DATABASE MANAGEMENT SYSTEM-

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

GYM MANAGEMENT SYSTEM

MEMBERS:

RIDHAM UPPAL
PALAK MAHAJAN

CONTENTS:

PROJECT DESCRIPTION:

Gym management software is a powerful tool designed to assist fitness businesses in managing all aspects of their operations. This software offers a variety of features that enable fitness owners and operators to manage their studio efficiently. Gym

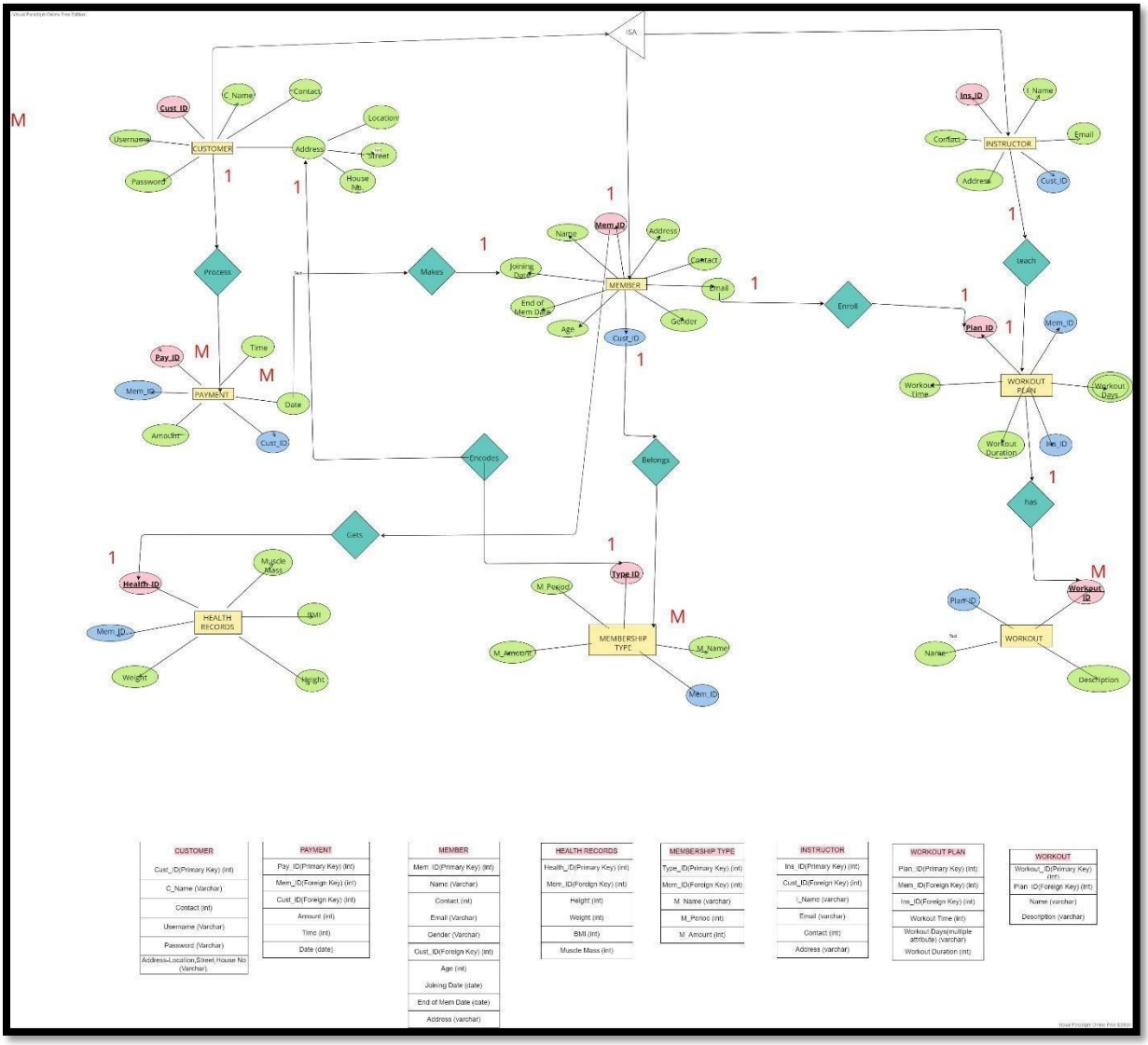
management software provides tools for scheduling classes and trainers, tracking members, communicating with clients, and processing payments. With the automation of various tasks, gym management software helps fitness businesses streamline their operations and achieve greater efficiency.

FUNCTIONALITIES:

Gym management software provides a range of useful features to help fitness businesses efficiently manage their operations, including:

- Storing the details of visiting customers
- Maintaining a record of all payments made
- Keeping a separate record of all members
- Keeping a separate record of all instructors
- Recording all workout plans
- Tracking different workouts
- Maintaining a detailed record of membership types
- Maintaining separate health records for members

ENTITY RELATIONSHIP DIAGRAM:

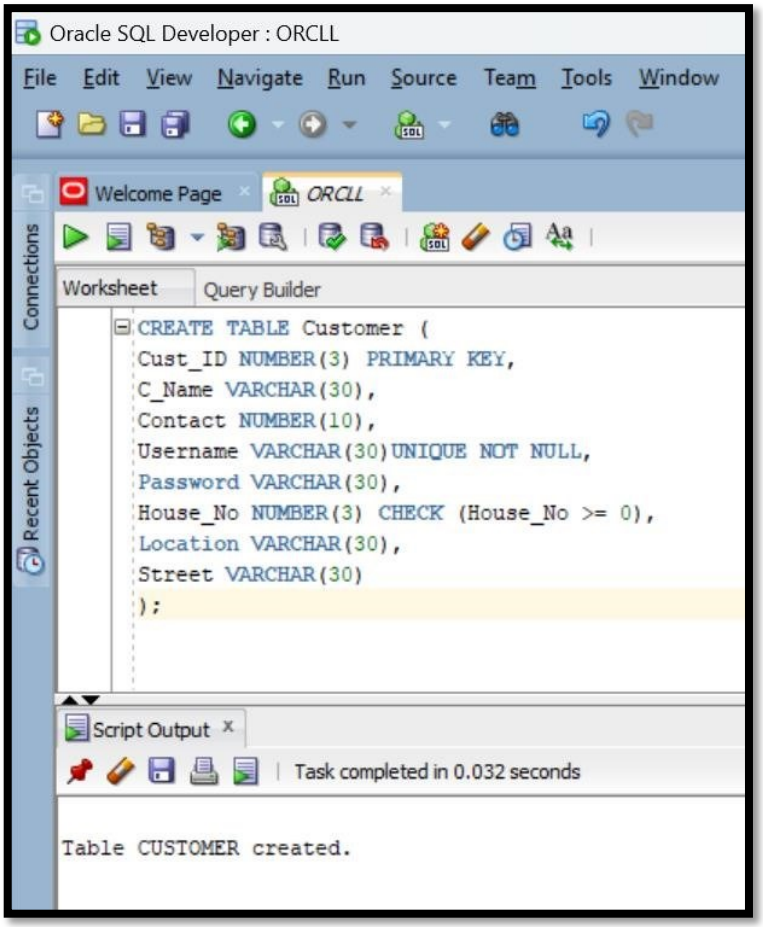


CREATING TABLES AND RELATIONS:

1. Customer Table

```
CREATE TABLE Customer (
Cust_ID NUMBER(3) PRIMARY KEY,
C_Name VARCHAR(30),
Contact NUMBER(10),
Username VARCHAR(30)UNIQUE NOT NULL,
Password VARCHAR(30),
House_No NUMBER(3) CHECK (House_No >= 0),
Location VARCHAR(30),
Street VARCHAR(30)
```

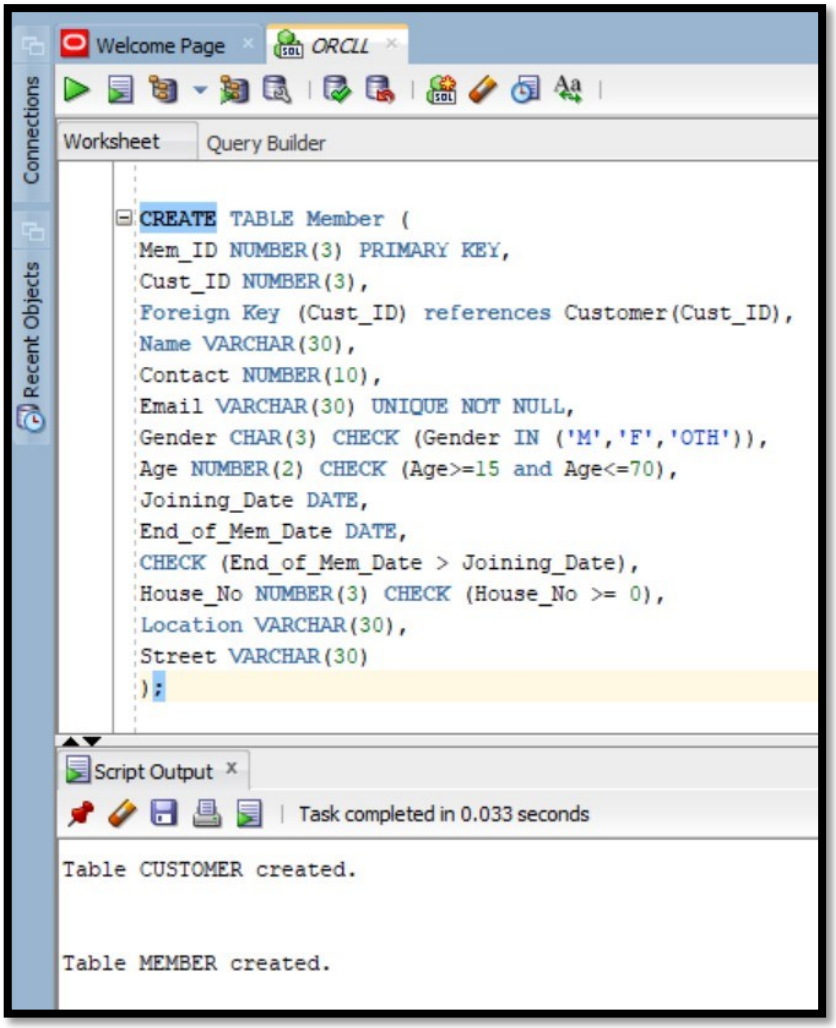
```
);
```



2. Member Table

```
CREATE TABLE Member (
Mem_ID NUMBER(3) PRIMARY KEY,
Cust_ID NUMBER(3),
Foreign Key (Cust_ID) references Customer(Cust_ID),
Name VARCHAR(30),
Contact NUMBER(10),
Email VARCHAR(30) UNIQUE NOT NULL,
Gender CHAR(3) CHECK (Gender IN ('M','F','OTH')),
Age NUMBER(2) CHECK (Age>=15 and Age<=70),
Joining_Date DATE,
End_of_Mem_Date DATE,
CHECK (End_of_Mem_Date > Joining_Date),
```

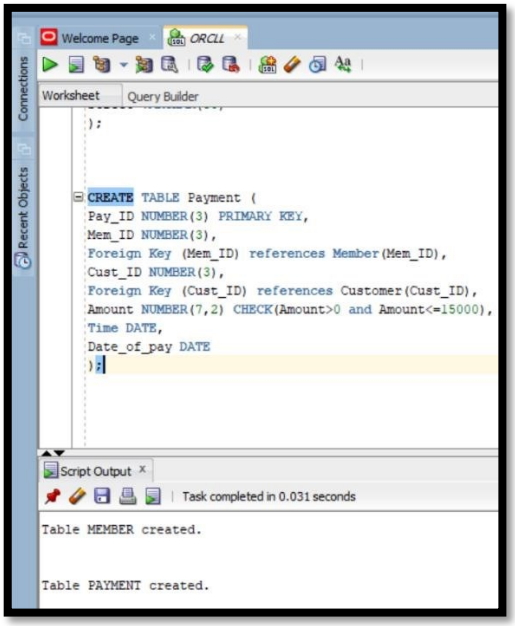
```
House_No NUMBER(3) CHECK (House_No >= 0),
Location VARCHAR(30),
Street VARCHAR(30)
);
```



3. Payment Table

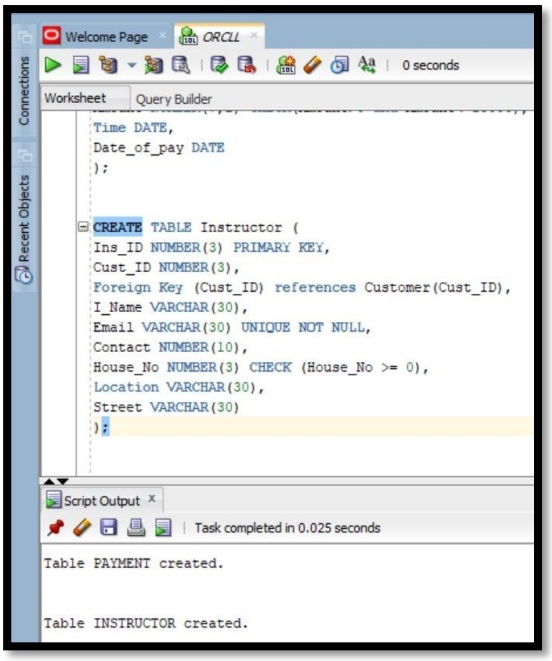
```
CREATE TABLE Payment (
Pay_ID NUMBER(3) PRIMARY KEY,
Mem_ID NUMBER(3),
Foreign Key (Mem_ID) references Member(Mem_ID),
Cust_ID NUMBER(3),
Foreign Key (Cust_ID) references Customer(Cust_ID),
Amount NUMBER(7,2) CHECK(Amount>0 and Amount<=15000),
Time DATE,
Date_of_pay DATE
```

```
);
```



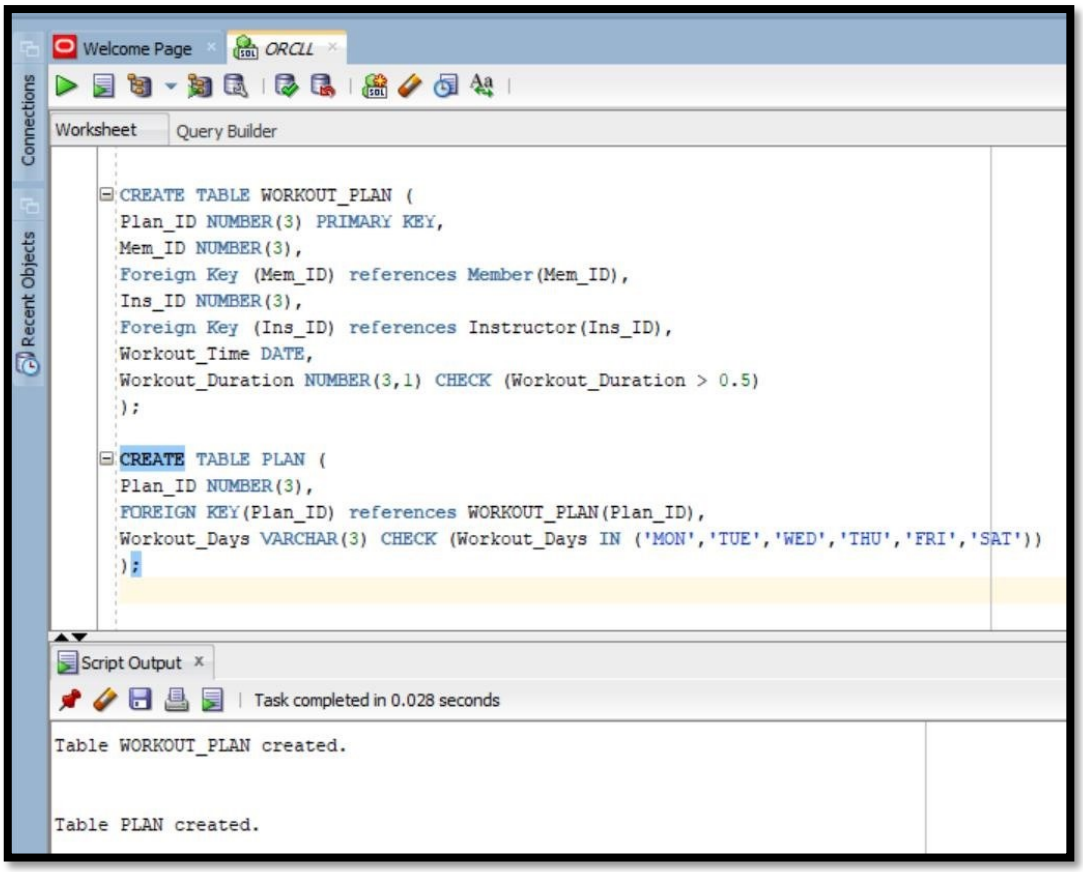
4. Instructor Table

```
CREATE TABLE Instructor (
Ins_ID NUMBER(3) PRIMARY KEY,
Cust_ID NUMBER(3),
Foreign Key (Cust_ID) references Customer(Cust_ID),
I_Name VARCHAR(30),
Email VARCHAR(30) UNIQUE NOT NULL,
Contact NUMBER(10),
House_No NUMBER(3) CHECK (House_No >= 0),
Location VARCHAR(30),
Street VARCHAR(30)
);
```



5. Workout Plan Table

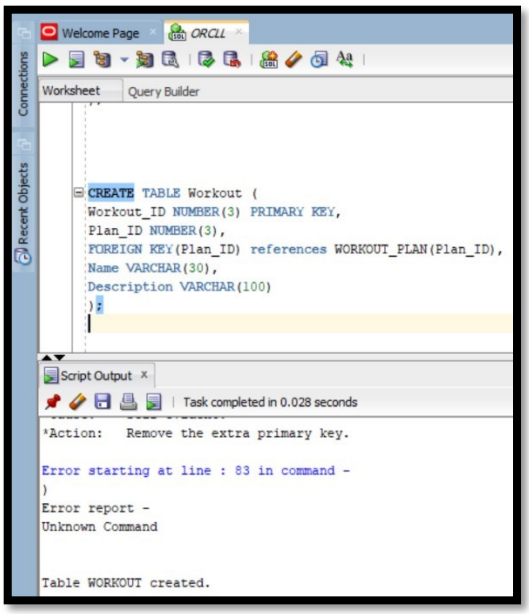
```
CREATE TABLE WORKOUT_PLAN (
Plan_ID NUMBER(3) PRIMARY KEY,
Mem_ID NUMBER(3),
Foreign Key (Mem_ID) references Member(Mem_ID),
Ins_ID NUMBER(3),
Foreign Key (Ins_ID) references Instructor(Ins_ID),
Workout_Time DATE,
Workout_Duration NUMBER(3,1) CHECK (Workout_Duration > 0.5)
);

CREATE TABLE PLAN (
Plan_ID NUMBER(3),
FOREIGN KEY(Plan_ID) references WORKOUT_PLAN(Plan_ID),
Workout_Days VARCHAR(3) CHECK (Workout_Days IN
('MON','TUE','WED','THU','FRI','SAT'))
);
```



6. Workout Table

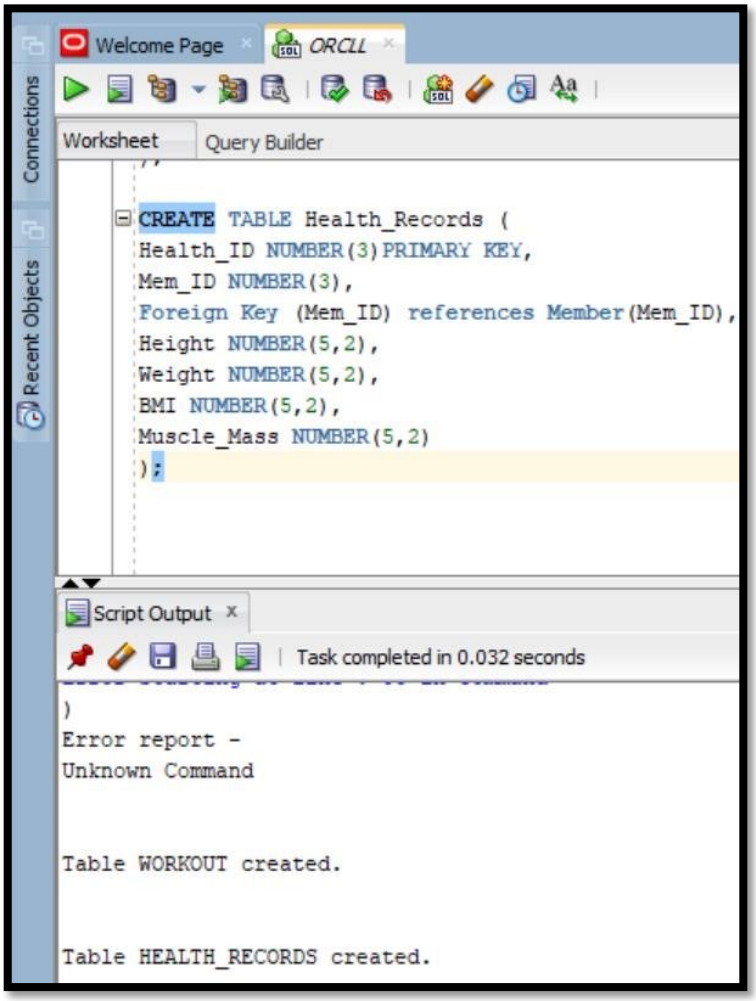
```
CREATE TABLE Workout (
Workout_ID NUMBER(3) PRIMARY KEY,
Plan_ID NUMBER(3),
FOREIGN KEY(Plan_ID) references WORKOUT_PLAN(Plan_ID),
Name VARCHAR(30),
Description VARCHAR(100)
```

```
);
```



7. Health Records Table

```
CREATE TABLE Health_Records (
Health_ID NUMBER(3)PRIMARY KEY,
Mem_ID NUMBER(3),
Foreign Key (Mem_ID) references Member(Mem_ID),
Height NUMBER(5,2),
Weight NUMBER(5,2),
BMI NUMBER(5,2),
Muscle_Mass NUMBER(5,2)
)
```



INSERTING RECORDS:

1. Customer Table

```
INSERT INTO Customer VALUES (114,'Kavya
Aggarwal',9849034012,'KAggarwal','kavya123',24,'Delhi','Dwarka');

INSERT INTO Customer VALUES (115,'Vikram
Gupta',9478273645,'VGupta','vikram22',37,'Mumbai','Bandra');

INSERT INTO Customer VALUES (116,'Shikha
Singh',9898787678,'SSingh','shikha09',28,'Delhi','Saket');
```

Welcome Page | ORCL | CUSTOMER

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	CUST_ID	C_NAME	CONTACT	USERNAME	PASSWORD	HOUSE_NO	LOCATION	STREET
1	114	Kavya Aggarwal	9849034012	KAggarwal	kavya123	24	Delhi	Dwarka
2	115	Vikram Gupta	9478273645	VGupta	vikram22	37	Mumbai	Bandra
3	116	Shikha Singh	9898787678	SSingh	shikha09	28	Delhi	Saket
4	117	Gaurav Kumar	9976770070	GKumar	gaurav321	31	Chennai	Adyar
5	118	Pooja Gupta	9999456789	PGupta	pooja456	23	Mumbai	Juhu
6	119	Rohit Kumar	9876501234	RKumar	rohit222	33	Bangalore	Indiranagar
7	120	Nikita Singh	9988776655	NSingh	nikita33	25	Delhi	Laxmi Nagar
8	121	Ajay Mehta	9654321098	AMehta	ajay123	45	Kolkata	Salt Lake
9	122	Ritu Gupta	9988776655	RGupta	ritu123	28	Bangalore	HSR Layout

2. Member Table

```
INSERT INTO Member VALUES (1,114, 'Kavya Aggarwal', 9849034012,
'kavya.aggarwal@example.com', 'F', 24, '01-Jan-2022', '01-Jan-
2023',101, 'Delhi', 'Dwarka');

INSERT INTO Member VALUES (2,115, 'Vikram Gupta', 9478273645,
'vikram.gupta@example.com', 'M', 37, '01-Jan-2022', '31-Dec-2023',201,
'Mumbai', 'Bandra');

INSERT INTO Member VALUES (3,116, 'Shikha Singh', 9898787678,
'shikha.singh@example.com', 'F', 28, '01-Jan-2022', '30-Jun-2023',301,
'Delhi', 'Saket');
```

Welcome Page | ORCL | MEMBER

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	MEM_ID	CUST_ID	NAME	CONTACT	EMAIL	GENDER	AGE	JOINING_DATE	END_OF_MEM_DATE	HOUSE_NO	LOCATION	STREET
1	1	114	Kavya Aggarwal	9849034012	kavya.aggarwal@example.com	F	24	01-01-22	01-01-23	101	Delhi	Dwarka
2	2	115	Vikram Gupta	9478273645	vikram.gupta@example.com	M	37	01-01-22	31-12-23	201	Mumbai	Bandra
3	3	116	Shikha Singh	9898787678	shikha.singh@example.com	F	28	01-01-22	30-06-23	301	Delhi	Saket
4	4	117	Gaurav Kumar	9976770070	gaurav.kumar@example.com	M	31	01-01-22	31-12-23	401	Chennai	Adyar
5	5	118	Pooja Gupta	9999456789	pooja.gupta@example.com	F	23	01-01-22	30-06-23	501	Mumbai	Juhu
6	6	119	Rohit Kumar	9876501234	rohit.kumar@example.com	M	33	01-01-22	31-12-23	601	Bangalore	Indiranagar
7	7	120	Nikita Singh	9988776655	nikita.singh@example.com	F	25	01-01-22	30-06-23	701	Delhi	Laxmi Nagar

3. Instructor Table

```
INSERT INTO Instructor VALUES (301,121,'Ajay Mehta',
'ajay.mehta@example.com',9654321098,45,'Kolkata','Salt Lake');

INSERT INTO Instructor VALUES (302,122,'Ritu Gupta',
'ritu.gupta@example.com',9988776655,28,'Bangalore','HSR Layout');
```

Welcome Page | ORCLL | INSTRUCTOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	INS_ID	CUST_ID	I_NAME	EMAIL	CONTACT	HOUSE_NO	LOCATION	STREET
1	301	121	Ajay Mehta	ajay.mehta@example.com	9654321098	45	Kolkata	Salt Lake
2	302	122	Ritu Gupta	ritu.gupta@example.com	9988776655	28	Bangalore	HSR Layout

4. Payment Table

```
--For Member Number 1-2 payments

INSERT INTO Payment VALUES(101,1,114,6000,to_date('2022-Jun-1',
'YYYYMON-DD'),'1-Jan- 2022');

INSERT INTO Payment VALUES(102,1,114,6000,to_date('2023-Jan-1', 'YYYY-
MON-DD'),'1-Jun-2022');
```

Welcome Page | ORCLL | PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies

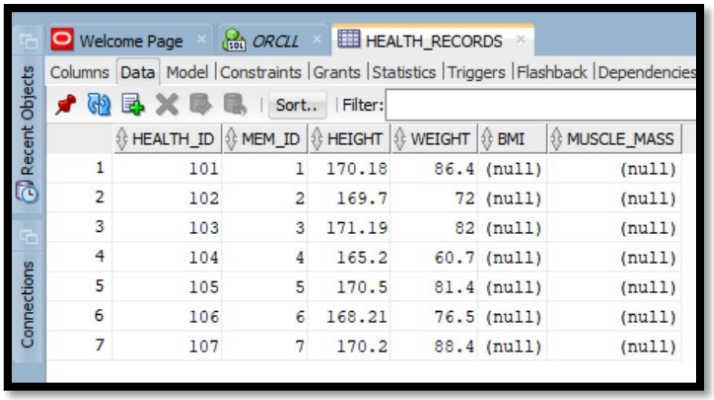
Sort.. | Filter:

	PAY_ID	MEM_ID	CUST_ID	AMOUNT	TIME	DATE_OF_PAY
1	101	1	114	6000	01-06-22	01-01-22
2	102	1	114	6000	01-01-23	01-06-22
3	201	2	115	6000	01-06-22	01-01-22
4	202	2	115	6000	01-12-22	01-06-22
5	203	2	115	6000	01-06-23	01-12-22
6	204	2	115	6000	31-12-23	01-06-23
7	301	3	116	6000	01-06-22	01-01-22
8	302	3	116	6000	01-12-22	01-06-22
9	303	3	116	6000	30-06-23	01-12-23
10	401	4	117	6000	01-06-22	01-01-22
11	402	4	117	6000	01-12-22	01-06-22
12	403	4	117	6000	01-06-23	01-12-22
13	404	4	117	6000	31-12-23	01-06-23
14	501	5	118	6000	01-06-22	01-01-22
15	502	5	118	6000	01-12-22	01-06-22
16	503	5	118	6000	30-06-23	01-12-22
17	601	6	119	6000	01-06-22	01-01-22
18	602	6	119	6000	01-12-22	01-06-22
19	603	6	119	6000	01-06-23	01-12-22
20	604	6	119	6000	31-12-23	01-06-23
21	701	7	120	6000	01-06-22	01-01-22
22	702	7	120	6000	01-12-22	01-06-22
23	703	7	120	6000	30-06-23	01-12-22

5. Health Records Table

```
INSERT INTO Health_Records VALUES(101,1,170.18,86.4,NULL,NULL);

INSERT INTO Health_Records VALUES(102,2,169.7,72,NULL,NULL);
```

```
INSERT INTO Health_Records VALUES(103,3,171.19,82,NULL,NULL);
```

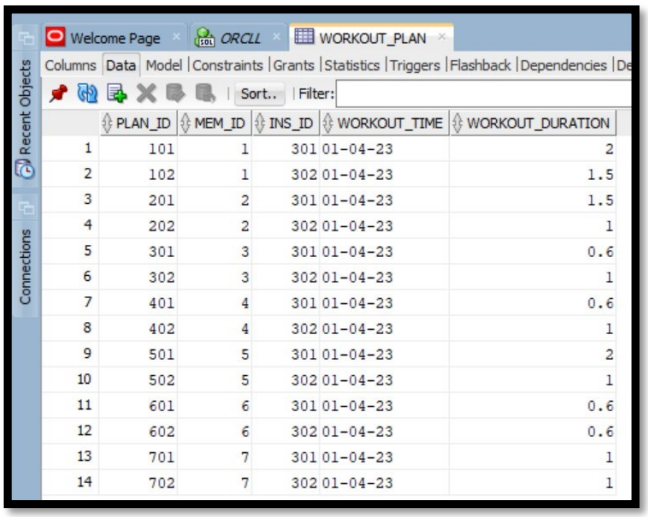


	HEALTH_ID	MEM_ID	HEIGHT	WEIGHT	BMI	MUSCLE_MASS
1	101	1	170.18	86.4	(null)	(null)
2	102	2	169.7	72	(null)	(null)
3	103	3	171.19	82	(null)	(null)
4	104	4	165.2	60.7	(null)	(null)
5	105	5	170.5	81.4	(null)	(null)
6	106	6	168.21	76.5	(null)	(null)
7	107	7	170.2	88.4	(null)	(null)

6. Workout Plan Table

--2 plans for each member

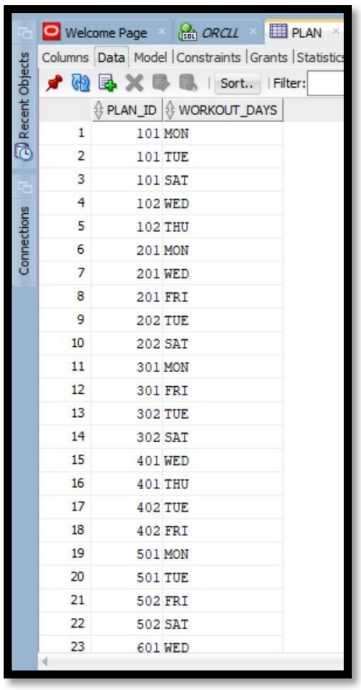
```
INSERT INTO Workout_Plan
   VALUES(101,1,301,to_date('16:00:00','hh24:mi:ss'),2);

INSERT INTO Workout_Plan
   VALUES(102,1,302,to_date('17:00:00','hh24:mi:ss'),1.5);

INSERT INTO Workout_Plan
   VALUES(201,2,301,to_date('7:00:00','hh24:mi:ss'),1.5);
```



	PLAN_ID	MEM_ID	INS_ID	WORKOUT_TIME	WORKOUT_DURATION
1	101	1	301	01-04-23	2
2	102	1	302	01-04-23	1.5
3	201	2	301	01-04-23	1.5
4	202	2	302	01-04-23	1
5	301	3	301	01-04-23	0.6
6	302	3	302	01-04-23	1
7	401	4	301	01-04-23	0.6
8	402	4	302	01-04-23	1
9	501	5	301	01-04-23	2
10	502	5	302	01-04-23	1
11	601	6	301	01-04-23	0.6
12	602	6	302	01-04-23	0.6
13	701	7	301	01-04-23	1
14	702	7	302	01-04-23	1

Plan Table

```
INSERT      INTO Plan VALUES(101,'MON');

INSERT      INTO Plan VALUES(101,'TUE');

INSERT      INTO Plan VALUES(101,'SAT');
```



	PLAN_ID	WORKOUT_DAYS
1	101	MON
2	101	TUE
3	101	SAT
4	102	WED
5	102	THU
6	201	MON
7	201	WED
8	201	FRI
9	202	TUE
10	202	SAT
11	301	MON
12	301	FRI
13	302	TUE
14	302	SAT
15	401	WED
16	401	THU
17	402	TUE
18	402	FRI
19	501	MON
20	501	TUE
21	502	FRI
22	502	SAT
23	601	WED

7. Workout Table

```
INSERT INTO Workout     VALUES(1,101,'Treadmill','Running practice');

INSERT INTO Workout     VALUES(2,101,'Cycling','Thigh control');

INSERT INTO Workout     VALUES(3,101,'Dumbell','Weight Training');

INSERT INTO Workout     VALUES(4,102,'Burpees','Strength Traning');
```

Welcome Page | ORCLL | WORKOUT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Depen

Sort.. | Filter:

	WORKOUT_ID	PLAN_ID	NAME	DESCRIPTION
1	1	101	Treadmill	Running practice
2	2	101	Cycling	Thigh control
3	3	101	Dumbell	Weight Training
4	4	102	Burpees	Strength Traning
5	5	101	Pushups	Strength Training
6	6	101	Side planks	Strength Training
7	7	102	Pull Ups	Waist Control
8	8	102	Squats	Thigh control
9	9	201	Treadmill	Running practice
10	10	201	Cycling	Thigh control
11	11	201	Dumbell	Weight Training
12	12	202	Pushups	Strength Training
13	13	202	Burpees	Strength Traning
14	14	301	Side planks	Strength Training
15	15	302	Squats	Thigh control
16	16	301	Treadmill	Running practice
17	17	302	Cycling	Thigh control
18	18	401	Dumbell	Weight Training
19	19	401	Burpees	Strength Traning
20	20	402	Treadmill	Running practice
21	21	402	Cycling	Thigh control
22	22	501	Side planks	Strength Training
23	23	501	Squats	Thigh control

NORMALIZATION OF TABLES:

1. Customer Table

Welcome Page | ORCLL | CUSTOMER

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

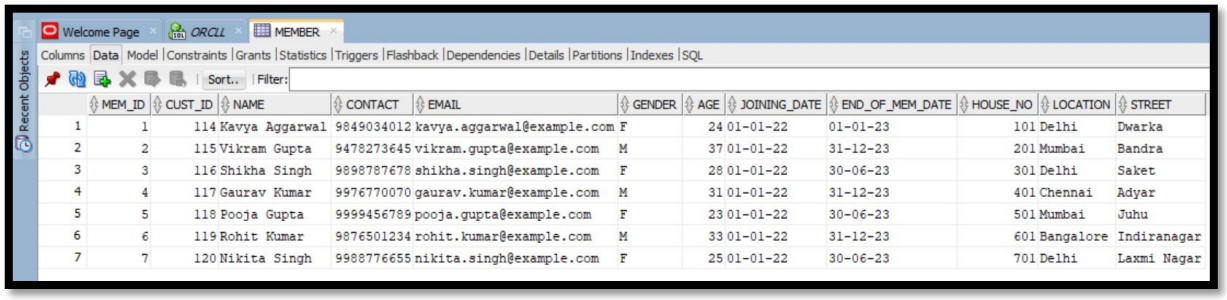
	CUST_ID	C_NAME	CONTACT	USERNAME	PASSWORD	HOUSE_NO	LOCATION	STREET
1	114	Kavya Aggarwal	9849034012	KAggarwal	kavya123	24	Delhi	Dwarka
2	115	Vikram Gupta	9478273645	VGupta	vikram22	37	Mumbai	Bandra
3	116	Shikha Singh	9898787678	SSingh	shikha09	28	Delhi	Saket
4	117	Gaurav Kumar	9976770070	GKumar	gaurav321	31	Chennai	Adyar
5	118	Pooja Gupta	9999456789	PGupta	pooja456	23	Mumbai	Juhu
6	119	Rohit Kumar	9876501234	RKumar	rohit222	33	Bangalore	Indiranagar
7	120	Nikita Singh	9988776655	NSingh	nikita33	25	Delhi	Laxmi Nagar
8	121	Ajay Mehta	9654321098	AMehta	ajay123	45	Kolkata	Salt Lake
9	122	Ritu Gupta	9988776655	RGupta	ritu123	28	Bangalore	HSR Layout

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the CUSTOMER table contains only single value in each cell, therefore it is in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

2. Member Table



	MEM_ID	CUST_ID	NAME	CONTACT	EMAIL	GENDER	AGE	JOINING_DATE	END_OF_MEM_DATE	HOUSE_NO	LOCATION	STREET
1	1	114	Kavya Aggarwal	9849034012	kavya.aggarwal@example.com	F	24	01-01-22	01-01-23	101	Delhi	Dwarka
2	2	115	Vikram Gupta	9478273645	vikram.gupta@example.com	M	37	01-01-22	31-12-23	201	Mumbai	Bandra
3	3	116	Shikha Singh	9898787678	shikha.singh@example.com	F	28	01-01-22	30-06-23	301	Delhi	Saket
4	4	117	Gaurav Kumar	9976770070	gaurav.kumar@example.com	M	31	01-01-22	31-12-23	401	Chennai	Adyar
5	5	118	Pooja Gupta	9999456789	pooja.gupta@example.com	F	23	01-01-22	30-06-23	501	Mumbai	Juhu
6	6	119	Rohit Kumar	9876501234	rohit.kumar@example.com	M	33	01-01-22	31-12-23	601	Bangalore	Indiranagar
7	7	120	Nikita Singh	9988776655	nikita.singh@example.com	F	25	01-01-22	30-06-23	701	Delhi	Laxmi Nagar

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the MEMBER table contains only single value in each cell, therefore it is in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

3. Payment Table

Welcome Page | ORCLL | PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies

Sort.. | Filter:

	PAY_ID	MEM_ID	CUST_ID	AMOUNT	TIME	DATE_OF_PAY
1	101	1	114	6000	01-06-22	01-01-22
2	102	1	114	6000	01-01-23	01-06-22
3	201	2	115	6000	01-06-22	01-01-22
4	202	2	115	6000	01-12-22	01-06-22
5	203	2	115	6000	01-06-23	01-12-22
6	204	2	115	6000	31-12-23	01-06-23
7	301	3	116	6000	01-06-22	01-01-22
8	302	3	116	6000	01-12-22	01-06-22
9	303	3	116	6000	30-06-23	01-12-23
10	401	4	117	6000	01-06-22	01-01-22
11	402	4	117	6000	01-12-22	01-06-22
12	403	4	117	6000	01-06-23	01-12-22
13	404	4	117	6000	31-12-23	01-06-23
14	501	5	118	6000	01-06-22	01-01-22
15	502	5	118	6000	01-12-22	01-06-22
16	503	5	118	6000	30-06-23	01-12-22
17	601	6	119	6000	01-06-22	01-01-22
18	602	6	119	6000	01-12-22	01-06-22
19	603	6	119	6000	01-06-23	01-12-22
20	604	6	119	6000	31-12-23	01-06-23
21	701	7	120	6000	01-06-22	01-01-22
22	702	7	120	6000	01-12-22	01-06-22
23	703	7	120	6000	30-06-23	01-12-22

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the PAYMENT table contains only single value in each cell, therefore it is in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

4. Instructor Table

Welcome Page | ORCLL | INSTRUCTOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

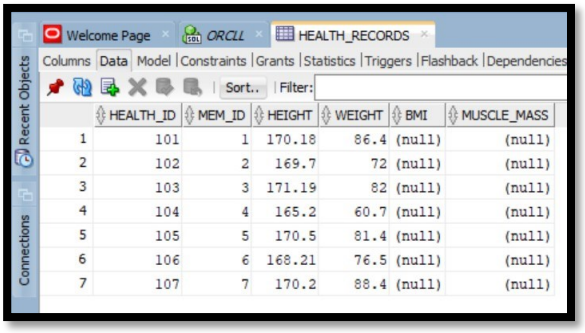
	INS_ID	CUST_ID	I_NAME	EMAIL	CONTACT	HOUSE_NO	LOCATION	STREET
1	301	121	Ajay Mehta	ajay.mehta@example.com	9654321098	45	Kolkata	Salt Lake
2	302	122	Ritu Gupta	ritu.gupta@example.com	9988776655	28	Bangalore	HSR Layout

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the INSTRUCTOR table contains only single value in each cell, therefore it is in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

5. Health records Table



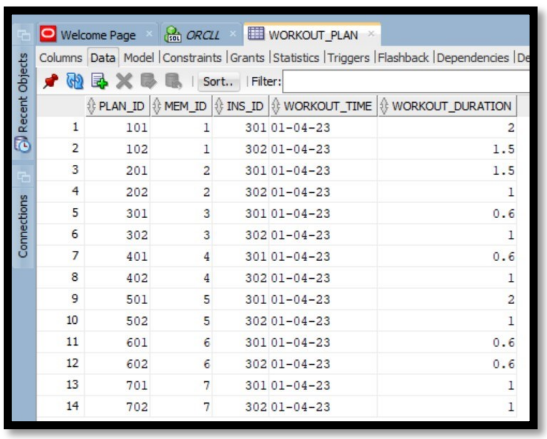
	HEALTH_ID	MEM_ID	HEIGHT	WEIGHT	BMI	MUSCLE_MASS
1	101	1	170.18	86.4	(null)	(null)
2	102	2	169.7	72	(null)	(null)
3	103	3	171.19	82	(null)	(null)
4	104	4	165.2	60.7	(null)	(null)
5	105	5	170.5	81.4	(null)	(null)
6	106	6	168.21	76.5	(null)	(null)
7	107	7	170.2	88.4	(null)	(null)

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the HEALTH RECORDS table contains only single value in each cell, therefore it is in 1NF.

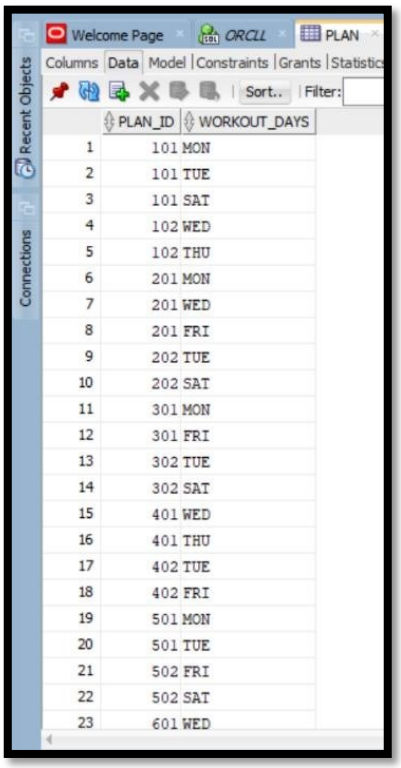
- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

6. Workout Plan Table



	PLAN_ID	MEM_ID	INS_ID	WORKOUT_TIME	WORKOUT_DURATION
1	101	1	301	01-04-23	2
2	102	1	302	01-04-23	1.5
3	201	2	301	01-04-23	1.5
4	202	2	302	01-04-23	1
5	301	3	301	01-04-23	0.6
6	302	3	302	01-04-23	1
7	401	4	301	01-04-23	0.6
8	402	4	302	01-04-23	1
9	501	5	301	01-04-23	2
10	502	5	302	01-04-23	1
11	601	6	301	01-04-23	0.6
12	602	6	302	01-04-23	0.6
13	701	7	301	01-04-23	1
14	702	7	302	01-04-23	1



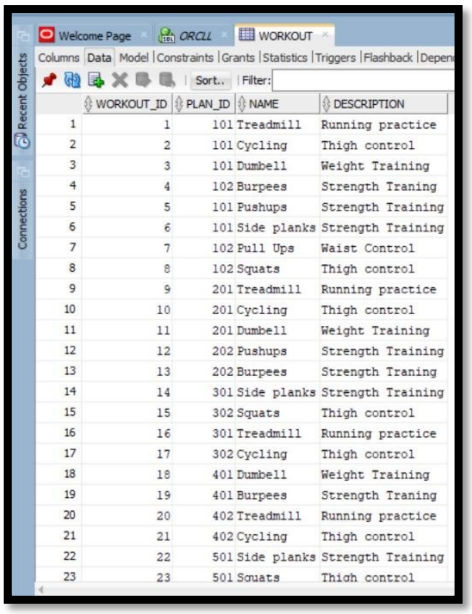
	PLAN_ID	WORKOUT_DAYS
1	101	MON
2	101	TUE
3	101	SAT
4	102	WED
5	102	THU
6	201	MON
7	201	WED
8	201	FRI
9	202	TUE
10	202	SAT
11	301	MON
12	301	FRI
13	302	TUE
14	302	SAT
15	401	WED
16	401	THU
17	402	TUE
18	402	FRI
19	501	MON
20	501	TUE
21	502	FRI
22	502	SAT
23	601	WED

- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Here, the column **workout_days** was earlier not normalized in 1NF as it contained multi-valued attributes. Therefore, we have normalized it and divided it into two separate tables. The first one contains all the other attributes and the second table contains the primary key **plan_ID** and the **workout_days.** Now, both the tables contain only single value in each cell, therefore they are in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Since the above table is in 1NF and contains only a single primary key and there is no partial dependency, therefore it is in 2NF.

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

7. Workout Table



	WORKOUT_ID	PLAN_ID	NAME	DESCRIPTION
1	1	101	Treadmill	Running practice
2	2	101	Cycling	Thigh control
3	3	101	Dumbell	Weight Training
4	4	102	Burpees	Strength Traning
5	5	101	Pushups	Strength Training
6	6	101	Side planks	Strength Training
7	7	102	Pull Ups	Waist Control
8	8	102	Squats	Thigh control
9	9	201	Treadmill	Running practice
10	10	201	Cycling	Thigh control
11	11	201	Dumbell	Weight Training
12	12	202	Pushups	Strength Training
13	13	202	Burpees	Strength Traning
14	14	301	Side planks	Strength Training
15	15	302	Squats	Thigh control
16	16	301	Treadmill	Running practice
17	17	302	Cycling	Thigh control
18	18	401	Dumbell	Weight Training
19	19	401	Burpees	Strength Traning
20	20	402	Treadmill	Running practice
21	21	402	Cycling	Thigh control
22	22	501	Side planks	Strength Training
23	23	501	Squats	Thigh control

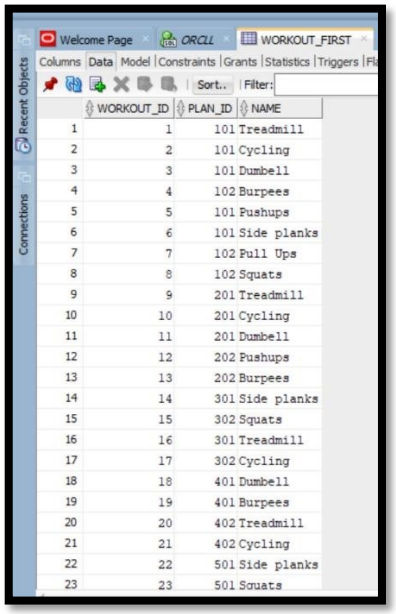
- **1NF:** A relation R is in first normal form (1NF) if and only if all underlying domains contain atomic values only. Since the WORKOUT table contains only single value in each cell, therefore it is in 1NF.

- **2NF:** A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-primary key attribute is fully dependent on the primary key. Here we need to divide it into two tables in order to bring it in the second normal form, **workout_first** (workout_ID, plan_ID, name) where the primary key is a combination of workout_ID and plan_ID. And, then there is **workout_second** (name, description) where the primary key is name.

a. Workout First

```
CREATE TABLE WORKOUT_FIRST (
Workout_ID NUMBER(3),
```

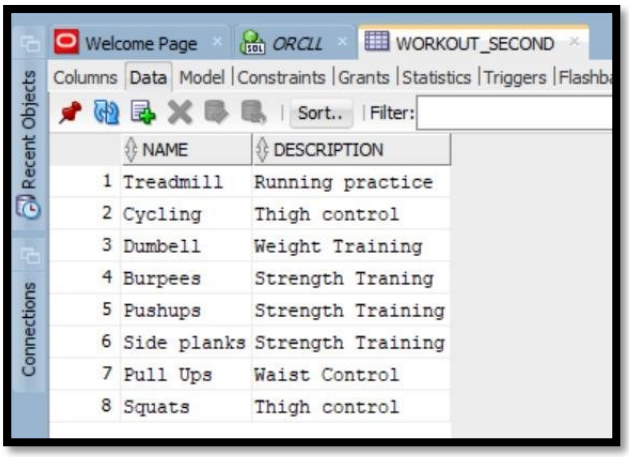
```
Plan_ID NUMBER(3),
Name VARCHAR(30),
PRIMARY KEY(Workout_ID,Plan_ID)
   );

  INSERT INTO Workout_first VALUES(1,101,'Treadmill');

  INSERT INTO Workout_first VALUES(2,101,'Cycling');

  INSERT INTO Workout_first VALUES(3,101,'Dumbell');
```



	WORKOUT_ID	PLAN_ID	NAME
1	1	101	Treadmill
2	2	101	Cycling
3	3	101	Dumbell
4	4	102	Burpees
5	5	101	Pushups
6	6	101	Side planks
7	7	102	Pull Ups
8	8	102	Squats
9	9	201	Treadmill
10	10	201	Cycling
11	11	201	Dumbell
12	12	202	Pushups
13	13	202	Burpees
14	14	301	Side planks
15	15	302	Squats
16	16	301	Treadmill
17	17	302	Cycling
18	18	401	Dumbell
19	19	401	Burpees
20	20	402	Treadmill
21	21	402	Cycling
22	22	501	Side planks
23	23	501	Squats

b. Workout Second

```
CREATE TABLE Workout_Second (
Name VARCHAR(30) PRIMARY KEY,
Description VARCHAR(100)
);


INSERT INTO Workout_second VALUES('Treadmill','Running practice');

INSERT INTO Workout_second VALUES('Cycling','Thigh control');
```



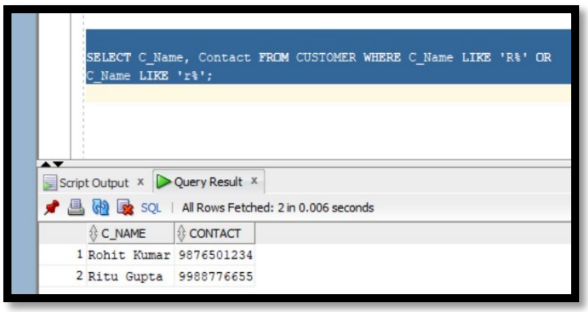
	NAME	DESCRIPTION
1	Treadmill	Running practice
2	Cycling	Thigh control
3	Dumbell	Weight Training
4	Burpees	Strength Traning
5	Pushups	Strength Training
6	Side planks	Strength Training
7	Pull Ups	Waist Control
8	Squats	Thigh control

- **3NF:** A relation R is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key. Since the above table does not have any transitive dependencies, therefore it is in 3NF.

BASIC SQL QUERIES

1) **If the Administrator wants to filter the customers on the basis of names.**

```
SELECT C_Name, Contact FROM CUSTOMER WHERE C_Name LIKE 'R%' OR C_Name
   LIKE 'r%';
```



```
SELECT C_Name, Contact FROM CUSTOMER WHERE C_Name LIKE 'R%' OR
C_Name LIKE 'r%';
```

Script Output | Query Result

All Rows Fetched: 2 in 0.006 seconds

	C_NAME	CONTACT
1	Rohit Kumar	9876501234
2	Ritu Gupta	9988776655

2) **If the customer loses his password, the administrator can give him on the basis of his Cust_ID.**

```
SELECT password FROM CUSTOMER WHERE Username='KAggarwal';
```

```
SELECT password FROM CUSTOMER WHERE Username='KAggarwal';
```

Script Output | Query Result

All Rows Fetched: 1 in 0.004 seconds

	PASSWORD
1	kavya123

3) **If any member has ended his membership, that can be updated in the Member table.**

```
UPDATE Member SET end_of_mem_date='15-MAY-2023' WHERE Mem_ID=5;
```

MEMBER

	MEM_ID	CUST_ID	NAME	CONTACT	EMAIL	GENDER	AGE	JOINING_DATE	END_OF_MEM_DATE	HOUSE_NO	LOCATION	STREET
1	1	114	Kavya Aggarwal	9849034012	kavya.aggarwal@example.com	F	24	01-01-22	01-01-23	101	Delhi	Dwarka
2	2	115	Vikram Gupta	9478273645	vikram.gupta@example.com	M	37	01-01-22	31-12-23	201	Mumbai	Bandra
3	3	116	Shikha Singh	9898787678	shikha.singh@example.com	F	28	01-01-22	30-06-23	301	Delhi	Saket
4	4	117	Gaurav Kumar	9976770070	gaurav.kumar@example.com	M	31	01-01-22	31-12-23	401	Chennai	Adyar
5	5	118	Pooja Gupta	9999456789	pooja.gupta@example.com	F	23	01-01-22	15-05-23	501	Mumbai	Juhu
6	6	119	Rohit Kumar	9876501234	rohit.kumar@example.com	M	33	01-01-22	31-12-23	601	Bangalore	Indiranagar
7	7	120	Nikita Singh	9988776655	nikita.singh@example.com	F	25	01-01-22	30-06-23	701	Delhi	Laxmi Nagar

4) **If the administrator wishes to locate all the members from Mumbai.**

```
SELECT Name from MEMBER where Location='Mumbai';
```

Script Output | Query Result

All Rows Fetched: 2 in 0.003 seconds

	NAME
1	Vikram Gupta
2	Pooja Gupta

5) **If the Administrator wishes to find the Cust_ID of an Instructor.**

```
SELECT Cust_ID from INSTRUCTOR where Ins_ID=302;
```

```
SELECT Cust_ID from INSTRUCTOR where Ins_ID=302;
```

Script Output | Query Result

All Rows Fetched: 1 in 0.004 seconds

	CUST_ID
1	122

6) **If the Administrator wants to see the members in the descending order of weight.**

```
SELECT * FROM HEALTH_RECORDS ORDER BY weight desc;
```

```
SELECT * FROM HEALTH_RECORDS ORDER BY weight desc;
```

Script Output | Query Result

All Rows Fetched: 7 in 0.007 seconds

	HEALTH_ID	MEM_ID	HEIGHT	WEIGHT	BMI	MUSCLE_MASS
1	107	7	170.2	88.4	(null)	(null)
2	101	1	170.18	86.4	(null)	(null)
3	103	3	171.19	82	(null)	(null)
4	105	5	170.5	81.4	(null)	(null)
5	106	6	168.21	76.5	(null)	(null)
6	102	2	169.7	72	(null)	(null)
7	104	4	165.2	60.7	(null)	(null)

7) **If the Administrator wants to see the average height of all the members.**

```
SELECT AVG(Height) FROM HEALTH_RECORDS;
```

```
SELECT AVG(Height) FROM HEALTH_RECORDS;
```

Script Output | Query Result

All Rows Fetched: 1 in 0.003 seconds

	AVG(HEIGHT)
1	169.3114285714285714285714285714285714 29

8) **If the Administrator wants to see the number of customers paying the same amount.**

```
SELECT COUNT(PAY_ID), AMOUNT FROM Payment WHERE AMOUNT >= 6000 GROUP
BY AMOUNT;
```

```
SELECT COUNT(PAY_ID),AMOUNT FROM Payment WHERE AMOUNT>=6000 GROUP BY AMOUNT;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.002 seconds

	COUNT(PAY_ID)	AMOUNT
1	23	6000

9) **If the Administrator wants to add column 'age' in the PAYMENT table.**

```
ALTER TABLE Payment add Age NUMBER(2);
```

Welcome Page | ORCLL | PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partition

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAY_ID	NUMBER(3,0)	No	(null)	1	(null)
2	MEM_ID	NUMBER(3,0)	Yes	(null)	2	(null)
3	CUST_ID	NUMBER(3,0)	Yes	(null)	3	(null)
4	AMOUNT	NUMBER(7,2)	Yes	(null)	4	(null)
5	TIME	DATE	Yes	(null)	5	(null)
6	DATE_OF_PAY	DATE	Yes	(null)	6	(null)
7	AGE	NUMBER(2,0)	Yes	(null)	7	(null)

10) **If the Administrator wants to Change column name from workout_time to workout_timing.**

```
ALTER TABLE workout_plan RENAME COLUMN workout_time TO workout_timing;
```

Welcome Page | ORCLL | WORKOUT_PLAN

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Inc

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PLAN_ID	NUMBER(3,0)	No	(null)	1	(null)
2	MEM_ID	NUMBER(3,0)	Yes	(null)	2	(null)
3	INS_ID	NUMBER(3,0)	Yes	(null)	3	(null)
4	WORKOUT_TIMING	DATE	Yes	(null)	4	(null)
5	WORKOUT_DURATION	NUMBER(3,1)	Yes	(null)	5	(null)

11) **If the Administrator wants see the number of members having the same workout duration and order it by plan_id.**

```
SELECT COUNT(PLAN_ID),workout_duration FROM workout_plan GROUP BY
workout_duration ORDER BY COUNT(PLAN_ID) DESC;
```

```
SELECT COUNT(PLAN_ID),workout_duration FROM workout_plan GROUP BY workout_duration ORDER BY COUNT(PLAN_ID) DESC
```

Script Output | Query Result

All Rows Fetched: 4 in 0.003 seconds

	COUNT(PLAN_ID)	WORKOUT_DURATION
1	6	1
2	4	0.6
3	2	2
4	2	1.5

12) JOINING:

```
SELECT M.Name, W.Ins_ID FROM MEMBER M
INNER JOIN WORKOUT_PLAN W ON W.Mem_ID = M.Mem_ID;
```

```
SELECT M.Name, W.Ins_ID FROM MEMBER M
INNER JOIN WORKOUT_PLAN W ON W.Mem_ID = M.Mem_ID;
```

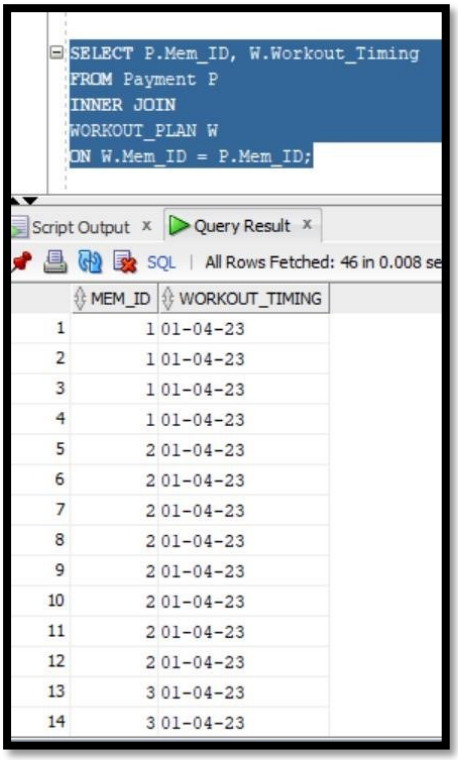
Script Output | Query Result

All Rows Fetched: 14 in 0.005 seconds

	NAME	INS_ID
1	Kavya Aggarwal	301
2	Kavya Aggarwal	302
3	Vikram Gupta	301
4	Vikram Gupta	302
5	Shikha Singh	301
6	Shikha Singh	302
7	Gaurav Kumar	301
8	Gaurav Kumar	302
9	Pooja Gupta	301
10	Pooja Gupta	302
11	Rohit Kumar	301
12	Rohit Kumar	302
13	Nikita Singh	301
14	Nikita Singh	302

```
SELECT P.Mem_ID, W.Workout_Timing
FROM Payment P  INNER
JOIN
WORKOUT_PLAN W
```

```
ON W.Mem_ID = P.Mem_ID;
```



```
SELECT P.Mem_ID, W.Workout_Timing
FROM Payment P
INNER JOIN
WORKOUT_PLAN W
ON W.Mem_ID = P.Mem_ID;
```

Script Output | Query Result

SQL | All Rows Fetched: 46 in 0.008 se

	MEM_ID	WORKOUT_TIMING
1	1	01-04-23
2	1	01-04-23
3	1	01-04-23
4	1	01-04-23
5	2	01-04-23
6	2	01-04-23
7	2	01-04-23
8	2	01-04-23
9	2	01-04-23
10	2	01-04-23
11	2	01-04-23
12	2	01-04-23
13	3	01-04-23
14	3	01-04-23