Tural Novruzov 28632

Project Step 3

Introduction

My project is titled "Life Expectancy and Inequality" where I will develop a database application to analyze inequality in life expectancy between countries and genders. A secondary objective of the application is to explore the relationship between life expectancy and population in different countries. The GitHub repository for the project: https://github.com/turalnovruzov/cs306-project. **I will have every step in different branches for easier browsing and grading experience. Please refer to the branch "step3" to see my project's state when I finished Step 3.**

Views

You can find the SQL code for the views in "Step 3/views.sql" and the logs in "Step 3/views_logs.log".

View #1

This view analyzes the relationship between health expenditure and life expectancy. The result table contains year, country name, and corresponding health expenditure and life expectancy

```
CREATE VIEW expenditure_vs_life_expectancy AS
SELECT h.year, c.name, h.expenditure, l.le
FROM country c
JOIN health_expenditure h ON c.iso = h.iso
JOIN life_expectancy_all l ON h.year = l.year AND c.iso = l.iso;
```

View #2

This view analyzes the difference in survival rate to 65 between males and females. The result table contains year, country name, and difference in the survival rate of males and females.

```
CREATE VIEW gender_survival_difference AS
SELECT s.year, c.name, (s.survival_female - s.survival_male) AS survival_diff
FROM country c
JOIN survival_to_65 s ON c.iso = s.iso;
```

View #3

This view finds the countries whose life expectancy decreased in the year 2000 compared to the year 1990. The result table contains the country names.

```
CREATE VIEW life_expectancy_decrease_1990_2000 AS
SELECT c.name
FROM (
  SELECT l1.iso, l1.year AS year1, l2.year AS year2, l1.le - l2.le AS diff
  FROM life_expectancy_all l1, life_expectancy_all l2
  WHERE l1.year = 1990 AND l2.year = 2000 AND l1.iso = l2.iso) r1
JOIN country c ON r1.iso = c.iso
WHERE r1.diff > 0;
```

View #4

This view finds countries with a decrease in survival_male and an increase in survival_female from 1960 to 1970 using IN. The result table contains the country names.

```
CREATE VIEW countries_with_survival_male_dec_female_inc_in AS
SELECT c.name
FROM country c
WHERE c.iso IN (
  SELECT s1.iso
  FROM survival_to_65 s1, survival_to_65 s2
  WHERE s1.year = 1960 AND s2.year = 1970 AND s1.iso = s2.iso AND s1.survival_male >
s2.survival_male AND s1.survival_female < s2.survival_female);
```

View #5

Find countries with a decrease in survival_male and an increase in survival_female from 1960 to 1970 using EXISTS. The result table contains the country names.

```
CREATE VIEW countries_with_survival_male_dec_female_inc_exists AS
SELECT c.name
FROM country c
WHERE EXISTS (
  SELECT 1
  FROM survival_to_65 s1, survival_to_65 s2
  WHERE s1.year = 1960 AND s2.year = 1970 AND s1.iso = s2.iso AND s1.survival_male >
s2.survival_male AND s1.survival_female < s2.survival_female AND c.iso = s1.iso);
```

View #6

This view calculates the average life expectancy for each country in the 90s (1990-1999). The result table contains the country names and the average life expectancy of those countries in the 90s.

```
CREATE VIEW avg_life_expectancy_90s AS
SELECT c.name, AVG(l.le) AS avg_life_expectancy
FROM country c
JOIN life_expectancy_all l ON c.iso = l.iso
WHERE l.year BETWEEN 1990 AND 1999
GROUP BY c.name;
```

View #7

This view calculates the total health expenditure for each country in the 2010s (2010-2019). The result table contains the country names and the total health expenditure for those countries in the 2010s.

```
CREATE VIEW total_health_expenditure_2010s AS
SELECT c.name, SUM(h.expenditure) AS total_expenditure
FROM country c
JOIN health_expenditure h ON c.iso = h.iso
WHERE h.year BETWEEN 2010 AND 2019
GROUP BY c.name;
```

View #8

This view finds the country with the highest life expectancy in the year 2000. The result table contains the country's name and its life expectancy in the year 2000.

```
CREATE VIEW highest_life_expectancy AS
SELECT c.name, MAX(l.le) AS max_life_expectancy
FROM country c
JOIN life_expectancy_all l ON c.iso = l.iso
WHERE l.year = 2000
GROUP BY c.name
ORDER BY max_life_expectancy DESC
LIMIT 1;
```

View #9

This view finds the minimum and maximum differences in survival rate to 65 between males and females for each country. The result table contains the country names, the minimum difference between their survival_male and survival_female, and the maximum difference between their survival_male and survival_female.

```
CREATE VIEW min_max_difference_survival_male_female AS
SELECT c.name, MIN(s.survival_female - s.survival_male) AS min_diff, MAX(s.survival_female
- s.survival_male) AS max_diff
FROM country c
JOIN survival_to_65 s ON c.iso = s.iso
GROUP BY c.name;
```

View #10

This view counts count the number of years in which life expectancy has increased compared to the previous year for each country. The result table contains the country names, and the count of years their life expectancy has increased.

```
CREATE VIEW count_years_life_expectacy_increased AS
SELECT c.name, COUNT(*) AS increased_years
FROM country c
JOIN life_expectancy_all l1 ON c.iso = l1.iso
JOIN life_expectancy_all l2 ON l1.year - 1 = l2.year AND l1.iso = l2.iso
WHERE l1.le > l2.le
GROUP BY c.name;
```

Constraint

I have created a constraint for the inequality_in_life table. The inequality column in this table stores percentage data, thus the values should be between 0 and 100. This constraint enforces that the value of inequality column is between 0 and 100. You can find the SQL code in “Step 3/constraint_inequality.sql” and the logs in “Step 3/trigger_logs.log”.

```
ALTER TABLE cs306_project.inequality_in_life
ADD CONSTRAINT check_inequality
CHECK (inequality >= 0 AND inequality <= 100)
```

Trigger

I added two triggers (before insert and before update) that limit the values of the inequality column in inequality_in_life. This trigger set the new values to the limits if the value exceeds the limits, which are 0 and 100. You can find the SQL code in “Step 3/trigger_inequality.sql” and the logs in “Step 3/trigger_logs.log”.

```
CREATE TRIGGER inequality_in_life_before_insert
BEFORE INSERT ON cs306_project.inequality_in_life
FOR EACH ROW
BEGIN
   IF NEW.inequality < 0 THEN
      SET NEW.inequality = 0;
   ELSEIF NEW.inequality > 100 THEN
      SET NEW.inequality = 100;
   END IF;
END //

CREATE TRIGGER inequality_in_life_before_UPDATE
BEFORE UPDATE ON cs306_project.inequality_in_life
FOR EACH ROW
BEGIN
   IF NEW.inequality < 0 THEN
      SET NEW.inequality = 0;
   ELSEIF NEW.inequality > 100 THEN
      SET NEW.inequality = 100;
   END IF;
END
```

The pros and cons of using a constraint vs a trigger depend on the project. I believe in this project using a constraint is a better idea, since I wouldn’t want any inaccurate data. However, in a project where you can have some spare room in terms of accuracy, it would be better to use a trigger since the trigger won’t stop the insertion or the update of data.

Stored Procedure

I created a procedure that accepts an iso code as input and returns a table with the minimum and maximum population of that country throughout the years. You can find the SQL code in “Step 3/get_min_max_population.sql” and the logs in “Step 3/procedure_logs.log”.

```
CREATE PROCEDURE `get_min_max_population` (IN iso_ CHAR(8))
BEGIN
        SELECT MIN(population) AS min_population, MAX(population) AS max_population
```

```
    FROM population
    WHERE iso = iso_;
END
```