Outdoor Air Pollution

Members: Asmaa Bashir (30401) - Kemal Ayhan (22511) - Mehmet Emin Er (27748) - Şükrü Baktır (23730) - Atahan Bozkuş (28471)

Repository link: https://github.com/baktirsukru/CS306Project
LogFile Link:
https://github.com/baktirsukru/CS306Project/blob/main/STEP%203/sql_actions_Local_instance_MySQL80.log

Part 1: Analyze Data

a) Create Views:

The following SQL is written to find all (country, year) pairs that have deaths caused by air pollution below 4%.

```
CREATE VIEW percentage_under_4
AS SELECT country_id, percentage, year
FROM percentage_of_deaths_by_air_pollution
WHERE percentage < 4;
```

The following SQL is written to find all (country, year) pairs that have 1000 deaths or above caused by outdoor air pollution.

```
CREATE VIEW causes_of_death_view AS
SELECT country_id, Years, Outdoor_air_pollution
FROM CAUSES_OF_DEATH
WHERE Outdoor_air_pollution > 999;
```

The following SQL is written to find all (country, year) pairs that have death rates above 50% for the age group of 5 and below.

```
CREATE VIEW pollution_rates_age_under_5 AS
SELECT country_id, year, death_under_5
FROM pollution_rates_age
WHERE death_under_5 > 50;
```

The following SQL is written to find all (country, year) pairs that have death rates above 100% for Matter Pollution and 6% OzonePollution.

```
CREATE VIEW Death_rates_from_ozone_and_particulate_matter_polution_VIEW AS
SELECT Entity , Years , MatterPollution , OzonePollution
FROM Death_rates_from_ozone_and_particulate_matter_polution
WHERE MatterPollution >= 100 AND OzonePollution >= 6 ;
```

The following SQL is written to find all (country, year) pairs that concentration above 6e18.

```
CREATE VIEW concentrations_VIEW AS
SELECT country_id, year, concentration
```

```
FROM concentrations_of_air_pollution
WHERE  concentration > 6e18;
```

b) Joins and Set Operators

Since EXCEPT operation is not allowed in MYSQL, instead NOT IN is used to find the (country, year) pairs that have child death rate above > 50 excluding the ones that have deaths caused by air pollution below 4%.

```
SELECT country_id, year FROM pollution_rates_age_under_5
WHERE country_id NOT IN (
   SELECT country_id FROM percentage_under_4
);
```

The following code piece gives the same query result with the one above, yet LEFT JOIN is used instead of NOT IN.

```
SELECT p.country_id, p.year
FROM pollution_rates_age_under_5 p
LEFT JOIN percentage_under_4 u ON p.country_id = u.country_id
WHERE u.country_id IS NULL;
```

c) IN and EXISTS

We ran the code below with IN, and then achieved the same results using EXISTS.

```
SELECT * FROM pollution_rates_age
WHERE country_id IN ( SELECT country_id FROM countries WHERE country_name = 'Turkey');
```

Following code is written to get all the data of Turkey in the pollution_rates_age.

```
SELECT * FROM pollution_rates_age
WHERE EXISTS (
  SELECT 1 FROM countries WHERE country_name = 'Turkey' AND countries.country_id =
pollution_rates_age.country_id);
```

d) Aggregate Operators

The following code is to find average values of MatterPollution, OzonePollution and the percentage of countries where the average percentage is above 10.

```
/* AVG */
SELECT percentage_of_deaths_by_air_pollution.country_id,
AVG(percentage_of_deaths_by_air_pollution.percentage) AS avg_percentage,
AVG(death_rates_from_ozone_and_particulate_matter_polution.MatterPollution) AS
avg_matter_pollution,
AVG(death_rates_from_ozone_and_particulate_matter_polution.OzonePollution) AS
avg_ozone_pollution
FROM percentage_of_deaths_by_air_pollution
LEFT JOIN death_rates_from_ozone_and_particulate_matter_polution
```

```
ON percentage_of_deaths_by_air_pollution.country_id =
death_rates_from_ozone_and_particulate_matter_polution.country_id
GROUP BY percentage_of_deaths_by_air_pollution.country_id
HAVING avg_percentage > 10;
```

The following code is to find the total number of deaths caused by outdoor air pollution and the total number of deaths below the age of 5 in countries where there are a 1000 deaths or more.

```
/* SUM */
SELECT causes_of_death.country_id,
SUM(causes_of_death.Outdoor_air_pollution) AS death_sum,
SUM(pollution_rates_age.death_under_5) AS under5_sum
FROM causes_of_death
LEFT JOIN pollution_rates_age
ON causes_of_death.country_id = pollution_rates_age.country_id
GROUP BY causes_of_death.country_id, causes_of_death.Years
HAVING death_sum > 999;
```

The following code is to find max values of Concentration values of air pollution, Death rates from ozone and particulate matter and comparison of them. For max Concentration we use >= 30.

```
/*MAX*/
SELECT concentrations_of_air_pollution.country_id,
max(concentrations_of_air_pollution.Concentration) AS Max_Concentration,
max(Death_rates_from_ozone_and_particulate_matter_polution.MatterPollution) AS
Max_MatterPollution,
max(Death_rates_from_ozone_and_particulate_matter_polution.OzonePollution) AS
Max_OzonePollution
FROM concentrations_of_air_pollution
LEFT JOIN Death_rates_from_ozone_and_particulate_matter_polution ON
concentrations_of_air_pollution.country_id =
Death_rates_from_ozone_and_particulate_matter_polution.country_id GROUP BY
concentrations_of_air_pollution.country_id
HAVING Max_Concentration >= 30 ;
```

The following code is to find max values of Concentration values of air pollution, Death rates from ozone and particulate matter. For min Concentration we use <= 6e18.

```
/* MIN */
SELECT concentrations_of_air_pollution.country_id,
min(concentrations_of_air_pollution.Concentration) AS Min_Concentration,
min(Death_rates_from_ozone_and_particulate_matter_polution.MatterPollution) AS
Min_MatterPollution,
min(Death_rates_from_ozone_and_particulate_matter_polution.OzonePollution) AS
Min_OzonePollution
FROM concentrations_of_air_pollution
LEFT JOIN Death_rates_from_ozone_and_particulate_matter_polution ON
```

```
concentrations_of_air_pollution.country_id =
Death_rates_from_ozone_and_particulate_matter_polution.country_id
GROUP BY concentrations_of_air_pollution.country_id
HAVING Min_Concentration <= 6e18 ;
```

The following code gives the count of years based on countries where there are at least 10 years where under_5 death rate is above 50. Unfortunately we had issues implementing a JOIN for this part.

```
/*COUNT*/
SELECT pollution_rates_age_under_5.country_id, count(*) AS total
FROM pollution_rates_age_under_5
GROUP BY pollution_rates_age_under_5.country_id
HAVING total > 10 ;
```

Part 2: Constraint and Trigger

The following is to find min and max percentage values in the percentage_of_deaths_by_air_pollution table.

```
SELECT MIN(percentage) AS min_perc, MAX(percentage) AS max_perc FROM
percentage_of_deaths_by_air_pollution;
```

Following code is written to add a constraint that checks if the entered data is between the min and max percentage values.

```
ALTER TABLE percentage_of_deaths_by_air_pollution ADD CONSTRAINT chk_perc_range CHECK
(percentage >= 0.23
AND percentage <= 17.20);
```

The code below is to check whether the constraint works. It gives an error since the average value is above the max value of it.

```
INSERT INTO percentage_of_deaths_by_air_pollution (country_id, year, percentage) VALUES (1, 2020,
18);
```

The code below is to add two triggers which will set the average part of the data which is updated or inserted to max or min value based on whether it is above or below these values.

```
DELIMITER $$
CREATE TRIGGER tr_insert BEFORE INSERT ON percentage_of_deaths_by_air_pollution FOR
EACH ROW
BEGIN
  IF NEW.percentage < 0.24 THEN
    SET NEW.percentage = 0.24;
  ELSEIF NEW.percentage > 17.19 THEN
    SET NEW.percentage = 17.19;
  END IF;
END $$
CREATE TRIGGER tr_update BEFORE UPDATE ON percentage_of_deaths_by_air_pollution FOR
EACH ROW
BEGIN
  IF NEW.percentage < 0.24 THEN
    SET NEW.percentage = 0.24;
  ELSEIF NEW.percentage > 17.19 THEN
    SET NEW.percentage = 17.19;
  END IF;
END $$
DELIMITER ;
```

Following code is to test whether the triggers are working correctly. As a result, it was seen that the value of the percentage which was 18 was set to 17.19.

```
INSERT INTO percentage_of_deaths_by_air_pollution (country_id, year, percentage) VALUES (1, 2020,
18);
```

Part 3: Stored Procedure

Following code is to create a Procedure that will give the data of pollution_rates_age table based on the parameter value.

```
DELIMITER $$
CREATE PROCEDURE get_data(IN c_name VARCHAR(255))
BEGIN
        SELECT * FROM pollution_rates_age
   WHERE pollution_rates_age.country_id = (
   SELECT country_id FROM countries where c_name = countries.country_name);
END $$
DELIMITER ;
```

These two calls were done to test the Procedure which seemed working correctly.

```
CALL get_data("Turkey");
CALL get_data("Albania");
```

Compare creating general constraints and creating trigger methods with pros and cons in terms of enforcing data integrity.

Ease of implementation: General constraints are easy to implement. Trigger methods are difficult to implement and test.

Efficiency/Performance: General constraints improve database performance by preventing the insertion of invalid data. Triggers reduce database performance, not the best choice for simple performance-critical tasks since they run extra code every single time.

Complexity: General constraints do not allow for complex logic to be performed when enforcing constraints. Trigger methods can enforce complex logic or computations to be performed that are difficult/impossible to enforce using general constraints.

Scope: General constraints can be applied to one or more columns in a table. It is possible to enforce constraints on individual columns or the entire table. Triggers can enforce constraints across multiple tables or databases.