# Dataset Creation and Life Expectancy Adjustment – SQL Workflow

This document outlines the step-by-step SQL-based workflow used to create and populate the `life\_expectancy\_data` dataset for the Final Year Project. Each section details the logic, purpose, and corresponding SQL queries executed in sequence to simulate realistic life expectancy data based on various health and lifestyle parameters.

## 1. Database and Schema Initialization

Create a new database named it “final\_year\_project”

Create a new schema in the connected server, named it “final\_year\_project”

## 2. Table Creation

Now we start applying queries to create life expectancy dataset base on lifestyle

USE final\_year\_project;

the SQL query to create the table with your specified columns:

CREATE TABLE life\_expectancy\_data (

id INT AUTO\_INCREMENT PRIMARY KEY,

Gender VARCHAR(10),

Country VARCHAR(100),

## 6. Populate Exercise Hours

Exercise\_hrs\_per\_week FLOAT,

## 7. Populate Diet Type

Diet\_Type VARCHAR(50),

## 8. Populate Medical History

Medical\_History TEXT,

## 9. Populate Work Stress Level

Work\_Stress\_Level VARCHAR(20),

Smoking VARCHAR(20),

Alcohol\_Consumption VARCHAR(20),

## 10. Populate Social Life

Social\_Life VARCHAR(20),

## 13. Populate Sleep Hours

Sleep\_hrs\_per\_day FLOAT,

BMI FLOAT,

Life\_Expectancy FLOAT

);

## 3. Insert Gender Data

insert exactly 4992 rows with alternating 'Male' and 'Female' values in the Gender column

DELIMITER $$

CREATE PROCEDURE insert\_gender\_rows()

BEGIN

DECLARE i INT DEFAULT 1;

WHILE i <= 4992 DO

INSERT INTO life\_expectancy\_data (Gender)

VALUES (

IF(MOD(i, 2) = 1, 'Male', 'Female')

);

SET i = i + 1;

END WHILE;

END$$

DELIMITER ;

-- Now call the procedure

CALL insert\_gender\_rows();

-- Drop it if you don't need it anymore

DROP PROCEDURE insert\_gender\_rows;

## 4. Assign Country

assign 8 countries to the existing 4992 rows (already inserted), such that each country gets 624 consecutive rows, in this order:

Australia, Brazil, Canada, France, Germany, India, Japan, South Africa

-- Australia: First 624 rows

UPDATE life\_expectancy\_data

SET Country = 'Australia'

ORDER BY id

LIMIT 624;

-- Brazil: Next 624 rows

UPDATE life\_expectancy\_data

SET Country = 'Brazil'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- Canada

UPDATE life\_expectancy\_data

SET Country = 'Canada'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- France

UPDATE life\_expectancy\_data

SET Country = 'France'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- Germany

UPDATE life\_expectancy\_data

SET Country = 'Germany'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- India

UPDATE life\_expectancy\_data

SET Country = 'India'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- Japan

UPDATE life\_expectancy\_data

SET Country = 'Japan'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

-- South Africa

UPDATE life\_expectancy\_data

SET Country = 'South Africa'

WHERE Country IS NULL

ORDER BY id

LIMIT 624;

## 5. Disable SQL Safe Updates

off the safe mode

SET SQL\_SAFE\_UPDATES = 0;

## 6. Populate Exercise Hours

the Exercise\_hrs\_per\_week values to be randomly distributed across all 4992 rows, not clustered.

 Randomly select 1747 rows and set values < 1.5

 Then set the remaining rows to random values between 1.5 and 14

-- Step 1: Set 1747 random rows with values < 1.5

UPDATE life\_expectancy\_data

## 6. Populate Exercise Hours

SET Exercise\_hrs\_per\_week = ROUND(RAND() \* 1.49, 2)

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 6. Populate Exercise Hours

WHERE Exercise\_hrs\_per\_week IS NULL

ORDER BY RAND()

LIMIT 1747

) AS sub

);

-- Step 2: Set remaining rows with values between 1.5 and 14

UPDATE life\_expectancy\_data

## 6. Populate Exercise Hours

SET Exercise\_hrs\_per\_week = ROUND(1.5 + (RAND() \* (14 - 1.5)), 2)

## 6. Populate Exercise Hours

WHERE Exercise\_hrs\_per\_week IS NULL;

## 7. Populate Diet Type

fill the Diet\_Type column in repeating order:  
Balanced, Unhealthy, Healthy, Balanced, Unhealthy, Healthy, ...

-- Initialize a row counter variable

SET @row := 0;

## 7. Populate Diet Type

-- Update Diet\_Type in repeating pattern using MOD logic

UPDATE life\_expectancy\_data

## 7. Populate Diet Type

SET Diet\_Type = (

CASE MOD(@row := @row + 1, 3)

WHEN 1 THEN 'Balanced'

WHEN 2 THEN 'Unhealthy'

ELSE 'Healthy'

END

)

ORDER BY id;

## 8. Populate Medical History

fill the Medical\_History column with:

40% Fit = 1996 rows

10% each of the other 6 conditions = 499 rows each

Total = 1996 + (6 × 499) = 4990 rows (we’ll fill 2 extra randomly with any value to make 4992)

-- 1. Fit (40%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Fit'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 1996

) AS sub

);

-- 2. Cancer (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Cancer'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- 3. Diabetes (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Diabetes'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- 4. Asthma (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Asthma'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- 5. HIV/AIDS (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'HIV/AIDS'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- 6. Heart Disease (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Heart Disease'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- 7. Kidney & Liver Diseases (10%)

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Kidney & Liver Diseases'

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 8. Populate Medical History

WHERE Medical\_History IS NULL

ORDER BY RAND()

LIMIT 499

) AS sub

);

-- Optional: Fill any remaining NULLs with 'Fit'

UPDATE life\_expectancy\_data

## 8. Populate Medical History

SET Medical\_History = 'Fit'

## 8. Populate Medical History

WHERE Medical\_History IS NULL;

## 9. Populate Work Stress Level

fill Work\_Stress\_Level columns with the values:

Low, Moderate, High  
…and distribute them randomly across all 4992 rows.

## 9. Populate Work Stress Level

-- Fill Work\_Stress\_Level randomly

UPDATE life\_expectancy\_data

## 9. Populate Work Stress Level

SET Work\_Stress\_Level = (

CASE FLOOR(1 + RAND() \* 3)

WHEN 1 THEN 'Low'

WHEN 2 THEN 'Moderate'

ELSE 'High'

END

)

## 9. Populate Work Stress Level

WHERE Work\_Stress\_Level IS NULL;

## 10. Populate Social Life

fill Social\_Life columns with the values:

Low, Moderate, High  
…and distribute them randomly across all 4992 rows.

## 10. Populate Social Life

-- Fill Social\_Life randomly

UPDATE life\_expectancy\_data

## 10. Populate Social Life

SET Social\_Life = (

CASE FLOOR(1 + RAND() \* 3)

WHEN 1 THEN 'Low'

WHEN 2 THEN 'Moderate'

ELSE 'High'

END

)

## 10. Populate Social Life

WHERE Social\_Life IS NULL;

## 11. Populate Smoking Levels

fill Smoking columns with random values from the following set:

'Never', 'Low', 'Moderate', 'High'

UPDATE life\_expectancy\_data

SET Smoking = (

CASE FLOOR(1 + RAND() \* 4)

WHEN 1 THEN 'Never'

WHEN 2 THEN 'Low'

WHEN 3 THEN 'Moderate'

ELSE 'High'

END

)

WHERE Smoking IS NULL;

## 12. Populate Alcohol Consumption

fill Alcohol\_Consumption columns with random values from the following set:

'Never', 'Low', 'Moderate', 'High'

UPDATE life\_expectancy\_data

SET Alcohol\_Consumption = (

CASE FLOOR(1 + RAND() \* 4)

WHEN 1 THEN 'Never'

WHEN 2 THEN 'Low'

WHEN 3 THEN 'Moderate'

ELSE 'High'

END

)

WHERE Alcohol\_Consumption IS NULL;

## 13. Populate Sleep Hours

fill the Sleep\_hrs\_per\_day column with:

40% (1996 rows): random values from 7–9

30% (≈1498 rows): random values from 3–6

30% (≈1498 rows): random values from 9–15

-- 1. 40% → 7 to 9 hours

UPDATE life\_expectancy\_data

## 13. Populate Sleep Hours

SET Sleep\_hrs\_per\_day = ROUND(7 + (RAND() \* (9 - 7)), 1)

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 13. Populate Sleep Hours

WHERE Sleep\_hrs\_per\_day IS NULL

ORDER BY RAND()

LIMIT 1996

) AS sub

);

-- 2. 30% → 3 to 6 hours

UPDATE life\_expectancy\_data

## 13. Populate Sleep Hours

SET Sleep\_hrs\_per\_day = ROUND(3 + (RAND() \* (6 - 3)), 1)

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

## 13. Populate Sleep Hours

WHERE Sleep\_hrs\_per\_day IS NULL

ORDER BY RAND()

LIMIT 1498

) AS sub

);

-- 3. 30% → 9 to 15 hours

UPDATE life\_expectancy\_data

## 13. Populate Sleep Hours

SET Sleep\_hrs\_per\_day = ROUND(9 + (RAND() \* (15 - 9)), 1)

## 13. Populate Sleep Hours

WHERE Sleep\_hrs\_per\_day IS NULL;

## 14. Populate BMI

14. the BMI column, you want:

40% (1996 rows): values between 18–25 (normal)

30% (1498 rows): values between 13–17 (underweight)

30% (1498 rows): values between 26–40 (overweight/obese)

Randomly distributed.

-- 1. 40% → 18 to 25

UPDATE life\_expectancy\_data

SET BMI = ROUND(18 + (RAND() \* (25 - 18)), 1)

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

WHERE BMI IS NULL

ORDER BY RAND()

LIMIT 1996

) AS sub

);

-- 2. 30% → 13 to 17

UPDATE life\_expectancy\_data

SET BMI = ROUND(13 + (RAND() \* (17 - 13)), 1)

WHERE id IN (

SELECT id FROM (

SELECT id FROM life\_expectancy\_data

WHERE BMI IS NULL

ORDER BY RAND()

LIMIT 1498

) AS sub

);

-- 3. 30% → 26 to 40

UPDATE life\_expectancy\_data

SET BMI = ROUND(26 + (RAND() \* (40 - 26)), 1)

WHERE BMI IS NULL;

## 15. Set Base Life Expectancy

fill the Life\_Expectancy column based on a mapping of Country and Gender, using the following values:

UPDATE life\_expectancy\_data

SET Life\_Expectancy =

CASE

WHEN Country = 'Japan' AND Gender = 'Male' THEN 81

WHEN Country = 'Japan' AND Gender = 'Female' THEN 87

WHEN Country = 'Australia' AND Gender = 'Male' THEN 81

WHEN Country = 'Australia' AND Gender = 'Female' THEN 85

WHEN Country = 'France' AND Gender = 'Male' THEN 80

WHEN Country = 'France' AND Gender = 'Female' THEN 86

WHEN Country = 'Canada' AND Gender = 'Male' THEN 80

WHEN Country = 'Canada' AND Gender = 'Female' THEN 84

WHEN Country = 'Germany' AND Gender = 'Male' THEN 79

WHEN Country = 'Germany' AND Gender = 'Female' THEN 84

WHEN Country = 'Brazil' AND Gender = 'Male' THEN 72

WHEN Country = 'Brazil' AND Gender = 'Female' THEN 79

WHEN Country = 'India' AND Gender = 'Male' THEN 69

WHEN Country = 'India' AND Gender = 'Female' THEN 72

WHEN Country = 'South Africa' AND Gender = 'Male' THEN 61

WHEN Country = 'South Africa' AND Gender = 'Female' THEN 67

ELSE NULL

END

WHERE Life\_Expectancy IS NULL;

## 6. Populate Exercise Hours

16. now adjust Life\_Expectancy based on Exercise\_hrs\_per\_week, such that:

## 6. Populate Exercise Hours

Only update if Exercise\_hrs\_per\_week > 1.53

The higher the value, the larger the added years

Addition should be between 0.43 and 6.9 years, proportional to the hours

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy + (

## 6. Populate Exercise Hours

0.43 + ((Exercise\_hrs\_per\_week - 1.53) / (14 - 1.53)) \* (6.9 - 0.43)

)

## 6. Populate Exercise Hours

WHERE Exercise\_hrs\_per\_week > 1.53;

## 7. Populate Diet Type

16. now update Life\_Expectancy based on Diet\_Type as follows:

Healthy → increase by +10.6 years

Unhealthy → decrease by −5.4 years

Balanced → no change

-- Increase for Healthy diet

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy + 10.6

## 7. Populate Diet Type

WHERE Diet\_Type = 'Healthy';

-- Decrease for Unhealthy diet

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 5.4

## 7. Populate Diet Type

WHERE Diet\_Type = 'Unhealthy';

## 8. Populate Medical History

17. update Life\_Expectancy based on Medical\_History with the following rules:

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 18.6

## 8. Populate Medical History

WHERE Medical\_History = 'Asthma';

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 10

## 8. Populate Medical History

WHERE Medical\_History = 'Diabetes';

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 30

## 8. Populate Medical History

WHERE Medical\_History IN ('HIV/AIDS', 'Cancer');

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 7

## 8. Populate Medical History

WHERE Medical\_History = 'Kidney & Liver Diseases';

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 12

## 8. Populate Medical History

WHERE Medical\_History = 'Heart Disease';

## 9. Populate Work Stress Level

18. let's update Life\_Expectancy based on Work\_Stress\_Level:

Low → increase by +0.85 years

Moderate → no change

High → decrease by −0.85 year

-- Increase for low stress

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy + 0.85

## 9. Populate Work Stress Level

WHERE Work\_Stress\_Level = 'Low';

-- Decrease for high stress

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 0.85

## 9. Populate Work Stress Level

WHERE Work\_Stress\_Level = 'High';

## 20. Adjust Life Expectancy: Smoking

19. et's update Life\_Expectancy based on the Smoking column with the following logic:

-- High smoking reduces life expectancy by 13 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 13

WHERE Smoking = 'High';

-- Moderate smoking reduces by 9 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 9

WHERE Smoking = 'Moderate';

-- Low smoking reduces by 5 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 5

WHERE Smoking = 'Low';

## 21. Adjust Life Expectancy: Alcohol

20. Let's update Life\_Expectancy based on the Alcohol\_Consumption column using the following rules:

-- High alcohol consumption reduces life expectancy by 4.5 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 4.5

WHERE Alcohol\_Consumption = 'High';

-- Moderate alcohol consumption reduces by 1.5 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 1.5

WHERE Alcohol\_Consumption = 'Moderate';

-- Low alcohol consumption reduces by 0.6 years

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 0.6

WHERE Alcohol\_Consumption = 'Low';

## 10. Populate Social Life

21. Let's update Life\_Expectancy based on the Social\_Life column using your specified adjustments:

-- Increase for high social life

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy + 0.20

## 10. Populate Social Life

WHERE Social\_Life = 'High';

-- Decrease for low social life

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 0.20

## 10. Populate Social Life

WHERE Social\_Life = 'Low';

## 13. Populate Sleep Hours

22. To update Life\_Expectancy based on Sleep\_hrs\_per\_day, we’ll apply:

-- Increase for sleep hours between 7 and 9 (inclusive)

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy + 2.5

## 13. Populate Sleep Hours

WHERE Sleep\_hrs\_per\_day BETWEEN 7 AND 9;

-- Decrease for sleep hours outside 7–9 range

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 2.5

## 13. Populate Sleep Hours

WHERE Sleep\_hrs\_per\_day < 7 OR Sleep\_hrs\_per\_day > 9;

## 24. Adjust Life Expectancy: BMI and Gender

23. Life Expectancy Adjustments Based on BMI and Gender

-- Obese males (BMI ≥ 30)

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 4.2

WHERE BMI >= 30 AND Gender = 'Male';

-- Obese females (BMI ≥ 30)

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 3.5

WHERE BMI >= 30 AND Gender = 'Female';

-- Underweight males (BMI < 18.5)

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 4.3

WHERE BMI < 18.5 AND Gender = 'Male';

-- Underweight females (BMI < 18.5)

UPDATE life\_expectancy\_data

SET Life\_Expectancy = Life\_Expectancy - 4.5

WHERE BMI < 18.5 AND Gender = 'Female';

24. ound the Life\_Expectancy column to 2 decimal places