



# MYSQL - PROJECT

DATABASE DESIGN FOR HEALTH AND FITNESS TRACKING



# OBJECTIVE



To track fitness details of clients using MySQL, Database for tracking and querying data related to fitness activities, user goals, and various health Nutrition. Below is an example of how you might structure your tables .



## Table structure



### Users:

Stores user information.



**Activities:** Stores different types of activities (e.g., running, cycling) and calories burned .



### Nutrition:

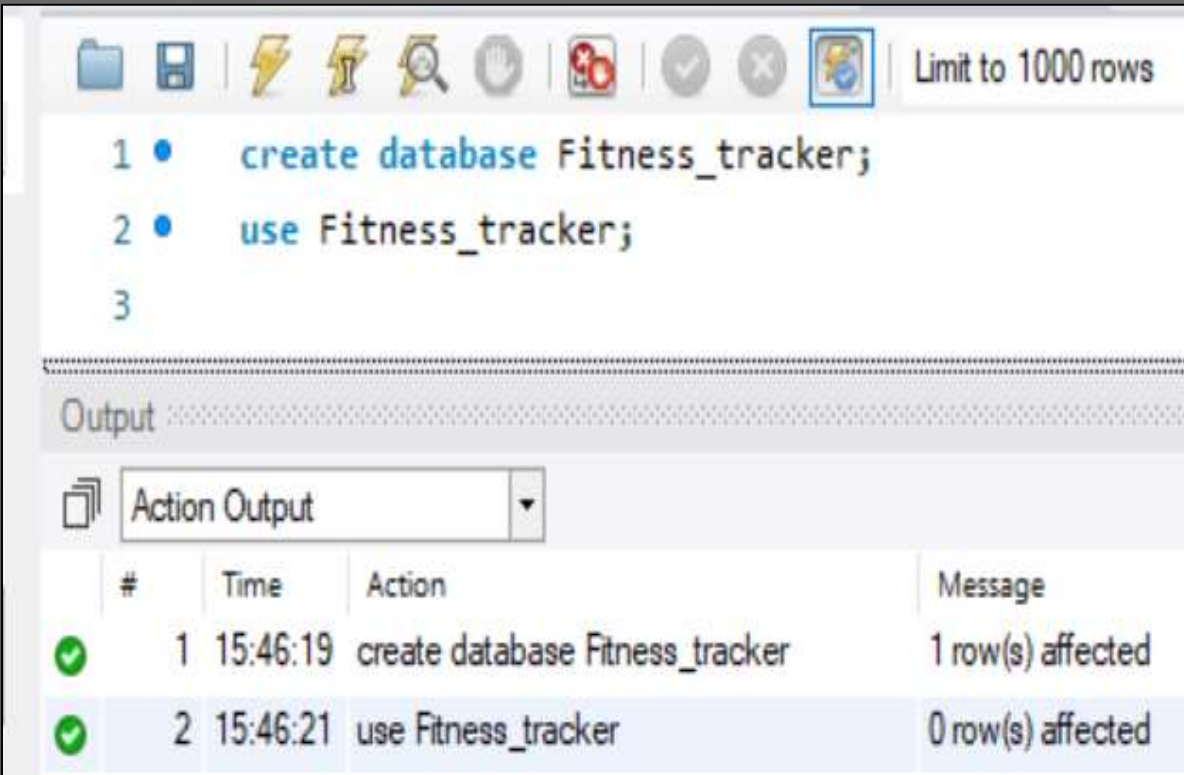
Stores performed by users , details like Meal Type, Food Type and calories .



**Goals:** Stores activities performed by users, including details like steps, active minute, and calories burned.

# Created a database 'Fitness'

## HOW TO CREATE, DROP AND USE A DATABASE IN MYSQL ?



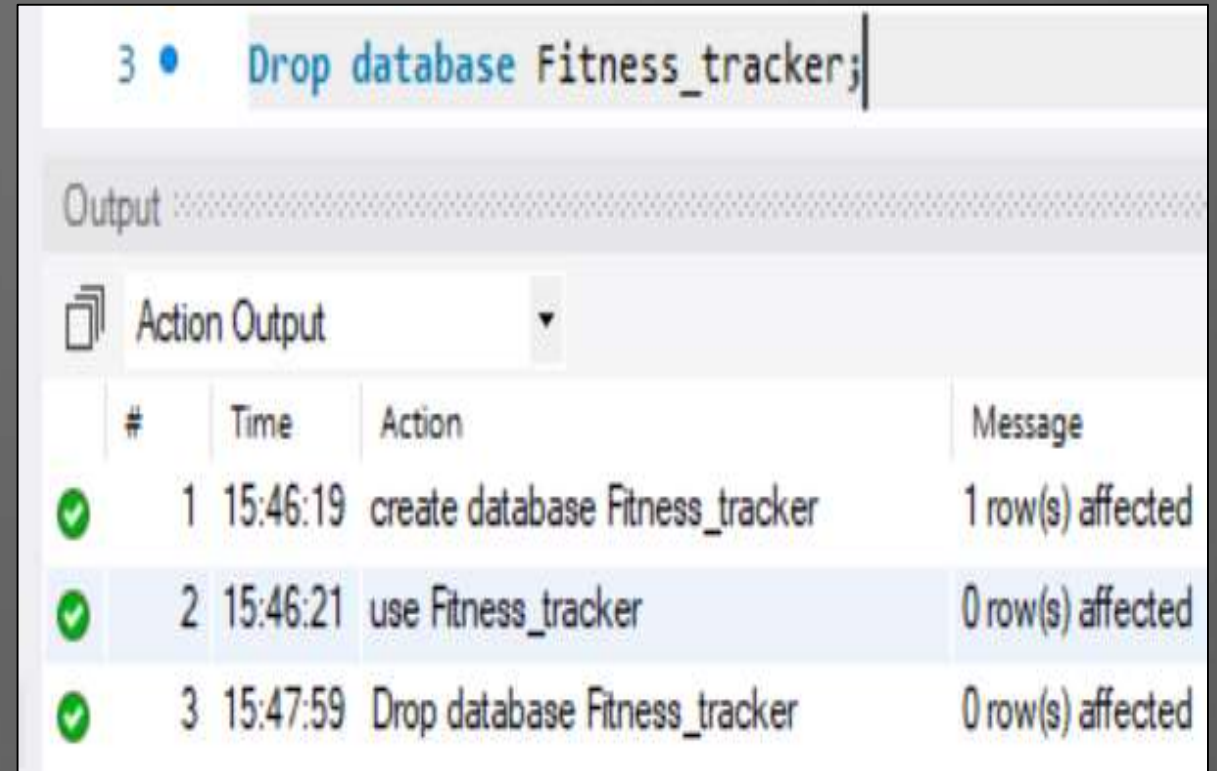
Limit to 1000 rows

```
1 • create database Fitness_tracker;  
2 • use Fitness_tracker;  
3
```

Output

Action Output

#	Time	Action	Message
✓ 1	15:46:19	create database Fitness_tracker	1 row(s) affected
✓ 2	15:46:21	use Fitness_tracker	0 row(s) affected



```
3 • Drop database Fitness_tracker;
```

Output

Action Output

#	Time	Action	Message
✓ 1	15:46:19	create database Fitness_tracker	1 row(s) affected
✓ 2	15:46:21	use Fitness_tracker	0 row(s) affected
✓ 3	15:47:59	Drop database Fitness_tracker	0 row(s) affected



# Created a table User, Activity, Nutrition and goals with primary key and foreign Key

## HOW TO CREATE A TABLE IN FITNESS DATABASE ?

```
5 • Create table users(  
6   user_id int primary key,  
7   username varchar(20) not null,  
8   age int,  
9   gender char(10),  
10  height varchar(10),  
11  weight int  
12 );  
13 • select*from users;
```

	user_id	username	age	gender	height	weight
▶	1	Chelsea	29	Male	158	90kg
	2	Ailey	50	Male	152	67kg
	3	Marietta	46	Male	163	25kg
	4	Olivier	14	Male	175	45kg
	5	Nathanael	16	Male	163	45kg

users 10 x

Output

	#	Time	Action	Message
✓	47	16:34:47	DEALLOCATE PREPARE stmt	OK
✓	48	16:35:05	select*from users LIMIT 0, 1000	50 row(s) returned

```
15 • create table activities03(  
16   activity_id int primary key,  
17   user_id int,  
18   foreign key(user_id) references users1(user_id),  
19   Exercise_Type varchar(20),  
20   weather_conditions varchar(20),  
21   CaloriesBurned int,  
22   Date date  
23 );  
24 • select*from activities;
```

	activity_id	user_id	Exercise_Type	weather_conditions	CaloriesBurned	Date
▶	1	1	aerobics	rainy	200	01-01-2024
	2	1	yoga	snowy	240	02-01-2024
	3	2	yoga	sunny	200	03-01-2024
	4	2	cycling	windy	240	04-01-2024
	5	3	zumba	windy	180	05-01-2024

activities 14 x

Output

	#	Time	Action	Message
✓	18	16:46:17	create table activities03( activity_id int...	0 row(s) affected
✓	19	16:46:26	select*from activities LIMIT 0, 1000	50 row(s) returned

```

26 • create table Nutrition9(
27     Nutritions_ID int primary key,
28     User_ID int,
29     foreign key (user_id) references users1(user_id),
30     MealType varchar(20),
31     FoodItem varchar(20),
32     Calories varchar(20)
33 );
34 • select*from Nutrition;
35

```

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

	Nutrition_id	user_id	meal_type	Food_type	Calories
▶	1	1	lunch	brown rics	150
	2	1	dinner	chicken breast	210
	3	2	snack	chicken breast	270
	4	2	breakfast	Oatmeal with Fruit & Nuts	250
	5	3	snack	salad	300

Nutrition 15 x

Output

Action Output

	#	Time	Action	Message
✓	20	16:47:52	create table Nutrition9( Nutritions_ID i...	0 row(s) affected
✓	21	16:47:56	select*from Nutrition LIMIT 0, 1000	50 row(s) returned

```

36 • create table goals(
37     goal_id int primary key,
38     user_id int,
39     foreign key (user_id) references users1(user_id),
40     Steps int,
41     active_minute float,
42     Date date
43 );
44 • select*from goals;
45

```

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

	Goal_id	user_id	steps	Active_minutes	Date
▶	1	1	16000	75	01-11-2023
	2	1	12000	45	02-11-2023
	3	2	5000	80	03-11-2023
	4	2	13500	55	04-11-2023
	5	3	9000	45	05-11-2023

goals 16 x

Output

Action Output

	#	Time	Action	Message
✓	30	16:50:22	DEALLOCATE PREPARE stmt	OK
✓	31	16:50:29	select*from goals LIMIT 0, 1000	50 row(s) returned



# TO FIND A QUERY TO UPDATE MEAL\_TYPE FOR SPECIFIC USER ?

```
50 • update nutrition set Meal_Type="mid snack",Food_Type="Butter Milk" where user_id=30;
51 • set sql_safe_updates=0;
52 • select*from Nutrition;
53
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	Nutrition_id	user_id	meal_type	Food_type	Calories
	39	29	dinner	Oatmeal with Fruit & Nuts	250
	40	30	mid snack	Butter Milk	190
	41	31	breakfast	chicken breast	270
	42	32	dinner	salad	250
	43	33	breakfast	sweet potato	150

Nutrition 66 x

Output

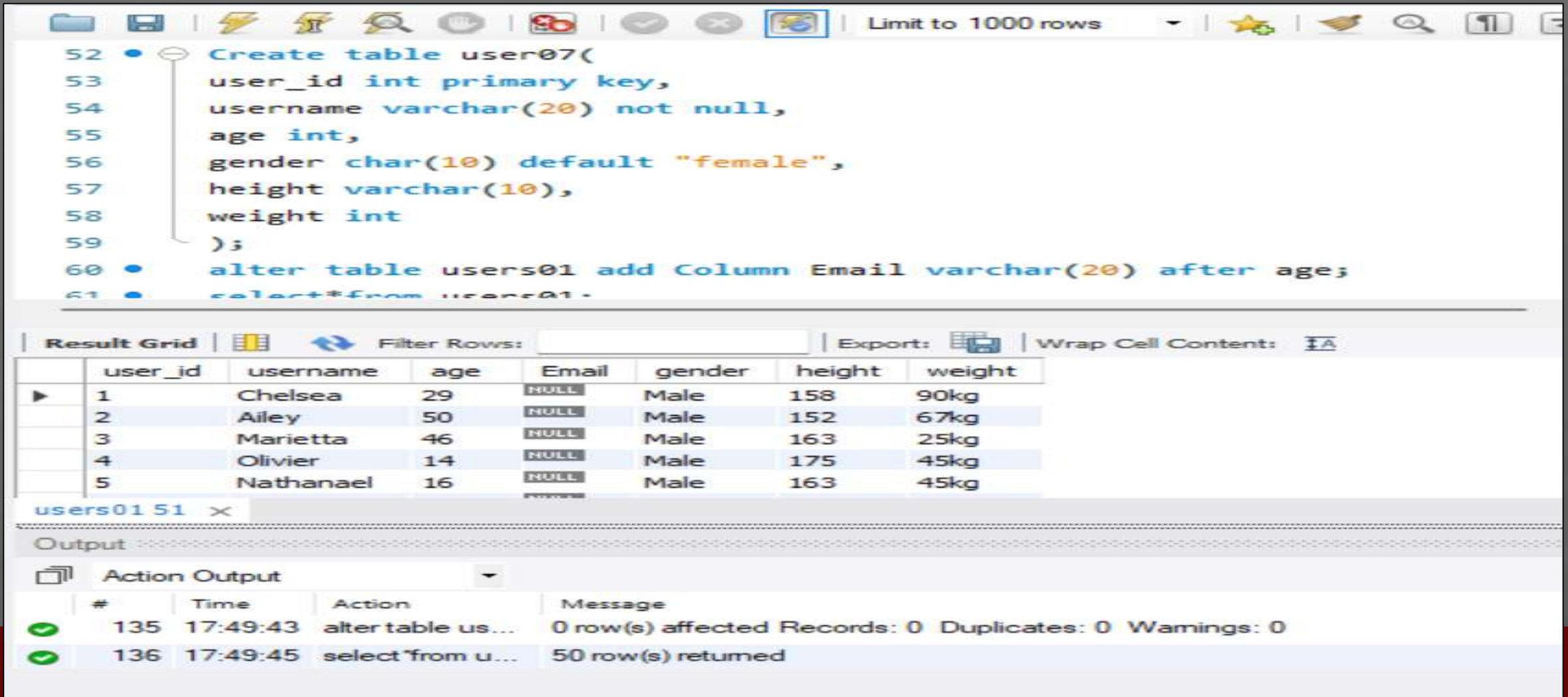


Action Output



	#	Time	Action	Message
✓	205	18:18:58	update nutritio...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓	206	18:19:26	select*from Nu...	50 row(s) returned

## TO FIND A QUERY USING CONSTRAINTS LIKE ADD DEFAULT, ADD COLUMN FROM USERS?



The screenshot displays a database management interface with a SQL editor at the top and a results pane at the bottom. The SQL editor contains three queries: a table creation statement for 'user07', an 'alter table' statement to add an 'Email' column to 'users01', and a 'select \* from users01;' statement. The results pane shows a table with 5 rows of user data. Below the table, the 'Output' section shows two successful actions: altering the table and selecting from it.

```
52 • Create table user07(  
53   user_id int primary key,  
54   username varchar(20) not null,  
55   age int,  
56   gender char(10) default "female",  
57   height varchar(10),  
58   weight int  
59 );  
60 • alter table users01 add Column Email varchar(20) after age;  
61 • select * from users01;
```

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

	user_id	username	age	Email	gender	height	weight
▶	1	Chelsea	29	NULL	Male	158	90kg
	2	Ailey	50	NULL	Male	152	67kg
	3	Marietta	46	NULL	Male	163	25kg
	4	Olivier	14	NULL	Male	175	45kg
	5	Nathanael	16	NULL	Male	163	45kg

users01 51 x

**Output**

Action Output

#	Time	Action	Message
✓ 135	17:49:43	alter table us...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 136	17:49:45	select * from u...	50 row(s) returned

# CALCULATE THE DIFFERENCE IN CALORIES BURNED BETWEEN 2 DAYS ?

```
65
66 • select user_id,(max(caloriesBurned)-min(caloriesBurned))as Calories_difference
67 From Activity
68 where date in ("01-01-2024","05-01-2024")
69 Group By user_id;
70
```

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

	user_id	Calories_difference
▶	1	40
	3	240

Result 96 x

Read Only

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
✓	255	19:36:57	select*From A...	50 row(s) returned	0.000 sec / 0.000 sec
✓	256	19:37:23	select user_id,...	2 row(s) returned	0.000 sec / 0.000 sec



FIND GOALS WHO HAVE EITHER MORE THAN 10,000 STEPS OR MORE THAN 60 ACTIVE MINUTES?

73 •  
74

```
select *from goals
where Steps>10000 or active_minutes >60;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Goal_id	user_id	steps	Active_minutes	Date
▶	1	1	16000	75	01-11-2023
	2	1	12000	45	02-11-2023
	3	2	5000	80	03-11-2023
	4	2	13500	55	04-11-2023
	6	3	10000	90	06-11-2023
	7	3	10000	90	07-11-2023
	11	5	1000	60	11-11-2023

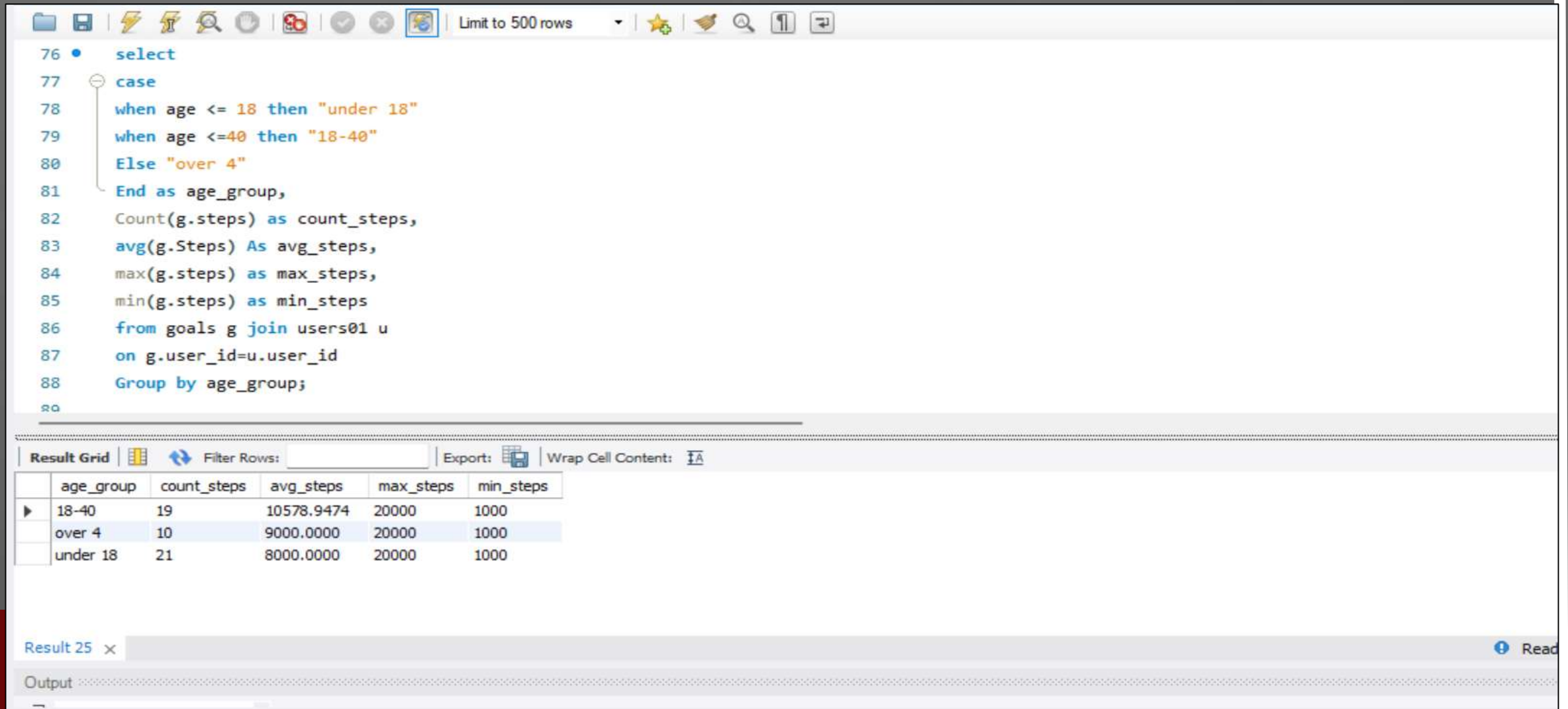
goals 2 ×

Output

Action Output

	#	Time	Action	Message
✓	3	20:13:48	select *from goals where Steps>10000 and active_minutes >60 LIMIT 0, 1000	10 row(s) returned
✓	4	20:14:23	select *from goals where Steps>10000 or active_minutes >60 LIMIT 0, 1000	35 row(s) returned

# CALCULATE AVG,COUNT,MAX,MIN OF THE STEPS USING GROUP BY THEIR AGE GROUP FROM THE USER TABLE ?



The screenshot displays a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 500 rows' dropdown. The SQL editor contains the following query:

```
76 • select
77 case
78 when age <= 18 then "under 18"
79 when age <=40 then "18-40"
80 Else "over 4"
81 End as age_group,
82 Count(g.steps) as count_steps,
83 avg(g.Steps) As avg_steps,
84 max(g.steps) as max_steps,
85 min(g.steps) as min_steps
86 from goals g join users01 u
87 on g.user_id=u.user_id
88 Group by age_group;
89
```

Below the editor, the 'Result Grid' tab is active, showing the query results in a table. The table has five columns: age\_group, count\_steps, avg\_steps, max\_steps, and min\_steps. The results are as follows:

	age_group	count_steps	avg_steps	max_steps	min_steps
▶	18-40	19	10578.9474	20000	1000
	over 4	10	9000.0000	20000	1000
	under 18	21	8000.0000	20000	1000

At the bottom, there is a 'Result 25' tab and an 'Output' section.

# TO FIND A QUERY THE SQRT, POWER, MODULO, EXP USING MATH FUNCTION?

```
90 • select goal_id,user_id,  
91      sqrt(1544)as active_minute_sqrt,  
92      power(50,2) as active_minute_power,  
93      mod(10,16000) as steps_mod_16000,  
94      exp(80) as active_minute_exp  
95 from goals;
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	goal_id	user_id	active_minute_sqrt	active_minute_power	steps_mod_16000	active_minute_exp
▶	1	1	39.293765408777	2500	10	5.54062238439351e34
	2	1	39.293765408777	2500	10	5.54062238439351e34
	3	2	39.293765408777	2500	10	5.54062238439351e34
	4	2	39.293765408777	2500	10	5.54062238439351e34
	5	3	39.293765408777	2500	10	5.54062238439351e34

Result 49 x

 Read On

Output


 Action Output ▼

#	Time	Action	Message	Duration / Fetch
✓ 83	18:43:25	select goal...	50 row(s) returned	0.000 sec / 0.000 sec
✓ 84	18:43:36	select goal...	50 row(s) returned	0.000 sec / 0.000 sec



## TO FIND A QUERY MAX CALORIES BY A USER?

```
99 • Select
100 user_id, max(caloriesBurned) as Exercise
101 from Activity
102 group by user_id having max(caloriesBurned)
103 order by user_id asc;
```

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

	user_id	Exercise
▶	1	240
	2	240
	3	420
	4	420
	5	420

Vertical Output **Result 80** x

Output



Action Output

	#	Time	Action	Message
✓	136	19:42:37	select*From...	50 row(s) returned
✓	137	19:43:04	Select user...	39 row(s) returned

To Find the middle 10 user ?

```
108 • select distinct(user_id) from users01 limit 11,10;  
109
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content

	user_id
▶	12
	13
	14
	15
	16

Vertical Output

Result 85



Output



Action Output



	#	Time	Action	Message
✓	141	19:58:55	select distin...	10 row(s) returned
✓	142	19:59:10	select distin...	10 row(s) returned

# WRITE A QUERY TO USING CONCATENATE ADD A NEW COULMN NAMED “DIET FOOD”?

```
109 • select Nutrition_id, Meal_Type, Food_type,  
110      Concat(Meal_Type, "-", Food_type) as Diet_Food, calories  
111      From nutrition;  
112
```

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

	Nutrition_id	Meal_Type	Food_type	Diet_Food	calories
▶	1	lunch	brown rics	lunch-brown rics	150
	2	dinner	chicken breast	dinner-chicken breast	210
	3	snack	chicken breast	snack-chicken breast	270
	4	breakfast	Oatmeal with Fruit & Nuts	breakfast-Oatmeal with Fruit & Nuts	250
	5	snack	salad	snack-salad	300

**Vertical Output** | **Result 87** x |

**Output**

**Action Output** ▼

	#	Time	Action	Message	Duration / Fetch
✓	148	20:08:30	select Nutrit...	50 row(s) returned	0.031 sec / 0.000 sec
✓	149	20:08:44	select Nutrit...	50 row(s) returned	0.000 sec / 0.000 sec




## TO FIND USERS WHO HAVE TAKE THE MAX STEPS ON THE SPECIFIC DATE “10-11-2023”

```
112
113 • select
114     u.user_id,u.username,g.steps,g.Date
115     from users01 u join goals g
116     on u.user_id=g.user_id
117     where g.date = "10-11-2023" and
118     g.steps < (select max(steps) from goals where date="10-11-2023");
119
120
```

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

	user_id	username	steps	Date
▶	5	Nathanael	1000	10-11-2023

Vertical Output Result 118 x

 Read Only

Output

 Action Output ▼

	#	Time	Action	Message	Duration / Fetch
✓	204	20:45:50	select u.us...	1 row(s) returned	0.000 sec / 0.000 sec
✓	205	20:46:01	select u.us	0 row(s) returned	0.000 sec / 0.000 sec

TO FIND THE QUERY INFORMATION ABOUT ALL USERS, THE WORKOUT PLANS THEY ARE FOLLOWING, THE ACTIVITIES THEY PERFORMED, AND DETAILS ABOUT THOSE ACTIVITIES. SPECIFICALLY FOR 15DAYS?

```
131 • select
132     distinct (u.user_id),u.username,u.age,
133     a.date,a.Exercise_type,
134     n.calories,
135     g.steps,g.minute
136 from users01 u
137 left join activity a on u.user_id=a.user_id
138 left join nutrition n on a.user_id=n.user_id
139 left join goals g on g.user_id=u.user_id
140 WHERE
141     a.Date BETWEEN "01-01-2024" AND "15-01-2024"
142 ORDER BY
143     u.user_id,a.date asc;
```

Result Grid				Filter Rows:	Export:		Wrap Cell Content:	
	user_id	username	age	date	Exercise_type	calories	steps	minute
▶	1	Chelsea	29	01-01-2024	aerobics	210	12000	45
	1	Chelsea	29	01-01-2024	aerobics	210	16000	75
	1	Chelsea	29	01-01-2024	aerobics	150	12000	45
	1	Chelsea	29	01-01-2024	aerobics	150	16000	75
	2	Alley	45	01-01-2024	yoga	250	13500	55

Result 70 

### Output

#### Action Output

**THANK YOU**

