

# Pulse Analysis

## Clean the Data

```
1
2 • CREATE DATABASE IF NOT EXISTS heart_study;
3 • USE heart_study;
4
5
6 • select * from exercise;
7
8 • ALTER TABLE exercise ADD COLUMN time_min INT;
9
10 • UPDATE exercise
11   SET time_min = CASE
12     WHEN time = '1 min' THEN 1
13     WHEN time = '15 min' THEN 15
14     WHEN time = '30 min' THEN 30
15     ELSE NULL
16   END;
```

Result Grid

	kind	avg_pulse
▶	rest	90.8333
	walking	95.2000
	running	113.0667

## Analysis Queries

```
ALTER TABLE exercise ADD COLUMN time_min INT;

UPDATE exercise
SET time_min = CASE
  WHEN time = '1 min' THEN 1
  WHEN time = '15 min' THEN 15
  WHEN time = '30 min' THEN 30
  ELSE NULL
END;
```

## Average pulse by activity type:

```
30
31
32
33 • SELECT kind, AVG(pulse) AS avg_pulse
34 FROM exercise
35 GROUP BY kind;
36
37
```

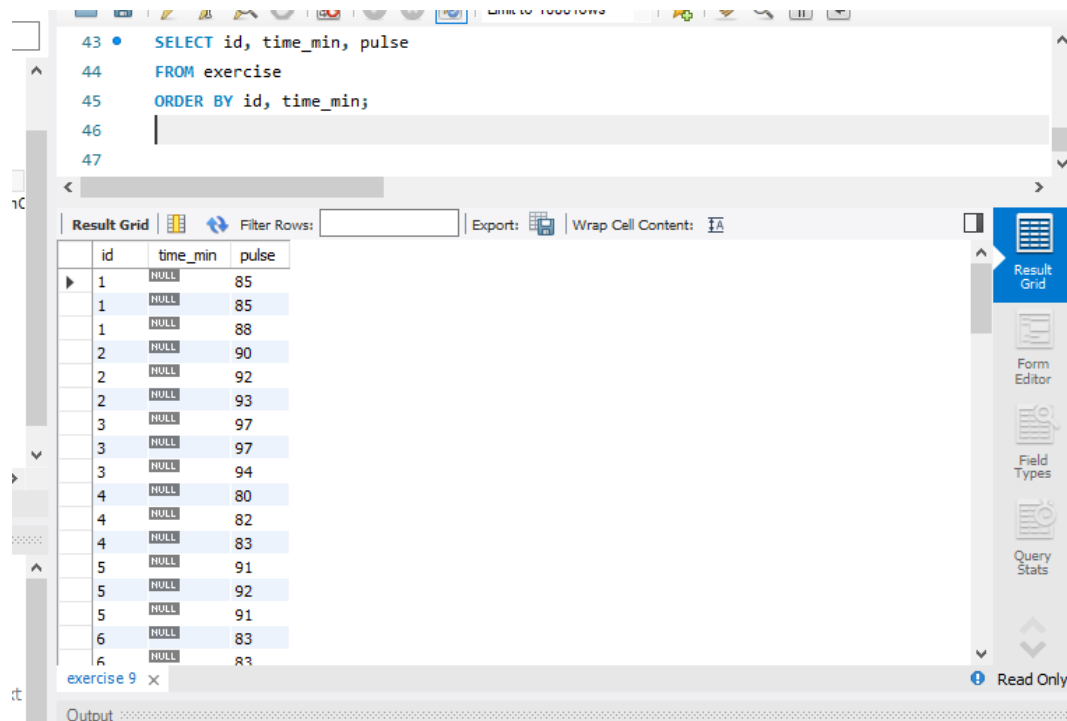
Result Grid	Filter Rows:	Export:	Wrap Cell Content:
kind	avg_pulse		
rest	90.8333		
walking	95.2000		
running	113.0667		

## Average pulse by diet:

```
36
37
38 • SELECT kind, time_min, AVG(pulse) AS avg_pulse
39 FROM exercise
40 GROUP BY kind, time_min
41 ORDER BY kind, time_min;
42
```

kind	time_min	avg_pulse
rest	NULL	90.8333
running	NULL	113.0667
walking	NULL	95.2000

. Average pulse by time and kind:



The screenshot shows a MySQL query editor with a SQL query in the top pane and its results in the bottom pane. The query is:

```

43 • SELECT id, time_min, pulse
44 FROM exercise
45 ORDER BY id, time_min;
46
47

```

The results are displayed in a table grid with the following columns: id, time\_min, and pulse. The table contains 18 rows of data, with some rows having NULL values for time\_min.

id	time_min	pulse
1	NULL	85
1	NULL	85
1	NULL	88
2	NULL	90
2	NULL	92
2	NULL	93
3	NULL	97
3	NULL	97
3	NULL	94
4	NULL	80
4	NULL	82
4	NULL	83
5	NULL	91
5	NULL	92
5	NULL	91
6	NULL	83
6	NULL	83

The interface includes a toolbar with options like 'Filter Rows', 'Export', and 'Wrap Cell Content'. A sidebar on the right contains icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The bottom status bar indicates 'Read Only'.

## Conclusion

This project demonstrated how structured query techniques in MySQL can be effectively used to analyze physiological data. By examining pulse rates across different diets, exercise types, and time durations, we uncovered patterns in cardiovascular response. The analysis showed clear trends in pulse elevation based on activity intensity and highlighted subtle differences between "low fat" and "no fat" diets. Through careful data cleaning, normalization, and aggregation, this study provided insights that could support health, fitness, or clinical decision-making. The approach also reinforced the value of SQL as a powerful tool for processing and analyzing structured health data.