



Mini project report on

GYM MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE20CS301 – DBMS Project

Submitted by:

CHANDAN KUMAR S PES2UG20CS804

Under the guidance of

Prof. Nivedita Kasturi

Assistant Professor

Designation

PES University

AUG - DEC 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

GYM MANAGEMENT SYSTEM

is a bonafide work carried out by

CHANDAN KUMAR S

PES2UG20CS804

In partial fulfilment for the completion of fifth semester DBMS Project (UE20CSS301) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2022 – DEC. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Prof. Nivedita Kasturi

Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **GYM MANAGEMENT SYSTEM** has been carried out by us under the guidance of **Prof. Nivedita Kasturi, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG–DEC2022

CHANDAN KUMAR S

PES2UG20CS804

<Signature>

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nivedita Kasturi, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE20CS301 - DBMS Project.

I take this opportunity to thank Dr. Sandesh B J, C, Professor, ChairPerson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

This project “Gym Management System” is solution fitness centres to manage the customers in an easier and more convenient way. The administrator, is able to view all the members of fitness centre as well as their details. The basic structure of the system as follows. This project is a computer-based program and it manages the gym members, the personnel and the inventory. This system also maintains the client details, to provide the valuable reports regarding the progress of the gym member. The aim of the project is to provide the online access to manage the gym members and trainers and their workouts. It enables the admin to control all over the gym system with single integrated application to provide the customers

Introduction

We all know health is a wealth. We do not need a fancy car, big apartment, a doctor degree without a health. Being healthy is a first thing we need to keep in mind. Because most of time our attitude depends on how we feel. Being healthy and fit gives us energy to do anything. Physical fitness is very necessary for a healthy and tension free life. Physical fitness includes diet, exercise and sleep. These three basic things have their own importance in each individual's life and everyone should be sensible with regard to these for a healthy life.

PROBLEM STATEMENT: -

The main objective of the project is to develop software that facilitates the data storage, data maintenance and its retrieval for the gym in an igneous way. To store the record of the customers, the staff that has the privileges to access, modify and delete any record and finally the service gym provides to its customers. Also, only staff have the privilege to access any database and make the required changes, if necessary. To develop easy-to-use software which handles the customer-staff relationship in an effective manner. To develop a user-friendly system that requires minimal user training. Most of the features and functions are similar to those on any windows platform.

Overview Of the Project

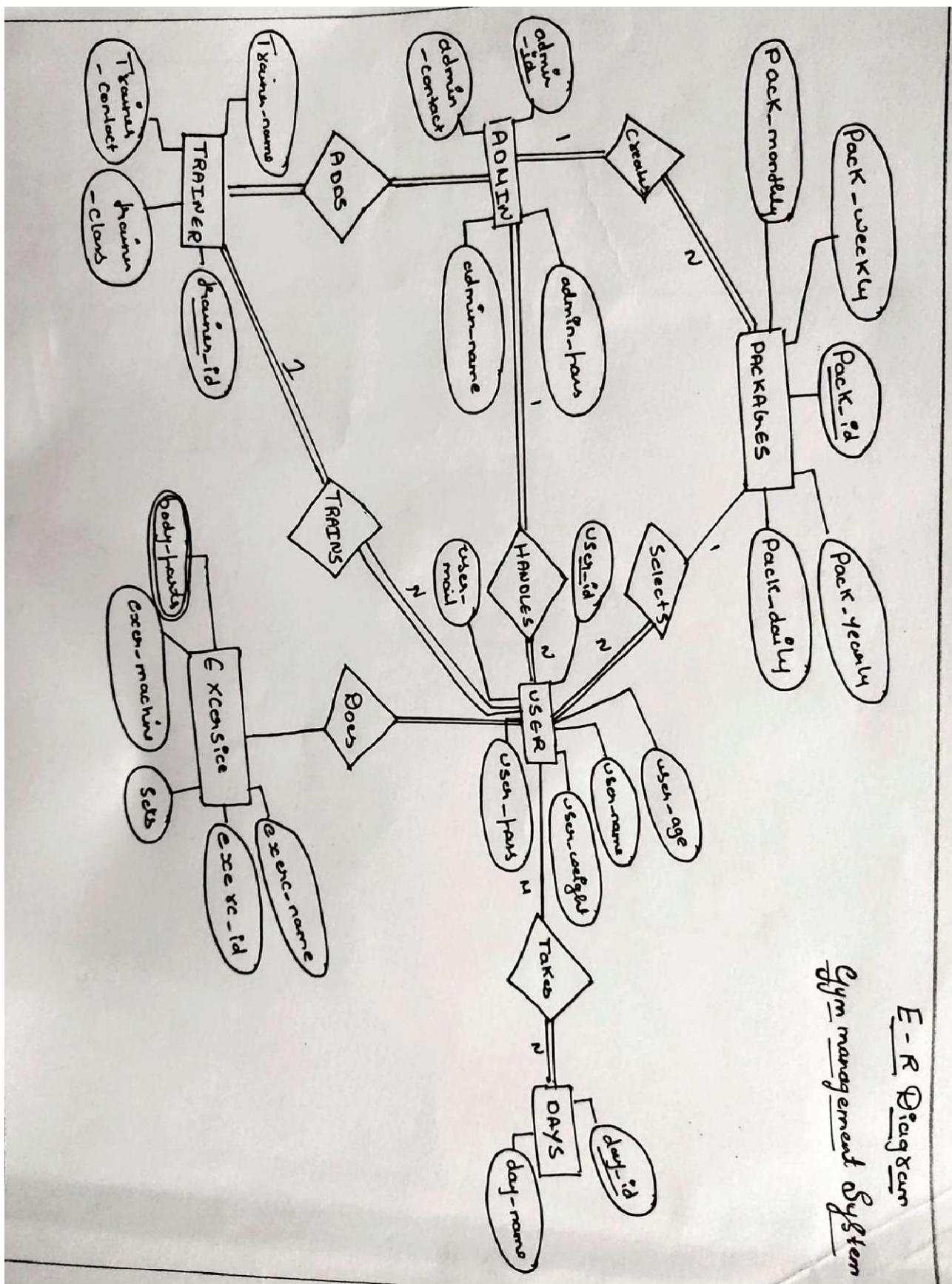
The “Gym management system” is for those who do a gym business. Before doing anything, we did decent research on major difficulties for gym owners. We examined carefully about how to make a huge registering system without failure as well as different function for different kind of user depending on their privilege.

Gym is the best way to get fitness and healthy life. There are more facility in gym that members get satisfied by there money and time. Gym management system holds the detailed information of the members and membership system. Gym management provides lots of functions such as registration of member, payment method, payment package, trainer details, tool requirement and so on. The admin of the gym will be responsibility of the whole gym management. There should be good communication between the members and gym master of the gym. This are helps to provide good quality of services to customer from gym admin.

This management system is become very simple to understand because of data flow diagram provided by us. Context level diagram and some chart are also available in this case study. The demo of using the software such as customer detail from, data base of software is also provided by us.

The gym management system aim is to provide good service for the customer, best facilities to satisfy the customer, provide total information about machine and data of coaches is also stored in it. Services provided by gym are also handled by the system. There is a good security for the details provided by the member in gym management system.

FINAL ER DIAGRAM: -



MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL

• Reasons for choosing relational database: -

1. A relational database is a type of database that stores and provides access to data points that are related to one another.
2. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables.
3. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.
4. The relational model means that the logical data structures—the data tables, views, and indexes—are separate from the physical storage structures. This separation means that database administrators can manage physical data storage without affecting access to that data as a logical structure.
5. To ensure that data is always accurate and accessible, relational databases follow certain integrity rules. For example, an integrity rule can specify that duplicate rows are not allowed in a table in order to eliminate the potential for erroneous information entering the database.

ER TO RELATIONAL MAPPING

STEPS OF ALGORITHM FOR CHOOSEN PROBLEM

Step1: Mapping of regular entity types

<u>User_id</u>	User_weight	User_contact	User_name	User_email	User_age	User_pass
----------------	-------------	--------------	-----------	------------	----------	-----------

Pack_id	Pack_weekly	Pack_monthly	Pack_yearly	Pack_daily
---------	-------------	--------------	-------------	------------

Trainer_id	Trainer_name	Trainer_class	Trainer_contact
------------	--------------	---------------	-----------------

Admin_id	Admin_name	Admin_contact	Admin_pass
----------	------------	---------------	------------

Excer_id	Excer_name	Excer_machine	sets
----------	------------	---------------	------

Days_id	Day_name
---------	----------

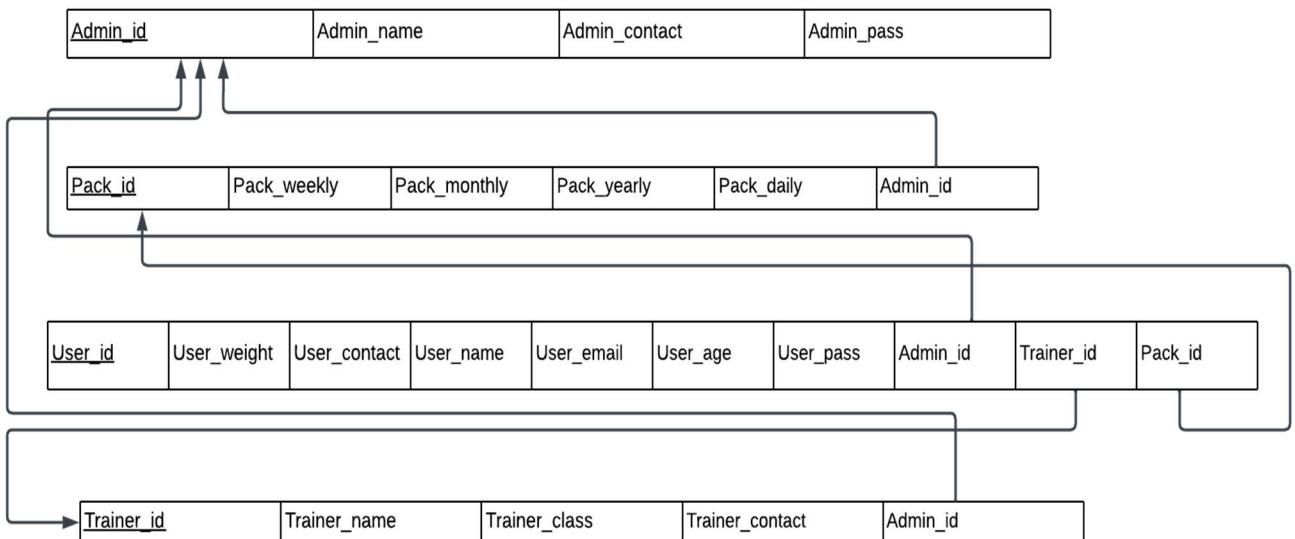
Step2: Mapping of weak entities

There are no weak entities

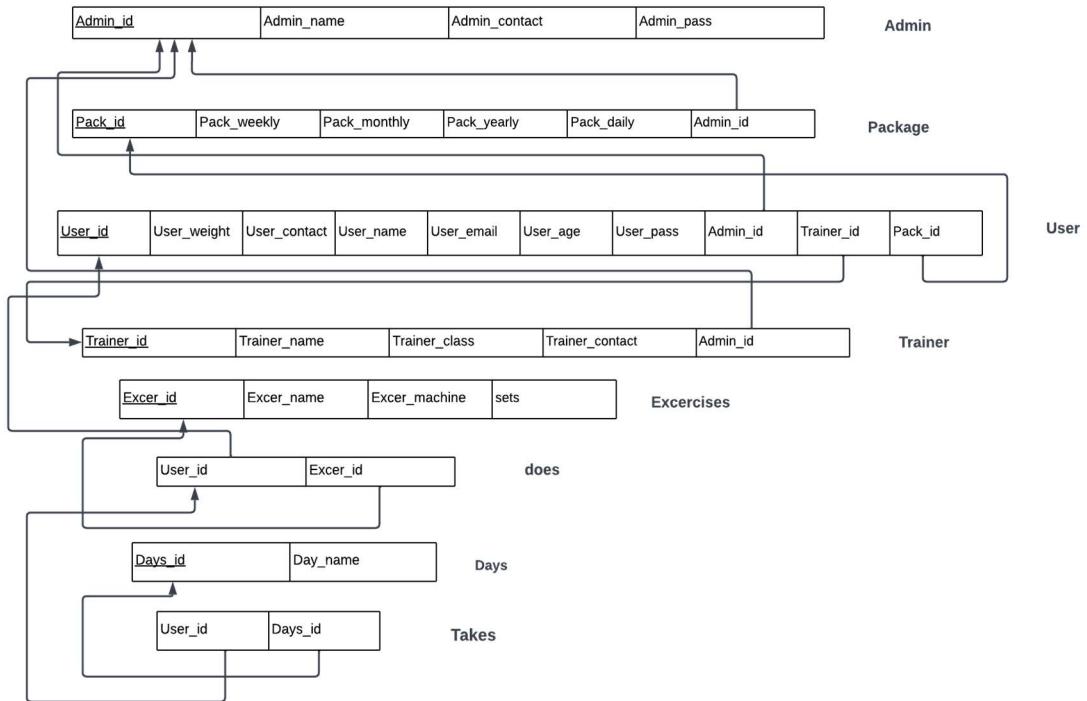
Step3: Mapping of Binary 1:1 Relation types

There are no 1:1 relation

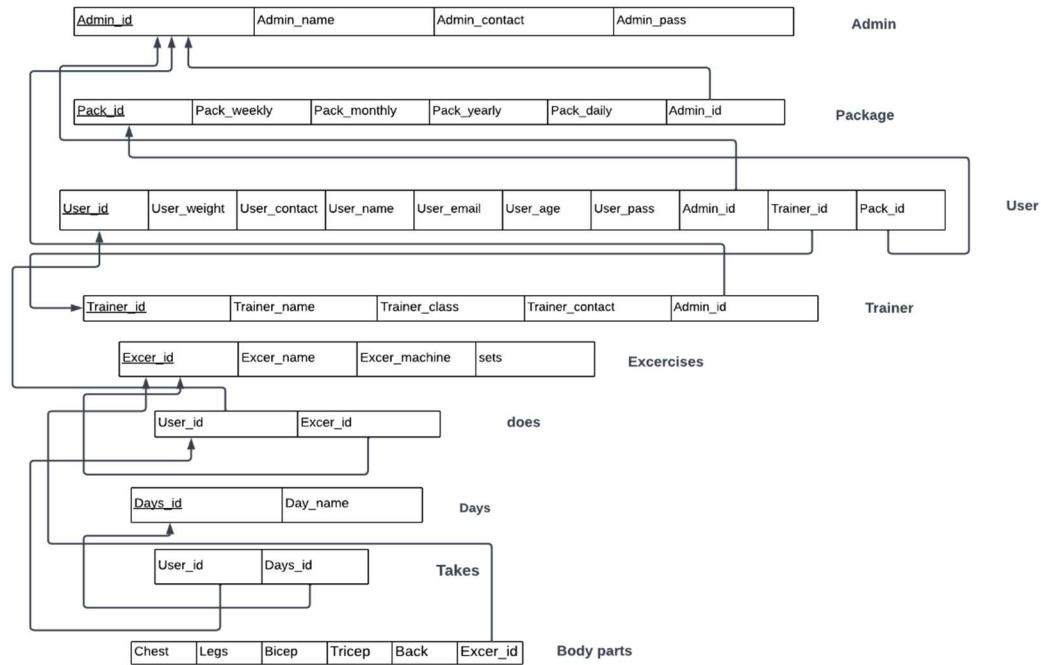
Step4: Mapping of 1: N Relationship Types



Step5: Mapping of M: N Relationship Types



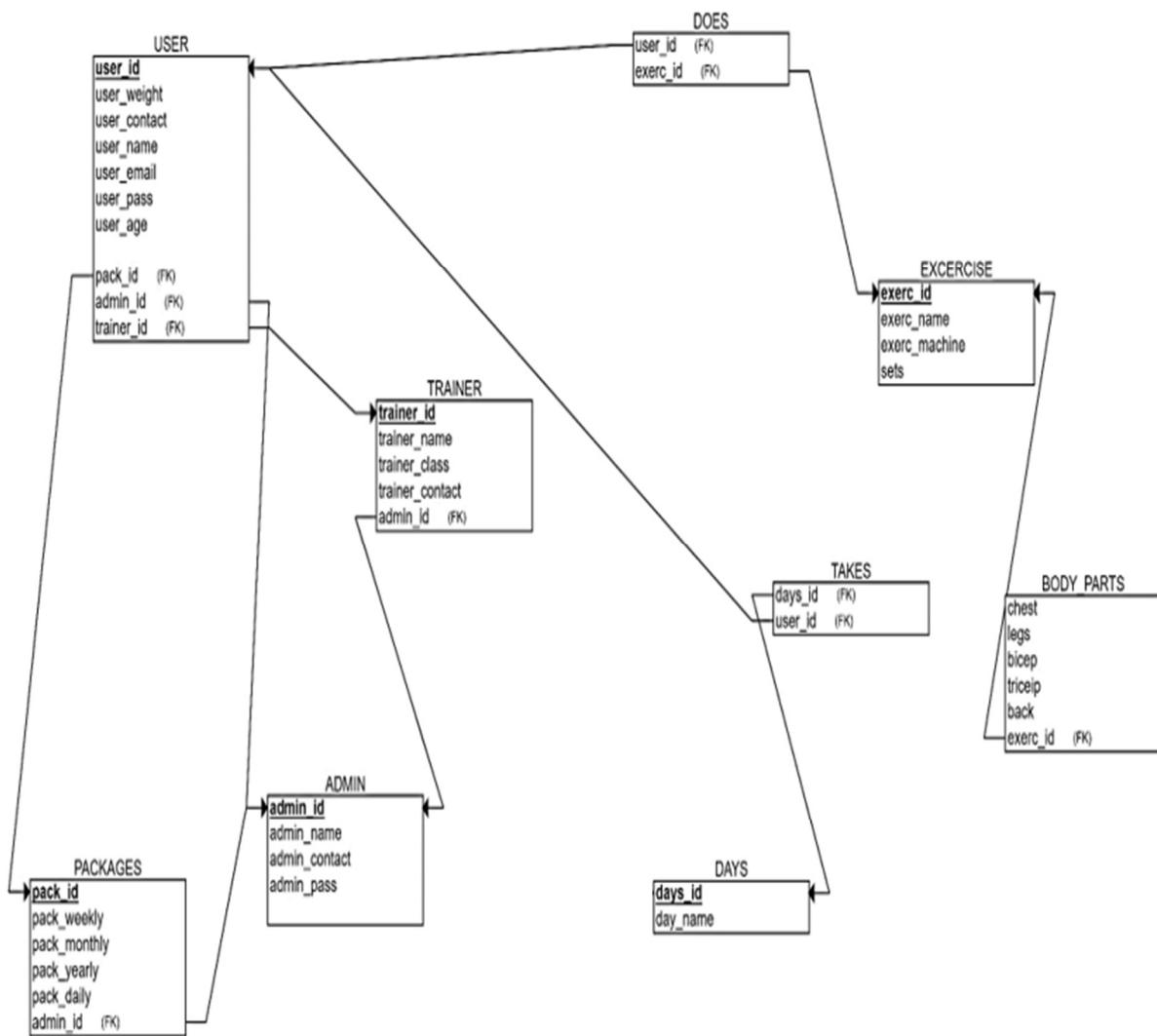
Step6: Mapping of Multivalued attribute



Step7: Mapping of N-ary relationship

There are no N-ary relationship types

COMPLETE RELATIONAL SCHEMA DIAGRAM: -



DDL STATEMENTS

Admin table

```
create table admins
(
    admin_id int ,
    admin_name varchar(40),
    admin_contact varchar(40),
    admin_pass varchar(40),

    PRIMARY KEY (admin_id)

) ;
```

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected for the 'admins' table. The table has four columns: admin_id, admin_name, admin_contact, and admin_pass. The admin_id column is defined as an int(11) with a primary key constraint. The other three columns are defined as varchar(40) with utf8mb4_general_ci collation. The 'Attributes' column includes information like 'Null' (No), 'Default' (None or NULL), and 'Comments'. The 'Action' column provides options to change, drop, or manage the column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	admin_id	int(11)			No	None			Change Drop More
2	admin_name	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	admin_contact	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	admin_pass	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Package table

```
create table packages
(
    pack_id int ,
    pack_weekly varchar(40) ,
    pack_monthly varchar(40) ,
    pack_daily varchar(40) ,
    pack_yearly varchar(40) ,
    admin_id int ,

    PRIMARY KEY (pack_id) ,
    foreign KEY (admin_id) references admins(admin_id)
);
```

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected for the 'packages' table. The table has six columns: pack_id, pack_weekly, pack_monthly, pack_daily, pack_yearly, and admin_id. The pack_id column is defined as an int(11) with a primary key constraint. The other five columns are defined as varchar(40) with utf8mb4_general_ci collation. The 'Attributes' column includes information like 'Null' (No or Yes), 'Default' (None or NULL), and 'Comments'. The 'Action' column provides options to change, drop, or manage the column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	pack_id	int(11)			No	None			Change Drop More
2	pack_weekly	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	pack_monthly	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	pack_daily	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	pack_yearly	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	admin_id	int(11)			Yes	NULL			Change Drop More

Trainer Table

```
create table trainer
(
    trainer_id int,
    trainer_name varchar(20),
    trainer_class varchar(20),
    trainer_contact varchar(20),
    admin_id int ,

    PRIMARY KEY (trainer_id) ,
    foreign KEY (admin_id) references admins(admin_id)

) ;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	trainer_id	int(11)			No	None			Drop
2	trainer_name	varchar(20)	utf8mb4_general_ci		Yes	NULL			Drop
3	trainer_class	varchar(20)	utf8mb4_general_ci		Yes	NULL			Drop
4	trainer_contact	varchar(20)	utf8mb4_general_ci		Yes	NULL			Drop
5	admin_id	int(11)			Yes	NULL			Drop

Exercise table

```
create table excercise
(
    excer_id  varchar(10) ,
    excer_name varchar(20) ,
    excer_machine varchar(20) ,
    sets int ,

    PRIMARY KEY (excer_id)

) ;
```

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	excer_id	varchar(10)	utf8mb4_general_ci		No	None			Drop
2	excer_name	varchar(20)	utf8mb4_general_ci		Yes	NULL			Drop
3	excer_machine	varchar(20)	utf8mb4_general_ci		Yes	NULL			Drop
4	sets	int(11)			Yes	NULL			Drop

Check all With selected: Drop Primary Unique Index Fulltext

Users table

```
create table users
(
    users_id int ,
    user_weight int ,
    users_name varchar(40) ,
    user_email varchar(40) ,
    user_pass varchar(40) ,
    user_age int ,
    user_contact bigint ,
    pack_id int ,
    admin_id int ,
    trainer_id int ,

    PRIMARY KEY (users_id),
    foreign KEY (pack_id) references packages(pack_id),
    foreign KEY (admin_id) references admins(admin_id),
    foreign KEY (trainer_id) references trainer(trainer_id)

);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 users_id	int(11)			No	None		Change Drop More	
<input type="checkbox"/>	2 user_weight	int(11)			Yes	NULL		Change Drop More	
<input type="checkbox"/>	3 users_name	varchar(40)	utf8mb4_general_ci		Yes	NULL		Change Drop More	
<input type="checkbox"/>	4 user_email	varchar(40)	utf8mb4_general_ci		Yes	NULL		Change Drop More	
<input type="checkbox"/>	5 user_pass	varchar(40)	utf8mb4_general_ci		Yes	NULL		Change Drop More	
<input type="checkbox"/>	6 user_age	int(11)			Yes	NULL		Change Drop More	
<input type="checkbox"/>	7 user_contact	bigint(20)			Yes	NULL		Change Drop More	
<input type="checkbox"/>	8 pack_id	int(11)			Yes	NULL		Change Drop More	
<input type="checkbox"/>	9 admin_id	int(11)			Yes	NULL		Change Drop More	
<input type="checkbox"/>	10 trainer_id	int(11)			Yes	NULL		Change Drop More	

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Table day

```
create table dayy
(
    days_id int ,
    day_name varchar(20) ,

    PRIMARY KEY (days_id)

);
```

Browse Structure SQL Search Insert Export Import Privileges Operations									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 days_id	int(11)			No	None		Change Drop More	
<input type="checkbox"/>	2 day_name	varchar(20)	utf8mb4_general_ci		Yes	NULL		Change Drop More	

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Table body parts

```
create table bodyparts
(
    chest varchar(40) ,
    legs  varchar(40) ,
    tricep varchar(40) ,
    bicep  varchar(40) ,
    back   varchar(40) ,
    excerpt_id varchar(40) ,

    foreign KEY (excerpt_id) references excercise(excerpt_id)

) ;
```

The screenshot shows the 'bodyparts' table structure in MySQL Workbench. At the top, there are two tabs: 'Table structure' (selected) and 'Relation view'. Below is a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The table has 6 rows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	chest	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
2	legs	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	tricep	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	bicep	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	back	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	excerpt_id	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Table Does

```
create table does
(
    users_id int ,
    excerpt_id varchar(10) ,

    foreign KEY (users_id) references users(users_id),
    foreign KEY (excerpt_id) references excercise(excerpt_id)

) ;
```

The screenshot shows the 'does' table structure in MySQL Workbench. At the top, there are two tabs: 'Table structure' (selected) and 'Relation view'. Below is a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The table has 2 rows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	days_id	int(11)			Yes	NULL			Change Drop More
2	users_id	int(11)			Yes	NULL			Change Drop More

At the bottom, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'.

Table takes

```
create table takes
(
    days_id int ,
    users_id int ,

    foreign KEY (users_id) references users(users_id),
    foreign KEY (days_id) references dayy(days_id)

) ;
```

The screenshot shows the 'takes' table structure in MySQL Workbench. At the top, there are two tabs: 'Table structure' (selected) and 'Relation view'. Below is a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. The table has 2 rows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	days_id	int(11)			Yes	NULL			Change Drop More
2	users_id	int(11)			Yes	NULL			Change Drop More

DML STATEMENTS

Admin values

```
insert into admins values (111,'Prabhas','9873528175','ABC');
insert into admins values (222,'Mahesh','8538268263','EFG');
insert into admins values (333,'Ayagaru','6489203642','JDH');
insert into admins values (444,'Kondanna','8459538434','HEH');
insert into admins values (555,'King','6474893567','JEI');
insert into admins values (666,'Nag','7452639644','USH');
insert into admins values (777,'Akhil','7564839353','DKM');
insert into admins values (888,'Chiru','9363537363','SJS');
```

Package values

```
insert into packages values (001,'once','10days','2hrs','300days',111);
insert into packages values (002,'twice','15days','3hrs','250days',222);
insert into packages values (003,'thrice','20days','4hrs','310days',333);
insert into packages values (004,'fourtimes','25days','4.5hrs','290days',444);
insert into packages values (005,'fivetimes','30days','5hrs','340days',555);
insert into packages values (006,'once','23days','2hrs','265days',666);
insert into packages values (007,'threetimes','16days','3hrs','150days',777);
insert into packages values (008,'fivetimes','28days','2.5hrs','280days',888);
```

Trainer values

```
insert into trainer values (1234,'varma','beginner','9374825362',111);
insert into trainer values (1425,'varun','intermediate','9465734261',222);
insert into trainer values (5456,'ram','advanced','6479374949',333);
insert into trainer values (5676,'gopal','beginner','7564849392',444);
insert into trainer values (7563,'hooda','intermediate','9464826453',555);
insert into trainer values (6353,'shyam','intermediate','9464826453',666);
insert into trainer values (8338,'manish','intermediate','9464826453',777);
insert into trainer values (3838,'karan','intermediate','9464826453',888);
```

exercise values

```
insert into excercise values ('10a','chestpress','dumbbellpressdown',5);
insert into excercise values ('10b','thighs','legpress',4);
insert into excercise values ('10c','bicepexercise','cablecurl',3);
insert into excercise values ('10d','tricep pressdpwn','close grip benchdown',2);
insert into excercise values ('10e','backpull','lowseatedrow',3);
insert into excercise values ('10f','pushups','null',5);
insert into excercise values ('10g','dumbbell rows','dumbles',3);
insert into excercise values ('10h','side planks','null',4);
```

User values

```
insert into users values (1,56,'siddardha','siddardha@gmail.com','sidd',19,6453635363,001,111,1234);
insert into users values (2,55,'ajay','ajay@gmail.com','aj',20,9573712473,002,222,1425);
insert into users values (3,78,'tharun','tharun@gmail.com','tha',20,7573582946,003,333,5456);
insert into users values (4,99,'vishnu','vishnu@gmail.com','vi',25,9874563434,004,444,5676);
insert into users values (5,99,'vikranth','vikranth@gmail.com','vikku',24,9756473546,005,555,7563);
insert into users values (6,58,'arun','arun@gmail.com','aru',20,9647546464,006,666,6353);
insert into users values (7,60,'sasank','sasank@gmail.com','sas',21,9474647474,007,777,8338);
insert into users values (8,78,'pradeep','pradeep@gmail.com','vibh',22,82653637373,008,888,3838);
```

day values

```
insert into dayy values (001,'monday');
insert into dayy values (010,'tuesday');
insert into dayy values (011,'wednesday');
insert into dayy values (100,'thrusday');
insert into dayy values (110,'friday');
```

Body parts values

```
insert into bodyparts values ('barbell bench press','squats','benchdip','hammer curl','bridges','10a');
insert into bodyparts values ('dumbbell bench press','lunges','board press','incline dumbbellcurl','pelvic tilts','10b');
insert into bodyparts values ('incline bench press','plank leglifts','dumbbell lowerheadextension','preacher','streches','10c');
insert into bodyparts values ('decline bench press','steps-ups','triceip dip','seated alteringcurl','cat streches','10d');
insert into bodyparts values ('dip','box jumps','cabeloverheadextension','standing cablecurl','lower backtilts','10e');
insert into bodyparts values ('chest fly','single leg deadlift','diamond pushups','EZ bar curl','lying lateral leglift','10f');
insert into bodyparts values ('dumbbell pull-over','speed skater jumps','close grip benchpress','spider curl','supermans','10g');
insert into bodyparts values ('machine chest press','bridge','cable ropetricep pushdown','seated zottmal curl','partial curls','10h');
```

Does values

```
insert into does values ( 1 , '10a' );
insert into does values ( 2 , '10b' );
insert into does values ( 3 , '10c' );
insert into does values ( 4 , '10d' );
insert into does values ( 5 , '10e' );
insert into does values ( 6 , '10f' );
insert into does values ( 7 , '10g' );
insert into does values ( 8 , '10h' );
```

Working model of the Database application

SQL QUERIES

1. Select the user's details who are working with exercise "chest press"

```
SELECT * FROM users where users_id IN(SELECT users_id FROM excercise where exer_name = 'chestpress');
```

Showing rows 0 - 8 (9 total, Query took 0.0013 seconds.)

```
SELECT * FROM users where users_id IN(SELECT users_id FROM excercise where exer_name = 'chestpress')
```

Profiling [Edit inline] [Edit]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	users_id	user_weight	users_name	user_email	user_pass	user_age	user_contact	pack_id	admin_id	trainer_id
<input type="checkbox"/>	1	56	siddu	sidda@gmail.com	sidd	19	6453635363	1	111	1234
<input type="checkbox"/>	2	55	chandan	ck@gmail.com	chan	20	9573712473	2	222	1425
<input type="checkbox"/>	3	78	tharun	tharun@gmamail.com	tharu	20	7573582906	3	333	5456
<input type="checkbox"/>	4	99	vish	vishnu@gmail.com	vish	25	9874563434	4	444	5676
<input type="checkbox"/>	5	99	vikranth	vikranth@gmail.com	vikku	24	9756473505	5	555	7563
<input type="checkbox"/>	6	58	arun	arungmail.com	aru	20	6475464646	6	666	6353
<input type="checkbox"/>	7	60	sasank	sasank@gmail.com	sas	21	9474647474	7	777	8338
<input type="checkbox"/>	8	28	pradeep	pradeepagmail.com	vibh	22	8265163773	8	888	61038
<input type="checkbox"/>	9	45	digant	duggu@gmail.com	duggu	15	6459453363	2	111	5456

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

2. Names of the trainer whose trainer class is intermediate

```
select trainer_name from trainer where trainer_class ='intermediate';
```

Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

```
select trainer_name from trainer where trainer_class ='intermediate'
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	trainer_name
<input type="checkbox"/>	varun
<input type="checkbox"/>	karan
<input type="checkbox"/>	shyam
<input type="checkbox"/>	hooda
<input type="checkbox"/>	manish
<input type="checkbox"/>	dharma

Check all With selected: Edit Copy Delete Copy Delete Export

3. Names of users and their contact numbers whose age is greater than 20

select users_name,user-contact from users where user_age>='20';

Show query box

✓ Showing rows 0 - 6 (7 total, Query took 0.0038 seconds.)

```
select users_name,user_contact from users where user_age>='20'
```

Profiling [E]

Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

	users_name	user_contact
<input type="checkbox"/>	chandan	9573712473
<input type="checkbox"/>	tarun	7573582906
<input type="checkbox"/>	vish	9874563434
<input type="checkbox"/>	vikranth	9756473505
<input type="checkbox"/>	arun	6475464644
<input type="checkbox"/>	sasank	9474647474
<input type="checkbox"/>	pradeep	8265163773

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

Query results operations

4. Names of Exercise and machine whose sets are equal to 5

select excerpt_name, excerpt_id,sets from excercise where sets='5';

Show query box

✓ Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
select excerpt_name, excerpt_id,sets from excercise where sets='5'
```

Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

	excerpt_name	excerpt_id	sets
<input type="checkbox"/>	chestpress	10a	5
<input type="checkbox"/>	pushups	10f	5

Check all With selected: Edit Copy Delete Export

5. Select userid , username who are trained by trainer' 5456'

```
select users_id, users_name, user_weight FROM users WHERE
trainer_id='5456'
```

Show query box

The screenshot shows the MySQL Workbench interface with the following details:

- Query Box:** Shows the executed query: `select users_id,users_name,user_weight FROM users WHERE trainer_id='5456'` and its result: "Showing rows 0 - 1 (2 total, Query took 0.0075 seconds.)".
- Table View:** A table with columns `users_id`, `users_name`, and `user_weight`. It contains two rows:

users_id	users_name	user_weight
3	tharun	78
9	digant	45
- Toolbar:** Includes buttons for Show all, Number of rows (set to 25), Filter rows, Search this table, Sort by key (set to None), and various edit and delete options.
- Operations Bar:** Includes buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

6. Names of Users who are trained by the trainer named ram.

```
select users_name from users where trainer_id in (select trainer_id from
trainer where trainer_name='ram');
```

Show query box

The screenshot shows the MySQL Workbench interface with the following details:

- Query Box:** Shows the executed query: `select users_name from users where trainer_id in (select trainer_id from trainer where trainer_name='ram')` and its result: "Showing rows 0 - 1 (2 total, Query took 0.0050 seconds.)".
- Table View:** A table with column `users_name`. It contains one row:

users_name
tharun
- Toolbar:** Includes buttons for Show all, Number of rows (set to 25), Filter rows, Search this table, Sort by key (set to None), and various edit and delete options.
- Operations Bar:** Includes buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

7. Names of the users who are performing exercise barbell bench press.

select users_name from users where users_id in (select users_id from does natural join bodyparts where chest='barbell bench press');

Show query box

The screenshot shows the MySQL Workbench interface with the following details:

- Query Results:** Shows the result of the query: "Showing rows 0 - 0 (1 total, Query took 0.0041 seconds.)". The query itself is: "select users_name from users where users_id in (select users_id from does natural join bodyparts where chest='barbell bench press')".
- Table View:** A table named "users_name" is displayed with one row: "siddu".
- Operations:** Buttons for Print, Copy to clipboard, Export, Display chart, and Create view are available.

8. Names of the users ,their email ids and contact numbers who workout on monday

select users_name,user_email,user_contact from users where users_id in (select users_id from takes t join dayy d on t.days_id=d.days_id where day_name='monday');

The screenshot shows the MySQL Workbench interface with the following details:

- Query Results:** Shows the result of the query: "Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)". The query is: "select users_name,user_email,user_contact from users where users_id in (select users_id from takes t join dayy d on t.days_id=d.days_id where day_name='monday')".
- Table View:** A table with columns "users_name", "user_email", and "user_contact" is displayed, showing two rows: "siddu" with email "sidda@gmail.com" and contact "6453635363", and "pradeep" with email "pradeepagmail.com" and contact "8265163773".
- Operations:** Buttons for Print, Copy to clipboard, Export, Display chart, and Create view are available.

9. select trainer_name,trainer_id from trainer natural join admins where admin_name ='kanul'

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view of databases and tables. The 'Tables' section includes 'admins', 'bodyparts', 'dayy', 'does', 'excercise', 'logs', 'packages', 'takes', 'trainer', and 'users'. Under 'mygym', there are 'New', 'admin', 'days', and 'exercises'. The main panel displays the results of a query:

```
Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)  
select trainer_name,trainer_id from trainer natural join admins where admin_name ='kanul'  
  
+ Options  
trainer_name | trainer_id  
varma | 1234  
dharma | 9876
```

Below the results, there are buttons for 'Query results operations' including Print, Copy to clipboard, Export, Display chart, and Create view.

10. Names of the users who uses dumbbells and perform more than or equal to 3 sets

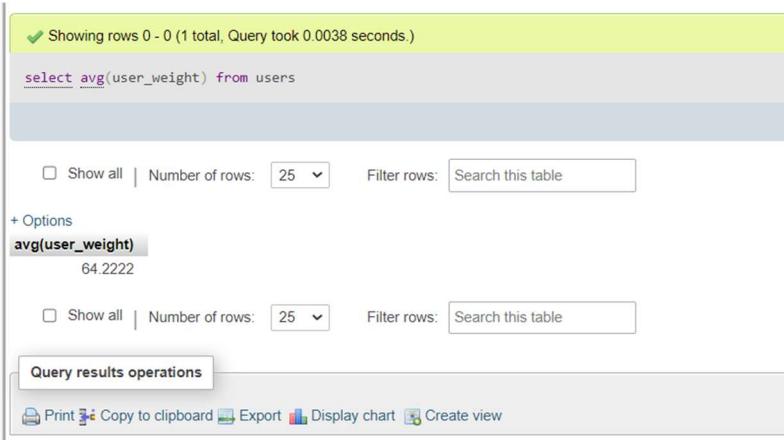
```
select users_name from users where users_id in (select users_id from excercise natural join does where exer_machine='dumbles' and sets>=3);
```

The screenshot shows the MySQL Workbench interface. The tree view on the left shows 'Tables' with 'users' selected. The main panel displays the results of the query:

```
Show query box  
Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)  
select users_name from users where users_id in (select users_id from excercise natural join does where exer_machine='dumbles' and sets>=3)  
  
+ Options  
users_name  
sasank  
Edit | Copy | Delete  
Check all | With selected: Edit | Copy | Delete | Export  
  
Query results operations  
Print | Copy to clipboard | Export | Display chart | Create view
```

12. The average weight users from the users table

select avg(user_weight) from users;



```
Showing rows 0 - 0 (1 total, Query took 0.0038 seconds.)  
select avg(user_weight) from users  
  
avg(user_weight)  
64.2222
```

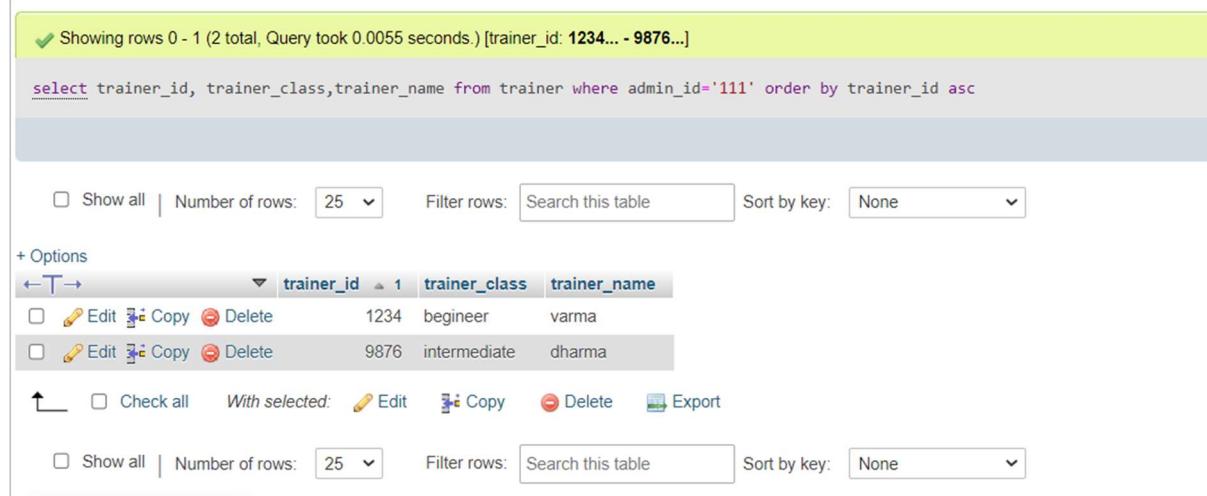
Show all | Number of rows: 25 Filter rows: Search this table

+ Options

Query results operations

Print Copy to clipboard Export Display chart Create view

13. Query to print trainer id , trainer class, trainer name of trainer who are managed by admin='111'



```
Showing rows 0 - 1 (2 total, Query took 0.0055 seconds.) [trainer_id: 1234... - 9876...]  
select trainer_id, trainer_class, trainer_name from trainer where admin_id='111' order by trainer_id asc  
  
trainer_id 1234 beginner varma  
9876 intermediate dharma
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

Edit Copy Delete

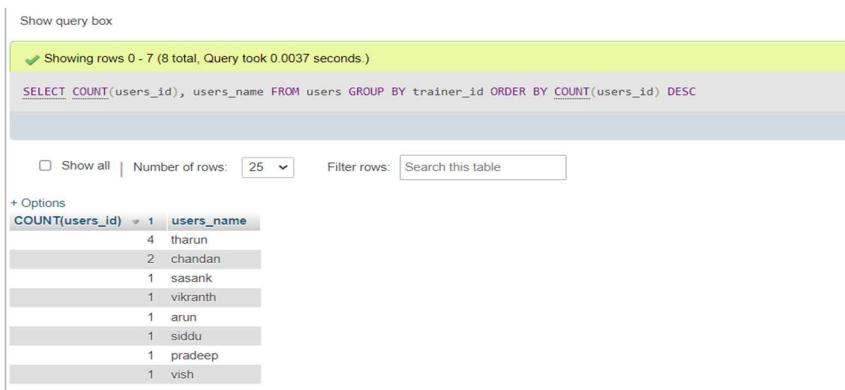
Edit Copy Delete

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

14. Query results with lists the number of customers trained by each trainer in descending order

SELECT COUNT(users_id), users_name FROM users GROUP BY trainer_id ORDER BY COUNT(users_id) DESC



```
Show query box  
Showing rows 0 - 7 (8 total, Query took 0.0037 seconds.)  
SELECT COUNT(users_id), users_name FROM users GROUP BY trainer_id ORDER BY COUNT(users_id) DESC  
  
COUNT(users_id) 1 users_name  
4 tharun  
2 chandan  
1 sasank  
1 vikranth  
1 arun  
1 siddu  
1 pradeep  
1 vish
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

11. Names of chest exercise which contain machine name as dumbbells or the number sets greater than or equal to 5

```
select    chest    from    bodyparts    natural    join    excercise    where  
excer_machine='dumbles' or sets>=5;
```

The screenshot shows a MySQL query results page. The query is:

```
select chest from bodyparts natural join excercise where excer_machine='dumbles' or sets>=5
```

The results table has one row:

chest
barbell bench press

Below the table are buttons for 'Query results operations': Print, Copy to clipboard, Export, Display chart, Create view.

13. The exercises performed by particular user siddardha on a given day

```
select chest,legs,triceps,biceps,back from bodyparts where excer_id in (select  
excer_id from does natural join users where users_name='chandan');
```

The screenshot shows a MySQL query results page. The query is:

```
select chest,legs,triceps,biceps,back from bodyparts where excer_id in (select excer_id from does natural join users where users_name='chandan')
```

The results table has two rows:

chest	legs	triceps	biceps	back
dumbbell bench press	lunges	board press	incline dumbbellcurl	pelvic tilts
dumbbell bench press	lunges	board press	incline dumbbellcurl	pelvic tilts

Below the table are buttons for 'Query results operations': Print, Copy to clipboard, Export, Display chart, Create view.

14. SELECT * FROM users where users_id IN(SELECT users_id FROM excercise where excer_name = 'chestpress');

```
SELECT admin_id AS user_id FROM trainer WHERE trainer_class = 'intermediate' UNION SELECT trainer_id FROM trainer where trainer_id = '1425';
```

The screenshot shows the phpMyAdmin interface with the database 'gms' selected. In the left sidebar, under the 'Tables' section, the 'users' table is highlighted. The main area displays the results of the executed query:

```
SELECT admin_id AS user_id FROM trainer WHERE trainer_class = 'intermediate' UNION SELECT trainer_id FROM trainer where trainer_id = '1425'
```

The results table shows the following data:

user_id
222
888
666
555
777
111
1425

STORED PROCEDURE:

15.. DELIMITER \$\$

```
CREATE procedure get_adminsinfo (IN var2 INT)
```

```
BEGIN
```

```
SELECT *FROM admins LIMIT var2;
```

```
SELECT COUNT(admin_id)AS TOTAL_ADMIN FROM admins;
```

```
END$$
```

```
DELIMITER $$
```

```
CALL getadminsinfo(4)
```

The screenshot shows the 'Routine parameters' dialog box in phpMyAdmin. It contains a table with the following data:

Name	Type	Function	Value
var2	INT		4

At the bottom right of the dialog are 'Go' and 'Close' buttons.

```

SET @p0='4'; CALL `get_adminsinfo`(@p0);

Execution results of routine `get_adminsinfo`:

+-----+-----+-----+-----+
| admin_id | admin_name | admin_contact | admin_pass |
+-----+-----+-----+-----+
| 111 | Kanul | 9873528175 | BXA |
| 222 | Mahesha | 8538268263 | BFG |
| 333 | charan | 6489283642 | JDH |
| 444 | anikha | 8660901237 | JDHPW |
+-----+-----+-----+-----+

TOTAL_ADMINs
8

```

Routines

Name	Action	Type	Returns
get_adminsinfo	Edit Execute Export Drop	PROCEDURE	

New Add routine

TRIGGERS: NEW UPDATES IN ADMIN TABLE OR DELETION LEADS TO INSERTION OF THEIR DETAILS ON LOGS TABLE

16. .DROP TRIGGER IF EXISTS `newadmin`;CREATE DEFINER=`root`@`localhost` TRIGGER `newadmin` AFTER UPDATE ON `admins` FOR EACH ROW INSERT INTO logs VALUES(NULL,NEW.admin_id,NEW.admin_contact,

NOW())

SELECT * FROM `logs`

sl_no	id	number	date
1	111	2147483647	2022-11-11 12:46:51
2	222	2147483647	2022-11-11 13:28:05
3	777	2147483647	2022-11-15 09:11:40
4	888	2147483647	2022-11-15 09:11:50

New amacclone doctoralm ecommerce ekart fitdb gms Procedures Tables New admins bodyparts dayy does excercise logs packages takes trainer users

Query results operations

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Triggers

Name	Action	Time	Event
newadmin	Edit Export Drop	AFTER	UPDATE

New Add trigger

Higher Level Programming

LANGUAGES USED FOR FRONT END :

- HTML
- CSS

LANGUAGES USED FOR DATABASE CONNECTIVITY:

- PHP

DEPENDENCIES INSTALLED:

XAMPP SERVER:

- XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.
- It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

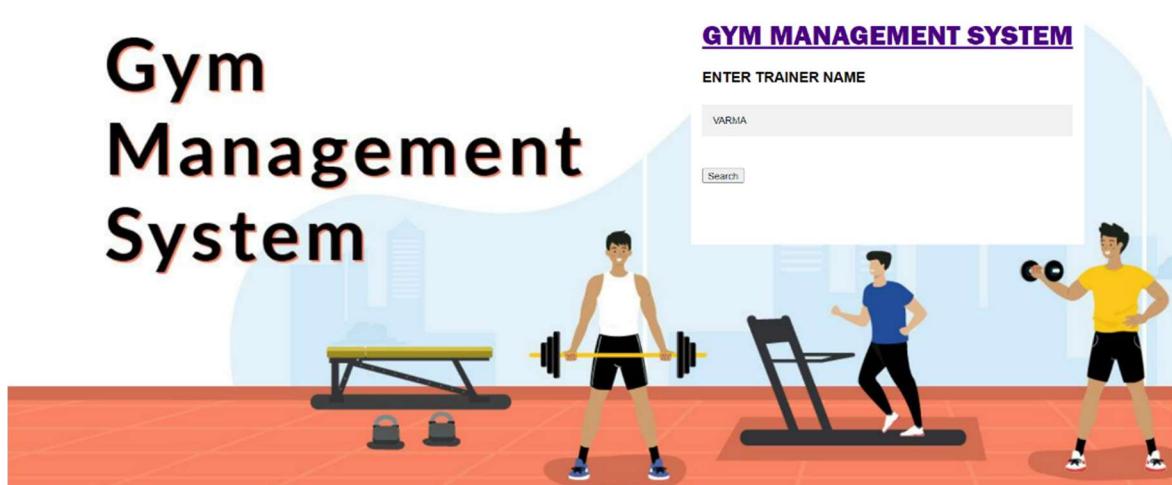
Query

List the customers name who are trained by particular trainer

```
select users_name,user_age,user_contact from users where trainer_id in  
(select trainer_id from trainer where trainer_name='ram')
```

Home Page

Gym Management System



Customer Details

THANK YOU FOR ENTERING

Here is the information displaying list of trainers:

GYM MANAGEMENT SYSTEM

USER NAME	USER AGE	USER CONTACT
tharun	20	7573582906
digant	15	6459453363
virat	19	785930221
hemanth	20	994447811