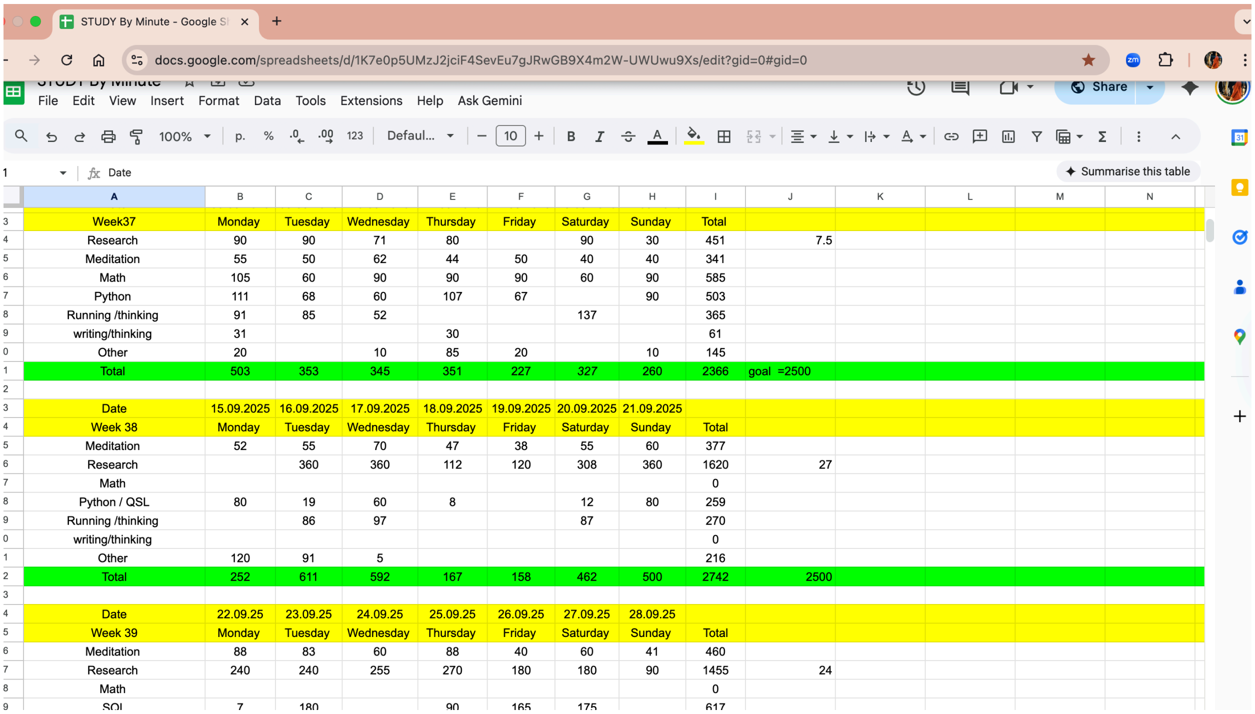
Study minutes project

I track my study or learning periods every day to see my progress visually.

I decided to visualise more efficiently using MySQL.

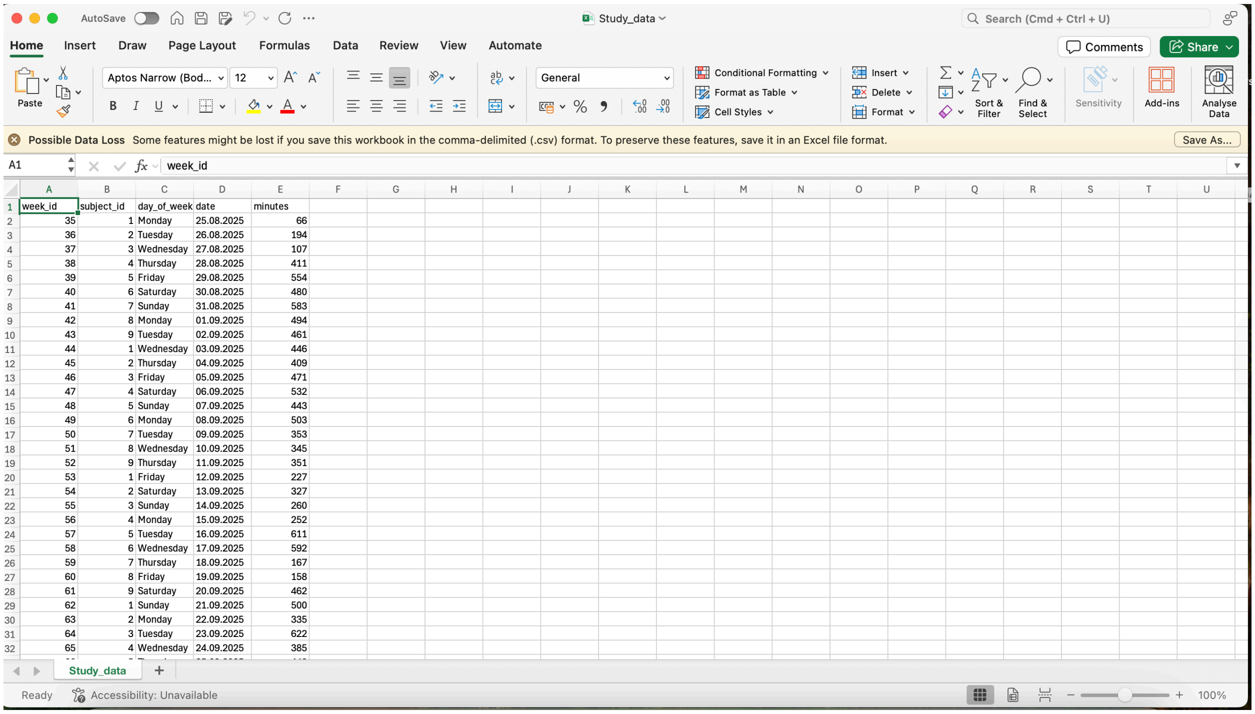
Initial Excel view of the table



STUDY By Minute - Google S
docs.google.com/spreadsheets/d/1K7e0p5UMzJ2jciF4SevEu7gJRwGB9X4m2W-UWUwu9Xs/edit?gid=0#gid=0
Share
File Edit View Insert Format Data Tools Extensions Help Ask Gemini
100%
Defaul...
10
Date
Summarise this table

A	B	C	D	E	F	G	H	I	J
Week37	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total	
Research	90	90	71	80		90	30	451	7.5
Meditation	55	50	62	44	50	40	40	341	
Math	105	60	90	90	90	60	90	585	
Python	111	68	60	107	67		90	503	
Running /thinking	91	85	52			137		365	
writing/thinking	31			30				61	
Other	20		10	85	20		10	145	
Total	503	353	345	351	227	327	260	2366	goal =2500
Date	15.09.2025	16.09.2025	17.09.2025	18.09.2025	19.09.2025	20.09.2025	21.09.2025		
Week 38	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total	
Meditation	52	55	70	47	38	55	60	377	
Research		360	360	112	120	308	360	1620	27
Math								0	
Python / QSL	80	19	60	8		12	80	259	
Running /thinking		86	97			87		270	
writing/thinking								0	
Other	120	91	5					216	
Total	252	611	592	167	158	462	500	2742	2500
Date	22.09.25	23.09.25	24.09.25	25.09.25	26.09.25	27.09.25	28.09.25		
Week 39	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total	
Meditation	88	83	60	88	40	60	41	460	
Research	240	240	255	270	180	180	90	1455	24
Math								0	
SQL	7	180		90	165	175		617	

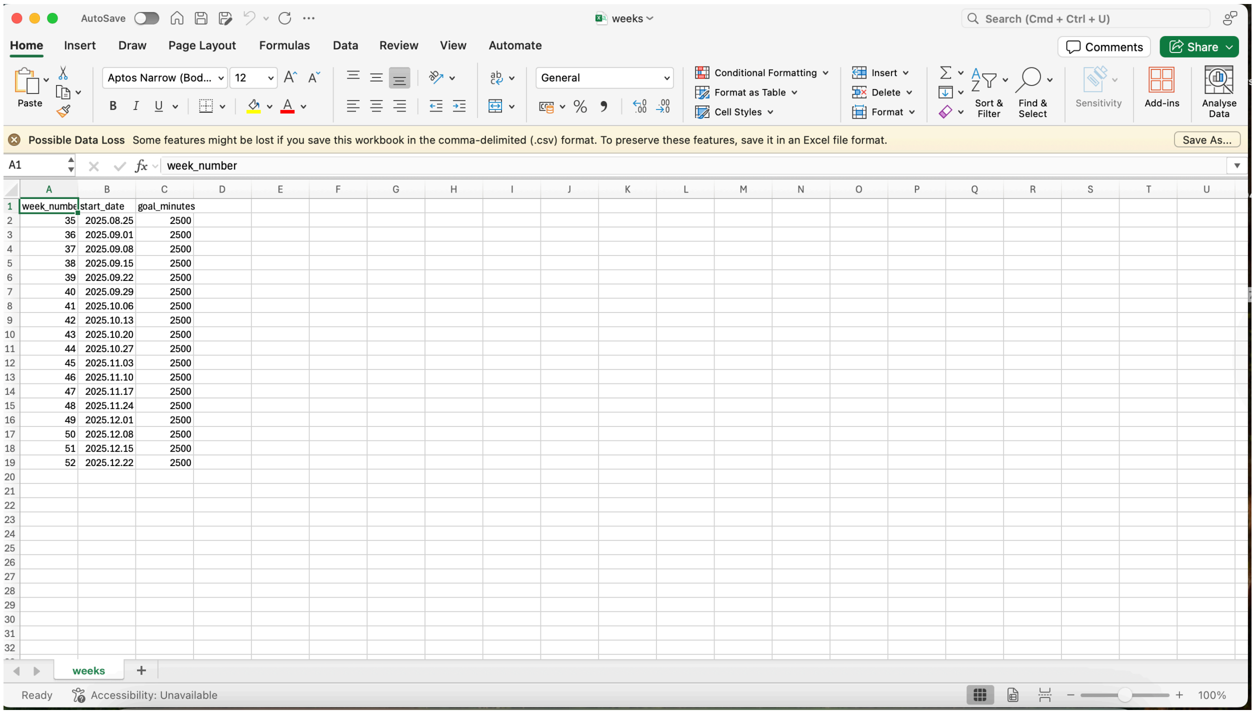
Then I turned it into 3 separate tabels



Possible Data Loss Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 week_id

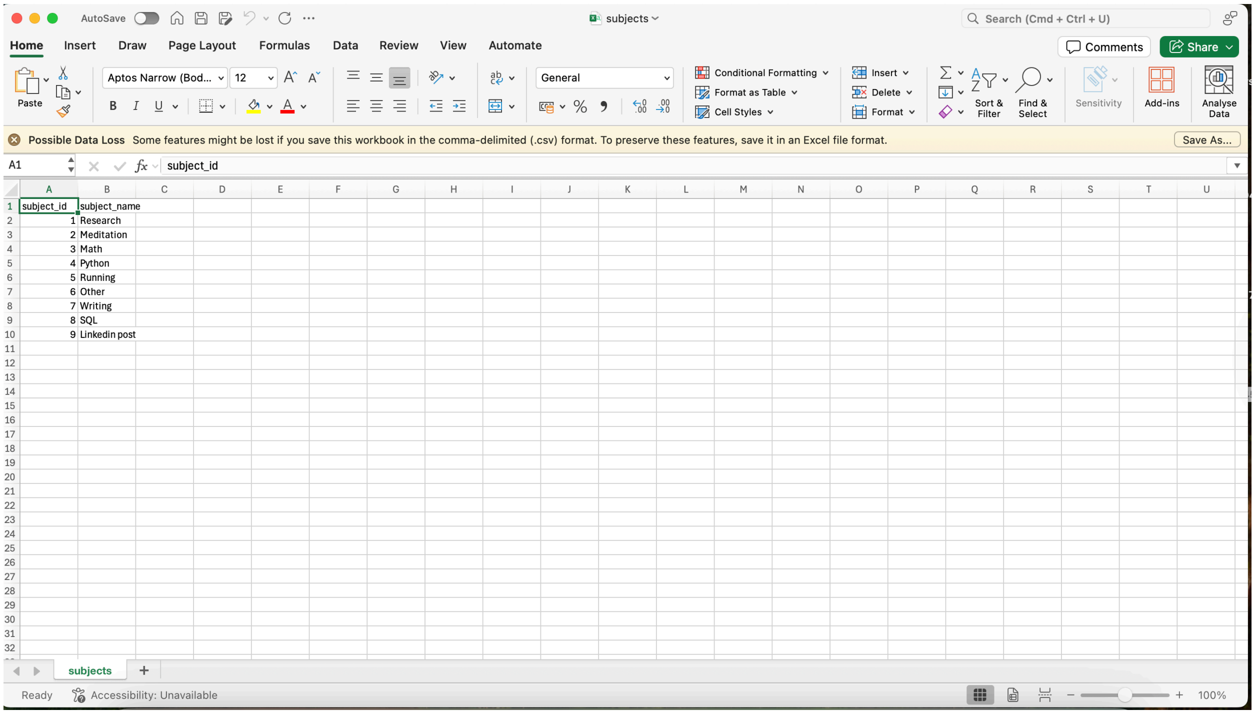
week_id	subject_id	day_of_week	date	minutes
35	1	Monday	25.08.2025	66
36	2	Tuesday	26.08.2025	194
37	3	Wednesday	27.08.2025	107
38	4	Thursday	28.08.2025	411
39	5	Friday	29.08.2025	554
40	6	Saturday	30.08.2025	480
41	7	Sunday	31.08.2025	583
42	8	Monday	01.09.2025	494
43	9	Tuesday	02.09.2025	461
44	1	Wednesday	03.09.2025	446
45	2	Thursday	04.09.2025	409
46	3	Friday	05.09.2025	471
47	4	Saturday	06.09.2025	532
48	5	Sunday	07.09.2025	443
49	6	Monday	08.09.2025	503
50	7	Tuesday	09.09.2025	353
51	8	Wednesday	10.09.2025	345
52	9	Thursday	11.09.2025	351
53	1	Friday	12.09.2025	227
54	2	Saturday	13.09.2025	327
55	3	Sunday	14.09.2025	260
56	4	Monday	15.09.2025	252
57	5	Tuesday	16.09.2025	611
58	6	Wednesday	17.09.2025	592
59	7	Thursday	18.09.2025	167
60	8	Friday	19.09.2025	158
61	9	Saturday	20.09.2025	462
62	1	Sunday	21.09.2025	500
63	2	Monday	22.09.2025	335
64	3	Tuesday	23.09.2025	622
65	4	Wednesday	24.09.2025	385



Possible Data Loss Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 week_number

week_numbe	start_date	goal_minutes
35	2025.08.25	2500
36	2025.09.01	2500
37	2025.09.08	2500
38	2025.09.15	2500
39	2025.09.22	2500
40	2025.09.29	2500
41	2025.10.06	2500
42	2025.10.13	2500
43	2025.10.20	2500
44	2025.10.27	2500
45	2025.11.03	2500
46	2025.11.10	2500
47	2025.11.17	2500
48	2025.11.24	2500
49	2025.12.01	2500
50	2025.12.08	2500
51	2025.12.15	2500
52	2025.12.22	2500

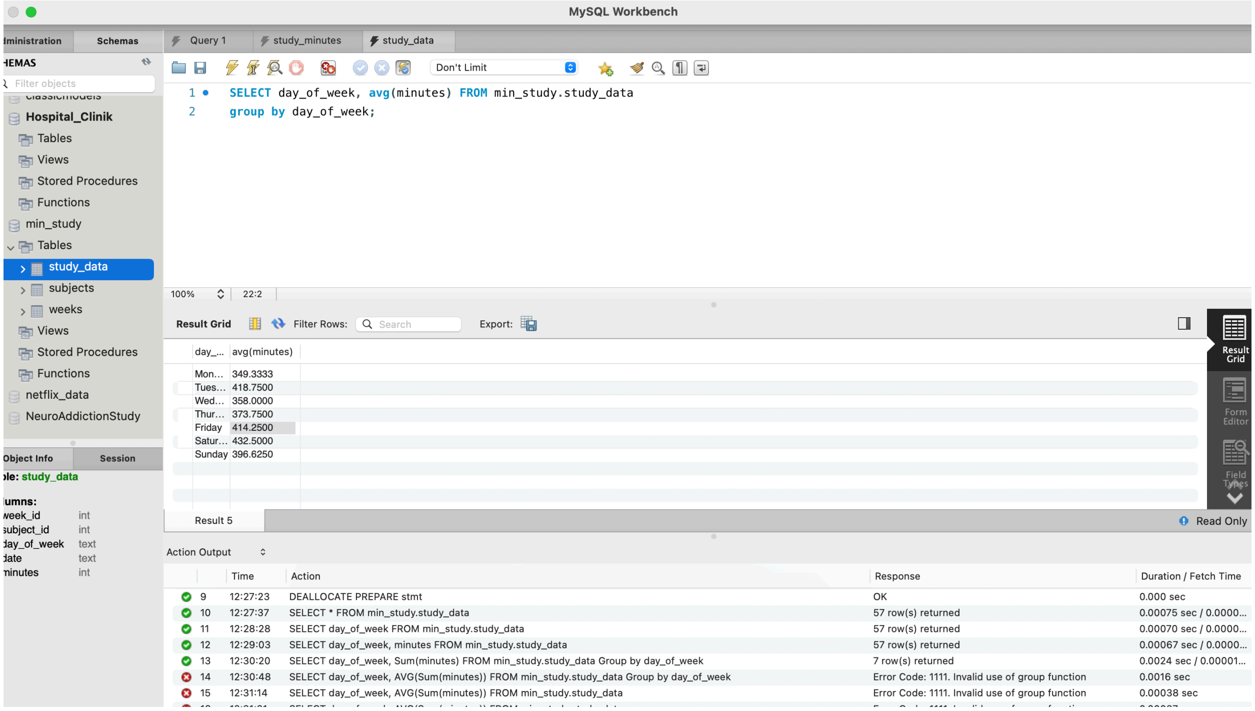


Possible Data Loss Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 subject_id

subject_id	subject_name
1	Research
2	Meditation
3	Math
4	Python
5	Running
6	Other
7	Writing
8	SQL
9	Linkedin post

After creating the schema, I created tables and column names and then imported the data into MySQL. Then I queried to find the average study minutes by day of the week. As it is shown in the picture, the most productive days of the week can be spotted



```sql
SELECT day_of_week, avg(minutes) FROM min_study.study_data
group by day_of_week;
```

ble: study_data

lumns:

week_id	int
subject_id	int
day_of_week	text
date	text
minutes	int

day_...	avg(minutes)
Mon...	349.3333
Tues...	418.7500
Wed...	358.0000
Thur...	373.7500
Friday	414.2500
Satur...	432.5000
Sunday	396.6250

```
    Time      Action                                                                        Response                                        Duration / Fetch Time
✓ 9   12:27:23  DEALLOCATE PREPARE stmt                                                       OK                                              0.000 sec
✓ 10  12:27:37  SELECT * FROM min_study.study_data                                            57 row(s) returned                              0.00075 sec / 0.0000...
✓ 11  12:28:28  SELECT day_of_week FROM min_study.study_data                                  57 row(s) returned                              0.00070 sec / 0.0000...
✓ 12  12:29:03  SELECT day_of_week, minutes FROM min_study.study_data                         57 row(s) returned                              0.00067 sec / 0.0000...
✓ 13  12:30:20  SELECT day_of_week, Sum(minutes) FROM min_study.study_data Group by day_of_week   7 row(s) returned                           0.0024 sec / 0.00001...
✗ 14  12:30:48  SELECT day_of_week, AVG(Sum(minutes)) FROM min_study.study_data Group by day_of_week   Error Code: 1111. Invalid use of group function   0.0016 sec
✗ 15  12:31:14  SELECT day_of_week, AVG(Sum(minutes)) FROM min_study.study_data               Error Code: 1111. Invalid use of group function   0.00038 sec
```

Although it was a relatively easy task, it was fun to see how in a real-life situation, Data Analytics can be used.