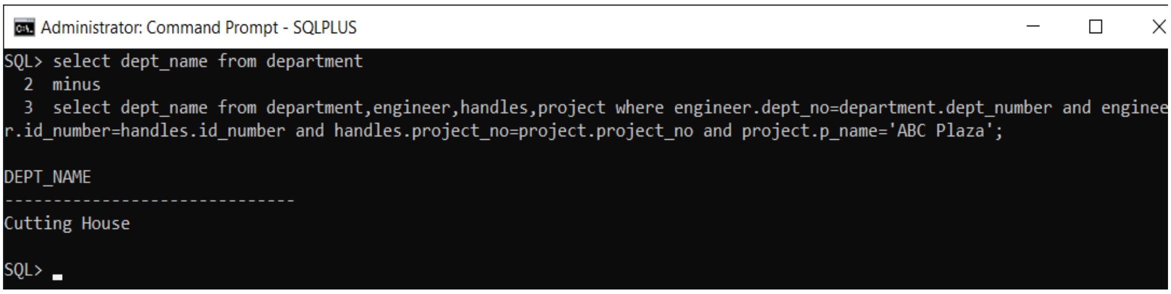
Count the quantity of raw woods based on their type.



```
SQL> select distinct name , id_number from engineer where dept_no in(select dept_number from department where ceo_id in
(select ceo_id from ceo where salary>3000000));

NAME                 ID_NUMBER
-------------------- ----------
John Doe             81267439
Mark Willinson       31046211
```

5) Set operation-

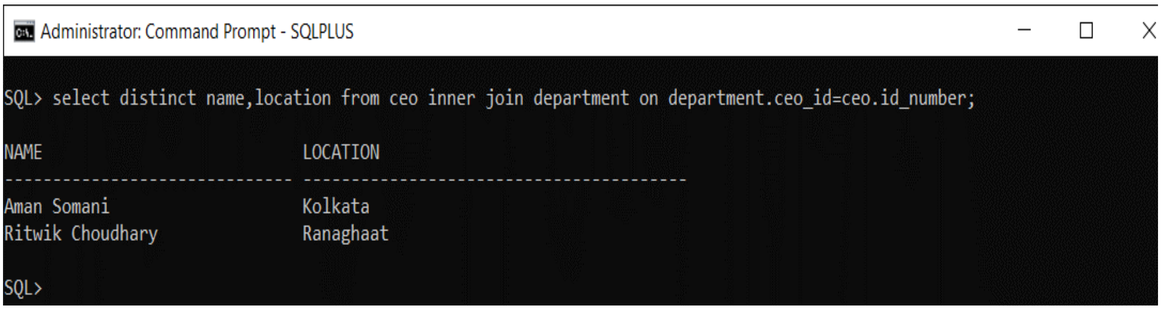
Display name of all the departments which are not handling the project named 'ABC Plaza'.



```
SQL> select dept_name from department
  2  minus
  3  select dept_name from department,engineer,handles,project where engineer.dept_no=department.dept_number and enginee
r.id_number=handles.id_number and handles.project_no=project.project_no and project.p_name='ABC Plaza';

DEPT_NAME
------------------------------
Cutting House

SQL> _
```

6) Inner join-

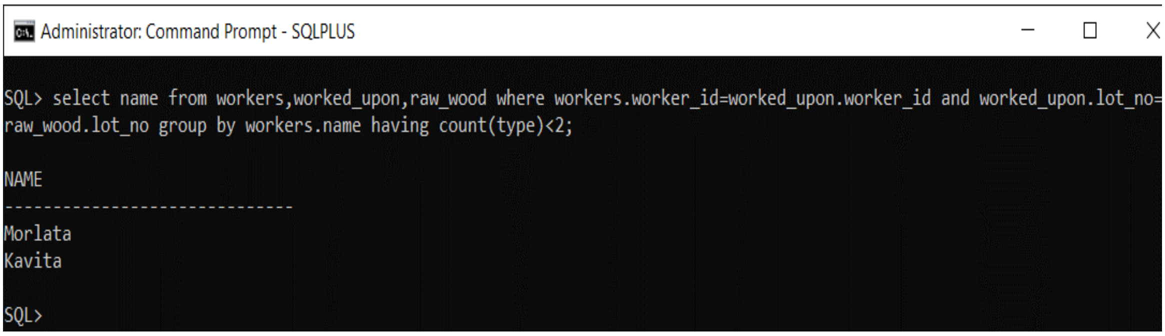
Display the names of ceo and city where their departments are located .



```
SQL> select distinct name,location from ceo inner join department on department.ceo_id=ceo.id_number;

NAME                           LOCATION
------------------------------ ----------------------------------------
Aman Somani                    Kolkata
Ritwik Choudhary               Ranaghaat

SQL>
```

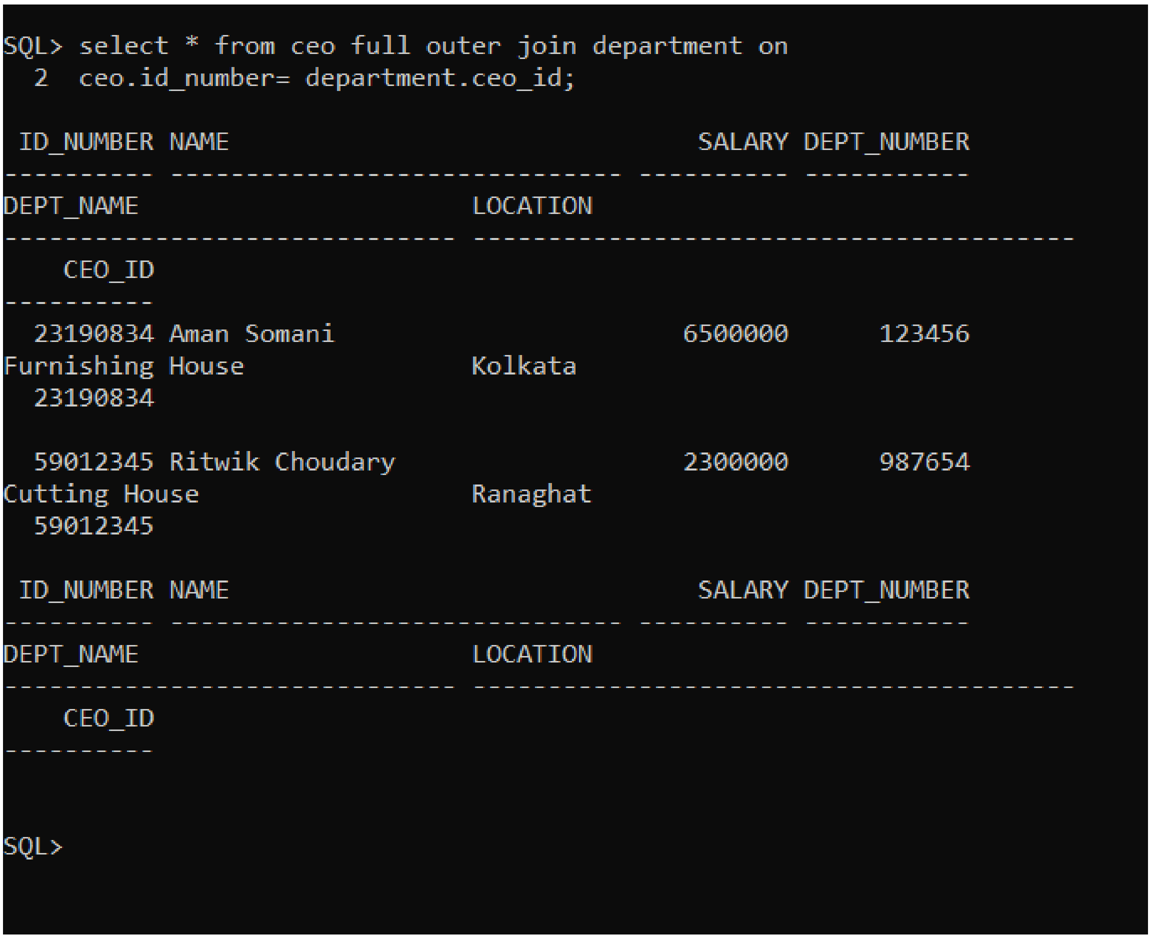
7) Group by having clause-

Display the name of workers who are working on less than two types of raw wood.



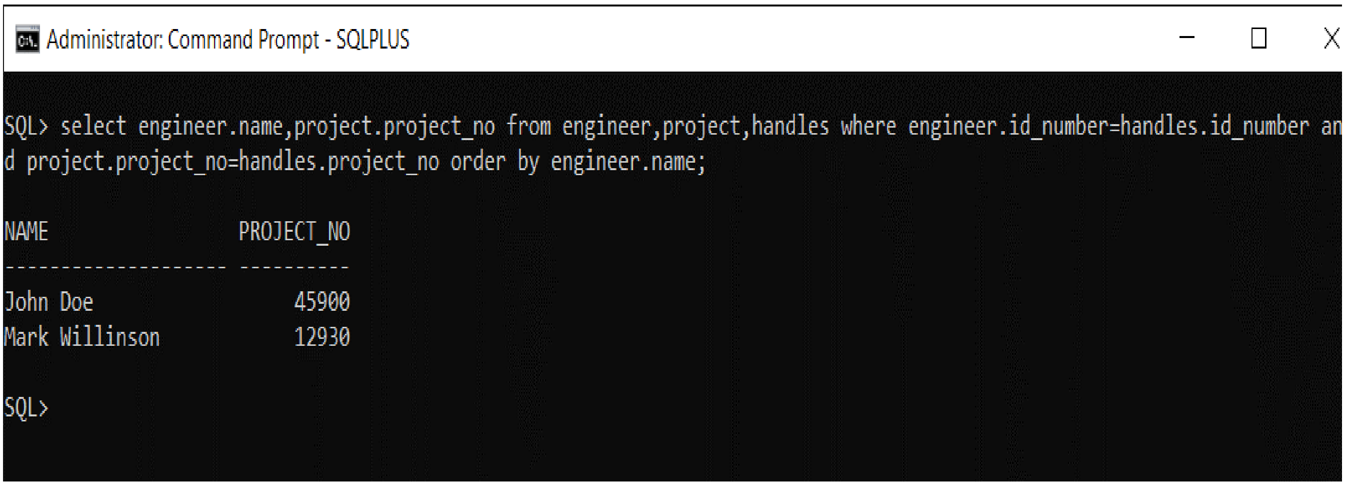
```
SQL> select name from workers,worked_upon,raw_wood where workers.worker_id=worked_upon.worker_id and worked_upon.lot_no=
raw_wood.lot_no group by workers.name having count(type)<2;

NAME
------------------------------
Morlata
Kavita

SQL>
```

8) To display ceo and department details using outer join query –



```
SQL> select * from ceo full outer join department on
  2  ceo.id_number= department.ceo_id;

 ID_NUMBER NAME                                SALARY DEPT_NUMBER
---------- ------------------------------ ---------- -----------
DEPT_NAME                      LOCATION
------------------------------ ----------------------------------------
    CEO_ID
----------
  23190834 Aman Somani                        6500000      123456
Furnishing House               Kolkata
  23190834

  59012345 Ritwik Choudary                    2300000      987654
Cutting House                  Ranaghat
  59012345

 ID_NUMBER NAME                                SALARY DEPT_NUMBER
---------- ------------------------------ ---------- -----------
DEPT_NAME                      LOCATION
------------------------------ ----------------------------------------
    CEO_ID
----------


SQL>
```

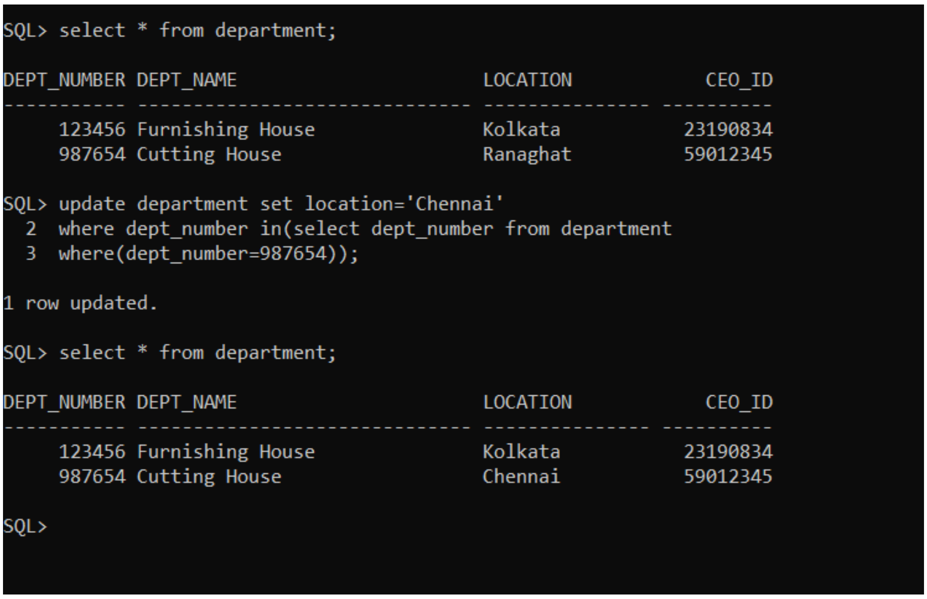
10) Display the name of engineers and the project no. they are handling in the alphabetical order of their name.



```
SQL> select engineer.name,project.project_no from engineer,project,handles where engineer.id_number=handles.id_number an
d project.project_no=handles.project_no order by engineer.name;

NAME                 PROJECT_NO
-------------------- ----------
John Doe                  45900
Mark Willinson            12930

SQL>
```

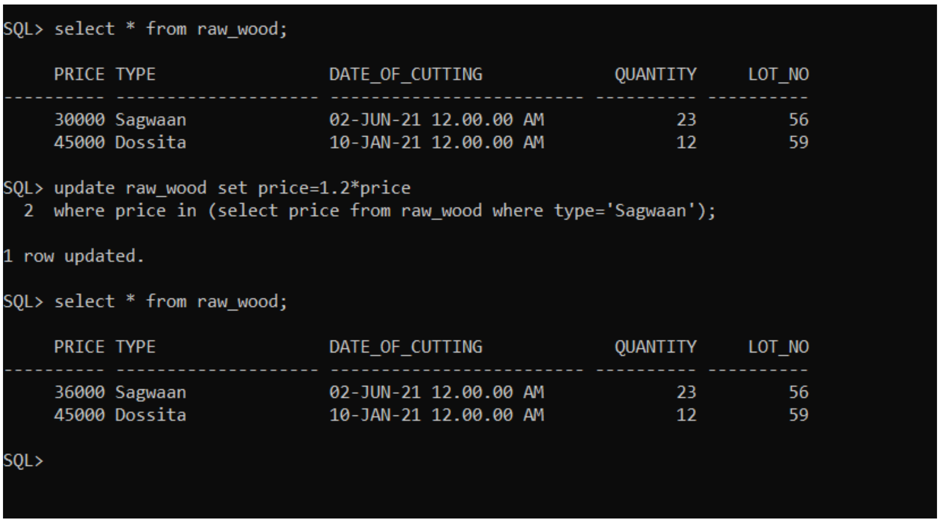
Update queries-

Update location of a depatment.



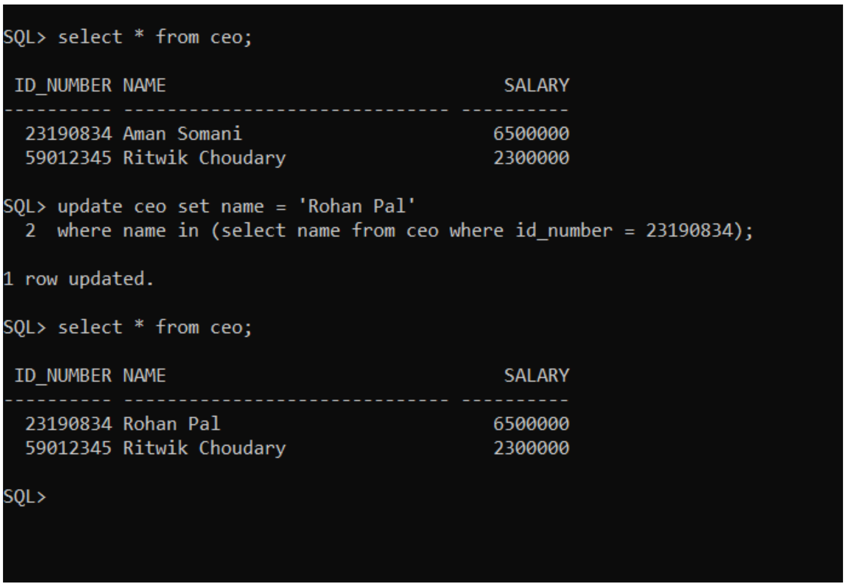
```
SQL> select * from department;

DEPT_NUMBER DEPT_NAME                      LOCATION           CEO_ID
----------- ------------------------------ --------------- ----------
     123456 Furnishing House               Kolkata           23190834
     987654 Cutting House                  Ranaghat          59012345

SQL> update department set location='Chennai'
  2  where dept_number in(select dept_number from department
  3  where(dept_number=987654));

1 row updated.

SQL> select * from department;

DEPT_NUMBER DEPT_NAME                      LOCATION           CEO_ID
----------- ------------------------------ --------------- ----------
     123456 Furnishing House               Kolkata           23190834
     987654 Cutting House                  Chennai           59012345

SQL>
```

Increase price of Sagwaan wood by 20%-



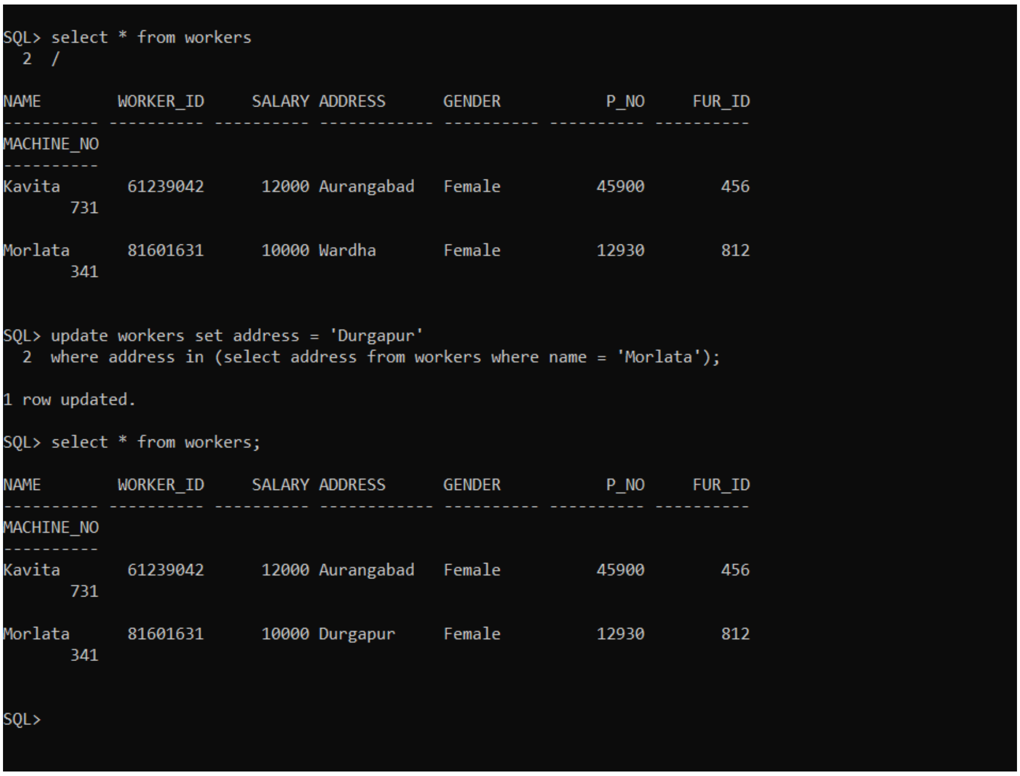
```
SQL> select * from raw_wood;

     PRICE TYPE                 DATE_OF_CUTTING                  QUANTITY     LOT_NO
---------- -------------------- ------------------------- ---------- ----------
     30000 Sagwaan              02-JUN-21 12.00.00 AM             23         56
     45000 Dossita              10-JAN-21 12.00.00 AM             12         59

SQL> update raw_wood set price=1.2*price
  2  where price in (select price from raw_wood where type='Sagwaan');

1 row updated.

SQL> select * from raw_wood;

     PRICE TYPE                 DATE_OF_CUTTING                  QUANTITY     LOT_NO
---------- -------------------- ------------------------- ---------- ----------
     36000 Sagwaan              02-JUN-21 12.00.00 AM             23         56
     45000 Dossita              10-JAN-21 12.00.00 AM             12         59

SQL>
```

Update CEO name-



```
SQL> select * from ceo;

 ID_NUMBER NAME                                SALARY
---------- ------------------------------ ----------
  23190834 Aman Somani                        6500000
  59012345 Ritwik Choudary                    2300000

SQL> update ceo set name = 'Rohan Pal'
  2  where name in (select name from ceo where id_number = 23190834);

1 row updated.

SQL> select * from ceo;

 ID_NUMBER NAME                                SALARY
---------- ------------------------------ ----------
  23190834 Rohan Pal                          6500000
  59012345 Ritwik Choudary                    2300000

SQL>
```

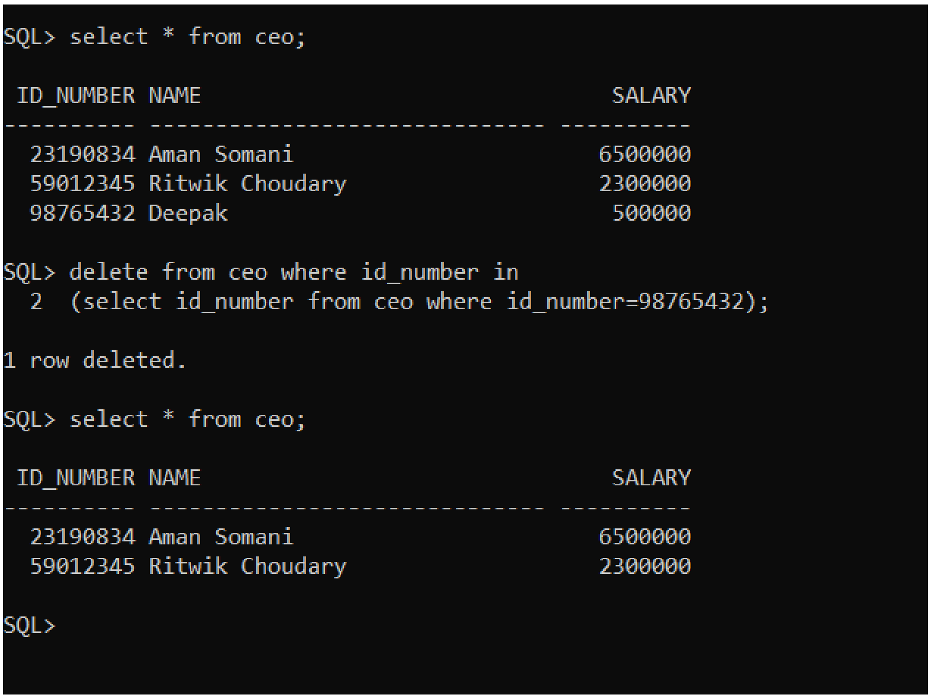
Update Address of a particular worker-



```
SQL> select * from workers
  2  /

NAME         WORKER_ID     SALARY ADDRESS     GENDER           P_NO     FUR_ID
---------- ---------- ---------- ----------- ---------- ---------- ----------
MACHINE_NO
----------
Kavita        61239042      12000 Aurangabad   Female          45900        456
       731

Morlata       81601631      10000 Wardha       Female          12930        812
       341


SQL> update workers set address = 'Durgapur'
  2  where address in (select address from workers where name = 'Morlata');

1 row updated.

SQL> select * from workers;

NAME         WORKER_ID     SALARY ADDRESS     GENDER           P_NO     FUR_ID
---------- ---------- ---------- ----------- ---------- ---------- ----------
MACHINE_NO
----------
Kavita        61239042      12000 Aurangabad   Female          45900        456
       731

Morlata       81601631      10000 Durgapur     Female          12930        812
       341


SQL>
```

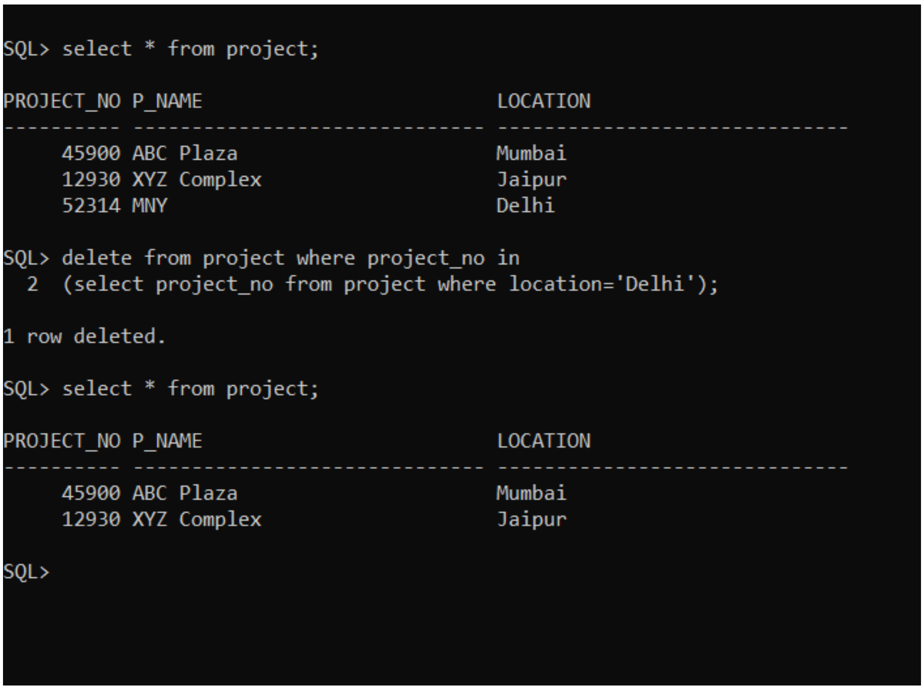
Delete queries-

To Remove CEO-



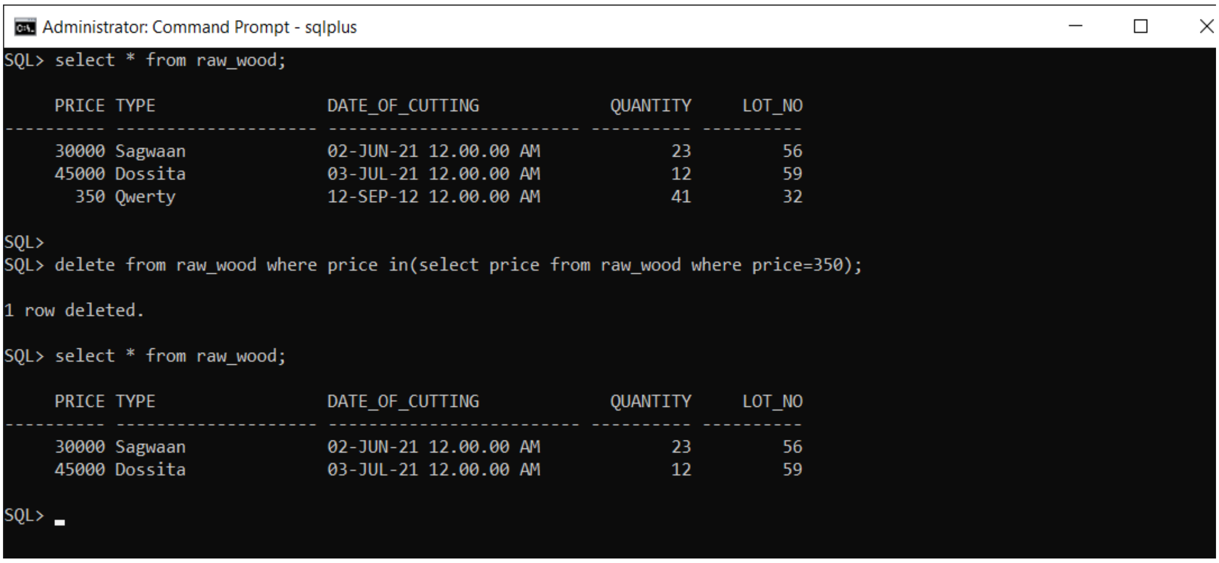
```
SQL> select * from ceo;

 ID_NUMBER NAME                                SALARY
---------- ------------------------------ ----------
  23190834 Aman Somani                       6500000
  59012345 Ritwik Choudary                   2300000
  98765432 Deepak                             500000

SQL> delete from ceo where id_number in
  2  (select id_number from ceo where id_number=98765432);

1 row deleted.

SQL> select * from ceo;

 ID_NUMBER NAME                                SALARY
---------- ------------------------------ ----------
  23190834 Aman Somani                       6500000
  59012345 Ritwik Choudary                   2300000

SQL>
```

To remove project where the location is in Delhi-



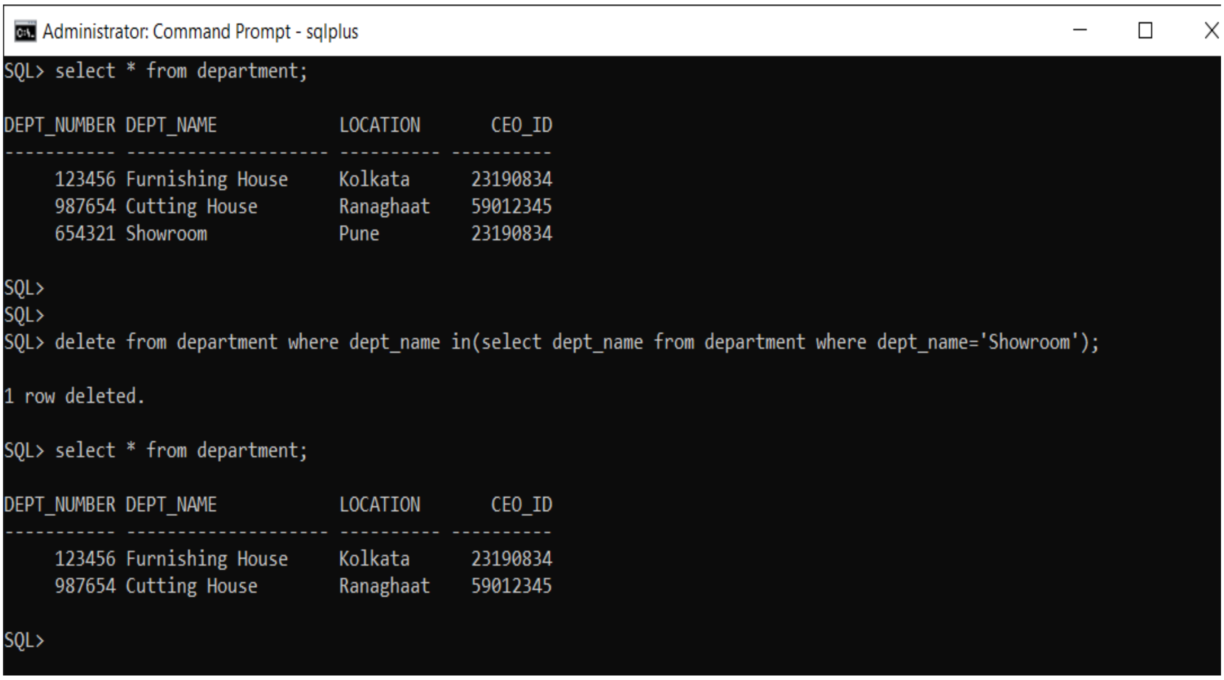
```
SQL> select * from project;

PROJECT_NO P_NAME                         LOCATION
---------- ------------------------------ ------------------------------
     45900 ABC Plaza                      Mumbai
     12930 XYZ Complex                    Jaipur
     52314 MNY                            Delhi

SQL> delete from project where project_no in
  2  (select project_no from project where location='Delhi');

1 row deleted.

SQL> select * from project;

PROJECT_NO P_NAME                         LOCATION
---------- ------------------------------ ------------------------------
     45900 ABC Plaza                      Mumbai
     12930 XYZ Complex                    Jaipur

SQL>
```

To remove raw wood in the basis of pricing.



```
SQL> select * from raw_wood;

     PRICE TYPE                 DATE_OF_CUTTING            QUANTITY     LOT_NO
---------- -------------------- ------------------------- ---------- ----------
     30000 Sagwaan              02-JUN-21 12.00.00 AM             23         56
     45000 Dossita              03-JUL-21 12.00.00 AM             12         59
       350 Qwerty               12-SEP-12 12.00.00 AM             41         32

SQL>
SQL> delete from raw_wood where price in(select price from raw_wood where price=350);

1 row deleted.

SQL> select * from raw_wood;

     PRICE TYPE                 DATE_OF_CUTTING            QUANTITY     LOT_NO
---------- -------------------- ------------------------- ---------- ----------
     30000 Sagwaan              02-JUN-21 12.00.00 AM             23         56
     45000 Dossita              03-JUL-21 12.00.00 AM             12         59

SQL>
```

To remove department in the basis of department name-

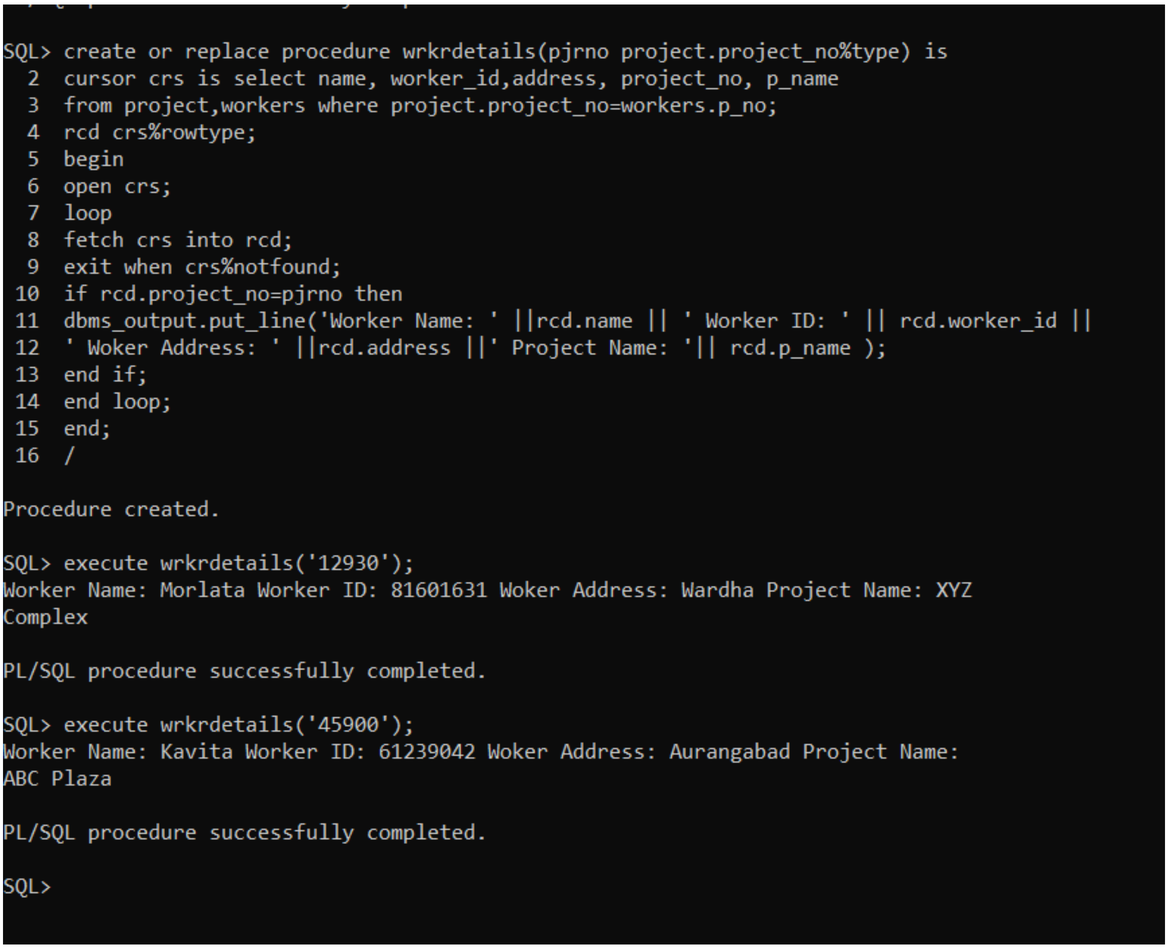


```
SQL> select * from department;

DEPT_NUMBER DEPT_NAME            LOCATION       CEO_ID
----------- -------------------- ---------- ----------
     123456 Furnishing House     Kolkata      23190834
     987654 Cutting House        Ranaghaat    59012345
     654321 Showroom             Pune         23190834

SQL>
SQL>
SQL> delete from department where dept_name in(select dept_name from department where dept_name='Showroom');

1 row deleted.

SQL> select * from department;

DEPT_NUMBER DEPT_NAME            LOCATION       CEO_ID
----------- -------------------- ---------- ----------
     123456 Furnishing House     Kolkata      23190834
     987654 Cutting House        Ranaghaat    59012345

SQL>
```

• Procedures and functions using PL/SQL-

1) Display worker name, id, address and Project Name of a given Project Number-

```
SQL>create or replace procedure
wrkrdetails(pjrno project.project_no%type) is2
cursor crs is select name, worker_id,address,
project_no, p_name
3 from project,workers where
project.project_no=workers.p_no;4  rcd
crs%rowtype;
5  begin
6  open crs;
7  loop
8  fetch crs into rcd;
9  exit when crs%notfound;
10  if rcd.project_no=pjrno then
11  dbms_output.put_line('Worker
Name: ' ||rcd.name || ' Worker ID: ' ||
rcd.worker_id ||
12  ' Woker Address: ' ||rcd.address ||'
Project Name: '|| rcd.p_name );
 13  end if;
 14  end loop;
 15  end;
 16  /
```

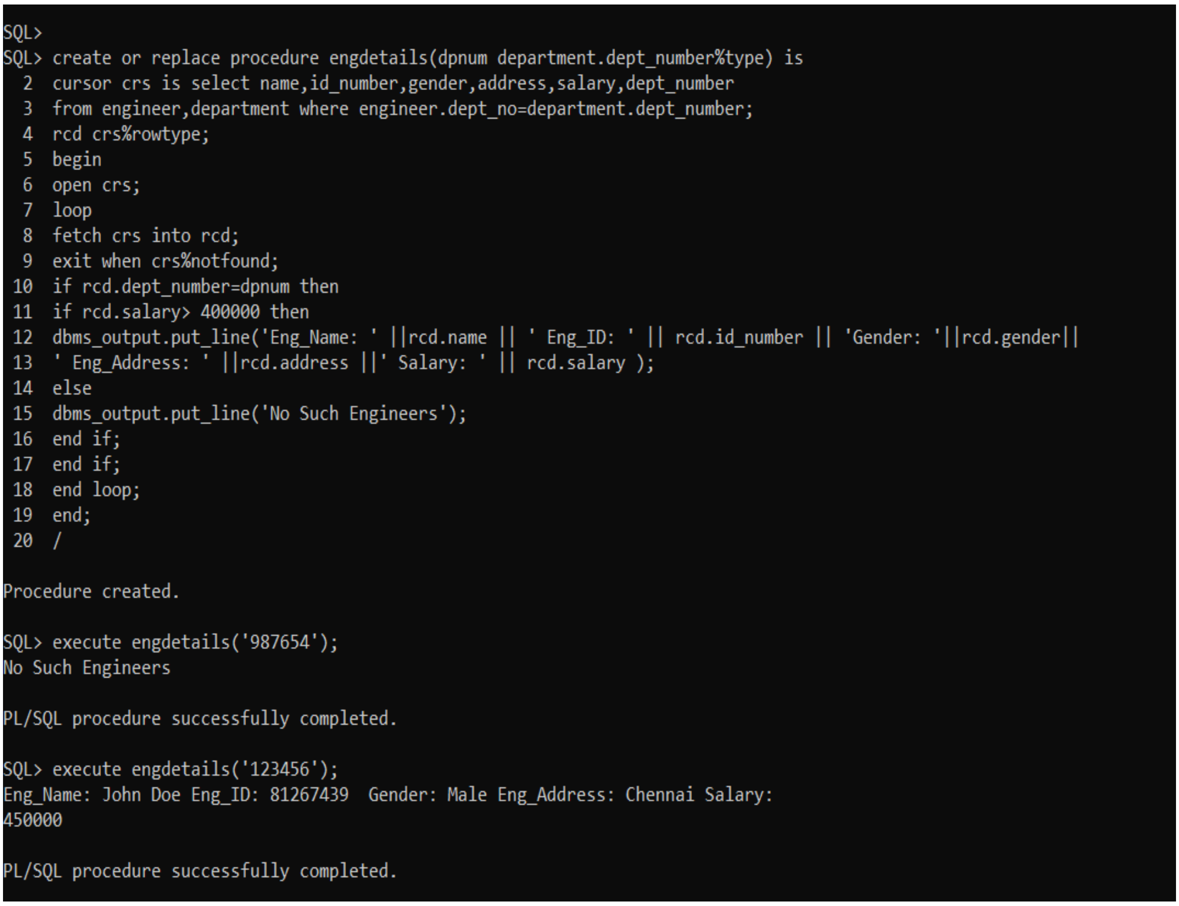


```
SQL> create or replace procedure wrkrdetails(pjrno project.project_no%type) is
cursor crs is select name, worker_id,address, project_no, p_name
from project,workers where project.project_no=workers.p_no;
rcd crs%rowtype;
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.project_no=pjrno then
dbms_output.put_line('Worker Name: ' ||rcd.name || ' Worker ID: ' || rcd.worker_id ||
' Woker Address: ' ||rcd.address ||' Project Name: '|| rcd.p_name );
end if;
end loop;
end;
/

Procedure created.

SQL> execute wrkrdetails('12930');
Worker Name: Morlata Worker ID: 81601631 Woker Address: Wardha Project Name: XYZ
Complex

PL/SQL procedure successfully completed.

SQL> execute wrkrdetails('45900');
Worker Name: Kavita Worker ID: 61239042 Woker Address: Aurangabad Project Name:
ABC Plaza

PL/SQL procedure successfully completed.

SQL>
```

2) Display Engineer details of a given Department Number whose salary is greater than 400000 else show ‘No Such Engineers’.

PL SQL procedure for this particular problem is as follows –

```
SQL> create or replace procedure engdetails(dpnum department.dept_number%type)
is
cursor crs is select name,id_number,gender,address,salary,dept_number
from engineer,department where engineer.dept_no=department.dept_number;
rcd crs%rowtype;
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
 if rcd.dept_number=dpnum then
 if rcd.salary> 400000 then
 dbms_output.put_line('Eng_Name: ' ||rcd.name || ' Eng_ID: ' || rcd.id_number || 'Gender: '||rcd.gender||
 ' Eng_Address: ' ||rcd.address ||' Salary: ' || rcd.salary );
 else
 dbms_output.put_line('No Such Engineers');
 end if;
 end if;
 end loop;
 end;
 /
```

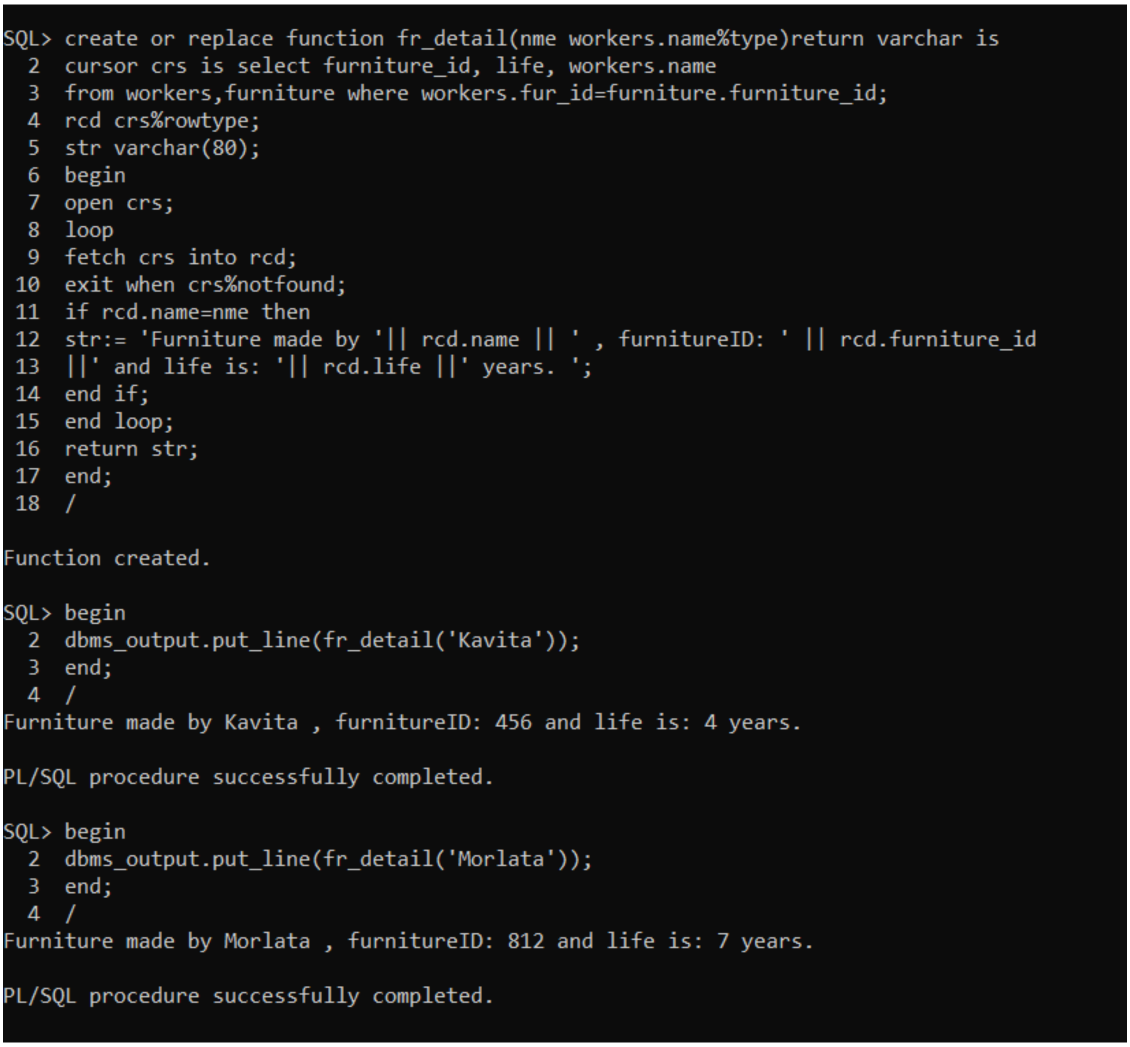


```
SQL>
SQL> create or replace procedure engdetails(dpnum department.dept_number%type) is
cursor crs is select name,id_number,gender,address,salary,dept_number
from engineer,department where engineer.dept_no=department.dept_number;
rcd crs%rowtype;
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.dept_number=dpnum then
if rcd.salary> 400000 then
dbms_output.put_line('Eng_Name: ' ||rcd.name || ' Eng_ID: ' || rcd.id_number || 'Gender: '||rcd.gender||
' Eng_Address: ' ||rcd.address ||' Salary: ' || rcd.salary );
else
dbms_output.put_line('No Such Engineers');
end if;
end if;
end loop;
end;
/

Procedure created.

SQL> execute engdetails('987654');
No Such Engineers

PL/SQL procedure successfully completed.

SQL> execute engdetails('123456');
Eng_Name: John Doe Eng_ID: 81267439  Gender: Male Eng_Address: Chennai Salary:
450000

PL/SQL procedure successfully completed.
```

3) Create a funtion to display Furniture id and life made by a Particular worker-

PL SQL procedure for this particular problem is as follows –

```
SQL> create or replace function fr_detail(nme workers.name%type)return varchar is
  2  cursor crs is select furniture_id, life, workers.name
  3  from workers,furniture where workers.fur_id=furniture.furniture_id;
  4  rcd crs%rowtype;
  5  str varchar(80);
  6  begin
  7  open crs;
  8  loop
  9  fetch crs into rcd;
 10  exit when crs%notfound;
 11  if rcd.name=nme then
 12  str:= 'Furniture made by '|| rcd.name || ' , furnitureID: ' || rcd.furniture_id
 13  ||' and life is: '|| rcd.life ||' years. ';
 14  end if;
 15  end loop;
 16  return str;
 17  end;
 18  /
```



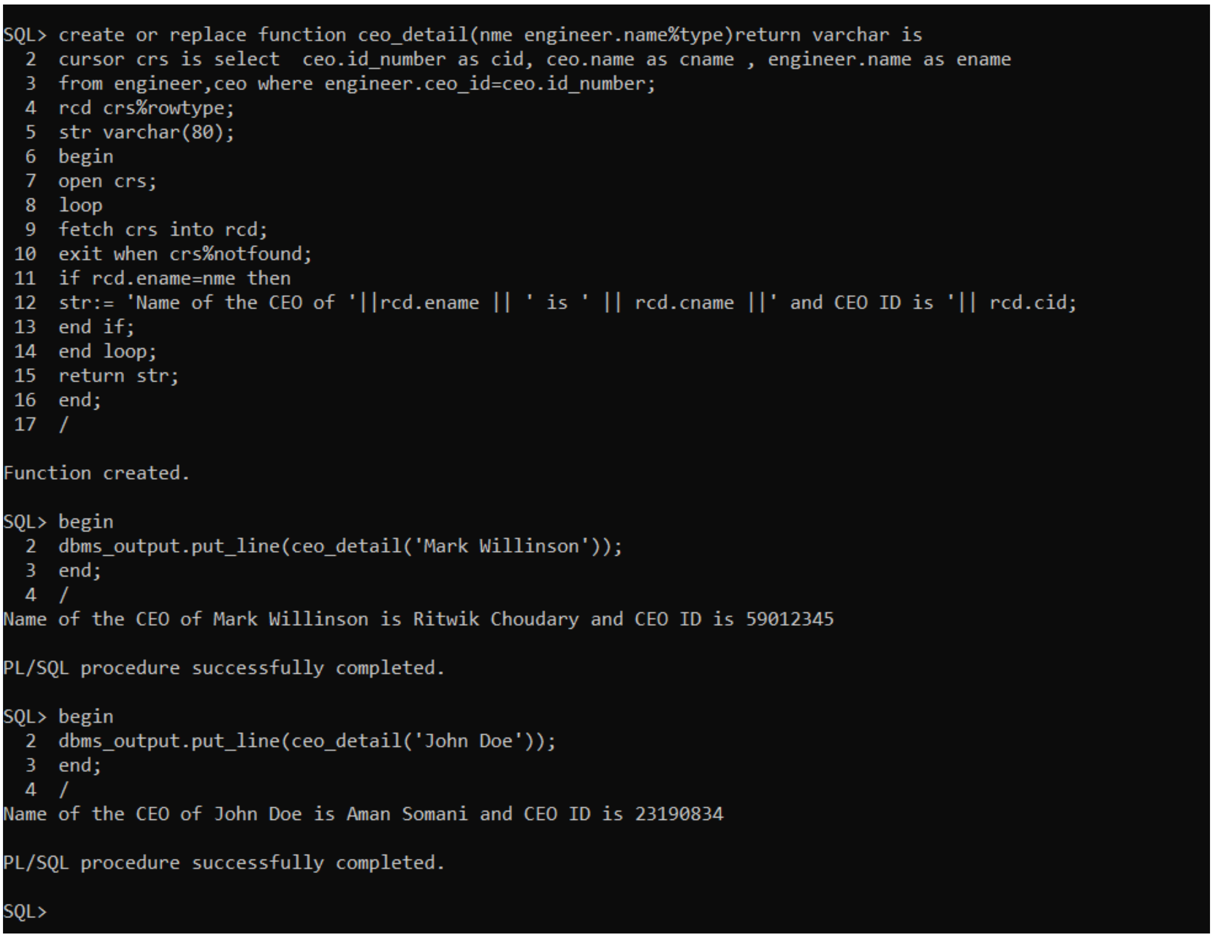
```
SQL> create or replace function fr_detail(nme workers.name%type)return varchar is
  2  cursor crs is select furniture_id, life, workers.name
  3  from workers,furniture where workers.fur_id=furniture.furniture_id;
  4  rcd crs%rowtype;
  5  str varchar(80);
  6  begin
  7  open crs;
  8  loop
  9  fetch crs into rcd;
 10  exit when crs%notfound;
 11  if rcd.name=nme then
 12  str:= 'Furniture made by '|| rcd.name || ' , furnitureID: ' || rcd.furniture_id
 13  ||' and life is: '|| rcd.life ||' years. ';
 14  end if;
 15  end loop;
 16  return str;
 17  end;
 18  /

Function created.

SQL> begin
  2  dbms_output.put_line(fr_detail('Kavita'));
  3  end;
  4  /
Furniture made by Kavita , furnitureID: 456 and life is: 4 years.

PL/SQL procedure successfully completed.

SQL> begin
  2  dbms_output.put_line(fr_detail('Morlata'));
  3  end;
  4  /
Furniture made by Morlata , furnitureID: 812 and life is: 7 years.

PL/SQL procedure successfully completed.
```

4) Create a function to get the CEO details of a particular Engineer Name.

PL SQL procedure for this particular problem is as follows –

```
SQL> create or replace function ceo_detail(nme engineer.name%type)return varchar is
2  cursor crs is select  ceo.id_number as cid, ceo.name as cname , engineer.name as ename
3  from engineer,ceo where engineer.ceo_id=ceo.id_number;
4  rcd crs%rowtype;
  5  str varchar(80);
  6  begin
  7  open crs;
  8  loop
  9  fetch crs into rcd;
 10  exit when crs%notfound;
 11  if rcd.ename=nme then
```

```
str:= 'Name of the CEO of '||rcd.ename || ' is ' || rcd.cname ||' and CEO ID is '|| rcd.cid;
end if;
end loop;
return str;
end;
/
```



```
SQL> create or replace function ceo_detail(nme engineer.name%type)return varchar is
cursor crs is select  ceo.id_number as cid, ceo.name as cname , engineer.name as ename
from engineer,ceo where engineer.ceo_id=ceo.id_number;
rcd crs%rowtype;
str varchar(80);
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.ename=nme then
str:= 'Name of the CEO of '||rcd.ename || ' is ' || rcd.cname ||' and CEO ID is '|| rcd.cid;
end if;
end loop;
return str;
end;
/

Function created.

SQL> begin
dbms_output.put_line(ceo_detail('Mark Willinson'));
end;
/
Name of the CEO of Mark Willinson is Ritwik Choudary and CEO ID is 59012345

PL/SQL procedure successfully completed.

SQL> begin
dbms_output.put_line(ceo_detail('John Doe'));
end;
/
Name of the CEO of John Doe is Aman Somani and CEO ID is 23190834

PL/SQL procedure successfully completed.

SQL>
```