DATABASE MANAGEMENT SYSTEM

Review 1

Prepared by:-

Tanishq :-19BIT0275

Dhruv Sood:- 19BIT0234

Real Estate Management System

Introduction:-

Data Requirements:-

Entities:-

1) ***Client:-*** This Entity stores information about the client. Client_id is going to be the unique identity of every client (Key) and the value stored of client_id is String. Other attributes that are required are Client's name with value stored String, address (so that we know he is an Indian resident) with datatype String, contact details (to contact him) which is a composite attribute consisting of email id, value stored String and phone number value stored String.

2) ***Property _details:-*** It is an entity with many attributes like address, type of property which uses value stored String also it contains a key called property id as there are a lot of properties so each property has been assigned a unique code so that they can be accessed easily. It contains attributes like construction starting date, a total value which uses value stored number.

3**)** ***Owner:-*** Owner is a very important entity for a real estate management system because 99 out of 100 clients ask for the information of the owner. So we added attributes like name, the place he is currently living, work profile which uses value stored String. It also contains attributes like properties owned by him, his personal contact details which uses value stored number. We have added a unique attribute named owner_id so that an owner can be accessed with creating a lot of confusion and also owner_id is the key.

4) ***Residential_ details:-*** As we told you we made this database management system very efficient or can say user friendly we have added a lot of information about the house in depth. We have added attributes like area, rent, sale which uses value stored String. We have added information about

the house in the attribute named BHK which uses value stored number. This entity will also be having a unique id(property_id) through which it can be easily accessed and will be the key.

5) Other _Details:- This is a weak entity type of the parent entity type House Details. This does not have any key, it only consists of features of the House. Property_id is the key here. It has attributes like furnishing value stored String, which side facing value stored String, distance from the nearest hospital, and airport of the value stored number. Does it have a Gym, swimming pool, park, etc?

6) Commercial _property _details:- This is the same as house details just the differences that these properties are for commercial use (for cooperates). For this, we added attributes like area, rent, sale which uses value stored String. We also have added attributes like floor numbers which use value stored number. We have given a unique code to this entity named property_id which is the key. So that all these properties can be accessed easily.

7) Broker:- Role of broker in this system is to be a middleman between client and properties. Broker will have a unique id called broker_id of the value stored String which will also be the key. Broker will have attributes like his/her name of value stored String, Qualifications so that we know how much knowledge he does have, of the value stored String, success rates, which shows how many successful deals have been performed by the broker. Also, his phone number to contact the broker, which is also of the value stored String.

8) Builder:- This is an entity that stores the information of the builder by which the property is established. It has Builder_id as the unique and key. Other attributes consist of the builder group of value stored String which specifies the name of the builder. And the last one is a review, so that client can choose a property with a builder group that has a good past review. The review is of the type number.

Relationship types:-

1) ***A client of (M: N)-(partial participation)*** This specifies the relationship between client and broker and (M: N) specifies that one person can be a client of many brokers and one broker may have many clients.
2) ***Sells (M: N)-(Total Participation)*** It tells the relationship between property and the broker and (M: N) specifies that one broker can sell many properties and one property could be sold by many brokers.
3) ***Owned by (M:1)-(Total Participation)*** It specifies the relationship between the property and the owner. (M:1) states that many houses or buildings can be owned by 1 owner but many owners can't have one or many properties.
4) ***Built by (M:1)-(Total Participation)*** It states that many properties could be built by a single builder but many builders can't build single property.
5) ***Features (1:1)-(Partial Participation)*** It specifies the relationship between house details and other details. It states that one house can have only one other detail.

Functional Requirements:-

1) VIEWER

The system shall allow users to log in if they enter the correct login id and password. Our users can be Client, broker, owner. The users must be able to see details of all properties whether it is house or commercial. Each client shall find their suitable broker and through that the property in which he is interested in. The area of each property shall be visible. The user interface of the website should be interactive and easy to use.

Brokers should be able to view their client details. Owners can also view client details and details of the broker. The client shall be able to view all the features of the house he is interested in.

The system shall display property details along with the price, location wise for the convenience of the client so that he/she is able to choose a property in that area. The details of the builder should be visible to the user. If a client should be able to see the brokerage of all the brokers available so that he can choose the most suitable broker for him/her.

The system should also provide reviews of the property that is open for sale or rent. No information should be hidden from the user and no hidden charges should be applied.

2) Administrator:-

The administrator is in charge of creating the website which is used to access the database. The administrator has all the privileges of the user but has the authority to add and remove data from the database which the user cannot do.

The administrator is accountable for creating user accounts and assigning them the id and password. They generate the fixtures and update them in the database. They should be allowed to enter the new properties available. He should have the authority to enter and modify the property details like the price of the property.

If now the owner doesn't want to sell his property due to personal problems then the administrator is responsible for the removal of that property from the database. A review of every broker must be updated after his successful deal.

Basic Analogy

- View the website with the browser
- Should have an interactive and easy to use user interface
- View all properties
- View all brokers
- View all clients

- View all details of residential property
- View all details of commercial property
- View the other details linked with the residential details
- View the reviews of the property
- View the builder
- View the starting date of construction
- View the brokerage of the broker

View property details location-wise:

I. Price
II. Area
III. Reviews

View all client details:

I. Name
II. Age
III. Contact details
IV. Address

View all Broker details:

I. Name
II. Brokerage
III. Success rate

View all builder details:

I. Name
II. Reviews

Removal of old data:-

1) Remove the property from the database if the owner is not interested in selling the property anymore.
2) Removal of the property if the property has been sold.
3) After the client takes the decision and he changes his mind and doesn't buy the property remove the client from the database.
4) Remove property if the builder ran after taking payments from people and the project is still under construction.

Modification of data:-

1) If the client buys the house, change the owner's details after the paperwork.
2) Modify the price of the property in the price changes.
3) We should update our database regarding the features that new features have been added to this property.
4) If client contact details change then modify them in the database.
5) After 1 successful deal of broker modify the success rate of the broker.

RETRIEVAL OF DATA

1) View Information on every property:
   The client can have an overall view of the property for example:
   a) Total Price
   b) Construction starting date
   c) Owner
2) View Information on Residential Property:

   Now the client can go through the details of the property

a) How many BHK
b) Type of residence, forex- villa, flat, standalone, etc.
c) Area (in sq. ft)
d) Sale price
e) Rent price

3) View information about Commercial property:

Here also a client can go through the details of the property

a) Floor number
b) Area
c) Sale price
d) Rent price

4) View information of owners:
If the client wishes to see the details of the owner
a) Name
b) Phone number
c) Properties Owned
d) Work profile
e) Address
f) Owner id

5) View information of brokers:

All broker information can be viewed by the client so he can choose wisely

a) Name
b) Brokerage
c) Qualification
d) Broker id
e) Phone number
f) Success rate

6) View information of every Client:

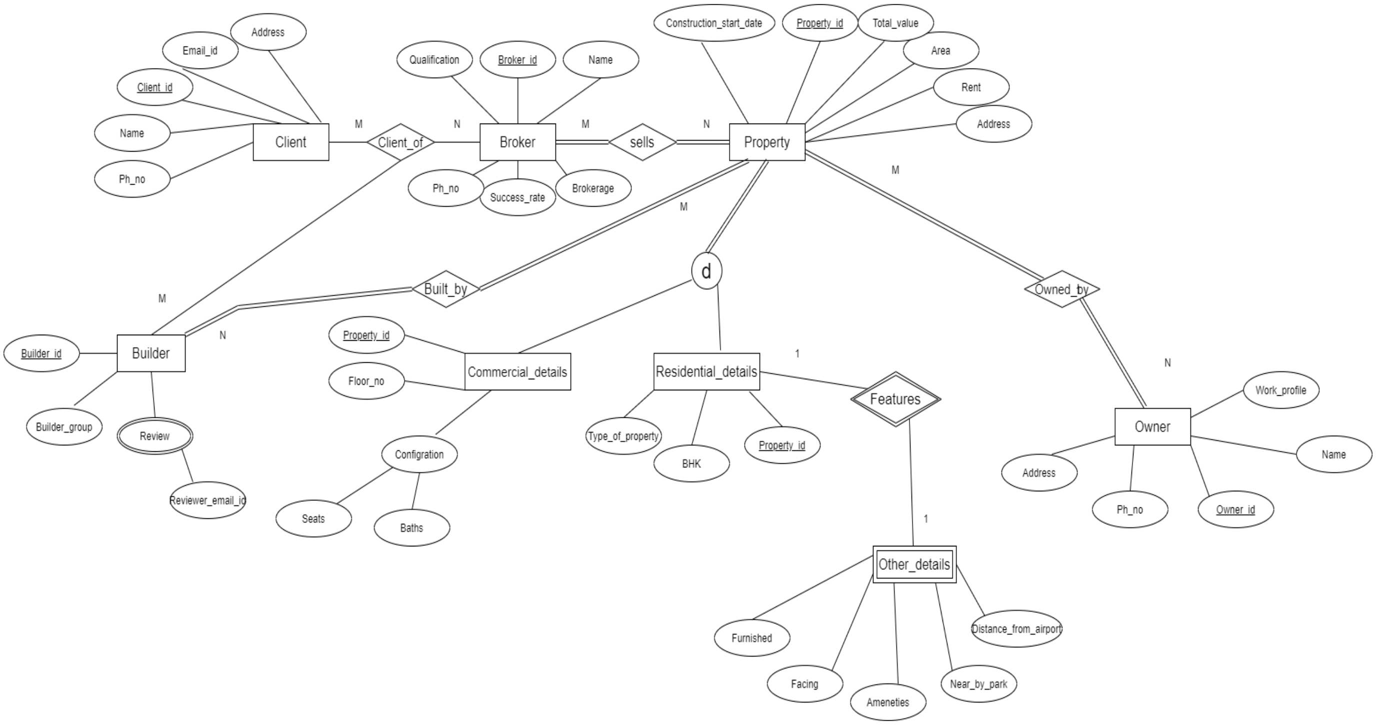
Even the owners can view the details of the clients who are buying the property.

a) Name
b) Address
c) Contact details

Some scenarios of data retrieval are:

1. When the client wants to see the properties according to the area required by the user.
2. If the client wants to view a residential property constructed by a particular builder group.
3. The client wants to know the name and address of the owner of the property.
4. The client wants to select a broker with respect to his qualification.
5. The broker wants to check the client's backgrounds.
6. A client wishing to see house according to its distance from hospitals and airports.
7. The client wants to view the house according to its BHK.
8. The client wants to purchase an office according to the area.
9. The owner wants to check the list of the clients.
10. The client wants to select a broker according to his brokerage.

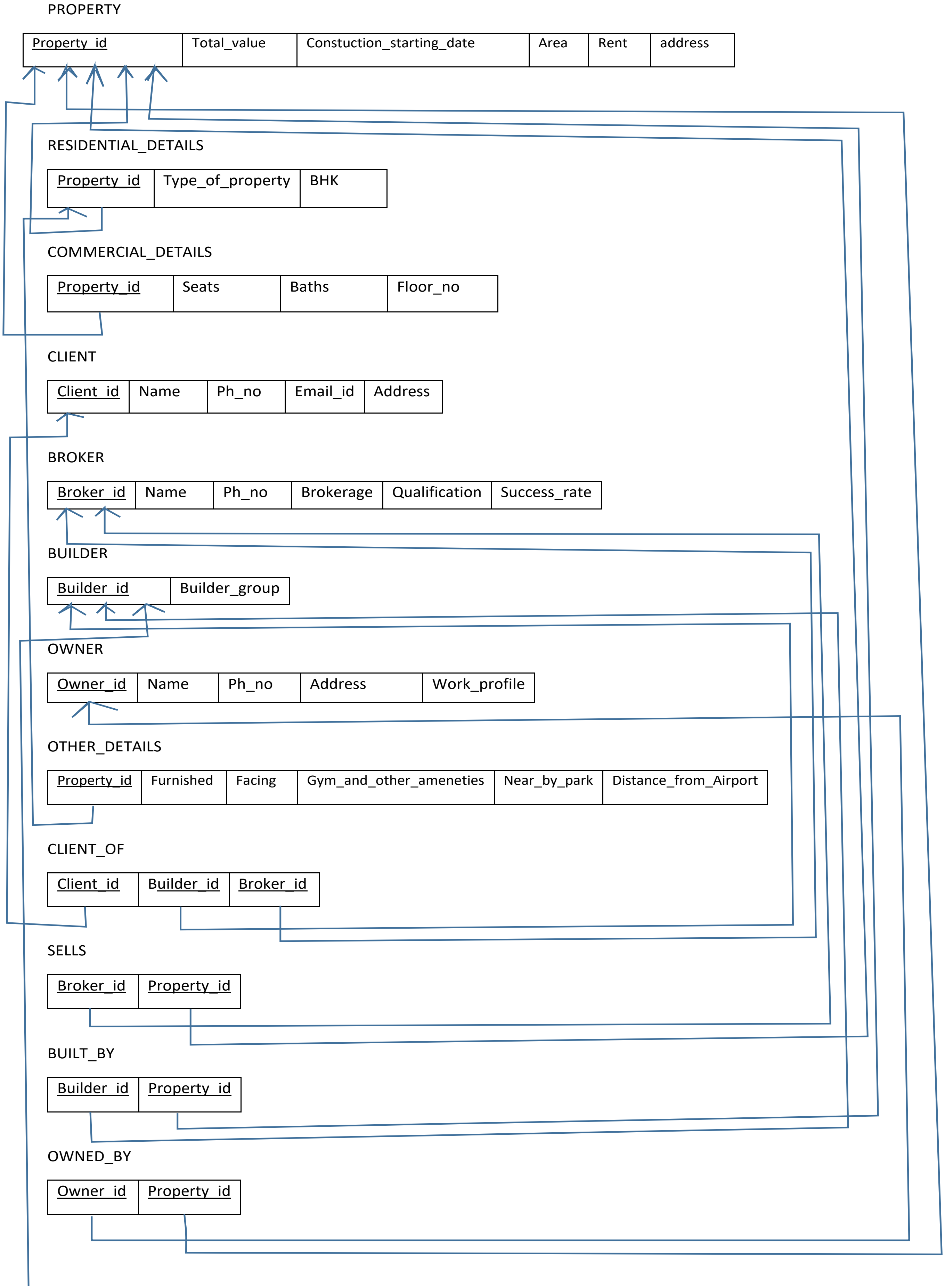
ER Diagram:-


Address
Email_id
Client_id
Name
Ph_no
Client
M
Client_of
N
Qualification
Broker_id
Name
Broker
Ph_no
Success_rate
Brokerage
M
sells
N
Construction_start_date
Property_id
Total_value
Area
Rent
Address
Property
M
M
M
Built_by
N
Builder_id
Builder
Builder_group
Review
Reviewer_email_id
d
Property_id
Floor_no
Commercial_details
Configration
Seats
Baths
Residential_details
Type_of_property
BHK
Property_id
1
Features
1
Other_details
Furnished
Facing
Ameneties
Near_by_park
Distance_from_airport
Owned_by
N
Work_profile
Owner
Name
Address
Ph_no
Owner_id

DATABASE MANAGEMENT SYSTEM

Review 2

Q.4 Relational schema



PROPERTY

Property_id	Total_value	Constuction_starting_date	Area	Rent	address

RESIDENTIAL_DETAILS

Property_id	Type_of_property	BHK

COMMERCIAL_DETAILS

Property_id	Seats	Baths	Floor_no

CLIENT

Client_id	Name	Ph_no	Email_id	Address

BROKER

Broker_id	Name	Ph_no	Brokerage	Qualification	Success_rate

BUILDER

Builder_id	Builder_group

OWNER

Owner_id	Name	Ph_no	Address	Work_profile

OTHER_DETAILS

Property_id	Furnished	Facing	Gym_and_other_ameneties	Near_by_park	Distance_from_Airport

CLIENT_OF

Client_id	Builder_id	Broker_id

SELLS

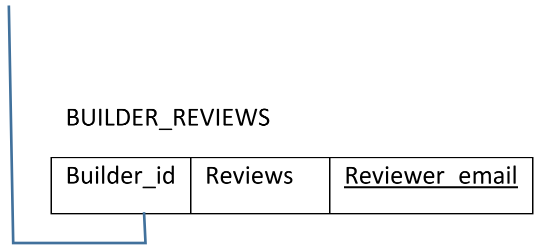
Broker_id	Property_id

BUILT_BY

Builder_id	Property_id

OWNED_BY

Owner_id	Property_id



BUILDER_REVIEWS

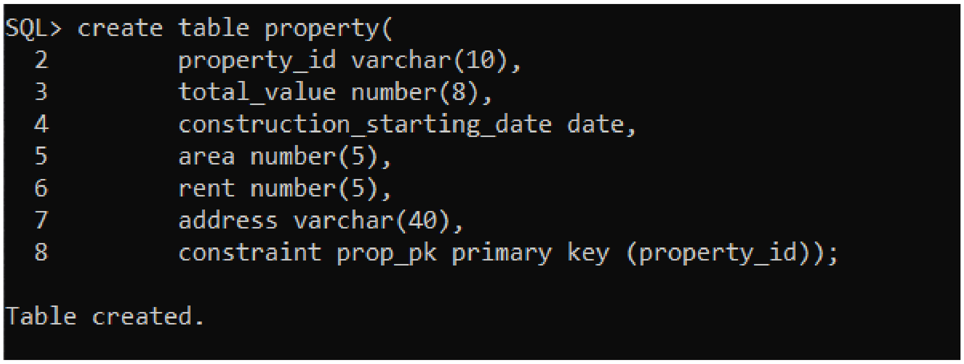
Builder_id	Reviews	Reviewer_email

Ques 5.

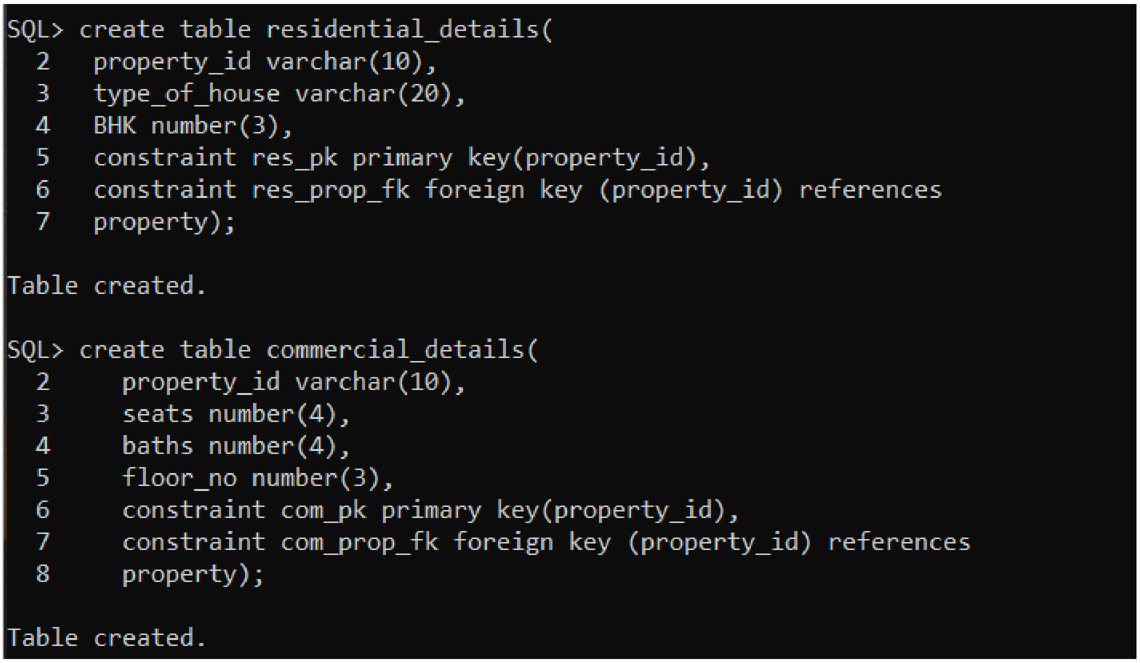
Tables

- Property
- Residential_details
- Commercial_details
- Client
- Broker
- Builder
- Owner
- Other_details
- Client_of
- Sells
- Built_by
- Owned_by
- Builder_reviews

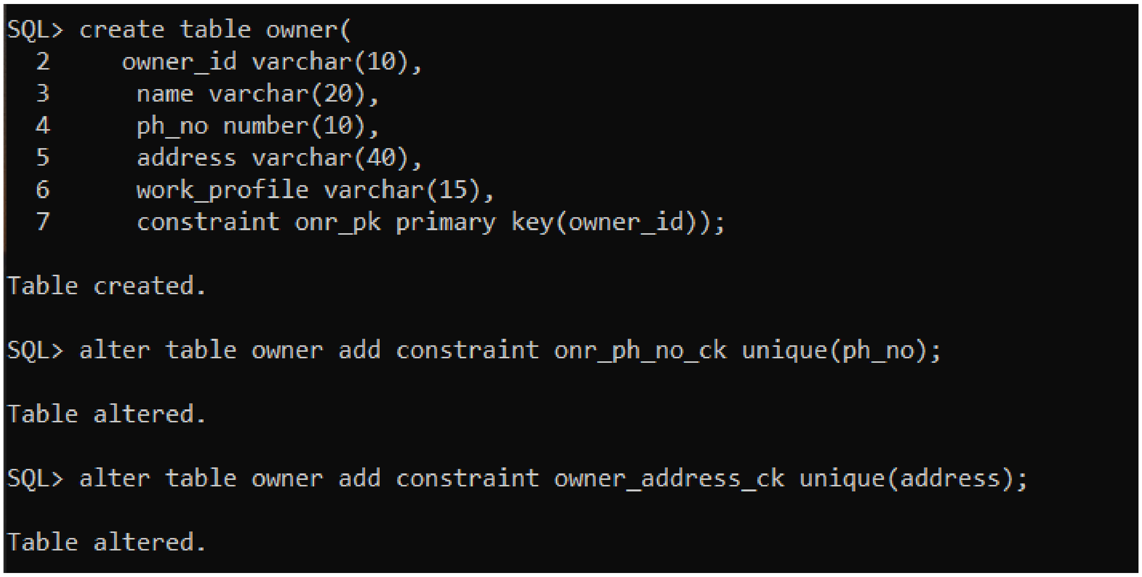
Code to create tables



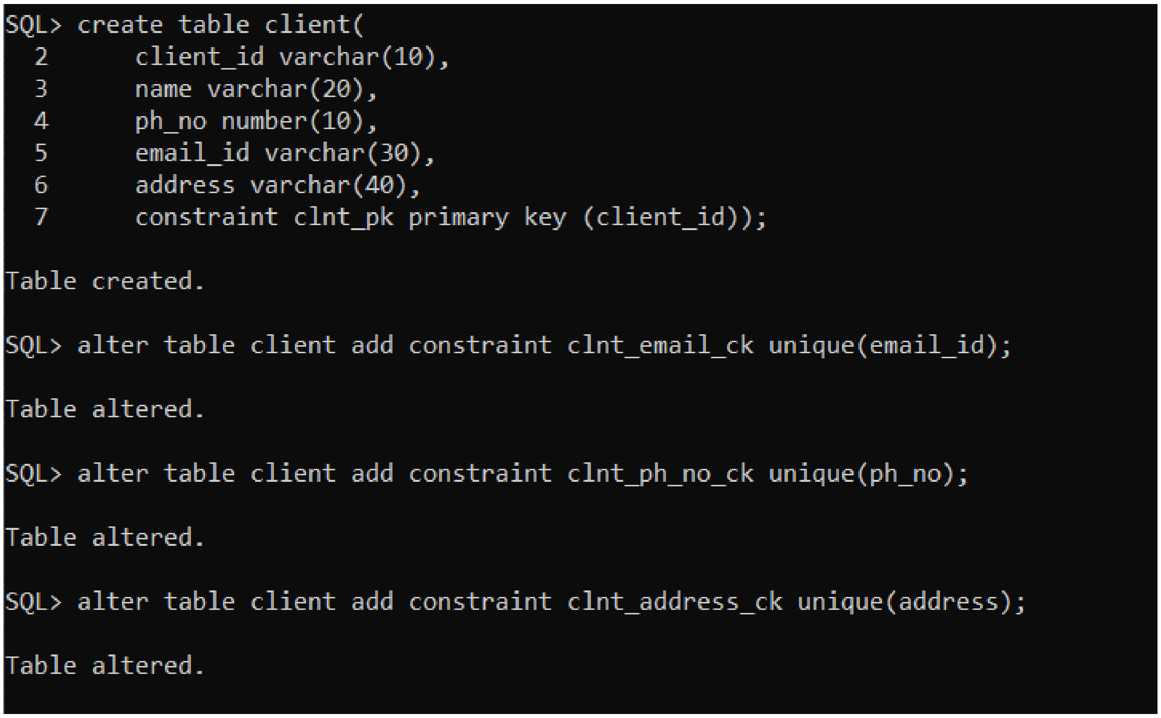
```
SQL> create table property(
        property_id varchar(10),
        total_value number(8),
        construction_starting_date date,
        area number(5),
        rent number(5),
        address varchar(40),
        constraint prop_pk primary key (property_id));

Table created.
```



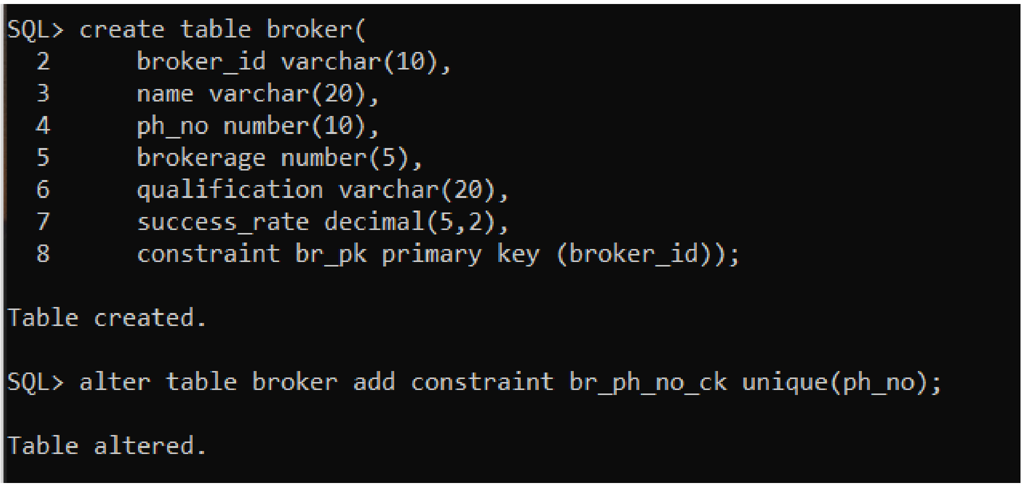
```
SQL> create table residential_details(
   property_id varchar(10),
   type_of_house varchar(20),
   BHK number(3),
   constraint res_pk primary key(property_id),
   constraint res_prop_fk foreign key (property_id) references
   property);

Table created.

SQL> create table commercial_details(
     property_id varchar(10),
     seats number(4),
     baths number(4),
     floor_no number(3),
     constraint com_pk primary key(property_id),
     constraint com_prop_fk foreign key (property_id) references
     property);

Table created.
```



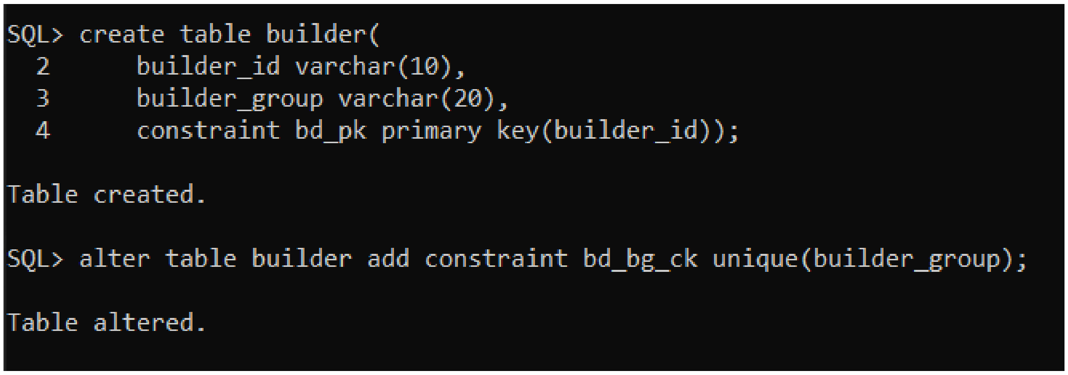
```
SQL> create table owner(
     owner_id varchar(10),
      name varchar(20),
      ph_no number(10),
      address varchar(40),
      work_profile varchar(15),
      constraint onr_pk primary key(owner_id));

Table created.

SQL> alter table owner add constraint onr_ph_no_ck unique(ph_no);

Table altered.

SQL> alter table owner add constraint owner_address_ck unique(address);

Table altered.
```



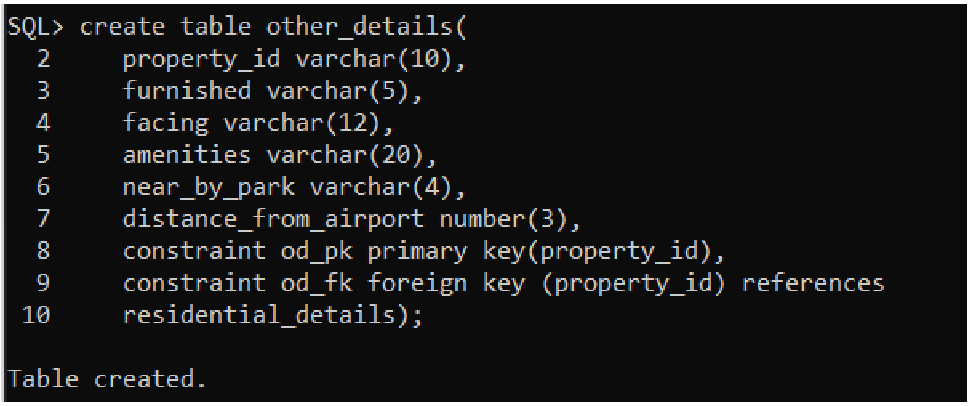
```
SQL> create table client(
        client_id varchar(10),
        name varchar(20),
        ph_no number(10),
        email_id varchar(30),
        address varchar(40),
        constraint clnt_pk primary key (client_id));

Table created.

SQL> alter table client add constraint clnt_email_ck unique(email_id);

Table altered.

SQL> alter table client add constraint clnt_ph_no_ck unique(ph_no);

Table altered.

SQL> alter table client add constraint clnt_address_ck unique(address);

Table altered.
```



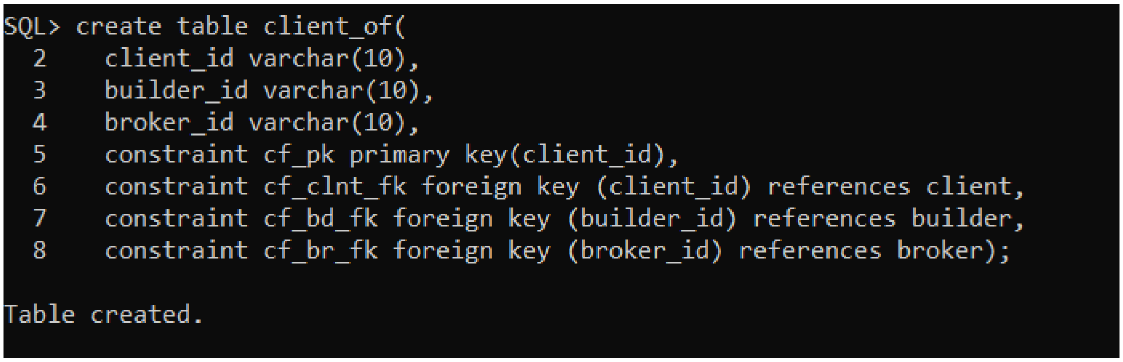
```
SQL> create table broker(
        broker_id varchar(10),
        name varchar(20),
        ph_no number(10),
        brokerage number(5),
        qualification varchar(20),
        success_rate decimal(5,2),
        constraint br_pk primary key (broker_id));

Table created.

SQL> alter table broker add constraint br_ph_no_ck unique(ph_no);

Table altered.
```



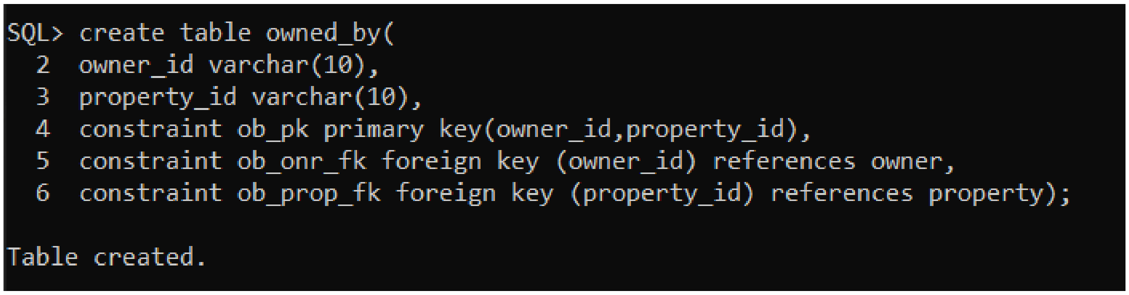
```
SQL> create table builder(
        builder_id varchar(10),
        builder_group varchar(20),
        constraint bd_pk primary key(builder_id));

Table created.

SQL> alter table builder add constraint bd_bg_ck unique(builder_group);

Table altered.
```



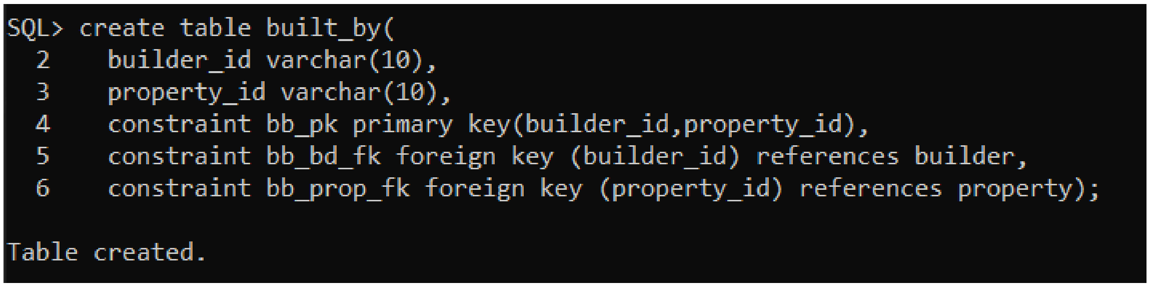
```
SQL> create table other_details(
       property_id varchar(10),
       furnished varchar(5),
       facing varchar(12),
       amenities varchar(20),
       near_by_park varchar(4),
       distance_from_airport number(3),
       constraint od_pk primary key(property_id),
       constraint od_fk foreign key (property_id) references
       residential_details);

Table created.
```



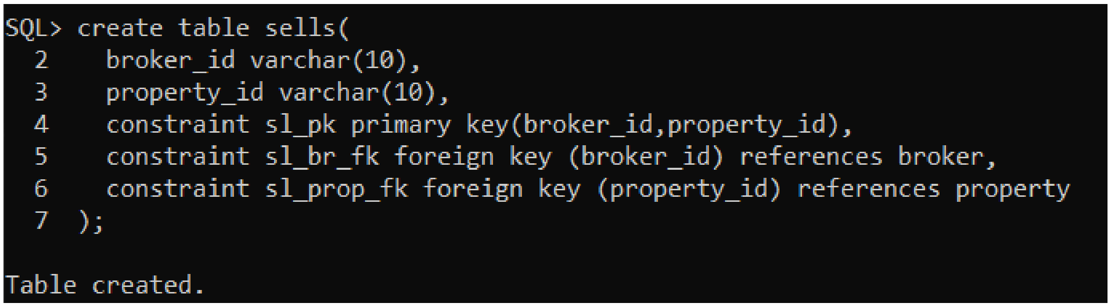
```
SQL> create table client_of(
      client_id varchar(10),
      builder_id varchar(10),
      broker_id varchar(10),
      constraint cf_pk primary key(client_id),
      constraint cf_clnt_fk foreign key (client_id) references client,
      constraint cf_bd_fk foreign key (builder_id) references builder,
      constraint cf_br_fk foreign key (broker_id) references broker);

Table created.
```



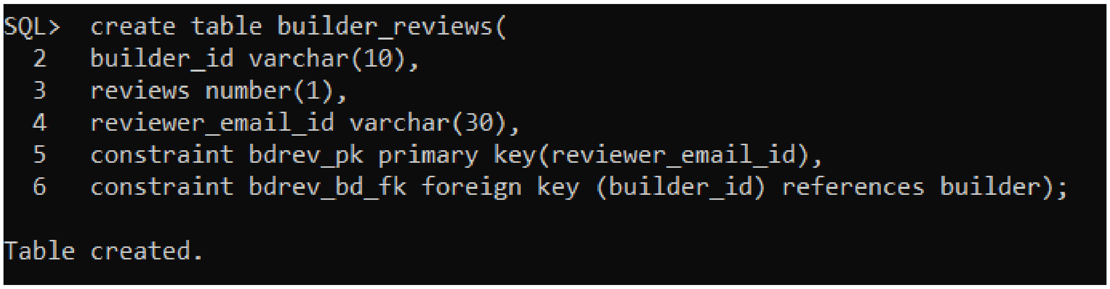
```
SQL> create table owned_by(
     owner_id varchar(10),
     property_id varchar(10),
     constraint ob_pk primary key(owner_id,property_id),
     constraint ob_onr_fk foreign key (owner_id) references owner,
     constraint ob_prop_fk foreign key (property_id) references property);

Table created.
```



```
SQL> create table built_by(
       builder_id varchar(10),
       property_id varchar(10),
       constraint bb_pk primary key(builder_id,property_id),
       constraint bb_bd_fk foreign key (builder_id) references builder,
       constraint bb_prop_fk foreign key (property_id) references property);

Table created.
```

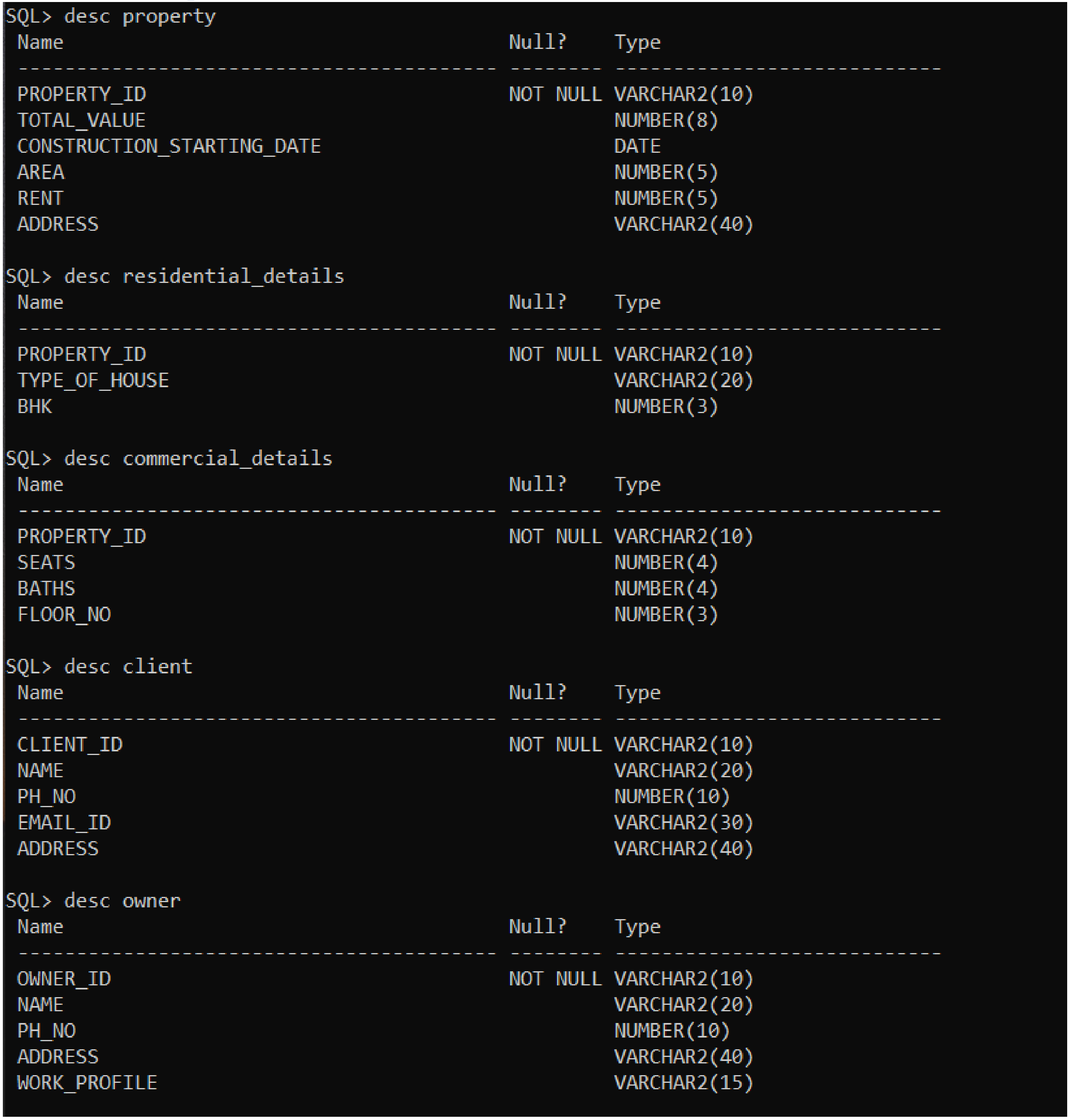


```
SQL> create table sells(
      broker_id varchar(10),
      property_id varchar(10),
      constraint sl_pk primary key(broker_id,property_id),
      constraint sl_br_fk foreign key (broker_id) references broker,
      constraint sl_prop_fk foreign key (property_id) references property
     );

Table created.
```

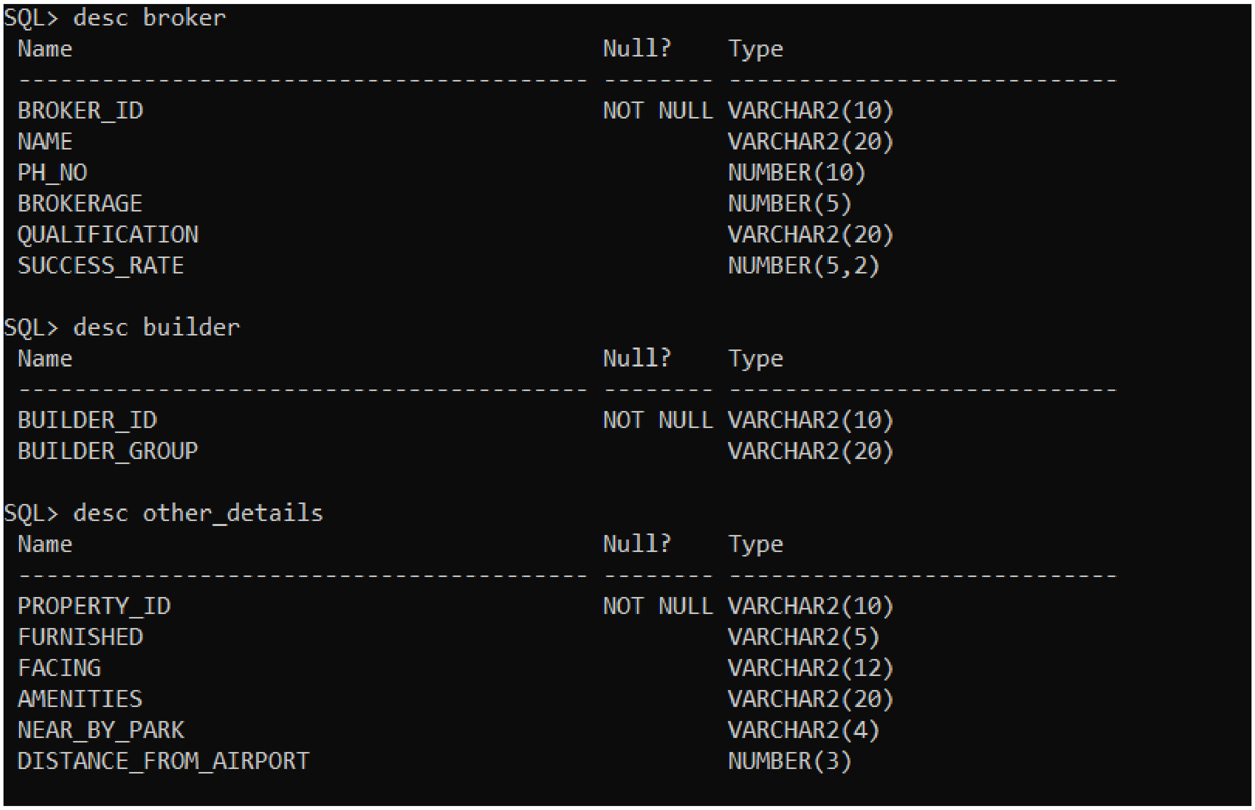


```
SQL>  create table builder_reviews(
      builder_id varchar(10),
      reviews number(1),
      reviewer_email_id varchar(30),
      constraint bdrev_pk primary key(reviewer_email_id),
      constraint bdrev_bd_fk foreign key (builder_id) references builder);

Table created.
```

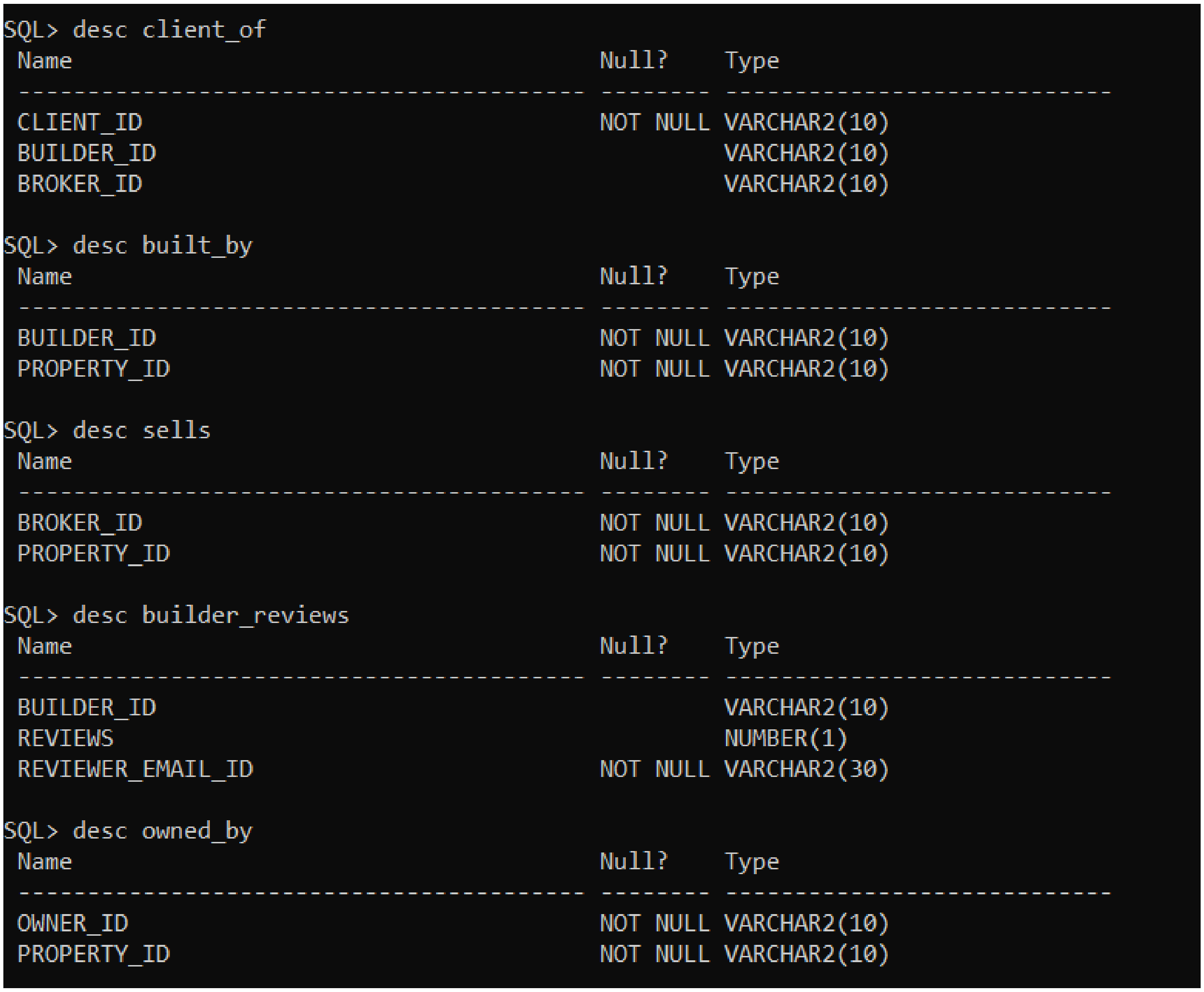
Table Descriptions



```
SQL> desc property
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROPERTY_ID                               NOT NULL VARCHAR2(10)
 TOTAL_VALUE                                        NUMBER(8)
 CONSTRUCTION_STARTING_DATE                         DATE
 AREA                                               NUMBER(5)
 RENT                                               NUMBER(5)
 ADDRESS                                            VARCHAR2(40)

SQL> desc residential_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROPERTY_ID                               NOT NULL VARCHAR2(10)
 TYPE_OF_HOUSE                                      VARCHAR2(20)
 BHK                                                NUMBER(3)

SQL> desc commercial_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROPERTY_ID                               NOT NULL VARCHAR2(10)
 SEATS                                              NUMBER(4)
 BATHS                                              NUMBER(4)
 FLOOR_NO                                           NUMBER(3)

SQL> desc client
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_ID                                 NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(20)
 PH_NO                                              NUMBER(10)
 EMAIL_ID                                           VARCHAR2(30)
 ADDRESS                                            VARCHAR2(40)

SQL> desc owner
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER_ID                                  NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(20)
 PH_NO                                              NUMBER(10)
 ADDRESS                                            VARCHAR2(40)
 WORK_PROFILE                                       VARCHAR2(15)
```

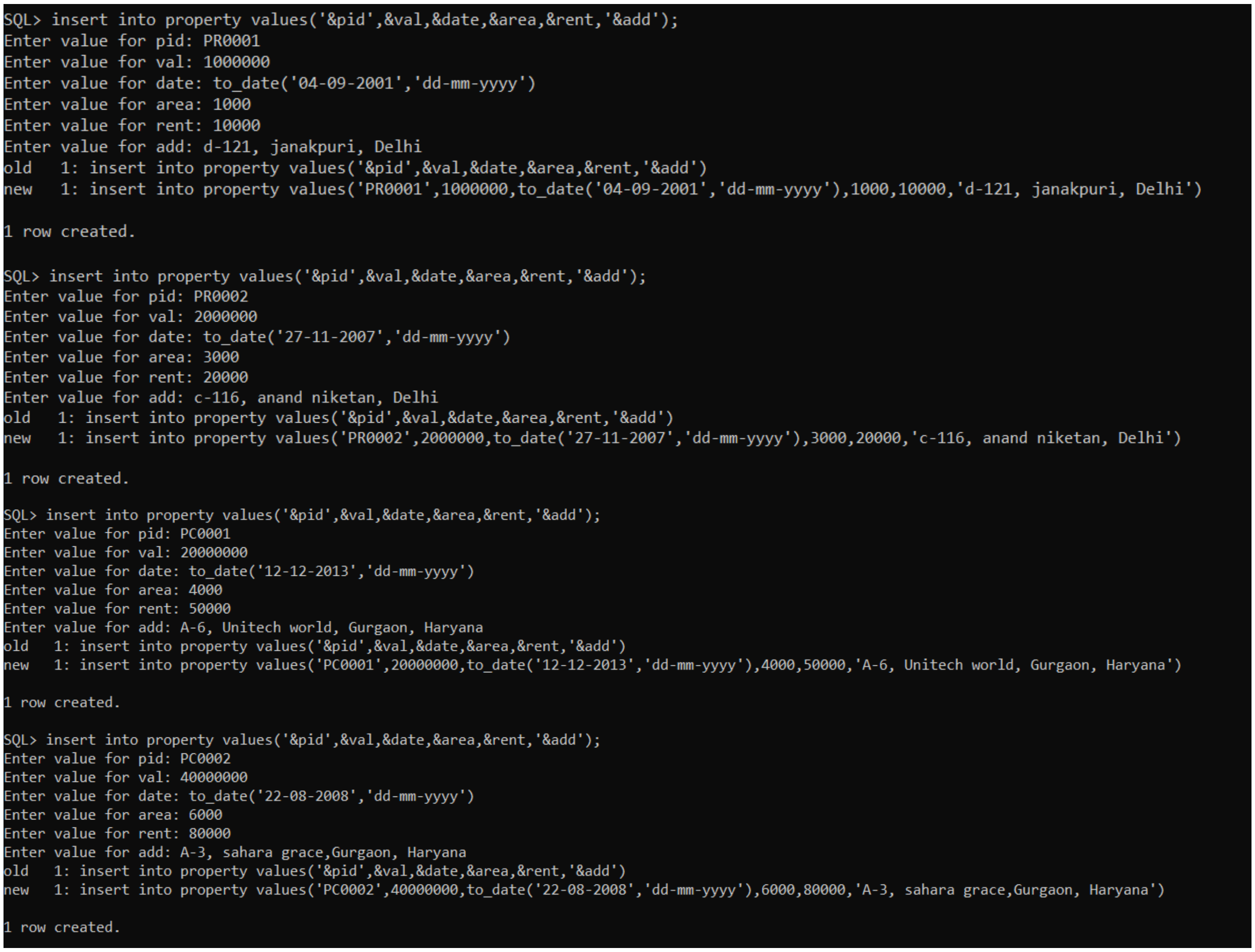


```
SQL> desc broker
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BROKER_ID                                 NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(20)
 PH_NO                                              NUMBER(10)
 BROKERAGE                                          NUMBER(5)
 QUALIFICATION                                      VARCHAR2(20)
 SUCCESS_RATE                                       NUMBER(5,2)

SQL> desc builder
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDER_ID                                NOT NULL VARCHAR2(10)
 BUILDER_GROUP                                      VARCHAR2(20)

SQL> desc other_details
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROPERTY_ID                               NOT NULL VARCHAR2(10)
 FURNISHED                                          VARCHAR2(5)
 FACING                                             VARCHAR2(12)
 AMENITIES                                          VARCHAR2(20)
 NEAR_BY_PARK                                       VARCHAR2(4)
 DISTANCE_FROM_AIRPORT                              NUMBER(3)
```

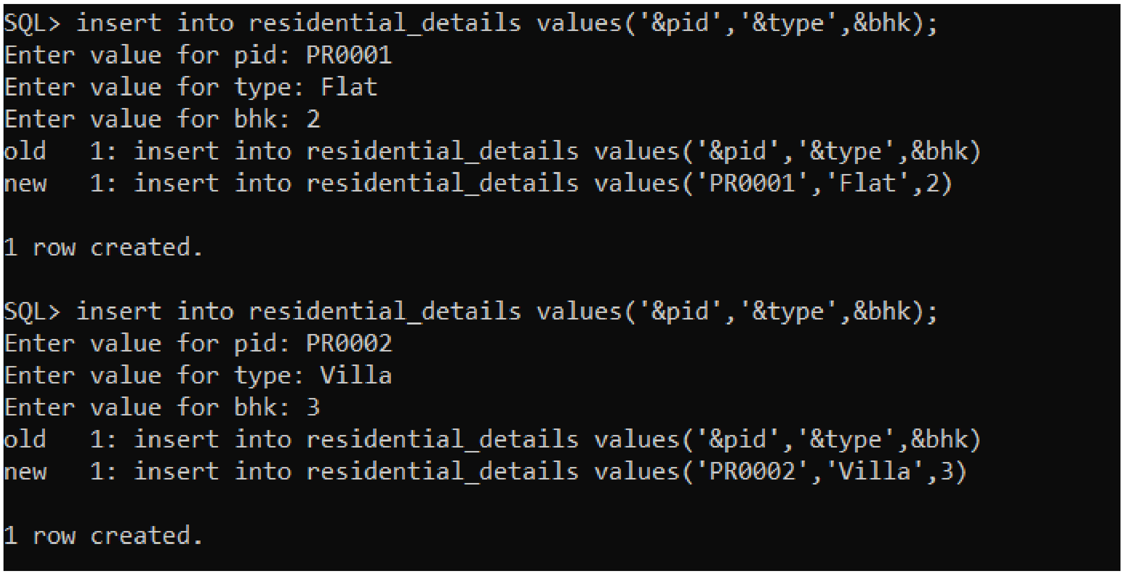


```
SQL> desc client_of
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_ID                                 NOT NULL VARCHAR2(10)
 BUILDER_ID                                         VARCHAR2(10)
 BROKER_ID                                          VARCHAR2(10)

SQL> desc built_by
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDER_ID                                NOT NULL VARCHAR2(10)
 PROPERTY_ID                               NOT NULL VARCHAR2(10)

SQL> desc sells
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BROKER_ID                                 NOT NULL VARCHAR2(10)
 PROPERTY_ID                               NOT NULL VARCHAR2(10)

SQL> desc builder_reviews
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BUILDER_ID                                         VARCHAR2(10)
 REVIEWS                                            NUMBER(1)
 REVIEWER_EMAIL_ID                         NOT NULL VARCHAR2(30)

SQL> desc owned_by
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER_ID                                  NOT NULL VARCHAR2(10)
 PROPERTY_ID                               NOT NULL VARCHAR2(10)
```

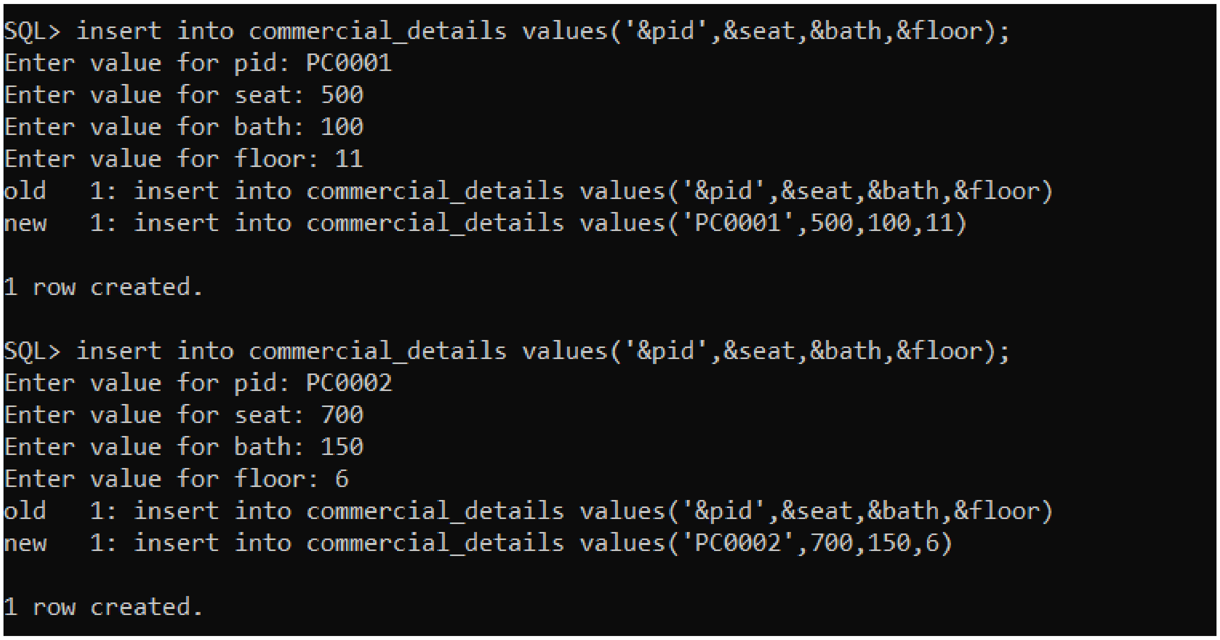
Table Insertions



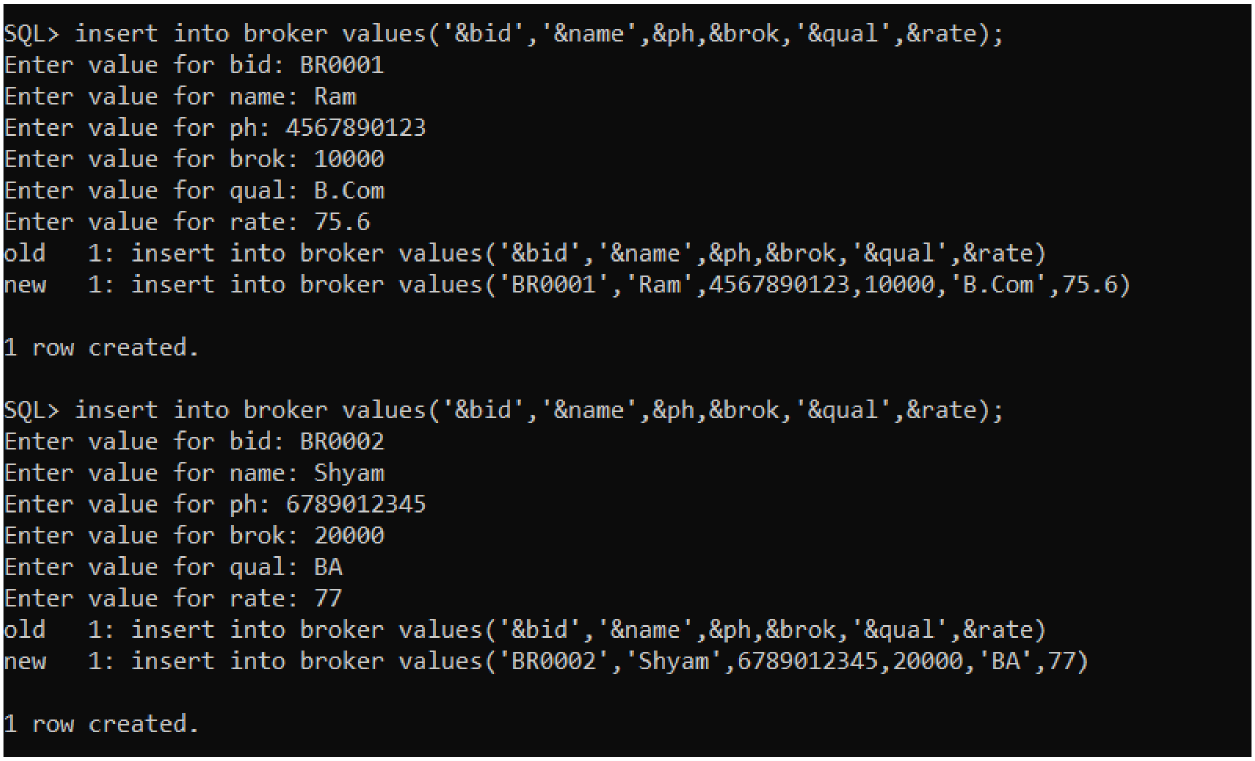
```
SQL> insert into property values('&pid',&val,&date,&area,&rent,'&add');
Enter value for pid: PR0001
Enter value for val: 1000000
Enter value for date: to_date('04-09-2001','dd-mm-yyyy')
Enter value for area: 1000
Enter value for rent: 10000
Enter value for add: d-121, janakpuri, Delhi
old   1: insert into property values('&pid',&val,&date,&area,&rent,'&add')
new   1: insert into property values('PR0001',1000000,to_date('04-09-2001','dd-mm-yyyy'),1000,10000,'d-121, janakpuri, Delhi')

1 row created.

SQL> insert into property values('&pid',&val,&date,&area,&rent,'&add');
Enter value for pid: PR0002
Enter value for val: 2000000
Enter value for date: to_date('27-11-2007','dd-mm-yyyy')
Enter value for area: 3000
Enter value for rent: 20000
Enter value for add: c-116, anand niketan, Delhi
old   1: insert into property values('&pid',&val,&date,&area,&rent,'&add')
new   1: insert into property values('PR0002',2000000,to_date('27-11-2007','dd-mm-yyyy'),3000,20000,'c-116, anand niketan, Delhi')

1 row created.

SQL> insert into property values('&pid',&val,&date,&area,&rent,'&add');
Enter value for pid: PC0001
Enter value for val: 20000000
Enter value for date: to_date('12-12-2013','dd-mm-yyyy')
Enter value for area: 4000
Enter value for rent: 50000
Enter value for add: A-6, Unitech world, Gurgaon, Haryana
old   1: insert into property values('&pid',&val,&date,&area,&rent,'&add')
new   1: insert into property values('PC0001',20000000,to_date('12-12-2013','dd-mm-yyyy'),4000,50000,'A-6, Unitech world, Gurgaon, Haryana')

1 row created.

SQL> insert into property values('&pid',&val,&date,&area,&rent,'&add');
Enter value for pid: PC0002
Enter value for val: 40000000
Enter value for date: to_date('22-08-2008','dd-mm-yyyy')
Enter value for area: 6000
Enter value for rent: 80000
Enter value for add: A-3, sahara grace,Gurgaon, Haryana
old   1: insert into property values('&pid',&val,&date,&area,&rent,'&add')
new   1: insert into property values('PC0002',40000000,to_date('22-08-2008','dd-mm-yyyy'),6000,80000,'A-3, sahara grace,Gurgaon, Haryana')

1 row created.
```



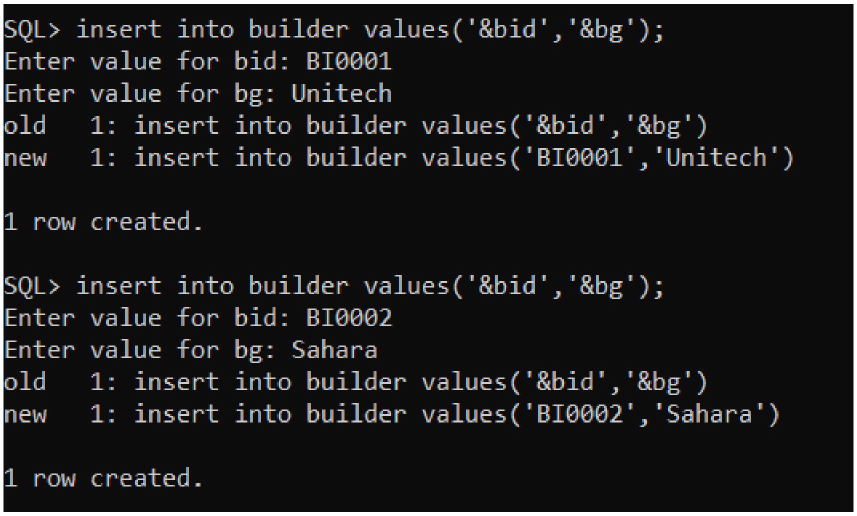
```
SQL> insert into residential_details values('&pid','&type',&bhk);
Enter value for pid: PR0001
Enter value for type: Flat
Enter value for bhk: 2
old   1: insert into residential_details values('&pid','&type',&bhk)
new   1: insert into residential_details values('PR0001','Flat',2)

1 row created.

SQL> insert into residential_details values('&pid','&type',&bhk);
Enter value for pid: PR0002
Enter value for type: Villa
Enter value for bhk: 3
old   1: insert into residential_details values('&pid','&type',&bhk)
new   1: insert into residential_details values('PR0002','Villa',3)

1 row created.
```



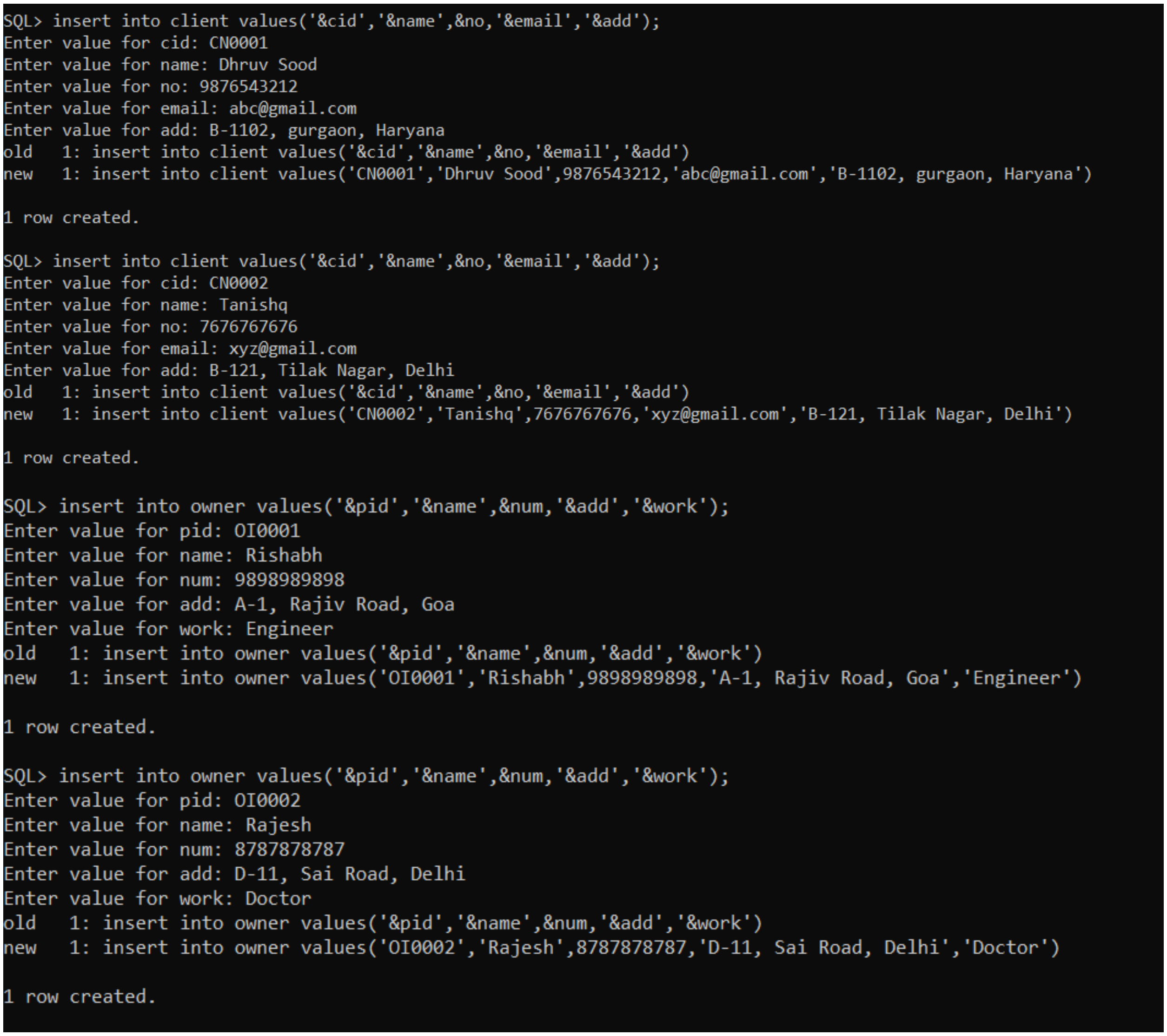
```
SQL> insert into commercial_details values('&pid',&seat,&bath,&floor);
Enter value for pid: PC0001
Enter value for seat: 500
Enter value for bath: 100
Enter value for floor: 11
old   1: insert into commercial_details values('&pid',&seat,&bath,&floor)
new   1: insert into commercial_details values('PC0001',500,100,11)

1 row created.

SQL> insert into commercial_details values('&pid',&seat,&bath,&floor);
Enter value for pid: PC0002
Enter value for seat: 700
Enter value for bath: 150
Enter value for floor: 6
old   1: insert into commercial_details values('&pid',&seat,&bath,&floor)
new   1: insert into commercial_details values('PC0002',700,150,6)

1 row created.
```



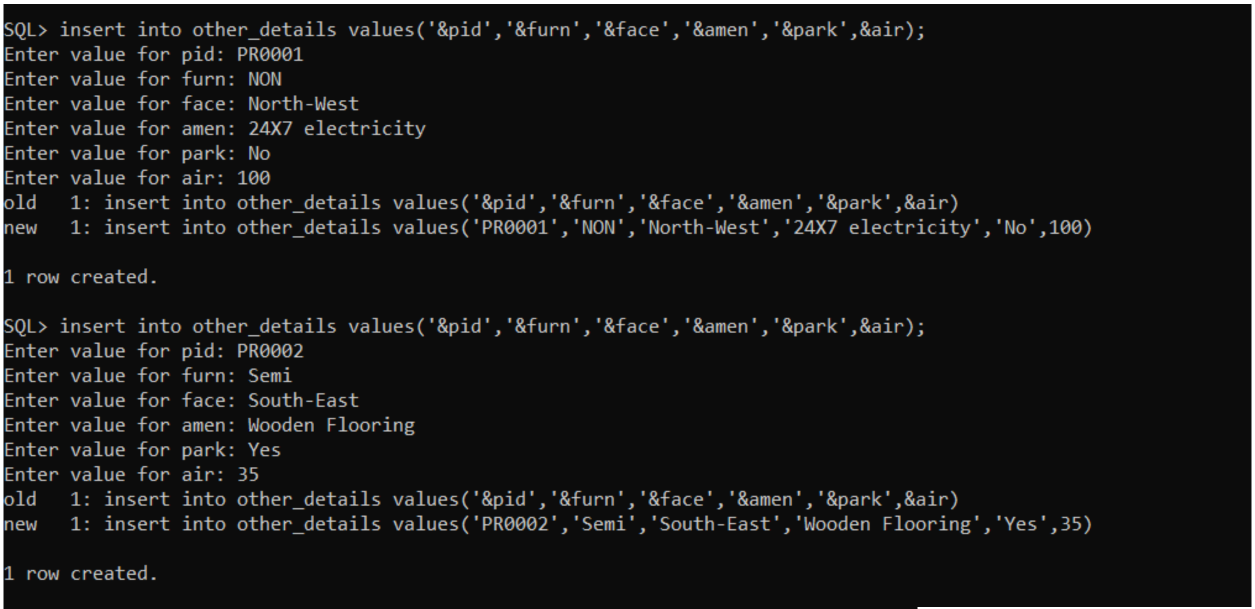
```
SQL> insert into broker values('&bid','&name',&ph,&brok,'&qual',&rate);
Enter value for bid: BR0001
Enter value for name: Ram
Enter value for ph: 4567890123
Enter value for brok: 10000
Enter value for qual: B.Com
Enter value for rate: 75.6
old   1: insert into broker values('&bid','&name',&ph,&brok,'&qual',&rate)
new   1: insert into broker values('BR0001','Ram',4567890123,10000,'B.Com',75.6)

1 row created.

SQL> insert into broker values('&bid','&name',&ph,&brok,'&qual',&rate);
Enter value for bid: BR0002
Enter value for name: Shyam
Enter value for ph: 6789012345
Enter value for brok: 20000
Enter value for qual: BA
Enter value for rate: 77
old   1: insert into broker values('&bid','&name',&ph,&brok,'&qual',&rate)
new   1: insert into broker values('BR0002','Shyam',6789012345,20000,'BA',77)

1 row created.
```



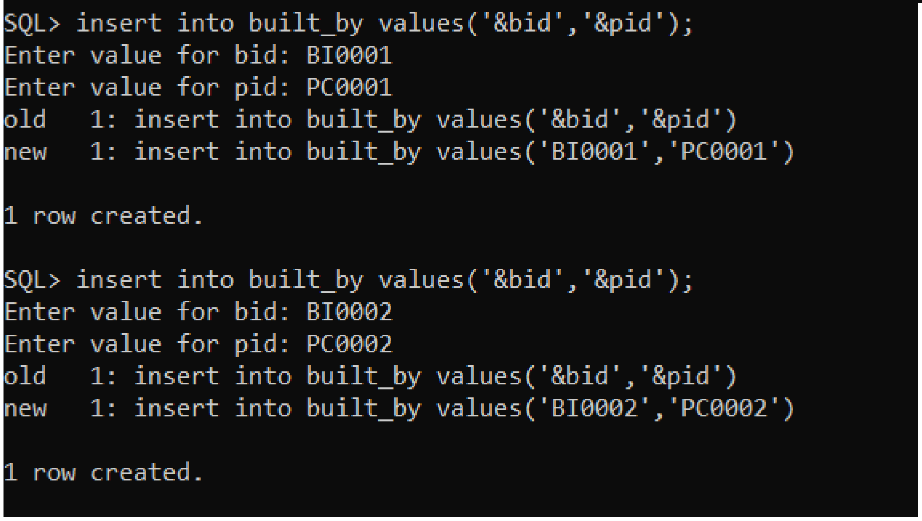
```
SQL> insert into builder values('&bid','&bg');
Enter value for bid: BI0001
Enter value for bg: Unitech
old   1: insert into builder values('&bid','&bg')
new   1: insert into builder values('BI0001','Unitech')

1 row created.

SQL> insert into builder values('&bid','&bg');
Enter value for bid: BI0002
Enter value for bg: Sahara
old   1: insert into builder values('&bid','&bg')
new   1: insert into builder values('BI0002','Sahara')

1 row created.
```



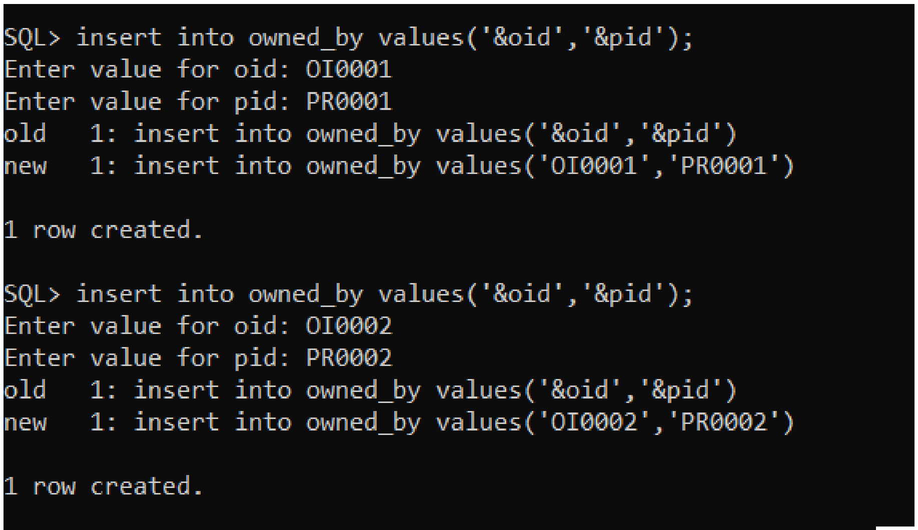
```
SQL> insert into client values('&cid','&name',&no,'&email','&add');
Enter value for cid: CN0001
Enter value for name: Dhruv Sood
Enter value for no: 9876543212
Enter value for email: abc@gmail.com
Enter value for add: B-1102, gurgaon, Haryana
old   1: insert into client values('&cid','&name',&no,'&email','&add')
new   1: insert into client values('CN0001','Dhruv Sood',9876543212,'abc@gmail.com','B-1102, gurgaon, Haryana')

1 row created.

SQL> insert into client values('&cid','&name',&no,'&email','&add');
Enter value for cid: CN0002
Enter value for name: Tanishq
Enter value for no: 7676767676
Enter value for email: xyz@gmail.com
Enter value for add: B-121, Tilak Nagar, Delhi
old   1: insert into client values('&cid','&name',&no,'&email','&add')
new   1: insert into client values('CN0002','Tanishq',7676767676,'xyz@gmail.com','B-121, Tilak Nagar, Delhi')

1 row created.

SQL> insert into owner values('&pid','&name',&num,'&add','&work');
Enter value for pid: OI0001
Enter value for name: Rishabh
Enter value for num: 9898989898
Enter value for add: A-1, Rajiv Road, Goa
Enter value for work: Engineer
old   1: insert into owner values('&pid','&name',&num,'&add','&work')
new   1: insert into owner values('OI0001','Rishabh',9898989898,'A-1, Rajiv Road, Goa','Engineer')

1 row created.

SQL> insert into owner values('&pid','&name',&num,'&add','&work');
Enter value for pid: OI0002
Enter value for name: Rajesh
Enter value for num: 8787878787
Enter value for add: D-11, Sai Road, Delhi
Enter value for work: Doctor
old   1: insert into owner values('&pid','&name',&num,'&add','&work')
new   1: insert into owner values('OI0002','Rajesh',8787878787,'D-11, Sai Road, Delhi','Doctor')

1 row created.
```



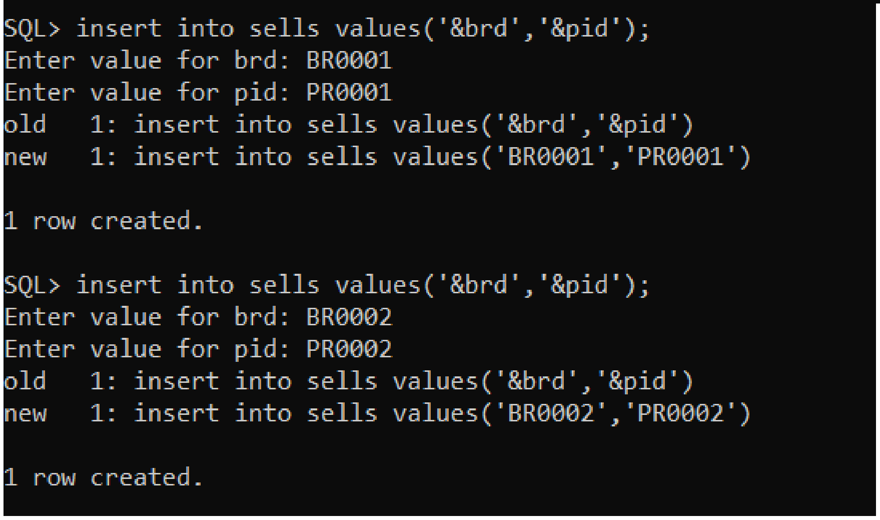
```
SQL> insert into other_details values('&pid','&furn','&face','&amen','&park',&air);
Enter value for pid: PR0001
Enter value for furn: NON
Enter value for face: North-West
Enter value for amen: 24X7 electricity
Enter value for park: No
Enter value for air: 100
old   1: insert into other_details values('&pid','&furn','&face','&amen','&park',&air)
new   1: insert into other_details values('PR0001','NON','North-West','24X7 electricity','No',100)

1 row created.

SQL> insert into other_details values('&pid','&furn','&face','&amen','&park',&air);
Enter value for pid: PR0002
Enter value for furn: Semi
Enter value for face: South-East
Enter value for amen: Wooden Flooring
Enter value for park: Yes
Enter value for air: 35
old   1: insert into other_details values('&pid','&furn','&face','&amen','&park',&air)
new   1: insert into other_details values('PR0002','Semi','South-East','Wooden Flooring','Yes',35)

1 row created.
```



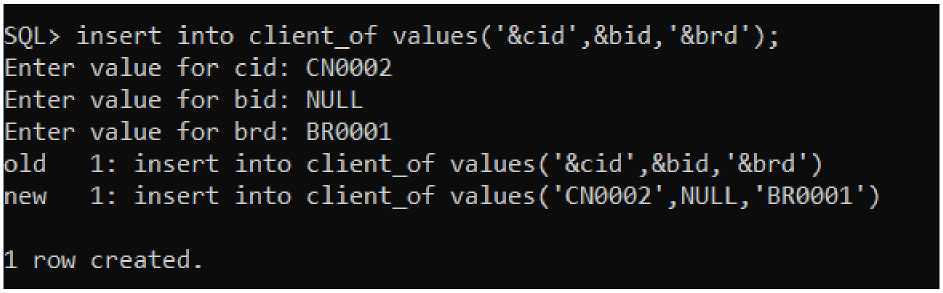
```
SQL> insert into built_by values('&bid','&pid');
Enter value for bid: BI0001
Enter value for pid: PC0001
old   1: insert into built_by values('&bid','&pid')
new   1: insert into built_by values('BI0001','PC0001')

1 row created.

SQL> insert into built_by values('&bid','&pid');
Enter value for bid: BI0002
Enter value for pid: PC0002
old   1: insert into built_by values('&bid','&pid')
new   1: insert into built_by values('BI0002','PC0002')

1 row created.
```



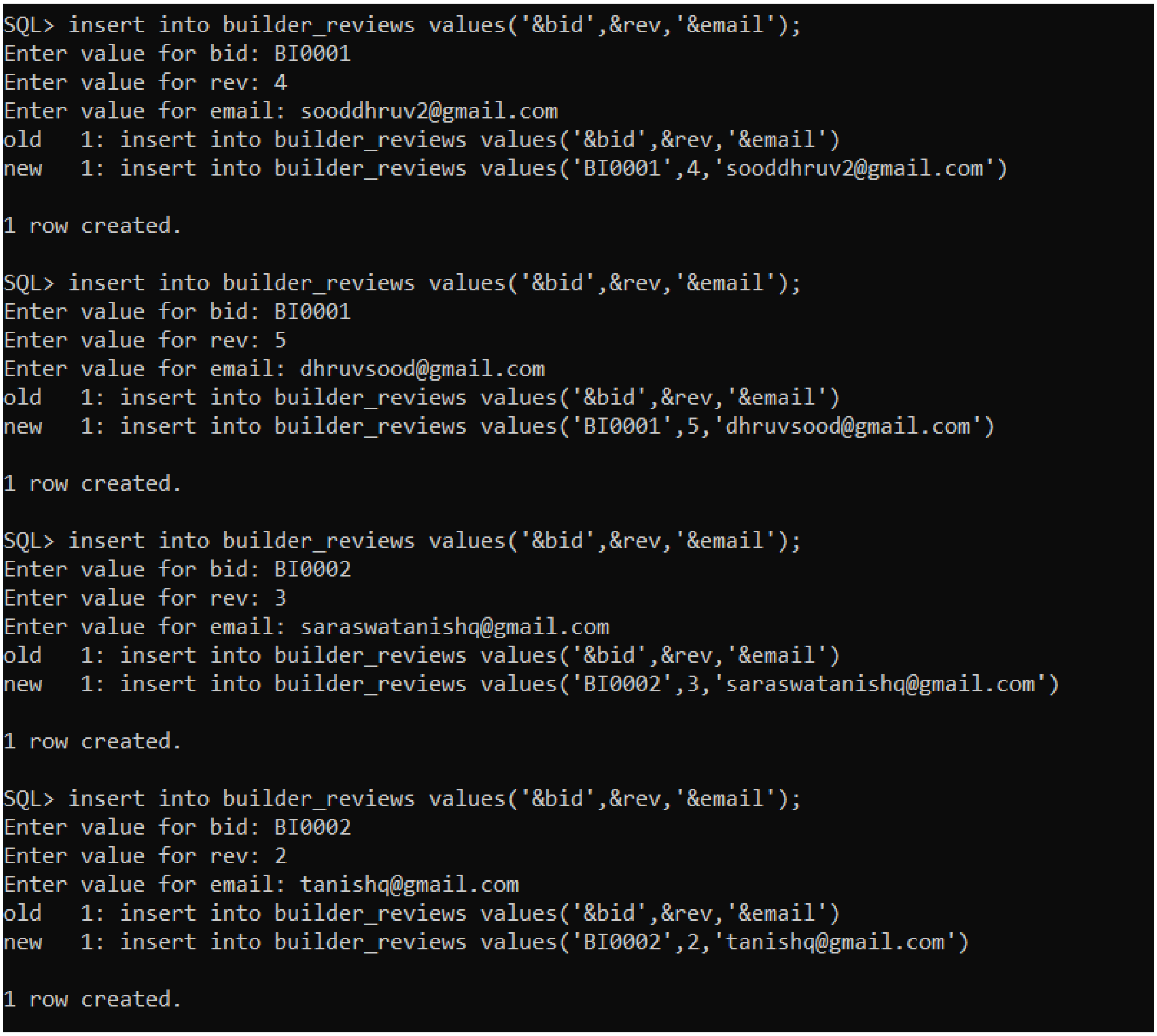
```
SQL> insert into owned_by values('&oid','&pid');
Enter value for oid: OI0001
Enter value for pid: PR0001
old   1: insert into owned_by values('&oid','&pid')
new   1: insert into owned_by values('OI0001','PR0001')

1 row created.

SQL> insert into owned_by values('&oid','&pid');
Enter value for oid: OI0002
Enter value for pid: PR0002
old   1: insert into owned_by values('&oid','&pid')
new   1: insert into owned_by values('OI0002','PR0002')

1 row created.
```



```
SQL> insert into sells values('&brd','&pid');
Enter value for brd: BR0001
Enter value for pid: PR0001
old   1: insert into sells values('&brd','&pid')
new   1: insert into sells values('BR0001','PR0001')

1 row created.

SQL> insert into sells values('&brd','&pid');
Enter value for brd: BR0002
Enter value for pid: PR0002
old   1: insert into sells values('&brd','&pid')
new   1: insert into sells values('BR0002','PR0002')

1 row created.
```

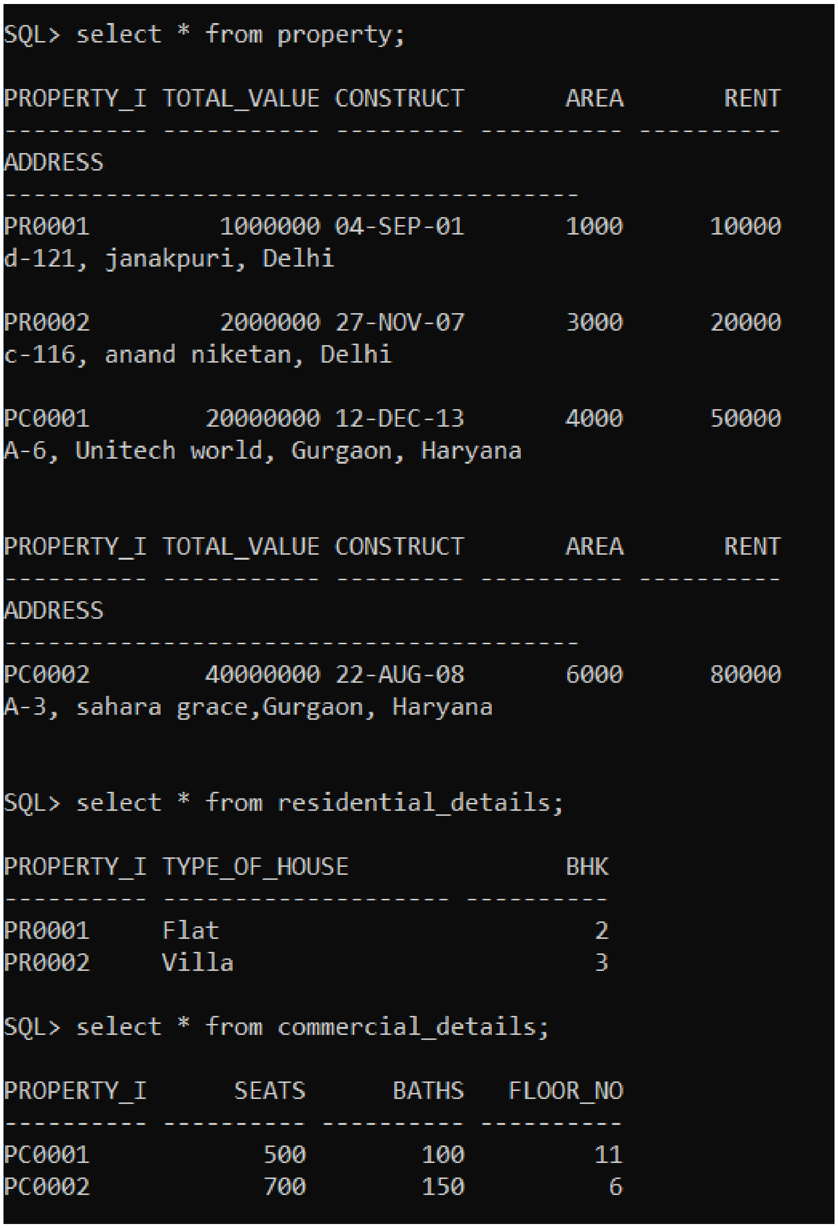


```
SQL> insert into client_of values('&cid',&bid,'&brd');
Enter value for cid: CN0002
Enter value for bid: NULL
Enter value for brd: BR0001
old   1: insert into client_of values('&cid',&bid,'&brd')
new   1: insert into client_of values('CN0002',NULL,'BR0001')

1 row created.
```

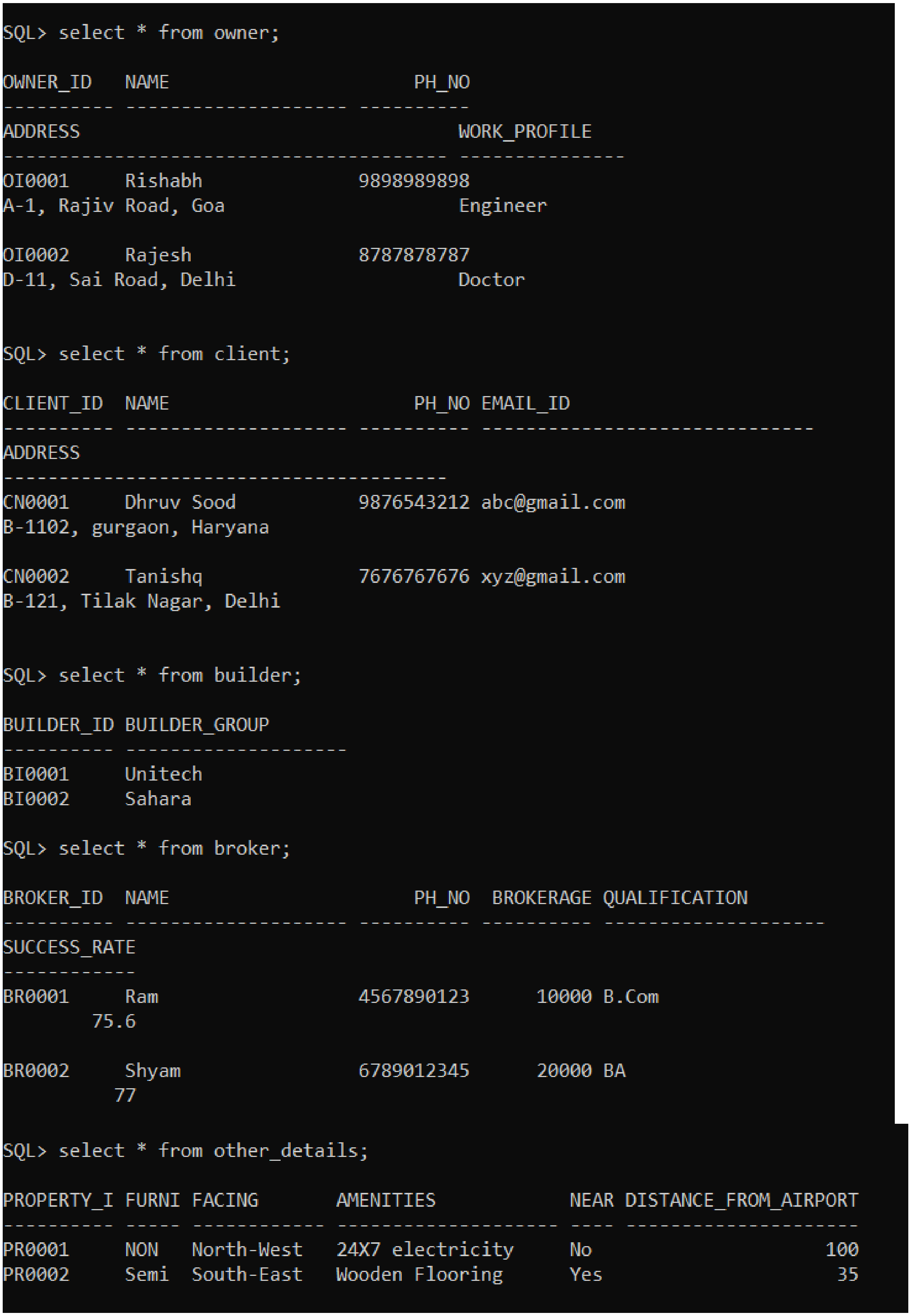


```
SQL> insert into builder_reviews values('&bid',&rev,'&email');
Enter value for bid: BI0001
Enter value for rev: 4
Enter value for email: sooddhruv2@gmail.com
old   1: insert into builder_reviews values('&bid',&rev,'&email')
new   1: insert into builder_reviews values('BI0001',4,'sooddhruv2@gmail.com')

1 row created.

SQL> insert into builder_reviews values('&bid',&rev,'&email');
Enter value for bid: BI0001
Enter value for rev: 5
Enter value for email: dhruvsood@gmail.com
old   1: insert into builder_reviews values('&bid',&rev,'&email')
new   1: insert into builder_reviews values('BI0001',5,'dhruvsood@gmail.com')

1 row created.

SQL> insert into builder_reviews values('&bid',&rev,'&email');
Enter value for bid: BI0002
Enter value for rev: 3
Enter value for email: saraswatanishq@gmail.com
old   1: insert into builder_reviews values('&bid',&rev,'&email')
new   1: insert into builder_reviews values('BI0002',3,'saraswatanishq@gmail.com')

1 row created.

SQL> insert into builder_reviews values('&bid',&rev,'&email');
Enter value for bid: BI0002
Enter value for rev: 2
Enter value for email: tanishq@gmail.com
old   1: insert into builder_reviews values('&bid',&rev,'&email')
new   1: insert into builder_reviews values('BI0002',2,'tanishq@gmail.com')

1 row created.
```

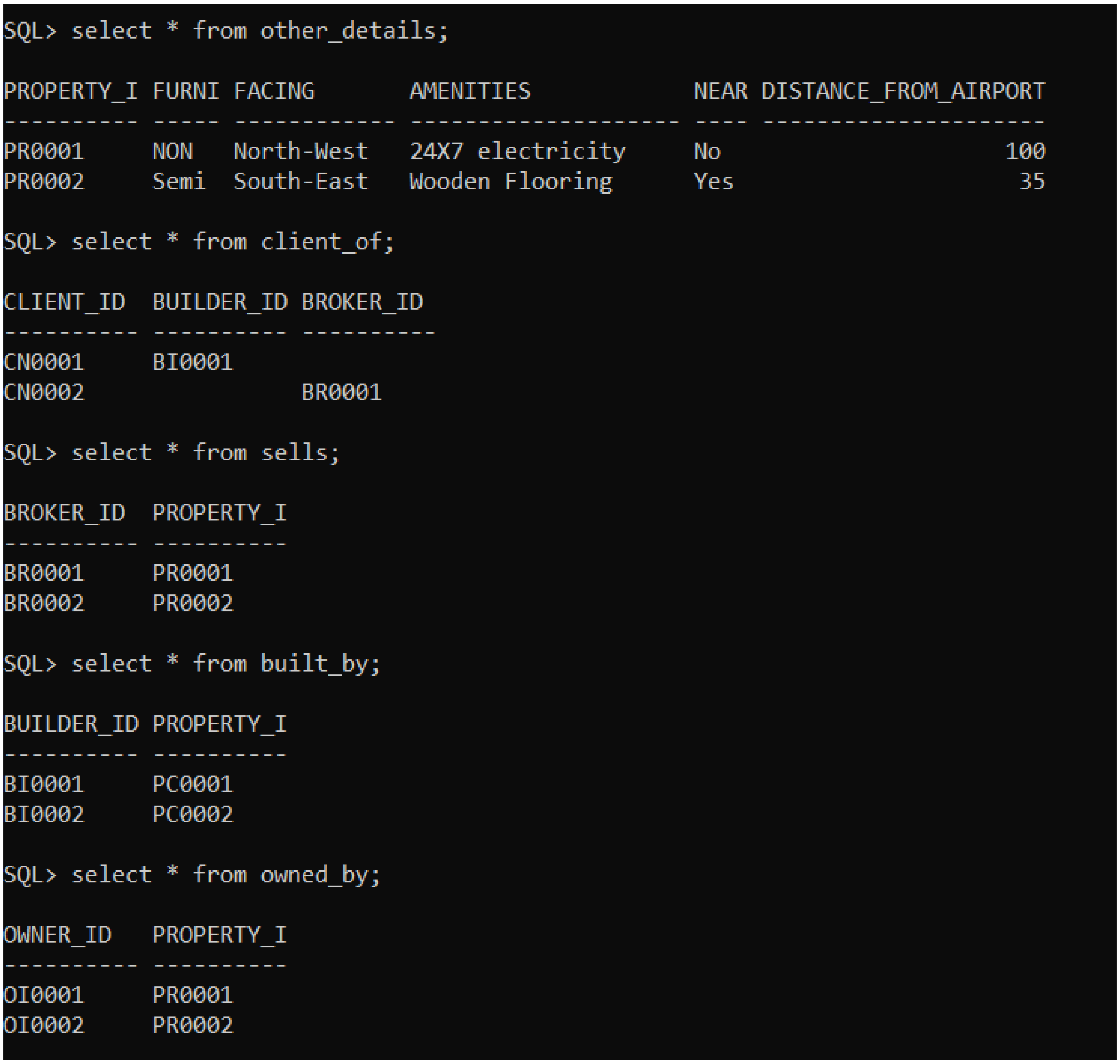
Table Data



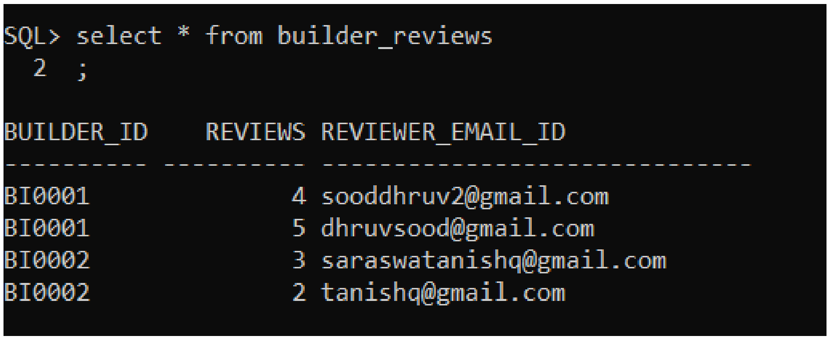
```
SQL> select * from property;

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PR0001         1000000 04-SEP-01       1000      10000
d-121, janakpuri, Delhi

PR0002         2000000 27-NOV-07       3000      20000
c-116, anand niketan, Delhi

PC0001        20000000 12-DEC-13       4000      50000
A-6, Unitech world, Gurgaon, Haryana


PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0002        40000000 22-AUG-08       6000      80000
A-3, sahara grace,Gurgaon, Haryana


SQL> select * from residential_details;

PROPERTY_I TYPE_OF_HOUSE               BHK
---------- -------------------- ----------
PR0001     Flat                          2
PR0002     Villa                         3

SQL> select * from commercial_details;

PROPERTY_I      SEATS      BATHS   FLOOR_NO
---------- ---------- ---------- ----------
PC0001            500        100         11
PC0002            700        150          6
```



```
SQL> select * from owner;

OWNER_ID   NAME                      PH_NO
---------- -------------------- ----------
ADDRESS                                  WORK_PROFILE
---------------------------------------- ---------------
OI0001     Rishabh              9898989898
A-1, Rajiv Road, Goa                     Engineer

OI0002     Rajesh               8787878787
D-11, Sai Road, Delhi                    Doctor


SQL> select * from client;

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS
----------------------------------------
CN0001     Dhruv Sood           9876543212 abc@gmail.com
B-1102, gurgaon, Haryana

CN0002     Tanishq              7676767676 xyz@gmail.com
B-121, Tilak Nagar, Delhi


SQL> select * from builder;

BUILDER_ID BUILDER_GROUP
---------- --------------------
BI0001     Unitech
BI0002     Sahara

SQL> select * from broker;

BROKER_ID  NAME                      PH_NO  BROKERAGE QUALIFICATION
---------- -------------------- ---------- ---------- --------------------
SUCCESS_RATE
------------
BR0001     Ram                  4567890123      10000 B.Com
        75.6

BR0002     Shyam                6789012345      20000 BA
          77

SQL> select * from other_details;

PROPERTY_I FURNI FACING       AMENITIES            NEAR DISTANCE_FROM_AIRPORT
---------- ----- ------------ -------------------- ---- --------------------
PR0001     NON   North-West   24X7 electricity     No                    100
PR0002     Semi  South-East   Wooden Flooring      Yes                    35
```



```
SQL> select * from other_details;

PROPERTY_I FURNI FACING       AMENITIES            NEAR DISTANCE_FROM_AIRPORT
---------- ----- ------------ -------------------- ---- ---------------------
PR0001     NON   North-West   24X7 electricity     No                     100
PR0002     Semi  South-East   Wooden Flooring      Yes                     35

SQL> select * from client_of;

CLIENT_ID  BUILDER_ID BROKER_ID
---------- ---------- ----------
CN0001     BI0001
CN0002                BR0001

SQL> select * from sells;

BROKER_ID  PROPERTY_I
---------- ----------
BR0001     PR0001
BR0002     PR0002

SQL> select * from built_by;

BUILDER_ID PROPERTY_I
---------- ----------
BI0001     PC0001
BI0002     PC0002

SQL> select * from owned_by;

OWNER_ID   PROPERTY_I
---------- ----------
OI0001     PR0001
OI0002     PR0002
```



```
SQL> select * from builder_reviews
  2  ;

BUILDER_ID    REVIEWS REVIEWER_EMAIL_ID
---------- ---------- ------------------------------
BI0001              4 sooddhruv2@gmail.com
BI0001              5 dhruvsood@gmail.com
BI0002              3 saraswatanishq@gmail.com
BI0002              2 tanishq@gmail.com
```

DATABASE MANAGEMENT SYSTEM

Review 3

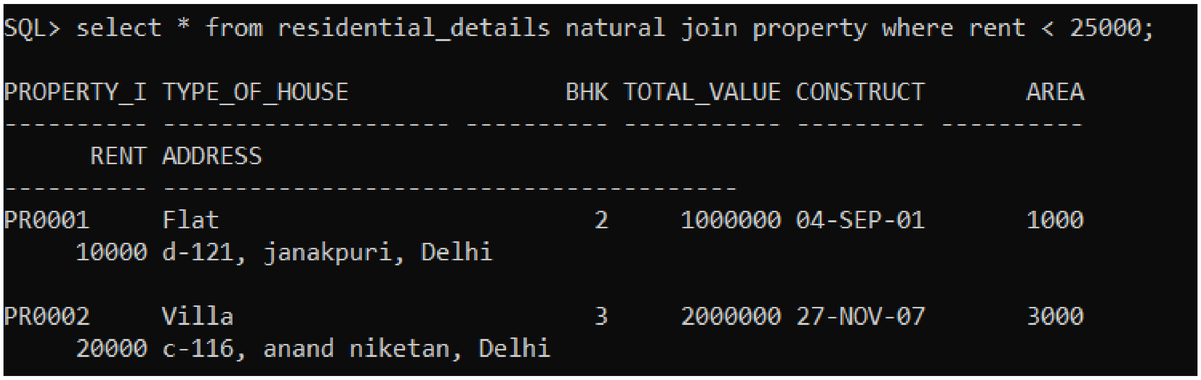
BY:

Dhruv Sood 19BIT023

Tanishq 19BIT0275

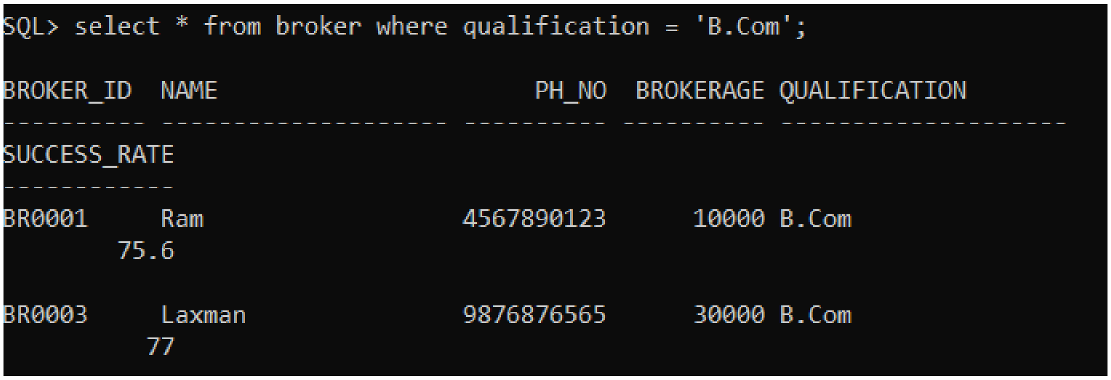
Select Query

Q.1 Display residential details where rent is less than 25k;



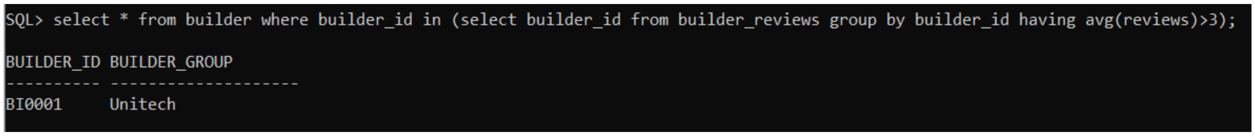
```
SQL> select * from residential_details natural join property where rent < 25000;

PROPERTY_I TYPE_OF_HOUSE               BHK TOTAL_VALUE CONSTRUCT       AREA
---------- -------------------- ---------- ----------- --------- ----------
      RENT ADDRESS
---------- ----------------------------------------
PR0001     Flat                          2     1000000 04-SEP-01       1000
     10000 d-121, janakpuri, Delhi

PR0002     Villa                         3     2000000 27-NOV-07       3000
     20000 c-116, anand niketan, Delhi
```

Q.2 Display broker details whose qualification is B.Com.



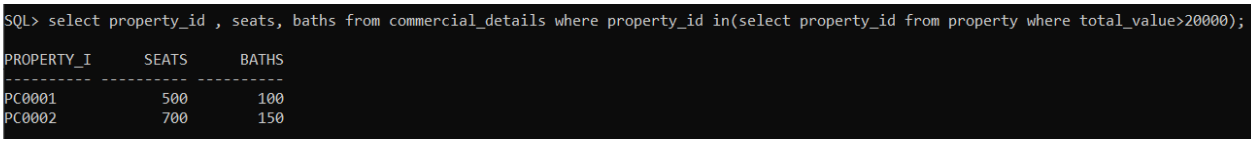
```
SQL> select * from broker where qualification = 'B.Com';

BROKER_ID  NAME                      PH_NO  BROKERAGE QUALIFICATION
---------- -------------------- ---------- ---------- --------------------
SUCCESS_RATE
------------
BR0001     Ram                  4567890123      10000 B.Com
        75.6

BR0003     Laxman               9876876565      30000 B.Com
          77
```

Q. Display builder details with builder review greater than 3.



```
SQL> select * from builder where builder_id in (select builder_id from builder_reviews group by builder_id having avg(reviews)>3);

BUILDER_ID BUILDER_GROUP
---------- --------------------
BI0001     Unitech
```

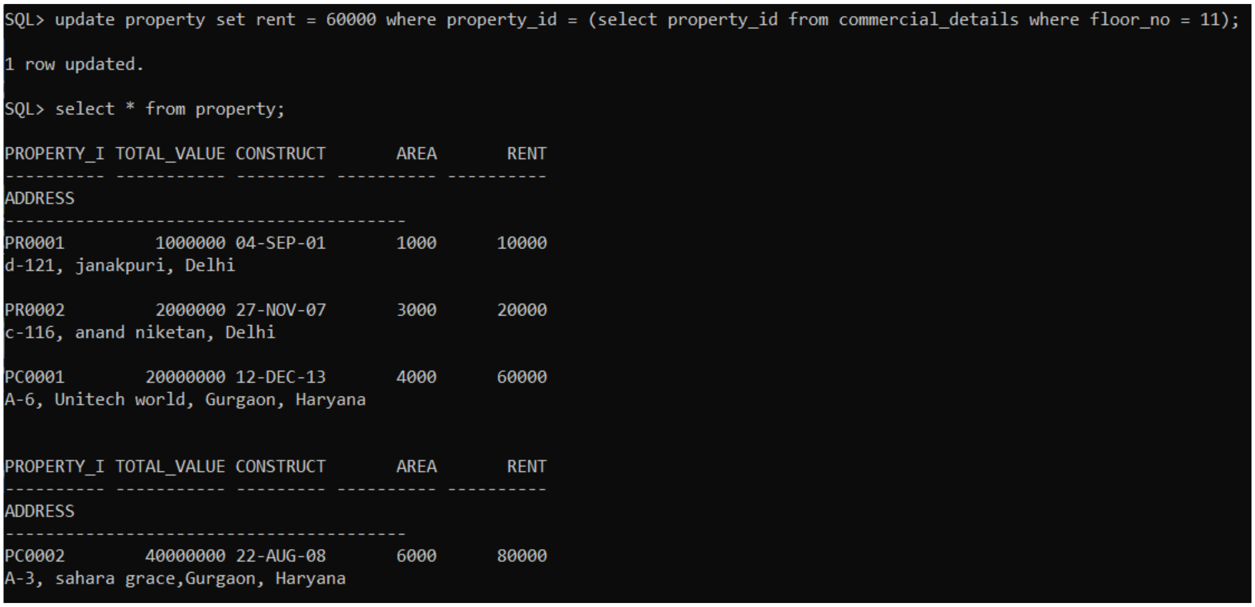
Q. Display property id seats and number of baths from commecial property whose total values is greater than 20000.



```
SQL> select property_id , seats, baths from commercial_details where property_id in(select property_id from property where total_value>20000);

PROPERTY_I      SEATS      BATHS
---------- ---------- ----------
PC0001            500        100
PC0002            700        150
```

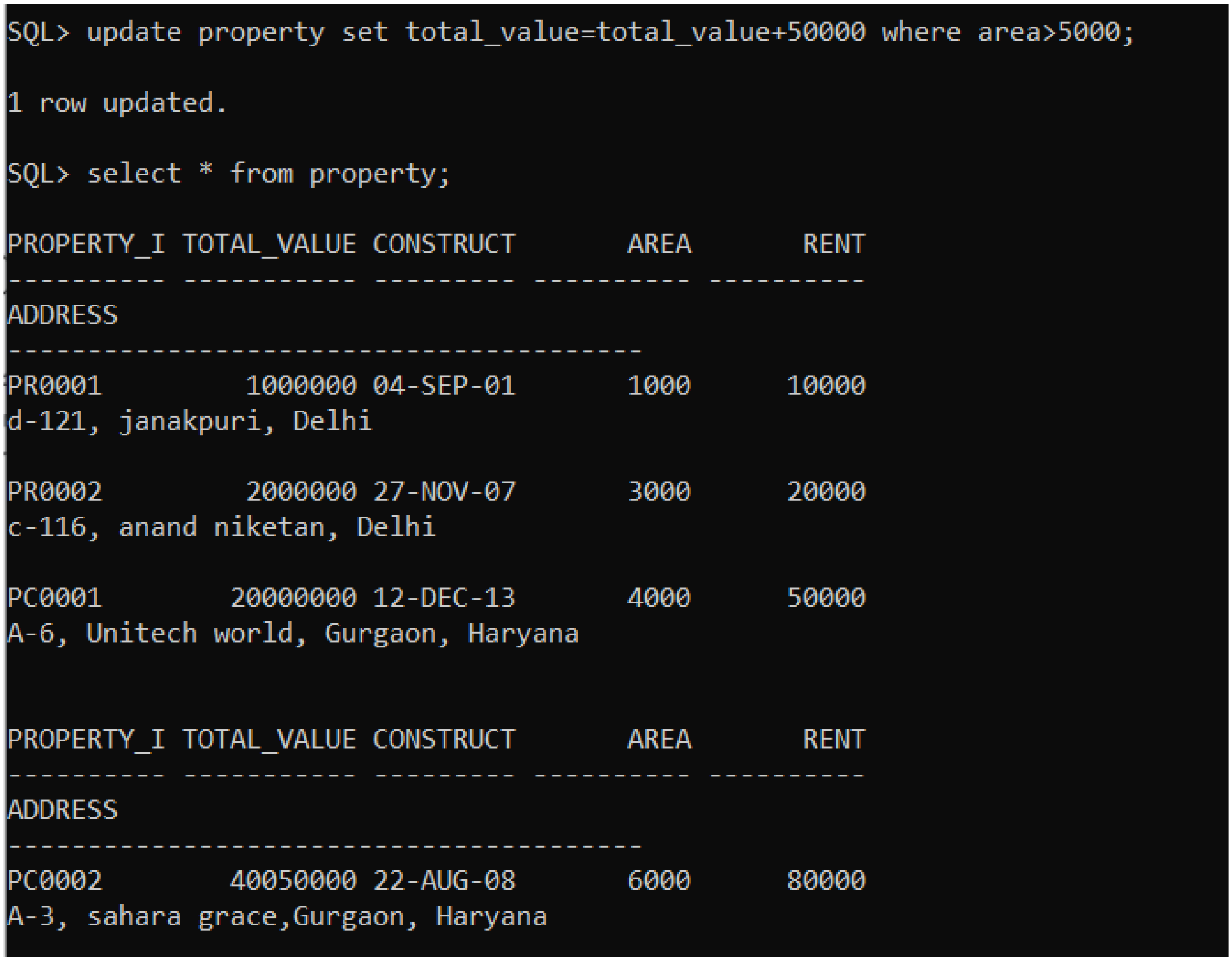
Update Query

Q.3 Update rent of the commercial property on floor no=11.



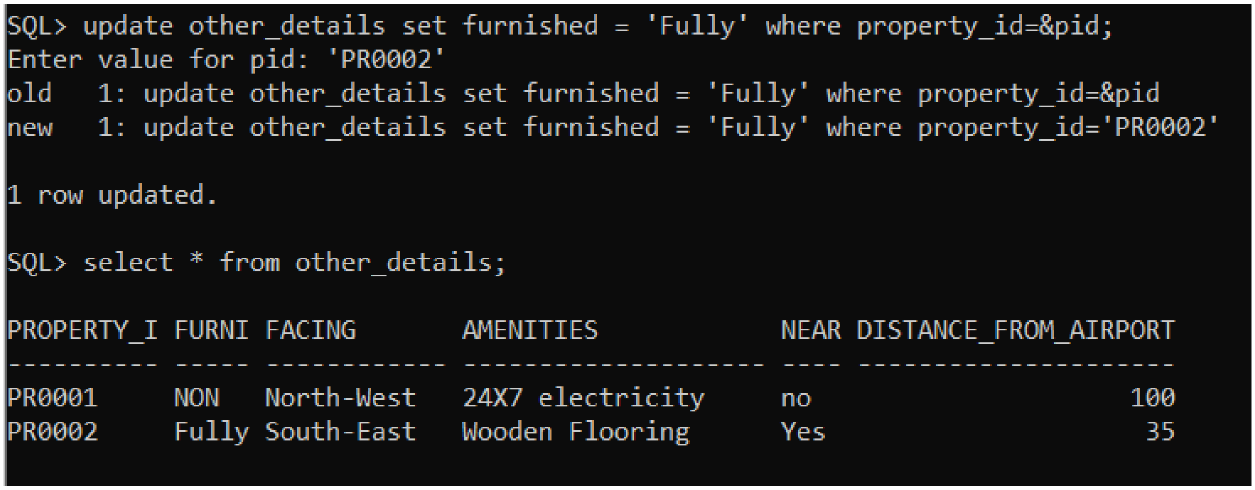
```
SQL> update property set rent = 60000 where property_id = (select property_id from commercial_details where floor_no = 11);

1 row updated.

SQL> select * from property;

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PR0001         1000000 04-SEP-01       1000      10000
d-121, janakpuri, Delhi

PR0002         2000000 27-NOV-07       3000      20000
c-116, anand niketan, Delhi

PC0001        20000000 12-DEC-13       4000      60000
A-6, Unitech world, Gurgaon, Haryana


PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0002        40000000 22-AUG-08       6000      80000
A-3, sahara grace,Gurgaon, Haryana
```

Q. Increase the total value of the properties by 50000 rs which have area greater 5000 sq ft.



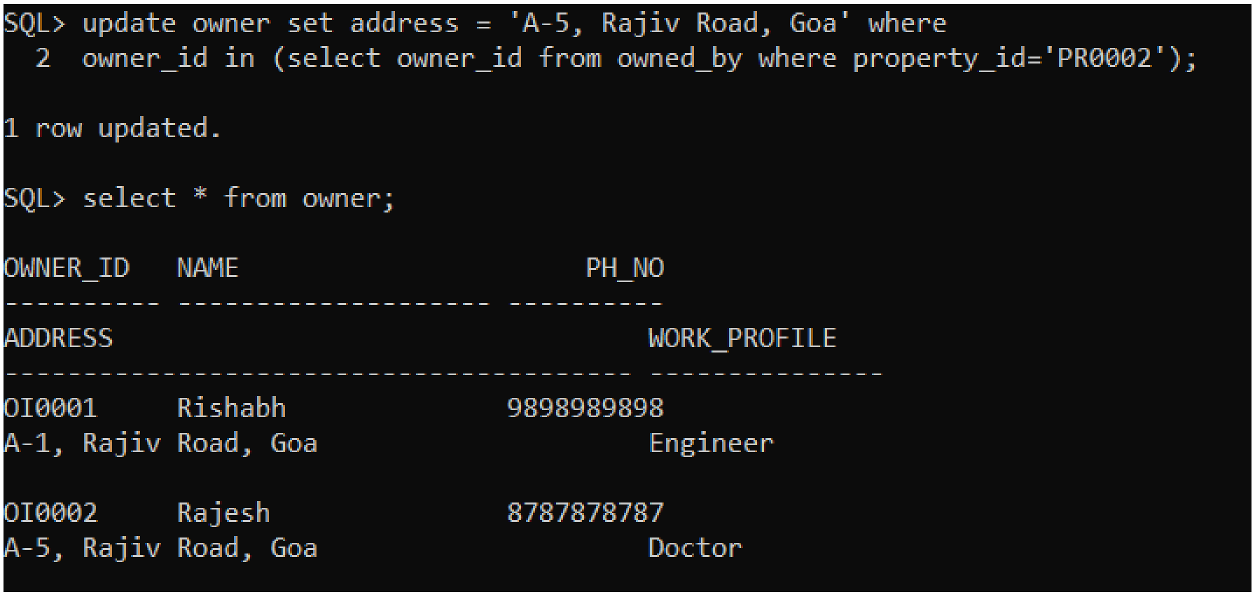
```
SQL> update property set total_value=total_value+50000 where area>5000;

1 row updated.

SQL> select * from property;

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PR0001         1000000 04-SEP-01       1000      10000
d-121, janakpuri, Delhi

PR0002         2000000 27-NOV-07       3000      20000
c-116, anand niketan, Delhi

PC0001        20000000 12-DEC-13       4000      50000
A-6, Unitech world, Gurgaon, Haryana


PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0002        40050000 22-AUG-08       6000      80000
A-3, sahara grace,Gurgaon, Haryana
```

Q. Update the furnishing status of the property with property id.



```
SQL> update other_details set furnished = 'Fully' where property_id=&pid;
Enter value for pid: 'PR0002'
old   1: update other_details set furnished = 'Fully' where property_id=&pid
new   1: update other_details set furnished = 'Fully' where property_id='PR0002'

1 row updated.

SQL> select * from other_details;

PROPERTY_I FURNI FACING       AMENITIES            NEAR DISTANCE_FROM_AIRPORT
---------- ----- ------------ -------------------- ---- ---------------------
PR0001     NON   North-West   24X7 electricity     no                     100
PR0002     Fully South-East   Wooden Flooring      Yes                     35
```

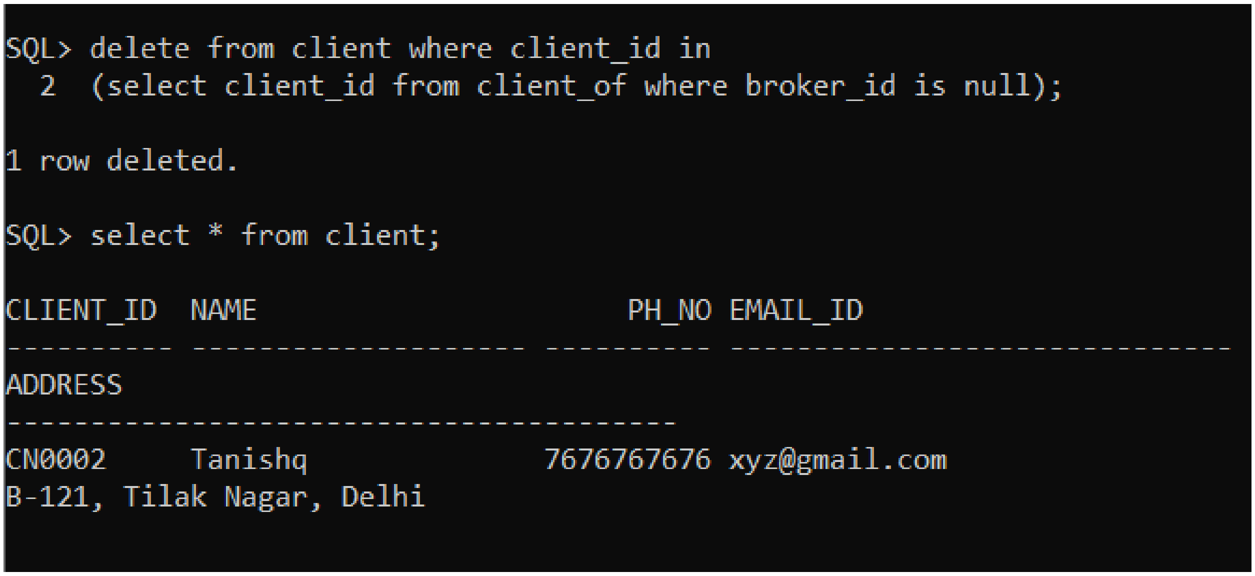
Q. Update current address of the owner of the property with property_id PR0002.



```
SQL> update owner set address = 'A-5, Rajiv Road, Goa' where
  2  owner_id in (select owner_id from owned_by where property_id='PR0002');

1 row updated.

SQL> select * from owner;

OWNER_ID   NAME                           PH_NO
---------- -------------------- ----------
ADDRESS                                  WORK_PROFILE
---------------------------------------- ---------------
OI0001     Rishabh              9898989898
A-1, Rajiv Road, Goa                     Engineer

OI0002     Rajesh               8787878787
A-5, Rajiv Road, Goa                     Doctor
```

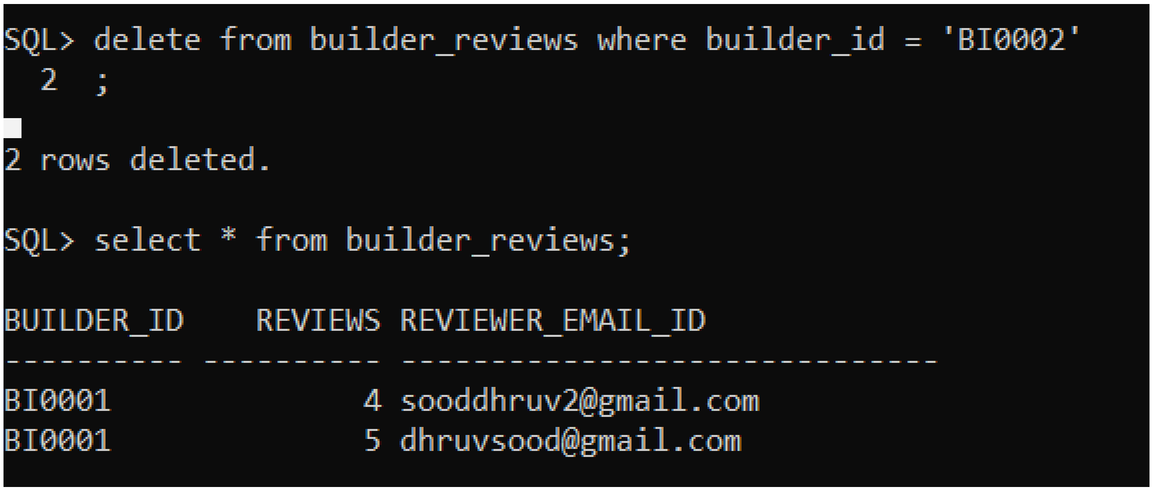
Delete Query

Q.4 delete client_details if there is no broker assosiated with client print no broker allotted.



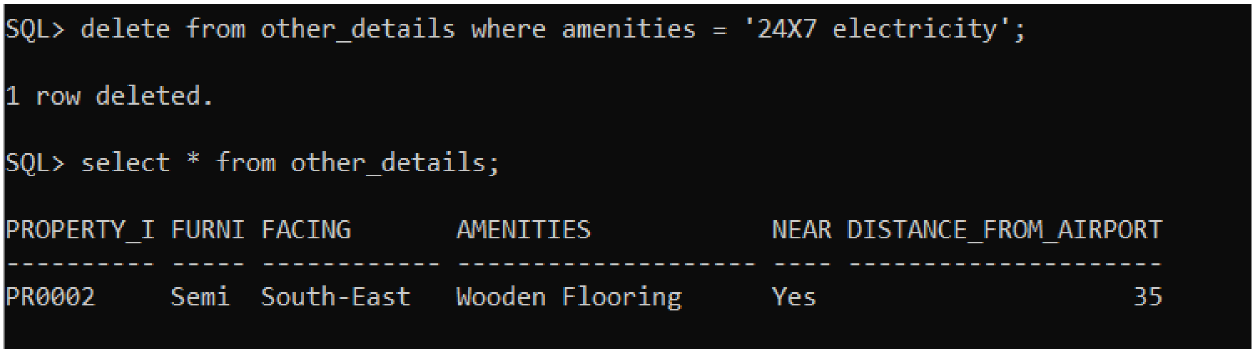
```
SQL> delete from client where client_id in
  2  (select client_id from client_of where broker_id is null);

1 row deleted.

SQL> select * from client;

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS
----------------------------------------
CN0002     Tanishq              7676767676 xyz@gmail.com
B-121, Tilak Nagar, Delhi
```

Q. delete the review of builder where builder_id = BI0002.



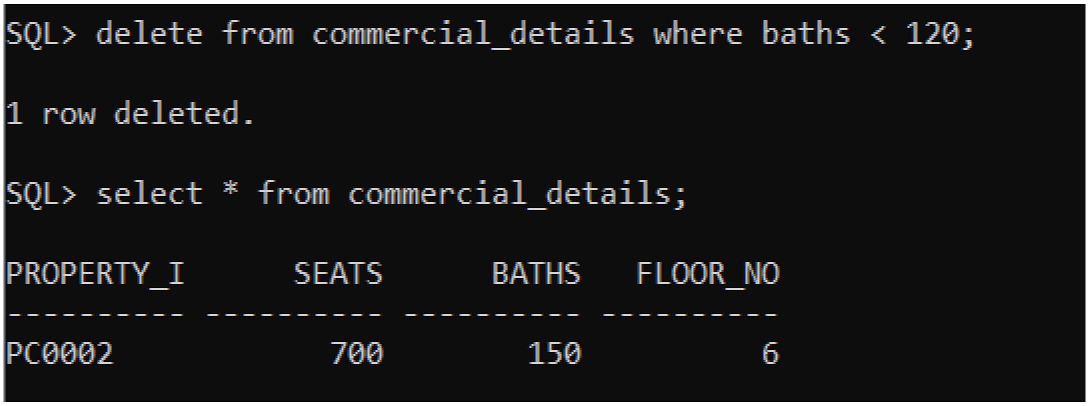
```
SQL> delete from builder_reviews where builder_id = 'BI0002'
  2  ;

2 rows deleted.

SQL> select * from builder_reviews;

BUILDER_ID    REVIEWS REVIEWER_EMAIL_ID
---------- ---------- ------------------------------
BI0001              4 sooddhruv2@gmail.com
BI0001              5 dhruvsood@gmail.com
```

Q. Delete other_details of property which have amenities as ‘24X7 electricit’.



```
SQL> delete from other_details where amenities = '24X7 electricity';

1 row deleted.

SQL> select * from other_details;

PROPERTY_I FURNI FACING       AMENITIES            NEAR DISTANCE_FROM_AIRPORT
---------- ----- ------------ -------------------- ---- ---------------------
PR0002     Semi  South-East   Wooden Flooring      Yes                     35
```

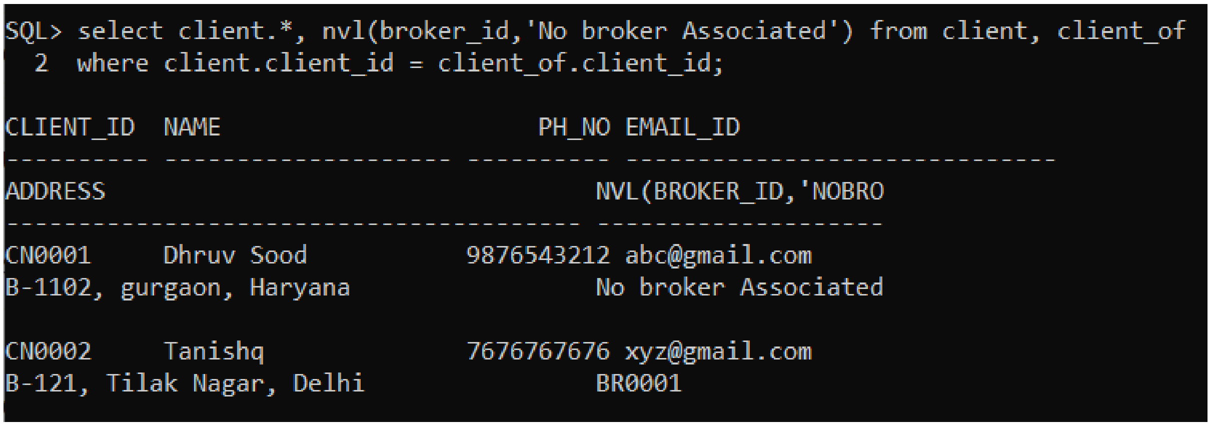
Q. Delete commercial_details where baths are less than 120;



```
SQL> delete from commercial_details where baths < 120;

1 row deleted.

SQL> select * from commercial_details;

PROPERTY_I      SEATS      BATHS   FLOOR_NO
---------- ---------- ---------- ----------
PC0002            700        150          6
```

NVL function Query

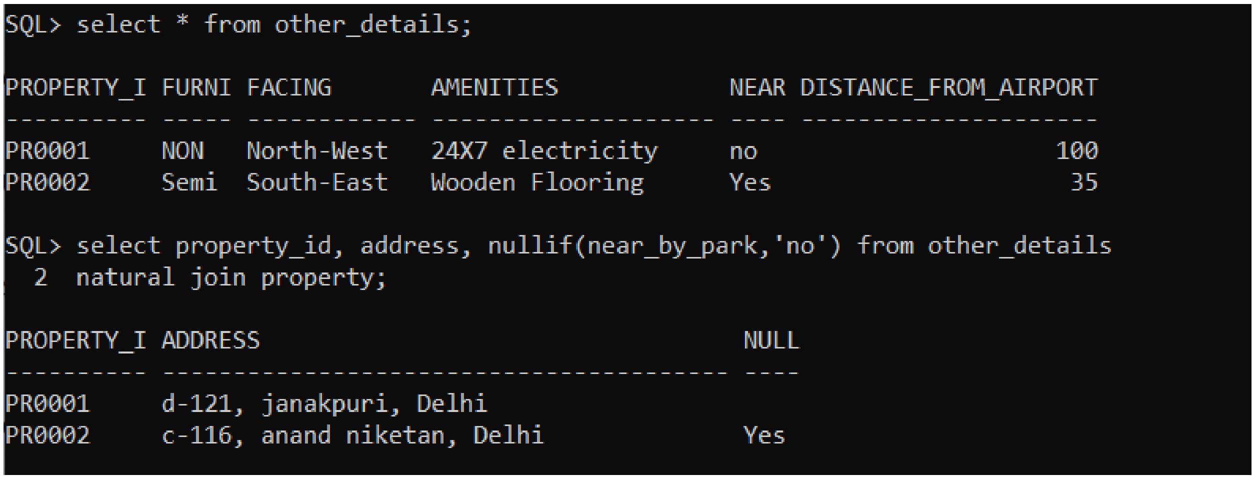
Q.5 select client_details if there is no broker assosiated with client print no broker allotted



```
SQL> select client.*, nvl(broker_id,'No broker Associated') from client, client_of
  2  where client.client_id = client_of.client_id;

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS                                  NVL(BROKER_ID,'NOBRO
---------------------------------------- --------------------
CN0001     Dhruv Sood           9876543212 abc@gmail.com
B-1102, gurgaon, Haryana                 No broker Associated

CN0002     Tanishq              7676767676 xyz@gmail.com
B-121, Tilak Nagar, Delhi                BR0001
```

Nullif function Query

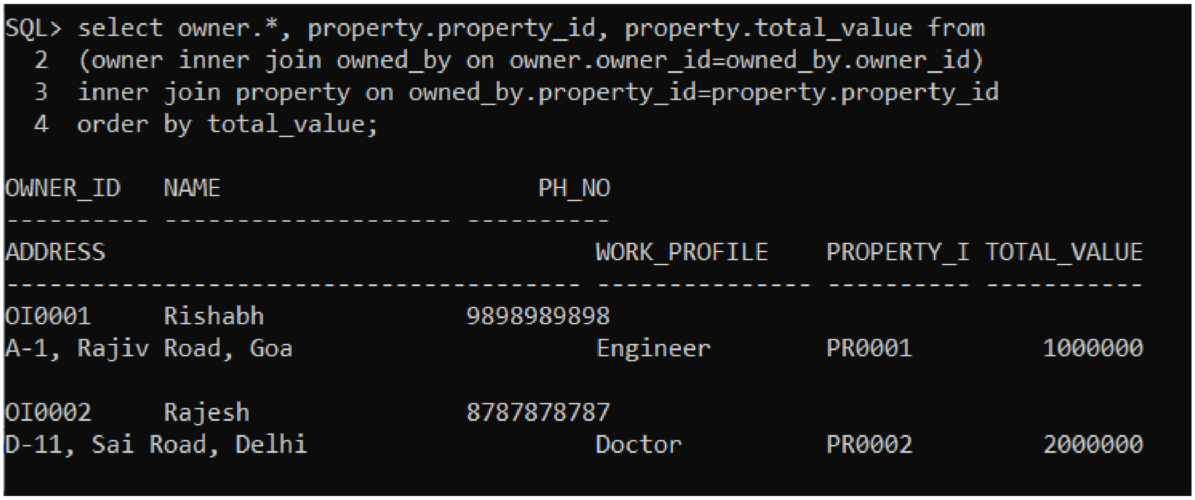
Q.6 Diplay address of residential_property if there is no near_by_park print null.



```
SQL> select * from other_details;

PROPERTY_I FURNI FACING       AMENITIES            NEAR DISTANCE_FROM_AIRPORT
---------- ----- ------------ -------------------- ---- ---------------------
PR0001     NON   North-West   24X7 electricity     no                     100
PR0002     Semi  South-East   Wooden Flooring      Yes                     35

SQL> select property_id, address, nullif(near_by_park,'no') from other_details
  2  natural join property;

PROPERTY_I ADDRESS                                  NULL
---------- ---------------------------------------- ----
PR0001     d-121, janakpuri, Delhi
PR0002     c-116, anand niketan, Delhi              Yes
```

Join query with order by clause

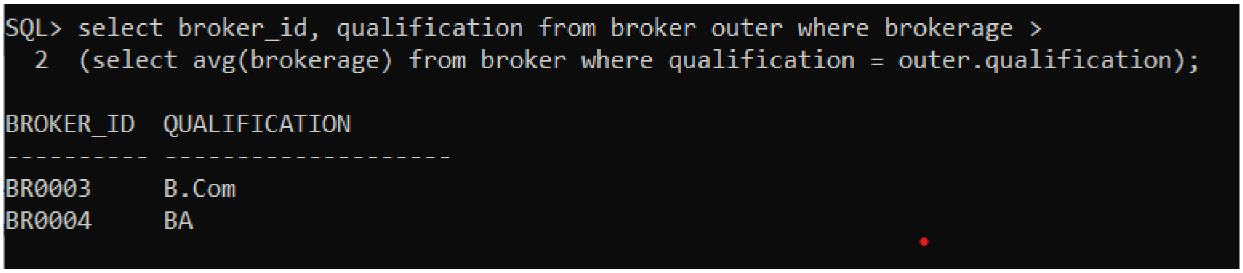
Q.7 Display owner details and property owned by them along with their value in ascending order of their price



```
SQL> select owner.*, property.property_id, property.total_value from
  2  (owner inner join owned_by on owner.owner_id=owned_by.owner_id)
  3  inner join property on owned_by.property_id=property.property_id
  4  order by total_value;

OWNER_ID   NAME                      PH_NO
---------- -------------------- ----------
ADDRESS                                  WORK_PROFILE    PROPERTY_I TOTAL_VALUE
---------------------------------------- --------------- ---------- -----------
OI0001     Rishabh              9898989898
A-1, Rajiv Road, Goa                     Engineer        PR0001         1000000

OI0002     Rajesh               8787878787
D-11, Sai Road, Delhi                    Doctor          PR0002         2000000
```

Correlated Query

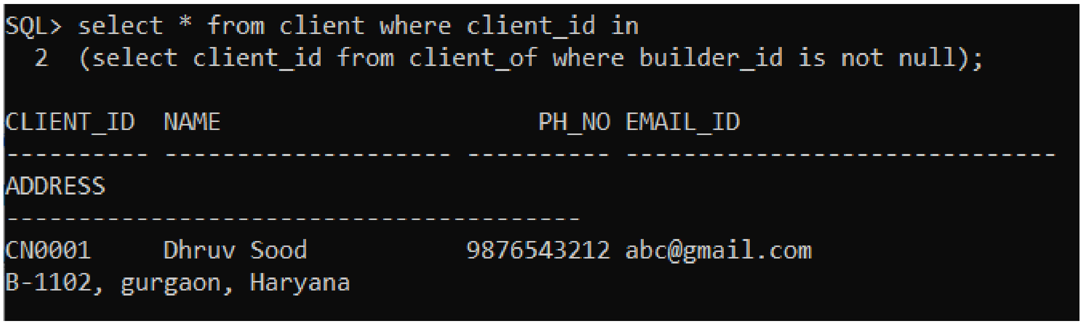
Q.8 Display broker id and qualification of brokers having brokerage higher than average brokerage of the brokers.



```
SQL> select broker_id, qualification from broker outer where brokerage >
  2  (select avg(brokerage) from broker where qualification = outer.qualification);

BROKER_ID  QUALIFICATION
---------- --------------------
BR0003     B.Com
BR0004     BA
```

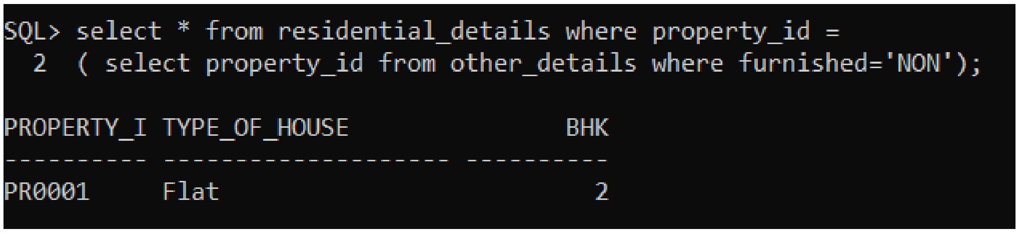
Uncorrelated Query

Q.9 Display client details who are associated with the builder, to purchase the property.



```
SQL> select * from client where client_id in
  2  (select client_id from client_of where builder_id is not null);

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS
----------------------------------------
CN0001     Dhruv Sood           9876543212 abc@gmail.com
B-1102, gurgaon, Haryana
```

Q.10 Display residential properties which are unfurnished.

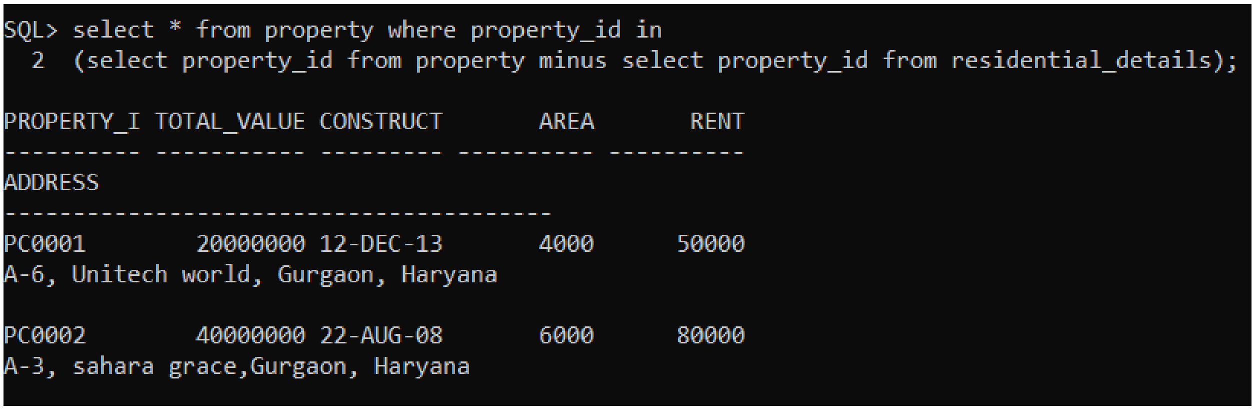


```
SQL> select * from residential_details where property_id =
  2  ( select property_id from other_details where furnished='NON');

PROPERTY_I TYPE_OF_HOUSE               BHK
---------- -------------------- ----------
PR0001     Flat                          2
```

Set Operation

Minus

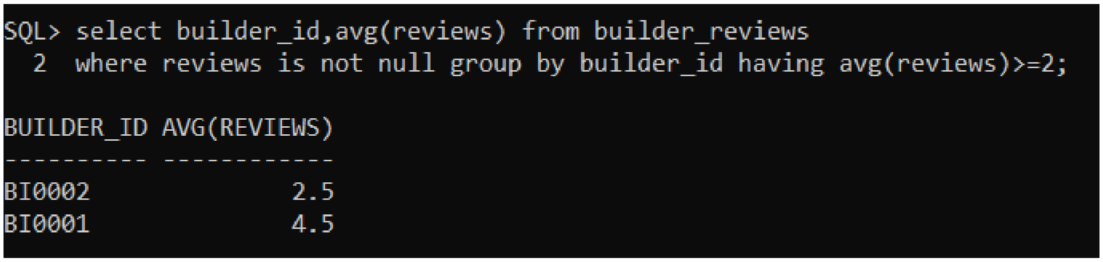
Q.11 Display details of commercial property only.



```
SQL> select * from property where property_id in
  2  (select property_id from property minus select property_id from residential_details);

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ---------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0001       20000000 12-DEC-13       4000      50000
A-6, Unitech world, Gurgaon, Haryana

PC0002       40000000 22-AUG-08       6000      80000
A-3, sahara grace,Gurgaon, Haryana
```

Group by, having and where clause

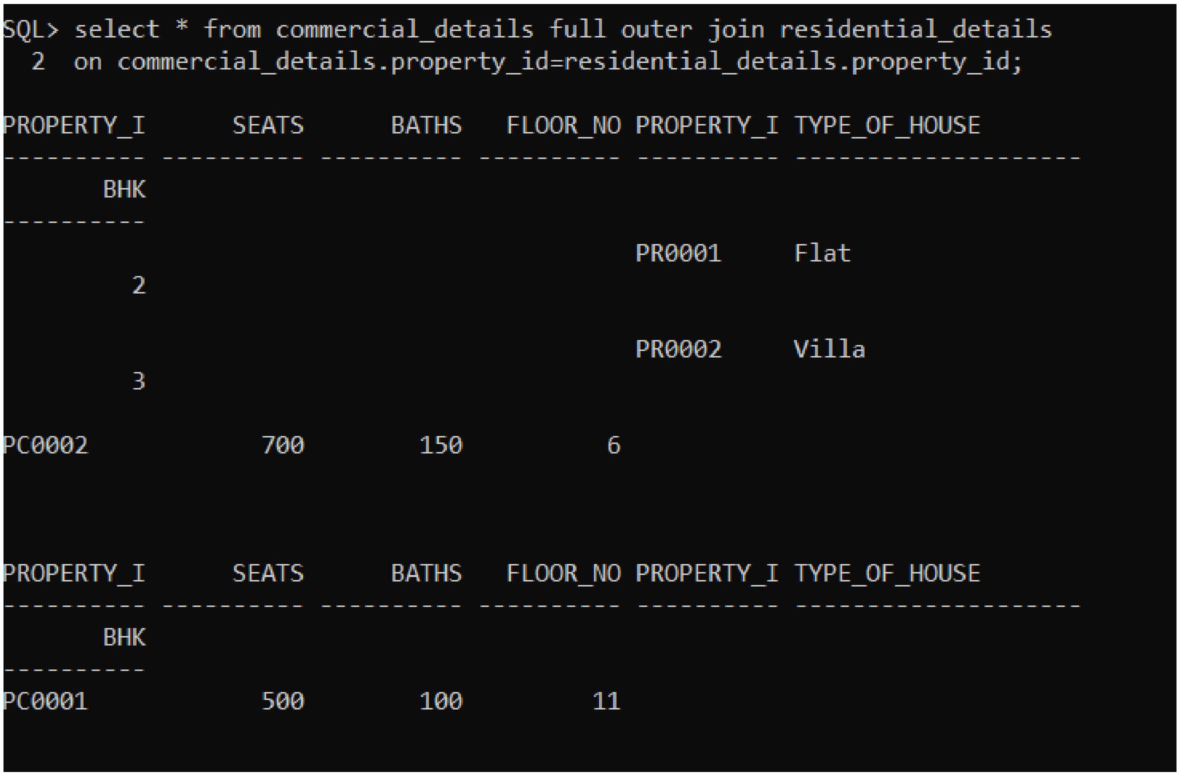
Q.12 Display average reviews of the builder with average review greater than 2 and review not null.



```
SQL> select builder_id,avg(reviews) from builder_reviews
  2  where reviews is not null group by builder_id having avg(reviews)>=2;

BUILDER_ID AVG(REVIEWS)
---------- ------------
BI0002              2.5
BI0001              4.5
```

Full Outer Join

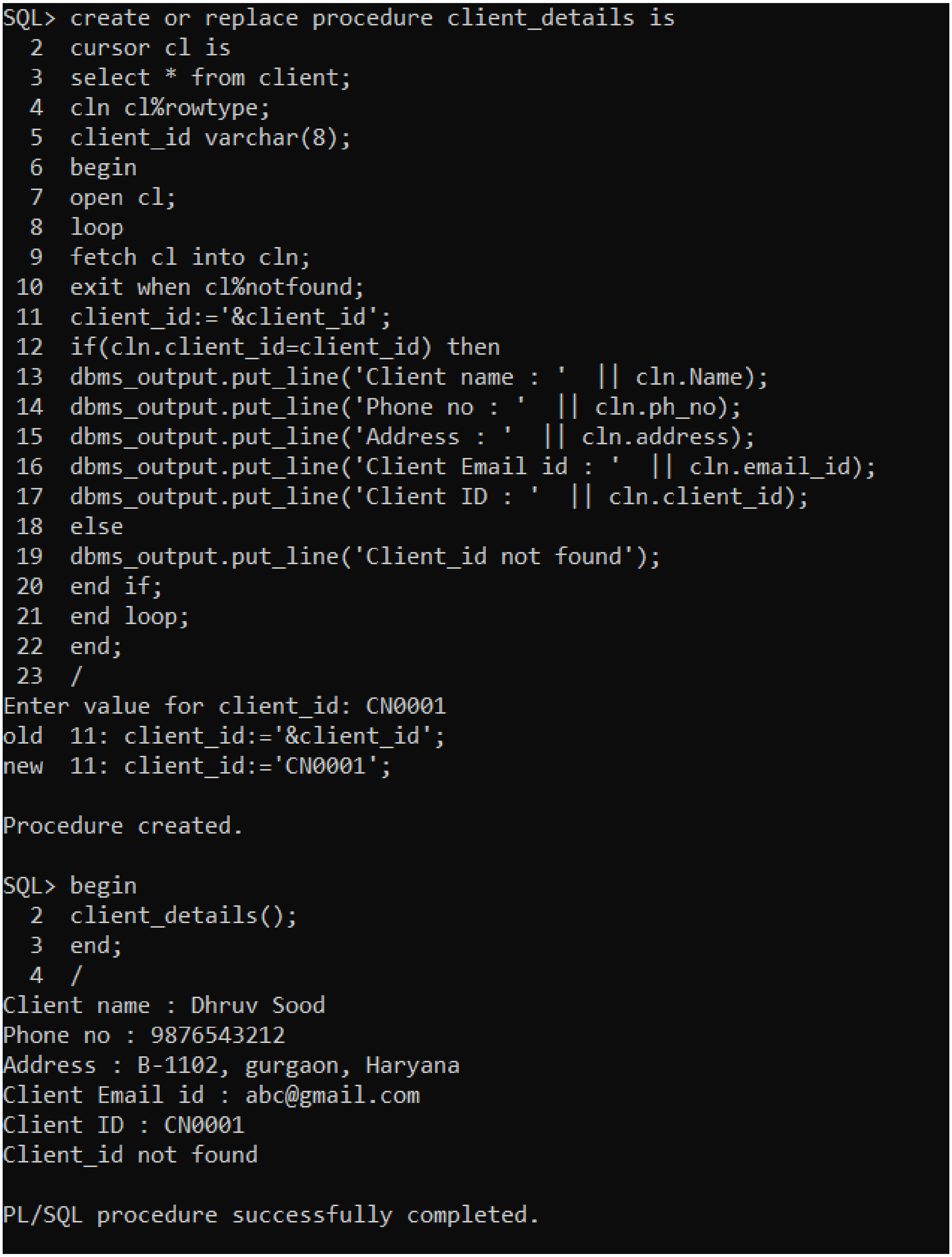
Q.13 Display commercial details along with residential details.



```
SQL> select * from commercial_details full outer join residential_details
  2  on commercial_details.property_id=residential_details.property_id;

PROPERTY_I      SEATS      BATHS   FLOOR_NO PROPERTY_I TYPE_OF_HOUSE
---------- ---------- ---------- ---------- ---------- --------------------
       BHK
----------
                                            PR0001     Flat
         2

                                            PR0002     Villa
         3

PC0002            700        150          6


PROPERTY_I      SEATS      BATHS   FLOOR_NO PROPERTY_I TYPE_OF_HOUSE
---------- ---------- ---------- ---------- ---------- --------------------
       BHK
----------
PC0001            500        100         11
```

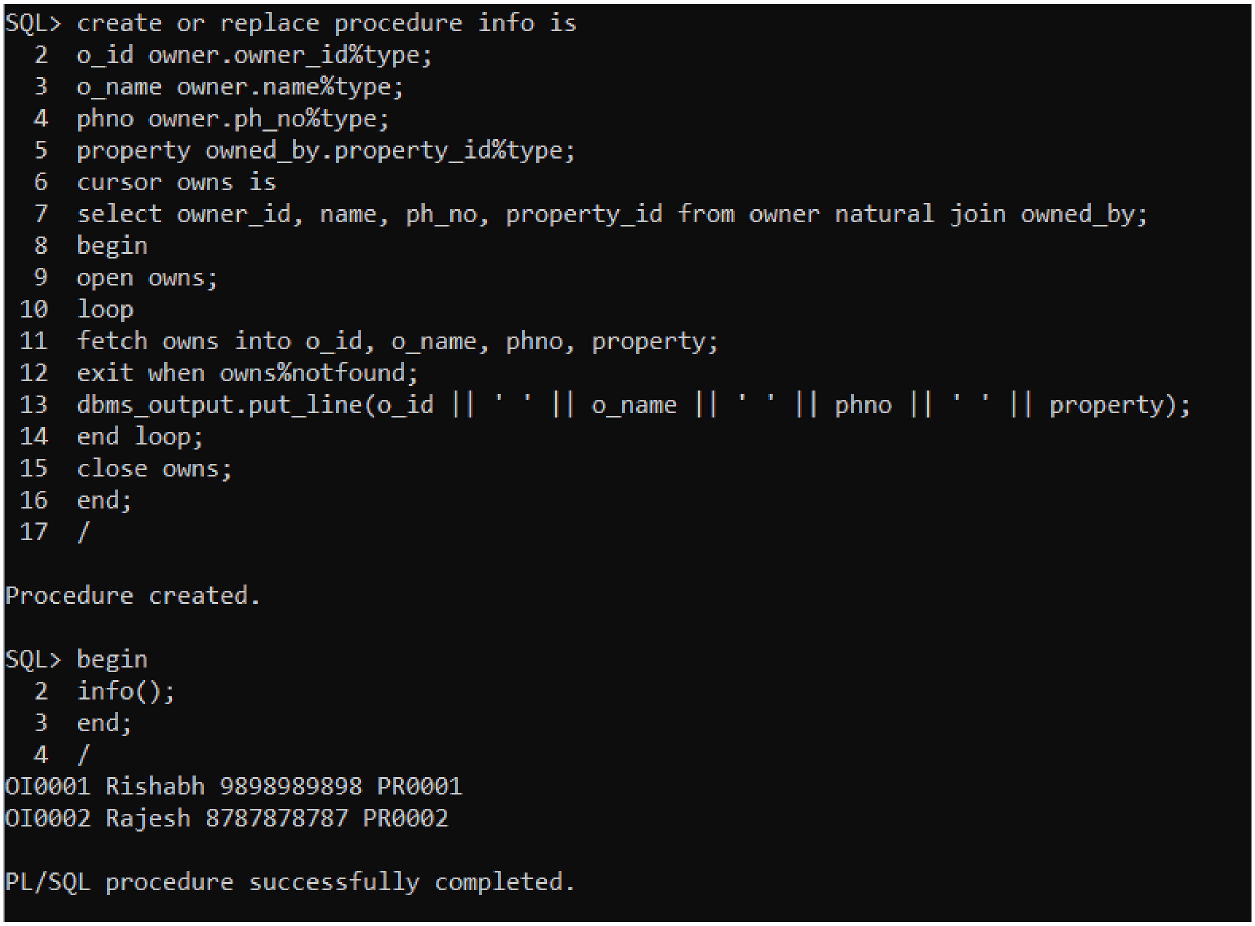
Procedures using cursor

1. If user wishes to fetch all the data associated with the client using his client id.



```
SQL> create or replace procedure client_details is
  2  cursor cl is
  3  select * from client;
  4  cln cl%rowtype;
  5  client_id varchar(8);
  6  begin
  7  open cl;
  8  loop
  9  fetch cl into cln;
 10  exit when cl%notfound;
 11  client_id:='&client_id';
 12  if(cln.client_id=client_id) then
 13  dbms_output.put_line('Client name : '  || cln.Name);
 14  dbms_output.put_line('Phone no : '  || cln.ph_no);
 15  dbms_output.put_line('Address : '  || cln.address);
 16  dbms_output.put_line('Client Email id : '  || cln.email_id);
 17  dbms_output.put_line('Client ID : '  || cln.client_id);
 18  else
 19  dbms_output.put_line('Client_id not found');
 20  end if;
 21  end loop;
 22  end;
 23  /
Enter value for client_id: CN0001
old  11: client_id:='&client_id';
new  11: client_id:='CN0001';

Procedure created.

SQL> begin
  2  client_details();
  3  end;
  4  /
Client name : Dhruv Sood
Phone no : 9876543212
Address : B-1102, gurgaon, Haryana
Client Email id : abc@gmail.com
Client ID : CN0001
Client_id not found

PL/SQL procedure successfully completed.
```

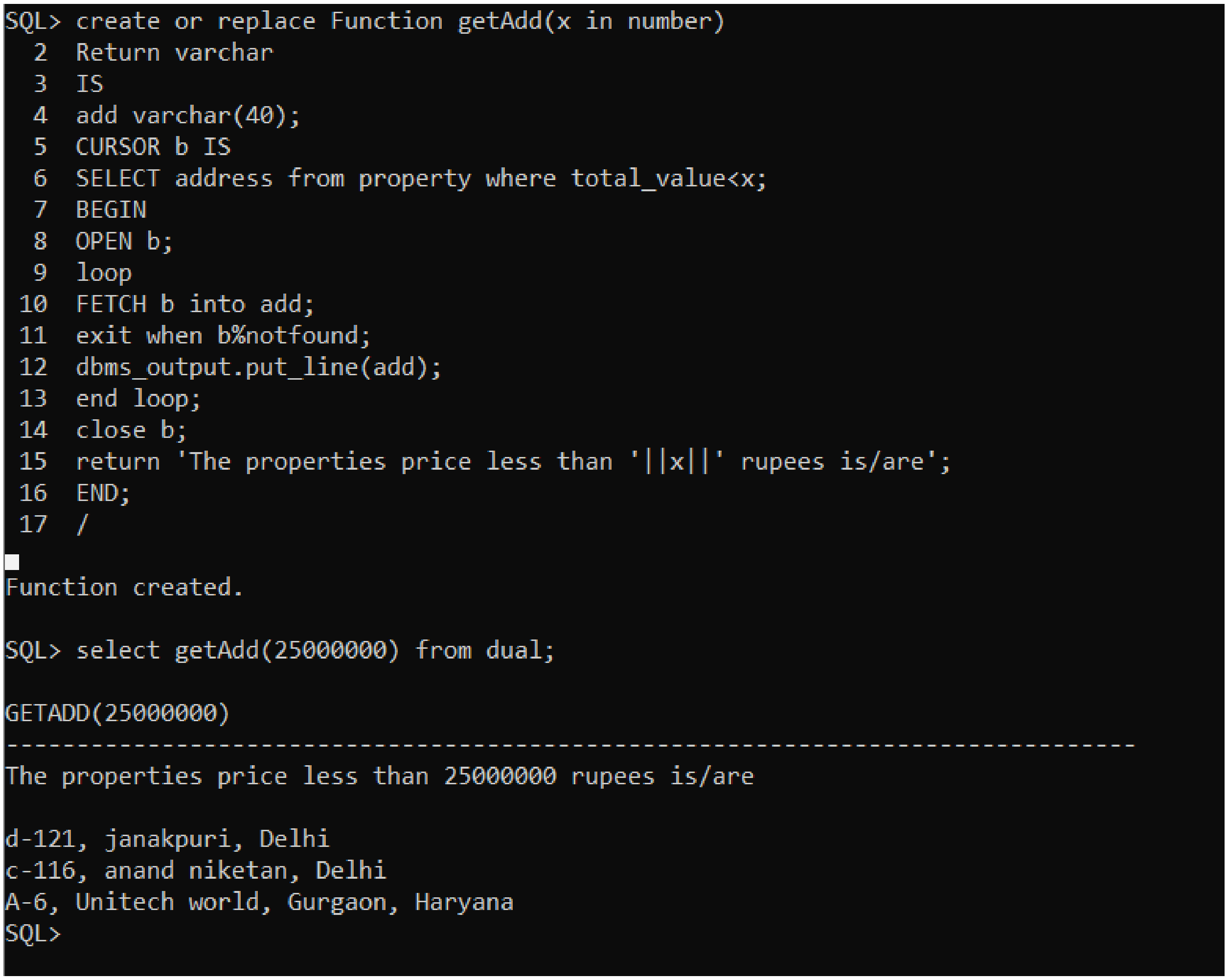
2. Consider a PL/SQl procedure to display owner details along with the property id using natural join on table owner and owned by.



```
SQL> create or replace procedure info is
o_id owner.owner_id%type;
o_name owner.name%type;
phno owner.ph_no%type;
property owned_by.property_id%type;
cursor owns is
select owner_id, name, ph_no, property_id from owner natural join owned_by;
begin
open owns;
loop
fetch owns into o_id, o_name, phno, property;
exit when owns%notfound;
dbms_output.put_line(o_id || ' ' || o_name || ' ' || phno || ' ' || property);
end loop;
close owns;
end;
/

Procedure created.

SQL> begin
info();
end;
/
OI0001 Rishabh 9898989898 PR0001
OI0002 Rajesh 8787878787 PR0002

PL/SQL procedure successfully completed.
```

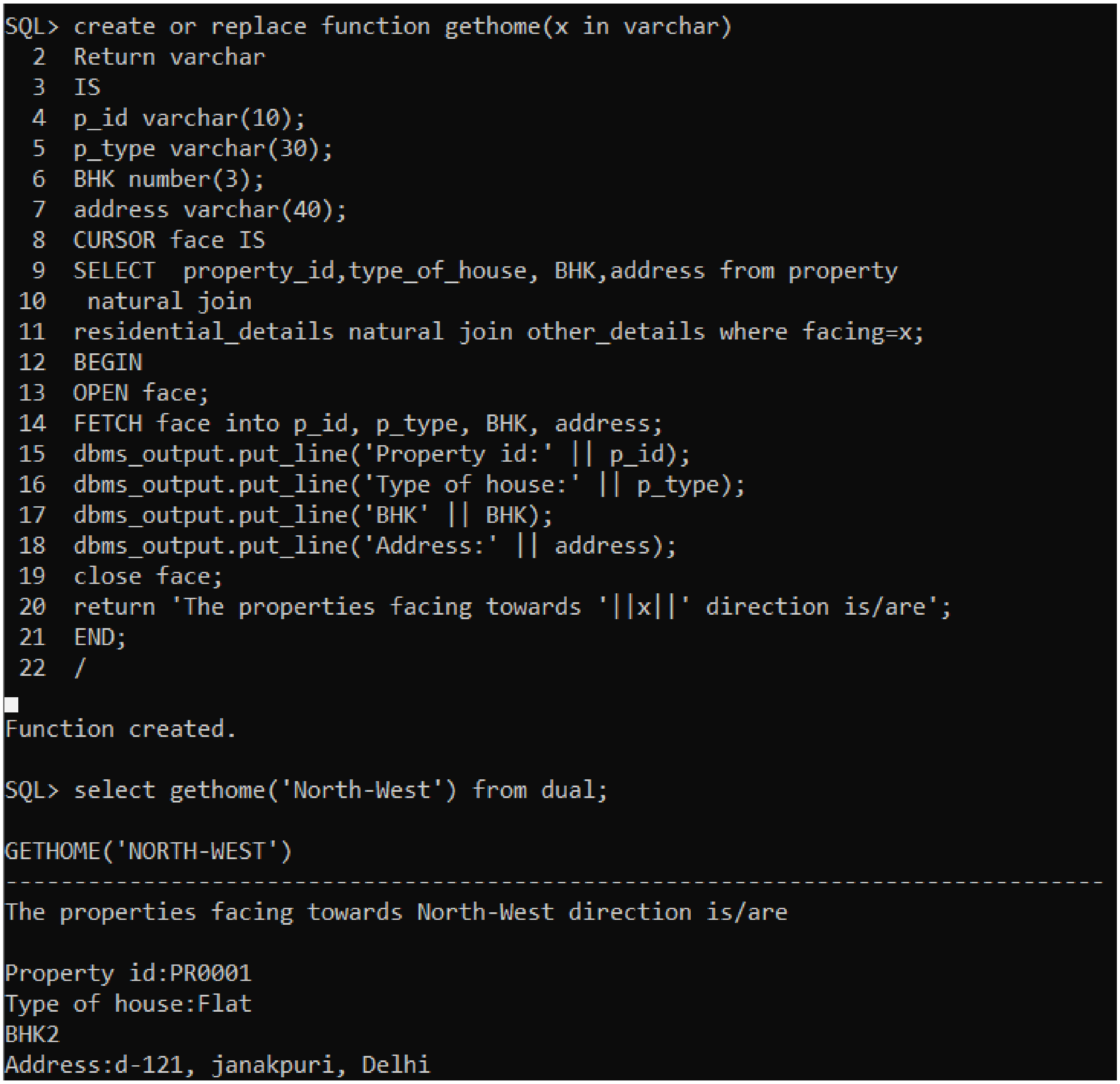
Functions using cursor

1. A function in PL/SQL using cursor to find property according to the client's budget.



```
SQL> create or replace Function getAdd(x in number)
Return varchar
IS
add varchar(40);
CURSOR b IS
SELECT address from property where total_value<x;
BEGIN
OPEN b;
loop
FETCH b into add;
exit when b%notfound;
dbms_output.put_line(add);
end loop;
close b;
return 'The properties price less than '||x||' rupees is/are';
END;
/

Function created.

SQL> select getAdd(25000000) from dual;

GETADD(25000000)
--------------------------------------------------------------------------------
The properties price less than 25000000 rupees is/are

d-121, janakpuri, Delhi
c-116, anand niketan, Delhi
A-6, Unitech world, Gurgaon, Haryana
SQL>
```

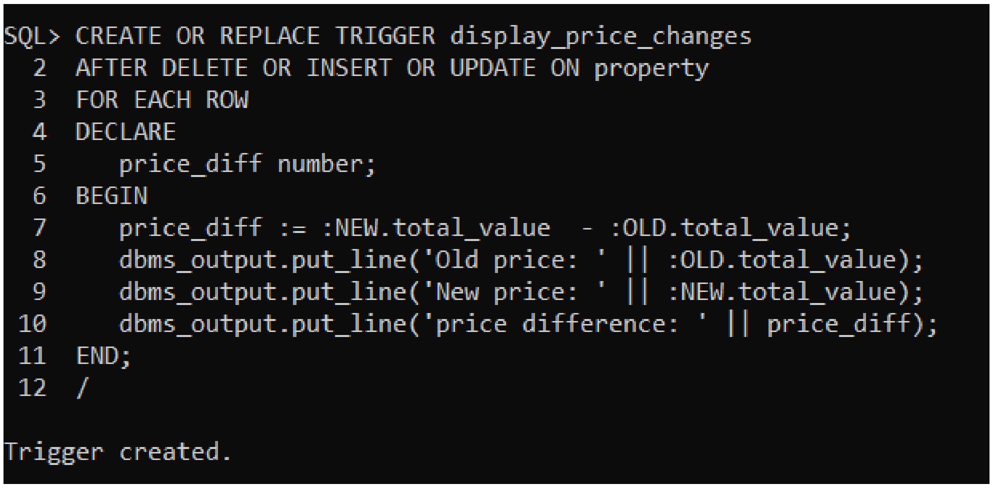
2. A function in PL/SQL using cursor to find a house according to it's facing(north facing, south facing etc).



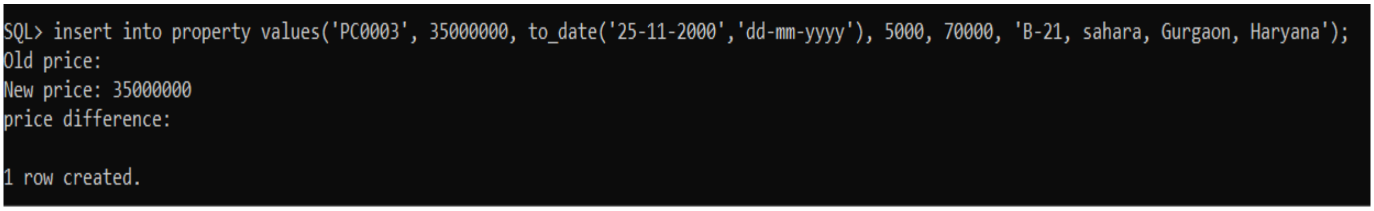
```
SQL> create or replace function gethome(x in varchar)
Return varchar
IS
p_id varchar(10);
p_type varchar(30);
BHK number(3);
address varchar(40);
CURSOR face IS
SELECT  property_id,type_of_house, BHK,address from property
 natural join
residential_details natural join other_details where facing=x;
BEGIN
OPEN face;
FETCH face into p_id, p_type, BHK, address;
dbms_output.put_line('Property id:' || p_id);
dbms_output.put_line('Type of house:' || p_type);
dbms_output.put_line('BHK' || BHK);
dbms_output.put_line('Address:' || address);
close face;
return 'The properties facing towards '||x||' direction is/are';
END;
/

Function created.

SQL> select gethome('North-West') from dual;

GETHOME('NORTH-WEST')
--------------------------------------------------------------------------------
The properties facing towards North-West direction is/are

Property id:PR0001
Type of house:Flat
BHK2
Address:d-121, janakpuri, Delhi
```

Tiggers

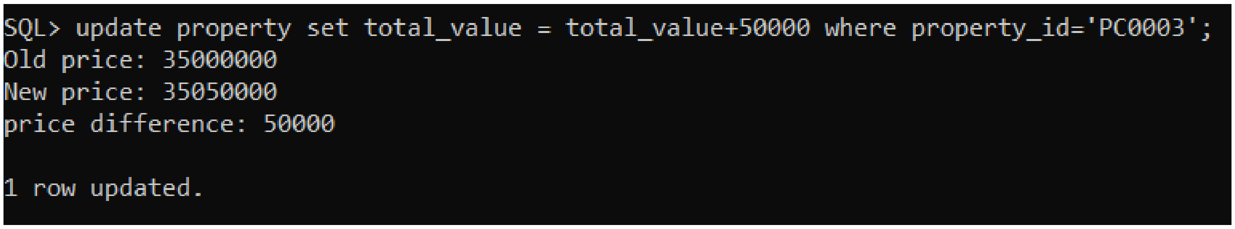
1. Whenever the property's price changes display the new and the old price along with their difference.



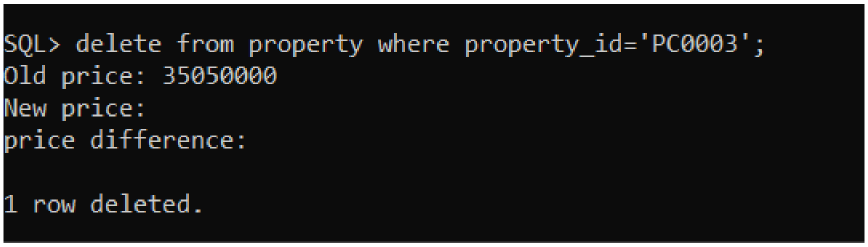
```
SQL> CREATE OR REPLACE TRIGGER display_price_changes
AFTER DELETE OR INSERT OR UPDATE ON property
FOR EACH ROW
DECLARE
   price_diff number;
BEGIN
   price_diff := :NEW.total_value  - :OLD.total_value;
   dbms_output.put_line('Old price: ' || :OLD.total_value);
   dbms_output.put_line('New price: ' || :NEW.total_value);
   dbms_output.put_line('price difference: ' || price_diff);
END;
/

Trigger created.
```



```
SQL> insert into property values('PC0003', 35000000, to_date('25-11-2000','dd-mm-yyyy'), 5000, 70000, 'B-21, sahara, Gurgaon, Haryana');
Old price:
New price: 35000000
price difference:

1 row created.
```

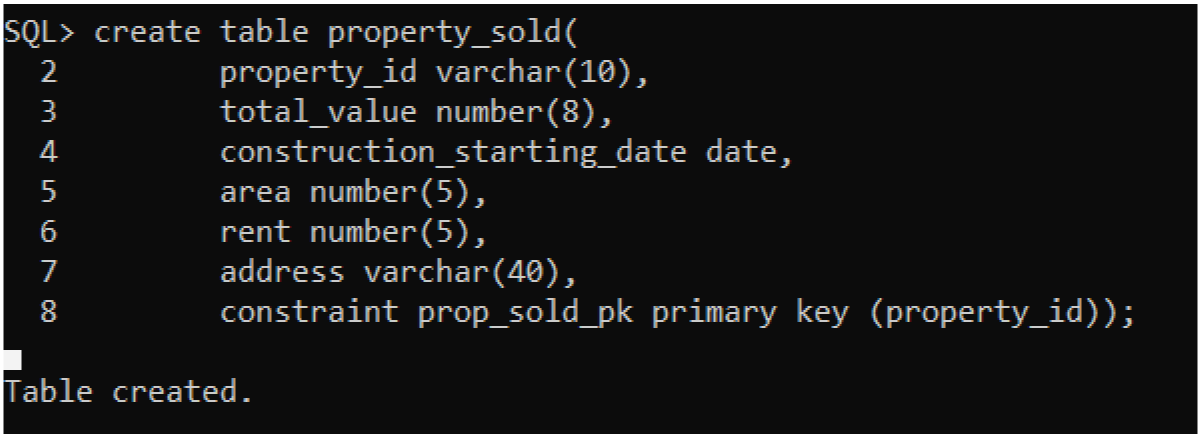


```
SQL> update property set total_value = total_value+50000 where property_id='PC0003';
Old price: 35000000
New price: 35050000
price difference: 50000

1 row updated.
```

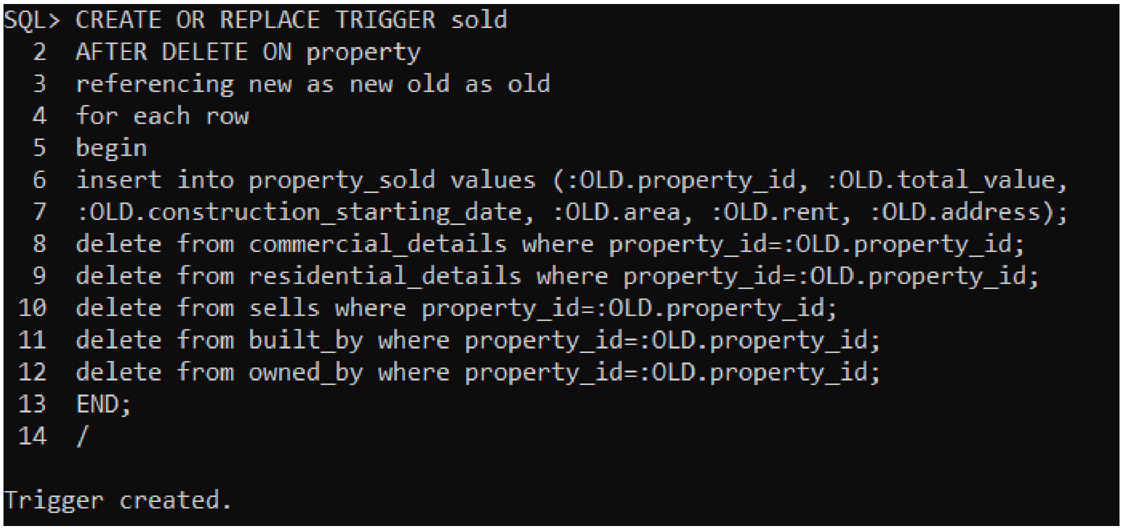


```
SQL> delete from property where property_id='PC0003';
Old price: 35050000
New price:
price difference:

1 row deleted.
```

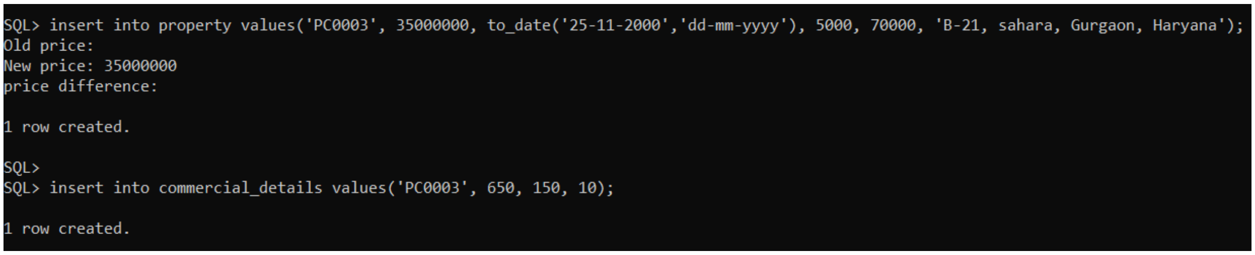
2. Whenever a property is sold, store its details in property sold table and delete all the current stored data of that property;



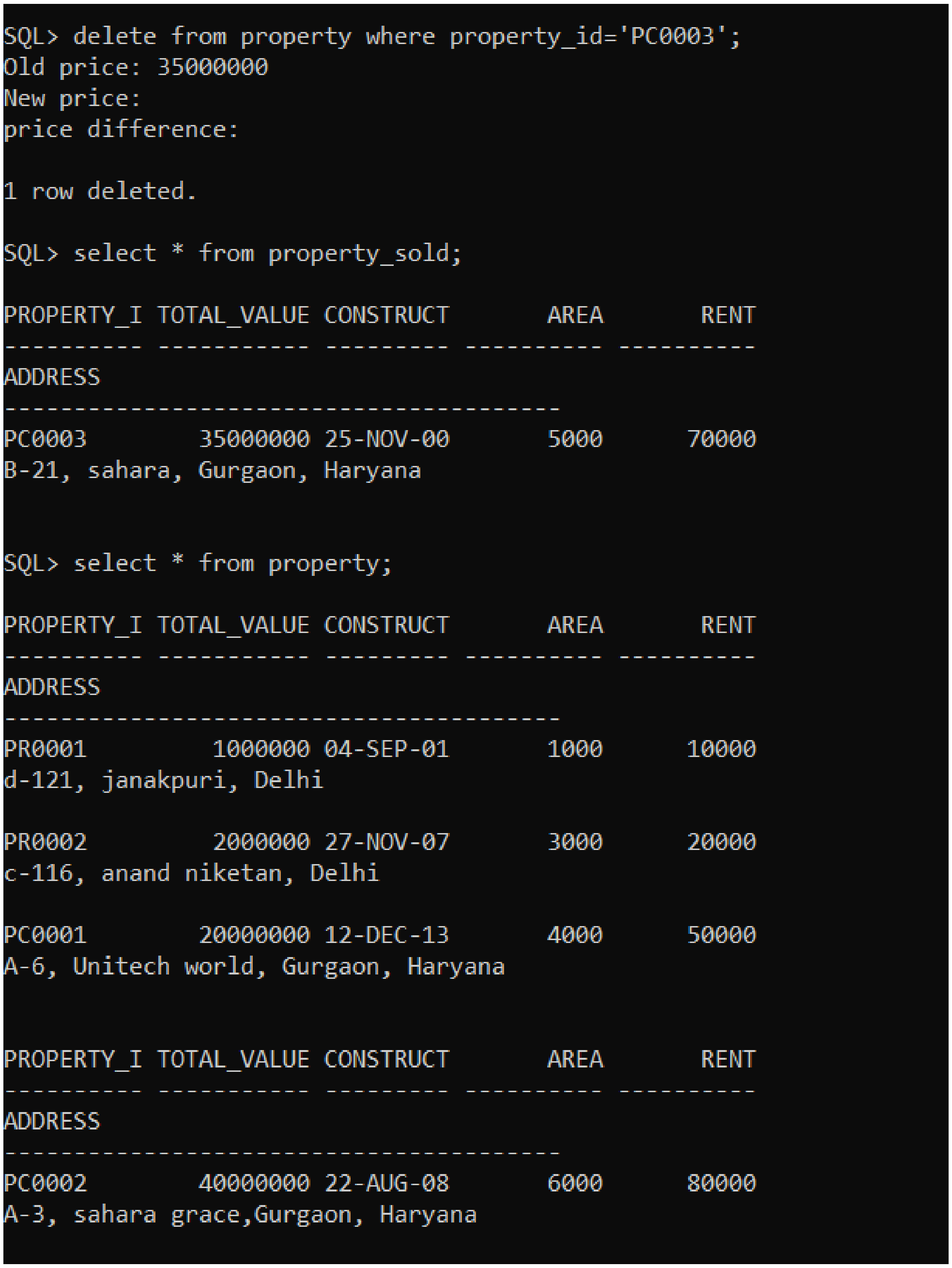
```
SQL> create table property_sold(
          property_id varchar(10),
          total_value number(8),
          construction_starting_date date,
          area number(5),
          rent number(5),
          address varchar(40),
          constraint prop_sold_pk primary key (property_id));

Table created.
```



```
SQL> CREATE OR REPLACE TRIGGER sold
     AFTER DELETE ON property
     referencing new as new old as old
     for each row
     begin
     insert into property_sold values (:OLD.property_id, :OLD.total_value,
     :OLD.construction_starting_date, :OLD.area, :OLD.rent, :OLD.address);
     delete from commercial_details where property_id=:OLD.property_id;
     delete from residential_details where property_id=:OLD.property_id;
     delete from sells where property_id=:OLD.property_id;
     delete from built_by where property_id=:OLD.property_id;
     delete from owned_by where property_id=:OLD.property_id;
     END;
     /

Trigger created.
```

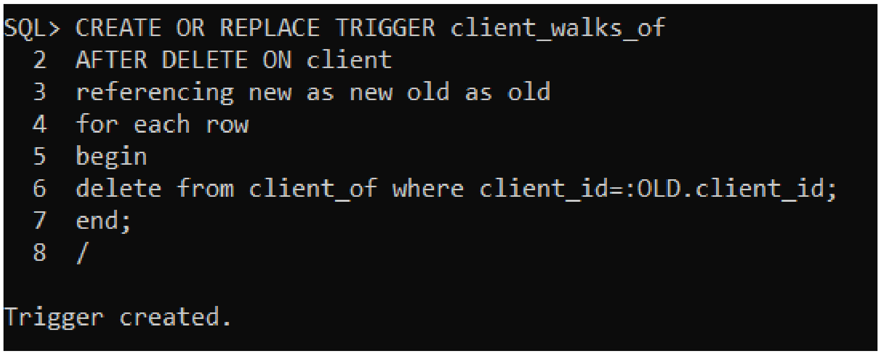


```
SQL> insert into property values('PC0003', 35000000, to_date('25-11-2000','dd-mm-yyyy'), 5000, 70000, 'B-21, sahara, Gurgaon, Haryana');
Old price:
New price: 35000000
price difference:

1 row created.

SQL>
SQL> insert into commercial_details values('PC0003', 650, 150, 10);

1 row created.
```

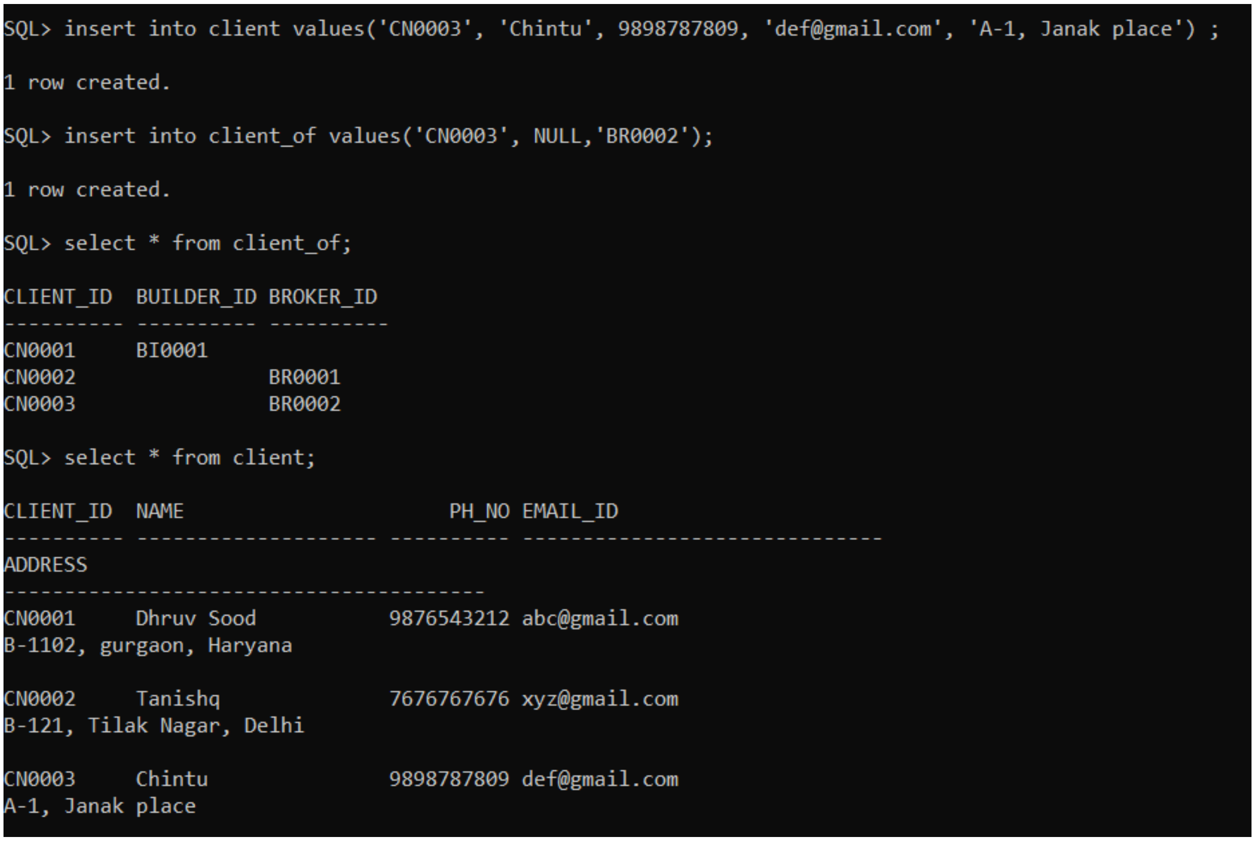


```
SQL> delete from property where property_id='PC0003';
Old price: 35000000
New price:
price difference:

1 row deleted.

SQL> select * from property_sold;

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0003        35000000 25-NOV-00       5000      70000
B-21, sahara, Gurgaon, Haryana


SQL> select * from property;

PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PR0001         1000000 04-SEP-01       1000      10000
d-121, janakpuri, Delhi

PR0002         2000000 27-NOV-07       3000      20000
c-116, anand niketan, Delhi

PC0001        20000000 12-DEC-13       4000      50000
A-6, Unitech world, Gurgaon, Haryana


PROPERTY_I TOTAL_VALUE CONSTRUCT       AREA       RENT
---------- ----------- --------- ---------- ----------
ADDRESS
----------------------------------------
PC0002        40000000 22-AUG-08       6000      80000
A-3, sahara grace,Gurgaon, Haryana
```

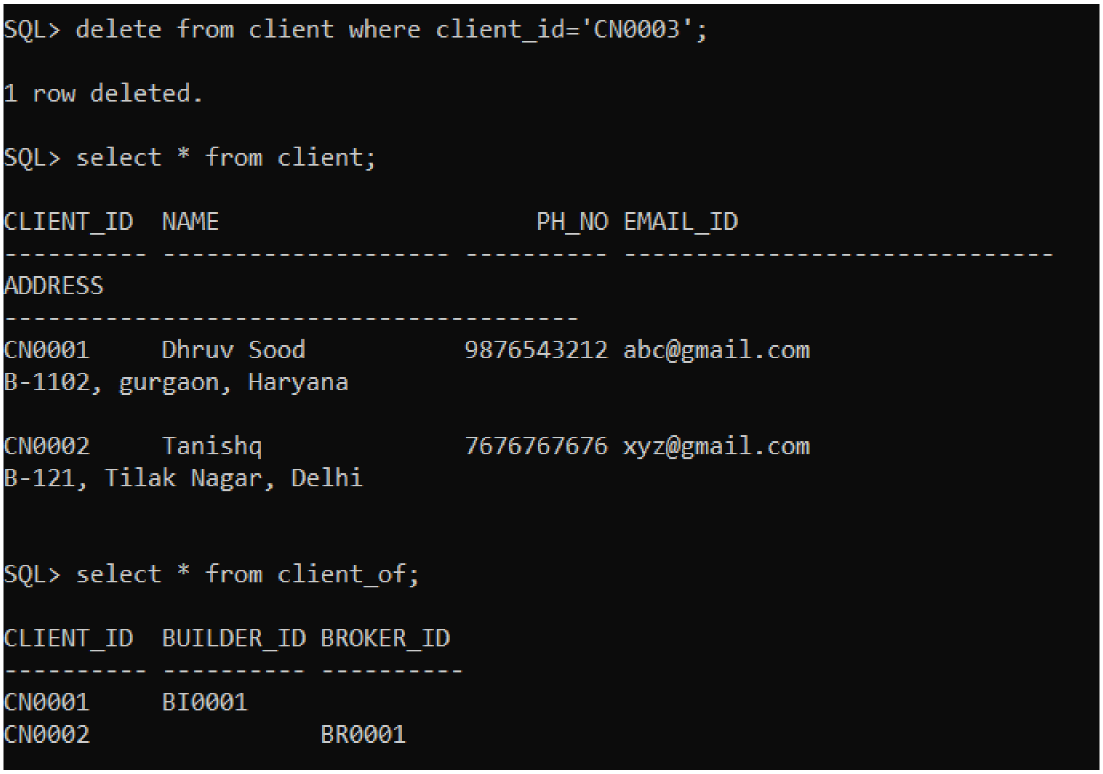
3. If the client walks of at anytime, then all his stored data in the database is to be deleted.



```
SQL> CREATE OR REPLACE TRIGGER client_walks_of
     AFTER DELETE ON client
     referencing new as new old as old
     for each row
     begin
     delete from client_of where client_id=:OLD.client_id;
     end;
     /

Trigger created.
```



```
SQL> insert into client values('CN0003', 'Chintu', 9898787809, 'def@gmail.com', 'A-1, Janak place') ;

1 row created.

SQL> insert into client_of values('CN0003', NULL,'BR0002');

1 row created.

SQL> select * from client_of;

CLIENT_ID  BUILDER_ID BROKER_ID
---------- ---------- ----------
CN0001     BI0001
CN0002                BR0001
CN0003                BR0002

SQL> select * from client;

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS
----------------------------------------
CN0001     Dhruv Sood           9876543212 abc@gmail.com
B-1102, gurgaon, Haryana

CN0002     Tanishq              7676767676 xyz@gmail.com
B-121, Tilak Nagar, Delhi

CN0003     Chintu               9898787809 def@gmail.com
A-1, Janak place
```



```
SQL> delete from client where client_id='CN0003';

1 row deleted.

SQL> select * from client;

CLIENT_ID  NAME                      PH_NO EMAIL_ID
---------- -------------------- ---------- ------------------------------
ADDRESS
----------------------------------------
CN0001     Dhruv Sood           9876543212 abc@gmail.com
B-1102, gurgaon, Haryana

CN0002     Tanishq              7676767676 xyz@gmail.com
B-121, Tilak Nagar, Delhi


SQL> select * from client_of;

CLIENT_ID  BUILDER_ID BROKER_ID
---------- ---------- ----------
CN0001     BI0001
CN0002                BR0001
```