A

Project Report

On

GYM MANAGEMENT SYSTEM

Developed By:

PRATIK M RANA (IT-092)

PARTH N PATEL (IT-080)

Guided By:

PROF. SUNIL K. VITHLANI
Department of Information Technology
Faculty of Technology
DDU University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad – 387001
October – 2018

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled GYM MANAGEMENT SYSTEM is a bonsaied report of the work carried out by

1) Mr. PRATIK M RANA, Student ID No: 16ITUBS001

2) Mr. PARTH N PATEL, Student ID No: 16ITUOS092

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2018-2019.

Prof. Sunil K. Vithlani (Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show my greatest appreciation to Prof. Sunil K. Vithlani. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time we attend his lectures. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Yours Sincerely

PRATIK M. RANA

PARTH N PATEL

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

1.1 CURRENT SYSTEM

The “Gym Management System” is to provide a system which handles the information of the people coming into the gym and maintaining their health care. It takes care of all their health information such as height, weight, BMI etc...

It also maintains the payment record of a particular member coming to our gym. Moreover, it provides facility for various payment modes. Different types of users can login to our website and can get the information about our gym. We also have a trainer associated with a particular member. We also provide various gym facility to our members.

As mentioned earlier, we measure BMI of a particular member so that the member can get the details related to his/her body.

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

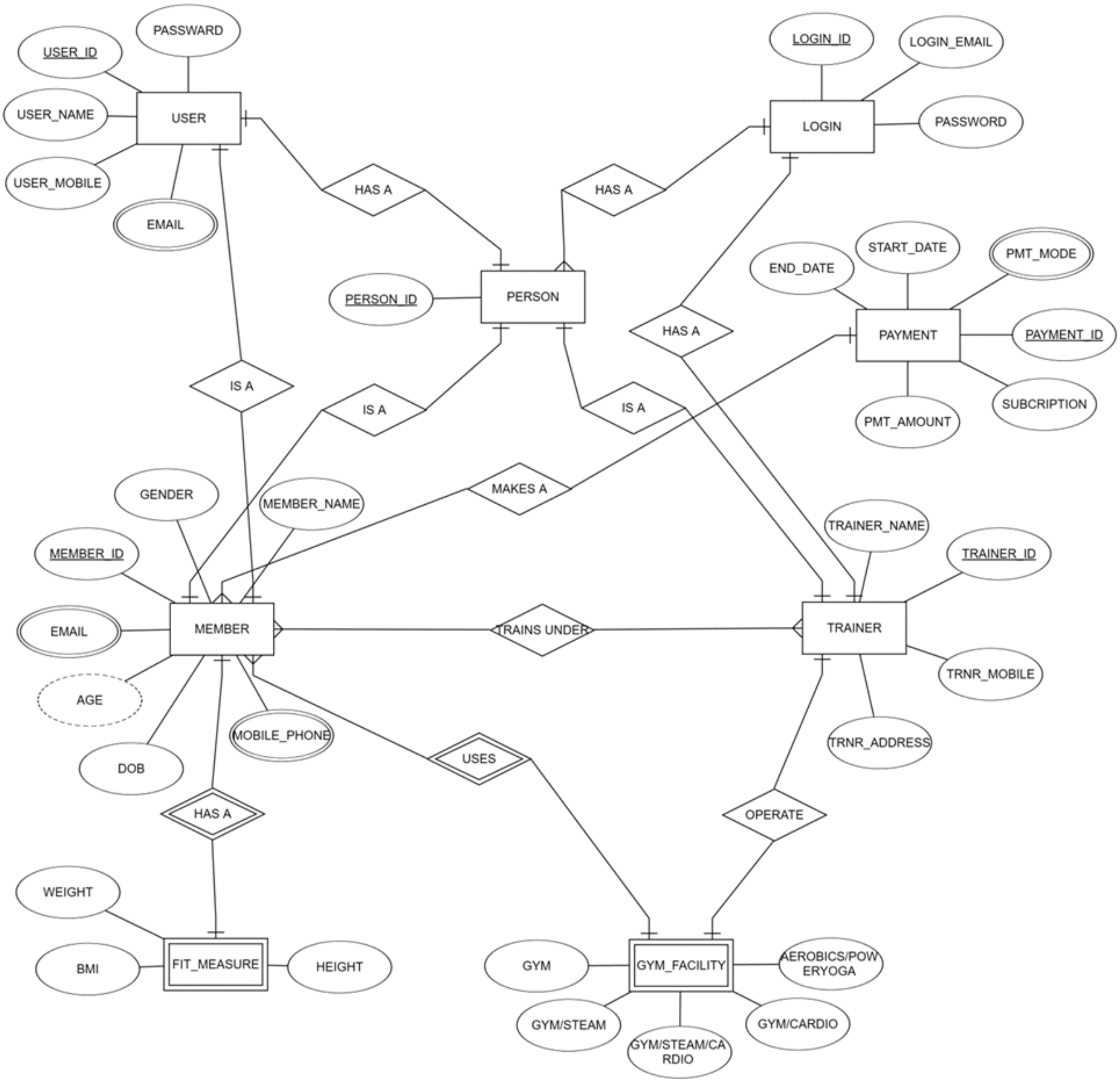
- Taking care of people's health information.
- To provide easy maintaining and handling of gym facilities
- Faster and secured access to system database
- Automatic handling of subscription for the gym
- Functional checks for invalid member and trainer ids

1.3 ADVANTAGES OF THE PROPOSED SYSTEM

While the gym management system is easy and affordable to consider as an option which saves time too, some of the benefits include:

- ✓ Ease of use
- ✓ Reduce the staff requirement
- ✓ Hassle free finance and fee collection processes
- ✓ Saves administration time
- ✓ Reliable
- ✓ Best for communicating with the member and building relationships.

2. E-R DIAGRAM


PASSWARD
USER_ID
USER_NAME
USER
USER_MOBILE
EMAIL
HAS A
HAS A
LOGIN_ID
LOGIN_EMAIL
LOGIN
PASSWORD
PERSON_ID
PERSON
HAS A
START_DATE
END_DATE
PMT_MODE
PAYMENT
PAYMENT_ID
SUBCRIPTION
PMT_AMOUNT
IS A
IS A
IS A
MAKES A
GENDER
MEMBER_NAME
MEMBER_ID
EMAIL
MEMBER
TRAINS UNDER
TRAINER
TRAINER_NAME
TRAINER_ID
TRNR_MOBILE
TRNR_ADDRESS
AGE
DOB
MOBILE_PHONE
USES
HAS A
OPERATE
WEIGHT
BMI
FIT_MEASURE
HEIGHT
GYM
GYM_FACILITY
AEROBICS/POWERYOGA
GYM/STEAM
GYM/STEAM/CARDIO
GYM/CARDIO

2.1 ENTITIES

- PERSON
- USER
- MEMBER
- TRAINER
- PAYMENT
- LOGIN
- FIT_MEASURE
- GYM_FACILITY
- TRAINS_UNDER

2.2 RELATIONSHIPS

- All the trainers and members are person
- Every person having access to website has login details
- Every member is a user
- Each member has a fit measure record
- Each member trains under a certain trainer
- Members have access to various gym facilities
- Trainers have login details
- Members make payment via various payment modes

2.3 MAPPING CONSTRAINTS

- One member has one-person record entry (one to one)
- One trainer has one-person record entry (one to one)
- One person can have many login accounts (one to many)
- One person can have only a single user account (one to one)
- One member can have only one user account (one to one)
- One member has only one fit measure record (one to one)
- One member has one gym facility record (one to one)
- One trainer has only one login account (one to one)
- One trainer has one gym facility record (one to one)
- One member can pay in more than one mode (one to many)
- One member can train under many trainers and each trainer can have many members train under him (many to many)

3. DATA DICTIONARY

PERSON

Column Name	Data Type	Nullable	Default	Primary Key
PERSON_ID	VARCHAR2(5)	No	-	1
LOGIN_ID	VARCHAR2(5)	No	-	-
MEMBER_ID	VARCHAR2(5)	No	-	-
TRAINER_ID	VARCHAR2(5)	No	-	-

USER

Column Name	Data Type	Nullable	Default	Primary Key
USER_ID	VARCHAR2(5)	No	-	1
PASSWORD	VARCHAR2(8)	No	-	-
USER_NAME	VARCHAR2(20)	No	-	-
USER_MOBILE	NUMBER(10)	No	-	-
PERSON_ID	VARCHAR2(5)	No	-	-
MEMBER_ID	VARCHAR2(5)	No	-	-

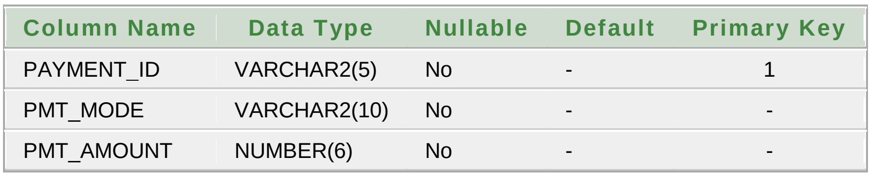
MEMBER

Column Name	Data Type	Nullable	Default	Primary Key
MEMBER_ID	VARCHAR2(5)	No	-	1
GENDER	VARCHAR2(1)	No	-	-
MEMBER_NAME	VARCHAR2(20)	No	-	-
START_DATE	DATE	No	-	-
END_DATE	DATE	No	-	-
DOB	DATE	No	-	-
AGE	NUMBER(2)	No	-	-
EMAIL	VARCHAR2(50)	No	-	-
SUBSCRIPTION	NUMBER(2)	No	-	-
PAYMENT_ID	VARCHAR2(5)	No	-	-

TRAINER

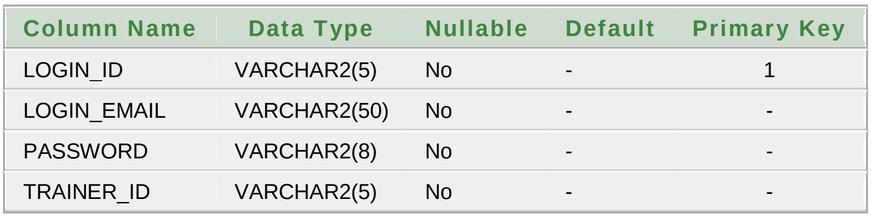
Column Name	Data Type	Nullable	Default	Primary Key
TRAINER_ID	VARCHAR2(5)	No	-	1
TRAINER_NAME	VARCHAR2(20)	Yes	-	-
TRNR_MOBILE	NUMBER(10)	Yes	-	-
TRNR_ADDRESS	VARCHAR2(100)	No	-	-

PAYMENT



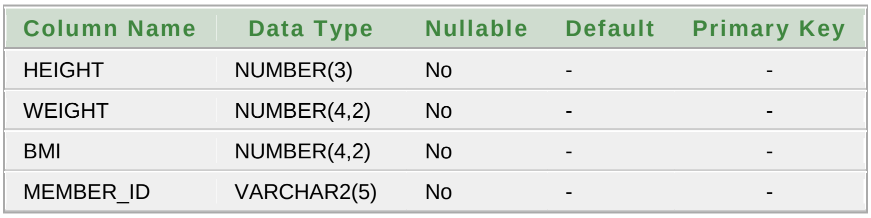
Column Name	Data Type	Nullable	Default	Primary Key
PAYMENT_ID	VARCHAR2(5)	No	-	1
PMT_MODE	VARCHAR2(10)	No	-	-
PMT_AMOUNT	NUMBER(6)	No	-	-

LOGIN



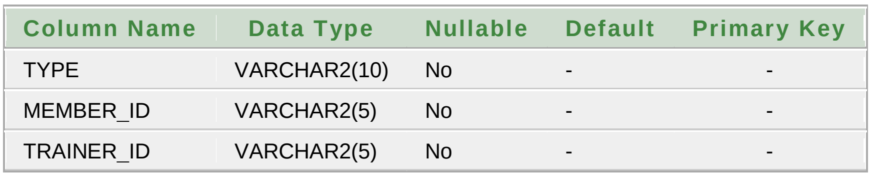
Column Name	Data Type	Nullable	Default	Primary Key
LOGIN_ID	VARCHAR2(5)	No	-	1
LOGIN_EMAIL	VARCHAR2(50)	No	-	-
PASSWORD	VARCHAR2(8)	No	-	-
TRAINER_ID	VARCHAR2(5)	No	-	-

FIT_MEASURE_3



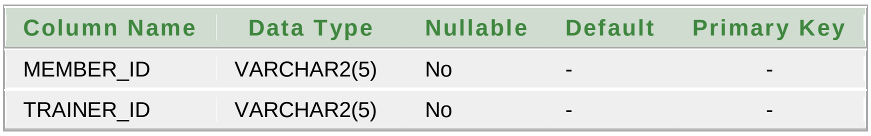
Column Name	Data Type	Nullable	Default	Primary Key
HEIGHT	NUMBER(3)	No	-	-
WEIGHT	NUMBER(4,2)	No	-	-
BMI	NUMBER(4,2)	No	-	-
MEMBER_ID	VARCHAR2(5)	No	-	-

GYM_FACILITY_1



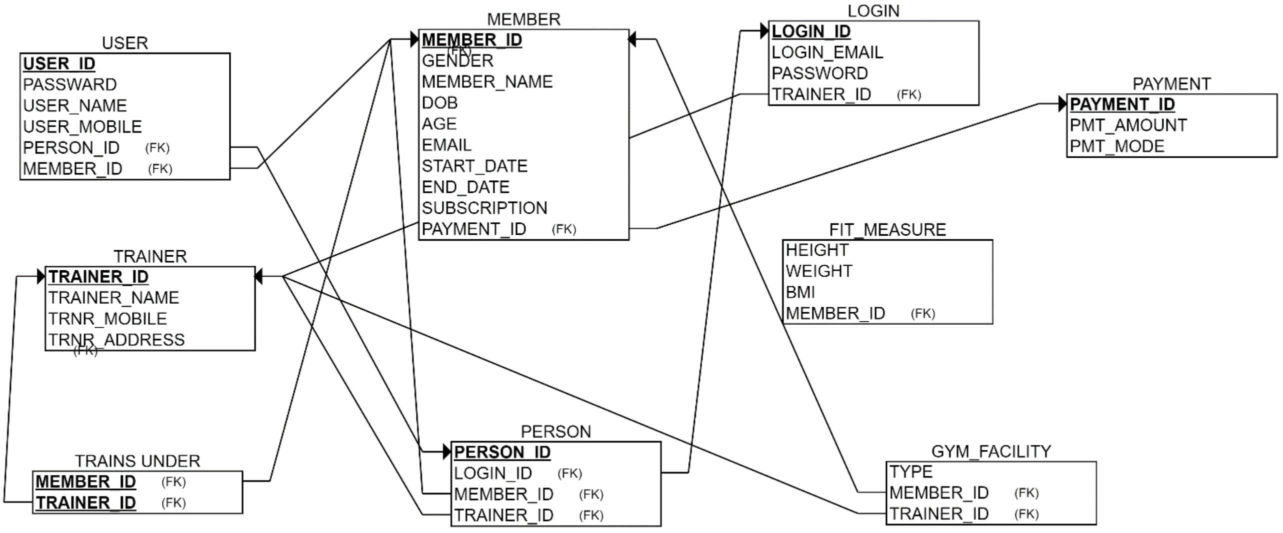
Column Name	Data Type	Nullable	Default	Primary Key
TYPE	VARCHAR2(10)	No	-	-
MEMBER_ID	VARCHAR2(5)	No	-	-
TRAINER_ID	VARCHAR2(5)	No	-	-

TRAINS_UNDER



Column Name	Data Type	Nullable	Default	Primary Key
MEMBER_ID	VARCHAR2(5)	No	-	-
TRAINER_ID	VARCHAR2(5)	No	-	-

4. SCHEMA DIAGRAM

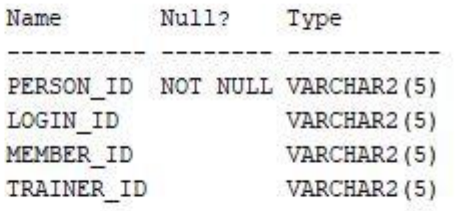


5. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

1. PERSON

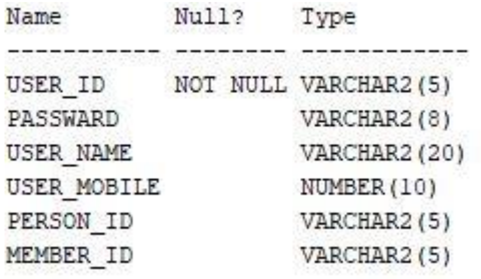
```
CREATE TABLE PERSON
(
        PERSON_ID  VARCHAR2(5) PRIMARY KEY ,
        LOGIN_ID   VARCHAR2(5) REFERENCES LOGIN(LOGIN_ID) ,
        MEMBER_ID  VARCHAR2(5) REFERENCES MEMBER(MEMBER_ID),
        TRAINER_ID VARCHAR2(5) REFERENCES TRAINER(TRAINER_ID)
);
```



```
Name       Null?    Type
---------- -------- -----------
PERSON_ID  NOT NULL VARCHAR2(5)
LOGIN_ID            VARCHAR2(5)
MEMBER_ID           VARCHAR2(5)
TRAINER_ID          VARCHAR2(5)
```

2. USER_1

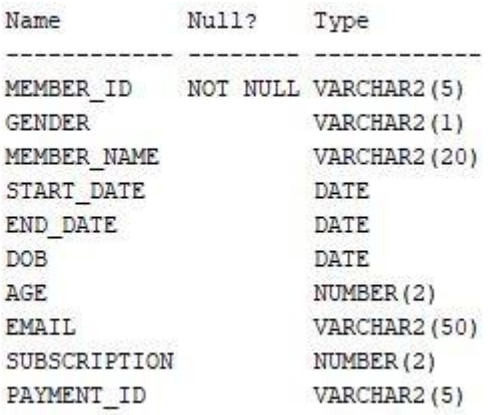
```
CREATE TABLE USER_1
(
        USER_ID     VARCHAR2(5) PRIMARY KEY ,
        PASSWARD    VARCHAR2(8) ,
        USER_NAME   VARCHAR2(20) ,
        USER_MOBILE NUMBER(10) ,
        PERSON_ID   VARCHAR2(5) REFERENCES PERSON(PERSON_ID) ,
        MEMBER_ID   VARCHAR2(5) REFERENCES MEMBER(MEMBER_ID)
);
```



```
Name        Null?    Type
----------- -------- ------------
USER_ID     NOT NULL VARCHAR2(5)
PASSWARD             VARCHAR2(8)
USER_NAME            VARCHAR2(20)
USER_MOBILE          NUMBER(10)
PERSON_ID            VARCHAR2(5)
MEMBER_ID            VARCHAR2(5)
```

3. MEMBER

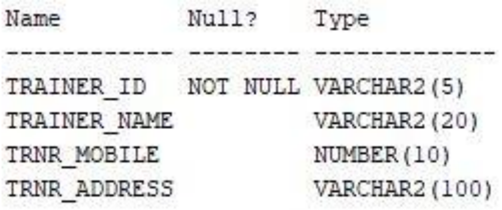
```
CREATE TABLE MEMBER
(
        MEMBER_ID    VARCHAR2(5) PRIMARY KEY ,
        GENDER       VARCHAR2(1) ,
        MEMBER_NAME  VARCHAR2(20) ,
        START_DATE   DATE ,
        END_DATE     DATE ,
        DOB         DATE ,
        AGE         NUMBER(2) ,
        EMAIL         VARCHAR2(50) ,
        SUBSCRIPTION NUMBER(2) ,
        PAYMENT_ID   VARCHAR2(5) REFERENCES
PAYMENT(PAYMENT_ID)
);
```



```
Name         Null?    Type
------------ -------- ------------
MEMBER_ID    NOT NULL VARCHAR2(5)
GENDER                VARCHAR2(1)
MEMBER_NAME           VARCHAR2(20)
START_DATE            DATE
END_DATE              DATE
DOB                   DATE
AGE                   NUMBER(2)
EMAIL                 VARCHAR2(50)
SUBSCRIPTION          NUMBER(2)
PAYMENT_ID            VARCHAR2(5)
```

4. TRAINER

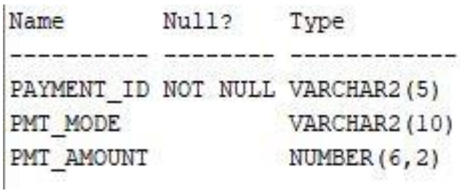
```
CREATE TABLE TRAINER
(
        TRAINER_ID   VARCHAR2(5) PRIMARY KEY ,
        TRAINER_NAME VARCHAR2(20) ,
        TRNR_MOBILE  NUMBER(10) ,
        TRNR_ADDRESS VARCHAR2(100)
);
```



```
Name         Null?    Type
------------ -------- -------------
TRAINER_ID   NOT NULL VARCHAR2(5)
TRAINER_NAME          VARCHAR2(20)
TRNR_MOBILE           NUMBER(10)
TRNR_ADDRESS          VARCHAR2(100)
```

5. PAYMENT

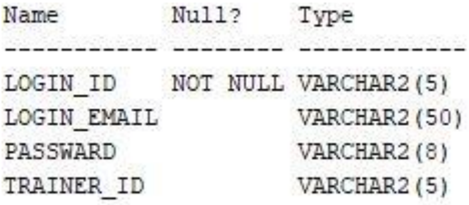
```
CREATE TABLE PAYMENT
(
        PAYMENT_ID   VARCHAR2(5) PRIMARY KEY ,
        PMT_MODE     VARCHAR2(10) ,
        PMT_AMOUNT   NUMBER(6)
);
```



```
Name       Null?    Type
---------- -------- ------------
PAYMENT_ID NOT NULL VARCHAR2(5)
PMT_MODE            VARCHAR2(10)
PMT_AMOUNT          NUMBER(6,2)
```

6. LOGIN

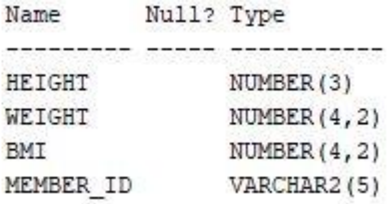
```
CREATE TABLE LOGIN
(
        LOGIN_ID    VARCHAR2(5) PRIMARY KEY ,
        LOGIN_EMAIL VARCHAR2(50) ,
        PASSWARD    VARCHAR2(8) ,
        TRAINER_ID  VARCHAR2(5) REFERENCES TRAINER(TRAINER_ID)
);
```



```
Name        Null?    Type
----------- -------- ------------
LOGIN_ID    NOT NULL VARCHAR2(5)
LOGIN_EMAIL          VARCHAR2(50)
PASSWARD             VARCHAR2(8)
TRAINER_ID           VARCHAR2(5)
```

7. FIT_MEASURE_3

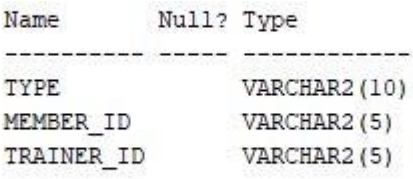
```
CREATE TABLE FIT_MEASURE_3
(
        HEIGHT    NUMBER(3) ,
        WEIGHT    NUMBER(4,2) ,
        BMI       NUMBER(4,2) ,
        MEMBER_ID VARCHAR2(5) REFERENCES MEMBER(MEMBER_ID)
);
```



```
Name      Null? Type
--------- ----- -----------
HEIGHT          NUMBER(3)
WEIGHT          NUMBER(4,2)
BMI             NUMBER(4,2)
MEMBER_ID       VARCHAR2(5)
```

8. GYM_FACILITY_1

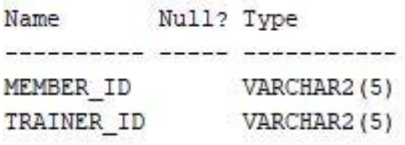
```
CREATE TABLE GYM_FACILITY_1
(
        TYPE       VARCHAR2(10),
        MEMBER_ID  VARCHAR2(5) REFERENCES MEMBER(MEMBER_ID) ,
        TRAINER_ID VARCHAR2(5) REFERENCES TRAINER(TRAINER_ID)
);
```



```
Name       Null? Type
---------- ----- ------------
TYPE             VARCHAR2(10)
MEMBER_ID        VARCHAR2(5)
TRAINER_ID       VARCHAR2(5)
```

9. TRAINS_UNDER

```
CREATE TABLE TRAINS_UNDER
(
        MEMBER_ID  VARCHAR2(5) REFERENCES MEMBER(MEMBER_ID) ,
        TRAINER_ID VARCHAR2(5) REFERENCES TRAINER(TRAINER_ID)
);
```

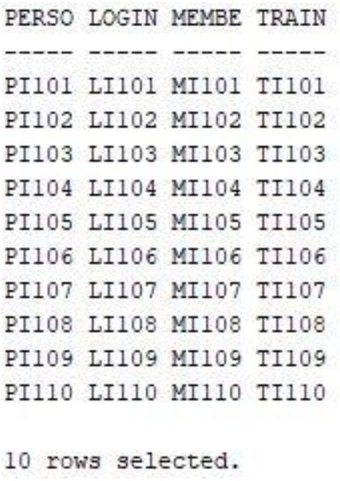


```
Name       Null? Type
---------- ----- -----------
MEMBER_ID        VARCHAR2(5)
TRAINER_ID       VARCHAR2(5)
```

5.2 INSERT DATA VALUES

1. PERSON

INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI101','LI101','MI101','TI101');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI102','LI102','MI102','TI102');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI103','LI103','MI103','TI103');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI104','LI104','MI104','TI104');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI105','LI105','MI105','TI105');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI106','LI106','MI106','TI106');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI107','LI107','MI107','TI107');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI108','LI108','MI108','TI108');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI109','LI109','MI109','TI109');
INSERT INTO PERSON(PERSON_ID,LOGIN_ID,MEMBER_ID,TRAINER_ID)
VALUES('PI110','LI110','MI110','TI110');

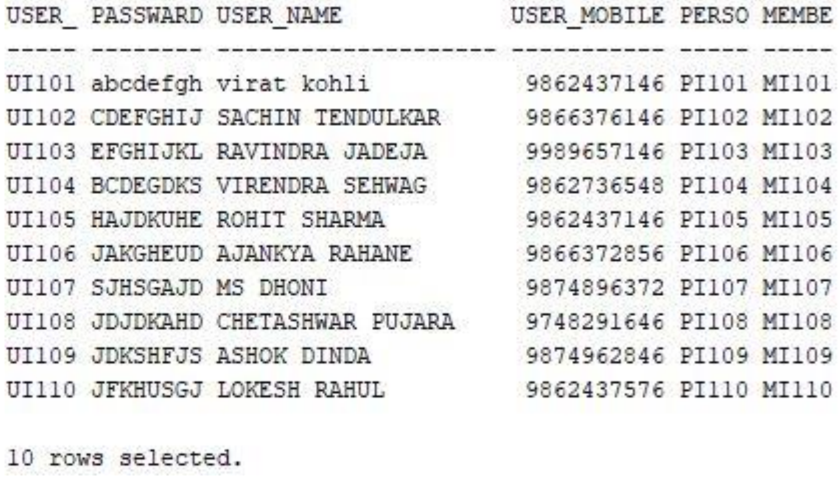


```
PERSO LOGIN MEMBE TRAIN
----- ----- ----- -----
PI101 LI101 MI101 TI101
PI102 LI102 MI102 TI102
PI103 LI103 MI103 TI103
PI104 LI104 MI104 TI104
PI105 LI105 MI105 TI105
PI106 LI106 MI106 TI106
PI107 LI107 MI107 TI107
PI108 LI108 MI108 TI108
PI109 LI109 MI109 TI109
PI110 LI110 MI110 TI110

10 rows selected.
```

2. USER_1

INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI101','abcdefgh','virat kohli',9862437146,'PI101','MI101');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI102','CDEFGHIJ','SACHIN
TENDULKAR',9866376146,'PI102','MI102');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI103','EFGHIJKL','RAVINDRA JADEJA',9989657146,'PI103','MI103');

INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI104','BCDEGDKS','VIRENDRA
SEHWAG',9862736548,'PI104','MI104');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI105','HAJDKUHE','ROHIT SHARMA',9862437146,'PI105','MI105');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI106','JAKGHEUD','AJANKYA
RAHANE',9866372856,'PI106','MI106');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI107','SJHSGAJD','MS DHONI',9874896372,'PI107','MI107');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI108','JDJDKAHD','CHETASHWAR
PUJARA',9748291646,'PI108','MI108');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI109','JDKSHFJS','ASHOK DINDA',9874962846,'PI109','MI109');
INSERT INTO USER_1 (USER_ID, PASSWARD, USER_NAME,
USER_MOBILE, PERSON_ID, MEMBER_ID)
VALUES('UI110','JFKHUSGJ','LOKESH RAHUL',9862437576,'PI110','MI110');

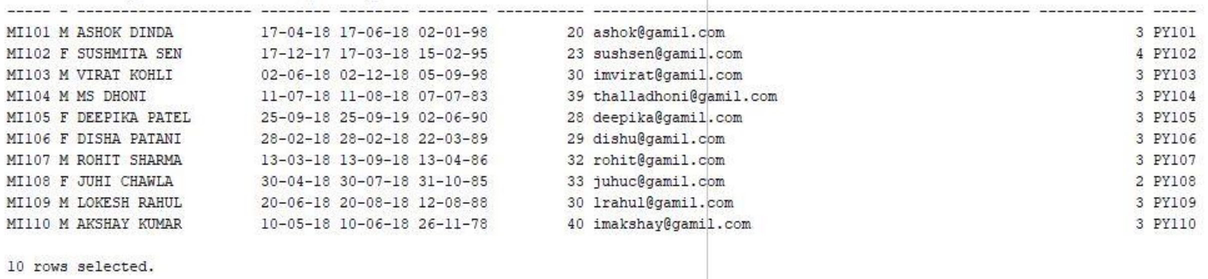


```
USER_ PASSWARD USER_NAME            USER_MOBILE PERSO MEMBE
----- -------- -------------------- ----------- ----- -----
UI101 abcdefgh virat kohli           9862437146 PI101 MI101
UI102 CDEFGHIJ SACHIN TENDULKAR      9866376146 PI102 MI102
UI103 EFGHIJKL RAVINDRA JADEJA       9989657146 PI103 MI103
UI104 BCDEGDKS VIRENDRA SEHWAG       9862736548 PI104 MI104
UI105 HAJDKUHE ROHIT SHARMA          9862437146 PI105 MI105
UI106 JAKGHEUD AJANKYA RAHANE        9866372856 PI106 MI106
UI107 SJHSGAJD MS DHONI              9874896372 PI107 MI107
UI108 JDJDKAHD CHETASHWAR PUJARA     9748291646 PI108 MI108
UI109 JDKSHFJS ASHOK DINDA           9874962846 PI109 MI109
UI110 JFKHUSGJ LOKESH RAHUL          9862437576 PI110 MI110

10 rows selected.
```

3. MEMBER

INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME,
START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION,
PAYMENT_ID)
VALUES('MI101','M','ASHOK DINDA','17-04-2018','17-06-2018','02-01-1998',20,'ashok@gamil.com',02,'PY101');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME,
START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION,
PAYMENT_ID)
VALUES('MI102','F','SUSHMITA SEN','17-12-2017','17-03-2018','15-02-1995',23,'sushsen@gamil.com',04,'PY102');

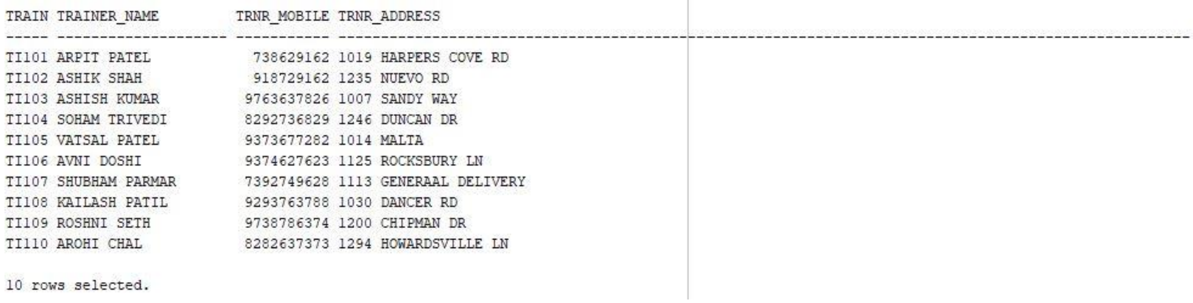
```
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI103','M','VIRAT KOHLI','02-06-2018','02-12-2018','05-09-1998',30,'imvirat@gamil.com',06,'PY103');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI104','M','MS DHONI','11-07-2018','11-08-2018','07-07-1983',39,'thalladhoni@gamil.com',01,'PY104');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI105','F','DEEPIKA PATEL','25-09-2018','25-09-2019','02-06-1990',28,'deepika@gamil.com','12','PY105');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI106','F','DISHA PATANI','28-02-2018','28-02-2018','22-03-1989',29,'dishu@gamil.com',02,'PY106');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI107','M','ROHIT SHARMA','13-03-2018','13-09-2018','13-04-1986',32,'rohit@gamil.com',06,'PY107');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI108','F','JUHI CHAWLA','30-04-2018','30-07-2018','31-10-1985',33,'juhuc@gamil.com',04,'PY108');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI109','M','LOKESH RAHUL','20-06-2018','20-08-2018','12-08-1988',30,'lrahul@gamil.com',02,'PY109');
INSERT INTO MEMBER (MEMBER_ID, GENDER, MEMBER_NAME, START_DATE, END_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, PAYMENT_ID)
VALUES('MI110','M','AKSHAY KUMAR','10-05-2018','10-06-2018','26-11-1978',40,'imakshay@gamil.com',01,'PY110');
```



```
MEMBE G MEMBER_NAME          START_DA END_DATE DOB             AGE EMAIL                                              SUBSCRIPTION PAYME
----- - -------------------- -------- -------- -------- ---------- -------------------------------------------------- ------------ -----
MI101 M ASHOK DINDA          17-04-18 17-06-18 02-01-98         20 ashok@gamil.com                                               3 PY101
MI102 F SUSHMITA SEN         17-12-17 17-03-18 15-02-95         23 sushsen@gamil.com                                             4 PY102
MI103 M VIRAT KOHLI          02-06-18 02-12-18 05-09-98         30 imvirat@gamil.com                                             3 PY103
MI104 M MS DHONI             11-07-18 11-08-18 07-07-83         39 thalladhoni@gamil.com                                         3 PY104
MI105 F DEEPIKA PATEL        25-09-18 25-09-19 02-06-90         28 deepika@gamil.com                                             3 PY105
MI106 F DISHA PATANI         28-02-18 28-02-18 22-03-89         29 dishu@gamil.com                                               3 PY106
MI107 M ROHIT SHARMA         13-03-18 13-09-18 13-04-86         32 rohit@gamil.com                                               3 PY107
MI108 F JUHI CHAWLA          30-04-18 30-07-18 31-10-85         33 juhuc@gamil.com                                               2 PY108
MI109 M LOKESH RAHUL         20-06-18 20-08-18 12-08-88         30 lrahul@gamil.com                                              3 PY109
MI110 M AKSHAY KUMAR         10-05-18 10-06-18 26-11-78         40 imakshay@gamil.com                                            3 PY110

10 rows selected.
```

4. TRAINER

```
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI101','ARPIT PATEL',738629162,'1019 HARPERS COVE RD');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI102','ASHIK SHAH',918729162,'1235 NUEVO RD');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI103','ASHISH KUMAR',9763637826,'1007 SANDY WAY');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI104','SOHAM TRIVEDI',8292736829,'1246 DUNCAN DR');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI105','VATSAL PATEL',9373677282,'1014 MALTA');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI106','AVNI DOSHI',9374627623,'1125 ROCKSBURY LN');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI107','SHUBHAM PARMAR',7392749628,'1113 GENERAAL
DELIVERY');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI108','KAILASH PATIL',9293763788,'1030 DANCER RD');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI109','ROSHNI SETH',9738786374,'1200 CHIPMAN DR');
INSERT INTO TRAINER (TRAINER_ID, TRAINER_NAME, TRNR_MOBILE,
TRNR_ADDRESS)
VALUES('TI110','AROHI CHAL',8282637373,'1294 HOWARDSVILLE LN');
```

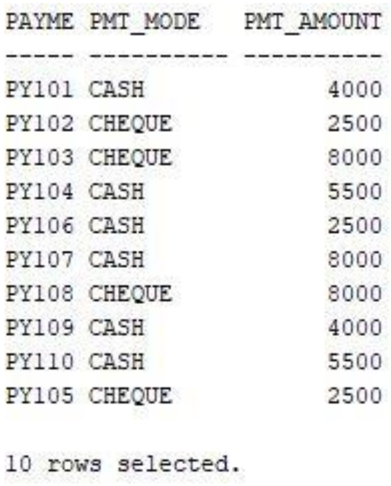


```
TRAIN TRAINER_NAME         TRNR_MOBILE TRNR_ADDRESS
----- -------------------- ----------- ----------------------------------------------------------------------------------------------------

TI101 ARPIT PATEL            738629162 1019 HARPERS COVE RD
TI102 ASHIK SHAH             918729162 1235 NUEVO RD
TI103 ASHISH KUMAR          9763637826 1007 SANDY WAY
TI104 SOHAM TRIVEDI         8292736829 1246 DUNCAN DR
TI105 VATSAL PATEL          9373677282 1014 MALTA
TI106 AVNI DOSHI            9374627623 1125 ROCKSBURY LN
TI107 SHUBHAM PARMAR        7392749628 1113 GENERAAL DELIVERY
TI108 KAILASH PATIL         9293763788 1030 DANCER RD
TI109 ROSHNI SETH           9738786374 1200 CHIPMAN DR
TI110 AROHI CHAL            8282637373 1294 HOWARDSVILLE LN

10 rows selected.
```

5. PAYMENT

```
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY101','CASH',3500);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY102','CHEQUE',2000);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY103','CHEQUE',7500);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
```

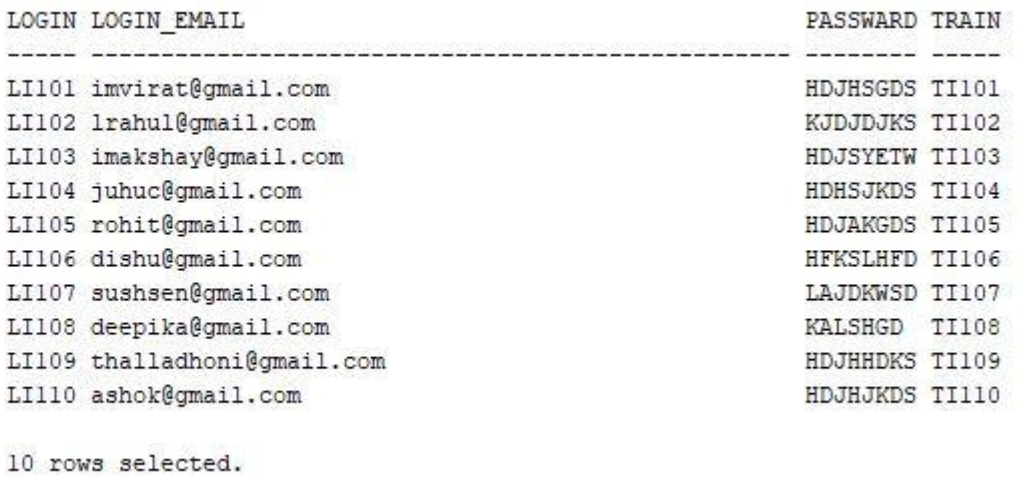
VALUES('PY104','CASH',5000);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY105','CHEQUE',2000);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY106','CASH',2000);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY107','CASH',7500);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY108','CHEQUE',7500);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY109','CASH',3500);
INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
VALUES('PY110','CASH',5000);



```
PAYME PMT_MODE   PMT_AMOUNT
----- ---------- ----------
PY101 CASH             4000
PY102 CHEQUE           2500
PY103 CHEQUE           8000
PY104 CASH             5500
PY106 CASH             2500
PY107 CASH             8000
PY108 CHEQUE           8000
PY109 CASH             4000
PY110 CASH             5500
PY105 CHEQUE           2500

10 rows selected.
```

6. LOGIN

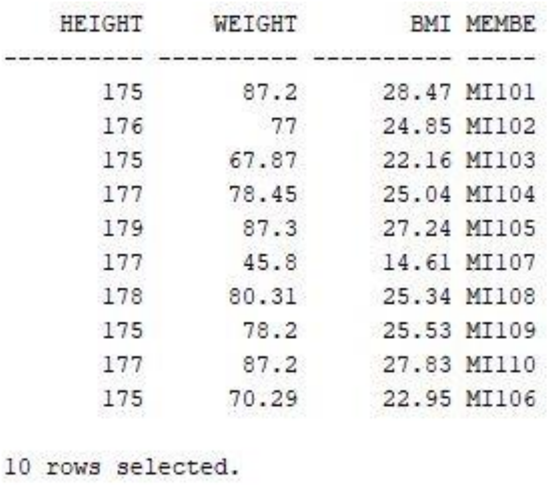
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI101','imvirat@gmail.com','HDJHSGDS','TI101');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI102','lrahul@gmail.com','KJDJDJKS','TI102');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI103','imakshay@gmail.com','HDJSYETW','TI103');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI104','juhuc@gmail.com','HDHSJKDS','TI104');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI105','rohit@gmail.com','HDJAKGDS','TI105');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI106','dishu@gmail.com','HFKSLHFD','TI106');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI107','sushsen@gmail.com','LAJDKWSD','TI107');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI108','deepika@gmail.com','KALSHGD','TI108');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI109','thalladhoni@gmail.com','HDJHHDKS','TI109');
INSERT INTO LOGIN(LOGIN_ID,LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('LI110','ashok@gmail.com','HDJHJKDS','TI110');



```
LOGIN LOGIN_EMAIL                                        PASSWARD TRAIN
----- -------------------------------------------------- -------- -----
LI101 imvirat@gmail.com                                  HDJHSGDS TI101
LI102 lrahul@gmail.com                                   KJDJDJKS TI102
LI103 imakshay@gmail.com                                 HDJSYETW TI103
LI104 juhuc@gmail.com                                    HDHSJKDS TI104
LI105 rohit@gmail.com                                    HDJAKGDS TI105
LI106 dishu@gmail.com                                    HFKSLHFD TI106
LI107 sushsen@gmail.com                                  LAJDKWSD TI107
LI108 deepika@gmail.com                                  KALSHGD  TI108
LI109 thalladhoni@gmail.com                              HDJHHDKS TI109
LI110 ashok@gmail.com                                    HDJHJKDS TI110

10 rows selected.
```

7. FIT_MEASURE_3

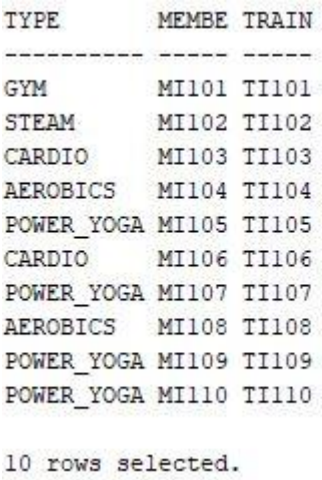
```
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(175,87.20,28.47,'MI101');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(176,77.00,24.85,'MI102');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(175,67.87,22.16,'MI103');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(177,78.45,25.04,'MI104');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(179,87.30,27.24,'MI105');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(175,70.29,22.95,'MI106');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(177,45.80,14.61,'MI107');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(178,80.31,25.34,'MI108');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(175,78.20,25.53,'MI109');
INSERT INTO FIT_MEASURE_3(HEIGHT,WEIGHT,BMI,MEMBER_ID)
VALUES(177,87.20,27.83,'MI110');
```



```
    HEIGHT     WEIGHT        BMI MEMBE
---------- ---------- ---------- -----
       175       87.2      28.47 MI101
       176         77      24.85 MI102
       175      67.87      22.16 MI103
       177      78.45      25.04 MI104
       179       87.3      27.24 MI105
       177       45.8      14.61 MI107
       178      80.31      25.34 MI108
       175       78.2      25.53 MI109
       177       87.2      27.83 MI110
       175      70.29      22.95 MI106

10 rows selected.
```

8. GYM_FACILITY_1

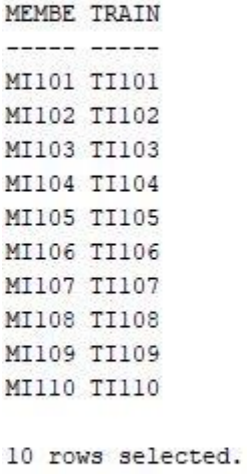
```
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('GYM','MI101','TI101');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('STEAM','MI102','TI102');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('CARDIO','MI103','TI103');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('AEROBICS','MI104','TI104');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('POWER_YOGA','MI105','TI105');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('CARDIO','MI106','TI106');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('POWER_YOGA','MI107','TI107');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('AEROBICS','MI108','TI108');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('GYM','MI109','TI109');
INSERT INTO GYM_FACILITY_1(TYPE,MEMBER_ID,TRAINER_ID)
VALUES('POWER_YOGA','MI110','TI110');
```



```
TYPE       MEMBE TRAIN
---------- ----- -----
GYM        MI101 TI101
STEAM      MI102 TI102
CARDIO     MI103 TI103
AEROBICS   MI104 TI104
POWER_YOGA MI105 TI105
CARDIO     MI106 TI106
POWER_YOGA MI107 TI107
AEROBICS   MI108 TI108
POWER_YOGA MI109 TI109
POWER_YOGA MI110 TI110

10 rows selected.
```

9. TRAINS_UNDER

```
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI101','TI101');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI102','TI102');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI103','TI103');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI104','TI104');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI105','TI105');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI106','TI106');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI107','TI107');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI108','TI108');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI109','TI109');
INSERT INTO TRAINS_UNDER(MEMBER_ID,TRAINER_ID)
VALUES('MI110','TI110');
```

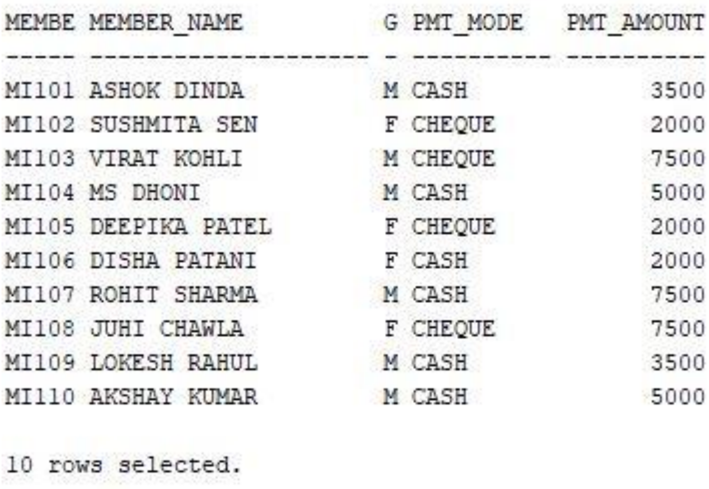


```
MEMBE TRAIN
----- -----
MI101 TI101
MI102 TI102
MI103 TI103
MI104 TI104
MI105 TI105
MI106 TI106
MI107 TI107
MI108 TI108
MI109 TI109
MI110 TI110

10 rows selected.
```

5.3 QUERIES

1. Find the payment details of all the members of the gym.

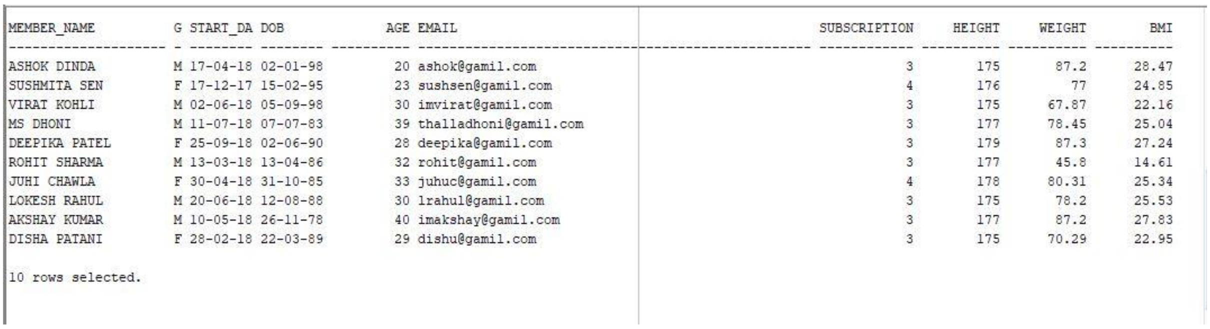
SELECT MEMBER_ID,MEMBER_NAME,GENDER,PMT_MODE,PMT_AMOUNT FROM MEMBER M INNER JOIN PAYMENT P ON M.PAYMENT_ID=P.PAYMENT_ID;



```
MEMBE MEMBER_NAME          G PMT_MODE   PMT_AMOUNT
----- -------------------- - ---------- ----------
MI101 ASHOK DINDA          M CASH             3500
MI102 SUSHMITA SEN         F CHEQUE           2000
MI103 VIRAT KOHLI          M CHEQUE           7500
MI104 MS DHONI             M CASH             5000
MI105 DEEPIKA PATEL        F CHEQUE           2000
MI106 DISHA PATANI         F CASH             2000
MI107 ROHIT SHARMA         M CASH             7500
MI108 JUHI CHAWLA          F CHEQUE           7500
MI109 LOKESH RAHUL         M CASH             3500
MI110 AKSHAY KUMAR         M CASH             5000

10 rows selected.
```

2. Find fitness details of all the members of the gym.

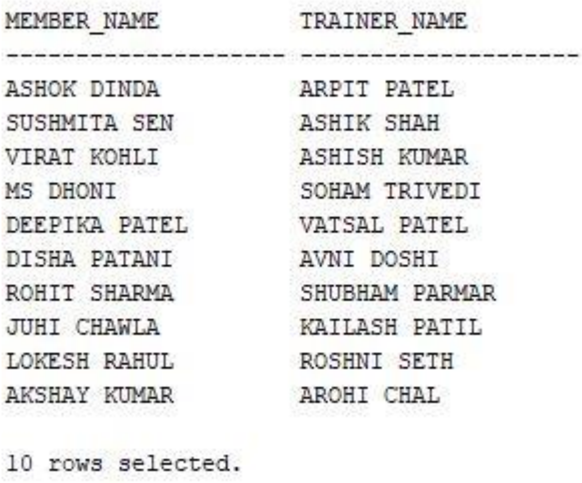
SELECT MEMBER_NAME, GENDER, START_DATE, DOB, AGE, EMAIL, SUBSCRIPTION, HEIGHT, WEIGHT, BMI FROM MEMBER M INNER JOIN FIT_MEASURE_3 FT ON M.MEMBER_ID=FT.MEMBER_ID;



```
MEMBER_NAME          G START_DA DOB             AGE EMAIL                                              SUBSCRIPTION     HEIGHT     WEIGHT        BMI
-------------------- - -------- -------- ---------- -------------------------------------------------- ------------ ---------- ---------- ----------
ASHOK DINDA          M 17-04-18 02-01-98         20 ashok@gamil.com                                               3        175       87.2      28.47
SUSHMITA SEN         F 17-12-17 15-02-95         23 sushsen@gamil.com                                             4        176         77      24.85
VIRAT KOHLI          M 02-06-18 05-09-98         30 imvirat@gamil.com                                             3        175      67.87      22.16
MS DHONI             M 11-07-18 07-07-83         39 thalladhoni@gamil.com                                         3        177      78.45      25.04
DEEPIKA PATEL        F 25-09-18 02-06-90         28 deepika@gamil.com                                             3        179       87.3      27.24
ROHIT SHARMA         M 13-03-18 13-04-86         32 rohit@gamil.com                                               3        177       45.8      14.61
JUHI CHAWLA          F 30-04-18 31-10-85         33 juhuc@gamil.com                                               4        178      80.31      25.34
LOKESH RAHUL         M 20-06-18 12-08-88         30 lrahul@gamil.com                                              3        175       78.2      25.53
AKSHAY KUMAR         M 10-05-18 26-11-78         40 imakshay@gamil.com                                            3        177       87.2      27.83
DISHA PATANI         F 28-02-18 22-03-89         29 dishu@gamil.com                                               3        175      70.29      22.95

10 rows selected.
```

3. Find the details about which member trains under which trainer.

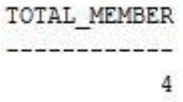
SELECT MEMBER_NAME,TRAINER_NAME FROM MEMBER M1 INNER JOIN TRAINS_UNDER TU ON M1.MEMBER_ID=TU.MEMBER_ID INNER JOIN TRAINER T ON T.TRAINER_ID=TU.TRAINER_ID;



```
MEMBER_NAME          TRAINER_NAME
-------------------- --------------------
ASHOK DINDA          ARPIT PATEL
SUSHMITA SEN         ASHIK SHAH
VIRAT KOHLI          ASHISH KUMAR
MS DHONI             SOHAM TRIVEDI
DEEPIKA PATEL        VATSAL PATEL
DISHA PATANI         AVNI DOSHI
ROHIT SHARMA         SHUBHAM PARMAR
JUHI CHAWLA          KAILASH PATIL
LOKESH RAHUL         ROSHNI SETH
AKSHAY KUMAR         AROHI CHAL

10 rows selected.
```

4. Find the details of the members whose age is less than 30.

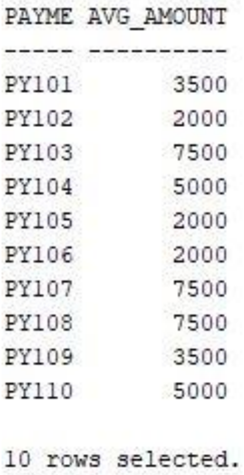
SELECT COUNT(MEMBER_ID) AS TOTAL_MEMBER FROM MEMBER WHERE AGE<30;



```
TOTAL_MEMBER
------------
           4
```

5. Find the average of payment amount of a particular member of the gym.

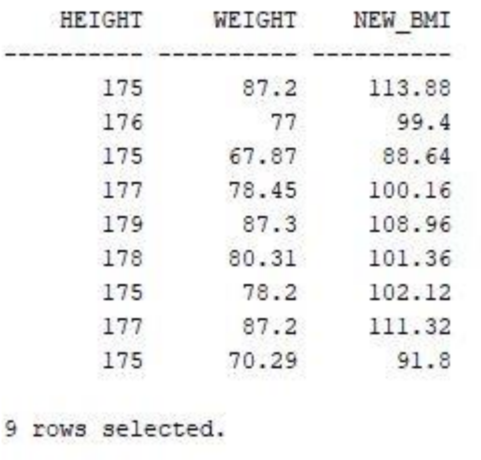
SELECT PAYMENT_ID,AVG(PMT_AMOUNT) AS AVG_AMOUNT FROM PAYMENT GROUP BY PAYMENT_ID;



```
PAYME AVG_AMOUNT
----- ----------
PY101       3500
PY102       2000
PY103       7500
PY104       5000
PY105       2000
PY106       2000
PY107       7500
PY108       7500
PY109       3500
PY110       5000

10 rows selected.
```

6. Find height, weight and BMI details of all the members whose BMI is greater than 20.

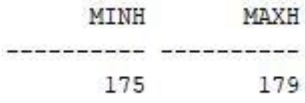
SELECT HEIGHT,WEIGHT,BMI*4 AS NEW_BMI FROM FIT_MEASURE_3 WHERE BMI>20.00;



```
    HEIGHT     WEIGHT    NEW_BMI
---------- ---------- ----------
       175       87.2     113.88
       176         77       99.4
       175      67.87      88.64
       177      78.45     100.16
       179       87.3     108.96
       178      80.31     101.36
       175       78.2     102.12
       177       87.2     111.32
       175      70.29       91.8

9 rows selected.
```

7. Find the members with maximum and minimum heights in the gym.

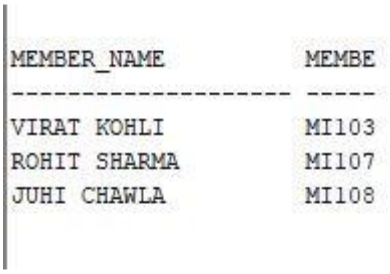
SELECT MIN(FIT_MEASURE_3.HEIGHT) AS MINH,
MAX(FIT_MEASURE_3.HEIGHT) AS MAXH FROM FIT_MEASURE_3;



```
      MINH       MAXH
---------- ----------
       175        179
```

8. Find the member whose payment amount is maximum among all the members.

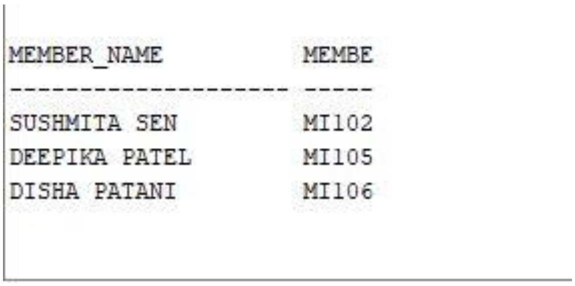
SELECT MEMBER_NAME,MEMBER_ID FROM MEMBER M INNER JOIN
PAYMENT P ON M.PAYMENT_ID=P.PAYMENT_ID WHERE
PMT_AMOUNT=(SELECT MAX(PMT_AMOUNT) FROM PAYMENT);



```
MEMBER_NAME          MEMBE
-------------------- -----
VIRAT KOHLI          MI103
ROHIT SHARMA         MI107
JUHI CHAWLA          MI108
```

9. Find the member whose payment amount is minimum among all the members.

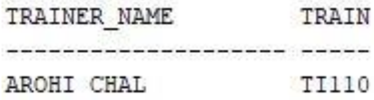
SELECT MEMBER_NAME,MEMBER_ID FROM MEMBER M INNER JOIN
PAYMENT P ON M.PAYMENT_ID=P.PAYMENT_ID WHERE
PMT_AMOUNT=(SELECT MIN(PMT_AMOUNT) FROM PAYMENT);



```
MEMBER_NAME          MEMBE
-------------------- -----
SUSHMITA SEN         MI102
DEEPIKA PATEL        MI105
DISHA PATANI         MI106
```

10. Find the details of the trainer whose trainer id is the highest.

SELECT TRAINER_NAME,TRAINER_ID FROM TRAINER WHERE TRAINER_ID=(SELECT MAX(TRAINER_ID) FROM TRAINER);



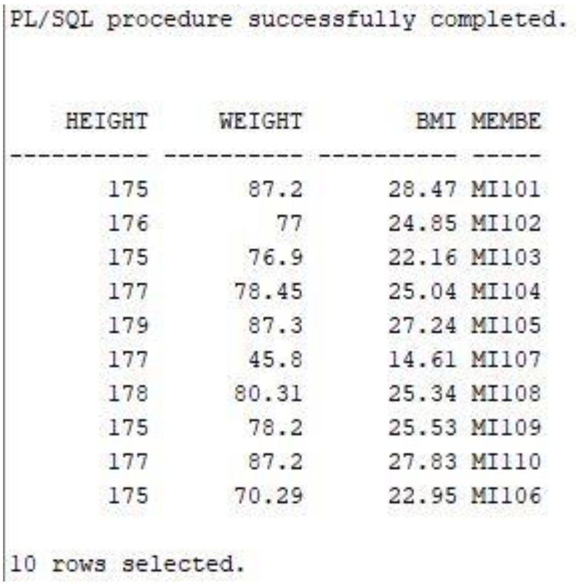
```
TRAINER_NAME         TRAIN
-------------------- -----
AROHI CHAL           TI110
```

5.4 PL/SQL BLOCKS (PROCEDURES AND FUNCTIONS)

PL-SQL BLOCK

1.To update the BMI of a member.

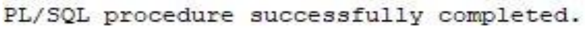
```
DECLARE
            HEIGHT    NUMBER(3);
            WEIGHT    NUMBER(4,2);
            PLBMI     NUMBER(4,2);
            MNO       VARCHAR2(5);
            MORE_THAN_BMI EXCEPTION;
      BEGIN
            HEIGHT:='&HEIGHT';
            WEIGHT:='&WEIGHT';
      MNO:='&MNO';
          PLBMI:=(WEIGHT*10000)/(HEIGHT*HEIGHT);
            SELECT BMI INTO PLBMI FROM FIT_MEASURE_3 WHERE
MEMBER_ID=MNO;
            IF PLBMI<18.50 THEN
                  RAISE MORE_THAN_BMI;
            ELSIF PLBMI>24.90 THEN
                  RAISE MORE_THAN_BMI;
            ELSE
                  UPDATE FIT_MEASURE_3 SET BMI=PLBMI WHERE
MEMBER_ID=MNO;
            END IF;
            EXCEPTION
                  WHEN MORE_THAN_BMI THEN
                        DBMS_OUTPUT.PUT_LINE(MNO||' YOUR BODY
MASS INDEX IS NOT IN THE SPECIFIED RANGE(THE BODY MASS INDEX
SHOULD BE BETWEEN 18.50 TO 24.90)');
            END;
```



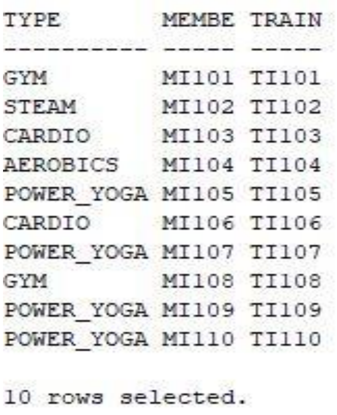
```
PL/SQL procedure successfully completed.


    HEIGHT     WEIGHT        BMI MEMBE
---------- ---------- ---------- -----
       175       87.2      28.47 MI101
       176         77      24.85 MI102
       175       76.9      22.16 MI103
       177      78.45      25.04 MI104
       179       87.3      27.24 MI105
       177       45.8      14.61 MI107
       178      80.31      25.34 MI108
       175       78.2      25.53 MI109
       177       87.2      27.83 MI110
       175      70.29      22.95 MI106

10 rows selected.
```

2. To update the type of gym or produce exception for a particular member.

```
DECLARE
        GYMTYPE VARCHAR2(10);
                MID VARCHAR(5);
                X VARCHAR(10);
                INVALID_TYPE EXCEPTION;
        BEGIN
                MID:='&MID';
                GYMTYPE:='&GYMTYPE';
                X:=GYMTYPE;
                IF X='GYM' THEN
                        UPDATE GYM_FACILITY_1 SET TYPE='GYM' WHERE
MEMBER_ID=MID;
                ELSIF X='STEAM' THEN
                        UPDATE GYM_FACILITY_1 SET TYPE='STEAM' WHERE
MEMBER_ID=MID;
                ELSIF X='CARDIO' THEN
                        UPDATE GYM_FACILITY_1 SET TYPE='CARDIO'
WHERE MEMBER_ID=MID;
                ELSIF X='AROBICS' THEN
                        UPDATE GYM_FACILITY_1 SET TYPE='AEROBICS'
WHERE MEMBER_ID=MID;
                ELSIF X='POWER_YOGA' THEN
                        UPDATE GYM_FACILITY_1 SET TYPE='POWER_YOGA'
WHERE MEMBER_ID=MID;
                ELSE
                        RAISE INVALID_TYPE;
                END IF;
        EXCEPTION
                WHEN INVALID_TYPE THEN
                        DBMS_OUTPUT.PUT_LINE('PLEASE ENTER THE VALID
GYM TYPE THAT YOU WANT TO PERSUE.');
        END;
```



```
PL/SQL procedure successfully completed.
```

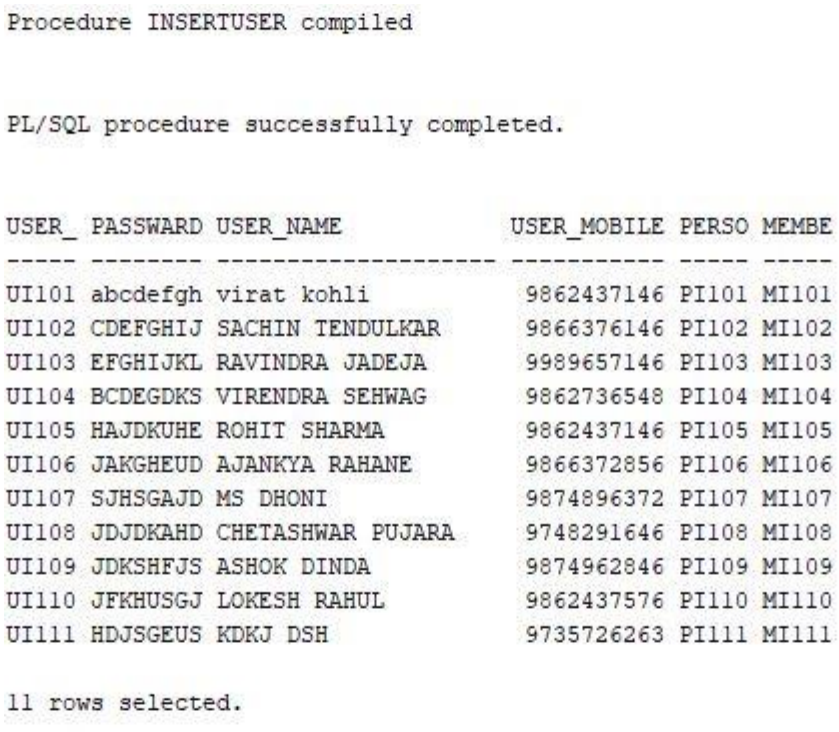


```
TYPE       MEMBE TRAIN
---------- ----- -----
GYM        MI101 TI101
STEAM      MI102 TI102
CARDIO     MI103 TI103
AEROBICS   MI104 TI104
POWER_YOGA MI105 TI105
CARDIO     MI106 TI106
POWER_YOGA MI107 TI107
GYM        MI108 TI108
POWER_YOGA MI109 TI109
POWER_YOGA MI110 TI110

10 rows selected.
```

PROCEDURE

1.Procedure to insert a record into the user table when a person visits the website.

```
CREATE OR REPLACE PROCEDURE INSERTUSER(USID IN
VARCHAR2,PASS IN VARCHAR2,
        UNAME IN VARCHAR2,UMOBILE IN NUMBER,PID IN
VARCHAR2,MID IN VARCHAR2)
        IS
        BEGIN
                INSERT INTO USER_1
                VALUES(USID,PASS,UNAME,UMOBILE,PID,MID);
        END;
RUN:BEGIN
  INSERTUSER('UI111','HDJSGEUS','KDKJ DSH',9735726263,'PI111','MI111');
  dbms_output.put_line('record inserted successfully');
END;
```



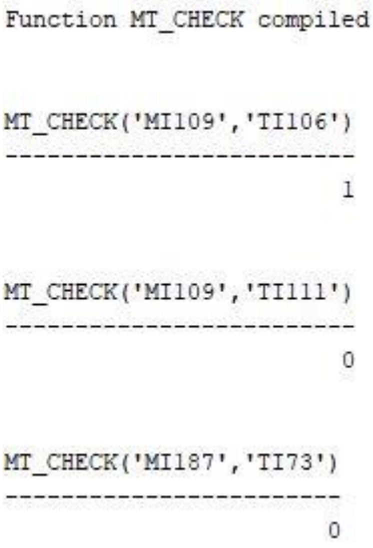
```
Procedure INSERTUSER compiled


PL/SQL procedure successfully completed.


USER_ PASSWARD USER_NAME            USER_MOBILE PERSO MEMBE
----- -------- -------------------- ----------- ----- -----
UI101 abcdefgh virat kohli           9862437146 PI101 MI101
UI102 CDEFGHIJ SACHIN TENDULKAR      9866376146 PI102 MI102
UI103 EFGHIJKL RAVINDRA JADEJA       9989657146 PI103 MI103
UI104 BCDEGDKS VIRENDRA SEHWAG       9862736548 PI104 MI104
UI105 HAJDKUHE ROHIT SHARMA          9862437146 PI105 MI105
UI106 JAKGHEUD AJANKYA RAHANE        9866372856 PI106 MI106
UI107 SJHSGAJD MS DHONI              9874896372 PI107 MI107
UI108 JDJDKAHD CHETASHWAR PUJARA     9748291646 PI108 MI108
UI109 JDKSHFJS ASHOK DINDA           9874962846 PI109 MI109
UI110 JFKHUSGJ LOKESH RAHUL          9862437576 PI110 MI110
UI111 HDJSGEUS KDKJ DSH              9735726263 PI111 MI111

11 rows selected.
```

FUNCTION

1.Function to check the trainer id and member id.

```
CREATE OR REPLACE FUNCTION MT_CHECK(MID IN VARCHAR2,TID IN
VARCHAR2)
        RETURN NUMBER IS
        DMID VARCHAR2(5);
        DTID VARCHAR2(5);
        BEGIN
```

```
      SELECT MEMBER_ID INTO DMID FROM PERSON WHERE
MEMBER_ID=MID;
                  SELECT TRAINER_ID INTO DTID FROM PERSON WHERE
TRAINER_ID=TID;
         RETURN 1;
         EXCEPTION
                  WHEN NO_DATA_FOUND THEN RETURN 0;
         END;
RUN:SELECT MT_CHECK(",") FROM DUAL;
```

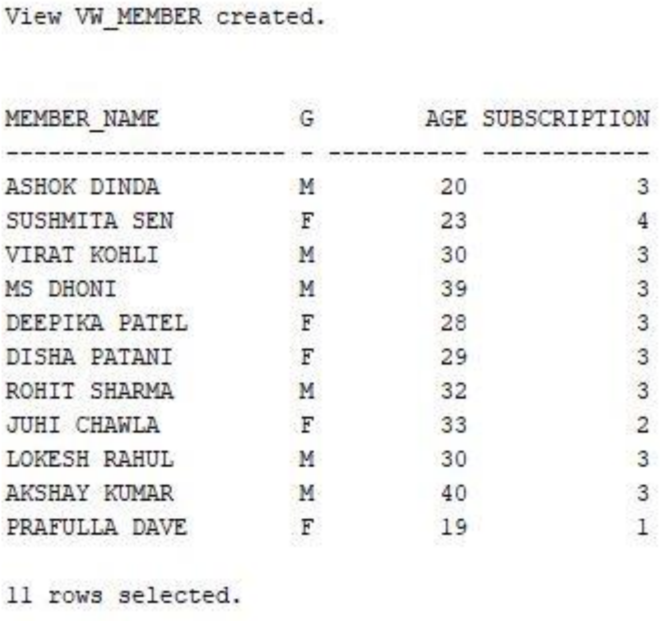


```
Function MT_CHECK compiled


MT_CHECK('MI109','TI106')
-------------------------
                        1


MT_CHECK('MI109','TI111')
-------------------------
                        0


MT_CHECK('MI187','TI73')
------------------------
                       0
```

5.5 VIEWS

1. View for displaying name, gender, age and subscription of a member.

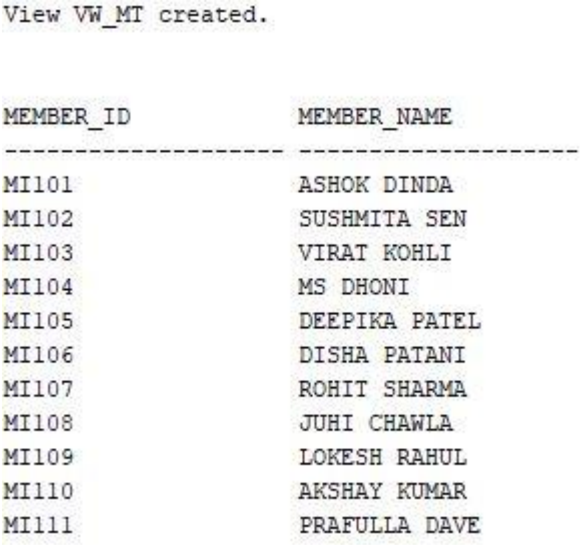
```
CREATE VIEW VW_MEMBER AS
        SELECT MEMBER_NAME,GENDER,AGE,SUBSCRIPTION
        FROM MEMBER;
RUN:SELECT * FROM VW_MEMBER;
```



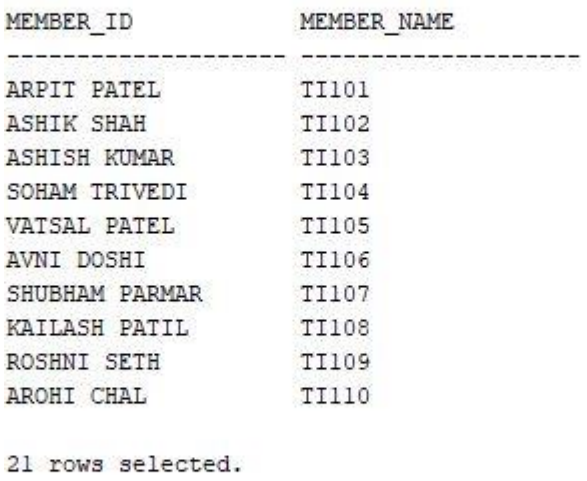
```
View VW_MEMBER created.


MEMBER_NAME          G        AGE SUBSCRIPTION
-------------------- - ---------- ------------
ASHOK DINDA          M         20            3
SUSHMITA SEN         F         23            4
VIRAT KOHLI          M         30            3
MS DHONI             M         39            3
DEEPIKA PATEL        F         28            3
DISHA PATANI         F         29            3
ROHIT SHARMA         M         32            3
JUHI CHAWLA          F         33            2
LOKESH RAHUL         M         30            3
AKSHAY KUMAR         M         40            3
PRAFULLA DAVE        F         19            1

11 rows selected.
```

2. View for getting information related to respective trainer for all the members.

```
CREATE OR REPLACE VIEW VW_MT AS
        SELECT MEMBER_ID,MEMBER_NAME FROM MEMBER
        UNION ALL
        SELECT TRAINER_NAME,TRAINER_ID FROM TRAINER;
RUN:SELECT * FROM VW_MT;
```



```
View VW_MT created.


MEMBER_ID            MEMBER_NAME
-------------------- --------------------
MI101                ASHOK DINDA
MI102                SUSHMITA SEN
MI103                VIRAT KOHLI
MI104                MS DHONI
MI105                DEEPIKA PATEL
MI106                DISHA PATANI
MI107                ROHIT SHARMA
MI108                JUHI CHAWLA
MI109                LOKESH RAHUL
MI110                AKSHAY KUMAR
MI111                PRAFULLA DAVE
```

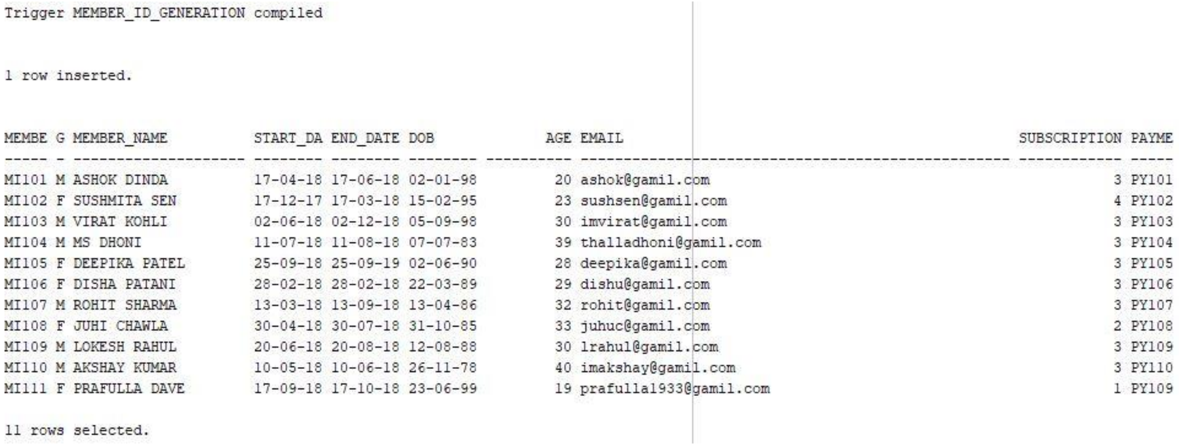


```
MEMBER_ID            MEMBER_NAME
-------------------- --------------------
ARPIT PATEL          TI101
ASHIK SHAH           TI102
ASHISH KUMAR         TI103
SOHAM TRIVEDI        TI104
VATSAL PATEL         TI105
AVNI DOSHI           TI106
SHUBHAM PARMAR       TI107
KAILASH PATIL        TI108
ROSHNI SETH          TI109
AROHI CHAL           TI110

21 rows selected.
```

5.6 TRIGGER

1. Trigger for primary key using lock table method.

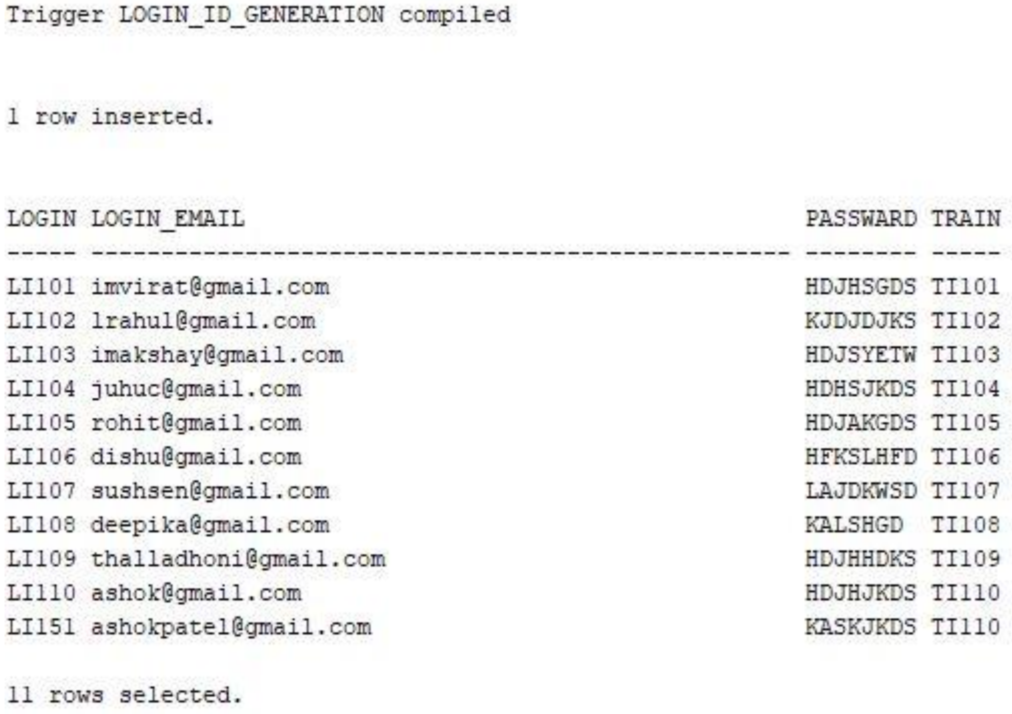
```
CREATE TABLE PKEY_LOOKUP
(
    PKEY_VALUE VARCHAR2(5)
);
    CREATE OR REPLACE TRIGGER MEMBER_ID_GENERATION
    BEFORE INSERT ON MEMBER
    FOR EACH ROW
    DECLARE
        LOOKUP_PKEY_VALUE VARCHAR2(5);
        NEW_PKEY_VALUE VARCHAR2(5);
    BEGIN
        BEGIN
            SELECT PKEY_VALUE INTO LOOKUP_PKEY_VALUE
FROM PKEY_LOOKUP;
            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    LOOKUP_PKEY_VALUE:='MI111';
        END;
        :NEW.MEMBER_ID:=LOOKUP_PKEY_VALUE;

    NEW_PKEY_VALUE:=TO_NUMBER(SUBSTR(LOOKUP_PKEY_VALU
E,4,2))+1;
        LOOKUP_PKEY_VALUE:='MI'||NEW_PKEY_VALUE;
        IF LOOKUP_PKEY_VALUE='MI110' THEN
            INSERT INTO PKEY_LOOKUP
VALUES(LOOKUP_PKEY_VALUE);
        ELSE
            UPDATE PKEY_LOOKUP SET
PKEY_VALUE=LOOKUP_PKEY_VALUE;
        END IF;
    END;
RUN:INSERT INTO
MEMBER(GENDER,MEMBER_NAME,START_DATE,END_DATE,DOB,AGE,E
MAIL,SUBSCRIPTION,PAYMENT_ID)
   VALUES('F','PRAFULLA DAVE','17-09-2018','17-10-2018','23-06-
1999',19,'prafulla1933@gamil.com',01,'PY109');
```



```
Trigger MEMBER_ID_GENERATION compiled


1 row inserted.


MEMBE G MEMBER_NAME          START_DA END_DATE DOB             AGE EMAIL                                              SUBSCRIPTION PAYME
----- - -------------------- -------- -------- -------- ---------- -------------------------------------------------- ------------ -----
MI101 M ASHOK DINDA          17-04-18 17-06-18 02-01-98         20 ashok@gamil.com                                               3 PY101
MI102 F SUSHMITA SEN         17-12-17 17-03-18 15-02-95         23 sushsen@gamil.com                                             4 PY102
MI103 M VIRAT KOHLI          02-06-18 02-12-18 05-09-98         30 imvirat@gamil.com                                             3 PY103
MI104 M MS DHONI             11-07-18 11-08-18 07-07-83         39 thalladhoni@gamil.com                                         3 PY104
MI105 F DEEPIKA PATEL        25-09-18 25-09-19 02-06-90         28 deepika@gamil.com                                             3 PY105
MI106 F DISHA PATANI         28-02-18 28-02-18 22-03-89         29 dishu@gamil.com                                               3 PY106
MI107 M ROHIT SHARMA         13-03-18 13-09-18 13-04-86         32 rohit@gamil.com                                               3 PY107
MI108 F JUHI CHAWLA          30-04-18 30-07-18 31-10-85         33 juhuc@gamil.com                                               2 PY108
MI109 M LOKESH RAHUL         20-06-18 20-08-18 12-08-88         30 lrahul@gamil.com                                              3 PY109
MI110 M AKSHAY KUMAR         10-05-18 10-06-18 26-11-78         40 imakshay@gamil.com                                            3 PY110
MI111 F PRAFULLA DAVE        17-09-18 17-10-18 23-06-99         19 prafulla1933@gamil.com                                        1 PY109

11 rows selected.
```

2. Trigger for creating primary key by sequence.

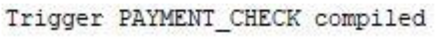
```
CREATE SEQUENCE LOGIN_SEQ_4 INCREMENT BY 001 START WITH 111;
          CREATE OR REPLACE TRIGGER LOGIN_ID_GENERATION
          BEFORE INSERT ON LOGIN
          FOR EACH ROW
          DECLARE
                    PRIMARY_KEY_VALUE VARCHAR2(5);
          BEGIN
                    SELECT 'LI'||TO_CHAR(LOGIN_SEQ_4.NEXTVAL)
                    INTO PRIMARY_KEY_VALUE FROM DUAL;
                    :NEW.LOGIN_ID:=PRIMARY_KEY_VALUE;
          END;
RUN:INSERT INTO LOGIN(LOGIN_EMAIL,PASSWARD,TRAINER_ID)
VALUES('ashokpatel@gmail.com','KASKJKDS','TI110');
```



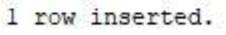
```
Trigger LOGIN_ID_GENERATION compiled


1 row inserted.


LOGIN LOGIN_EMAIL                                        PASSWARD TRAIN
----- -------------------------------------------------- -------- -----
LI101 imvirat@gmail.com                                  HDJHSGDS TI101
LI102 lrahul@gmail.com                                   KJDJDJKS TI102
LI103 imakshay@gmail.com                                 HDJSYETW TI103
LI104 juhuc@gmail.com                                    HDHSJKDS TI104
LI105 rohit@gmail.com                                    HDJAKGDS TI105
LI106 dishu@gmail.com                                    HFKSLHFD TI106
LI107 sushsen@gmail.com                                  LAJDKWSD TI107
LI108 deepika@gmail.com                                  KALSHGD  TI108
LI109 thalladhoni@gmail.com                              HDJHHDKS TI109
LI110 ashok@gmail.com                                    HDJHJKDS TI110
LI151 ashokpatel@gmail.com                               KASKJKDS TI110

11 rows selected.
```

3. Trigger for checking payment details.

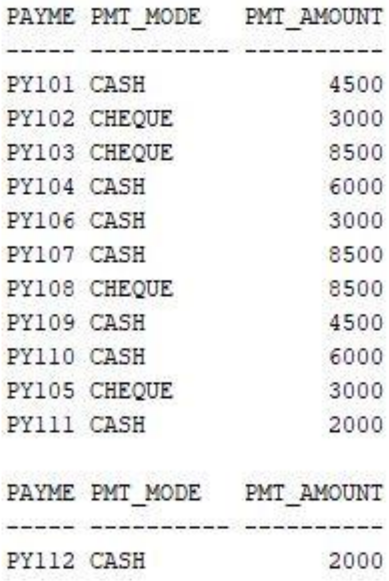
```
CREATE OR REPLACE TRIGGER PAYMENT_CHECK
        BEFORE INSERT ON PAYMENT
        FOR EACH ROW
        DECLARE
                C_PAY_ID VARCHAR2(5);
        BEGIN
                IF :NEW.PMT_AMOUNT<=0 THEN
                        RAISE_APPLICATION_ERROR(-2001,'THE
TRANSACTION AMOUNT IS ZERO OR NAGATIVE');
                END IF;
                IF :NEW.PMT_MODE='CASH' THEN
                        DBMS_OUTPUT.PUT_LINE('VALID');
                ELSIF :NEW.PMT_MODE='CHEQUE' THEN
                        DBMS_OUTPUT.PUT_LINE('VALID');
                ELSE
                        RAISE_APPLICATION_ERROR(-2002,'INVALID
PAYMENT MODE.');
                END IF;
        END;
RUN:INSERT INTO PAYMENT(PAYMENT_ID,PMT_MODE,PMT_AMOUNT)
   VALUES('PY106','CASH',2000);
```



```
Trigger PAYMENT_CHECK compiled
```



```
1 row inserted.
```

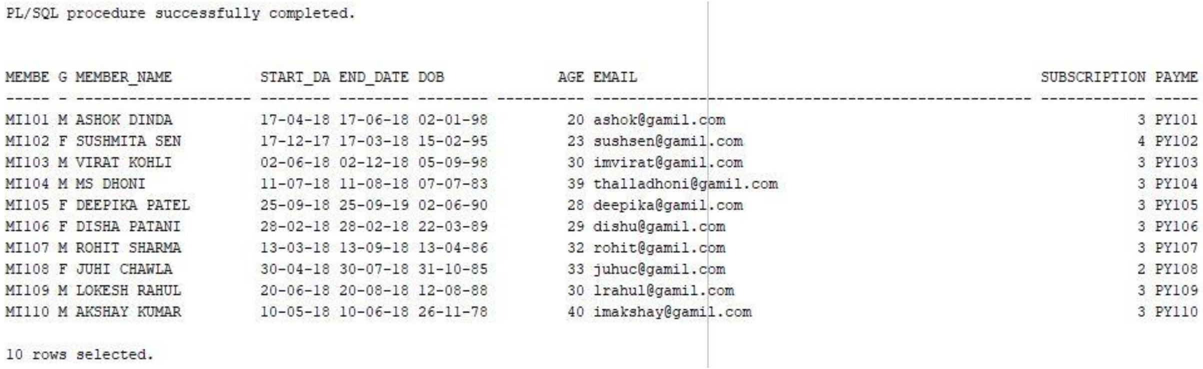


```
PAYME PMT_MODE   PMT_AMOUNT
----- ---------- ----------
PY101 CASH             4500
PY102 CHEQUE           3000
PY103 CHEQUE           8500
PY104 CASH             6000
PY106 CASH             3000
PY107 CASH             8500
PY108 CHEQUE           8500
PY109 CASH             4500
PY110 CASH             6000
PY105 CHEQUE           3000
PY111 CASH             2000

PAYME PMT_MODE   PMT_AMOUNT
----- ---------- ----------
PY112 CASH             2000
```

5.7 CURSORS

1. Implicit cursor to update the member subscription if the member wants to continue.

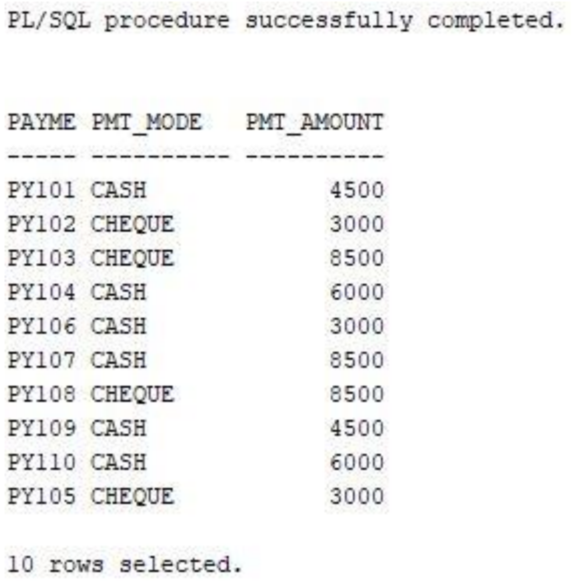
```
DECLARE
		MNO VARCHAR2(5);
	BEGIN
		MNO:='&MNO';
		UPDATE MEMBER SET SUBSCRIPTION='&SUBSCRIPTION'
WHERE MEMBER_ID=MNO;
		IF SQL%FOUND THEN
			DBMS_OUTPUT.PUT_LINE('MEMBER SUBSCRIPTION
EXTENDED SUCCESSFULLY.');
		END IF;
		IF SQL%NOTFOUND THEN
			DBMS_OUTPUT.PUT_LINE('INVALID MEMBER ID.');
		END IF;
	END;
```



```
PL/SQL procedure successfully completed.


MEMBE G MEMBER_NAME          START_DA END_DATE DOB             AGE EMAIL                                              SUBSCRIPTION PAYME
----- - -------------------- -------- -------- -------- ---------- -------------------------------------------------- ------------ -----
MI101 M ASHOK DINDA          17-04-18 17-06-18 02-01-98         20 ashok@gamil.com                                               3 PY101
MI102 F SUSHMITA SEN         17-12-17 17-03-18 15-02-95         23 sushsen@gamil.com                                             4 PY102
MI103 M VIRAT KOHLI          02-06-18 02-12-18 05-09-98         30 imvirat@gamil.com                                             3 PY103
MI104 M MS DHONI             11-07-18 11-08-18 07-07-83         39 thalladhoni@gamil.com                                         3 PY104
MI105 F DEEPIKA PATEL        25-09-18 25-09-19 02-06-90         28 deepika@gamil.com                                             3 PY105
MI106 F DISHA PATANI         28-02-18 28-02-18 22-03-89         29 dishu@gamil.com                                               3 PY106
MI107 M ROHIT SHARMA         13-03-18 13-09-18 13-04-86         32 rohit@gamil.com                                               3 PY107
MI108 F JUHI CHAWLA          30-04-18 30-07-18 31-10-85         33 juhuc@gamil.com                                               2 PY108
MI109 M LOKESH RAHUL         20-06-18 20-08-18 12-08-88         30 lrahul@gamil.com                                              3 PY109
MI110 M AKSHAY KUMAR         10-05-18 10-06-18 26-11-78         40 imakshay@gamil.com                                            3 PY110

10 rows selected.
```

2. Implicit cursor to increase the payment of the member.

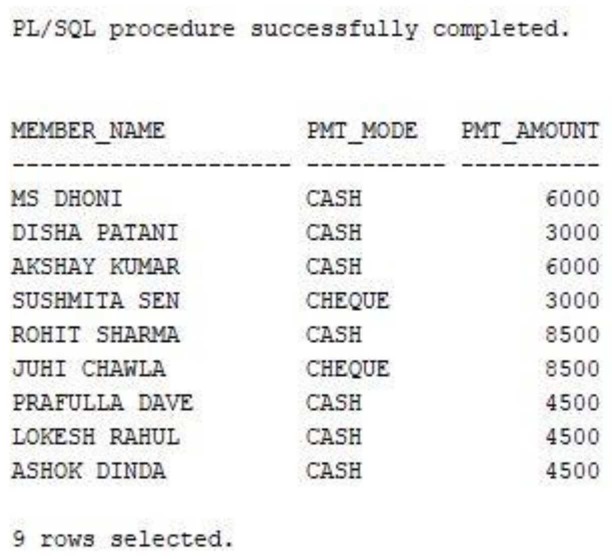
```
DECLARE
		NOMEM NUMBER(2);
	BEGIN
		UPDATE PAYMENT SET PMT_AMOUNT=PMT_AMOUNT + 500;
		IF SQL%FOUND THEN
			NOMEM:=SQL%ROWCOUNT;
			DBMS_OUTPUT.PUT_LINE('MEMBER PAYMENT OF
'||NOMEM||' UPDATED SUCCESSFULLY.');
		END IF;
		IF SQL%NOTFOUND THEN
			DBMS_OUTPUT.PUT_LINE('MEMBER PAYMENT NOT
YET UPDATED.');
		END IF;
	END;
```



```
PL/SQL procedure successfully completed.


PAYME PMT_MODE   PMT_AMOUNT
----- ---------- ----------
PY101 CASH             4500
PY102 CHEQUE           3000
PY103 CHEQUE           8500
PY104 CASH             6000
PY106 CASH             3000
PY107 CASH             8500
PY108 CHEQUE           8500
PY109 CASH             4500
PY110 CASH             6000
PY105 CHEQUE           3000

10 rows selected.
```

3. Explicit cursor to disable the membership of a particular member.

```
CREATE TABLE INACTIVE_PAYMENT
(
        PAYMENT_ID   VARCHAR2(5) PRIMARY KEY ,
        PMT_MODE     VARCHAR2(10) ,
        PMT_AMOUNT   NUMBER(6)
);
DECLARE
        MNO VARCHAR(5);
        CURSOR NO_MEMBER IS
        SELECT PAYMENT_ID,PMT_MODE,PMT_AMOUNT FROM PAYMENT
        WHERE PAYMENT_ID IN
        (SELECT PAYMENT_ID FROM MEMBER GROUP BY PAYMENT_ID
HAVING MAX(SYSDATE-END_DATE)>=1);
BEGIN
        MNO:='&MNO';
        FOR TREC IN NO_MEMBER
        LOOP
                INSERT INTO INACTIVE_PAYMENT

        VALUES(TREC.PAYMENT_ID,TREC.PMT_MODE,TREC.PMT_AMOUN
T);
        END LOOP;
        COMMIT;
END;
```



```
PL/SQL procedure successfully completed.


MEMBER_NAME          PMT_MODE   PMT_AMOUNT
-------------------- ---------- ----------
MS DHONI             CASH             6000
DISHA PATANI         CASH             3000
AKSHAY KUMAR         CASH             6000
SUSHMITA SEN         CHEQUE           3000
ROHIT SHARMA         CASH             8500
JUHI CHAWLA          CHEQUE           8500
PRAFULLA DAVE        CASH             4500
LOKESH RAHUL         CASH             4500
ASHOK DINDA          CASH             4500

9 rows selected.
```

6. FUTURE ENHANCEMENTS OF THE SYSTEM

This project has been done very systematically and all the efforts have been taken so that the project would be efficient and a success. Though, there still exists a scope of improvement for our project. The system has been developed in such a way that it can accept modifications and further changes.

The following could be the enhancements that can be incorporated in our project:

- ✓ Automated fitness suggestion by evaluating the health conditions on a regular basis
- ✓ Provide more modes for payment
- ✓ The system would automatically assign the member and login id whenever a new member joins or a new person creates an account on the website
- ✓ Provide more data accuracy and a greater efficiency
- ✓ More user friendly and interactive
- ✓ System would take a lesser time to execute than the current system.

7. BIBLIOGRAPHY

For the successful working of our current project, various sources have been referred. Even though most of the references were taken from the book: Database System Concepts, we have referenced a few websites for further help.

Reference Book:
Database System Concepts
- Henry F.Korth & A.Silberschatz. 2nd Ed. McGraw-Hill 1991

Website References:

- Google for problem solving
- http://www.tutorialspoint.com/mysql/
- https://apache.org/docs/2.0/mics/tutorials.html
- https://stackoverflow.com/
- https://docs.oracle.com/en/