

STRAVA FITNESS — SQL Analysis Report

Tool Used: DB Browser for SQLite

Submitted by: Ashwin kanth Marapally

Section 1: Activity Trends

Query 1: Average Steps per Day

SQL Query:

```
SELECT
    ActivityDate,
    ROUND(AVG(TotalSteps),2) AS avg_steps
FROM daily_activity
GROUP BY ActivityDate
ORDER BY ActivityDate;
```

Result Table:

ActivityDate	avg_steps
2016-04-12	8756.59
2016-04-13	7663.16
2016-04-14	8243.16
2016-04-15	7533.85
2016-04-16	9080.47
2016-04-17	7217.43
2016-04-18	8151.71
2016-04-19	9198.46
2016-04-20	8707.17
2016-04-21	9421.25
2016-04-22	7686.58
2016-04-23	8904.13
2016-04-24	7887.37
2016-04-25	9066.04
2016-04-26	8644.41
2016-04-27	8914.34
2016-04-28	7838.58
2016-04-29	8367.46
2016-04-30	9240.21
2016-05-01	7388.21

2016-05-02	7537.12
2016-05-03	8747.7
2016-05-04	7406.19
2016-05-05	9738.46
2016-05-06	7760.25
2016-05-07	7976.38
2016-05-08	8275.39
2016-05-09	9279.92
2016-05-10	8988.57
2016-05-11	8203.09
2016-05-12	3848.89

Query 2: Total Calories Burned per Day

SQL Query:

```
SELECT
    ActivityDay,
    SUM(Calories) AS total_calories
FROM daily_calories
GROUP BY ActivityDay
ORDER BY ActivityDay;
```

Result Table:

ActivityDay	total_calories
2016-04-12	78893
2016-04-13	75459
2016-04-14	77761
2016-04-15	77721
2016-04-16	76574
2016-04-17	71391
2016-04-18	74668
2016-04-19	75491
2016-04-20	76647
2016-04-21	77500
2016-04-22	74485
2016-04-23	76709
2016-04-24	73326
2016-04-25	75186
2016-04-26	74604

2016-04-27	74514
2016-04-28	74114
2016-04-29	72722
2016-04-30	73592
2016-05-01	66913
2016-05-02	65988
2016-05-03	71163
2016-05-04	66211
2016-05-05	70037
2016-05-06	68877
2016-05-07	65141
2016-05-08	62193
2016-05-09	63063
2016-05-10	57963
2016-05-11	52562
2016-05-12	23925

Query 3: Active vs. Sedentary Minutes

SQL Query:

```
SELECT
    ActivityDate,
    SUM(VeryActiveMinutes + FairlyActiveMinutes +
    LightlyActiveMinutes) AS active_minutes,
    SedentaryMinutes
FROM daily_activity
GROUP BY ActivityDate
ORDER BY ActivityDate;
```

Result Table:

ActivityDate	active_minutes	SedentryMinutes
2016-04-12	7562	728
2016-04-13	7018	776
2016-04-14	7733	1218
2016-04-15	8016	726
2016-04-16	7577	773
2016-04-17	6275	539
2016-04-18	7322	1149
2016-04-19	7669	775
2016-04-20	7889	818

2016-04-21	7182	838
2016-04-22	7463	1217
2016-04-23	8535	732
2016-04-24	7074	709
2016-04-25	7445	814
2016-04-26	7606	833
2016-04-27	7424	1108
2016-04-28	7647	782
2016-04-29	7527	815
2016-04-30	7916	712
2016-05-01	5958	730
2016-05-02	6266	798
2016-05-03	7050	816
2016-05-04	5942	1179
2016-05-05	7098	857
2016-05-06	6776	754
2016-05-07	6261	833
2016-05-08	5920	574
2016-05-09	6467	835
2016-05-10	5777	746
2016-05-11	5287	669
2016-05-12	2208	890

Section 2: Sleep Patterns

Query 4: Average Sleep Duration per User

SQL Query:

```
SELECT
    Id,
    ROUND(AVG(TotalMinutesAsleep),2) AS avg_sleep_minutes
FROM sleep_day
GROUP BY Id
ORDER BY avg_sleep_minutes DESC;
```

Result Table:

Id	avg_sleep_minutes
1844505072	652.0
2026352035	506.18

6117666160	478.78
4319703577	476.65
5553957443	463.48
7086361926	453.13
6962181067	448.0
2347167796	446.8
8378563200	445.13
8792009665	435.67
5577150313	432.0
4702921684	417.48
1927972279	417.0
4388161847	400.17
4445114986	385.18
1503960366	360.28
6775888955	349.67
4020332650	349.38
8053475328	297.0
1644430081	294.0
3977333714	293.64
4558609924	127.6
7007744171	68.5
2320127002	61.0

Query 5: Average Sleep Duration by Weekday

SQL Query:

```
SELECT
    strftime('%w', SleepDate) AS weekday,
    ROUND(AVG(TotalMinutesAsleep),2) AS avg_sleep_minutes
FROM sleep_day
GROUP BY weekday
ORDER BY weekday;
```

Result Table:

weekday	avg_sleep_minutes
0	452.75
1	419.5
2	404.54
3	434.68

4	401.3
5	405.42
6	419.07

Section 3: Steps vs Calories

Query 6: Steps vs Calories

SQL Query:

```
SELECT
    TotalSteps,
    Calories
FROM daily_activity
ORDER BY ActivityDate;
```

Note: Full result contains 861 rows. Only top 10 shown below for illustration .

Result Table:

TotalSteps	Calories
13162	1985
8163	1432
10694	3199
6697	2030
678	2220
11875	2390
4414	1459
10725	2124
10113	2344
8796	1982

Query 7: Top 5 Active Users (Steps & Calories)

SQL Query:

```
SELECT
    Id,
    ROUND(AVG(TotalSteps),2) AS avg_steps,
    ROUND(AVG(Calories),2) AS avg_calories
FROM daily_activity
GROUP BY Id
ORDER BY avg_steps DESC
```

```
LIMIT 5;
```

Result Table:

Id	avg_steps	Avg_calories
8877689391	16040.03	3420.26
8053475328	14763.29	2945.81
1503960366	12520.63	1876.97
7007744171	11776.36	2583.48
2022484408	11370.65	2509.97

Section 4: Weight Trends

Query 8: Weight Trend Over Time

SQL Query:

```
SELECT
    Id,
    WeightDate,
    ROUND(WeightKg,2) As WeightKg
FROM weight_log
ORDER BY Id, WeightDate;
```

Note: Only first 10 rows displayed here. Full query returned 67 rows.

Result Table:

Id	WeightDate	WeightKg
1503960366	2016-05-02 23:59:59	52.6
1503960366	2016-05-03 23:59:59	52.6
1927972279	2016-04-13 01:08:52	133.5
2873212765	2016-04-21 23:59:59	56.7
2873212765	2016-05-12 23:59:59	57.3
4319703577	2016-04-17 23:59:59	72.4
4319703577	2016-05-04 23:59:59	72.3
4558609924	2016-04-18 23:59:59	69.7
4558609924	2016-04-25 23:59:59	70.3
4558609924	2016-05-01 23:59:59	69.9

Section 5: Hourly Activity

Query 9: Most Active Hour of the Day

SQL Query:

```
SELECT
    strftime('%H', ActivityDateTime) AS hour_of_day,
    SUM(StepTotal) AS total_steps
FROM hourly_steps
GROUP BY hour_of_day
ORDER BY total_steps DESC;
```

Result Table:

hour_of_day	total_steps
18	542848
19	528552
12	505848
17	498511
14	497813
13	495220
16	450639
10	447467
11	423534
09	403404
08	398044
15	371782
20	320638
07	284932
21	278865
22	215141
06	166191
23	110286
05	40886
00	39404
01	21555
02	15964
04	11836
03	5996

Query 10: Avg Calories Burned per Hour

SQL Query:

```
SELECT
    strftime('%H', ActivityDateTime) AS hour_of_day,
    ROUND(AVG(Calories),2) AS avg_calories
FROM hourly_calories
GROUP BY hour_of_day
ORDER BY hour_of_day;
```

Result Table:

hour_of_day	avg_calories
00	71.81
01	70.17
02	69.19
03	67.54
04	68.26
05	81.71
06	87.0
07	94.48
08	103.34
09	106.14
10	110.46
11	109.81
12	117.2
13	115.31
14	115.73
15	106.64
16	113.33
17	122.75
18	123.49
19	121.48
20	102.36
21	96.06
22	88.27
23	77.59

Section 6: Summary Insights

Query 11: Top 5 Most Active Users

SQL Query:

```
SELECT
    Id,
    ROUND(AVG(TotalSteps),2) AS avg_steps
FROM daily_activity
GROUP BY Id
ORDER BY avg_steps DESC
LIMIT 5;
```

Result Table:

Id	avg_steps
8877689391	16040.03
8053475328	14763.29
1503960366	12520.63
7007744171	11776.36
2022484408	11370.65

Query 12: Top 5 Best Sleepers

SQL Query:

```
SELECT
    Id,
    ROUND(AVG(TotalMinutesAsleep),2) AS avg_sleep_minutes
FROM sleep_day
GROUP BY Id
ORDER BY avg_sleep_minutes DESC
LIMIT 5;
```

Result Table:

Id	Avg_sleep_minutes
1844505072	652.0
2026352035	506.18
6117666160	478.78
4319703577	476.65
5553957443	463.48

Section 7: Business Impact and Conclusion

1. User Activity is Inconsistent (Query 1, 3, 11)

Some users walk a lot while others remain sedentary. This suggests Strava should:

- Promote daily step goals, gamification, and push notifications
- Personalize activity nudges based on user behaviour

2. Calories Burned Directly Follows Steps (Query 2, 6, 7)

Steps and calories are strongly correlated, confirming device tracking accuracy. This supports:

- Encouraging users to log diverse workout types for holistic tracking
- Creating smarter calorie feedback loops in the app

3. Sleep Patterns Vary by Weekday (Query 4, 5, 12)

Users tend to sleep more on weekends. Strava can:

- Send weekday sleep reminders
- Educate users on sleep hygiene
- Offer sleep coaching features

4. Hourly Data Highlights Peak Engagement Times (Query 9, 10)

Most activity and calorie burn happen during certain hours. Strava can:

- Push notifications during low-activity hours to drive mid-day movement
- Tailor in-app content by time of day

5. Weight Data Shows Stability (Query 8)

Users tracking weight show stable trends. This means:

- Weight logs are meaningful for long-term habit building
- Opportunity to reinforce tracking with progress visuals and milestones

6. Top Performing Users Identified (Query 11, 12)

Strava can:

- Recognize top users as brand advocates
- Target low-engagement users with encouragement campaigns

Conclusion:

The SQL analysis provided essential visibility into user activity, calorie burn, sleep behaviour, and engagement trends using raw smart device data. These insights form a foundational layer for Strava's:

- Marketing Strategy → personalized campaigns, nudges, challenges
- Product Innovation → smart recommendations, habit building tools
- User Retention → progress tracking, engagement triggers

With data-backed evidence, Strava can now enhance the app experience, boost user wellness, and drive deeper engagement across its platform.