Fitness Data Hub 💪

A DBMS Project Presentation

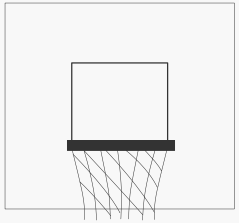
Table of contents

Members!

Siva Nandu S

Vishnu V P

Sidharth S



01

Introduction

This PPT presents the DBMS group project titled "Fitness Data Hub," submitted by Siva Nandu S, Vishnu V P, and Sidharth S. The project revolves around the management and utilization of data in a gym environment.

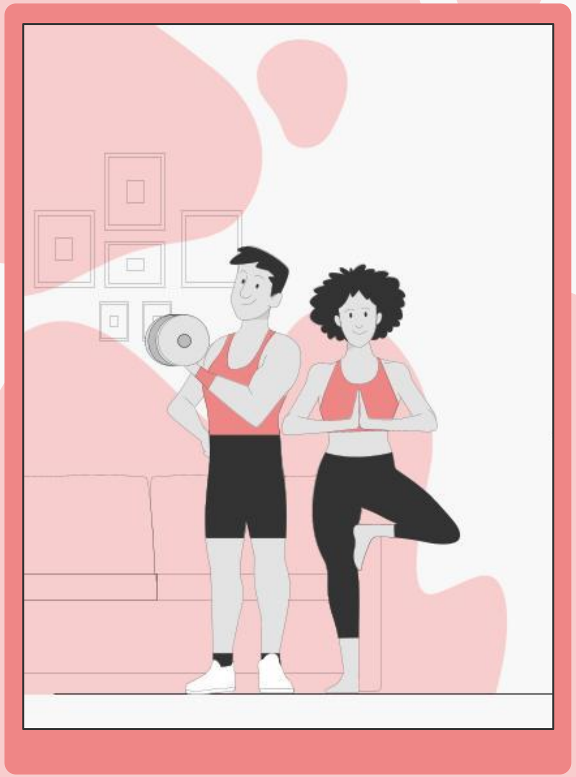
What is our Objective?

The objective of the project is to develop a comprehensive database management system (DBMS) that effectively handles the various aspects of gym data, including member profiles, workout routines, equipment inventory, classes, and fitness goals. The Fitness Data Hub aims to provide a centralized platform for efficient data organization, analysis, and reporting, empowering gym administrators, trainers, and members with valuable insights.

Different Relations in Database

Tables/Relations

- Gym
- Member
- Trainer
- Equipment
- Membership Plan
- Using Equipment
- Gives Supplement
- Log Book
- Competition



02 Design

You can divide the design into :

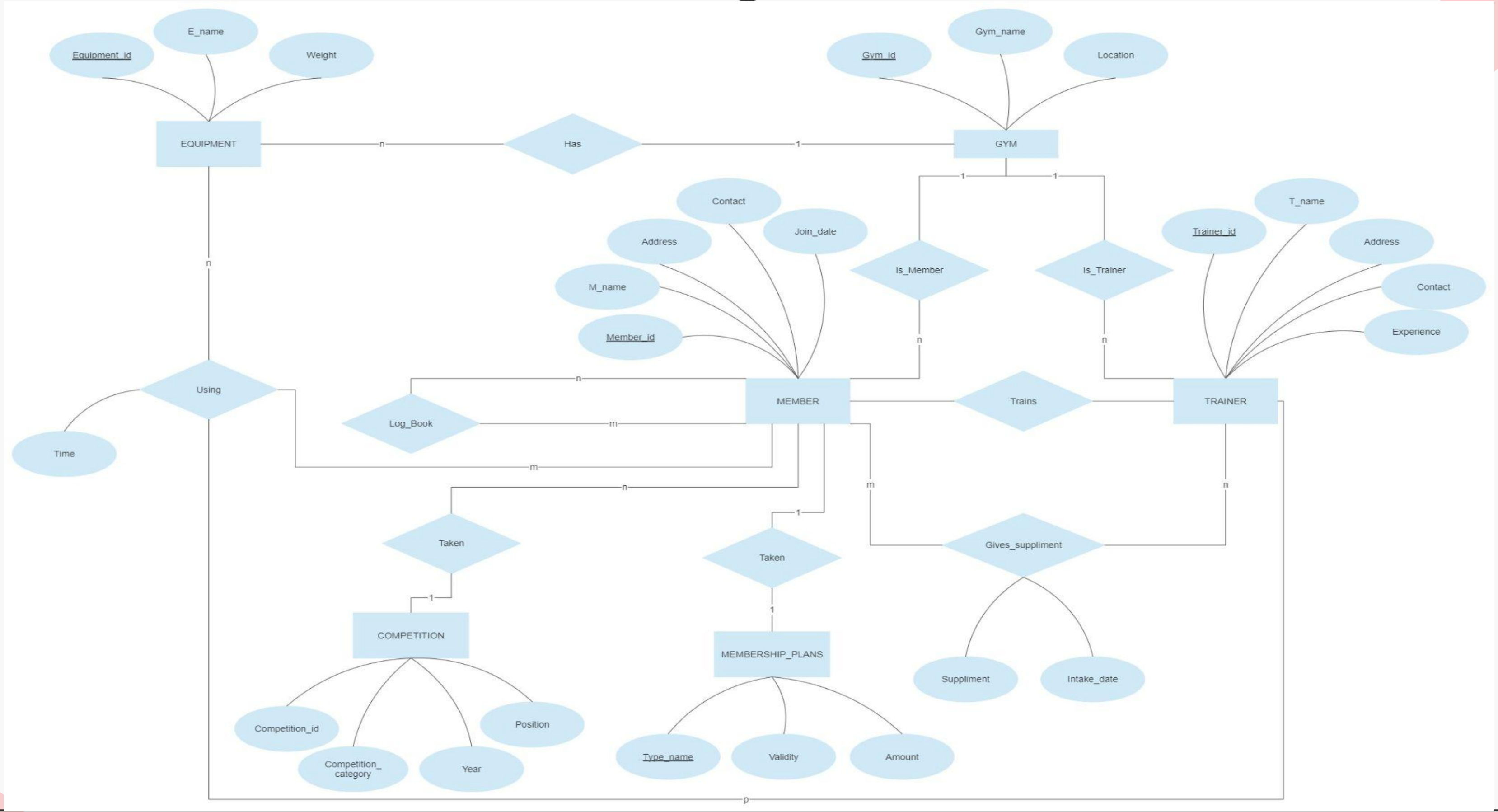
ER Diagram

A visual representation that illustrates the relationships between entities in a database system

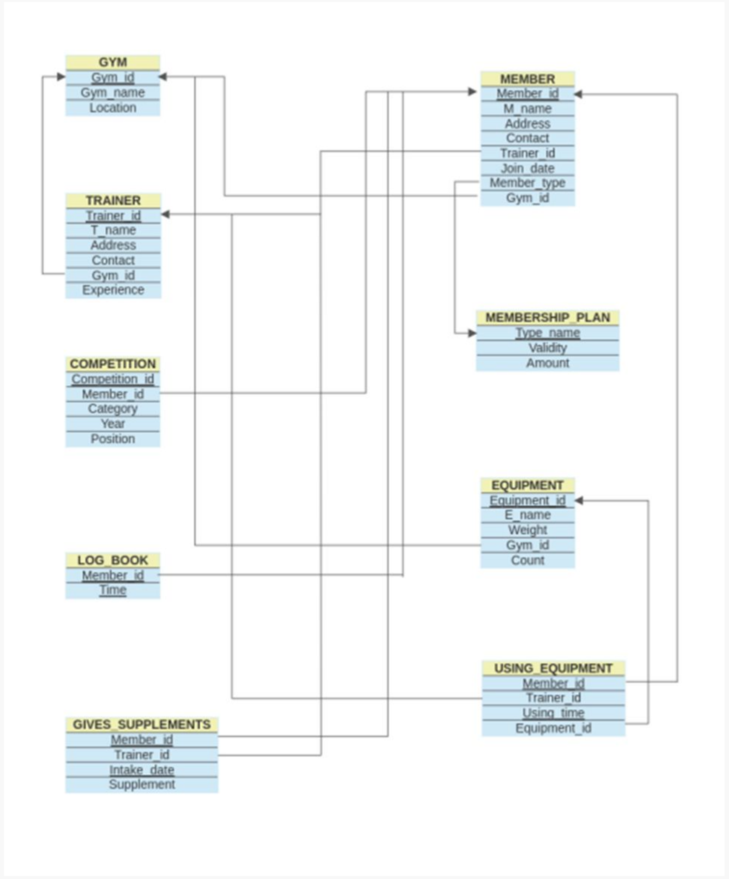
Relational Schema

Logical representation that defines the structure and organization of a relational database, including tables, columns, and their relationships

ER Diagram

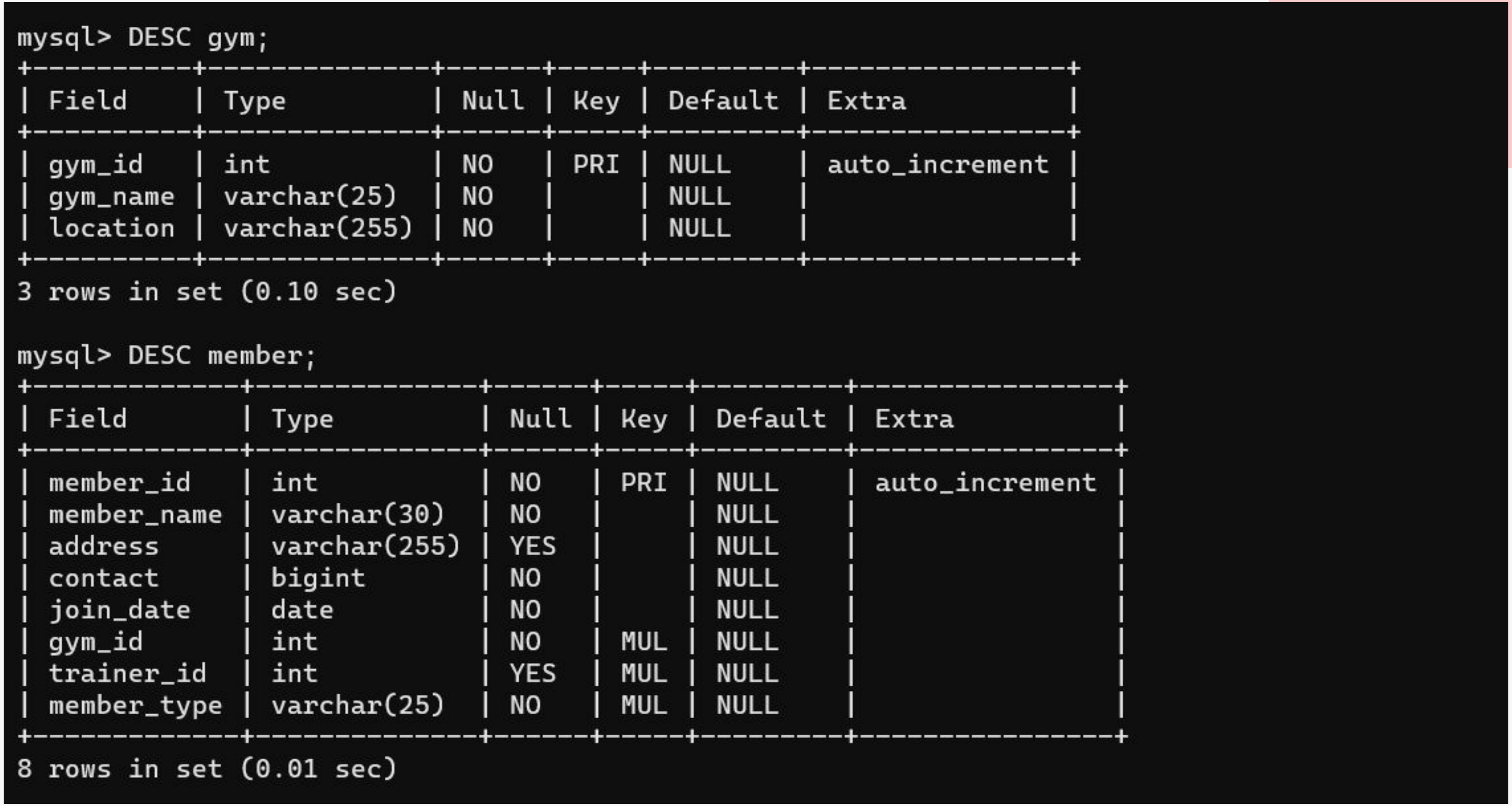

Equipment_id
E_name
Weight
EQUIPMENT
n
Has
1
GYM
Gym_id
Gym_name
Location
1
1
Is_Member
Is_Trainer
n
n
Contact
Join_date
Address
M_name
Member_id
MEMBER
T_name
Trainer_id
Address
Contact
Experience
TRAINER
Trains
n
Using
Time
n
Log_Book
m
m
n
Taken
1
COMPETITION
Competition_id
Competition_category
Year
Position
1
Taken
1
MEMBERSHIP_PLANS
Type_name
Validity
Amount
m
n
Gives_suppliment
Suppliment
Intake_date
p

Relational Schema

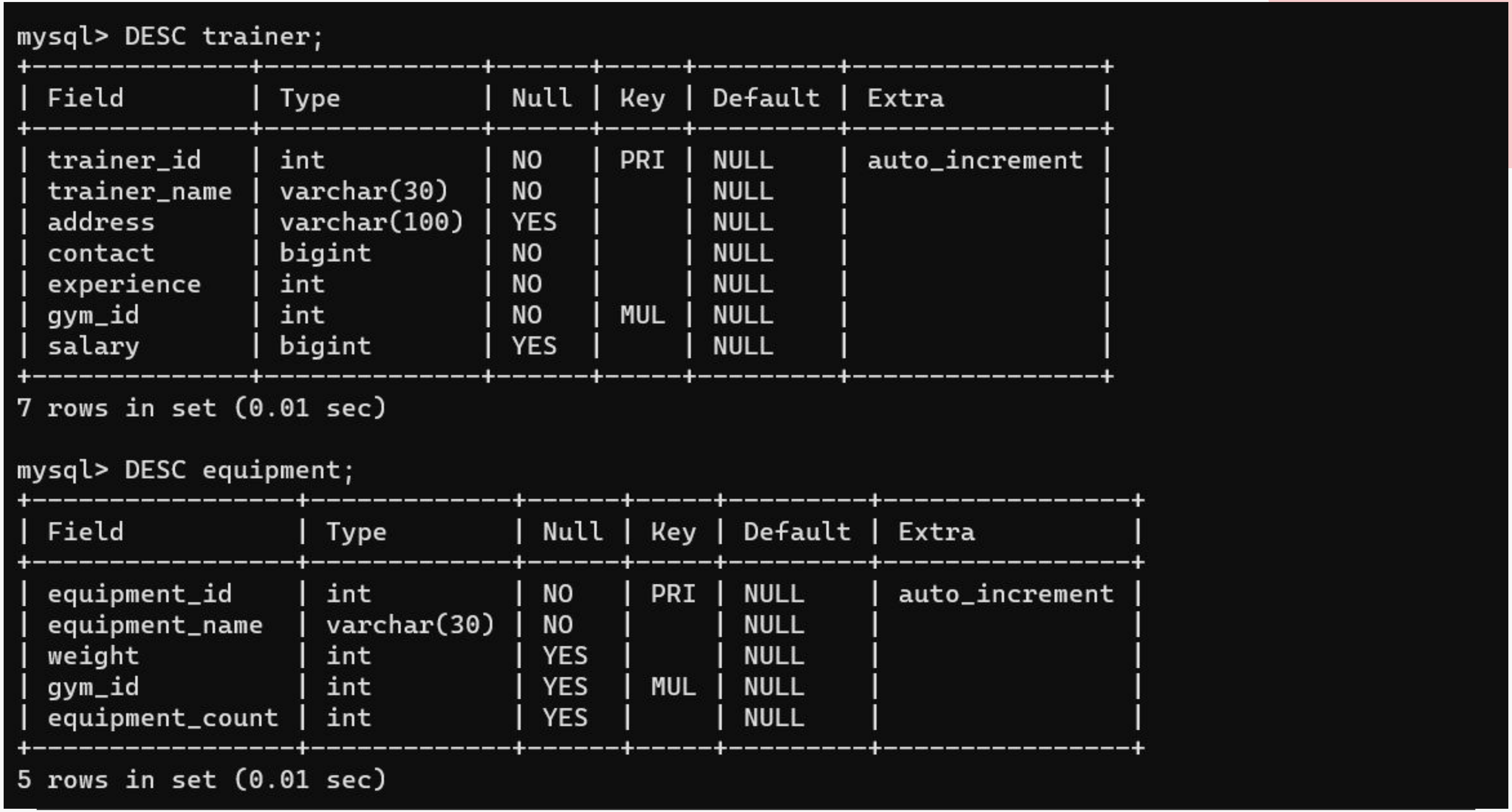


03

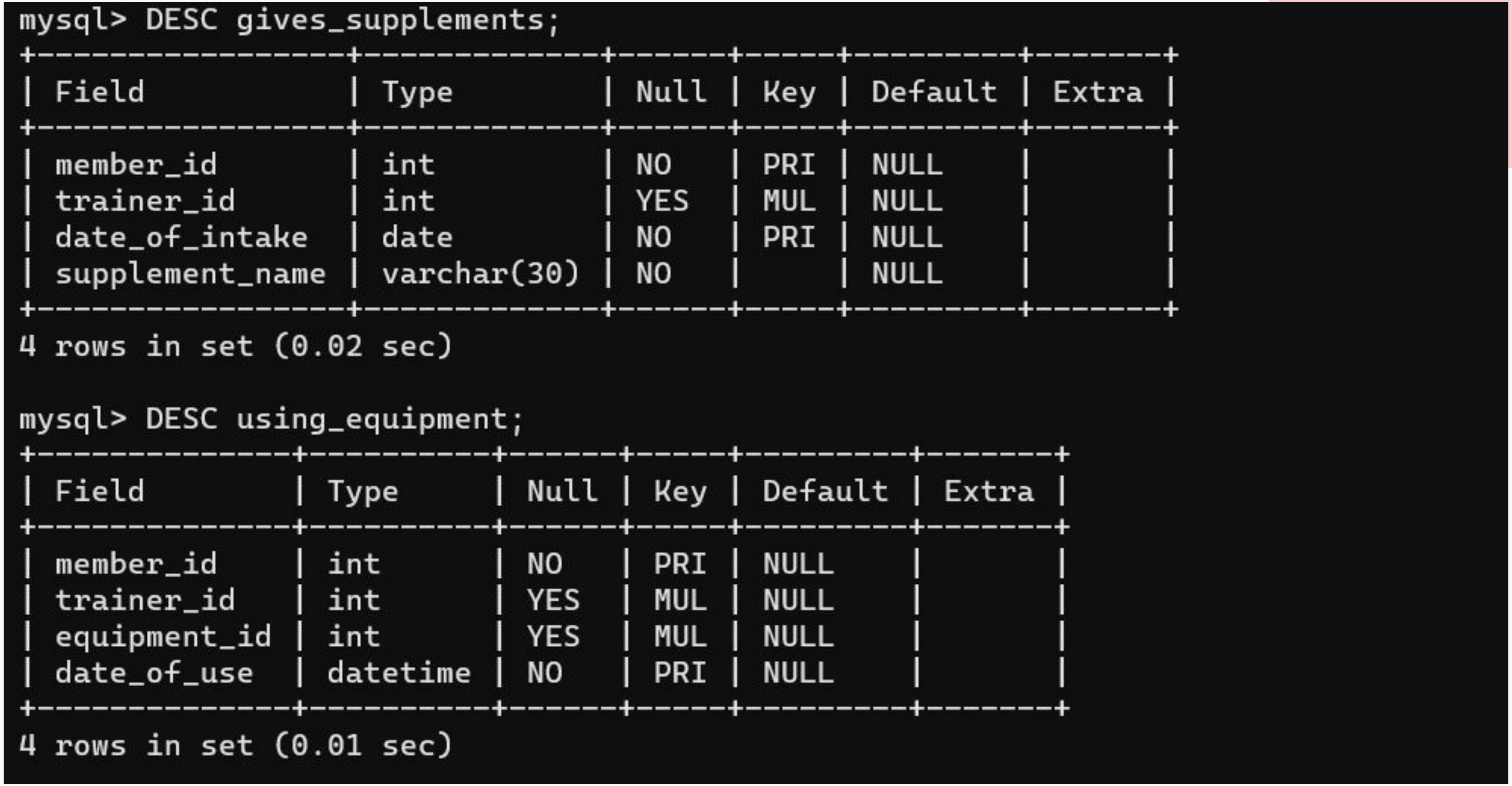
Implementation



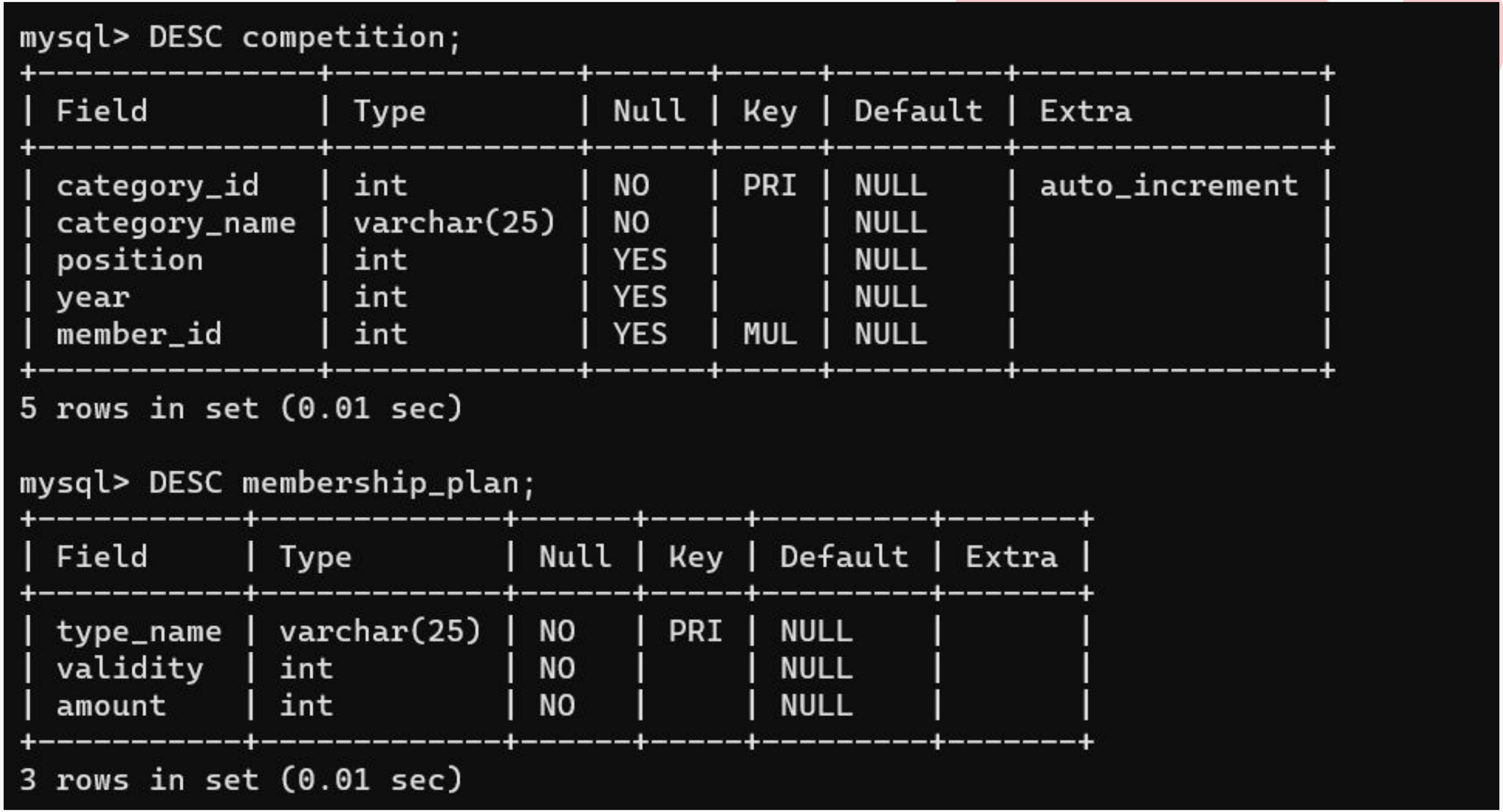
```
mysql> DESC gym;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| gym_id   | int          | NO   | PRI | NULL    | auto_increment |
| gym_name | varchar(25)  | NO   |     | NULL    |                |
| location | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.10 sec)

mysql> DESC member;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| member_id   | int          | NO   | PRI | NULL    | auto_increment |
| member_name | varchar(30)  | NO   |     | NULL    |                |
| address     | varchar(255) | YES  |     | NULL    |                |
| contact     | bigint       | NO   |     | NULL    |                |
| join_date   | date         | NO   |     | NULL    |                |
| gym_id      | int          | NO   | MUL | NULL    |                |
| trainer_id  | int          | YES  | MUL | NULL    |                |
| member_type | varchar(25)  | NO   | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```



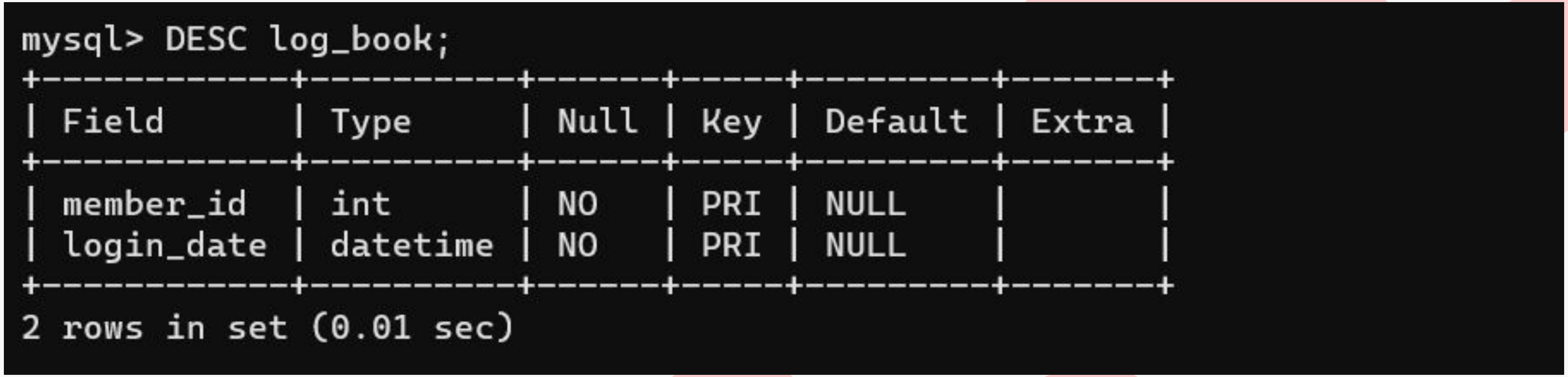
```
mysql> DESC trainer;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| trainer_id   | int          | NO   | PRI | NULL    | auto_increment |
| trainer_name | varchar(30)  | NO   |     | NULL    |                |
| address      | varchar(100) | YES  |     | NULL    |                |
| contact      | bigint       | NO   |     | NULL    |                |
| experience   | int          | NO   |     | NULL    |                |
| gym_id       | int          | NO   | MUL | NULL    |                |
| salary       | bigint       | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)

mysql> DESC equipment;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| equipment_id    | int         | NO   | PRI | NULL    | auto_increment |
| equipment_name  | varchar(30) | NO   |     | NULL    |                |
| weight          | int         | YES  |     | NULL    |                |
| gym_id          | int         | YES  | MUL | NULL    |                |
| equipment_count | int         | YES  |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```



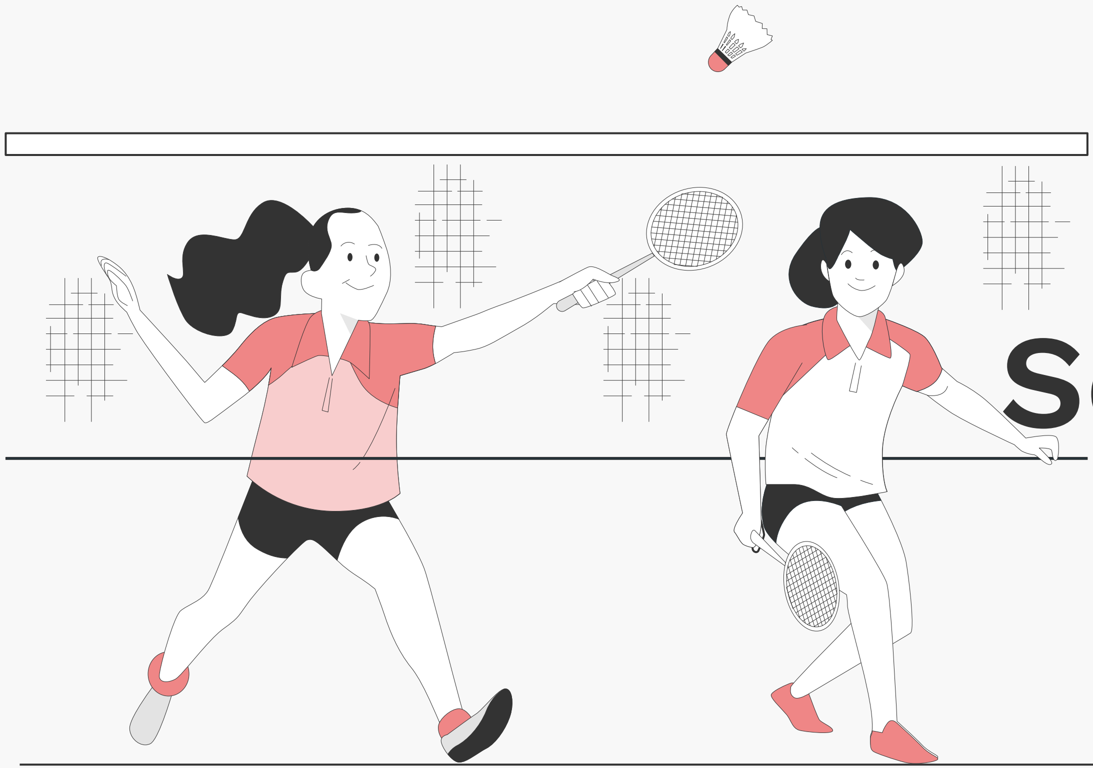
```
mysql> DESC gives_supplements;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| member_id       | int         | NO   | PRI | NULL    |       |
| trainer_id      | int         | YES  | MUL | NULL    |       |
| date_of_intake  | date        | NO   | PRI | NULL    |       |
| supplement_name | varchar(30) | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.02 sec)

mysql> DESC using_equipment;
+--------------+----------+------+-----+---------+-------+
| Field        | Type     | Null | Key | Default | Extra |
+--------------+----------+------+-----+---------+-------+
| member_id    | int      | NO   | PRI | NULL    |       |
| trainer_id   | int      | YES  | MUL | NULL    |       |
| equipment_id | int      | YES  | MUL | NULL    |       |
| date_of_use  | datetime | NO   | PRI | NULL    |       |
+--------------+----------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```



```
mysql> DESC competition;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| category_id   | int         | NO   | PRI | NULL    | auto_increment |
| category_name | varchar(25) | NO   |     | NULL    |                |
| position      | int         | YES  |     | NULL    |                |
| year          | int         | YES  |     | NULL    |                |
| member_id     | int         | YES  | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> DESC membership_plan;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| type_name | varchar(25) | NO   | PRI | NULL    |       |
| validity  | int         | NO   |     | NULL    |       |
| amount    | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

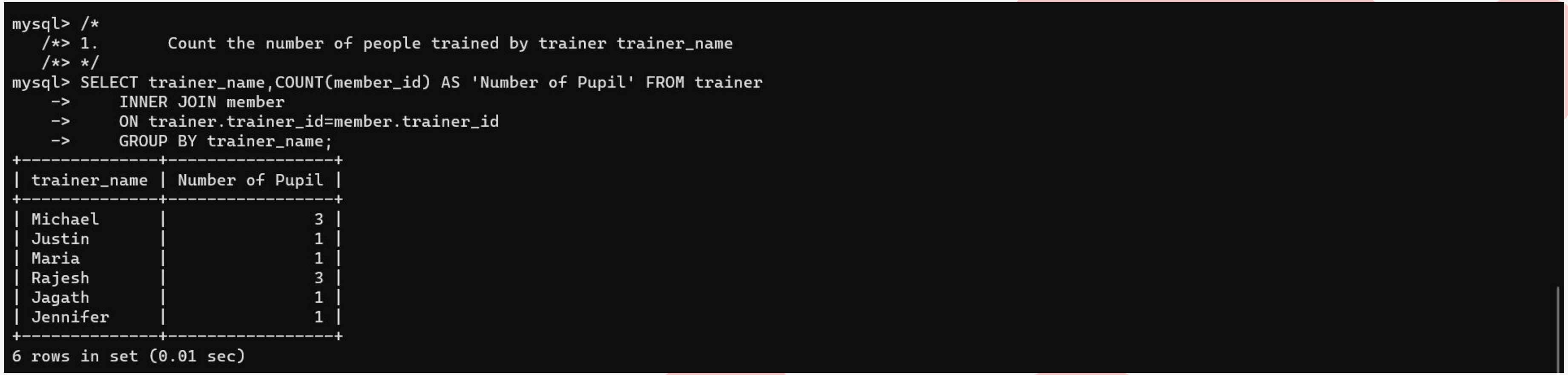


```
mysql> DESC log_book;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| member_id  | int      | NO   | PRI | NULL    |       |
| login_date | datetime | NO   | PRI | NULL    |       |
+------------+----------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

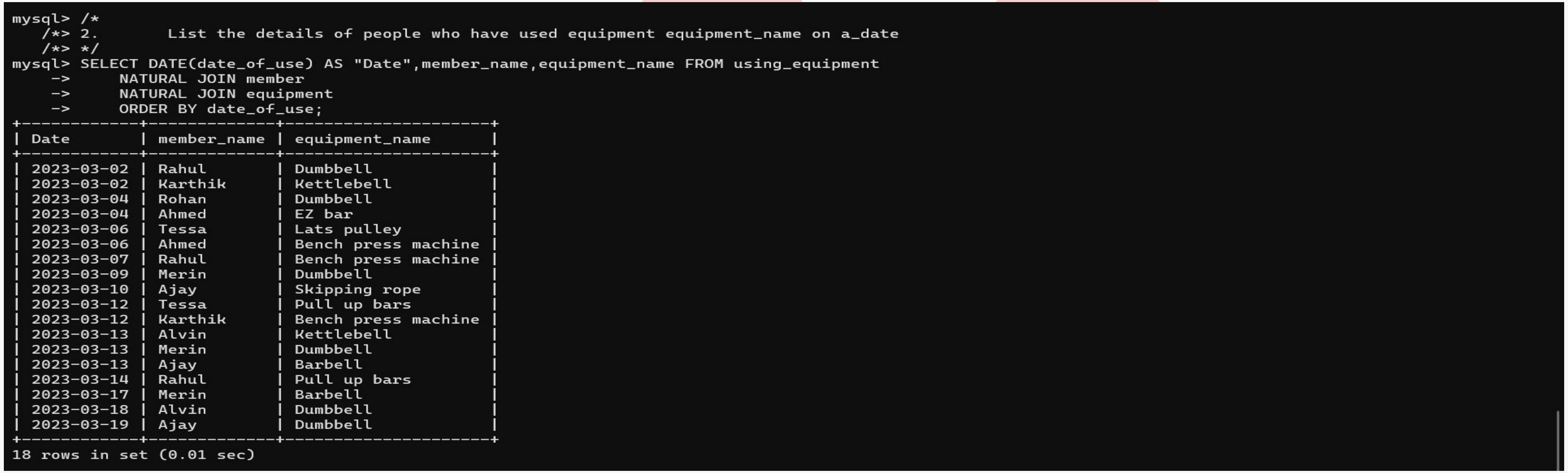


04

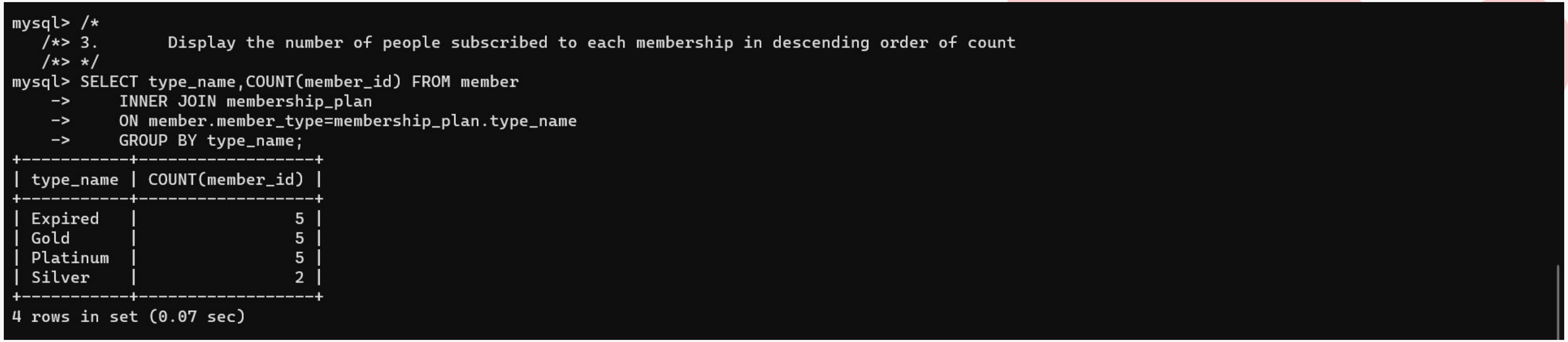
Output Screenshots

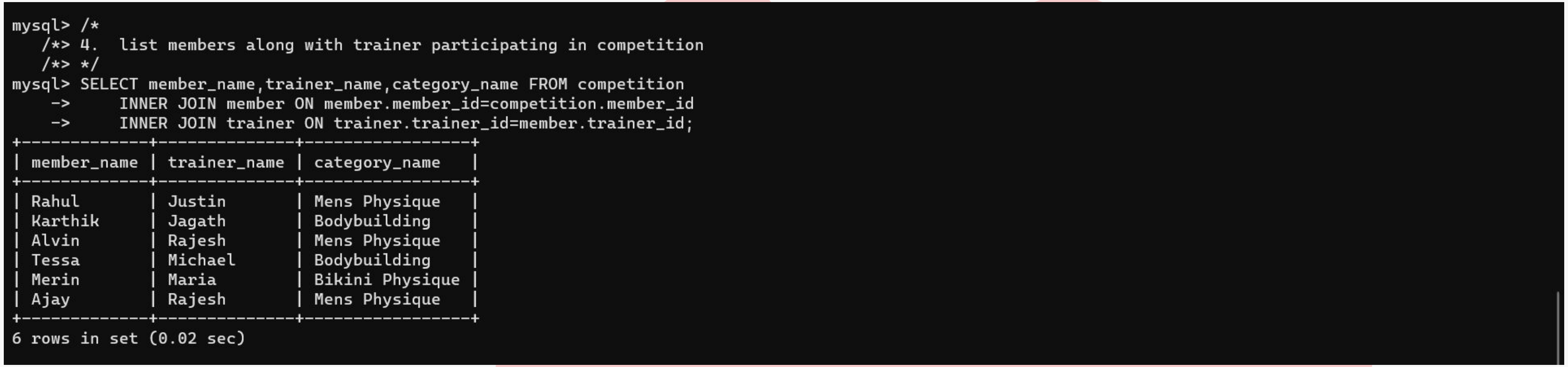


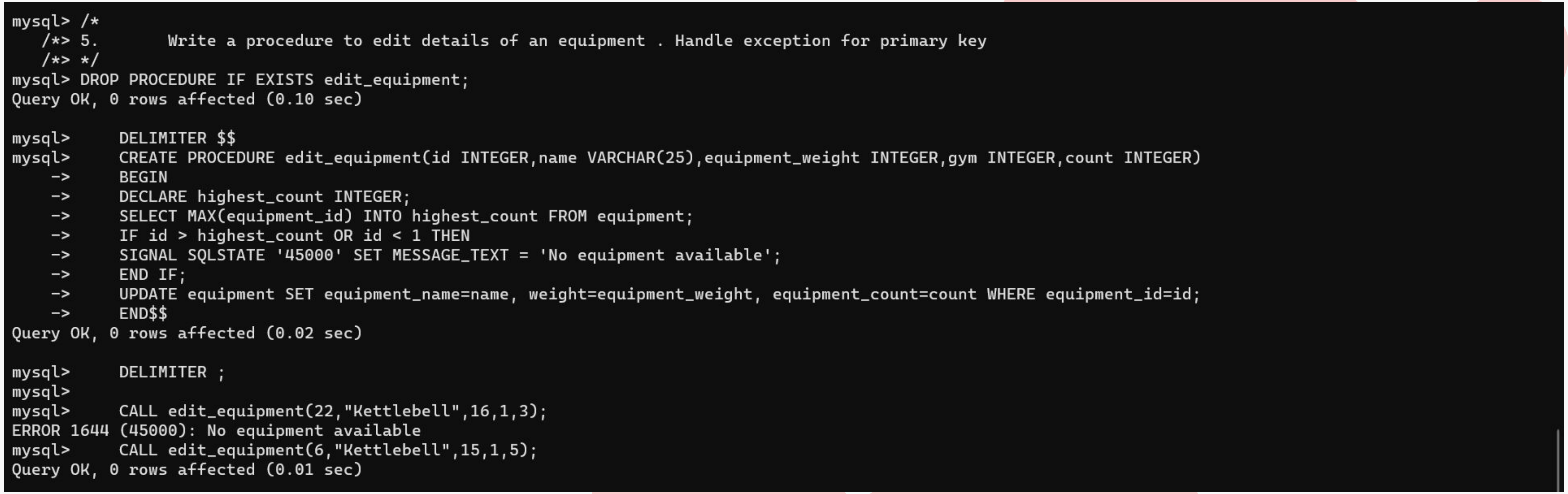
```
mysql> /*
   /*> 1.       Count the number of people trained by trainer trainer_name
   /*> */
mysql> SELECT trainer_name,COUNT(member_id) AS 'Number of Pupil' FROM trainer
    ->     INNER JOIN member
    ->     ON trainer.trainer_id=member.trainer_id
    ->     GROUP BY trainer_name;
+--------------+-----------------+
| trainer_name | Number of Pupil |
+--------------+-----------------+
| Michael      |               3 |
| Justin       |               1 |
| Maria        |               1 |
| Rajesh       |               3 |
| Jagath       |               1 |
| Jennifer     |               1 |
+--------------+-----------------+
6 rows in set (0.01 sec)
```

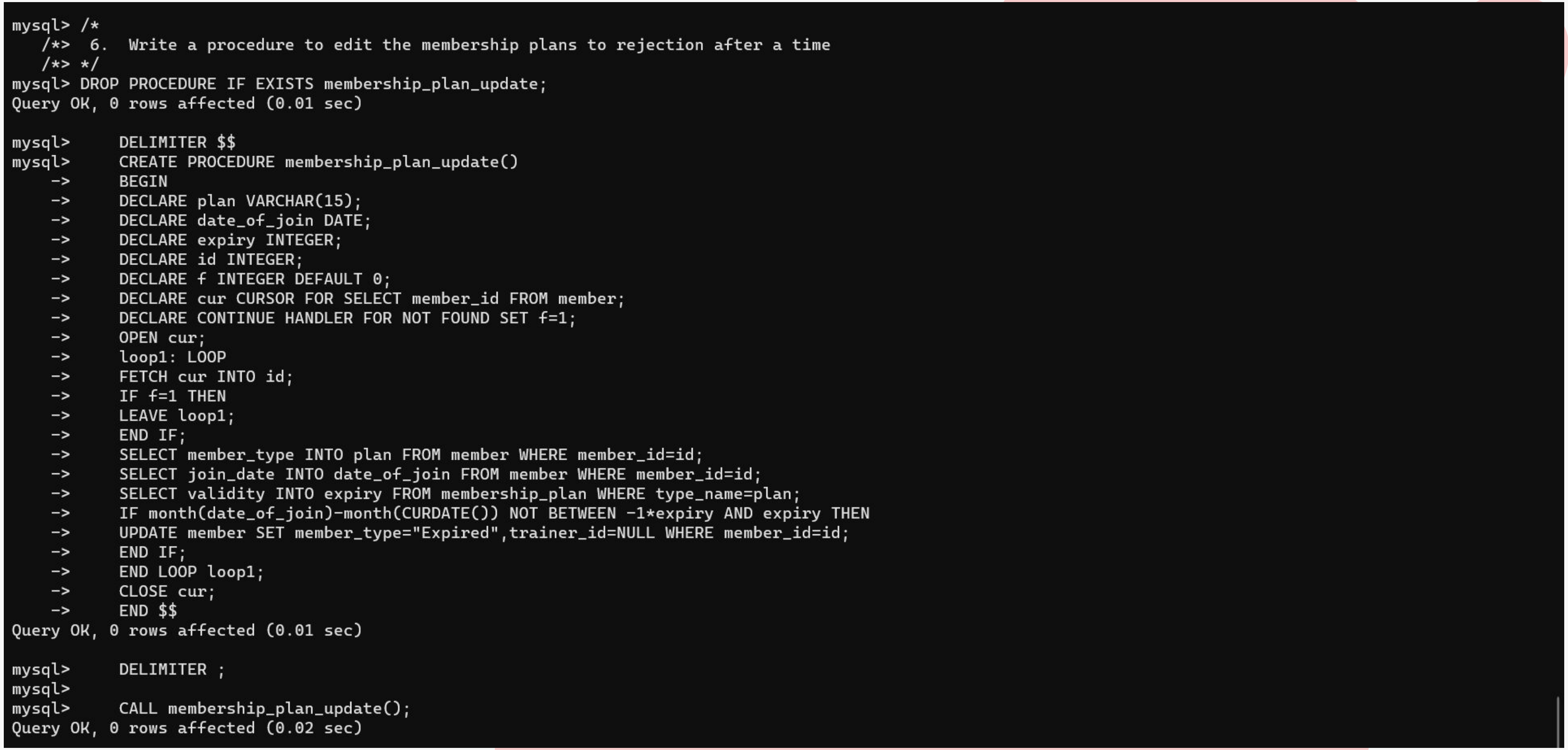


```
mysql> /*
   /*> 2.       List the details of people who have used equipment equipment_name on a_date
   /*> */
mysql> SELECT DATE(date_of_use) AS "Date",member_name,equipment_name FROM using_equipment
    ->     NATURAL JOIN member
    ->     NATURAL JOIN equipment
    ->     ORDER BY date_of_use;
+------------+-------------+---------------------+
| Date       | member_name | equipment_name      |
+------------+-------------+---------------------+
| 2023-03-02 | Rahul       | Dumbbell            |
| 2023-03-02 | Karthik     | Kettlebell          |
| 2023-03-04 | Rohan       | Dumbbell            |
| 2023-03-04 | Ahmed       | EZ bar              |
| 2023-03-06 | Tessa       | Lats pulley         |
| 2023-03-06 | Ahmed       | Bench press machine |
| 2023-03-07 | Rahul       | Bench press machine |
| 2023-03-09 | Merin       | Dumbbell            |
| 2023-03-10 | Ajay        | Skipping rope       |
| 2023-03-12 | Tessa       | Pull up bars        |
| 2023-03-12 | Karthik     | Bench press machine |
| 2023-03-13 | Alvin       | Kettlebell          |
| 2023-03-13 | Merin       | Dumbbell            |
| 2023-03-13 | Ajay        | Barbell             |
| 2023-03-14 | Rahul       | Pull up bars        |
| 2023-03-17 | Merin       | Barbell             |
| 2023-03-18 | Alvin       | Dumbbell            |
| 2023-03-19 | Ajay        | Dumbbell            |
+------------+-------------+---------------------+
18 rows in set (0.01 sec)
```

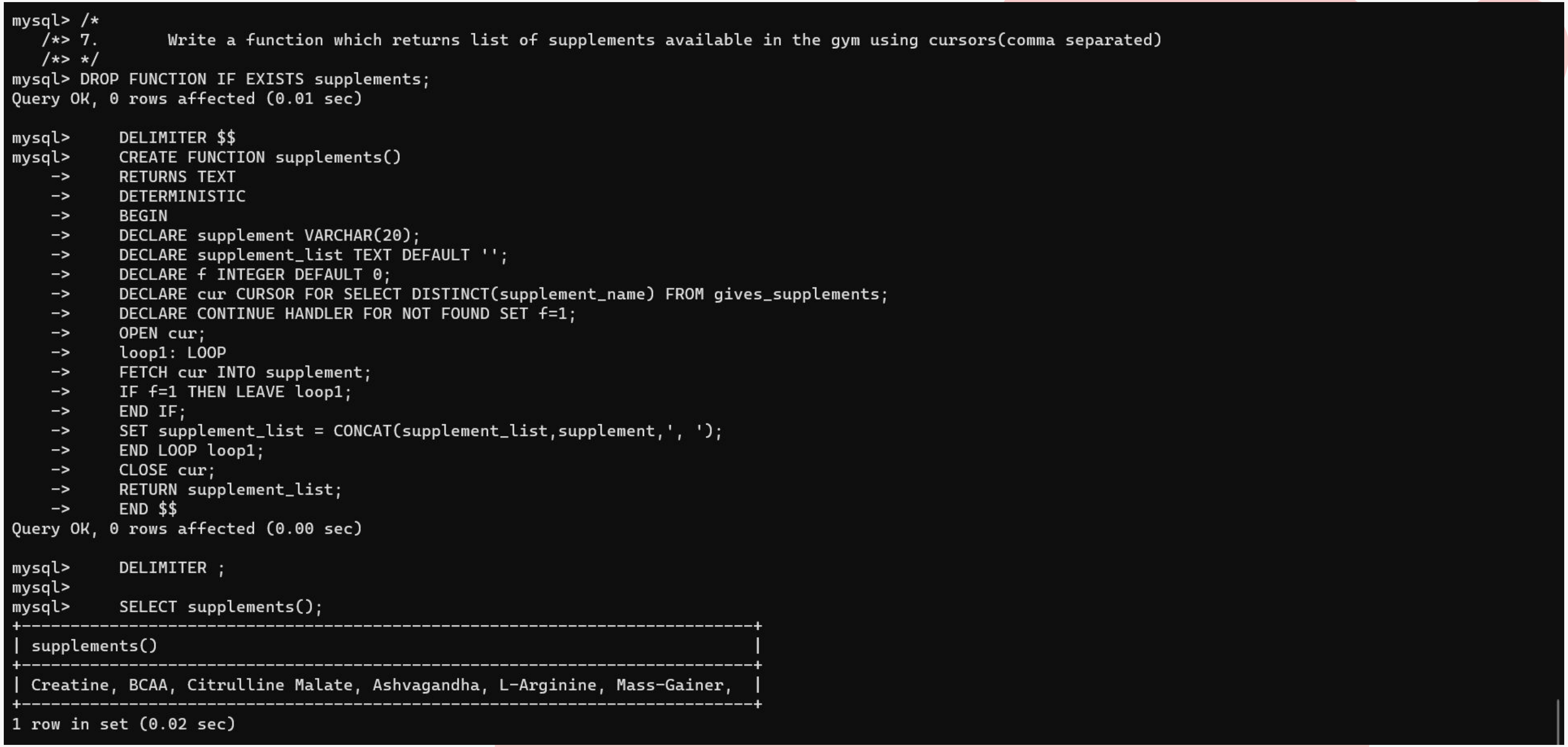

```
mysql> /*
   /*> 3.       Display the number of people subscribed to each membership in descending order of count
   /*> */
mysql> SELECT type_name,COUNT(member_id) FROM member
    ->     INNER JOIN membership_plan
    ->     ON member.member_type=membership_plan.type_name
    ->     GROUP BY type_name;
+-----------+------------------+
| type_name | COUNT(member_id) |
+-----------+------------------+
| Expired   |                5 |
| Gold      |                5 |
| Platinum  |                5 |
| Silver    |                2 |
+-----------+------------------+
4 rows in set (0.07 sec)
```


```
mysql> /*
   /*> 4.  list members along with trainer participating in competition
   /*> */
mysql> SELECT member_name,trainer_name,category_name FROM competition
    ->     INNER JOIN member ON member.member_id=competition.member_id
    ->     INNER JOIN trainer ON trainer.trainer_id=member.trainer_id;
+-------------+--------------+-----------------+
| member_name | trainer_name | category_name   |
+-------------+--------------+-----------------+
| Rahul       | Justin       | Mens Physique   |
| Karthik     | Jagath       | Bodybuilding    |
| Alvin       | Rajesh       | Mens Physique   |
| Tessa       | Michael      | Bodybuilding    |
| Merin       | Maria        | Bikini Physique |
| Ajay        | Rajesh       | Mens Physique   |
+-------------+--------------+-----------------+
6 rows in set (0.02 sec)
```

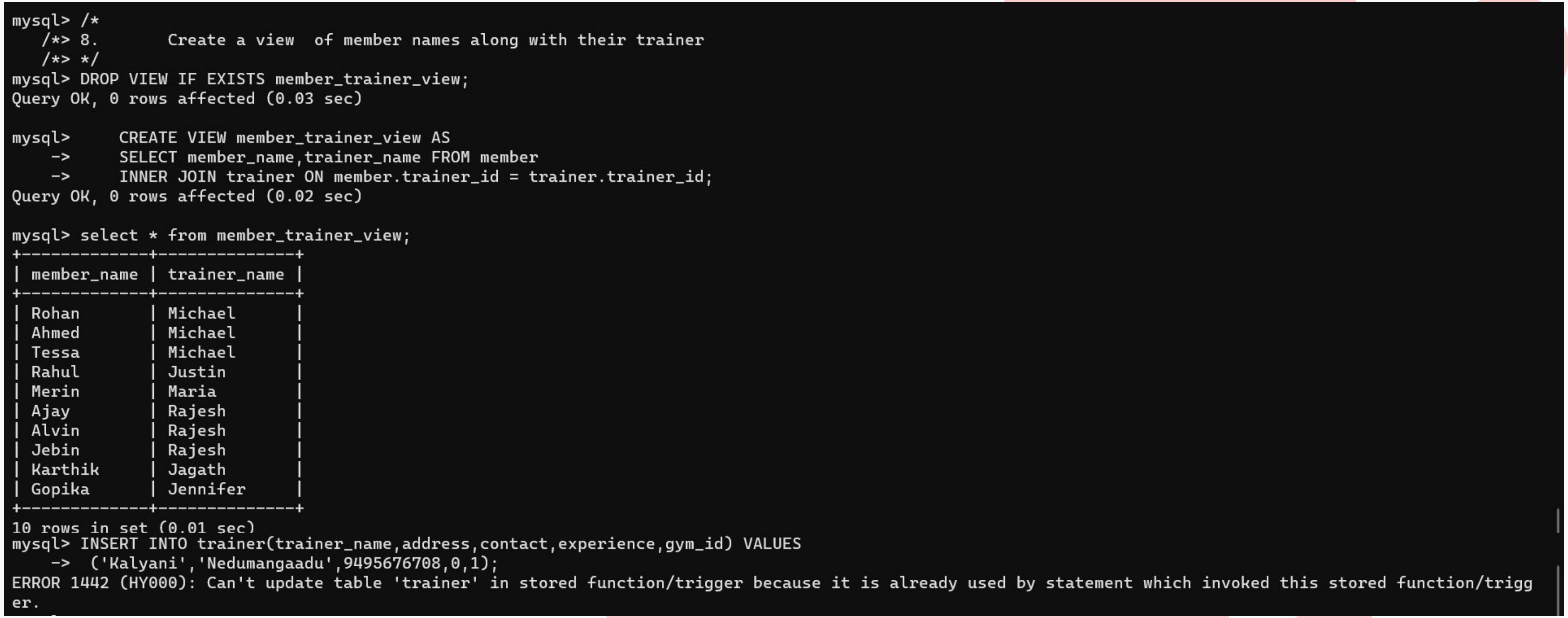

```
mysql> /*
   /*> 5.       Write a procedure to edit details of an equipment . Handle exception for primary key
   /*> */
mysql> DROP PROCEDURE IF EXISTS edit_equipment;
Query OK, 0 rows affected (0.10 sec)

mysql>     DELIMITER $$
mysql>     CREATE PROCEDURE edit_equipment(id INTEGER,name VARCHAR(25),equipment_weight INTEGER,gym INTEGER,count INTEGER)
    ->     BEGIN
    ->     DECLARE highest_count INTEGER;
    ->     SELECT MAX(equipment_id) INTO highest_count FROM equipment;
    ->     IF id > highest_count OR id < 1 THEN
    ->     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No equipment available';
    ->     END IF;
    ->     UPDATE equipment SET equipment_name=name, weight=equipment_weight, equipment_count=count WHERE equipment_id=id;
    ->     END$$
Query OK, 0 rows affected (0.02 sec)

mysql>     DELIMITER ;
mysql>
mysql>     CALL edit_equipment(22,"Kettlebell",16,1,3);
ERROR 1644 (45000): No equipment available
mysql>     CALL edit_equipment(6,"Kettlebell",15,1,5);
Query OK, 0 rows affected (0.01 sec)
```



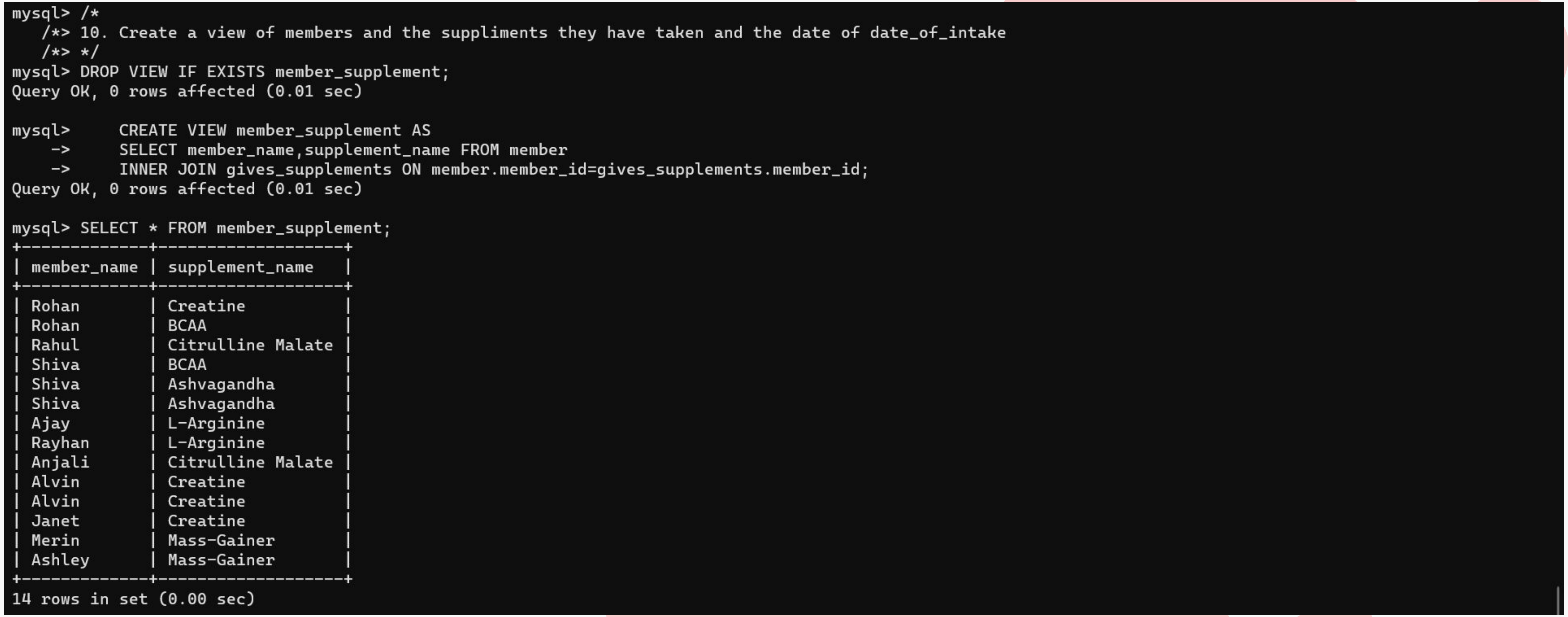
```
mysql> /*
   /*>  6.  Write a procedure to edit the membership plans to rejection after a time
   /*> */
mysql> DROP PROCEDURE IF EXISTS membership_plan_update;
Query OK, 0 rows affected (0.01 sec)

mysql>     DELIMITER $$
mysql>     CREATE PROCEDURE membership_plan_update()
    ->     BEGIN
    ->     DECLARE plan VARCHAR(15);
    ->     DECLARE date_of_join DATE;
    ->     DECLARE expiry INTEGER;
    ->     DECLARE id INTEGER;
    ->     DECLARE f INTEGER DEFAULT 0;
    ->     DECLARE cur CURSOR FOR SELECT member_id FROM member;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
    ->     OPEN cur;
    ->     loop1: LOOP
    ->     FETCH cur INTO id;
    ->     IF f=1 THEN
    ->     LEAVE loop1;
    ->     END IF;
    ->     SELECT member_type INTO plan FROM member WHERE member_id=id;
    ->     SELECT join_date INTO date_of_join FROM member WHERE member_id=id;
    ->     SELECT validity INTO expiry FROM membership_plan WHERE type_name=plan;
    ->     IF month(date_of_join)-month(CURDATE()) NOT BETWEEN -1*expiry AND expiry THEN
    ->     UPDATE member SET member_type="Expired",trainer_id=NULL WHERE member_id=id;
    ->     END IF;
    ->     END LOOP loop1;
    ->     CLOSE cur;
    ->     END $$
Query OK, 0 rows affected (0.01 sec)

mysql>     DELIMITER ;
mysql>
mysql>     CALL membership_plan_update();
Query OK, 0 rows affected (0.02 sec)
```



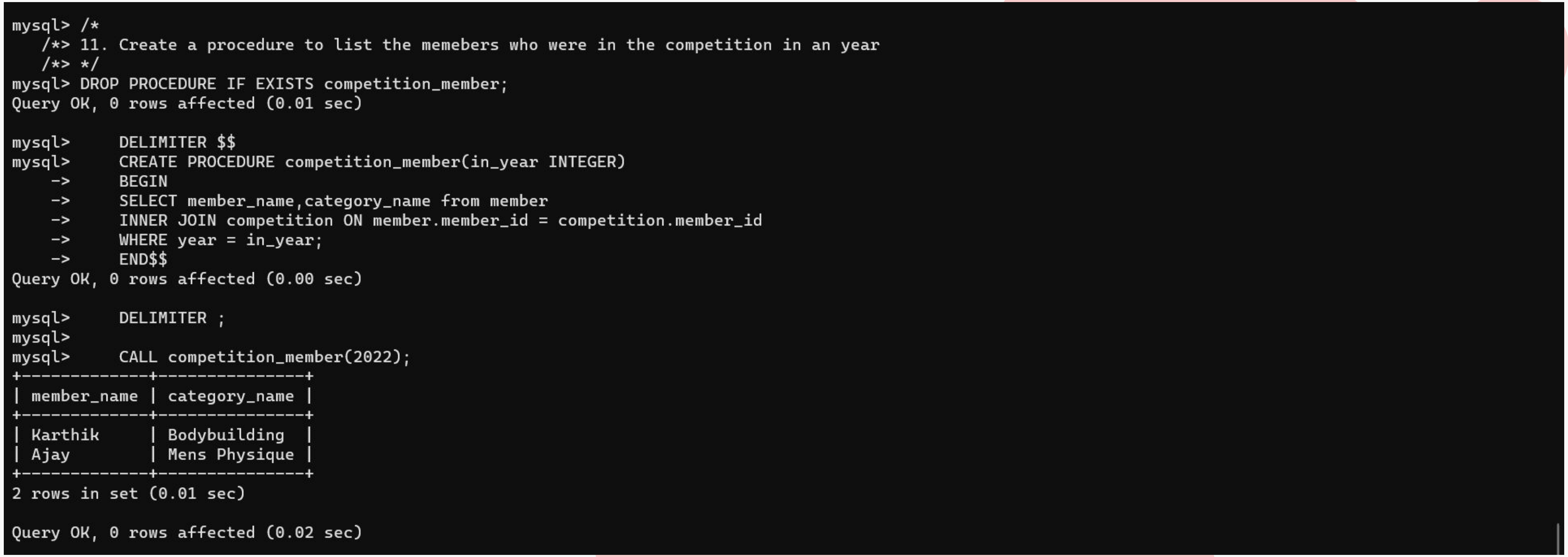
```
mysql> /*
   /*> 7.       Write a function which returns list of supplements available in the gym using cursors(comma separated)
   /*> */
mysql> DROP FUNCTION IF EXISTS supplements;
Query OK, 0 rows affected (0.01 sec)

mysql>     DELIMITER $$
mysql>     CREATE FUNCTION supplements()
    ->     RETURNS TEXT
    ->     DETERMINISTIC
    ->     BEGIN
    ->     DECLARE supplement VARCHAR(20);
    ->     DECLARE supplement_list TEXT DEFAULT '';
    ->     DECLARE f INTEGER DEFAULT 0;
    ->     DECLARE cur CURSOR FOR SELECT DISTINCT(supplement_name) FROM gives_supplements;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET f=1;
    ->     OPEN cur;
    ->     loop1: LOOP
    ->     FETCH cur INTO supplement;
    ->     IF f=1 THEN LEAVE loop1;
    ->     END IF;
    ->     SET supplement_list = CONCAT(supplement_list,supplement,', ');
    ->     END LOOP loop1;
    ->     CLOSE cur;
    ->     RETURN supplement_list;
    ->     END $$
Query OK, 0 rows affected (0.00 sec)

mysql>     DELIMITER ;
mysql>
mysql>     SELECT supplements();
+---------------------------------------------------------------------------+
| supplements()                                                             |
+---------------------------------------------------------------------------+
| Creatine, BCAA, Citrulline Malate, Ashvagandha, L-Arginine, Mass-Gainer,  |
+---------------------------------------------------------------------------+
1 row in set (0.02 sec)
```



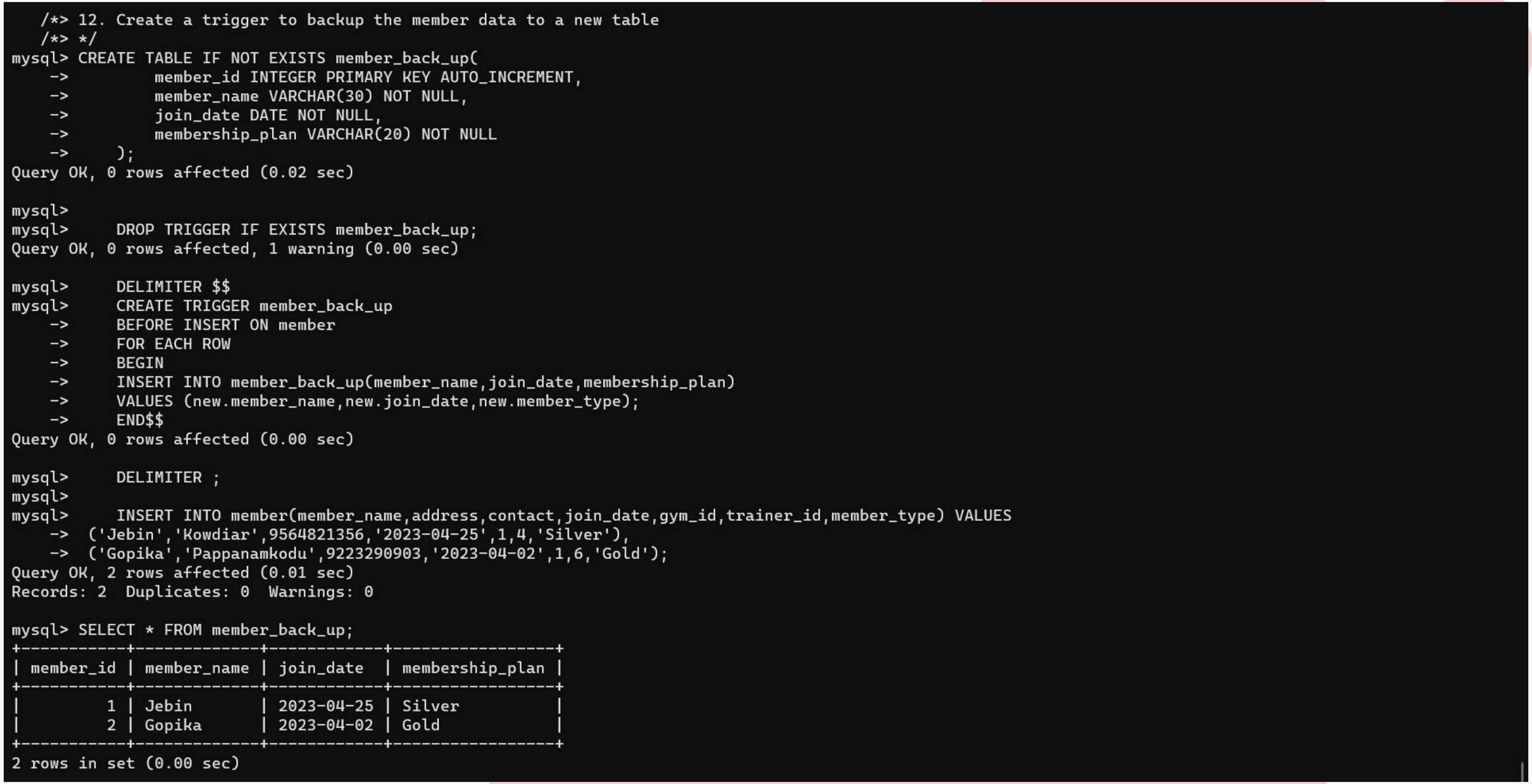
```
mysql> /*
   /*> 8.       Create a view  of member names along with their trainer
   /*> */
mysql> DROP VIEW IF EXISTS member_trainer_view;
Query OK, 0 rows affected (0.03 sec)

mysql>     CREATE VIEW member_trainer_view AS
    ->     SELECT member_name,trainer_name FROM member
    ->     INNER JOIN trainer ON member.trainer_id = trainer.trainer_id;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from member_trainer_view;
+-------------+--------------+
| member_name | trainer_name |
+-------------+--------------+
| Rohan       | Michael      |
| Ahmed       | Michael      |
| Tessa       | Michael      |
| Rahul       | Justin       |
| Merin       | Maria        |
| Ajay        | Rajesh       |
| Alvin       | Rajesh       |
| Jebin       | Rajesh       |
| Karthik     | Jagath       |
| Gopika      | Jennifer     |
+-------------+--------------+
10 rows in set (0.01 sec)
mysql> INSERT INTO trainer(trainer_name,address,contact,experience,gym_id) VALUES
    ->  ('Kalyani','Nedumangaadu',9495676708,0,1);
ERROR 1442 (HY000): Can't update table 'trainer' in stored function/trigger because it is already used by statement which invoked this stored function/trigg
er.
```



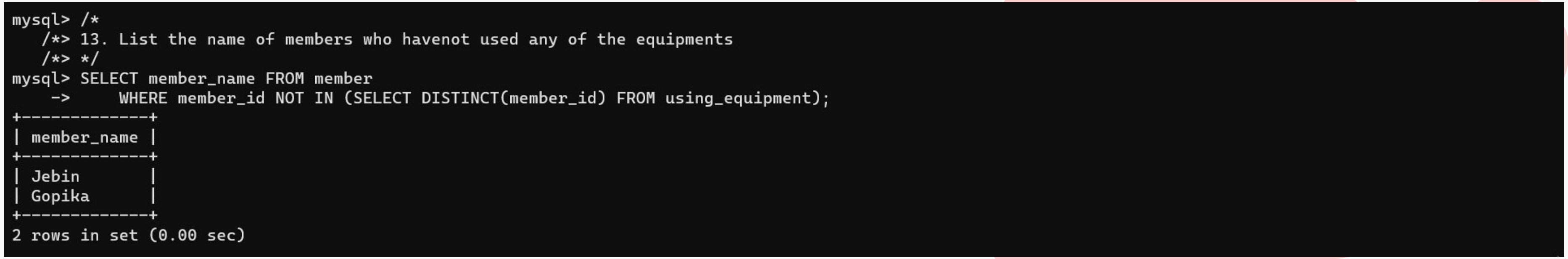
```
mysql> /*
   /*> 10. Create a view of members and the suppliments they have taken and the date of date_of_intake
   /*> */
mysql> DROP VIEW IF EXISTS member_supplement;
Query OK, 0 rows affected (0.01 sec)

mysql>     CREATE VIEW member_supplement AS
    ->     SELECT member_name,supplement_name FROM member
    ->     INNER JOIN gives_supplements ON member.member_id=gives_supplements.member_id;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM member_supplement;
+-------------+-------------------+
| member_name | supplement_name   |
+-------------+-------------------+
| Rohan       | Creatine          |
| Rohan       | BCAA              |
| Rahul       | Citrulline Malate |
| Shiva       | BCAA              |
| Shiva       | Ashvagandha       |
| Shiva       | Ashvagandha       |
| Ajay        | L-Arginine        |
| Rayhan      | L-Arginine        |
| Anjali      | Citrulline Malate |
| Alvin       | Creatine          |
| Alvin       | Creatine          |
| Janet       | Creatine          |
| Merin       | Mass-Gainer       |
| Ashley      | Mass-Gainer       |
+-------------+-------------------+
14 rows in set (0.00 sec)
```



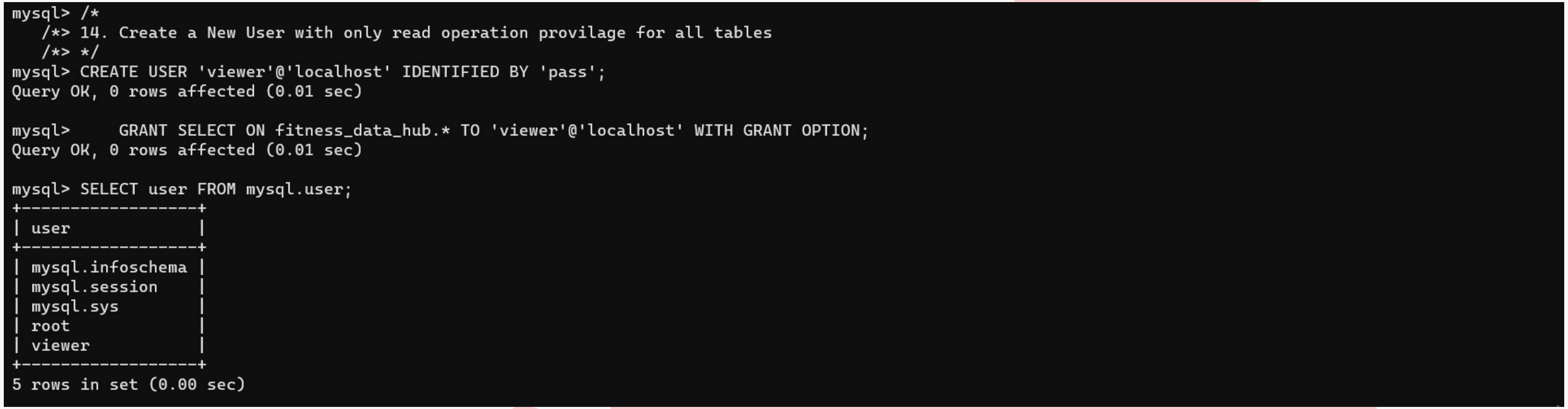
```
mysql> /*
   /*> 11. Create a procedure to list the memebers who were in the competition in an year
   /*> */
mysql> DROP PROCEDURE IF EXISTS competition_member;
Query OK, 0 rows affected (0.01 sec)

mysql>     DELIMITER $$
mysql>     CREATE PROCEDURE competition_member(in_year INTEGER)
    ->     BEGIN
    ->     SELECT member_name,category_name from member
    ->     INNER JOIN competition ON member.member_id = competition.member_id
    ->     WHERE year = in_year;
    ->     END$$
Query OK, 0 rows affected (0.00 sec)

mysql>     DELIMITER ;
mysql>
mysql>     CALL competition_member(2022);
+-------------+---------------+
| member_name | category_name |
+-------------+---------------+
| Karthik     | Bodybuilding  |
| Ajay        | Mens Physique |
+-------------+---------------+
2 rows in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```



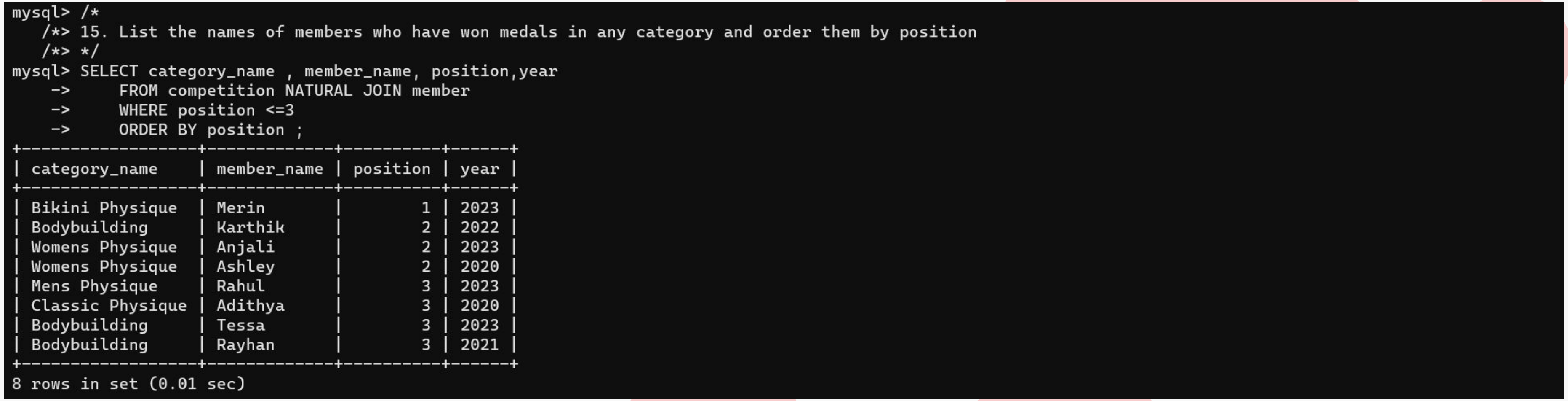
```
   /*> 12. Create a trigger to backup the member data to a new table
   /*> */
mysql> CREATE TABLE IF NOT EXISTS member_back_up(
    ->          member_id INTEGER PRIMARY KEY AUTO_INCREMENT,
    ->          member_name VARCHAR(30) NOT NULL,
    ->          join_date DATE NOT NULL,
    ->          membership_plan VARCHAR(20) NOT NULL
    ->      );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql>     DROP TRIGGER IF EXISTS member_back_up;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>     DELIMITER $$
mysql>     CREATE TRIGGER member_back_up
    ->     BEFORE INSERT ON member
    ->     FOR EACH ROW
    ->     BEGIN
    ->     INSERT INTO member_back_up(member_name,join_date,membership_plan)
    ->     VALUES (new.member_name,new.join_date,new.member_type);
    ->     END$$
Query OK, 0 rows affected (0.00 sec)

mysql>     DELIMITER ;
mysql>
mysql>     INSERT INTO member(member_name,address,contact,join_date,gym_id,trainer_id,member_type) VALUES
    ->  ('Jebin','Kowdiar',9564821356,'2023-04-25',1,4,'Silver'),
    ->  ('Gopika','Pappanamkodu',9223290903,'2023-04-02',1,6,'Gold');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM member_back_up;
+-----------+-------------+------------+-----------------+
| member_id | member_name | join_date  | membership_plan |
+-----------+-------------+------------+-----------------+
|         1 | Jebin       | 2023-04-25 | Silver          |
|         2 | Gopika      | 2023-04-02 | Gold            |
+-----------+-------------+------------+-----------------+
2 rows in set (0.00 sec)
```



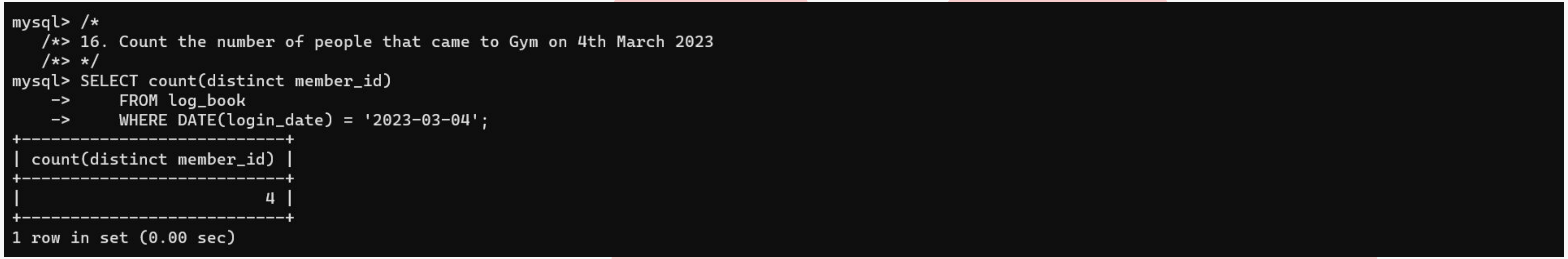
```
mysql> /*
   /*> 13. List the name of members who havenot used any of the equipments
   /*> */
mysql> SELECT member_name FROM member
    ->     WHERE member_id NOT IN (SELECT DISTINCT(member_id) FROM using_equipment);
+-------------+
| member_name |
+-------------+
| Jebin       |
| Gopika      |
+-------------+
2 rows in set (0.00 sec)
```



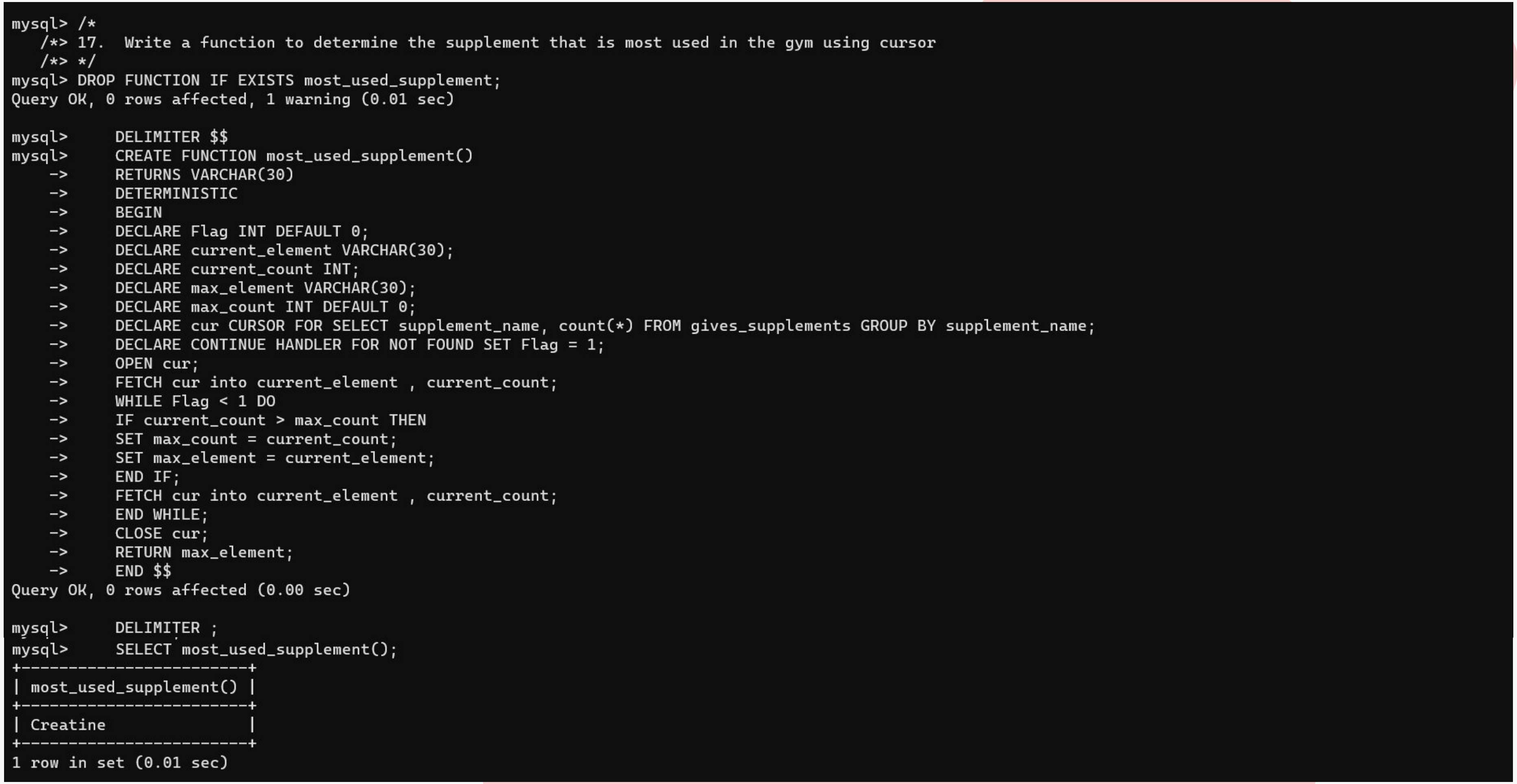
```
mysql> /*
   /*> 14. Create a New User with only read operation provilage for all tables
   /*> */
mysql> CREATE USER 'viewer'@'localhost' IDENTIFIED BY 'pass';
Query OK, 0 rows affected (0.01 sec)

mysql>     GRANT SELECT ON fitness_data_hub.* TO 'viewer'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT user FROM mysql.user;
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
| viewer           |
+------------------+
5 rows in set (0.00 sec)
```



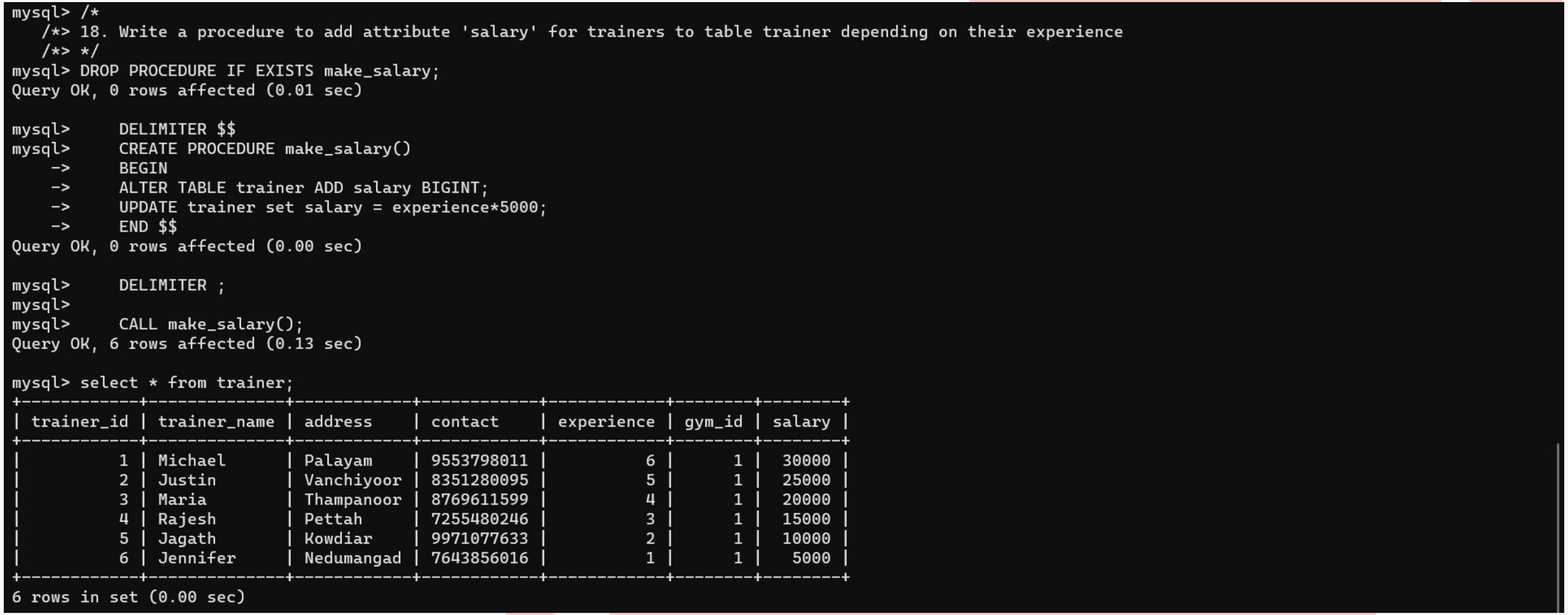
```
mysql> /*
   /*> 15. List the names of members who have won medals in any category and order them by position
   /*> */
mysql> SELECT category_name , member_name, position,year
    ->     FROM competition NATURAL JOIN member
    ->     WHERE position <=3
    ->     ORDER BY position ;
+------------------+-------------+----------+------+
| category_name    | member_name | position | year |
+------------------+-------------+----------+------+
| Bikini Physique  | Merin       |        1 | 2023 |
| Bodybuilding     | Karthik     |        2 | 2022 |
| Womens Physique  | Anjali      |        2 | 2023 |
| Womens Physique  | Ashley      |        2 | 2020 |
| Mens Physique    | Rahul       |        3 | 2023 |
| Classic Physique | Adithya     |        3 | 2020 |
| Bodybuilding     | Tessa       |        3 | 2023 |
| Bodybuilding     | Rayhan      |        3 | 2021 |
+------------------+-------------+----------+------+
8 rows in set (0.01 sec)
```



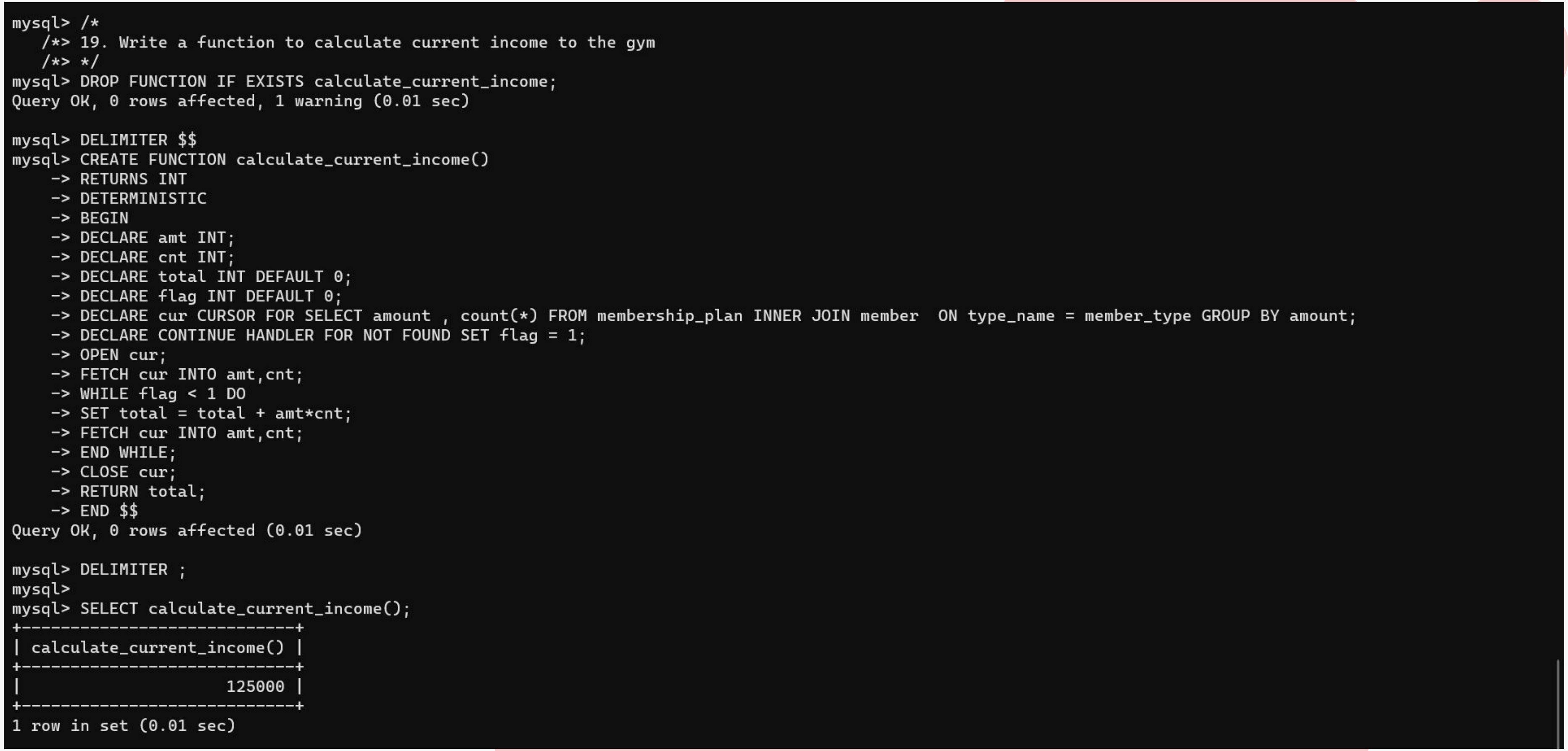
```
mysql> /*
   /*> 16. Count the number of people that came to Gym on 4th March 2023
   /*> */
mysql> SELECT count(distinct member_id)
    ->     FROM log_book
    ->     WHERE DATE(login_date) = '2023-03-04';
+---------------------------+
| count(distinct member_id) |
+---------------------------+
|                         4 |
+---------------------------+
1 row in set (0.00 sec)
```



```
mysql> /*
   /*> 17.  Write a function to determine the supplement that is most used in the gym using cursor
   /*> */
mysql> DROP FUNCTION IF EXISTS most_used_supplement;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql>     DELIMITER $$
mysql>     CREATE FUNCTION most_used_supplement()
    ->     RETURNS VARCHAR(30)
    ->     DETERMINISTIC
    ->     BEGIN
    ->     DECLARE Flag INT DEFAULT 0;
    ->     DECLARE current_element VARCHAR(30);
    ->     DECLARE current_count INT;
    ->     DECLARE max_element VARCHAR(30);
    ->     DECLARE max_count INT DEFAULT 0;
    ->     DECLARE cur CURSOR FOR SELECT supplement_name, count(*) FROM gives_supplements GROUP BY supplement_name;
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET Flag = 1;
    ->     OPEN cur;
    ->     FETCH cur into current_element , current_count;
    ->     WHILE Flag < 1 DO
    ->     IF current_count > max_count THEN
    ->     SET max_count = current_count;
    ->     SET max_element = current_element;
    ->     END IF;
    ->     FETCH cur into current_element , current_count;
    ->     END WHILE;
    ->     CLOSE cur;
    ->     RETURN max_element;
    ->     END $$
Query OK, 0 rows affected (0.00 sec)

mysql>     DELIMITER ;
mysql>     SELECT most_used_supplement();
+------------------------+
| most_used_supplement() |
+------------------------+
| Creatine               |
+------------------------+
1 row in set (0.01 sec)
```



```
mysql> /*
   /*> 18. Write a procedure to add attribute 'salary' for trainers to table trainer depending on their experience
   /*> */
mysql> DROP PROCEDURE IF EXISTS make_salary;
Query OK, 0 rows affected (0.01 sec)

mysql>     DELIMITER $$
mysql>     CREATE PROCEDURE make_salary()
    ->     BEGIN
    ->     ALTER TABLE trainer ADD salary BIGINT;
    ->     UPDATE trainer set salary = experience*5000;
    ->     END $$
Query OK, 0 rows affected (0.00 sec)

mysql>     DELIMITER ;
mysql>
mysql>     CALL make_salary();
Query OK, 6 rows affected (0.13 sec)

mysql> select * from trainer;
+------------+--------------+------------+------------+------------+--------+--------+
| trainer_id | trainer_name | address    | contact    | experience | gym_id | salary |
+------------+--------------+------------+------------+------------+--------+--------+
|          1 | Michael      | Palayam    | 9553798011 |          6 |      1 |  30000 |
|          2 | Justin       | Vanchiyoor | 8351280095 |          5 |      1 |  25000 |
|          3 | Maria        | Thampanoor | 8769611599 |          4 |      1 |  20000 |
|          4 | Rajesh       | Pettah     | 7255480246 |          3 |      1 |  15000 |
|          5 | Jagath       | Kowdiar    | 9971077633 |          2 |      1 |  10000 |
|          6 | Jennifer     | Nedumangad | 7643856016 |          1 |      1 |   5000 |
+------------+--------------+------------+------------+------------+--------+--------+
6 rows in set (0.00 sec)
```



```
mysql> /*
   /*> 19. Write a function to calculate current income to the gym
   /*> */
mysql> DROP FUNCTION IF EXISTS calculate_current_income;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> DELIMITER $$
mysql> CREATE FUNCTION calculate_current_income()
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE amt INT;
    -> DECLARE cnt INT;
    -> DECLARE total INT DEFAULT 0;
    -> DECLARE flag INT DEFAULT 0;
    -> DECLARE cur CURSOR FOR SELECT amount , count(*) FROM membership_plan INNER JOIN member  ON type_name = member_type GROUP BY amount;
    -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET flag = 1;
    -> OPEN cur;
    -> FETCH cur INTO amt,cnt;
    -> WHILE flag < 1 DO
    -> SET total = total + amt*cnt;
    -> FETCH cur INTO amt,cnt;
    -> END WHILE;
    -> CLOSE cur;
    -> RETURN total;
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
mysql> SELECT calculate_current_income();
+----------------------------+
| calculate_current_income() |
+----------------------------+
|                     125000 |
+----------------------------+
1 row in set (0.01 sec)
```

05

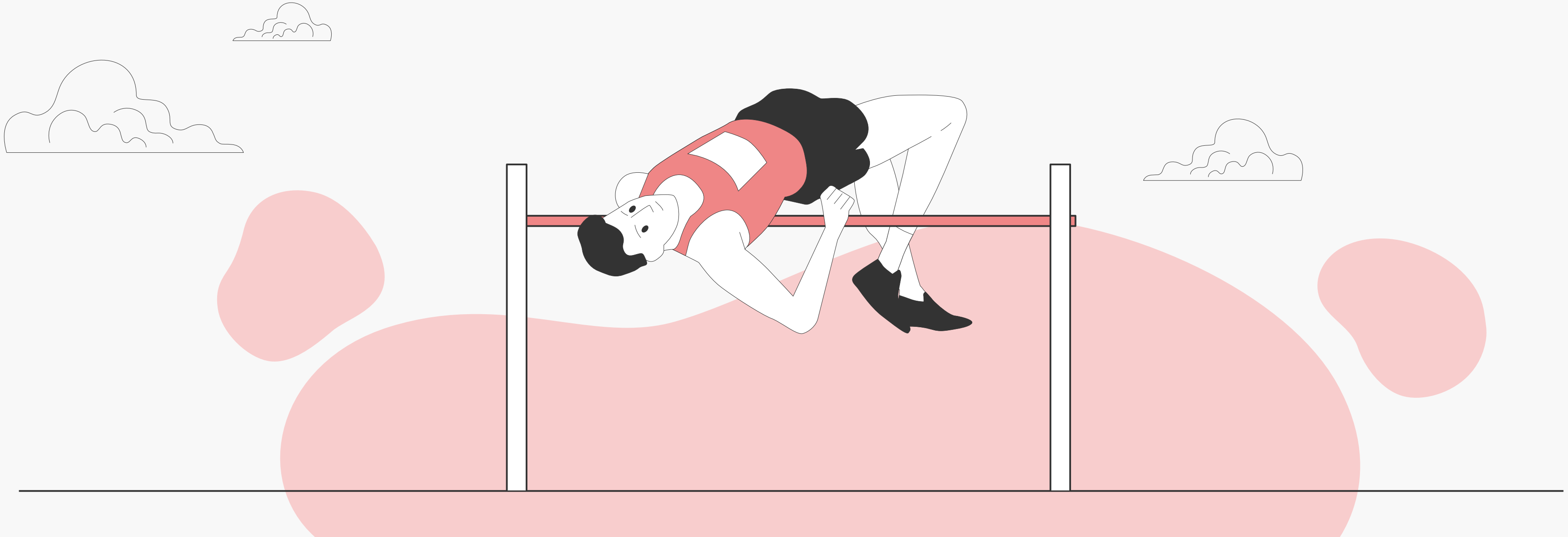
Conclusion

Benefits of database

Database Benefits

1. Robust data analysis capabilities
2. Track individual member progress
3. Systematic approach to collect and manage data
4. Efficiently captures and organizes essential information

In conclusion, our group project, Fitness Data Hub has successfully addressed the challenges and requirements of managing data in a fictional gym through the implementation of a comprehensive database management system (DBMS)



Thank You