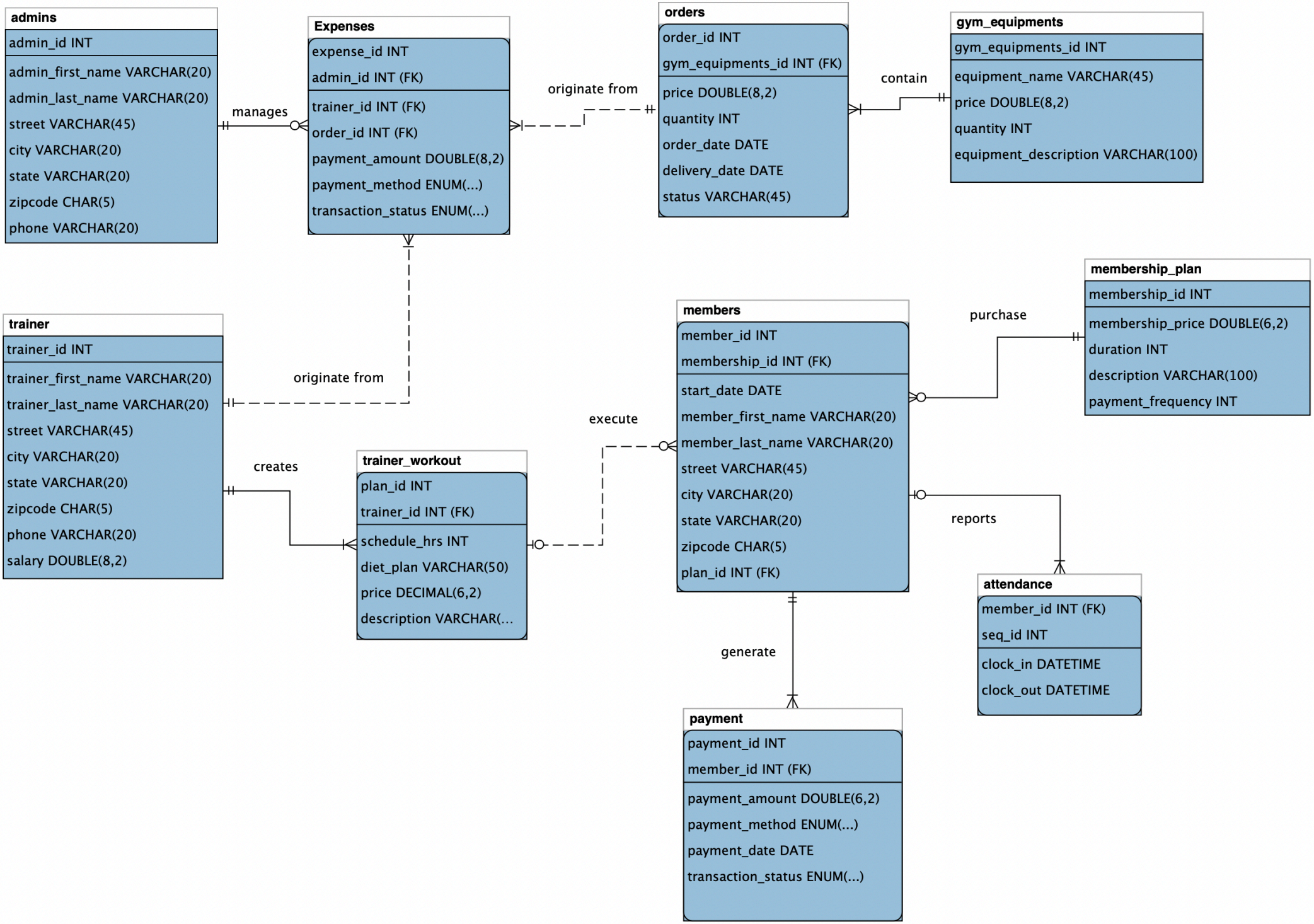
Work-it-Out: Gym Fitness Management System - Milestone 2

Group 4:
Alex Churchill
Justus Ngunjiri
Likhitha Veganti
Kewal Deepak Tadas


admins
admin_id INT
admin_first_name VARCHAR(20)
admin_last_name VARCHAR(20)
street VARCHAR(45)
city VARCHAR(20)
state VARCHAR(20)
zipcode CHAR(5)
phone VARCHAR(20)
manages
Expenses
expense_id INT
admin_id INT (FK)
trainer_id INT (FK)
order_id INT (FK)
payment_amount DOUBLE(8,2)
payment_method ENUM(...)
transaction_status ENUM(...)
originate from
orders
order_id INT
gym_equipments_id INT (FK)
price DOUBLE(8,2)
quantity INT
order_date DATE
delivery_date DATE
status VARCHAR(45)
contain
gym_equipments
gym_equipments_id INT
equipment_name VARCHAR(45)
price DOUBLE(8,2)
quantity INT
equipment_description VARCHAR(100)
trainer
trainer_id INT
trainer_first_name VARCHAR(20)
trainer_last_name VARCHAR(20)
street VARCHAR(45)
city VARCHAR(20)
state VARCHAR(20)
zipcode CHAR(5)
phone VARCHAR(20)
salary DOUBLE(8,2)
originate from
creates
trainer_workout
plan_id INT
trainer_id INT (FK)
schedule_hrs INT
diet_plan VARCHAR(50)
price DECIMAL(6,2)
description VARCHAR(...
execute
members
member_id INT
membership_id INT (FK)
start_date DATE
member_first_name VARCHAR(20)
member_last_name VARCHAR(20)
street VARCHAR(45)
city VARCHAR(20)
state VARCHAR(20)
zipcode CHAR(5)
plan_id INT (FK)
purchase
membership_plan
membership_id INT
membership_price DOUBLE(6,2)
duration INT
description VARCHAR(100)
payment_frequency INT
reports
attendance
member_id INT (FK)
seq_id INT
clock_in DATETIME
clock_out DATETIME
generate
payment
payment_id INT
member_id INT (FK)
payment_amount DOUBLE(6,2)
payment_method ENUM(...)
payment_date DATE
transaction_status ENUM(...)

Set of Relevant Queries

We created a set of 11 SQL database queries that can be used to extract valuable insights and information from the gym database. These queries cover a range of factors, including financial performance, member engagement, trainer workload, and inventory management. By running these queries, gym owners and administrators can gain a better understanding of their business operations and make data-driven decisions to optimize their services and resources. Some of the key insights that can be gained from these queries include identifying high-value members, optimizing trainer workload, managing inventory, and targeting specific demographics for marketing campaigns. Overall, these queries can help gym owners and administrators optimize their operations and improve member engagement and satisfaction. Please note, we framed our queries as questions in order to help readers understand the business question that each query is designed to answer.

1. **How many members are following each diet plan type and how many trainers made each type of plan (keto, diabetic etc.)?**

Relevance: This query shows each diet plan offered by the gym and the number of trainers offering the plan, as well as the number of members following it. By tracking the number of members following each diet plan and the number of trainers who create those plans, gym owners, in collaboration with admins, can optimize resources and tailor members services to meet the needs of the members.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```sql
USE projectml2;

-- 1. How many trainers are associated with each type of diet plan?
-- And How many members are following each type of diet plan? And (keto, diabetic etc.)?
SELECT W.DIET_PLAN,
  COUNT(DISTINCT T.TRAINER_ID) AS NUM_OF_TRAINERS,
  COUNT(DISTINCT M.MEMBER_ID) AS NUM_OF_MEMBERS
FROM
  TRAINER T
  JOIN TRAINER_WORKOUT W
  ON T.TRAINER_ID = W.TRAINER_ID
  LEFT JOIN MEMBERS M
  ON W.PLAN_ID = M.PLAN_ID
GROUP BY W.DIET_PLAN
ORDER BY NUM_OF_TRAINERS DESC, NUM_OF_MEMBERS DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DIET_PLAN	NUM_OF_TRAINERS	NUM_OF_MEMBERS
Low Fat	5	8
High Protein	4	1
Diabetic	3	4
Keto	3	4
High Protein Non Veg	2	3

2. What times of day do most members come to the gym?

Relevance: Owners need to know the most efficient way to allocate staff in order to meet demand while minimizing cost. By analyzing attendance trends, owners can determine the optimal amount of staff needed at any point and schedule their employees accordingly.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```sql
-- 2. What are the top 3 peakest hours?
SELECT EXTRACT(HOUR FROM CLOCK_TIME) AS HOUR_OF_DAY, COUNT(*) AS NUM_OF_VISITS
FROM (
  SELECT CLOCK_IN AS CLOCK_TIME
  FROM ATTENDANCE
  UNION ALL
  SELECT CLOCK_OUT AS CLOCK_TIME
  FROM ATTENDANCE
) AS ALL_VISITS
GROUP BY EXTRACT(HOUR FROM CLOCK_TIME)
ORDER BY NUM_OF_VISITS DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

HOUR_OF_DAY	NUM_OF_VISITS
20	84
9	55
18	39

3. **What are the most commonly used payment methods?**

Relevance: This query shows the top three most popular payment methods members are using to pay the gym. By understanding which payment methods are used most frequently by their members, gym owners can streamline their payment systems, reduce processing costs, and offer more convenient payment options to members.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```sql
-- 3. What are the 3 most commonly used payment methods from both expenses and payments?
SELECT PAYMENT_METHOD, SUM(NUM_TRANSACTIONS) AS TOTAL_TRANSACTIONS
FROM (
  SELECT PAYMENT_METHOD, COUNT(*) AS NUM_TRANSACTIONS
  FROM EXPENSES
  GROUP BY PAYMENT_METHOD
  UNION ALL
  SELECT PAYMENT_METHOD, COUNT(*) AS NUM_TRANSACTIONS
  FROM PAYMENTS
  GROUP BY PAYMENT_METHOD
) AS ALL_TRANSACTIONS
GROUP BY PAYMENT_METHOD
ORDER BY TOTAL_TRANSACTIONS DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

PAYMENT_METHOD	TOTAL_TRANSACTIONS
cheque	33
venmo	29
credit_card	27

4. **How much of each equipment type does the gym currently have?**

Relevance: This query returns a count for each type of equipment the gym currently has on hand. By tracking the quantity of each equipment type, owners can ensure they have enough equipment to meet demand, anticipate replacements, and plan for any expansions.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```sql
-- 4. How much of each equipment type does the gym currently have?
SELECT G.EQUIPMENT_NAME, count(G.EQUIPMENT_NAME) as NUM_OF_EQUIPMENTS
FROM
ORDERS O
INNER JOIN GYM_EQUIPMENTS G
ON O.GYM_EQUIPMENTS_ID = G.GYM_EQUIPMENTS_ID
WHERE O.STATUS = 'Delivered'
GROUP BY G.EQUIPMENT_NAME
ORDER BY NUM_OF_EQUIPMENTS DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EQUIPMENT_NAME	NUM_OF_EQUIPMENTS
Training bench	4
Dumbbell set	4
Rowing machine	3
Treadmill	2
Stationary bicycle	1
Barbell set	1
Low impact treadmills	1
Resistance bands	1

5. **How many members is each trainer currently training?**

Relevance: This query returns each trainer's name and the number of members they are currently working with. By monitoring the number of members assigned to each trainer, owners can suggest new trainers for members, stop assigning overworked trainers, and direct new members to trainers who have openings.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 5.How many members is each trainer currently training?
SELECT T.TRAINER_FIRST_NAME, COUNT(M.MEMBER_ID) AS NUM_OF_MEMBERS
FROM TRAINER T
INNER JOIN TRAINER_WORKOUT W
ON T.TRAINER_ID = W.TRAINER_ID
INNER JOIN MEMBERS M
ON W.PLAN_ID = M.PLAN_ID
GROUP BY T.TRAINER_ID
ORDER BY NUM_OF_MEMBERS DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TRAINER_FIRST_NAME	NUM_OF_MEMBERS
Velvet	7
Karry	4
Shae	4
Gracie	2
Maud	2
Frasquito	1

6. **How many members are signed up for each type of membership plan?**

Relevance: This query returns every membership plan (membership_id) and the number of members who are signed up for that plan. This information can be used to maximize revenue by adjusting the membership offerings based on the popularity of membership plans.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 6. How many members are signed up for each type of membership plan?
SELECT P.MEMBERSHIP_ID, P.DESCRIPTION, COUNT(M.MEMBER_ID) AS NUM_OF_MEMBERS
FROM MEMBERS M
INNER JOIN MEMBERSHIP_PLAN P
ON M.MEMBERSHIP_ID = P.MEMBERSHIP_ID
GROUP BY M.MEMBERSHIP_ID
ORDER BY NUM_OF_MEMBERS DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MEMBERSHIP_ID	DESCRIPTION	NUM_OF_MEMBERS
3	Half yearly membership plan with single payment	4
7	Yearly membership plan with half yearly payment	4
4	Half yearly membership plan with 3 months pay...	3
10	Yearly membership plan with monthly payment	3
1	3 months membership plan with single payment	2
9	Yearly membership plan with 3 months payment	2
2	3 months membership plan with montly payment	1
5	Half yearly membership plan with monthly paym...	1

7. **How does the amount of time members spend in the gym compare to their scheduled hours?**

Relevance: This query provides a side by side comparison of the time each member spent in the gym vs the hours they scheduled so that members can see how they have been keeping up with their fitness goals and make adjustments as needed.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
FROM ATTENDANCE A
INNER JOIN MEMBERS M
ON A.MEMBER_ID = M.MEMBER_ID
INNER JOIN TRAINER_WORKOUT W
ON M.PLAN_ID = W.PLAN_ID
WHERE DATE(A.CLOCK_IN) >= '2023-01-21' AND DATE(A.CLOCK_OUT) <= '2023-01-27'
GROUP BY M.MEMBER_ID
ORDER BY SCHEDULE_HRS DESC, TIME_SPEND DESC;

-- 8. Every rejected transaction and the associated member.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MEMBER_ID	SCHEDULE_HRS	TIME_SPEND
3	20	12
10	19	11
15	18	20
1	18	16
12	18	12
13	18	11
11	18	10
4	18	8
7	16	14
14	16	11
20	16	8
18	11	6
9	9	13
5	9	12
8	4	5
17	3	16
16	3	12
19	3	12
6	3	11
2	3	4

8. **Are there any rejected transactions and which members have not paid the gym?**

Relevance: This query shows all the rejected payment transactions and the associated member. This information will allow owners and administrators to reach out to members who have not paid their membership fees to collect updated payment information and any money owed to the gym.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 8. Every rejected transaction and the associated member.
SELECT M.MEMBER_ID , M.MEMBER_FIRST_NAME, M.MEMBER_LAST_NAME, P.PAYMENT_ID, P.PAYMENT_AMOUNT, P.PAYMENT_DATE
FROM MEMBERS M
INNER JOIN PAYMENTS P
ON M.MEMBER_ID = P.MEMBER_ID
WHERE P.TRANSACTION_STATUS = 'fail'
ORDER BY P.PAYMENT_DATE;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MEMBER_ID	MEMBER_FIRST_NAME	MEMBER_LAST_NAME	PAYMENT_ID	PAYMENT_AMOUNT	PAYMENT_DATE
16	Mozes	Shattock	47	265.00	2022-04-10
6	Jere	Tschursch	48	60.00	2022-04-11
19	Sim	Langham	49	1977.00	2022-04-12
13	Tania	Tammadge	50	346.00	2022-04-13
6	Jere	Tschursch	51	332.00	2022-04-14
7	Bryna	Cosley	52	439.00	2022-04-15
17	Alva	Guerro	53	872.00	2022-04-16
14	Dinnie	Francesconi	54	1077.00	2022-04-17
1	Ameline	Lamas	55	1094.00	2022-04-18
11	Natividad	Pett	28	840.00	2022-07-22

9. What is the ratio of trainers to members?

Relevance : This query shows the number of members divided by the number of trainers. It provides a quick overview of whether or not the gym needs to consider hiring more trainers to keep up with demand.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 9. Ratio of members to trainers
SELECT COUNT(DISTINCT M.MEMBER_ID) / COUNT(DISTINCT T.TRAINER_ID) AS MEMBER_TO_TRAINER_RAT
FROM TRAINER T
JOIN TRAINER_WORKOUT W
ON T.TRAINER_ID = W.TRAINER_ID
JOIN MEMBERS M
ON W.PLAN ID = M.PLAN ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MEMBER_TO_TRAINER_RATIO
3.3333

10. How are expenses split between trainer salaries and equipment purchases?

Relevance: This query will allow owners to see how much money they are spending on trainer salaries and orders. This information will be useful for owners who want a breakdown of their operating expenses for accounting and cash flow purposes.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 10. How are expenses divided between trainers and orders?
SELECT
  SUM(CASE WHEN TRAINER_ID IS NOT NULL THEN PAYMENT_AMOUNT ELSE 0 END) AS TRAINER_EXPENSE,
  SUM(CASE WHEN order_id IS NOT NULL THEN PAYMENT_AMOUNT ELSE 0 END) AS ORDER_EXPENSE
FROM
  EXPENSES;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TRAINER_EXPENSE	ORDER_EXPENSE
19833.69	29556.99

11. What is the total profit to date

Relevance: This query calculates the overall financial performance of the gym over its lifetime by determining the net profit or loss of the gym. This information can be used to review past performance, resource allocation, and business growth strategies by reviewing financial performance.

MILESTONE2DUMP | MILESTONE2QUERY*

Limit to 1000 rows

```
-- 11. Total profit of the gym.(Total income - total expenses)
SELECT
(SELECT SUM(PAYMENT_AMOUNT) FROM PAYMENTS)
 -
 (SELECT SUM(PAYMENT_AMOUNT) FROM EXPENSES) AS PROFIT;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PROFIT
52317.32

Stored Procedure

Relevance: We created a stored procedure named ‘get_member_details‘ which takes a member_id integer as an input and outputs that members first name, the price of their membership, the duration of their membership, their diet plan and their scheduled gym hours. This stored procedure provides a quick method for accessing an overview of a particular member and will be useful in many situations such as accessing a members information when they call in about their membership.

Procedure Details

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_member_details`(IN member_id INT)
BEGIN
    SELECT
        members.member_first_name,
        membership_plan.membership_price,
        membership_plan.duration,
        trainer_workout.diet_plan,
        trainer_workout.schedule_hrs
    FROM
        members
        INNER JOIN membership_plan ON members.membership_id = membership_plan.membership_id
        INNER JOIN trainer_workout ON members.plan_id = trainer_workout.plan_id
    WHERE
        members.member_id = member_id;
END
```

Calling the Procedure

```
call project.get_member_details(4);
```

Output for member_id = 4

member_first_name	membership_price	duration	diet_plan	schedule_hrs
Chrissy	900.00	6	Low Fat	18