SQL Data Preparation and Cleaning Report

PROJECT TITLE: STRAVA FITNESS DATA ANALYSIS

OBJECTIVE

Cleaning, normalising, and standardising the raw statistics gathered from different fitness trackers (such as steps, sleep, calories, etc.) in order to get them ready for visual analytics in Power BI and Python was the aim of this SQL phase. I completed necessary data cleaning activities in the SQLite database to get the Fitbit datasets ready for analysis. This required adding many CSV files to the database, including hourlyCalories_merged, sleepDay_merged, and dailyActivity_merged. I handled missing data, made sure that column formats were consistent (particularly for date and time), and eliminated duplicate entries using GROUP BY and HAVING clauses. I converted datetime fields to normal date formats as necessary. Additionally, I used common keys like Id and ActivityDate to confirm the links across tables. These procedures made sure the data was accurate and dependable for further Power BI and Python analysis.

Tools Used: SQLite Workbench

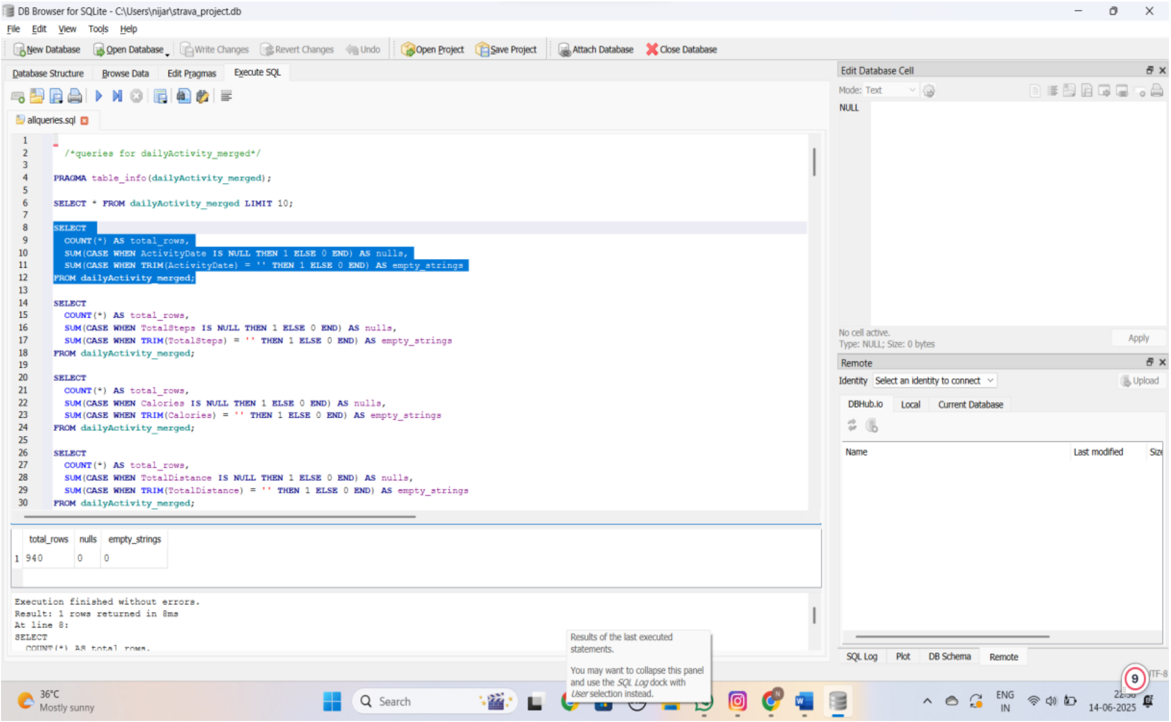
Data Files Handled:

- dailyActivity_merged
- heartrate_daily
- hourlyCalories_cleaned
- hourlyIntensities_cleaned
- hourlySteps_cleaned
- sleepDay_cleaned
- weightLog_dates_fixed

We performed date format corrections, column standardisations, and ensured all datasets are free from NULLs and empty Strings. Also checked for duplicates and finally invalid values (like negative steps, calories, etc.)

SQL QUERIES AND LOGIC

1. Checking NULLS and Invalid values:


DB Browser for SQLite - C:\Users\nijar\strava_project.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
allqueries.sql
Edit Database Cell
Mode: Text
NULL
No cell active.
Type: NULL; Size: 0 bytes
Apply
Remote
Identity Select an identity to connect
Upload
DBHub.io Local Current Database
Name Last modified Size
SQL Log Plot DB Schema Remote
36°C
Mostly sunny
Search
ENG
IN
14-06-2025

```sql

  /*queries for dailyActivity_merged*/

PRAGMA table_info(dailyActivity_merged);

SELECT * FROM dailyActivity_merged LIMIT 10;

SELECT
  COUNT(*) AS total_rows,
  SUM(CASE WHEN ActivityDate IS NULL THEN 1 ELSE 0 END) AS nulls,
  SUM(CASE WHEN TRIM(ActivityDate) = '' THEN 1 ELSE 0 END) AS empty_strings
FROM dailyActivity_merged;

SELECT
  COUNT(*) AS total_rows,
  SUM(CASE WHEN TotalSteps IS NULL THEN 1 ELSE 0 END) AS nulls,
  SUM(CASE WHEN TRIM(TotalSteps) = '' THEN 1 ELSE 0 END) AS empty_strings
FROM dailyActivity_merged;

SELECT
  COUNT(*) AS total_rows,
  SUM(CASE WHEN Calories IS NULL THEN 1 ELSE 0 END) AS nulls,
  SUM(CASE WHEN TRIM(Calories) = '' THEN 1 ELSE 0 END) AS empty_strings
FROM dailyActivity_merged;

SELECT
  COUNT(*) AS total_rows,
  SUM(CASE WHEN TotalDistance IS NULL THEN 1 ELSE 0 END) AS nulls,
  SUM(CASE WHEN TRIM(TotalDistance) = '' THEN 1 ELSE 0 END) AS empty_strings
FROM dailyActivity_merged;
```

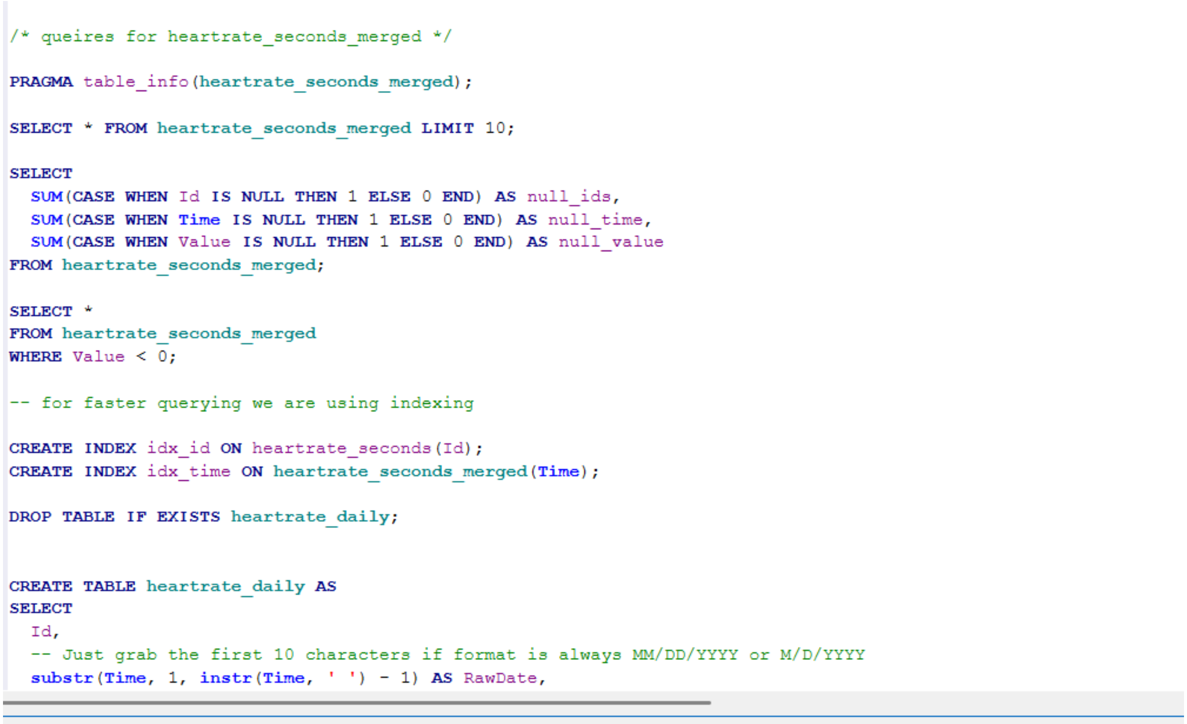
	total_rows	nulls	empty_strings
1	940	0	0

```
Execution finished without errors.
Result: 1 rows returned in 8ms
At line 8:
SELECT
  COUNT(*) AS total_rows,
```

Results of the last executed statements.

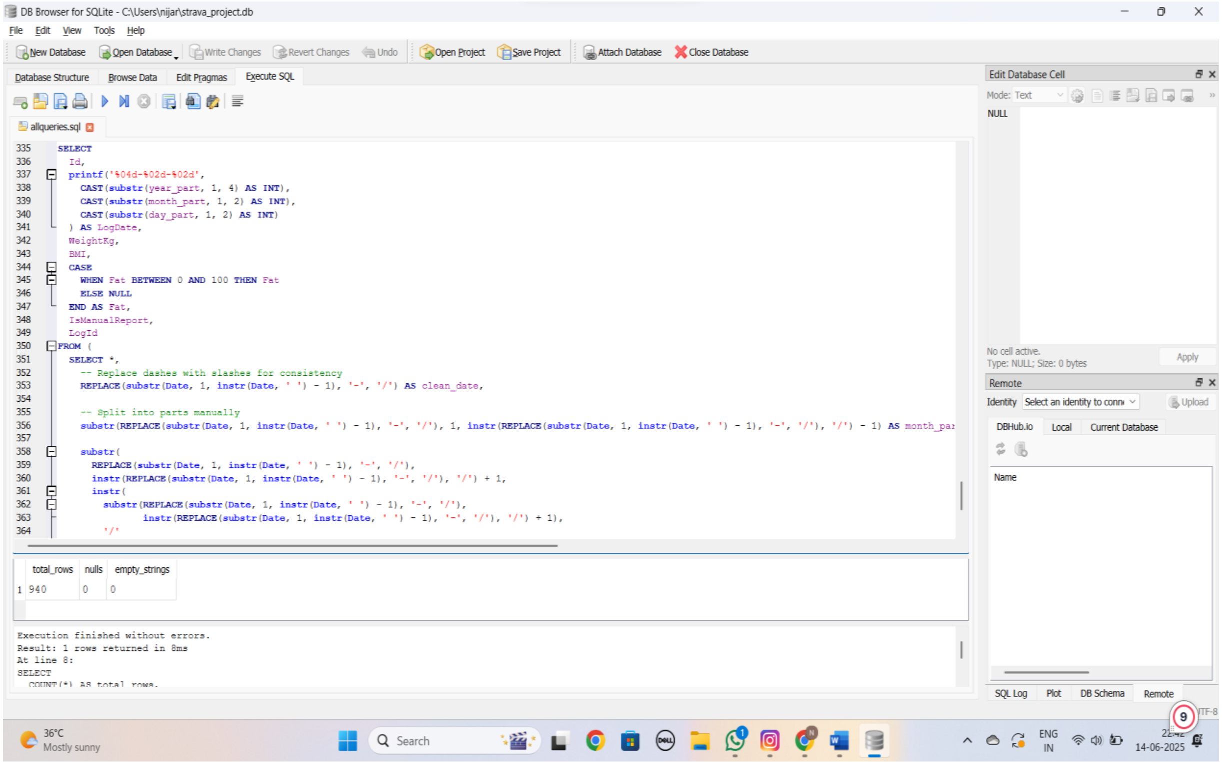
You may want to collapse this panel and use the *SQL Log* dock with *User* selection instead.

2. Checking Duplicates:



```sql
/* queires for heartrate_seconds_merged */

PRAGMA table_info(heartrate_seconds_merged);

SELECT * FROM heartrate_seconds_merged LIMIT 10;

SELECT
  SUM(CASE WHEN Id IS NULL THEN 1 ELSE 0 END) AS null_ids,
  SUM(CASE WHEN Time IS NULL THEN 1 ELSE 0 END) AS null_time,
  SUM(CASE WHEN Value IS NULL THEN 1 ELSE 0 END) AS null_value
FROM heartrate_seconds_merged;

SELECT *
FROM heartrate_seconds_merged
WHERE Value < 0;

-- for faster querying we are using indexing

CREATE INDEX idx_id ON heartrate_seconds(Id);
CREATE INDEX idx_time ON heartrate_seconds_merged(Time);

DROP TABLE IF EXISTS heartrate_daily;


CREATE TABLE heartrate_daily AS
SELECT
  Id,
  -- Just grab the first 10 characters if format is always MM/DD/YYYY or M/D/YYYY
  substr(Time, 1, instr(Time, ' ') - 1) AS RawDate,
```

3. Standardising Date Formats:



```
SELECT
  Id,
  printf('%04d-%02d-%02d',
    CAST(substr(year_part, 1, 4) AS INT),
    CAST(substr(month_part, 1, 2) AS INT),
    CAST(substr(day_part, 1, 2) AS INT)
  ) AS LogDate,
  WeightKg,
  BMI,
  CASE
    WHEN Fat BETWEEN 0 AND 100 THEN Fat
    ELSE NULL
  END AS Fat,
  IsManualReport,
  LogId
FROM (
  SELECT *,
    -- Replace dashes with slashes for consistency
    REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/') AS clean_date,

    -- Split into parts manually
    substr(REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'), 1, instr(REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'), '/') - 1) AS month_par

    substr(
      REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'),
      instr(REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'), '/') + 1,
      instr(
        substr(REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'),
               instr(REPLACE(substr(Date, 1, instr(Date, ' ') - 1), '-', '/'), '/') + 1),
        '/'
```

	total_rows	nulls	empty_strings
1	940	0	0

```
Execution finished without errors.
Result: 1 rows returned in 8ms
At line 8:
SELECT
  COUNT(*) AS total_rows,
```

Final Output and Observations

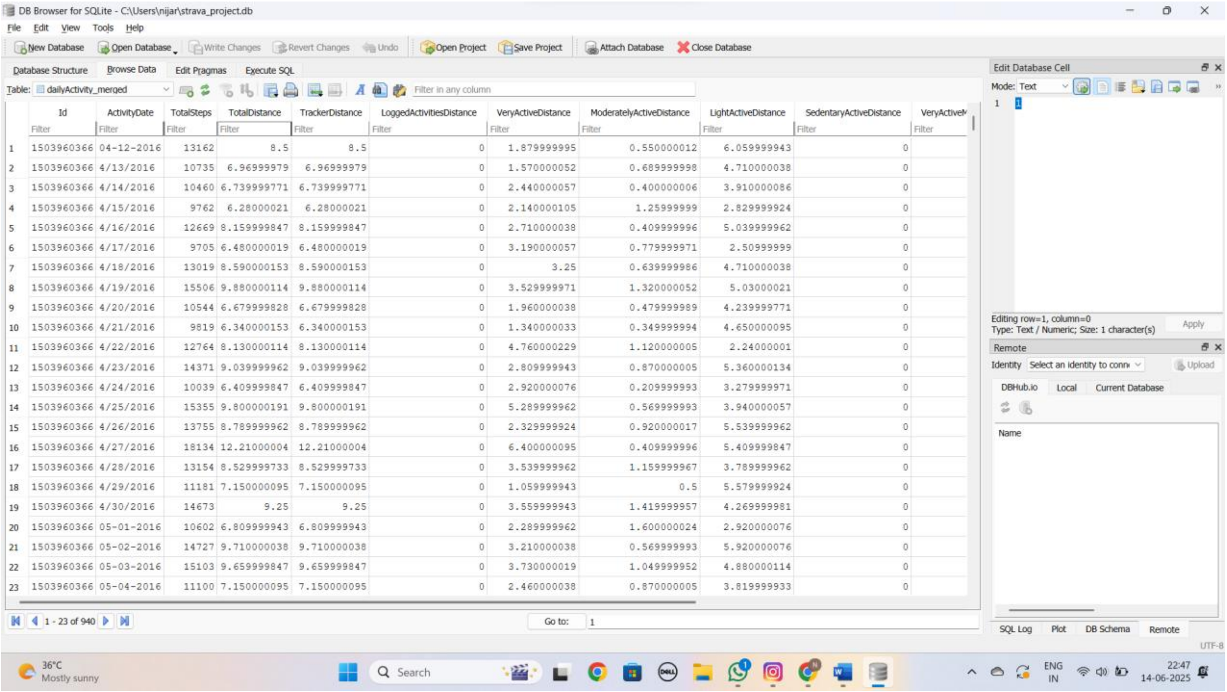
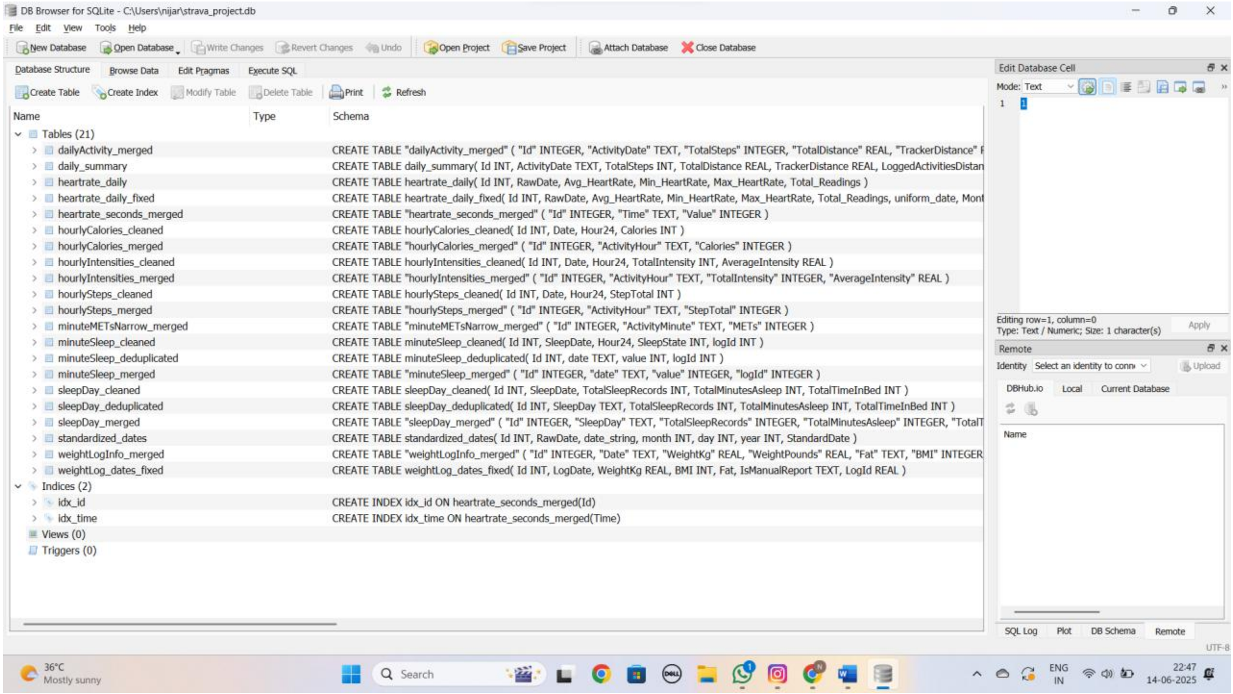


Table: dailyActivity_merged

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDistance	SedentaryActiveDistance	VeryActiveM
1	1503960366	04-12-2016	13162	8.5	8.5	0	1.879999995	0.550000012	6.059999943	0	
2	1503960366	4/13/2016	10735	6.96999979	6.96999979	0	1.570000052	0.689999998	4.710000038	0	
3	1503960366	4/14/2016	10460	6.739999771	6.739999771	0	2.440000057	0.400000006	3.910000086	0	
4	1503960366	4/15/2016	9762	6.28000021	6.28000021	0	2.140000105	1.25999999	2.829999924	0	
5	1503960366	4/16/2016	12669	8.159999847	8.159999847	0	2.710000038	0.409999996	5.039999962	0	
6	1503960366	4/17/2016	9705	6.480000019	6.480000019	0	3.190000057	0.779999971	2.50999999	0	
7	1503960366	4/18/2016	13019	8.590000153	8.590000153	0	3.25	0.639999986	4.710000038	0	
8	1503960366	4/19/2016	15506	9.880000114	9.880000114	0	3.529999971	1.320000052	5.03000021	0	
9	1503960366	4/20/2016	10544	6.679999828	6.679999828	0	1.960000038	0.479999989	4.239999771	0	
10	1503960366	4/21/2016	9819	6.340000153	6.340000153	0	1.340000033	0.349999994	4.650000095	0	
11	1503960366	4/22/2016	12764	8.130000114	8.130000114	0	4.760000229	1.120000005	2.24000001	0	
12	1503960366	4/23/2016	14371	9.039999962	9.039999962	0	2.809999943	0.870000005	5.360000134	0	
13	1503960366	4/24/2016	10039	6.409999847	6.409999847	0	2.920000076	0.209999993	3.279999971	0	
14	1503960366	4/25/2016	15355	9.800000191	9.800000191	0	5.289999962	0.569999993	3.940000057	0	
15	1503960366	4/26/2016	13755	8.789999962	8.789999962	0	2.329999924	0.920000017	5.539999962	0	
16	1503960366	4/27/2016	18134	12.21000004	12.21000004	0	6.400000095	0.409999996	5.409999847	0	
17	1503960366	4/28/2016	13154	8.529999733	8.529999733	0	3.539999962	1.159999967	3.789999962	0	
18	1503960366	4/29/2016	11181	7.150000095	7.150000095	0	1.059999943	0.5	5.579999924	0	
19	1503960366	4/30/2016	14673	9.25	9.25	0	3.559999943	1.419999957	4.269999981	0	
20	1503960366	05-01-2016	10602	6.809999943	6.809999943	0	2.289999962	1.600000024	2.920000076	0	
21	1503960366	05-02-2016	14727	9.710000038	9.710000038	0	3.210000038	0.569999993	5.920000076	0	
22	1503960366	05-03-2016	15103	9.659999847	9.659999847	0	3.730000019	1.049999952	4.880000114	0	
23	1503960366	05-04-2016	11100	7.150000095	7.150000095	0	2.460000038	0.870000005	3.819999933	0	

1 - 23 of 940



Name	Type	Schema
Tables (21)		
dailyActivity_merged		CREATE TABLE "dailyActivity_merged" ("Id" INTEGER, "ActivityDate" TEXT, "TotalSteps" INTEGER, "TotalDistance" REAL, "TrackerDistance" F
daily_summary		CREATE TABLE daily_summary(Id INT, ActivityDate TEXT, TotalSteps INT, TotalDistance REAL, TrackerDistance REAL, LoggedActivitiesDistan
heartrate_daily		CREATE TABLE heartrate_daily(Id INT, RawDate, Avg_HeartRate, Min_HeartRate, Max_HeartRate, Total_Readings)
heartrate_daily_fixed		CREATE TABLE heartrate_daily_fixed(Id INT, RawDate, Avg_HeartRate, Min_HeartRate, Max_HeartRate, Total_Readings, uniform_date, Mont
heartrate_seconds_merged		CREATE TABLE "heartrate_seconds_merged" ("Id" INTEGER, "Time" TEXT, "Value" INTEGER)
hourlyCalories_cleaned		CREATE TABLE hourlyCalories_cleaned(Id INT, Date, Hour24, Calories INT)
hourlyCalories_merged		CREATE TABLE "hourlyCalories_merged" ("Id" INTEGER, "ActivityHour" TEXT, "Calories" INTEGER)
hourlyIntensities_cleaned		CREATE TABLE hourlyIntensities_cleaned(Id INT, Date, Hour24, TotalIntensity INT, AverageIntensity REAL)
hourlyIntensities_merged		CREATE TABLE "hourlyIntensities_merged" ("Id" INTEGER, "ActivityHour" TEXT, "TotalIntensity" INTEGER, "AverageIntensity" REAL)
hourlySteps_cleaned		CREATE TABLE hourlySteps_cleaned(Id INT, Date, Hour24, StepTotal INT)
hourlySteps_merged		CREATE TABLE "hourlySteps_merged" ("Id" INTEGER, "ActivityHour" TEXT, "StepTotal" INTEGER)
minuteMETsNarrow_merged		CREATE TABLE "minuteMETsNarrow_merged" ("Id" INTEGER, "ActivityMinute" TEXT, "METs" INTEGER)
minuteSleep_cleaned		CREATE TABLE minuteSleep_cleaned(Id INT, SleepDate, Hour24, SleepState INT, logId INT)
minuteSleep_deduplicated		CREATE TABLE minuteSleep_deduplicated(Id INT, date TEXT, value INT, logId INT)
minuteSleep_merged		CREATE TABLE "minuteSleep_merged" ("Id" INTEGER, "date" TEXT, "value" INTEGER, "logId" INTEGER)
sleepDay_cleaned		CREATE TABLE sleepDay_cleaned(Id INT, SleepDate, TotalSleepRecords INT, TotalMinutesAsleep INT, TotalTimeInBed INT)
sleepDay_deduplicated		CREATE TABLE sleepDay_deduplicated(Id INT, SleepDay TEXT, TotalSleepRecords INT, TotalMinutesAsleep INT, TotalTimeInBed INT)
sleepDay_merged		CREATE TABLE "sleepDay_merged" ("Id" INTEGER, "SleepDay" TEXT, "TotalSleepRecords" INTEGER, "TotalMinutesAsleep" INTEGER, "TotalT
standardized_dates		CREATE TABLE standardized_dates(Id INT, RawDate, date_string, month INT, day INT, year INT, StandardDate)
weightLogInfo_merged		CREATE TABLE "weightLogInfo_merged" ("Id" INTEGER, "Date" TEXT, "WeightKg" REAL, "WeightPounds" REAL, "Fat" TEXT, "BMI" INTEGER
weightLog_dates_fixed		CREATE TABLE weightLog_dates_fixed(Id INT, LogDate, WeightKg REAL, BMI INT, Fat, IsManualReport TEXT, LogId REAL)
Indices (2)		
idx_id		CREATE INDEX idx_id ON heartrate_seconds_merged(Id)
idx_time		CREATE INDEX idx_time ON heartrate_seconds_merged(Time)
Views (0)		
Triggers (0)		

After cleaning:

- All datasets use either the mm-dd-yyy or mm/dd/yyy date format.
- NULL values and zero entries were removed.
- Column data types (like integers for steps and floats for weight) were validated.

Challenges Faced:

- Inconsistent date formats like yyyy-mm-dd 00:00 AM/PM required conditional handling.
- Some datasets lacked primary keys, so joins required caution.