VIT®

Vellore Institute of Technology

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

DATABASE MANAGEMENT SYSTEM PROJECT

TOPIC: GYM MANAGEMENT SYSTEM

SUBMITTED TO: PROF. BIMAL KUMAR RAY

SUBMITTED BY:

Aayush Samaiyar - 19BIT0057

Dhruvil Dave - 19BIT0065

Deepak Kumar - 19BIT0154

UNIVERSE OF DISCOURSE

Our project entitled "GYM MANAGEMENT SYSTEM", is a database system that provides hassle-free supervision of the gym. It maintains the account of the owner, managers, trainers, and gym members.

For each member, who enrolls in the gym a trainer is assigned. The manager examines the membership details and also intimates them for their payment. The owner runs the gym and also looks for the maintenance of equipment and hires managers who provide trainer to the members.

ENTITIES

- Owner
- Gym
- Room
- Trainer
- Member
- Manager
- Membership_Plan
- Equipment
- Payment

ENTITY	ATTRIBUTES
Owner	owner_id(pk), owner_name, owner_phone, owner_email, owner_address
Gym	gym_id(pk), gym_name, gym_address, gym_landmark
Manager	manager_id(pk), manager_name, manager_email, manager_phone, manager_gender
Room	room_id(pk), room_floor, room_type, room_no
Equipment	equipment_id (pk) , equipment_name, equipment_desc, equipment_puchase_date
Trainer	trainer_id (pk), trainer_name, trainer_email, trainer_age, trainer_phone, trainer_speciality
Member	member_id (pk), member_name, member_age, member_phone, member_email, member_gender.
Payment	payment_id (pk), payment_date, payment_amount, payment_mode(multivalued), payment_month
Membership_Plan	plan_id (pk), plan_tenure, plan_discount, signup_fee, plan_name

Data Requirements:

Consider a Gym Database where data about the gym is stored. The data requirements will be as follows

Owner of the Gym is identified by his owner id, every owner has a name, email, phone and address.

Owner owns the gym, one owner might have many gyms and one gym might have many owners. Also a gym must have a owner

Gym is identified by gym id, Every gym has a name, an address and a landmark name.

Every Gym has different rooms. Rooms are identified by room_id, every room has a floor_number, room_type and a room number.

Every gym should have a room and one gym might have more than one room.

Every room has a particular number of trainers who train the members of the gym. Trainer is identified by Trainer id, every trainer has Name, Age, email, phone number and speciality of the trainer. One room can have many trainers and one trainer can work in more than one room. Every room should have a trainer.

Trainer trains members of the gym. Members are identified by Member id, it also contains name, email, phone number, age and gender of the member. Every member has one trainer and a Trainer may train more than one member. But a member must have a trainer

Every member belongs to a particular membership plan, Membership plan is identified by plan id, every membership plan has a discount, membership tenure, plan name and signup fee.

A particular membership plan can be taken by many members and a member has to compulsorily take a membership plan.

Members make payment, Every payment is identified by payment id, every payment has a Date of payment, amount of payment, mode of payment and pay month.

One member can make more than one payment but payment should be made by the member.

Furthermore, Owner appoints a manager to the gym, the manager is identified by manager id, every manager has the name, email, phone number and gender of the manager. One owner might assign more than one manager, and a manager must be appointed by the owner.

Manager Manages Gym, One manager might manage more than one gym and every gym can have more than one manager. A gym must have a manager.

Manager receives payment, one manager might receive more than one payment. But payment is compulsorily received by a manager.

Also every room contains equipment, each equipment is identified by Equipment id, every equipment has name, description and purchase date. One room might have more than one equipment, and equipment must be contained in a room.

Relationships and Cardinality

- Owner OWNS Gym (M : N)
- Owner APPOINTS Manager(1 : M)
- Manager MANAGES Gym (M : N)
- Gym HAS Room (1 : M)
- Room HAS Trainer (M : N)
- Trainer TRAINS Member (1 : M)
- Member BELONGS Membership plan (M : 1)
- Room CONTAINS Equipments (1 : M)
- Member MAKES Payment (1 : M)
- Manager RECEIVES Payment (1 : M)

FUNCTIONAL REQUIREMENTS: (RETRIEVAL OF DATA)

1. Customers can look at the type of membership plans.
2. Customers can find the nearest gym by searching the gym address.
3. Customers can see the details of managers and trainers.
4. Customers can see their own details and membership plan details from the database.
5. Owners can see the payment description of their customers.
6. Customers/trainers can see the equipment present in the gym.
7. Trainer can see their assigned room and members.
8. Owner can have a check of which gym is managed by which manager.

FUNCTIONAL REQUIREMENTS: (MODIFICATION OF DATA)

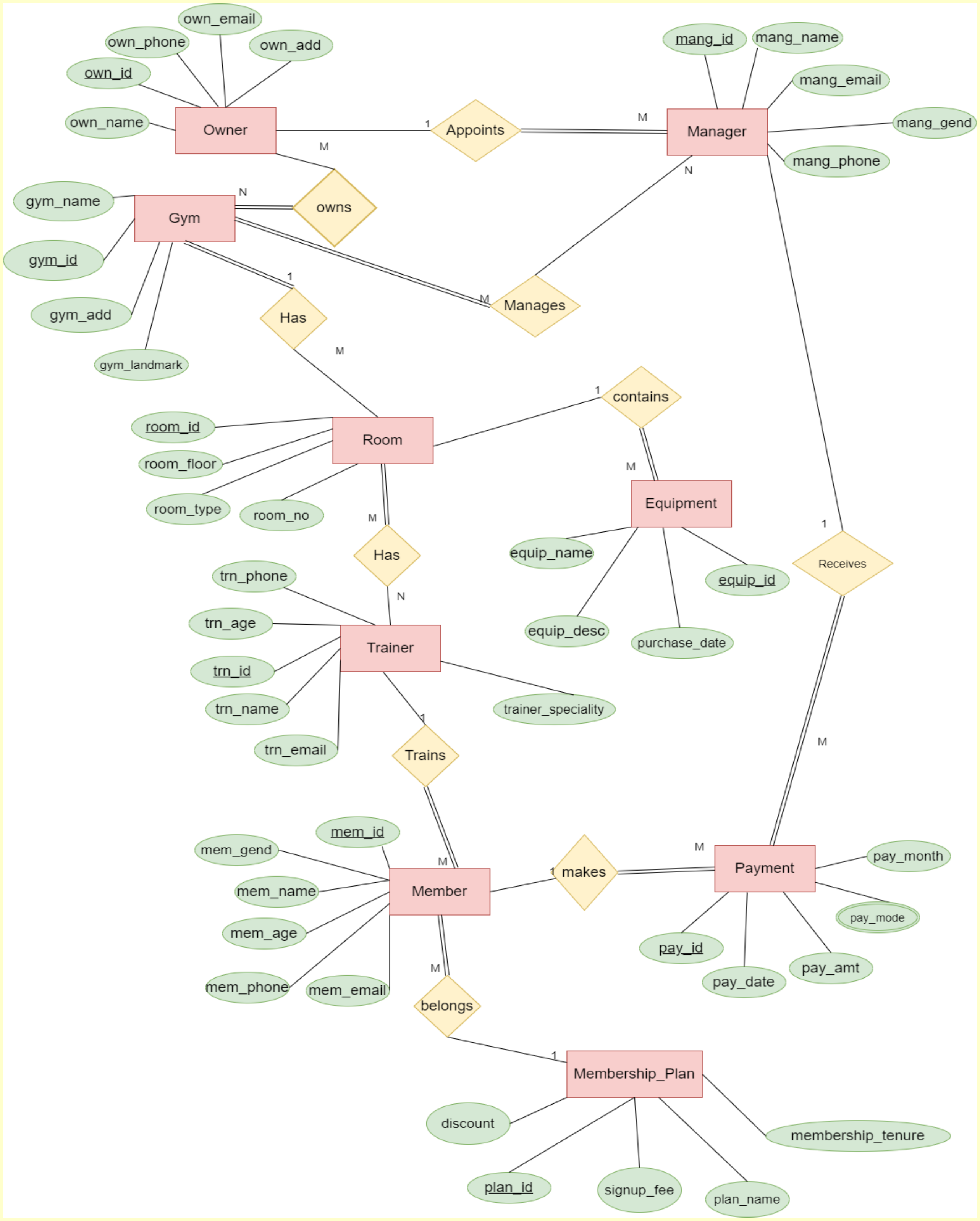
1. Owners can change the manager and trainer.
2. Manager can change the allotted rooms for the trainer.
3. Customers can change their membership plans.
4. Customers can change their personal details and mode of payment.
5. Customers can ask to change their assigned trainer.
6. Manager can add or replace the equipment.

FUNCTIONAL REQUIREMENTS: (DELETION OF DATA)

1. Owner can remove the manager.
2. Customers can cancel their membership.
3. Owner/Manager can remove any discount on membership plans.

4. Manager can change the equipment in the room.

ER DIAGRAM:


own_email
own_phone
own_add
own_id
own_name
Owner
1
Appoints
M
Manager
mang_id
mang_name
mang_email
mang_gend
mang_phone
M
N
gym_name
Gym
owns
gym_id
gym_add
gym_landmark
1
Has
M
Manages
N
M
contains
1
room_id
Room
room_floor
room_type
room_no
M
Equipment
M
Has
N
equip_name
equip_id
equip_desc
purchase_date
1
Receives
trn_phone
trn_age
Trainer
trn_id
trn_name
trn_email
trainer_speciality
1
Trains
M
mem_id
mem_gend
mem_name
Member
mem_age
mem_phone
mem_email
M
1
makes
M
Payment
pay_month
pay_mode
pay_id
pay_date
pay_amt
M
belongs
1
Membership_Plan
discount
membership_tenure
plan_id
signup_fee
plan_name

RELATIONAL DATABASE SCHEMA

OWNER

own_id	own_name	own_phone	own_email	own_add

GYM

gym_id	gym_name	gym_add	gym_landmark

MANAGER

mang_id	mang_name	mang_email	mang_gender	mang_phone	own_id

ROOM

room_id	room_floor	room_type	room_no	gym_id

EQUIPMENT

eq_id	eq_name	eq_desc	purchase_date	room_id

TRAINER

trn_id	trn_name	trn_age	trn_phone	trn_email	speciality

MEMBERSHIP_PLAN

plan_id	discount	signup_fee	plan_name	tenure

MEMBER

mem_id	mem_name	mem_gen	mem_age	mem_phone	mem_email	trn_id	plan_id

PAYMENT

pay_id	pay_date	pay_amt	pay_month	mem_id	mang_id

PAYMENT MODE

pay_id	pay_mode

OWNS

own_id	gym_id

MANAGES

gym_id	mang_id

HAS

room_id	trn_id

IMPLEMENTATION

CREATING TABLES WITH CONSTRAINTS:

1. **OWNER**

create table owner (own_id varchar(4) constraint pk_own primary key, own_name varchar(30), own_phone number(10), own_add varchar(100));

alter table owner add email varchar(50);

2. **GYM**

create table gym (gym_id varchar(4) constraint pk_gym primary key, gym_name varchar(20), gym_add varchar(100), gym_landmark varchar(20));

3. **MANAGER**

create table manager (mang_id varchar(4) constraint pk_manager primary key, mang_name varchar(20), mang_gender varchar(6), mang_phone number(10), own_id constraint fkown references owner);

alter table manager add mang_email varchar(50);

4. **ROOM**

create table room (room_id varchar(4) constraint pk_room primary key, room_floor varchar(10), room_type varchar(20), room_number number(4), gym_id constraint fkgym references gym);

5. **EQUIPMENT**

create table equipment (eq_id varchar(4) constraint pk_eq primary key, eq_name varchar(20), eq_desc varchar(100), purchase_date date, room_id constraint fkroom references room);

6. **TRAINER**

create table trainer (trn_id varchar(4) constraint pk_trn primary key, trn_name varchar(20), trn_age number(2), trn_phone number(10), trn_email varchar(50), speciality varchar(20));

7. **MEMBERSHIP_PLAN**

create table membership_plan (plan_id varchar(4) constraint pk_memplan primary key, discount varchar(3), signup_fee number(5), plan_name varchar(20), tenure varchar(20));

8. MEMBER

create table member (mem_id varchar(4) constraint pk_mem primary key, mem_name varchar(20), mem_gender varchar(8), mem_age number(2), mem_phone number(10), mem_email varchar(50), trn_id constraint fktrn references trainer, plan_id constraint fkmemplan references membership_plan);

9. PAYMENT

create table payment (pay_id varchar(4) constraint pk_pay primary key, pay_date date, pay_amt number(5), pay_month varchar(20), mem_id constraint fkmem references member, mang_id constraint fkmang references manager);

10. PAYMENT_MODE

create table payment_mode (pay_id varchar(4), pay_mode varchar(20), constraint pk_paymode primary key(pay_id,pay_mode));

alter table payment_mode add constraint fkpaym foreign key(pay_id) references payment;

11. OWNS

create table owns (own_id varchar(4), gym_id varchar(4), constraint pk_owns primary key(own_id, gym_id));

alter table owns add constraint fkowid foreign key(own_id) references owner;

alter table owns add constraint fkgymid foreign key(gym_id) references gym;

12. MANAGES

create table manages (gym_id varchar(4), mang_id varchar(4), constraint pk_manages primary key(gym_id, mang_id));

alter table manages add constraint fkmgid foreign key(mang_id) references manager;

alter table manages add constraint fkgmid foreign key(gym_id) references gym;

13. HAS

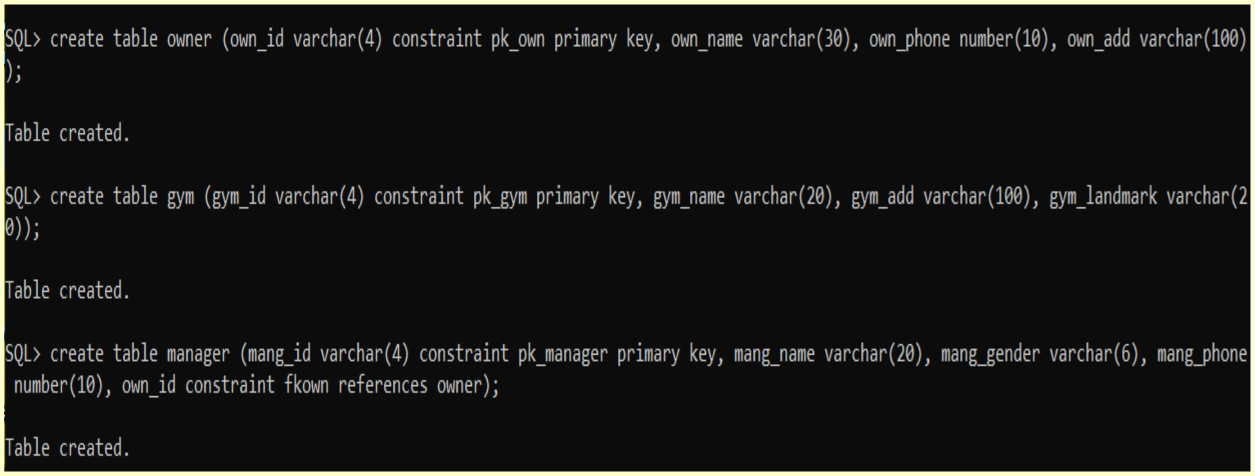
create table has (room_id varchar(4), trn_id varchar(4), constraint pk_has primary key(room_id, trn_id));

alter table has add constraint fktrid foreign key(trn_id) references trainer;

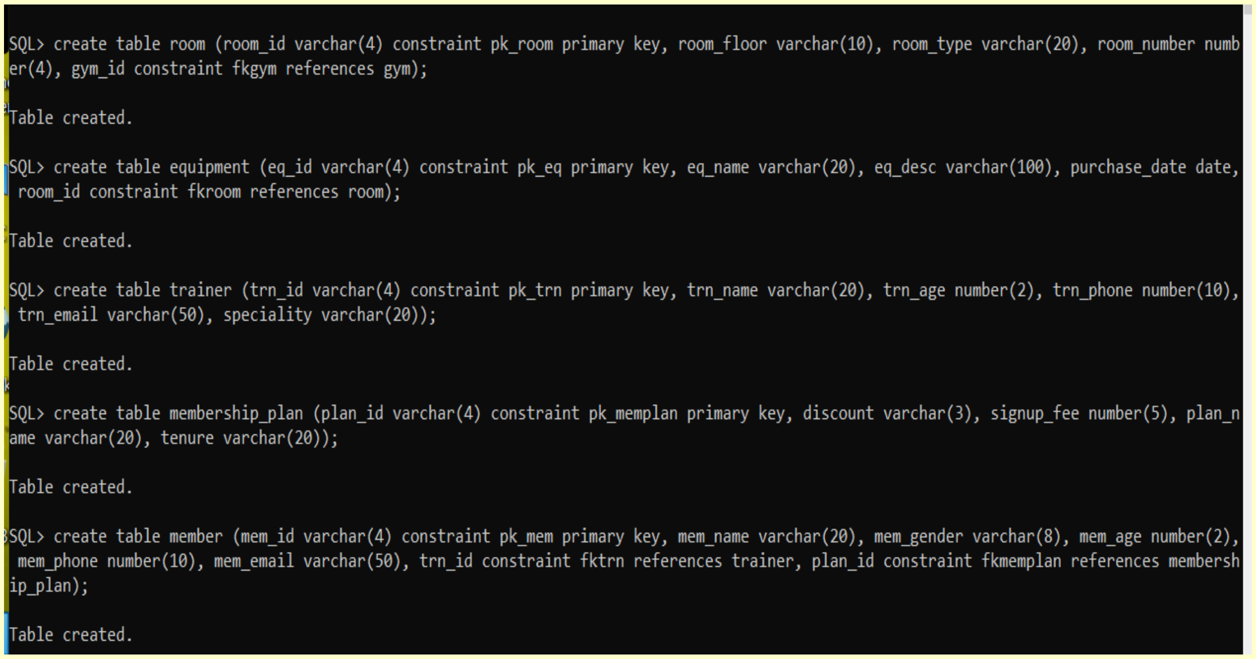
alter table has add constraint fkrmid foreign key(room_id) references room;

SCREENSHOTS

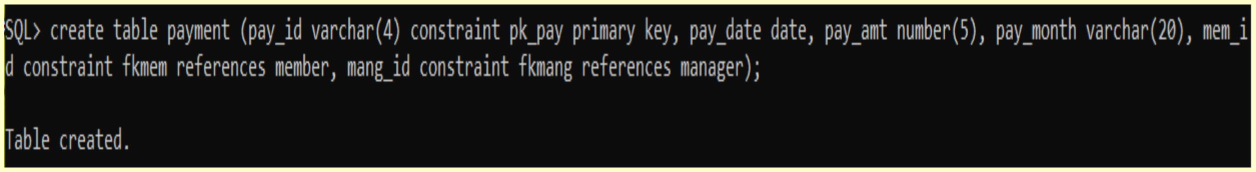
CREATING TABLES:



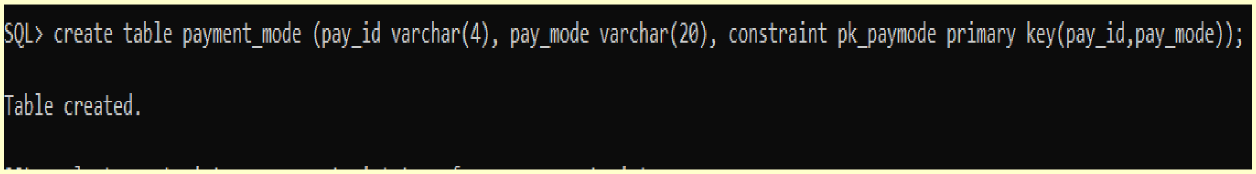
```
SQL> create table owner (own_id varchar(4) constraint pk_own primary key, own_name varchar(30), own_phone number(10), own_add varchar(100)
);

Table created.

SQL> create table gym (gym_id varchar(4) constraint pk_gym primary key, gym_name varchar(20), gym_add varchar(100), gym_landmark varchar(2
0));

Table created.

SQL> create table manager (mang_id varchar(4) constraint pk_manager primary key, mang_name varchar(20), mang_gender varchar(6), mang_phone
 number(10), own_id constraint fkown references owner);

Table created.
```



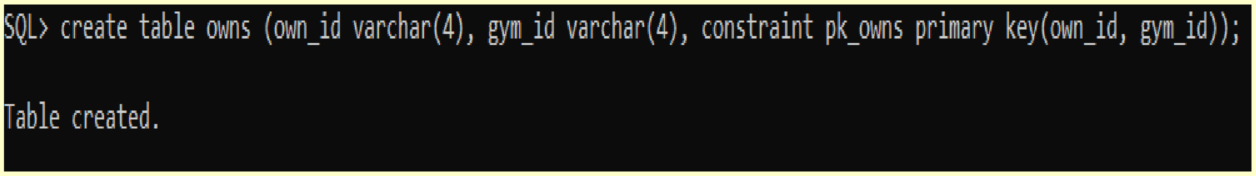
```
SQL> create table room (room_id varchar(4) constraint pk_room primary key, room_floor varchar(10), room_type varchar(20), room_number numb
er(4), gym_id constraint fkgym references gym);

Table created.

SQL> create table equipment (eq_id varchar(4) constraint pk_eq primary key, eq_name varchar(20), eq_desc varchar(100), purchase_date date,
 room_id constraint fkroom references room);

Table created.

SQL> create table trainer (trn_id varchar(4) constraint pk_trn primary key, trn_name varchar(20), trn_age number(2), trn_phone number(10),
 trn_email varchar(50), speciality varchar(20));

Table created.

SQL> create table membership_plan (plan_id varchar(4) constraint pk_memplan primary key, discount varchar(3), signup_fee number(5), plan_n
ame varchar(20), tenure varchar(20));

Table created.

SQL> create table member (mem_id varchar(4) constraint pk_mem primary key, mem_name varchar(20), mem_gender varchar(8), mem_age number(2),
 mem_phone number(10), mem_email varchar(50), trn_id constraint fktrn references trainer, plan_id constraint fkmemplan references membersh
ip_plan);

Table created.
```



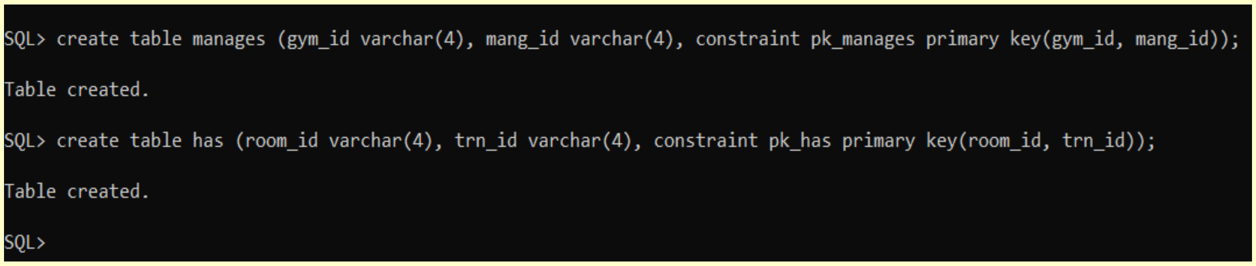
```
SQL> create table payment (pay_id varchar(4) constraint pk_pay primary key, pay_date date, pay_amt number(5), pay_month varchar(20), mem_i
d constraint fkmem references member, mang_id constraint fkmang references manager);

Table created.
```



```
SQL> create table payment_mode (pay_id varchar(4), pay_mode varchar(20), constraint pk_paymode primary key(pay_id,pay_mode));

Table created.
```

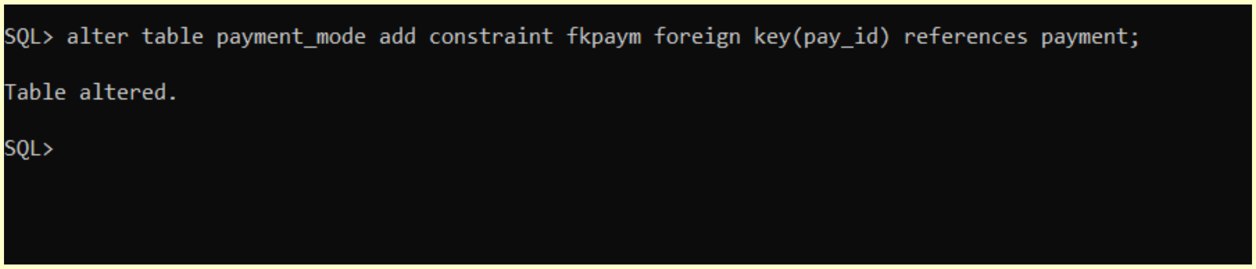


```
SQL> create table owns (own_id varchar(4), gym_id varchar(4), constraint pk_owns primary key(own_id, gym_id));

Table created.
```

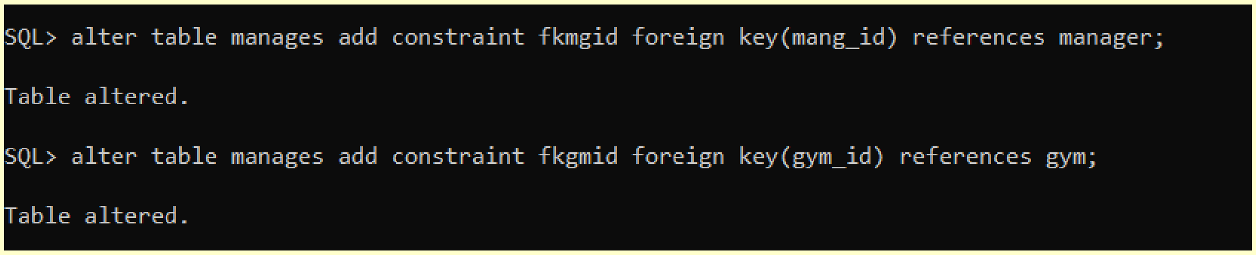


```
SQL> create table manages (gym_id varchar(4), mang_id varchar(4), constraint pk_manages primary key(gym_id, mang_id));

Table created.

SQL> create table has (room_id varchar(4), trn_id varchar(4), constraint pk_has primary key(room_id, trn_id));

Table created.

SQL>
```

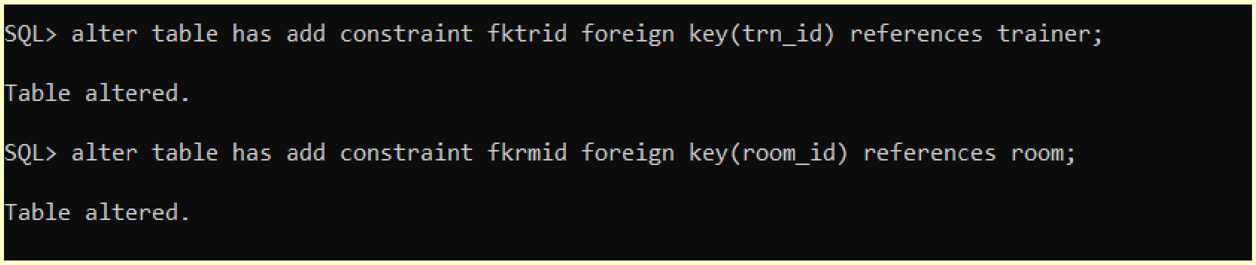
ALTERING TABLES



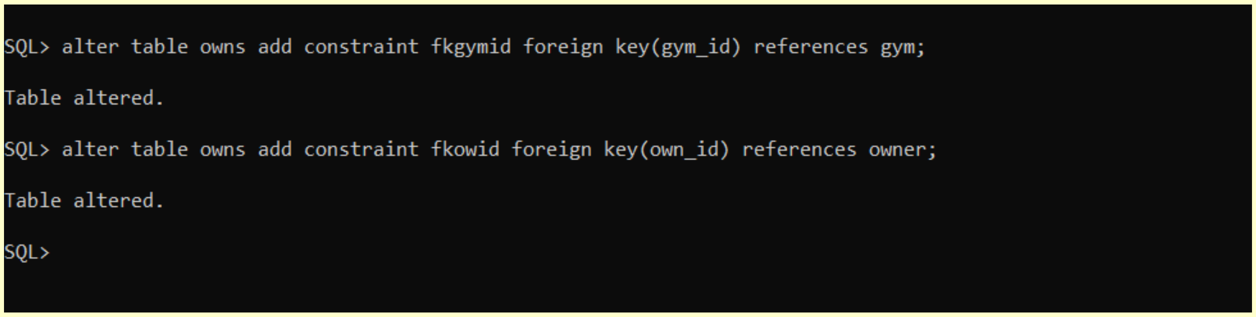
```
SQL> alter table payment_mode add constraint fkpaym foreign key(pay_id) references payment;

Table altered.

SQL>
```



```
SQL> alter table manages add constraint fkmgid foreign key(mang_id) references manager;

Table altered.

SQL> alter table manages add constraint fkgmid foreign key(gym_id) references gym;

Table altered.
```

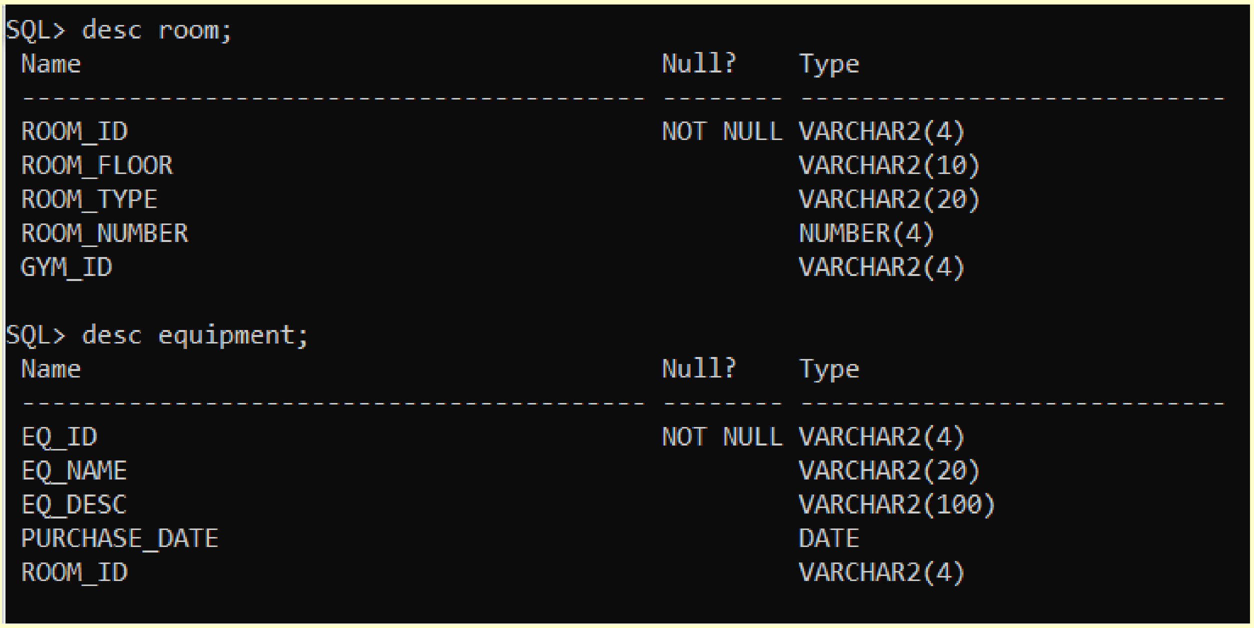


```
SQL> alter table has add constraint fktrid foreign key(trn_id) references trainer;

Table altered.

SQL> alter table has add constraint fkrmid foreign key(room_id) references room;

Table altered.
```

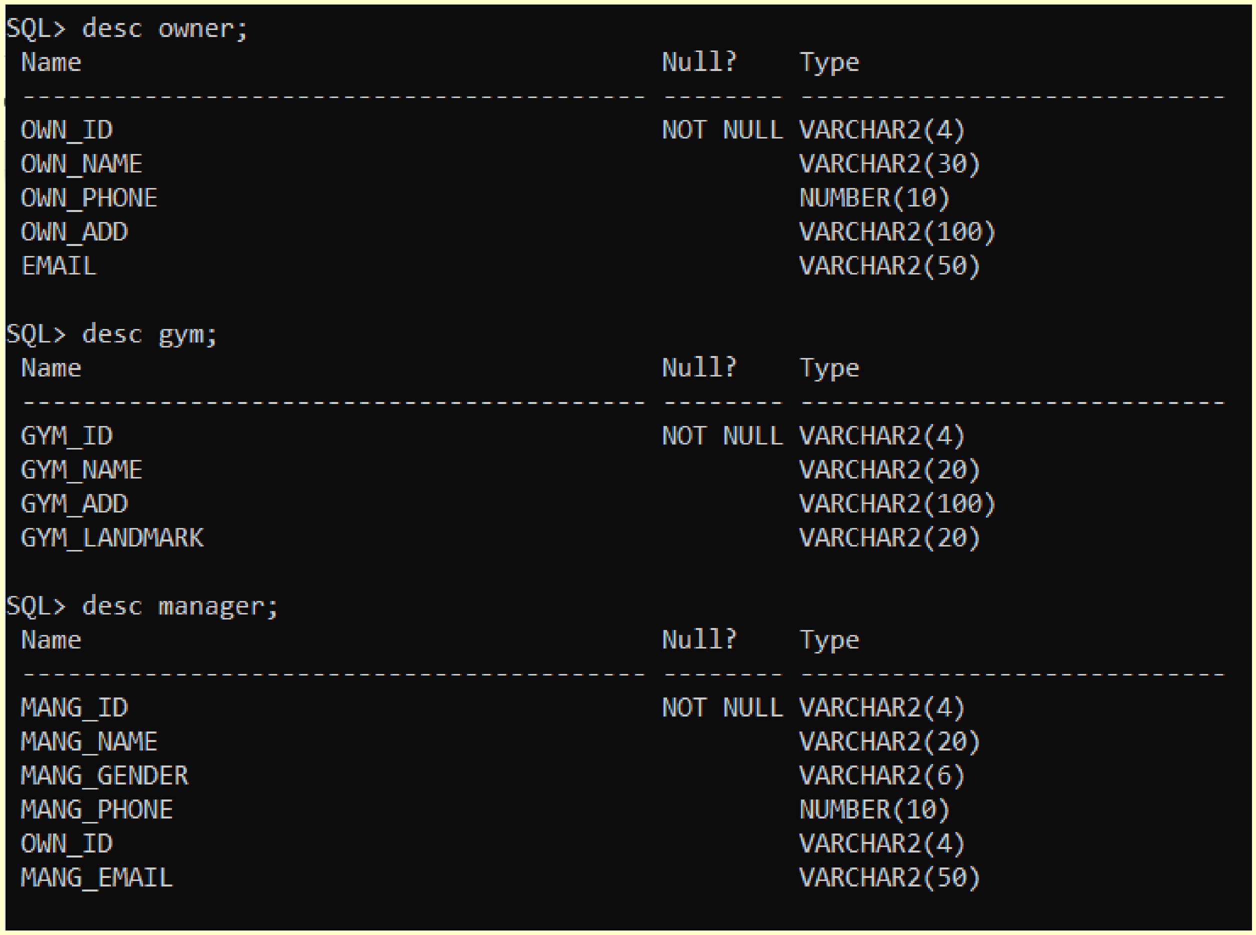


```
SQL> alter table owns add constraint fkgymid foreign key(gym_id) references gym;

Table altered.

SQL> alter table owns add constraint fkowid foreign key(own_id) references owner;

Table altered.

SQL>
```

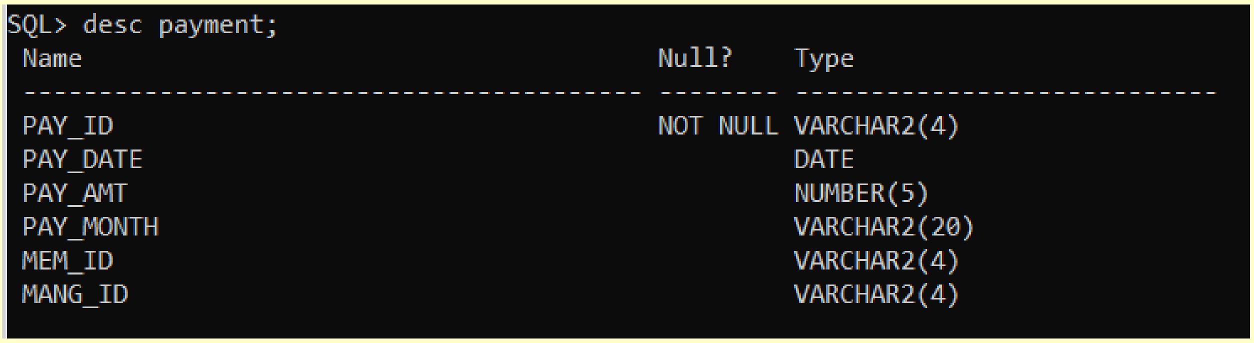
DESCRIBING TABLES:



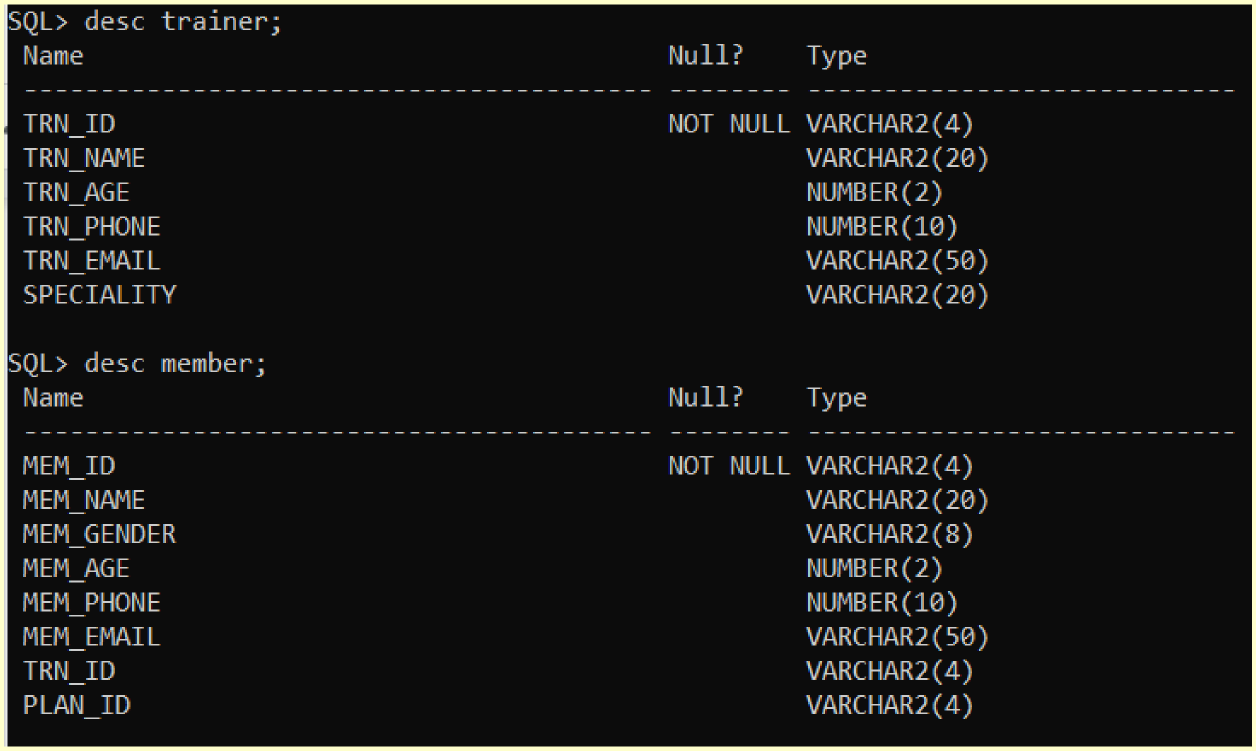
```
SQL> desc room;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROOM_ID                                   NOT NULL VARCHAR2(4)
 ROOM_FLOOR                                         VARCHAR2(10)
 ROOM_TYPE                                          VARCHAR2(20)
 ROOM_NUMBER                                        NUMBER(4)
 GYM_ID                                             VARCHAR2(4)

SQL> desc equipment;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EQ_ID                                     NOT NULL VARCHAR2(4)
 EQ_NAME                                            VARCHAR2(20)
 EQ_DESC                                            VARCHAR2(100)
 PURCHASE_DATE                                      DATE
 ROOM_ID                                            VARCHAR2(4)
```



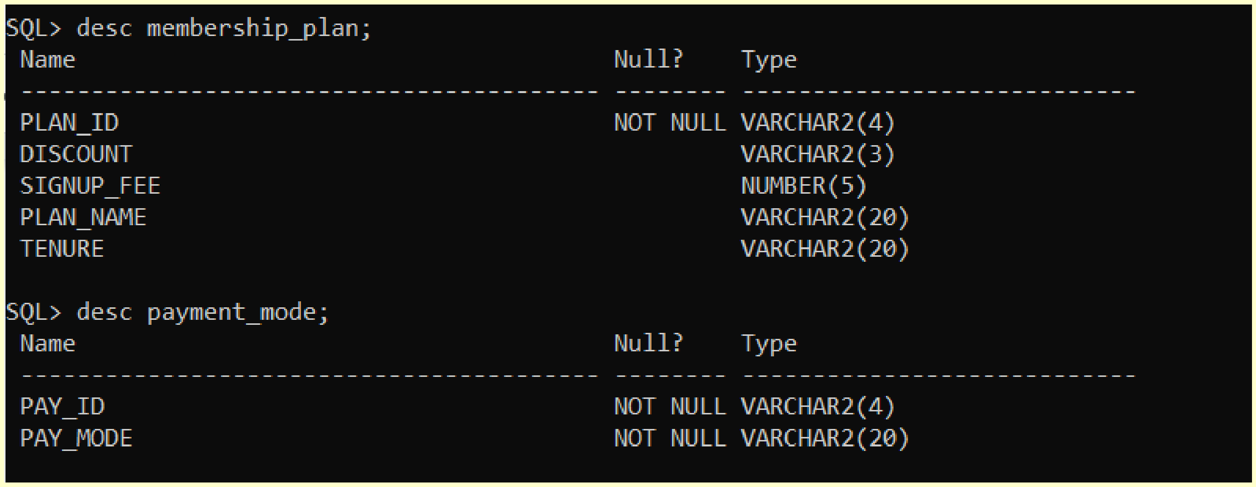
```
SQL> desc owner;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWN_ID                                    NOT NULL VARCHAR2(4)
 OWN_NAME                                           VARCHAR2(30)
 OWN_PHONE                                          NUMBER(10)
 OWN_ADD                                            VARCHAR2(100)
 EMAIL                                              VARCHAR2(50)

SQL> desc gym;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GYM_ID                                    NOT NULL VARCHAR2(4)
 GYM_NAME                                           VARCHAR2(20)
 GYM_ADD                                            VARCHAR2(100)
 GYM_LANDMARK                                       VARCHAR2(20)

SQL> desc manager;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MANG_ID                                   NOT NULL VARCHAR2(4)
 MANG_NAME                                          VARCHAR2(20)
 MANG_GENDER                                        VARCHAR2(6)
 MANG_PHONE                                         NUMBER(10)
 OWN_ID                                             VARCHAR2(4)
 MANG_EMAIL                                         VARCHAR2(50)
```



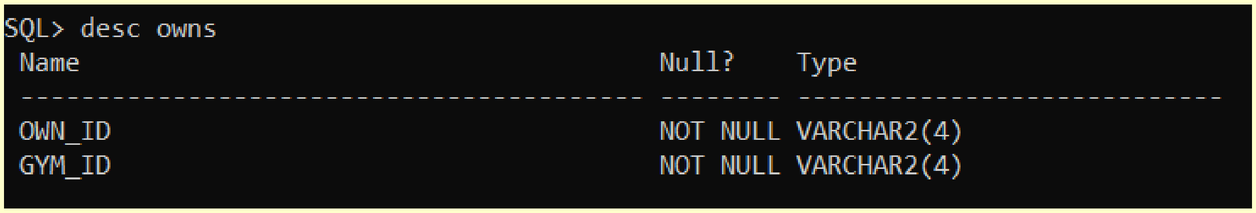
```
SQL> desc payment;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAY_ID                                    NOT NULL VARCHAR2(4)
 PAY_DATE                                           DATE
 PAY_AMT                                            NUMBER(5)
 PAY_MONTH                                          VARCHAR2(20)
 MEM_ID                                             VARCHAR2(4)
 MANG_ID                                            VARCHAR2(4)
```



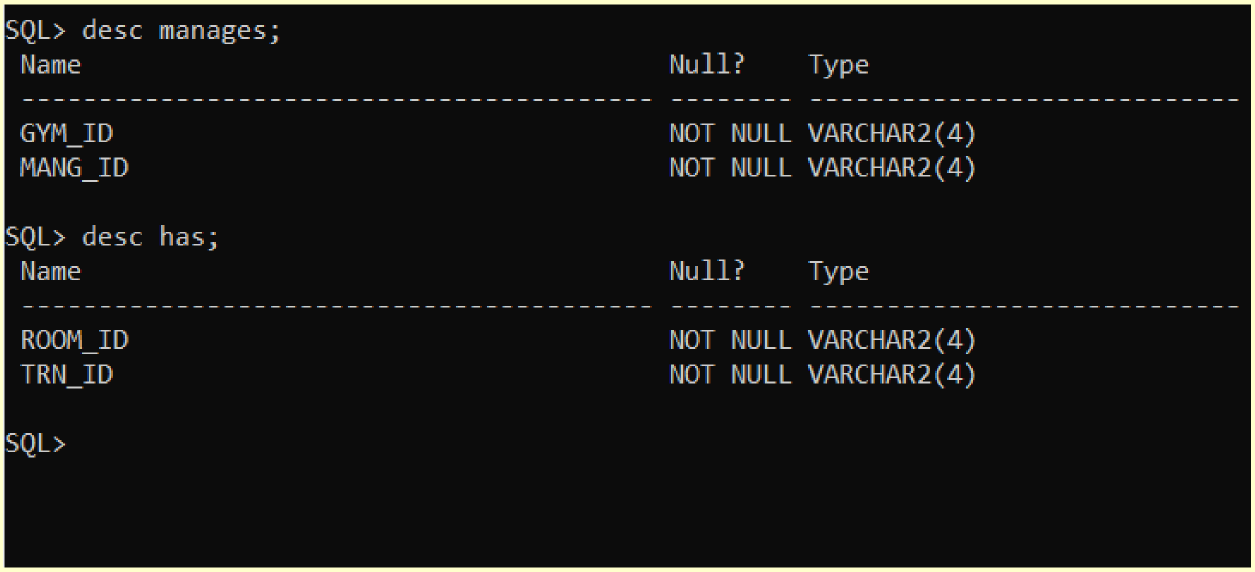
```
SQL> desc trainer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TRN_ID                                    NOT NULL VARCHAR2(4)
 TRN_NAME                                           VARCHAR2(20)
 TRN_AGE                                            NUMBER(2)
 TRN_PHONE                                          NUMBER(10)
 TRN_EMAIL                                          VARCHAR2(50)
 SPECIALITY                                         VARCHAR2(20)

SQL> desc member;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEM_ID                                    NOT NULL VARCHAR2(4)
 MEM_NAME                                           VARCHAR2(20)
 MEM_GENDER                                         VARCHAR2(8)
 MEM_AGE                                            NUMBER(2)
 MEM_PHONE                                          NUMBER(10)
 MEM_EMAIL                                          VARCHAR2(50)
 TRN_ID                                             VARCHAR2(4)
 PLAN_ID                                            VARCHAR2(4)
```



```
SQL> desc membership_plan;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PLAN_ID                                   NOT NULL VARCHAR2(4)
 DISCOUNT                                           VARCHAR2(3)
 SIGNUP_FEE                                         NUMBER(5)
 PLAN_NAME                                          VARCHAR2(20)
 TENURE                                             VARCHAR2(20)

SQL> desc payment_mode;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAY_ID                                    NOT NULL VARCHAR2(4)
 PAY_MODE                                  NOT NULL VARCHAR2(20)
```

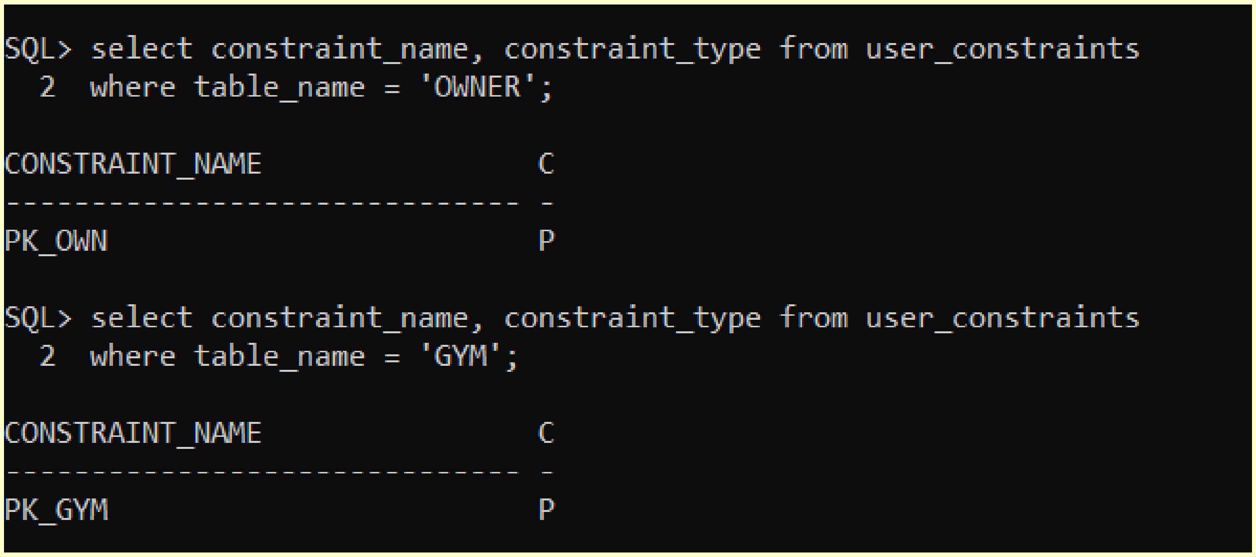


```
SQL> desc owns
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWN_ID                                    NOT NULL VARCHAR2(4)
 GYM_ID                                    NOT NULL VARCHAR2(4)
```

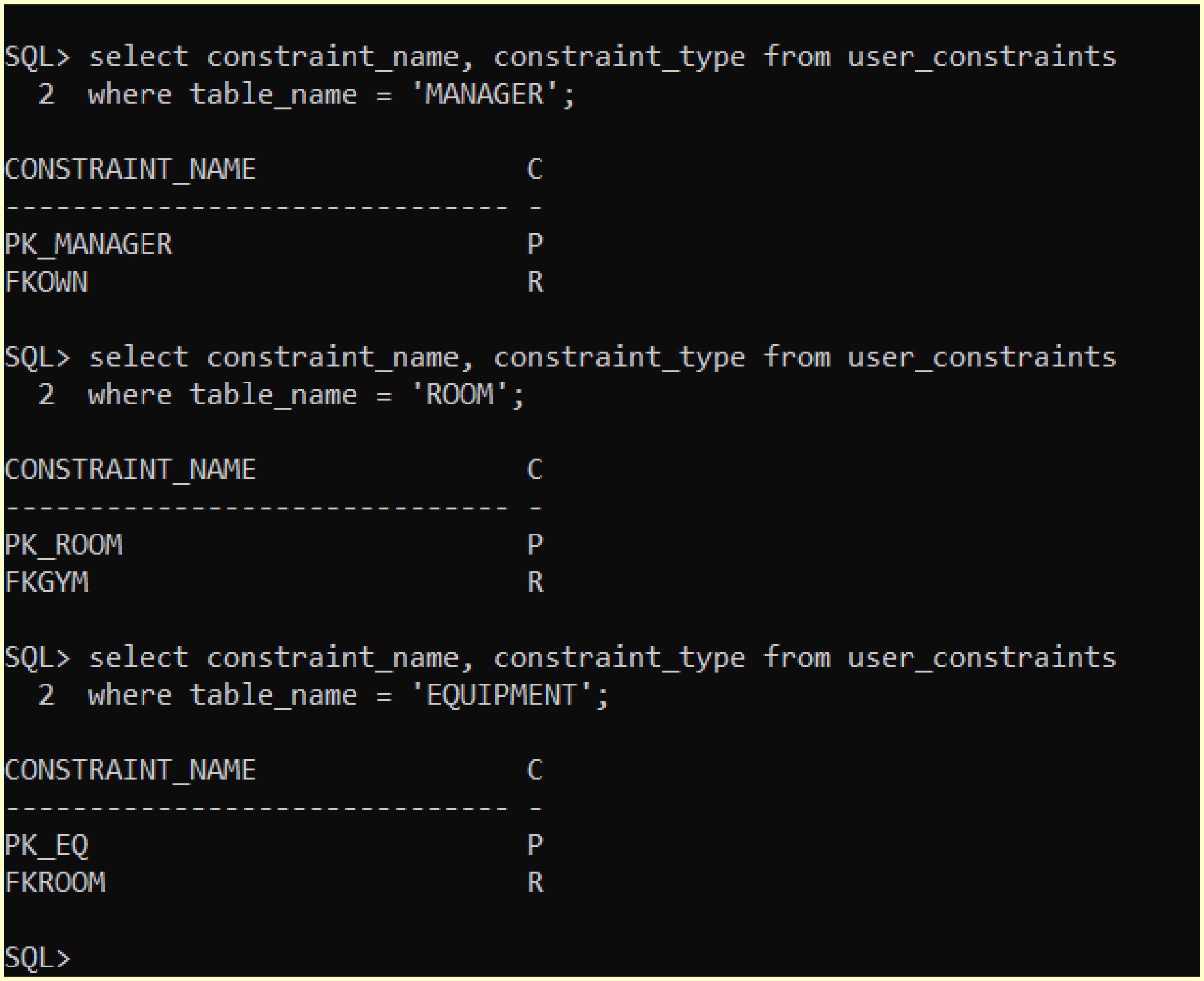


```
SQL> desc manages;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GYM_ID                                    NOT NULL VARCHAR2(4)
 MANG_ID                                   NOT NULL VARCHAR2(4)

SQL> desc has;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROOM_ID                                   NOT NULL VARCHAR2(4)
 TRN_ID                                    NOT NULL VARCHAR2(4)

SQL>
```

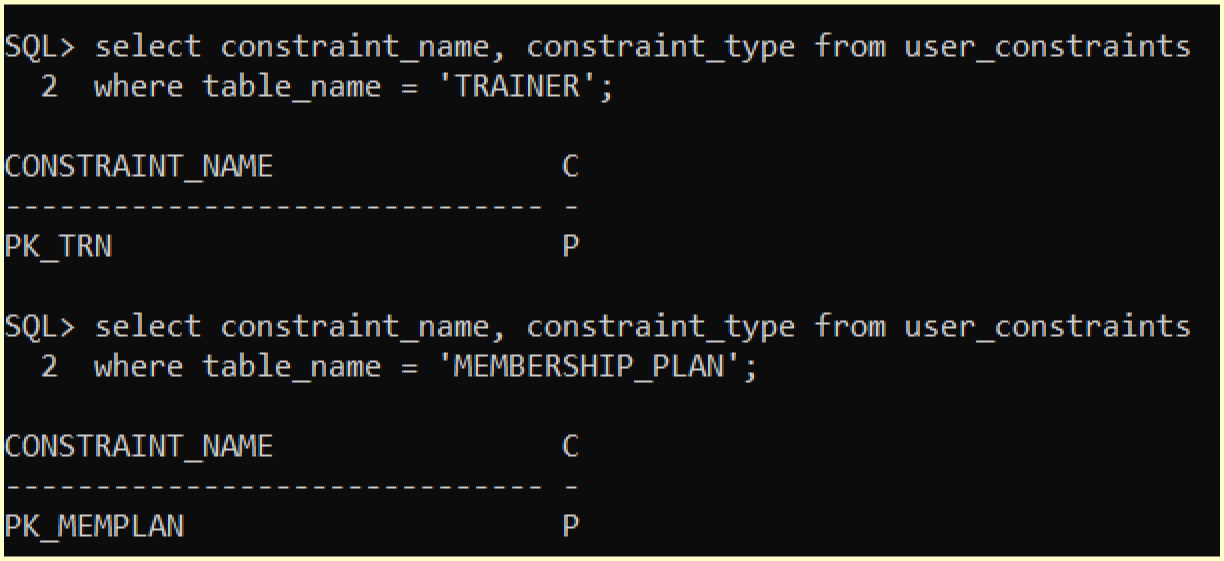
NAMED CONSTRAINTS OF EACH TABLE:



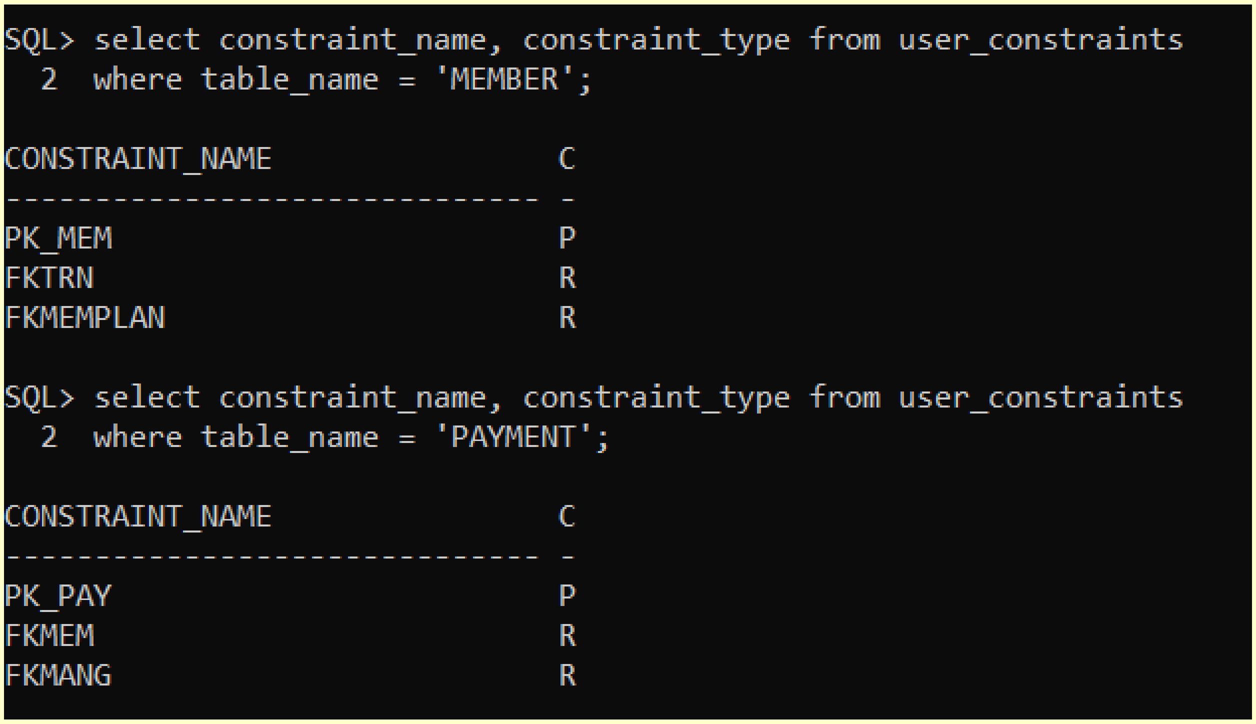
```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'OWNER';

CONSTRAINT_NAME                C
------------------------------ -
PK_OWN                         P

SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'GYM';

CONSTRAINT_NAME                C
------------------------------ -
PK_GYM                         P
```



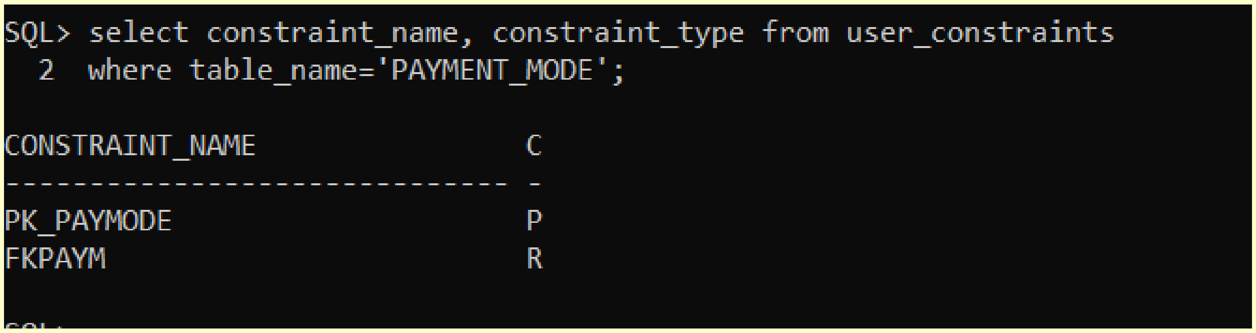
```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'MANAGER';

CONSTRAINT_NAME                C
------------------------------ -
PK_MANAGER                     P
FKOWN                          R

SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'ROOM';

CONSTRAINT_NAME                C
------------------------------ -
PK_ROOM                        P
FKGYM                          R

SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'EQUIPMENT';

CONSTRAINT_NAME                C
------------------------------ -
PK_EQ                          P
FKROOM                         R

SQL>
```



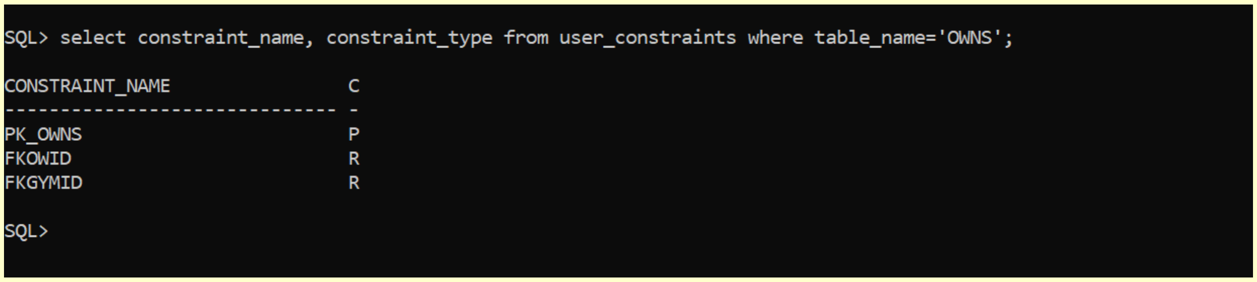
```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'TRAINER';

CONSTRAINT_NAME                C
------------------------------ -
PK_TRN                         P

SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'MEMBERSHIP_PLAN';

CONSTRAINT_NAME                C
------------------------------ -
PK_MEMPLAN                     P
```



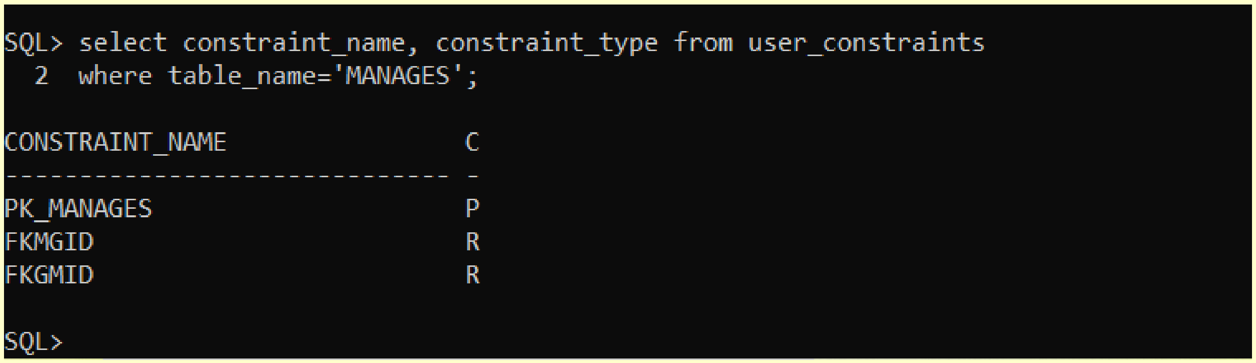
```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'MEMBER';

CONSTRAINT_NAME                C
------------------------------ -
PK_MEM                         P
FKTRN                          R
FKMEMPLAN                      R

SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name = 'PAYMENT';

CONSTRAINT_NAME                C
------------------------------ -
PK_PAY                         P
FKMEM                          R
FKMANG                         R
```



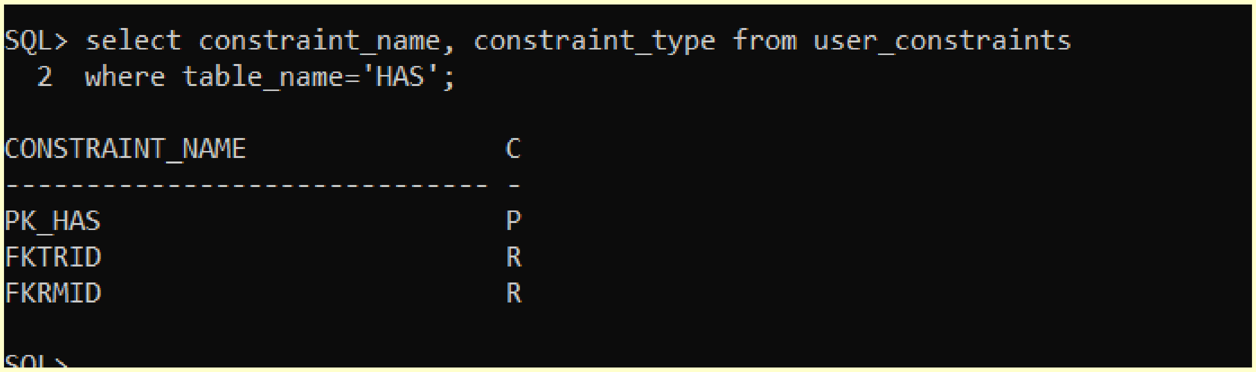
```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name='PAYMENT_MODE';

CONSTRAINT_NAME                C
------------------------------ -
PK_PAYMODE                     P
FKPAYM                         R
```



```
SQL> select constraint_name, constraint_type from user_constraints where table_name='OWNS';

CONSTRAINT_NAME                C
------------------------------ -
PK_OWNS                        P
FKOWID                         R
FKGYMID                        R

SQL>
```



```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name='MANAGES';

CONSTRAINT_NAME                C
------------------------------ -
PK_MANAGES                     P
FKMGID                         R
FKGMID                         R

SQL>
```



```
SQL> select constraint_name, constraint_type from user_constraints
  2  where table_name='HAS';

CONSTRAINT_NAME                C
------------------------------ -
PK_HAS                         P
FKTRID                         R
FKRMID                         R
```

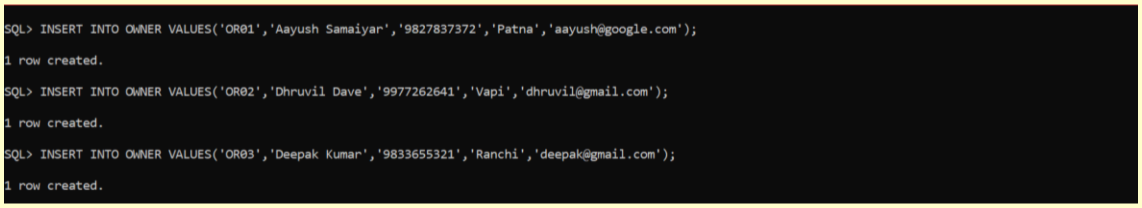
INSERTION OF VALUES IN TABLES

1. inserting values in owner table

INSERT INTO OWNER VALUES('OR01','Aayush Samaiyar','9827837372','Patna','aayush@google.com');

INSERT INTO OWNER VALUES('OR02','Dhruvil Dave','9977262641','Vapi','dhruvil@gmail.com');

INSERT INTO OWNER VALUES('OR03','Deepak Kumar','9833655321','Ranchi','deepak@gmail.com');



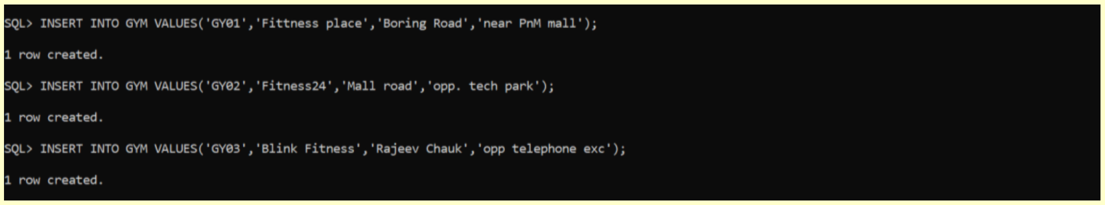
```
SQL> INSERT INTO OWNER VALUES('OR01','Aayush Samaiyar','9827837372','Patna','aayush@google.com');

1 row created.

SQL> INSERT INTO OWNER VALUES('OR02','Dhruvil Dave','9977262641','Vapi','dhruvil@gmail.com');

1 row created.

SQL> INSERT INTO OWNER VALUES('OR03','Deepak Kumar','9833655321','Ranchi','deepak@gmail.com');

1 row created.
```

2. inserting values into gym table

INSERT INTO GYM VALUES('GY01','Fitness place','Boring Road','near PnM mall');

INSERT INTO GYM VALUES('GY02','Fitness24','Mall road','opp. tech park');

INSERT INTO GYM VALUES('GY03','Blink Fitness','Rajeev Chauk','opp telephone exc');



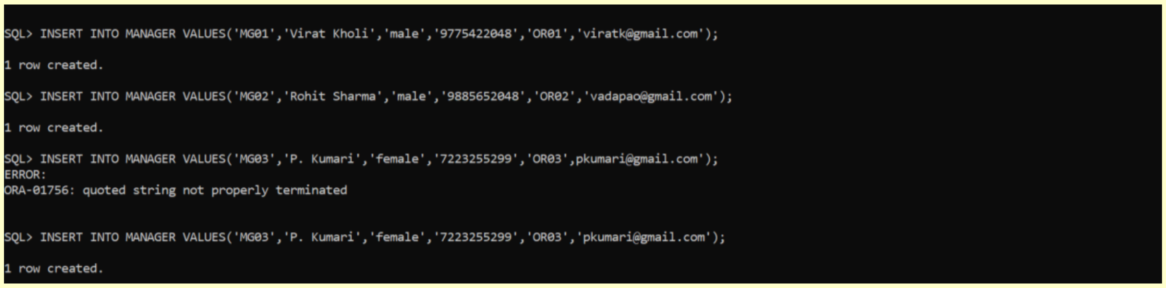
```
SQL> INSERT INTO GYM VALUES('GY01','Fittness place','Boring Road','near PnM mall');

1 row created.

SQL> INSERT INTO GYM VALUES('GY02','Fitness24','Mall road','opp. tech park');

1 row created.

SQL> INSERT INTO GYM VALUES('GY03','Blink Fitness','Rajeev Chauk','opp telephone exc');

1 row created.
```

3. I**nserting values into manager table**

INSERT INTO MANAGER VALUES('MG01','Virat Kholi','male','9775422048','OR01','viratk@gmail.com');

INSERT INTO MANAGER VALUES('MG02','Rohit Sharma','male','9885652048','OR02','vadapao@gmail.com');

INSERT INTO MANAGER VALUES('MG03','P. Kumari','female','7223255299','OR03','pkumari@gmail.com');



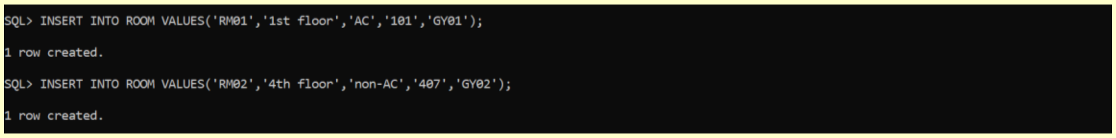
```
SQL> INSERT INTO MANAGER VALUES('MG01','Virat Kholi','male','9775422048','OR01','viratk@gmail.com');

1 row created.

SQL> INSERT INTO MANAGER VALUES('MG02','Rohit Sharma','male','9885652048','OR02','vadapao@gmail.com');

1 row created.

SQL> INSERT INTO MANAGER VALUES('MG03','P. Kumari','female','7223255299','OR03',pkumari@gmail.com');
ERROR:
ORA-01756: quoted string not properly terminated


SQL> INSERT INTO MANAGER VALUES('MG03','P. Kumari','female','7223255299','OR03','pkumari@gmail.com');

1 row created.
```

4. Inserting values into room table

INSERT INTO ROOM VALUES('RM01','1st floor','AC','101','GY01');

INSERT INTO ROOM VALUES('RM02','4th floor', 'non-AC', '407', 'GY02');

INSERT INTO ROOM VALUES('RM03','2nd loor', 'AC','203','GY03');



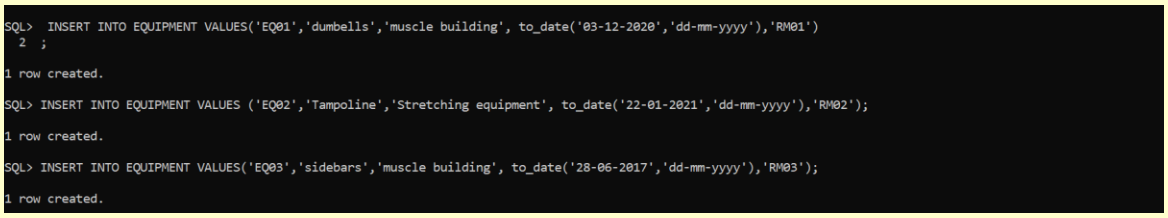
```
SQL> INSERT INTO ROOM VALUES('RM01','1st floor','AC','101','GY01');

1 row created.

SQL> INSERT INTO ROOM VALUES('RM02','4th floor','non-AC','407','GY02');

1 row created.
```

5. Inserting values into equipment table

INSERT INTO EQUIPMENT VALUES('EQ01','dumbells','muscle building', to_date('03-12-2020','dd-mm-yyyy'),'RM01');

INSERT INTO EQUIPMENT VALUES ('EQ02', 'Tampoline', 'Stretching equipment', to_date('22-01-2021','dd-mm-yyyy'), 'RM02');

INSERT INTO EQUIPMENT VALUES('EQ03','sidebars','muscle building', to_date('28-06-2017','dd-mm-yyyy'),'RM03');



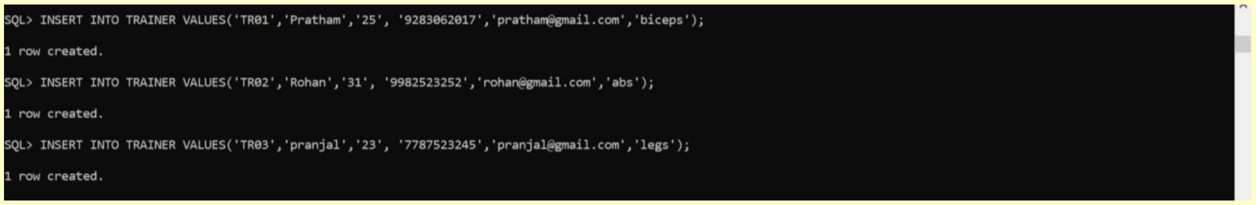
```
SQL>  INSERT INTO EQUIPMENT VALUES('EQ01','dumbells','muscle building', to_date('03-12-2020','dd-mm-yyyy'),'RM01')
  2  ;

1 row created.

SQL> INSERT INTO EQUIPMENT VALUES ('EQ02','Tampoline','Stretching equipment', to_date('22-01-2021','dd-mm-yyyy'),'RM02');

1 row created.

SQL> INSERT INTO EQUIPMENT VALUES('EQ03','sidebars','muscle building', to_date('28-06-2017','dd-mm-yyyy'),'RM03');

1 row created.
```

6. Inserting values into the trainer table.

INSERT INTO TRAINER VALUES('TR01','Pratham','25', '9283062017','pratham@gmail.com','biceps');

INSERT INTO TRAINER VALUES('TR02','Rohan','31', '9982523252','rohan@gmail.com','abs');

INSERT INTO TRAINER VALUES('TR03','pranjal','23', '7787523245','pranjal@gmail.com','legs');



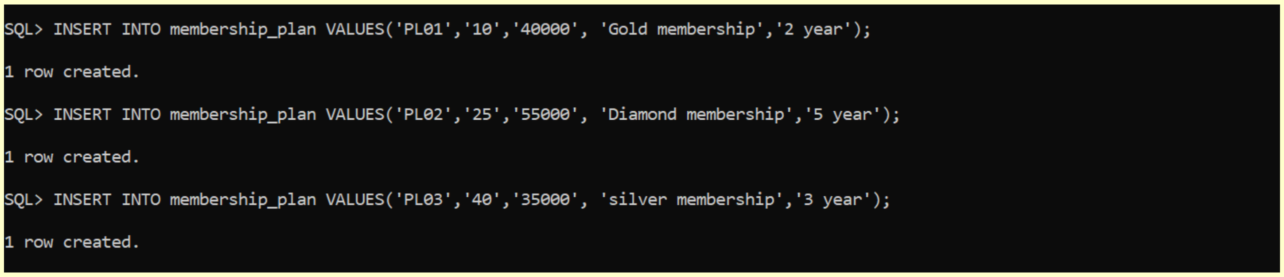
```
SQL> INSERT INTO TRAINER VALUES('TR01','Pratham','25', '9283062017','pratham@gmail.com','biceps');

1 row created.

SQL> INSERT INTO TRAINER VALUES('TR02','Rohan','31', '9982523252','rohan@gmail.com','abs');

1 row created.

SQL> INSERT INTO TRAINER VALUES('TR03','pranjal','23', '7787523245','pranjal@gmail.com','legs');

1 row created.
```

7. Inserting values into membership_plan table

INSERT INTO membership_plan VALUES('PL01','10','40000', 'Gold membership','2 year');

INSERT INTO membership_plan VALUES('PL02','25','55000', 'Diamond membership','5 year');

INSERT INTO membership_plan VALUES('PL03','40','35000', 'silver membership','3 year');



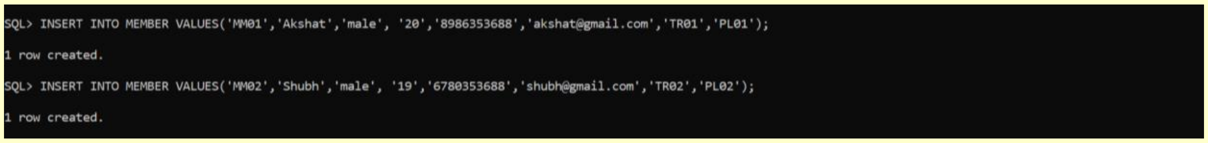
```
SQL> INSERT INTO membership_plan VALUES('PL01','10','40000', 'Gold membership','2 year');

1 row created.

SQL> INSERT INTO membership_plan VALUES('PL02','25','55000', 'Diamond membership','5 year');

1 row created.

SQL> INSERT INTO membership_plan VALUES('PL03','40','35000', 'silver membership','3 year');

1 row created.
```

8. Inserting values into member table

INSERT INTO MEMBER VALUES('MM01','Akshat','male', '20','8986353688','akshat@gmail.com','TR01','PL01');

INSERT INTO MEMBER VALUES('MM02','Shubh','male', '19','6780353688','shubh@gmail.com','TR02','PL02');

INSERT INTO MEMBER VALUES('MM03','Ritika','female', '21','8986353772','rikika@gmail.com','TR03','PL03');



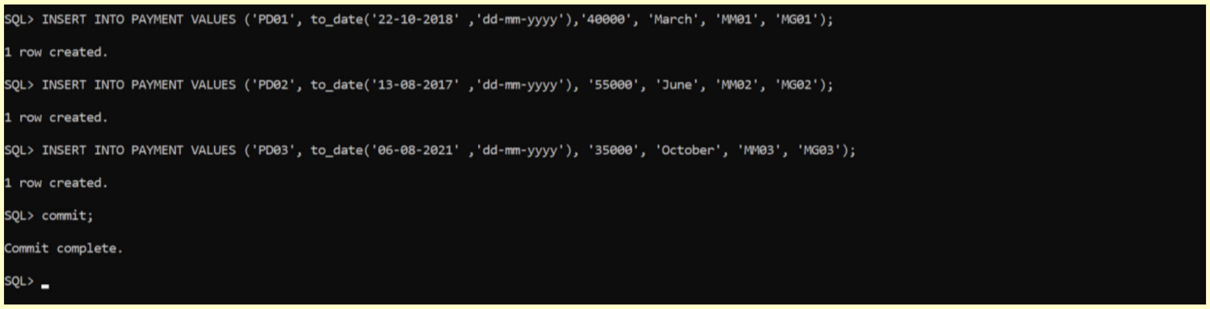
```
SQL> INSERT INTO MEMBER VALUES('MM01','Akshat','male', '20','8986353688','akshat@gmail.com','TR01','PL01');

1 row created.

SQL> INSERT INTO MEMBER VALUES('MM02','Shubh','male', '19','6780353688','shubh@gmail.com','TR02','PL02');

1 row created.
```

9. Inserting values into payment table

INSERT INTO PAYMENT VALUES ('PD01', to_date('22-10-2018' ,'dd-mm-yyyy'),'40000', 'March', 'MM01', 'MG01');

INSERT INTO PAYMENT VALUES ('PD02', to_date('13-08-2017' ,'dd-mm-yyyy'), '55000', 'June', 'MM02', 'MG02');

INSERT INTO PAYMENT VALUES ('PD03', to_date('06-08-2021' ,'dd-mm-yyyy'), '35000', 'October', 'MM03', 'MG03');



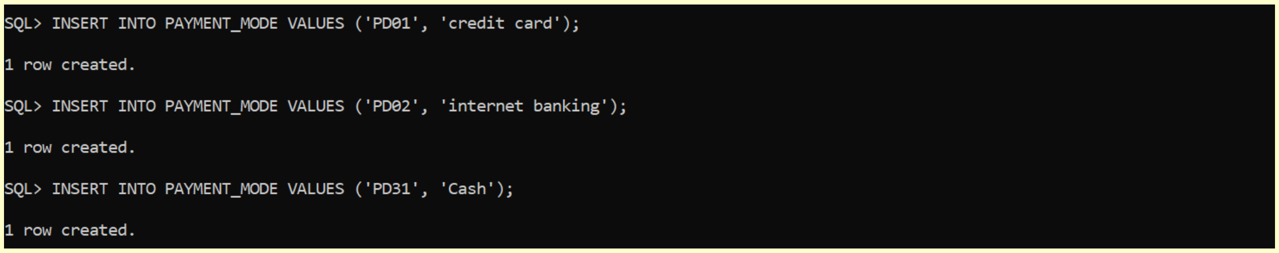
```
SQL> INSERT INTO PAYMENT VALUES ('PD01', to_date('22-10-2018' ,'dd-mm-yyyy'),'40000', 'March', 'MM01', 'MG01');

1 row created.

SQL> INSERT INTO PAYMENT VALUES ('PD02', to_date('13-08-2017' ,'dd-mm-yyyy'), '55000', 'June', 'MM02', 'MG02');

1 row created.

SQL> INSERT INTO PAYMENT VALUES ('PD03', to_date('06-08-2021' ,'dd-mm-yyyy'), '35000', 'October', 'MM03', 'MG03');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

10. Inserting values into payment_mode table

INSERT INTO PAYMENT_MODE VALUES ('PD01', 'credit card');

INSERT INTO PAYMENT_MODE VALUES ('PD02', 'internet banking');

INSERT INTO PAYMENT_MODE VALUES ('PD03', 'Cash');



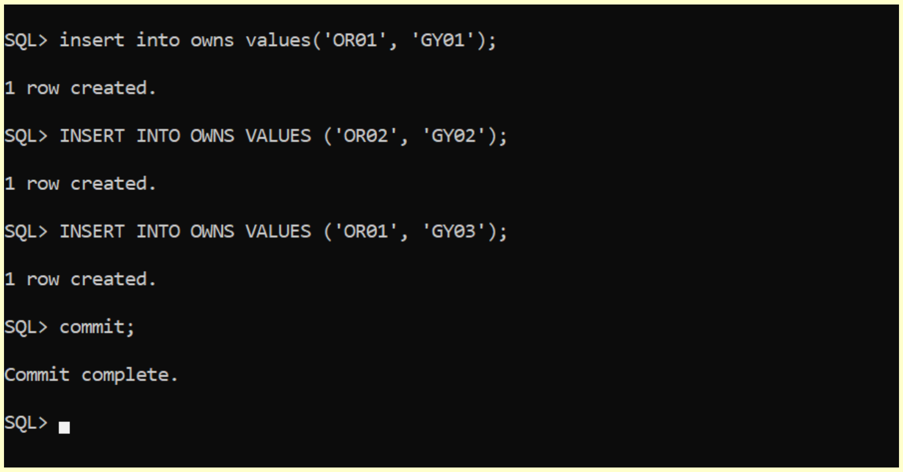
```
SQL> INSERT INTO PAYMENT_MODE VALUES ('PD01', 'credit card');

1 row created.

SQL> INSERT INTO PAYMENT_MODE VALUES ('PD02', 'internet banking');

1 row created.

SQL> INSERT INTO PAYMENT_MODE VALUES ('PD31', 'Cash');

1 row created.
```

11. Inserting values into owns table

INSERT INTO OWNS VALUES ('OR01', 'GY01');

INSERT INTO OWNS VALUES ('OR02', 'GY02');

INSERT INTO OWNS VALUES ('OR01', 'GY03');



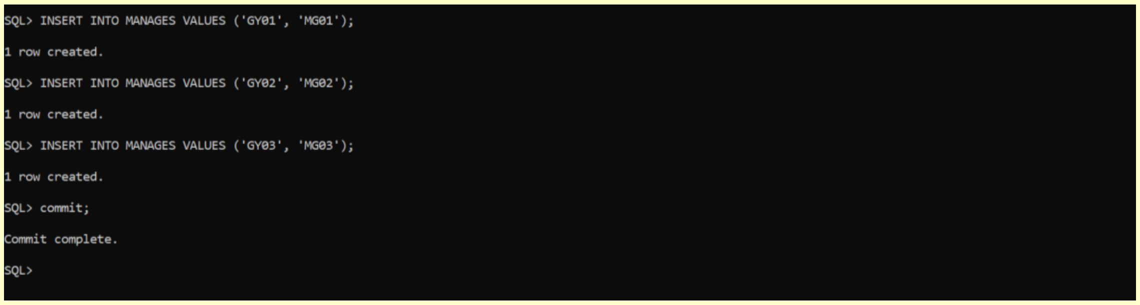
```
SQL> insert into owns values('OR01', 'GY01');

1 row created.

SQL> INSERT INTO OWNS VALUES ('OR02', 'GY02');

1 row created.

SQL> INSERT INTO OWNS VALUES ('OR01', 'GY03');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

12. Inserting values into manages table

INSERT INTO MANAGES VALUES ('GY01', 'MG01');

INSERT INTO MANAGES VALUES ('GY02', 'MG02');

INSERT INTO MANAGES VALUES ('GY03', 'MG03');



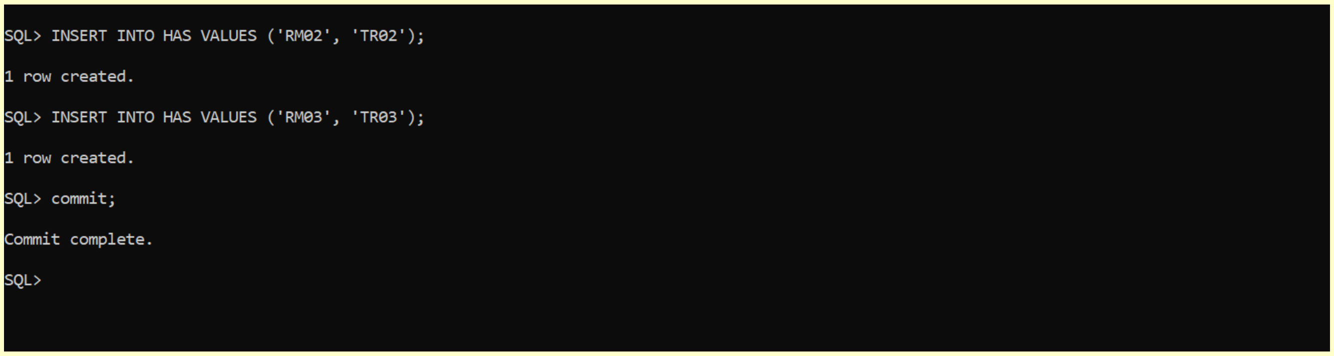
```
SQL> INSERT INTO MANAGES VALUES ('GY01', 'MG01');

1 row created.

SQL> INSERT INTO MANAGES VALUES ('GY02', 'MG02');

1 row created.

SQL> INSERT INTO MANAGES VALUES ('GY03', 'MG03');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

13. Inserting values into has table

INSERT INTO HAS VALUES ('RM01', 'TR01');

INSERT INTO HAS VALUES ('RM02', 'TR02');

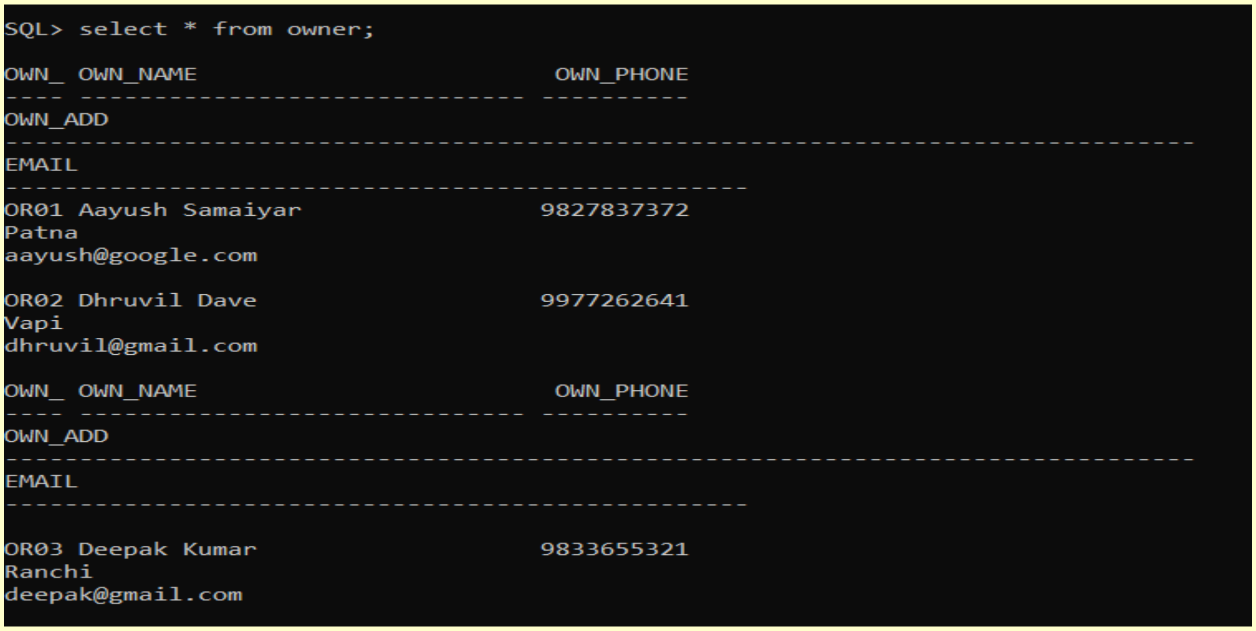
INSERT INTO HAS VALUES ('RM03', 'TR03');



```
SQL> INSERT INTO HAS VALUES ('RM02', 'TR02');

1 row created.

SQL> INSERT INTO HAS VALUES ('RM03', 'TR03');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

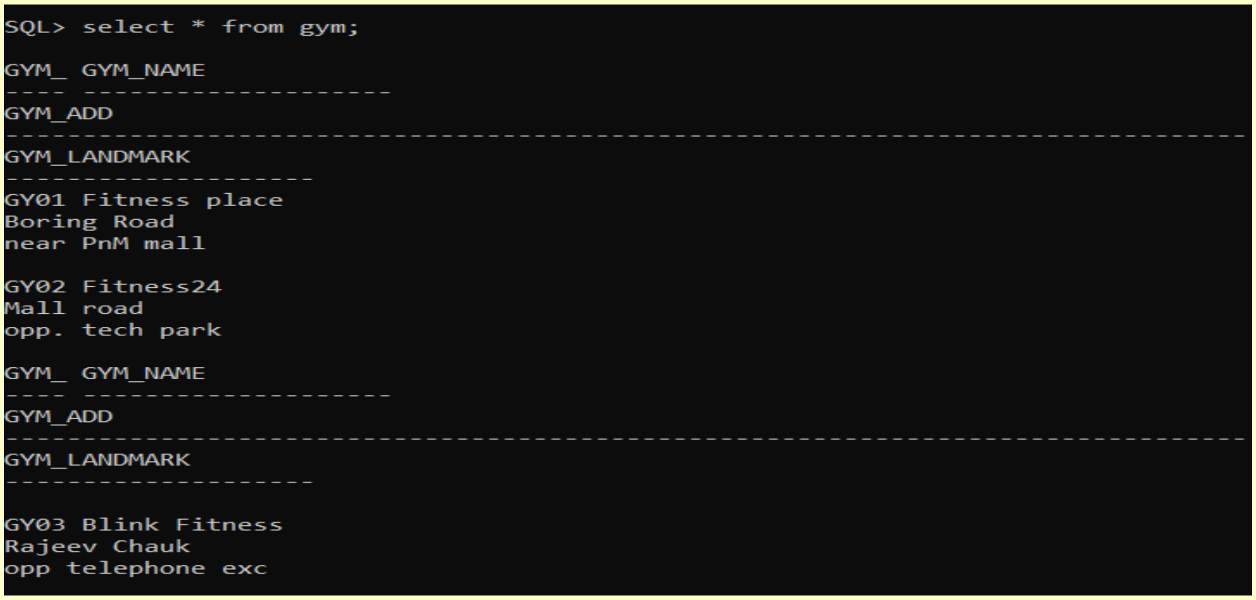
<u>Displaying content of each table:</u>

1. Owner



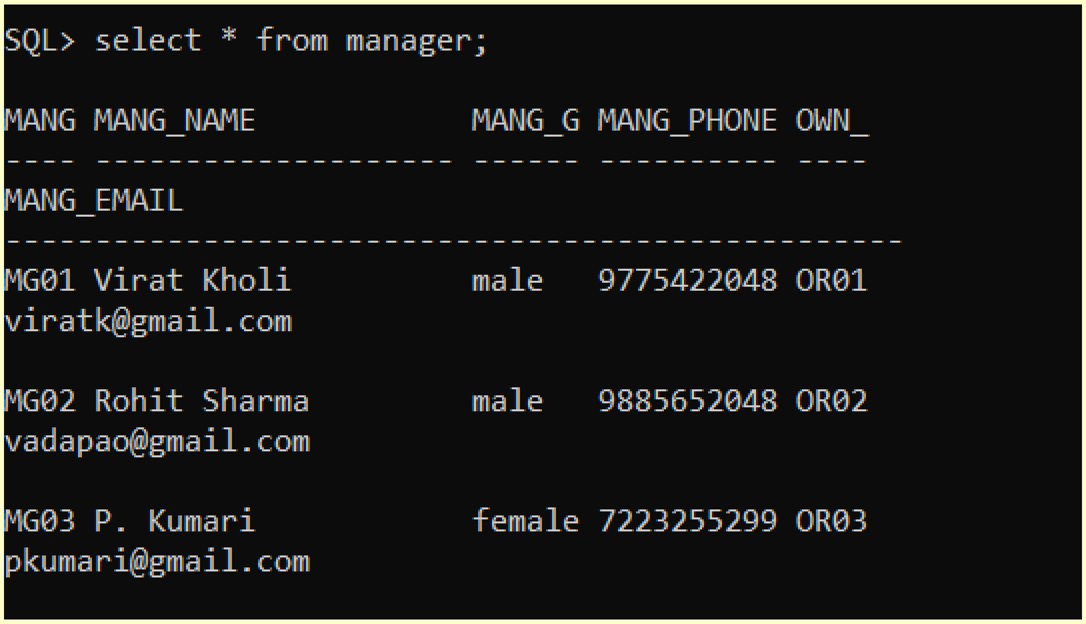
```
SQL> select * from owner;

OWN_ OWN_NAME                        OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------
OR01 Aayush Samaiyar                9827837372
Patna
aayush@google.com

OR02 Dhruvil Dave                   9977262641
Vapi
dhruvil@gmail.com

OWN_ OWN_NAME                        OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------

OR03 Deepak Kumar                   9833655321
Ranchi
deepak@gmail.com
```

2. GYM



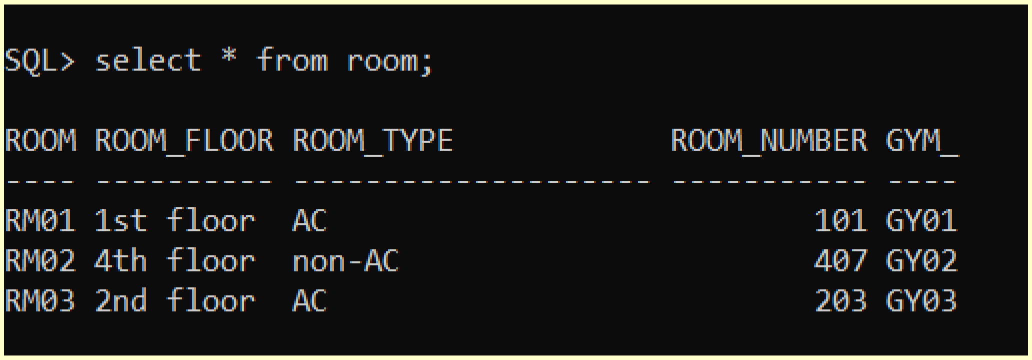
```
SQL> select * from gym;

GYM_ GYM_NAME
---- --------------------
GYM_ADD
--------------------------------------------------------------------------------
GYM_LANDMARK
--------------------
GY01 Fitness place
Boring Road
near PnM mall

GY02 Fitness24
Mall road
opp. tech park

GYM_ GYM_NAME
---- --------------------
GYM_ADD
--------------------------------------------------------------------------------
GYM_LANDMARK
--------------------

GY03 Blink Fitness
Rajeev Chauk
opp telephone exc
```

3. Manager



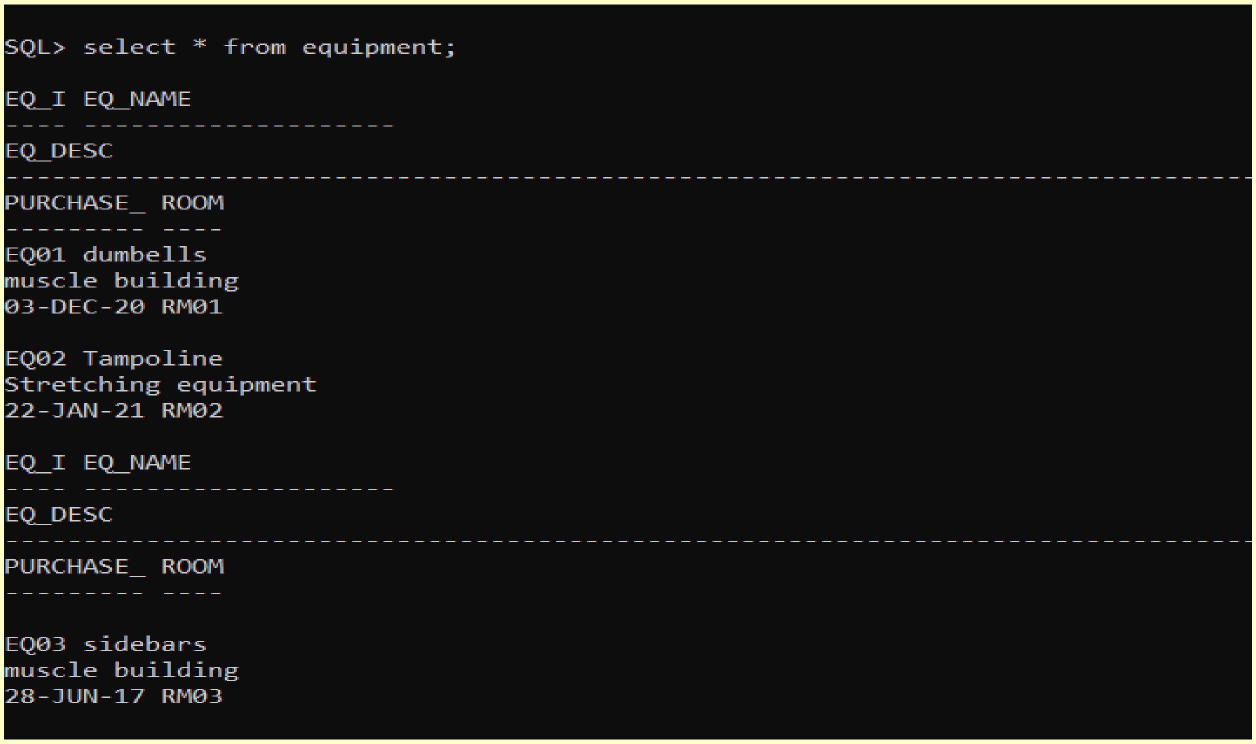
```
SQL> select * from manager;

MANG MANG_NAME            MANG_G MANG_PHONE OWN_
---- -------------------- ------ ---------- ----
MANG_EMAIL
--------------------------------------------------
MG01 Virat Kholi          male   9775422048 OR01
viratk@gmail.com

MG02 Rohit Sharma         male   9885652048 OR02
vadapao@gmail.com

MG03 P. Kumari            female 7223255299 OR03
pkumari@gmail.com
```

4. Room



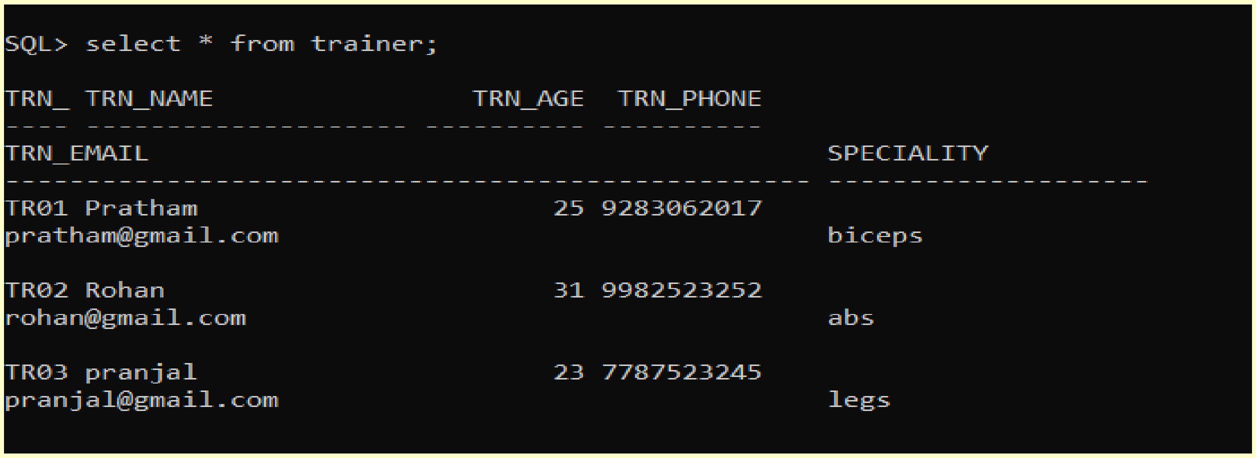
```
SQL> select * from room;

ROOM ROOM_FLOOR ROOM_TYPE            ROOM_NUMBER GYM_
---- ---------- -------------------- ----------- ----
RM01 1st floor  AC                           101 GY01
RM02 4th floor  non-AC                       407 GY02
RM03 2nd floor  AC                           203 GY03
```

5. Equipment



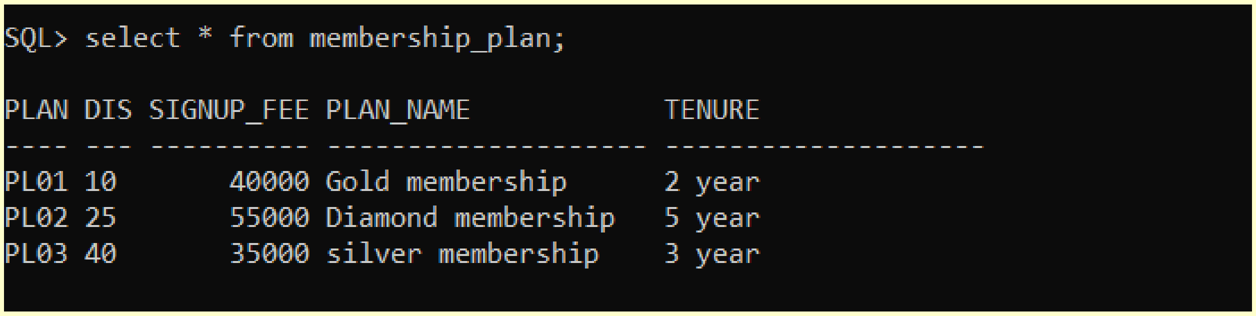
```
SQL> select * from equipment;

EQ_I EQ_NAME
---- --------------------
EQ_DESC
--------------------------------------------------------------------------------
PURCHASE_ ROOM
--------- ----
EQ01 dumbells
muscle building
03-DEC-20 RM01

EQ02 Tampoline
Stretching equipment
22-JAN-21 RM02

EQ_I EQ_NAME
---- --------------------
EQ_DESC
--------------------------------------------------------------------------------
PURCHASE_ ROOM
--------- ----

EQ03 sidebars
muscle building
28-JUN-17 RM03
```

6. Trainer



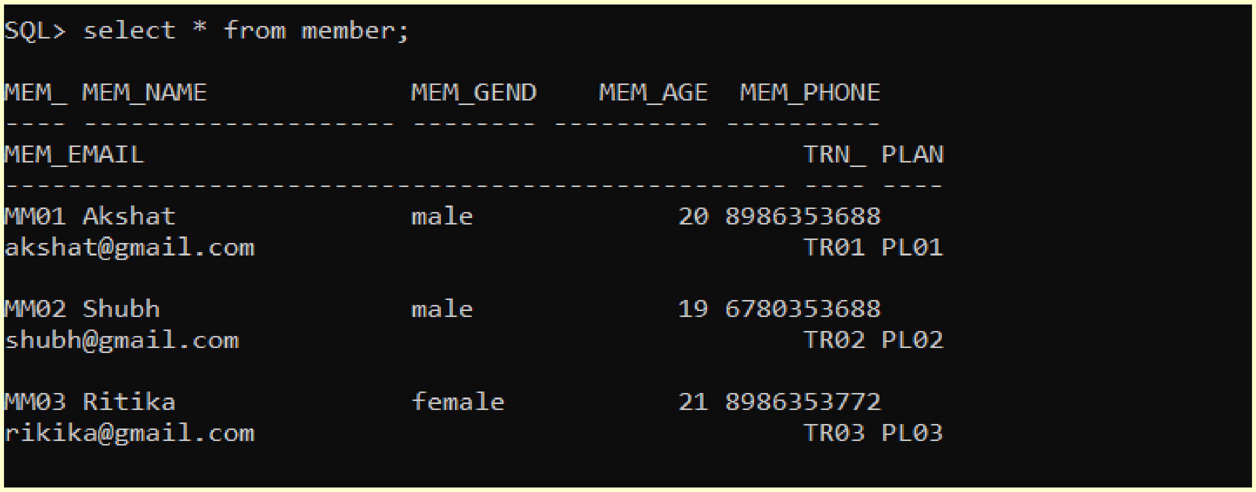
```
SQL> select * from trainer;

TRN_ TRN_NAME                  TRN_AGE  TRN_PHONE
---- -------------------- ---------- ----------
TRN_EMAIL                                          SPECIALITY
-------------------------------------------------- --------------------
TR01 Pratham                      25 9283062017
pratham@gmail.com                                  biceps

TR02 Rohan                        31 9982523252
rohan@gmail.com                                    abs

TR03 pranjal                      23 7787523245
pranjal@gmail.com                                  legs
```

7. Membership_plan

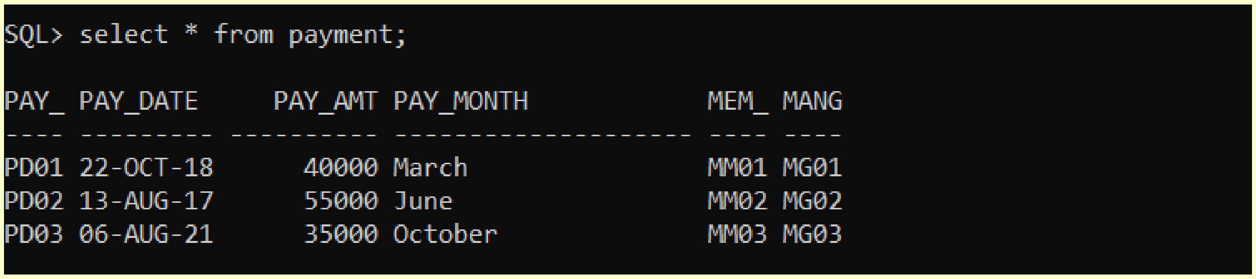


```
SQL> select * from membership_plan;

PLAN DIS SIGNUP_FEE PLAN_NAME            TENURE
---- --- ---------- -------------------- --------------------
PL01 10       40000 Gold membership      2 year
PL02 25       55000 Diamond membership   5 year
PL03 40       35000 silver membership    3 year
```

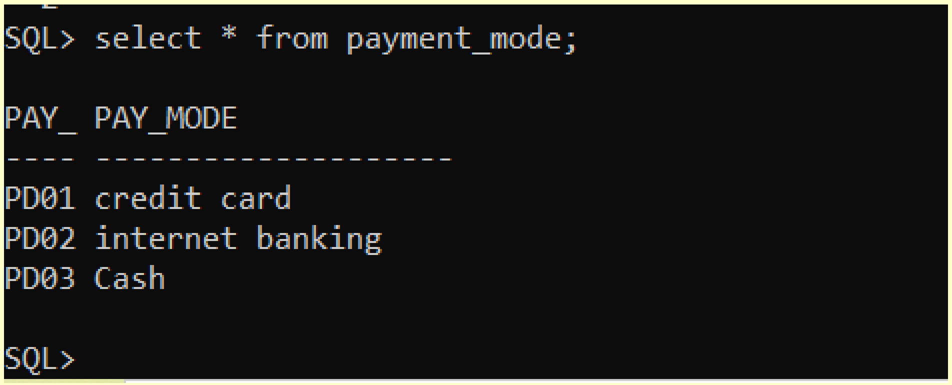
8. Member


```
SQL> select * from member;

MEM_ MEM_NAME             MEM_GEND    MEM_AGE  MEM_PHONE
---- -------------------- -------- ---------- ----------
MEM_EMAIL                                          TRN_ PLAN
-------------------------------------------------- ---- ----
MM01 Akshat               male             20 8986353688
akshat@gmail.com                                   TR01 PL01

MM02 Shubh                male             19 6780353688
shubh@gmail.com                                    TR02 PL02

MM03 Ritika               female           21 8986353772
rikika@gmail.com                                   TR03 PL03
```

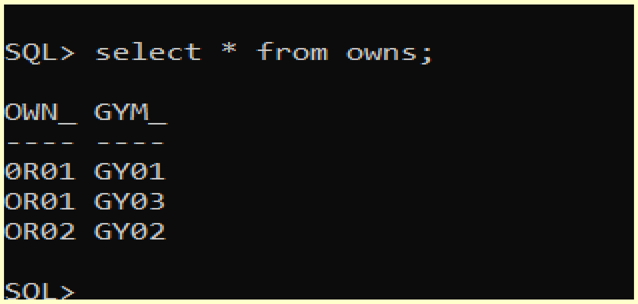
9. Payment


```
SQL> select * from payment;

PAY_ PAY_DATE     PAY_AMT PAY_MONTH            MEM_ MANG
---- --------- ---------- -------------------- ---- ----
PD01 22-OCT-18      40000 March                MM01 MG01
PD02 13-AUG-17      55000 June                 MM02 MG02
PD03 06-AUG-21      35000 October              MM03 MG03
```

10. Payment_mode

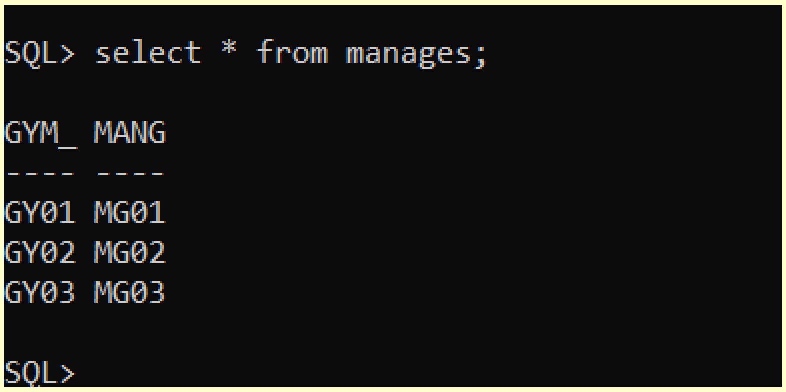

```
SQL> select * from payment_mode;

PAY_ PAY_MODE
---- --------------------
PD01 credit card
PD02 internet banking
PD03 Cash

SQL>
```

11. Owns



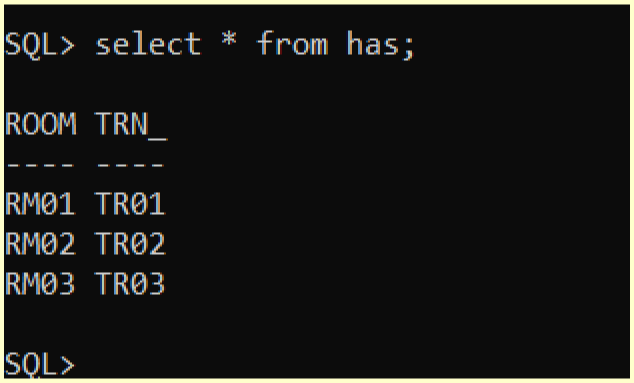
```
SQL> select * from owns;

OWN_ GYM_
---- ----
0R01 GY01
OR01 GY03
OR02 GY02

SQL>
```

12. Manages



```
SQL> select * from manages;

GYM_ MANG
---- ----
GY01 MG01
GY02 MG02
GY03 MG03

SQL>
```

13. Has



```
SQL> select * from has;

ROOM TRN_
---- ----
RM01 TR01
RM02 TR02
RM03 TR03

SQL>
```

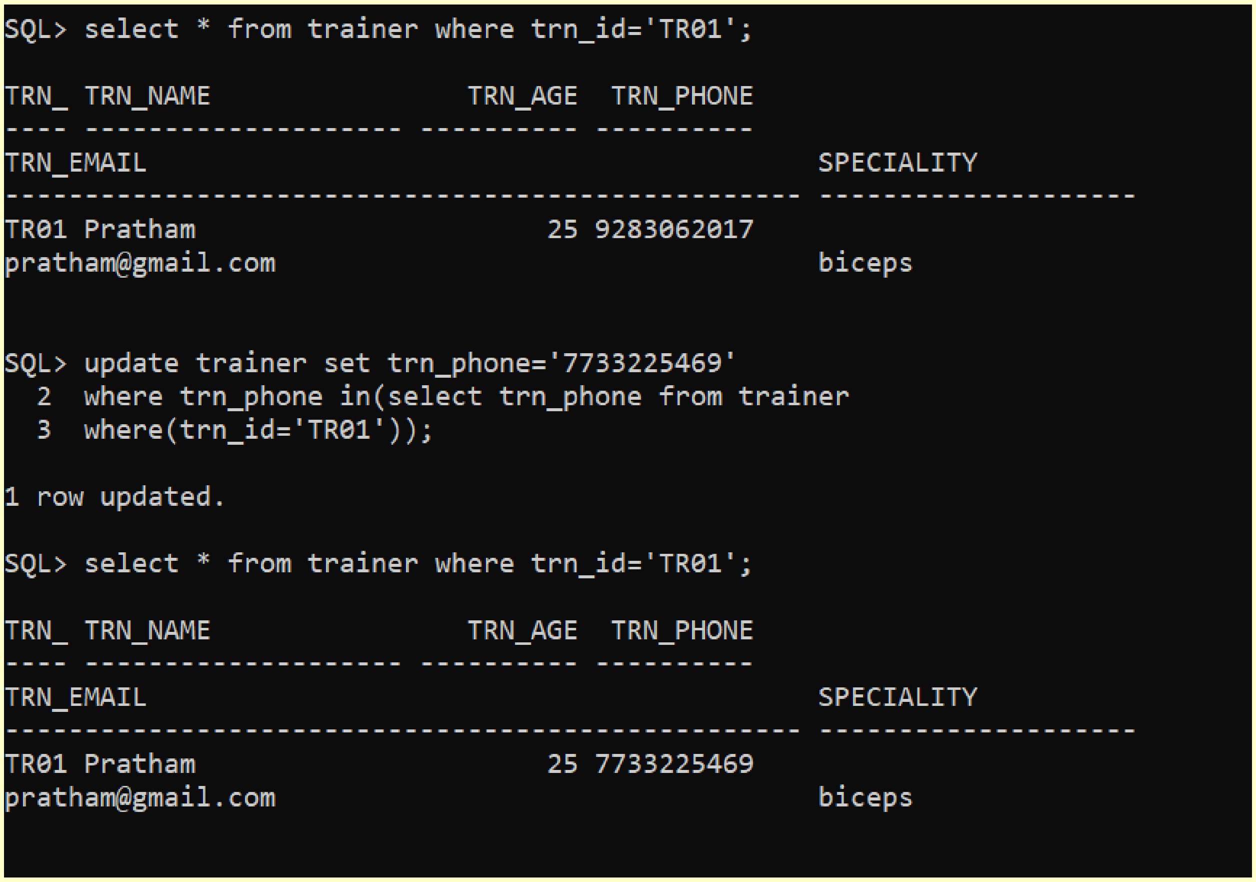
Modification Of Data:

1. **<u>To update the trainer phone number:</u>**

SQL> update trainer set trn_phone='7733225469'

 where trn_phone in(select trn_phone from trainer

 where(trn_id='TR01'));



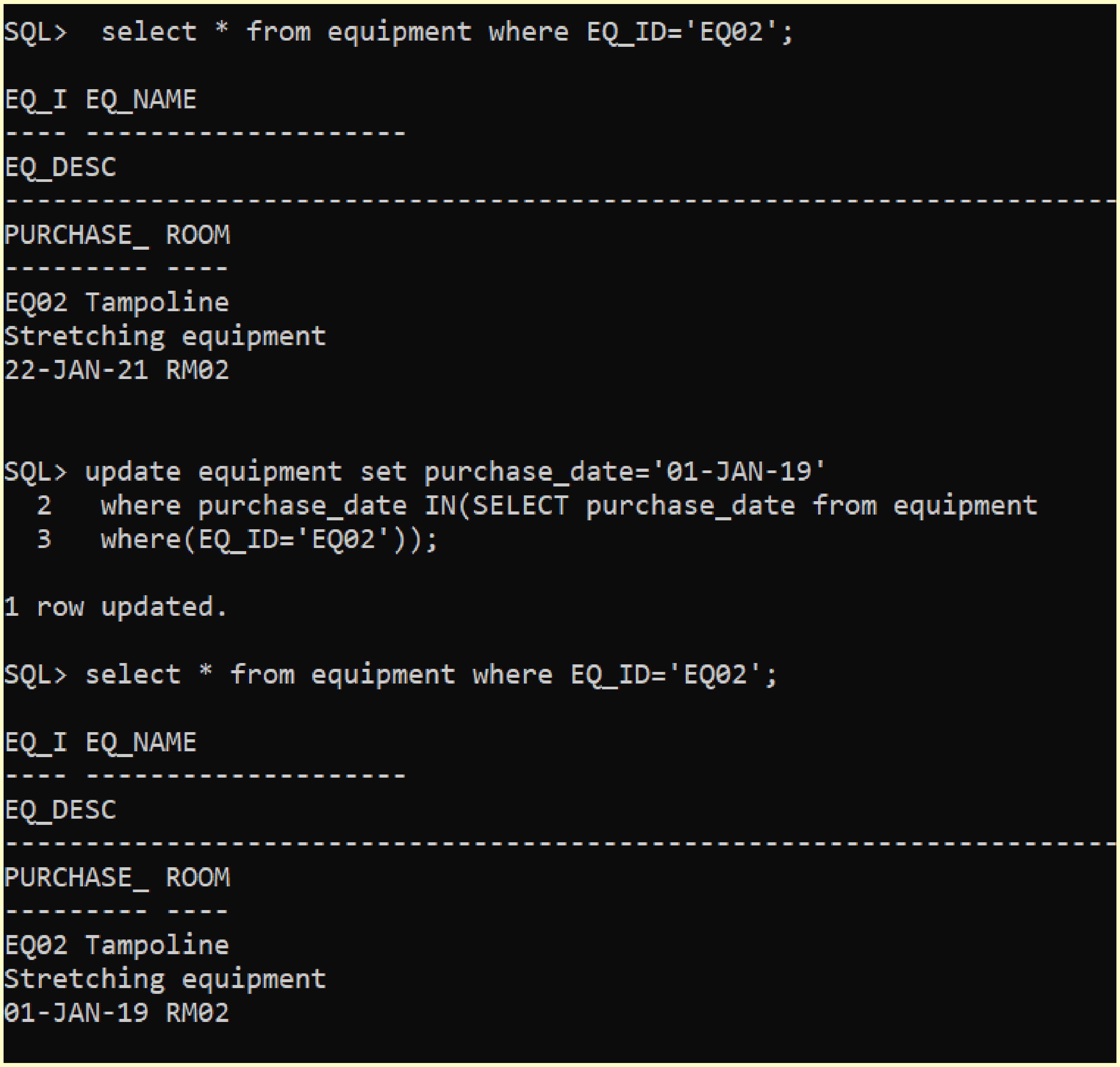
```
SQL> select * from trainer where trn_id='TR01';

TRN_ TRN_NAME                TRN_AGE  TRN_PHONE
---- -------------------- ---------- ----------
TRN_EMAIL                                          SPECIALITY
-------------------------------------------------- --------------------
TR01 Pratham                      25 9283062017
pratham@gmail.com                                  biceps


SQL> update trainer set trn_phone='7733225469'
  2  where trn_phone in(select trn_phone from trainer
  3  where(trn_id='TR01'));

1 row updated.

SQL> select * from trainer where trn_id='TR01';

TRN_ TRN_NAME                TRN_AGE  TRN_PHONE
---- -------------------- ---------- ----------
TRN_EMAIL                                          SPECIALITY
-------------------------------------------------- --------------------
TR01 Pratham                      25 7733225469
pratham@gmail.com                                  biceps
```

2. **<u>To change the equipment purchase date:</u>**

SQL> update equipment set purchase_date='01-JAN-19'

 where purchase_date IN(SELECT purchase_date from equipment

 where(EQ_ID='EQ02'));

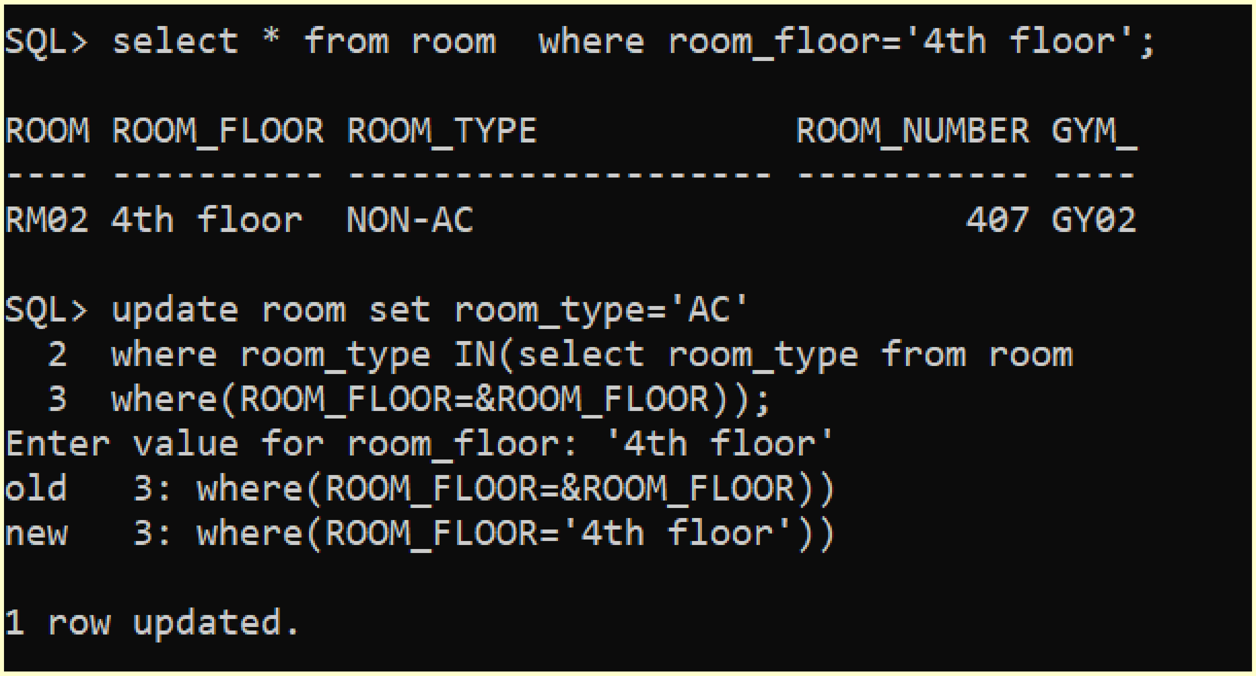


```
SQL>  select * from equipment where EQ_ID='EQ02';

EQ_I EQ_NAME
---- --------------------
EQ_DESC
--------------------------------------------------------------------
PURCHASE_ ROOM
--------- ----
EQ02 Tampoline
Stretching equipment
22-JAN-21 RM02


SQL> update equipment set purchase_date='01-JAN-19'
  2   where purchase_date IN(SELECT purchase_date from equipment
  3   where(EQ_ID='EQ02'));

1 row updated.

SQL> select * from equipment where EQ_ID='EQ02';

EQ_I EQ_NAME
---- --------------------
EQ_DESC
--------------------------------------------------------------------
PURCHASE_ ROOM
--------- ----
EQ02 Tampoline
Stretching equipment
01-JAN-19 RM02
```

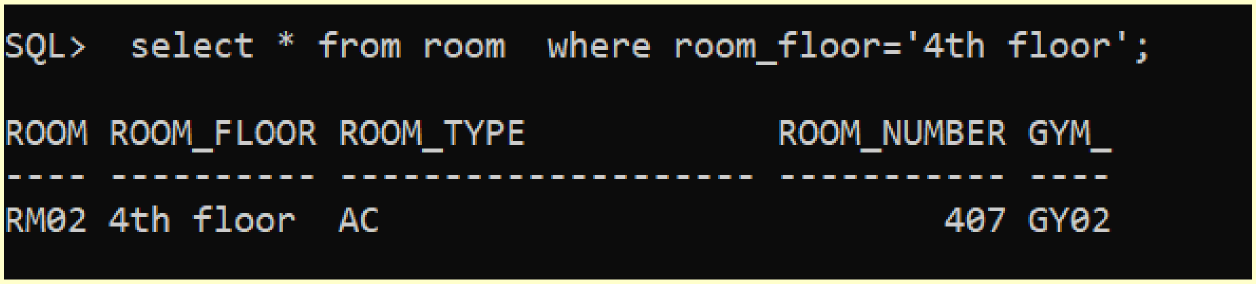
3. **To update the room type after the renovation of Gym:**

```
SQL> update room set room_type='AC'

   where room_type IN(select room_type from room

  where(ROOM_FLOOR=&ROOM_FLOOR));

Enter value for room_floor: '4th floor'

old   3: where(ROOM_FLOOR=&ROOM_FLOOR))
```

new   3: where(ROOM_FLOOR='4th floor'))



```
SQL> select * from room  where room_floor='4th floor';

ROOM ROOM_FLOOR ROOM_TYPE            ROOM_NUMBER GYM_
---- ---------- -------------------- ----------- ----
RM02 4th floor  NON-AC                       407 GY02

SQL> update room set room_type='AC'
  2  where room_type IN(select room_type from room
  3  where(ROOM_FLOOR=&ROOM_FLOOR));
Enter value for room_floor: '4th floor'
old   3: where(ROOM_FLOOR=&ROOM_FLOOR))
new   3: where(ROOM_FLOOR='4th floor'))

1 row updated.
```



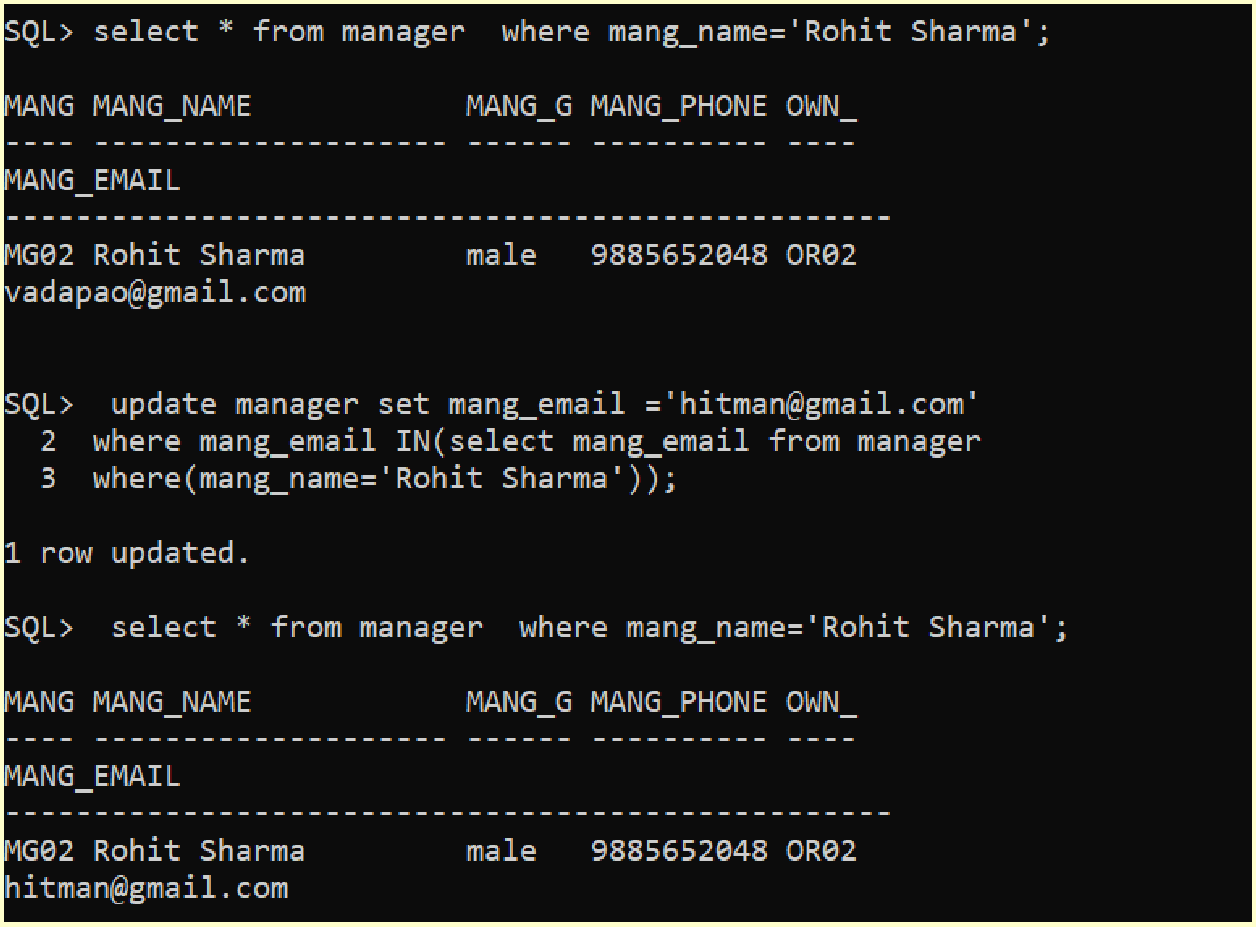
```
SQL>  select * from room  where room_floor='4th floor';

ROOM ROOM_FLOOR ROOM_TYPE            ROOM_NUMBER GYM_
---- ---------- -------------------- ----------- ----
RM02 4th floor  AC                           407 GY02
```

4. **To Change the manager email address**:

SQL> update manager set mang_email ='hitman@gmail.com'

where mang_email IN(select mang_email from manager

where(mang_name='Rohit Sharma'));



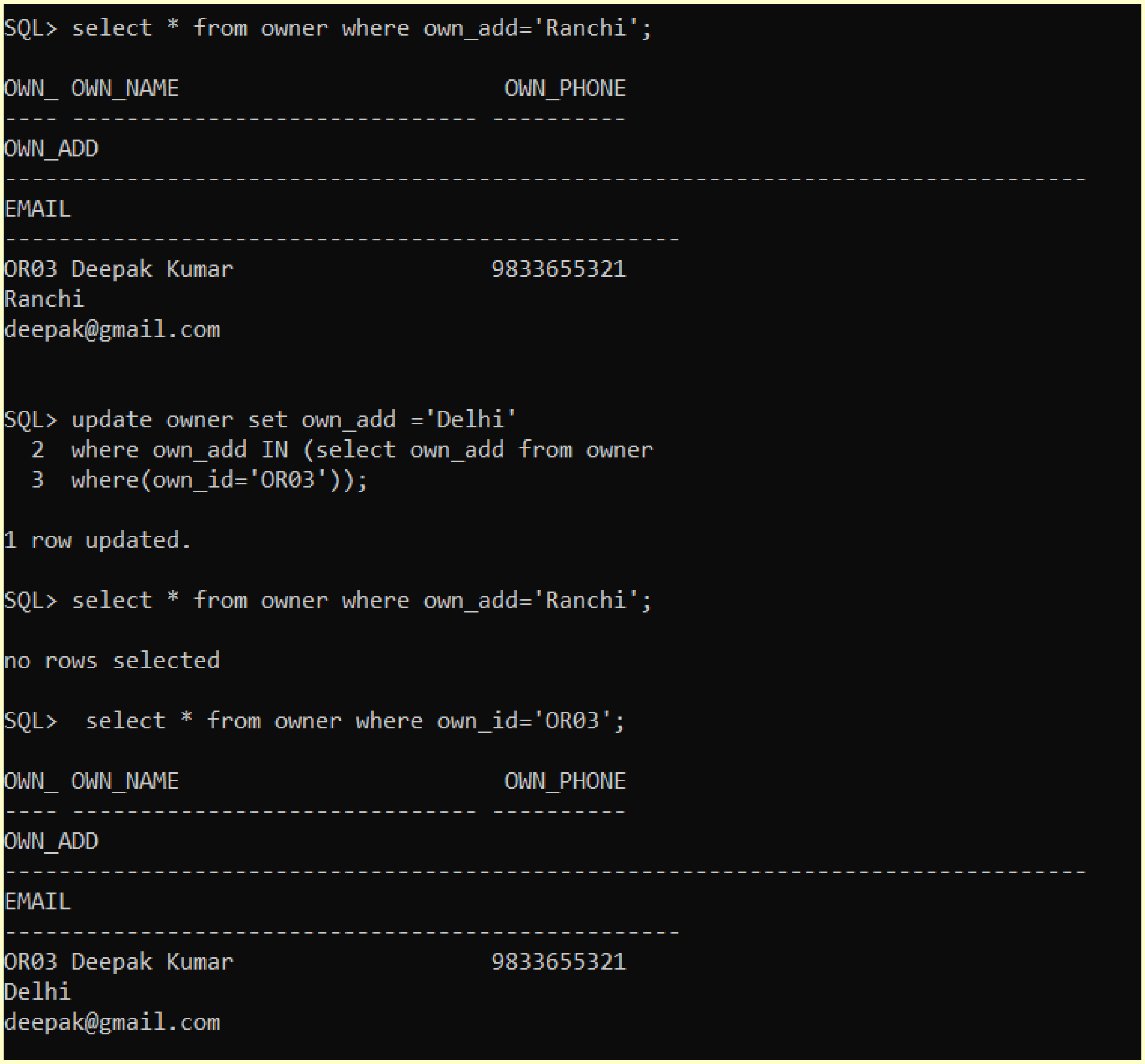
```
SQL> select * from manager  where mang_name='Rohit Sharma';

MANG MANG_NAME            MANG_G MANG_PHONE OWN_
---- -------------------- ------ ---------- ----
MANG_EMAIL
--------------------------------------------------
MG02 Rohit Sharma         male   9885652048 OR02
vadapao@gmail.com


SQL>  update manager set mang_email ='hitman@gmail.com'
  2  where mang_email IN(select mang_email from manager
  3  where(mang_name='Rohit Sharma'));

1 row updated.

SQL>  select * from manager  where mang_name='Rohit Sharma';

MANG MANG_NAME            MANG_G MANG_PHONE OWN_
---- -------------------- ------ ---------- ----
MANG_EMAIL
--------------------------------------------------
MG02 Rohit Sharma         male   9885652048 OR02
hitman@gmail.com
```

5. <u>To update owner address:</u>

SQL> update owner set own_add ='Delhi'

where own_add IN (select own_add from owner

where(own_id='OR03'));



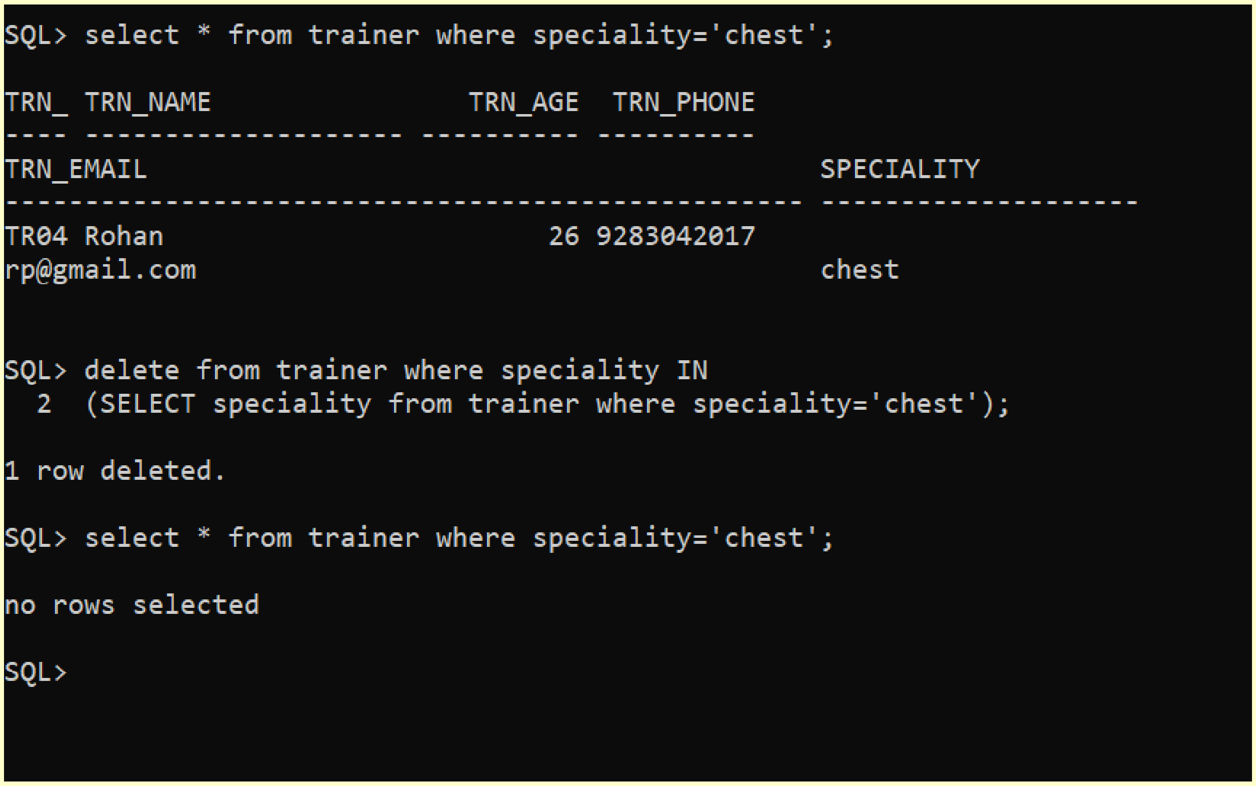
```
SQL> select * from owner where own_add='Ranchi';

OWN_ OWN_NAME                        OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------
OR03 Deepak Kumar                   9833655321
Ranchi
deepak@gmail.com


SQL> update owner set own_add ='Delhi'
  2  where own_add IN (select own_add from owner
  3  where(own_id='OR03'));

1 row updated.

SQL> select * from owner where own_add='Ranchi';

no rows selected

SQL>  select * from owner where own_id='OR03';

OWN_ OWN_NAME                        OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------
OR03 Deepak Kumar                   9833655321
Delhi
deepak@gmail.com
```

<u>Deletion of Data:</u>

i) <u>To remove the trainer:</u>

SQL> delete from trainer where speciality IN

  2 (SELECT speciality from trainer where speciality='chest');

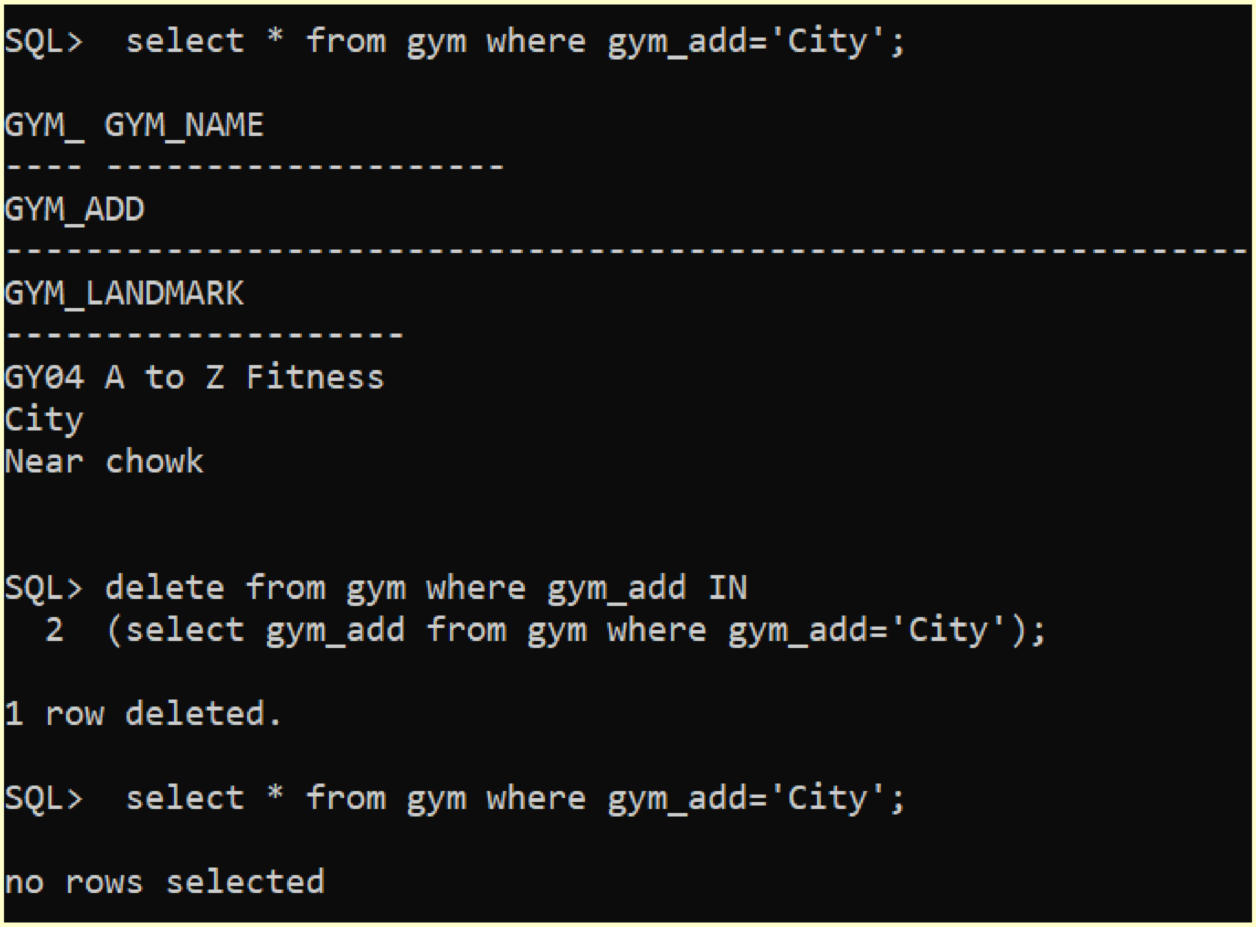


```
SQL> select * from trainer where speciality='chest';

TRN_ TRN_NAME                TRN_AGE  TRN_PHONE
---- -------------------- ---------- ----------
TRN_EMAIL                                          SPECIALITY
-------------------------------------------------- --------------------
TR04 Rohan                        26 9283042017
rp@gmail.com                                       chest


SQL> delete from trainer where speciality IN
  2  (SELECT speciality from trainer where speciality='chest');

1 row deleted.

SQL> select * from trainer where speciality='chest';

no rows selected

SQL>
```

ii) **<u>To Remove the gym old address</u>:**

SQL> delete from gym where gym_add IN

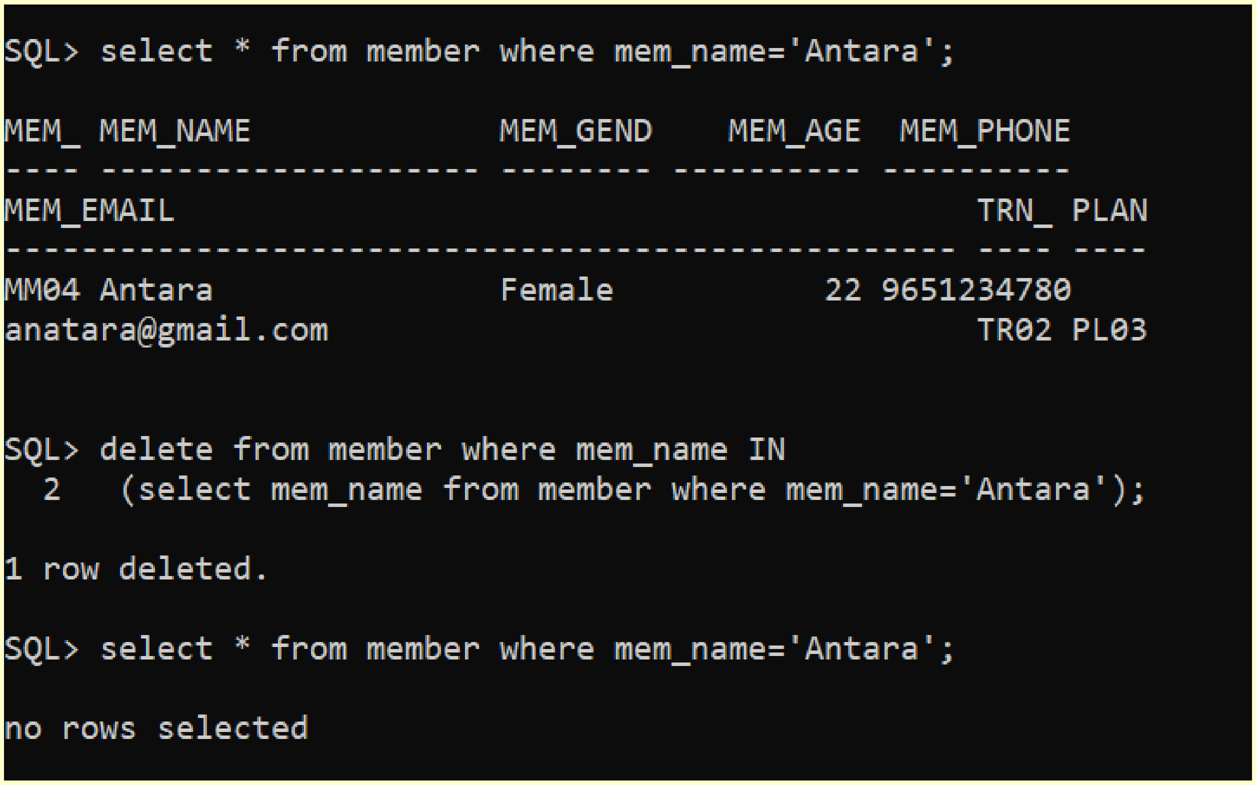
   2 (select gym_add from gym where gym_add='City');



```
SQL>  select * from gym where gym_add='City';

GYM_ GYM_NAME
---- --------------------
GYM_ADD
--------------------------------------------------------------
GYM_LANDMARK
--------------------
GY04 A to Z Fitness
City
Near chowk


SQL> delete from gym where gym_add IN
  2  (select gym_add from gym where gym_add='City');

1 row deleted.

SQL>  select * from gym where gym_add='City';

no rows selected
```

iii) <u>To remove a member whose membership period is over</u>:

```
SQL> delete from member where mem_name IN

    2   (select mem_name from member where mem_name='Antara');
```

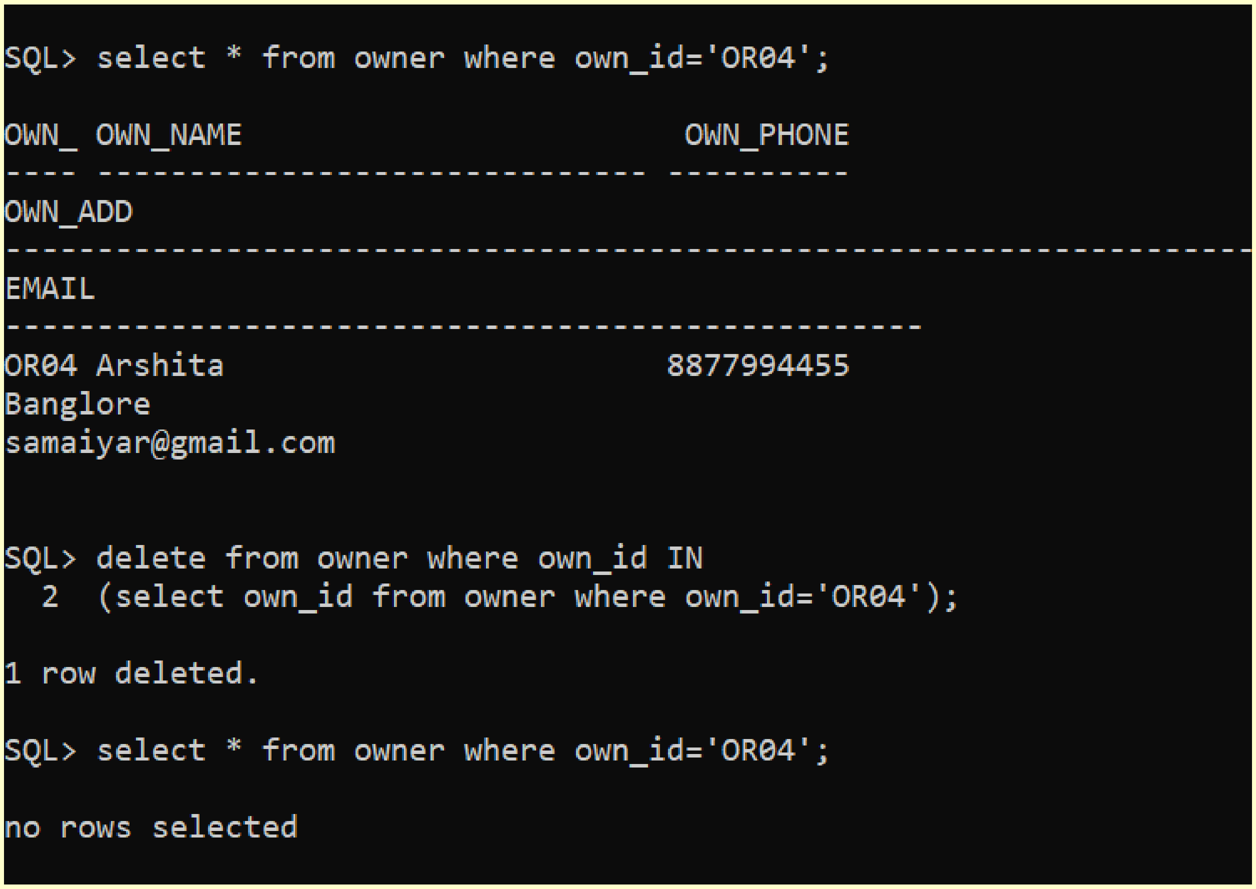


```
SQL> select * from member where mem_name='Antara';

MEM_ MEM_NAME             MEM_GEND    MEM_AGE  MEM_PHONE
---- -------------------- -------- ---------- ----------
MEM_EMAIL                                          TRN_ PLAN
-------------------------------------------------- ---- ----
MM04 Antara               Female           22 9651234780
anatara@gmail.com                                  TR02 PL03


SQL> delete from member where mem_name IN
  2   (select mem_name from member where mem_name='Antara');

1 row deleted.

SQL> select * from member where mem_name='Antara';

no rows selected
```

iv)<u>Removal of details of an owner:</u>

SQL> delete from owner where own_id IN

   2 (select own_id from owner where own_id='OR04';



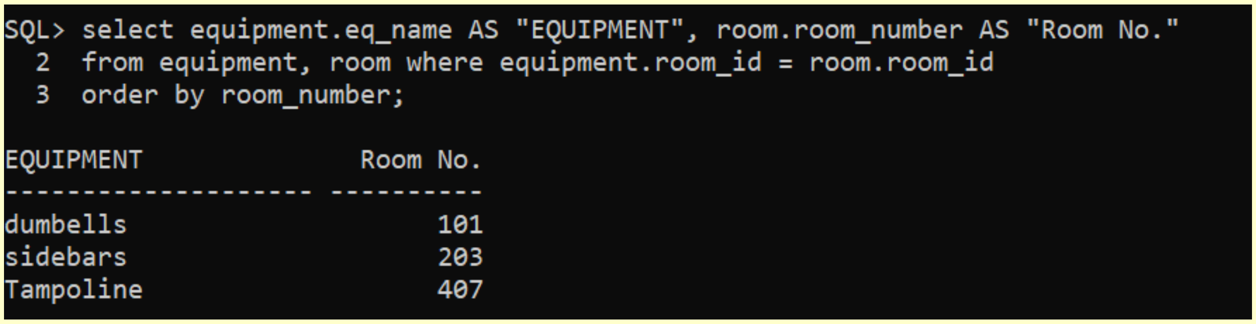
```
SQL> select * from owner where own_id='OR04';

OWN_ OWN_NAME                        OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------
OR04 Arshita                        8877994455
Banglore
samaiyar@gmail.com


SQL> delete from owner where own_id IN
  2  (select own_id from owner where own_id='OR04');

1 row deleted.

SQL> select * from owner where own_id='OR04';

no rows selected
```

<u>SELECT QUERIES or Retrieval Of Data:</u>

<u>Join with order by</u>

To display which equipment belongs to which room and ordering them by room number.

1. select equipment.eq_name AS "EQUIPMENT", room.room_number AS "Room No." from equipment, room where equipment.room_id = room.room_id order by room_number;

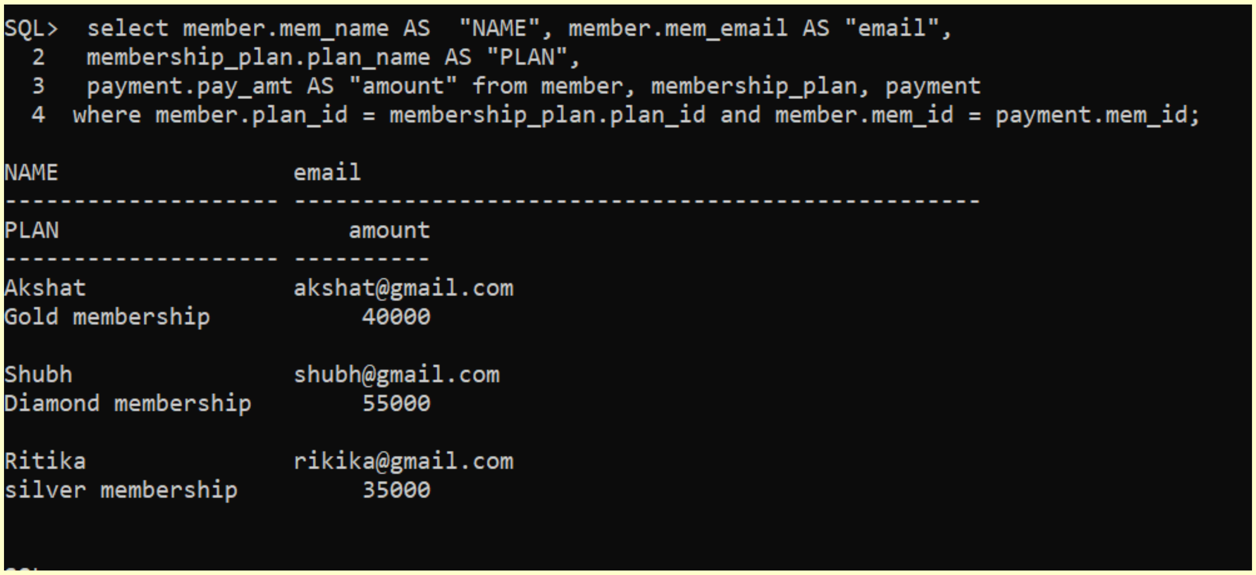


```
SQL> select equipment.eq_name AS "EQUIPMENT", room.room_number AS "Room No."
  2  from equipment, room where equipment.room_id = room.room_id
  3  order by room_number;

EQUIPMENT             Room No.
-------------------- ----------
dumbells                   101
sidebars                   203
Tampoline                  407
```

Join query

To display members name, email, gym plan under which they belong and the membership amount.

2. select member.mem_name AS "NAME", member.mem_email AS "email", membership_plan.plan_name AS "PLAN", payment.pay_amt AS "amount" from member, membership_plan, payment where member.plan_id = membership_plan.plan_id and member.mem_id = payment.mem_id;

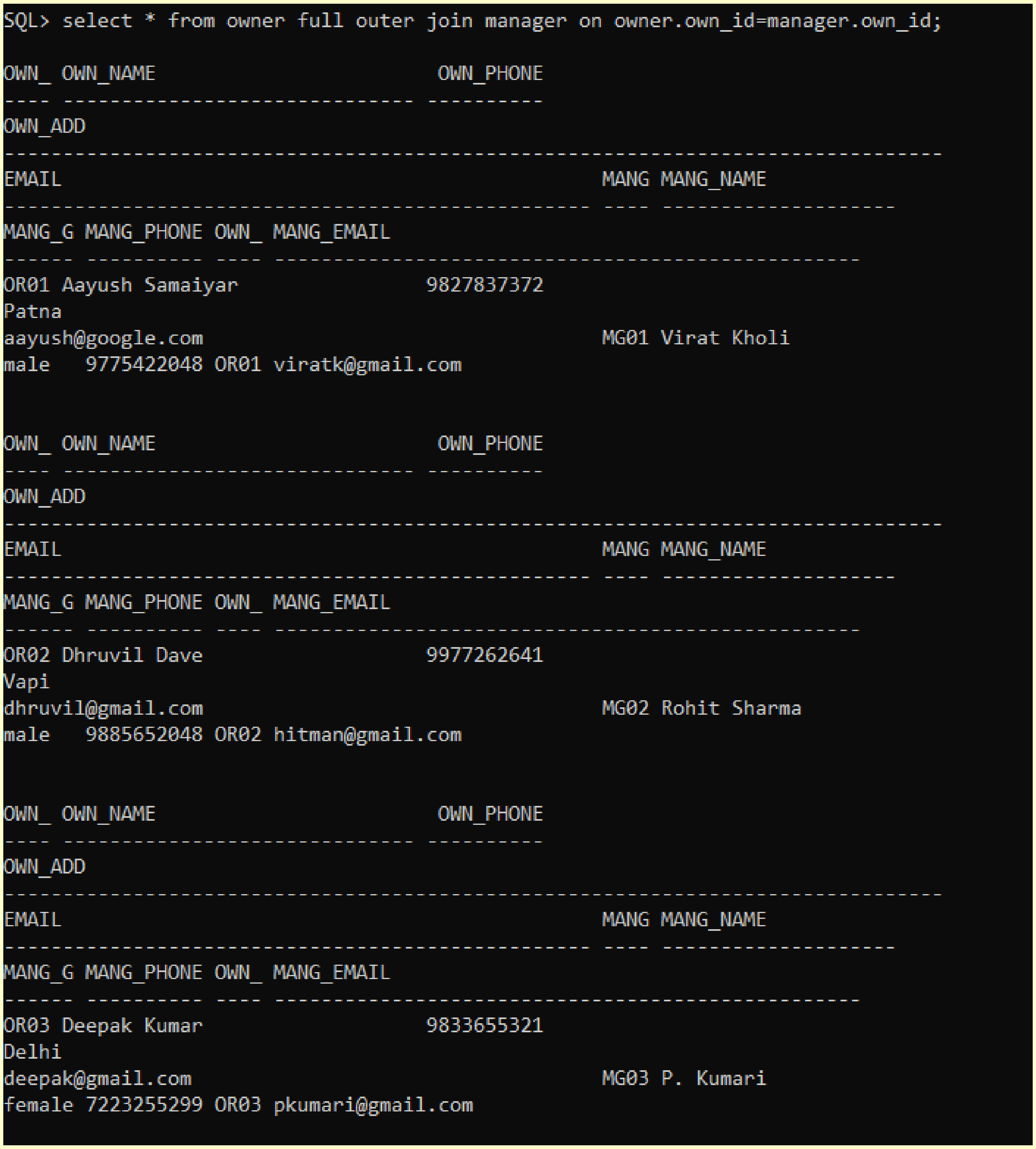


```
SQL>  select member.mem_name AS  "NAME", member.mem_email AS "email",
  2   membership_plan.plan_name AS "PLAN",
  3   payment.pay_amt AS "amount" from member, membership_plan, payment
  4  where member.plan_id = membership_plan.plan_id and member.mem_id = payment.mem_id;

NAME                 email
-------------------- --------------------------------------------------
PLAN                     amount
-------------------- ----------
Akshat               akshat@gmail.com
Gold membership           40000

Shubh                shubh@gmail.com
Diamond membership        55000

Ritika               rikika@gmail.com
silver membership         35000
```

Outer join

To display owner and manager details using outer join query.

3. select * from owner full outer join manager on owner.own_id=manager.own_id;

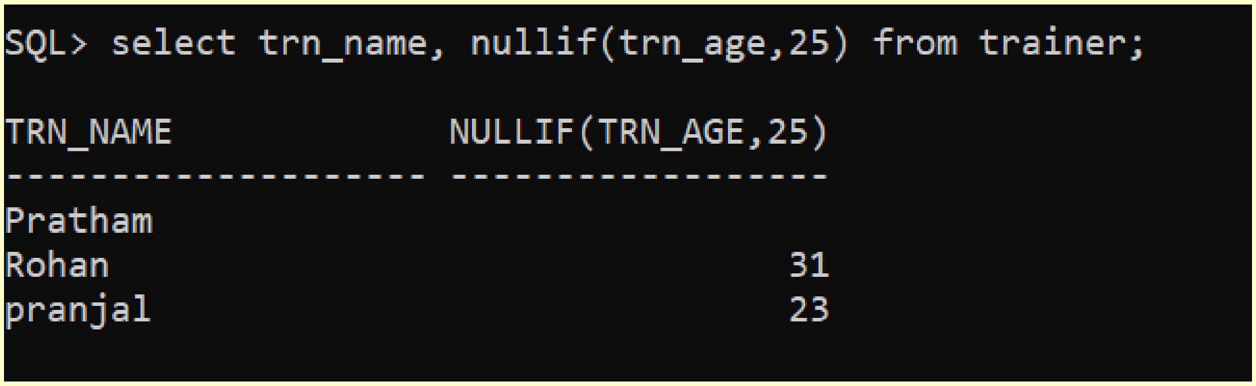


```
SQL> select * from owner full outer join manager on owner.own_id=manager.own_id;

OWN_ OWN_NAME                       OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL                                              MANG MANG_NAME
-------------------------------------------------- ---- --------------------
MANG_G MANG_PHONE OWN_ MANG_EMAIL
------ ---------- ---- --------------------------------------------------
OR01 Aayush Samaiyar                 9827837372
Patna
aayush@google.com                                  MG01 Virat Kholi
male   9775422048 OR01 viratk@gmail.com


OWN_ OWN_NAME                       OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL                                              MANG MANG_NAME
-------------------------------------------------- ---- --------------------
MANG_G MANG_PHONE OWN_ MANG_EMAIL
------ ---------- ---- --------------------------------------------------
OR02 Dhruvil Dave                    9977262641
Vapi
dhruvil@gmail.com                                  MG02 Rohit Sharma
male   9885652048 OR02 hitman@gmail.com


OWN_ OWN_NAME                       OWN_PHONE
---- ------------------------------ ----------
OWN_ADD
--------------------------------------------------------------------------------
EMAIL                                              MANG MANG_NAME
-------------------------------------------------- ---- --------------------
MANG_G MANG_PHONE OWN_ MANG_EMAIL
------ ---------- ---- --------------------------------------------------
OR03 Deepak Kumar                    9833655321
Delhi
deepak@gmail.com                                   MG03 P. Kumari
female 7223255299 OR03 pkumari@gmail.com
```

Query using Nullif function

To display trainer name and age, and if the trainer age is 25 then remove the trainer age.

4. select trn_name, nullif(trn_age,25) from trainer;

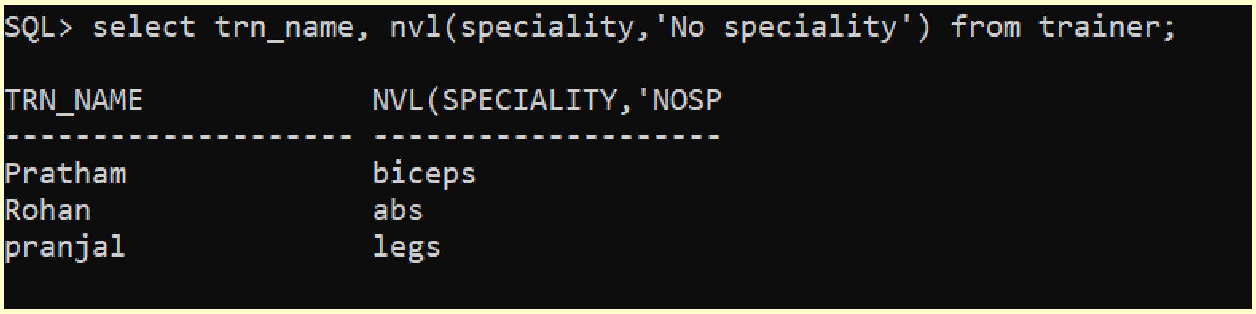


```
SQL> select trn_name, nullif(trn_age,25) from trainer;

TRN_NAME             NULLIF(TRN_AGE,25)
-------------------- ------------------
Pratham
Rohan                                31
pranjal                              23
```

Query using Nvl function

To display trainer name and his speciality, and if there is no speciality then display as no speciality mentioned.

5. select trn_name, nvl(speciality,'No speciality') from trainer;

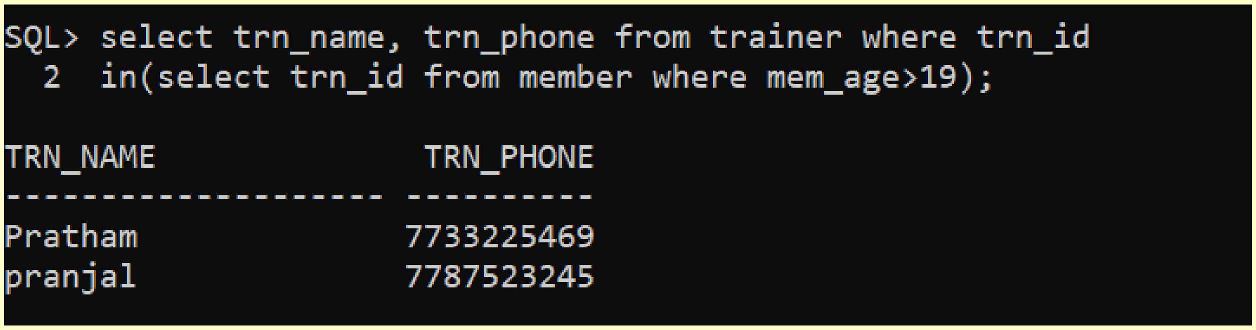


```
SQL> select trn_name, nvl(speciality,'No speciality') from trainer;

TRN_NAME             NVL(SPECIALITY,'NOSP
-------------------- --------------------
Pratham              biceps
Rohan                abs
pranjal              legs
```

Uncorrelated nested query

To display the trainer name, phone appointed for all members who are above the age of 19

6. select trn_name, trn_phone from trainer where trn_id in(select trn_id from member where mem_age>19);

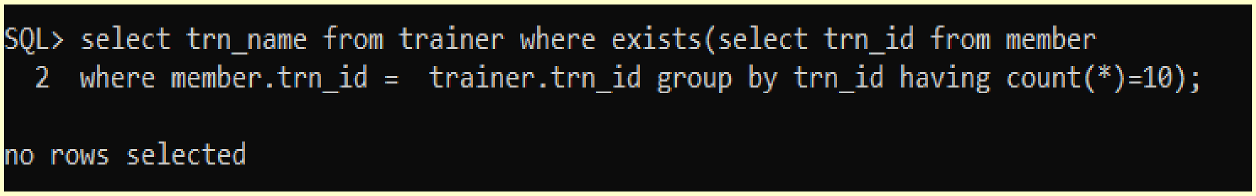


```
SQL> select trn_name, trn_phone from trainer where trn_id
  2  in(select trn_id from member where mem_age>19);

TRN_NAME               TRN_PHONE
--------------------- ----------
Pratham               7733225469
pranjal               7787523245
```

Correlated nested query

Display all the trainer names who have 10 members assigned under them.

7. select trn_name from trainer where exists(select trn_id from member where member.trn_id = trainer.trn_id group by trn_id having count(*)=10);



```
SQL> select trn_name from trainer where exists(select trn_id from member
  2  where member.trn_id =  trainer.trn_id group by trn_id having count(*)=10);

no rows selected
```

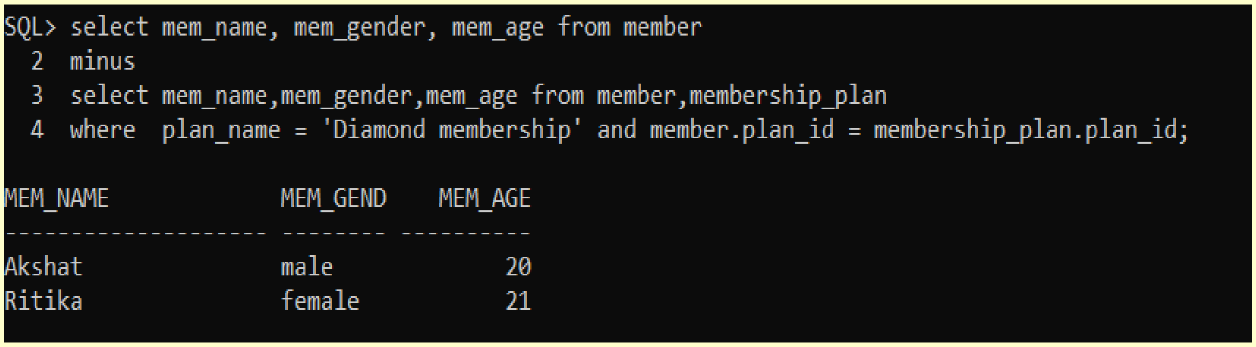
Set query

Display member name, gender and age who have diamond membership plan.

8. select mem_name, mem_gender, mem_age from member

   minus

   select mem_name,mem_gender,mem_age from member,membership_plan where plan_name = 'Diamond membership' and member.plan_id = membership_plan.plan_id;

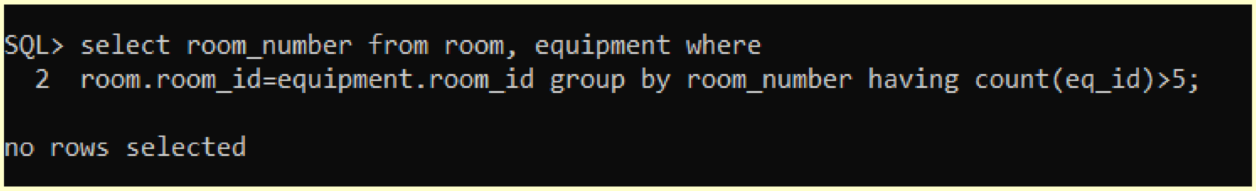


```
SQL> select mem_name, mem_gender, mem_age from member
  2  minus
  3  select mem_name,mem_gender,mem_age from member,membership_plan
  4  where  plan_name = 'Diamond membership' and member.plan_id = membership_plan.plan_id;

MEM_NAME             MEM_GEND    MEM_AGE
-------------------- -------- ----------
Akshat               male             20
Ritika               female           21
```

Query involving group by, having and where clause

Display the room number that has more than five equipment.

9. select room_number from room, equipment where room.room_id=equipment.room_id group by room_number having count(eq_id)>5;



```
SQL> select room_number from room, equipment where
  2  room.room_id=equipment.room_id group by room_number having count(eq_id)>5;

no rows selected
```

PL/SQL PROCEDURES:

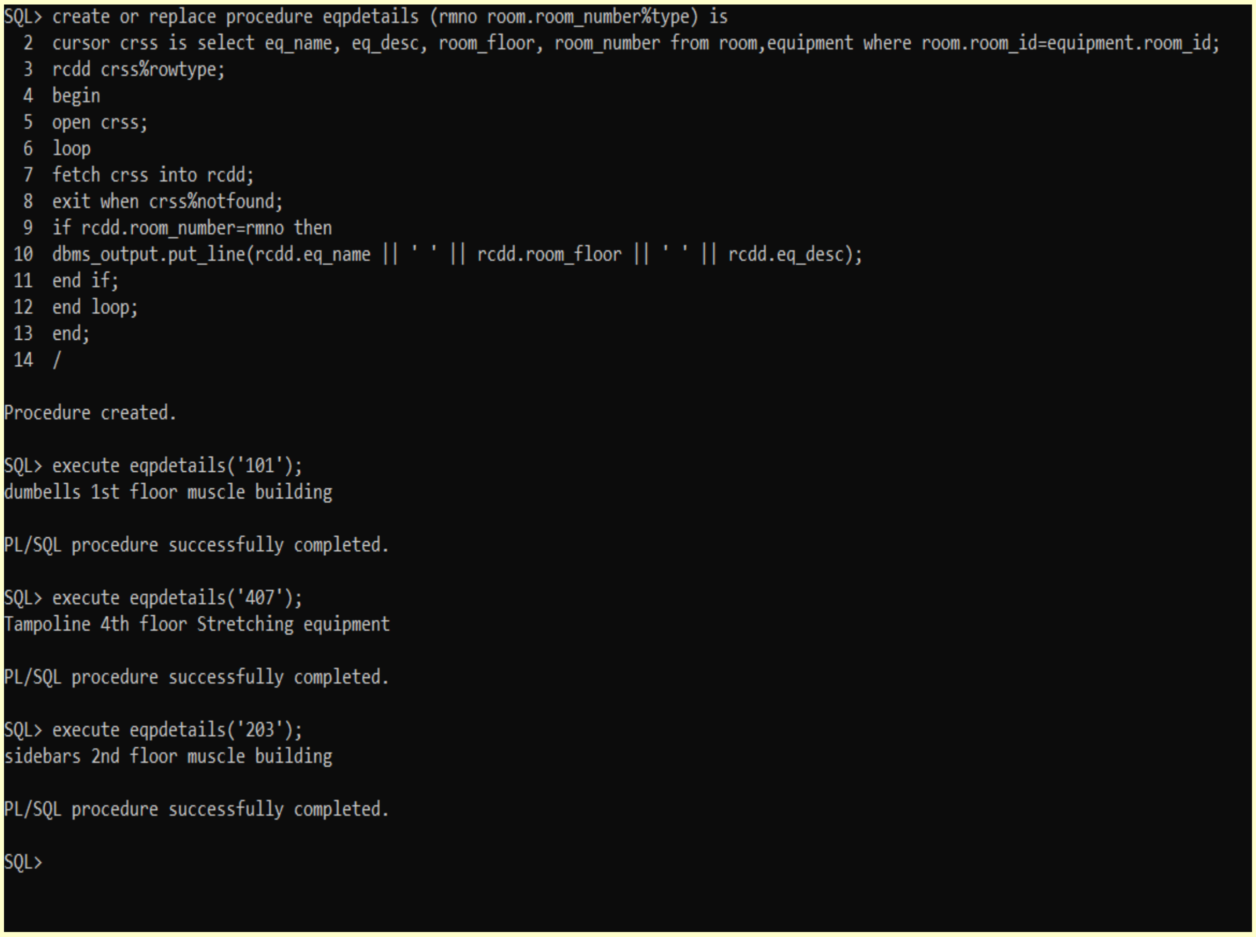
1. Display equipment details and room floor of a given room number

- PL SQL procedure for this particular problem is as follows -

SQL> create or replace procedure eqpdetails (rmno room.room_number%type) is

  2 cursor crss is select eq_name, eq_desc, room_floor, room_number from room,equipment where room.room_id=equipment.room_id;

```
rcdd crss%rowtype;
begin
open crss;
loop
fetch crss into rcdd;
exit when crss%notfound;
if rcdd.room_number=rmno then
dbms_output.put_line(rcdd.eq_name || ' ' || rcdd.room_floor || ' ' || rcdd.eq_desc);
end if;
end loop;
end;
/
```



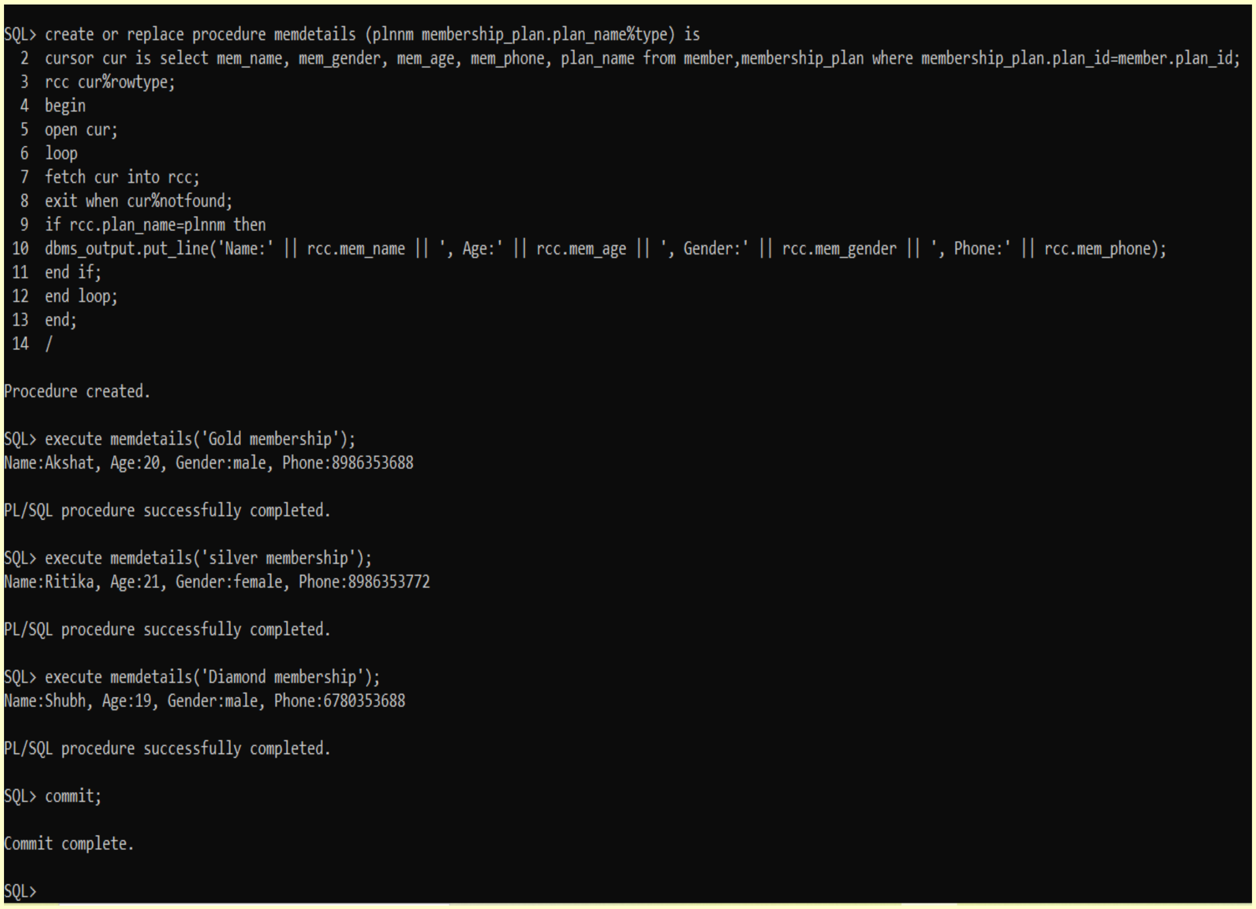
```
SQL> create or replace procedure eqpdetails (rmno room.room_number%type) is
cursor crss is select eq_name, eq_desc, room_floor, room_number from room,equipment where room.room_id=equipment.room_id;
rcdd crss%rowtype;
begin
open crss;
loop
fetch crss into rcdd;
exit when crss%notfound;
if rcdd.room_number=rmno then
dbms_output.put_line(rcdd.eq_name || ' ' || rcdd.room_floor || ' ' || rcdd.eq_desc);
end if;
end loop;
end;
/

Procedure created.

SQL> execute eqpdetails('101');
dumbells 1st floor muscle building

PL/SQL procedure successfully completed.

SQL> execute eqpdetails('407');
Tampoline 4th floor Stretching equipment

PL/SQL procedure successfully completed.

SQL> execute eqpdetails('203');
sidebars 2nd floor muscle building

PL/SQL procedure successfully completed.

SQL>
```

2. Displaying Member Details for a given membership plan by its name

- PL SQL procedure for this particular problem is as follows -

```
SQL> create or replace procedure memdetails (plnnm membership_plan.plan_name%type) is
cursor cur is select mem_name, mem_gender, mem_age, mem_phone, plan_name from member,membership_plan where membership_plan.plan_id=member.plan_id;
rcc cur%rowtype;
```

```
begin
open cur;
loop
fetch cur into rcc;
exit when cur%notfound;
if rcc.plan_name=plnnm then
dbms_output.put_line('Name:' || rcc.mem_name || ', Age:' || rcc.mem_age || ', Gender:' || rcc.mem_gender || ', Phone:' || rcc.mem_phone);
end if;
end loop;
end;
/
```



```
SQL> create or replace procedure memdetails (plnnm membership_plan.plan_name%type) is
cursor cur is select mem_name, mem_gender, mem_age, mem_phone, plan_name from member,membership_plan where membership_plan.plan_id=member.plan_id;
rcc cur%rowtype;
begin
open cur;
loop
fetch cur into rcc;
exit when cur%notfound;
if rcc.plan_name=plnnm then
dbms_output.put_line('Name:' || rcc.mem_name || ', Age:' || rcc.mem_age || ', Gender:' || rcc.mem_gender || ', Phone:' || rcc.mem_phone);
end if;
end loop;
end;
/

Procedure created.

SQL> execute memdetails('Gold membership');
Name:Akshat, Age:20, Gender:male, Phone:8986353688

PL/SQL procedure successfully completed.

SQL> execute memdetails('silver membership');
Name:Ritika, Age:21, Gender:female, Phone:8986353772

PL/SQL procedure successfully completed.

SQL> execute memdetails('Diamond membership');
Name:Shubh, Age:19, Gender:male, Phone:6780353688

PL/SQL procedure successfully completed.

SQL> commit;

Commit complete.

SQL>
```

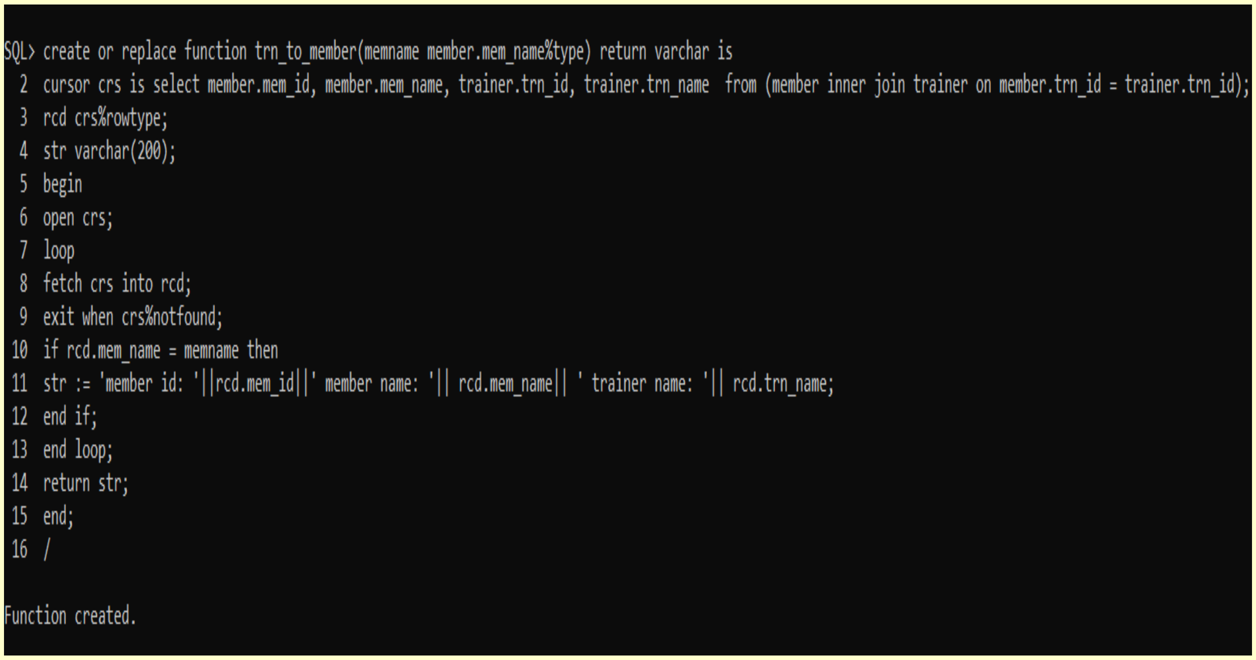
PL/SQL FUNCTIONS:

1. Function to find the trainer name allocated to a member.

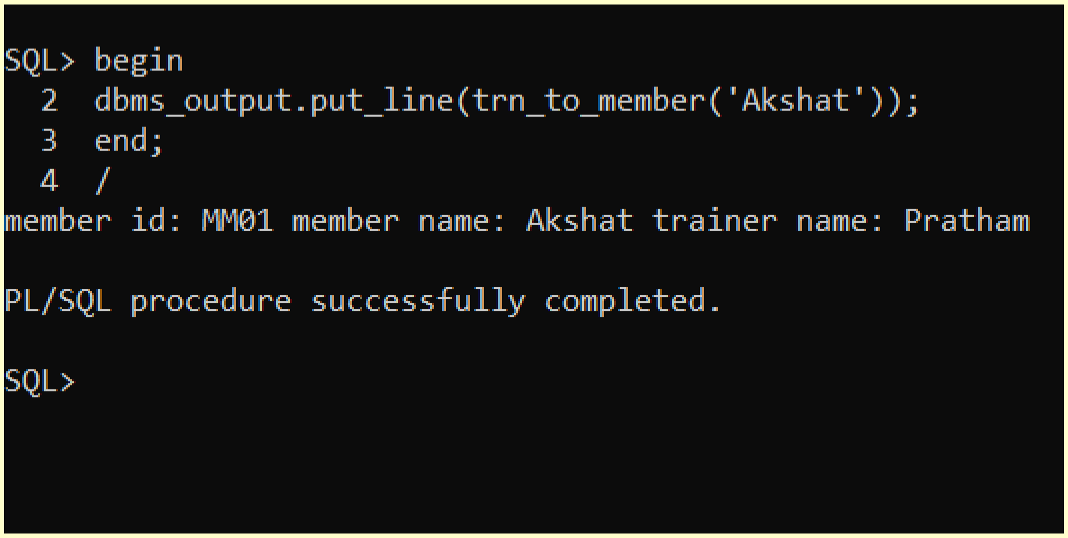
SQL> create or replace function trn_to_member(memname member.mem_name%type) return varchar is

2 cursor crs is select member.mem_id, member.mem_name, trainer.trn_id, trainer.trn_name from (member inner join trainer on member.trn_id = trainer.trn_id);

```
rcd crs%rowtype;
str varchar(200);
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.mem_name = memname then
str := 'member id: '||rcd.mem_id||' member name: '|| rcd.mem_name|| ' trainer name: '|| rcd.trn_name;
end if;
end loop;
return str;
end;
/
```



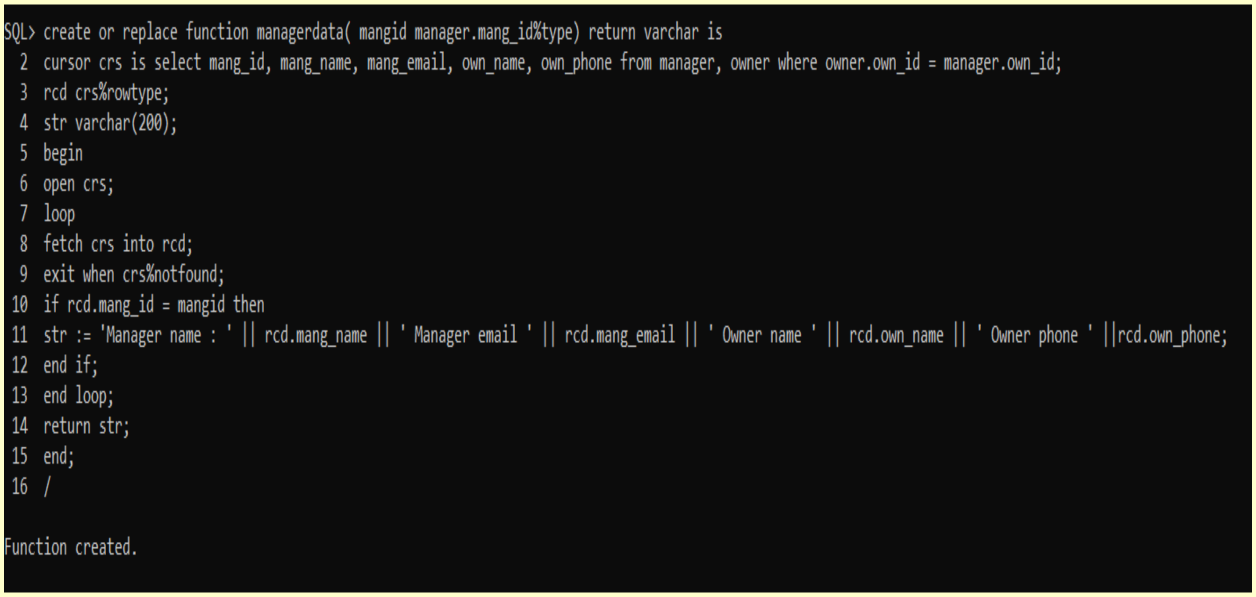
```
SQL> create or replace function trn_to_member(memname member.mem_name%type) return varchar is
cursor crs is select member.mem_id, member.mem_name, trainer.trn_id, trainer.trn_name  from (member inner join trainer on member.trn_id = trainer.trn_id);
rcd crs%rowtype;
str varchar(200);
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.mem_name = memname then
str := 'member id: '||rcd.mem_id||' member name: '|| rcd.mem_name|| ' trainer name: '|| rcd.trn_name;
end if;
end loop;
return str;
end;
/

Function created.
```



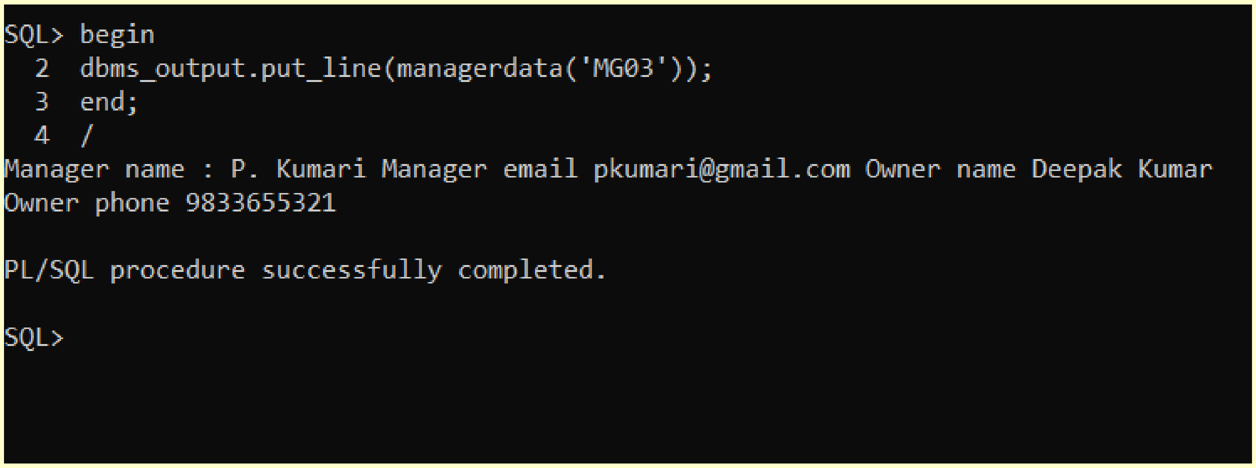
```
SQL> begin
dbms_output.put_line(trn_to_member('Akshat'));
end;
/
member id: MM01 member name: Akshat trainer name: Pratham

PL/SQL procedure successfully completed.

SQL>
```

2. Function to get the manager details using the owner 's name.

```
create or replace function managerdata( mangid manager.mang_id%type) return varchar is

 cursor crs is select mang_id, mang_name, mang_email, own_name, own_phone from manager, owner where owner.own_id = manager.own_id;

 rcd crs%rowtype;

 str varchar(200);

 begin

 open crs;

 loop

 fetch crs into rcd;

 exit when crs%notfound;

 if rcd.mang_id = mangid then

 str := 'Manager name : ' || rcd.mang_name || ' Manager email ' || rcd.mang_email || ' Owner name ' || rcd.own_name || ' Owner phone ' ||rcd.own_phone;

 end if;

 end loop;

 return str;

 end;
```



```
SQL> create or replace function managerdata( mangid manager.mang_id%type) return varchar is
cursor crs is select mang_id, mang_name, mang_email, own_name, own_phone from manager, owner where owner.own_id = manager.own_id;
rcd crs%rowtype;
str varchar(200);
begin
open crs;
loop
fetch crs into rcd;
exit when crs%notfound;
if rcd.mang_id = mangid then
str := 'Manager name : ' || rcd.mang_name || ' Manager email ' || rcd.mang_email || ' Owner name ' || rcd.own_name || ' Owner phone ' ||rcd.own_phone;
end if;
end loop;
return str;
end;
/

Function created.
```



```
SQL> begin
dbms_output.put_line(managerdata('MG03'));
end;
/
Manager name : P. Kumari Manager email pkumari@gmail.com Owner name Deepak Kumar
Owner phone 9833655321

PL/SQL procedure successfully completed.

SQL>
```