

```

USE stravafitness;
-- Sleep vs Calories Burned
SELECT
    ROUND(AVG(s.TotalMinutesAsleep), 1) AS avg_sleep_mins,
    ROUND(AVG(a.Calories), 1) AS avg_calories_burned
FROM dailyActivity_merged a
JOIN sleepDay_merged s
    ON a.Id = s.Id
    AND a.ActivityDate = DATE(s.SleepDay);
-- Step Consistency Over Time
SELECT
    user_Id,
    ActivityDay,
    StepTotal
FROM cleaned_dailysteps_merged
ORDER BY user_Id, ActivityDay;
-- Calories Burned on High Sleep vs Low Sleep Days
SELECT
    CASE
        WHEN s.TotalMinutesAsleep >= 420 THEN 'Well Rested (7+ hrs)'
        ELSE 'Sleep Deprived (<7 hrs)'
    END AS sleep_quality,
    ROUND(AVG(a.Calories), 0) AS avg_calories_burned
FROM dailyActivity_merged a
JOIN sleepDay_merged s
    ON a.Id = s.Id
    AND a.ActivityDate = DATE(s.SleepDay)
GROUP BY sleep_quality;
-- Top 10 Most Active Users (Based on Total Steps)
SELECT
    Id AS user_id,
    ROUND(AVG(TotalSteps), 0) AS avg_daily_steps,
    ROUND(AVG(Calories), 0) AS avg_calories
FROM dailyActivity_merged
GROUP BY Id
ORDER BY avg_daily_steps DESC
LIMIT 10;
-- Daily trends for each user
SELECT
    Id,
    ActivityDate,
    TotalSteps,
    Calories,
    VeryActiveMinutes,
    SedentaryMinutes
FROM dailyActivity_merged;
-- Merged engagement table
CREATE TABLE unified_engagement_data1 AS
SELECT
    da.Id,
    da.ActivityDate,
    da.TotalSteps,
    da.SedentaryMinutes,
    da.Calories,
    sd.TotalMinutesAsleep,

```

```
        sd.TotalTimeInBed
FROM dailyActivity_merged da
JOIN sleepDay_merged sd
    ON da.Id = sd.Id AND da.ActivityDate = sd.SleepDay;
-- Check how weight & BMI relate to active lifestyle
SELECT
    wl.user_id,
    wl.WeightKg,
    wl.BMI,
    da.TotalSteps,
    da.Calories
FROM weightloginfo wl
JOIN dailyactivity_merged da
    ON wl.user_id = da.Id
WHERE wl.WeightKg IS NOT NULL
    AND wl.BMI IS NOT NULL;
```



```

SELECT
    a.Id AS user_id,
    a.ActivityDate,
    a.TotalSteps,
    a.SedentaryMinutes,
    a.VeryActiveMinutes,
    a.FairlyActiveMinutes,
    a.LightlyActiveMinutes,
    s.TotalMinutesAsleep,
    s.TotalTimeInBed,
    c.Calories,
    ds.StepTotal
FROM dailyActivity_merged a
LEFT JOIN sleepDay_merged s
    ON a.Id = s.Id
    AND a.ActivityDate = DATE(s.SleepDay)
LEFT JOIN dailyCalories_merged c
    ON a.Id = c.Id
    AND a.ActivityDate = c.ActivityDay
LEFT JOIN cleaned_dailysteps_merged ds
    ON a.Id = ds.user_Id
    AND a.ActivityDate = ds.ActivityDay;
-- Engagement by Day of Week
-- öŸ,, Create merged table from activity and sleep data
CREATE TABLE unified_engagement_data AS
SELECT
    da.Id,
    da.ActivityDate,
    da.TotalSteps,
    da.SedentaryMinutes,
    da.Calories,
    sd.TotalMinutesAsleep,
    sd.TotalTimeInBed
FROM dailyActivity_merged da
JOIN sleepDay_merged sd
    ON da.Id = sd.Id
    AND da.ActivityDate = sd.SleepDay;
SELECT
    DAYNAME(ActivityDate) AS day_name,
    ROUND(AVG(TotalSteps), 0) AS avg_steps,
    ROUND(AVG(SedentaryMinutes), 0) AS avg_sedentary,
    ROUND(AVG(TotalMinutesAsleep), 0) AS avg_sleep
FROM unified_engagement_data
GROUP BY day_name
ORDER BY FIELD(day_name, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday');
-- Correlation Between Sleep and Steps: do well-rested users walk more?
SELECT
    TotalMinutesAsleep,
    TotalSteps
FROM unified_engagement_data
WHERE TotalMinutesAsleep IS NOT NULL AND TotalSteps IS NOT NULL;
-- User Segmentation: Sleep & Activity Profile- Prepare for user clustering in Power BI
SELECT
    Id,

```

```

ROUND(AVG(TotalSteps), 0) AS avg_steps,
ROUND(AVG(Calories), 0) AS avg_calories,
ROUND(AVG(SedentaryMinutes), 0) AS avg_sedentary,
ROUND(AVG(TotalMinutesAsleep), 0) AS avg_sleep,
ROUND(AVG(TotalTimeInBed), 0) AS avg_time_in_bed
FROM unified_engagement_data
GROUP BY Id;
-- Engagement Bins by Steps
SELECT
CASE
    WHEN TotalSteps >= 10000 THEN 'Highly Active'
    WHEN TotalSteps >= 5000 THEN 'Moderately Active'
    ELSE 'Low Activity'
END AS activity_level,
COUNT(*) AS num_days,
ROUND(AVG(TotalMinutesAsleep), 0) AS avg_sleep,
ROUND(AVG(SedentaryMinutes), 0) AS avg_sedentary
FROM unified_engagement_data
GROUP BY activity_level;
-- When are users most active in a day?
SELECT
    HOUR(ActivityHour) AS hour_of_day,
    AVG(StepTotal) AS avg_steps
FROM hourlysteps
GROUP BY hour_of_day
ORDER BY hour_of_day;
-- Calories burned at different hours
SELECT
    HOUR(ActivityHour) AS hour_of_day,
    AVG(Calories) AS avg_calories
FROM hourlycalories
GROUP BY hour_of_day
ORDER BY hour_of_day;
-- When are people doing very active movements?
SELECT
    HOUR(ActivityHour) AS hour_of_day,
    AVG(TotalIntensity) AS avg_intensity
FROM hourlyIntensities_merged
GROUP BY hour_of_day
ORDER BY hour_of_day;

```



```

USE stravafitness
-- Average daily steps, sleep, and calories per user
SELECT
    da.Id,
    ROUND(AVG(da.TotalSteps), 0) AS avg_steps,
    ROUND(AVG(sd.TotalMinutesAsleep), 0) AS avg_sleep,
    ROUND(AVG(da.Calories), 0) AS avg_calories
FROM dailyactivity_merged da
JOIN sleepday_merged sd
    ON da.Id = sd.Id AND da.ActivityDate = sd.SleepDay
GROUP BY da.Id;
-- Merge BMI with average steps
SELECT
    wl.user_id,
    wl.BMI,
    ROUND(AVG(da.TotalSteps), 0) AS avg_steps
FROM weightloginfo wl
JOIN dailyactivity_merged da
    ON wl.user_id = da.Id
WHERE wl.BMI IS NOT NULL
GROUP BY wl.user_id, wl.BMI;
-- Steps by day of week per user
SELECT
    Id,
    DAYNAME(ActivityDate) AS day_of_week,
    SUM(TotalSteps) AS total_steps
FROM dailyactivity_merged
GROUP BY Id, day_of_week;
-- Chart 4 (Updated): Steps Contribution by Activity Level (Column Chart)
-- Calculate percentage contribution of each activity level to total steps
-- Average distance types per user
SELECT
    Id,
    ROUND(AVG(VeryActiveDistance), 2) AS very_active_km,
    ROUND(AVG(LightActiveDistance), 2) AS light_active_km,
    ROUND(AVG(TotalDistance), 2) AS total_km
FROM dailyactivity_merged
GROUP BY Id;
-- Chart 5: Calories Burned by Day of Week (Column Chart)
SELECT
    DAYNAME(ActivityDate) AS day_name,
    ROUND(AVG(Calories), 0) AS avg_calories
FROM dailyactivity_merged
GROUP BY day_name
ORDER BY FIELD(day_name,
'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday');
-- Chart 6: User Clustering (K-means style Prewrite)
-- Steps + Sleep + Calories per user
SELECT
    da.Id,
    ROUND(AVG(da.TotalSteps), 0) AS avg_steps,
    ROUND(AVG(sd.TotalMinutesAsleep), 0) AS avg_sleep,
    ROUND(AVG(da.Calories), 0) AS avg_calories,
    ROUND(AVG(da.SedentaryMinutes), 0) AS avg_sedentary
FROM dailyactivity_merged da

```

```
JOIN sleepday_merged sd
    ON da.Id = sd.Id AND da.ActivityDate = sd.SleepDay
GROUP BY da.Id;
-- Chart 7: BMI Category Distribution (Pie Chart)
-- Group users by BMI category
SELECT
    CASE
        WHEN BMI < 18.5 THEN 'Underweight'
        WHEN BMI BETWEEN 18.5 AND 24.9 THEN 'Normal'
        WHEN BMI BETWEEN 25 AND 29.9 THEN 'Overweight'
        ELSE 'Obese'
    END AS bmi_category,
    COUNT(*) AS user_count
FROM weightloginfo
WHERE BMI IS NOT NULL
GROUP BY bmi_category;
```


