# CS306 GROUP 6 PROJECT STEP 3 REPORT

**Burak Ersöz 29378**
**Cansin Narşahin 29126**
**Ozan Parlayan 29283**
**Ufuk Özdek 29498**
**Göktuğ Gökyılmaz 28846**

**Github Repository :** ***https://github.com/burakmert824/CS306_GROUP_PROJECT***
***Log File :***
***https://github.com/burakmert824/CS306_GROUP_PROJECT/blob/main/step3/logFiles.txt***
***SQL Files :***
***https://github.com/burakmert824/CS306_GROUP_PROJECT/tree/main/step3***

In project step 3, we mainly did 3 things. Firstly we create 5 view that are:

## Question 1)

## -- 1.a part

**-- 1) In this view, we have found the country codes and their average obesity rates.**

```
CREATE VIEW avg_obesity_rate as
(SELECT ob.C_CODE, AVG(ob.death_rate) AS avg_deathrate
FROM obesity_report ob,countries c
WHERE c.C_Code = ob.C_Code and ob.YYear BETWEEN 2000 AND 2010
GROUP BY C_CODE);
```

**-- 2) In this view, we have founded the countries that meet two conditions: having more than 2500 deaths and lower than the average life expectancy. We have also retrieved the corresponding years for these countries.**

```
CREATE VIEW high_airpol_low_lifeex AS
SELECT a.C_Code, a.YYear, a.deathcounts, a.Household_FosilFuel
FROM airpol_occure a
INNER JOIN LifeExpact e ON a.C_Code = e.C_Code AND a.YYear = e.YYear
WHERE a.deathcounts > 2500 AND e.Life_ex_value < (SELECT AVG(Life_ex_value) FROM
LifeExpact);
```

**-- 3) Below code shows the countries with a life expectancy below the average and drug deaths greater than 100.**

```
CREATE VIEW high_drugdeaths_low_lifeex AS
SELECT s.C_Code, s.YYear, s.drugdeaths, e.Life_ex_value
FROM substance_use s
```

```
INNER JOIN LifeExpact e ON s.C_Code = e.C_Code AND s.YYear = e.YYear
WHERE s.drugdeaths > 100 AND e.Life_ex_value < (SELECT AVG(Life_ex_value) FROM
LifeExpact);
```

**-- 4) This view shows the countries that have average life_exp over and equal to 75 from 1990 and 2012**

```
CREATE VIEW high_life_exp_countries AS
SELECT
   c.C_Name AS Country,
   AVG(l.Life_ex_value) AS Avg_Life_Expectancy
FROM countries c
JOIN LifeExpact l ON c.C_Code = l.C_Code
WHERE l.YYear BETWEEN 1990 AND 2012
GROUP BY c.C_Name
HAVING AVG(l.Life_ex_value) >= 75;
```

**-- 5) Below view statement shows the countries that have under 80 deaths per 100k people from smoking**

```
CREATE VIEW low_death_rate_countries AS
SELECT c.C_Name AS Country,
   AVG(s.death_rate_per_100000_people) AS Avg_Death_Rate_Per_100k_People
FROM countries c
JOIN Smoke_Examine s ON c.C_Code = s.C_code
WHERE s.YYear BETWEEN 1990 AND 2012
GROUP BY c.C_Name
HAVING AVG(s.death_rate_per_100000_people) <= 80;
```

## -- 1.B Part

Then in part B we use set operators between two views in our SQL statement.

**--  Below statement shows the union of the countries with high life expectancy and low death rate from smoking  (it uses 4th and 5th view statements)**

```
SELECT Country
FROM high_life_exp_countries
UNION
```

```
SELECT Country
FROM low_death_rate_countries;
```

**-- Below statement shows the outer join of the countries with high life expectancy and low death rate from smoking ( to do that unions lift and right joins) The UNION operator and the simplified FULL OUTER JOIN simulation should return the same count of tuples.**

```
SELECT h.Country
FROM high_life_exp_countries h
LEFT JOIN low_death_rate_countries l ON h.Country = l.Country
WHERE l.Country IS NULL
UNION
SELECT l.Country
FROM high_life_exp_countries h
RIGHT JOIN low_death_rate_countries l ON h.Country = l.Country
WHERE h.Country IS NULL
UNION
SELECT h.Country
FROM high_life_exp_countries h
JOIN low_death_rate_countries l ON h.Country = l.Country;
```

## -- 1.C Part

```
SELECT h.Country
FROM high_life_exp_countries h
WHERE h.Country IN ( SELECT l.Country
                     FROM low_death_rate_countries l );

SELECT h.Country
FROM high_life_exp_countries h
WHERE EXISTS ( SELECT 1
               FROM low_death_rate_countries l
               WHERE h.Country = l.Country);
```

## -- 1.D Part

In this part we **use all five aggregate operators (SUM, AVG, COUNT, MIN, MAX) in a 5 SELECT statements,**

**-- 1) It counts the number of countries in 2000 to 2010 and if their life expectancy average is greater then 75 and their death rate below then 9.66.**

```
SELECT COUNT(*) AS num_countries
FROM (
  SELECT c.C_Code, AVG(ob.death_rate) AS avg_death_rate
  FROM countries c
  JOIN obesity_report ob ON c.C_Code = ob.C_code
  JOIN high_life_exp_countries hlec ON c.C_Name = hlec.Country
  WHERE ob.YYear BETWEEN 2000 AND 2010
  GROUP BY c.C_Code
  HAVING avg_death_rate > 9.66
) AS t;
```

**– 2) This query calculates the following aggregate values for each country where the average consumption per smoker per day is greater than 10, use all five aggregate operators (SUM, AVG, COUNT, MIN, MAX) in a SELECT statement, joining the Smoke_Examine and LifeExpact tables using the countries table. The results are grouped by country and ordered by the total number of deaths in descending order.**

```
SELECT
    c.C_Name AS Country,
    COUNT(s.YYear) AS NumberOfYears,
    AVG(s.consumption_per_smoker_per_day) AS Avg_Consumption_Per_Smoker_Per_Day,
    SUM(s.deaths) AS Total_Deaths,
    MIN(l.Life_ex_value) AS Min_Life_Expectancy,
    MAX(l.Life_ex_value) AS Max_Life_Expectancy
FROM countries c
JOIN Smoke_Examine s ON c.C_Code = s.C_code
JOIN LifeExpact l ON c.C_Code = l.C_Code AND s.YYear = l.YYear
WHERE s.YYear BETWEEN 1990 AND 2000
GROUP BY c.C_Name
HAVING AVG(s.consumption_per_smoker_per_day) > 10
ORDER BY Total_Deaths DESC;
```

**– 3) below code sum the death numbers caused by air pollution of the country between 2000 and 2010**

```
SELECT c.C_Code, c.C_Name, SUM(a.deathcounts) AS total_deathcounts
FROM airpol_occure a
JOIN countries c ON c.C_Code = a.C_Code
WHERE a.YYear BETWEEN 2000 AND 2010
GROUP BY a.C_Code, c.C_Name
HAVING SUM(a.deathcounts) > 0;
```

**– 4) Below code shows the maximum death number caused by drug use for countries in their history**

```
SELECT s.C_Code, s.YYear, s.drugdeaths
FROM substance_use s
INNER JOIN (
   SELECT C_Code, MAX(drugdeaths) AS max_deaths
   FROM substance_use
   GROUP BY C_Code
   HAVING MAX(drugdeaths) > 0
) AS m
ON s.C_Code = m.C_Code AND s.drugdeaths = m.max_deaths;
```

**– 5) Below code shows the minimum death number caused by alcohol use for countries in their history**

```
SELECT s.C_Code, s.YYear, s.alcoholdeaths
FROM substance_use s
INNER JOIN (
   SELECT C_Code, MIN(alcoholdeaths) AS min_deaths
   FROM substance_use
   GROUP BY C_Code
   HAVING MIN(alcoholdeaths) > 0
) AS m ON s.C_Code = m.C_Code AND s.alcoholdeaths = m.min_deaths;
```

## Question 2)

**We have set up constraints and triggers for a numeric field in the "obesity_report" table.We added a constraint called "check_year_range" to ensure only values within the specified range are entered. "Before insert" and "before update" triggers fix values outside the range to the minimum or maximum value. These constraints and triggers maintain data integrity and consistency in the table.**

```
ALTER TABLE obesity_report
ADD CONSTRAINT check_year_range
CHECK (YYear BETWEEN 0 AND 2023);

delimiter //
-- create trigger for inserting
CREATE TRIGGER fix_year_range_insert BEFORE INSERT ON obesity_report
FOR EACH ROW
```

```
BEGIN
  IF NEW.YYear < 0 THEN
    SET NEW.YYear = 0;
  ELSEIF NEW.YYear > 2023 THEN
    SET NEW.YYear = 2023;
  END IF;
END //

-- create trigger for updating
CREATE TRIGGER fix_year_range_update
BEFORE UPDATE ON obesity_report
FOR EACH ROW
BEGIN
  IF NEW.YYear < 0 THEN
    SET NEW.YYear = 0;
  ELSEIF NEW.YYear > 2023 THEN
    SET NEW.YYear = 2023;
  END IF;
END //
delimiter ;
```

## Question 3)

**We created a stored procedure named "get_max_obesity_rate_of_country" that takes one parameter, "iso_code", checks its validity, and retrieves the maximum death rate value and its corresponding year for the specified country code from the "obesity_report" table. The procedure outputs a concatenated message string containing the retrieved values. Overall, the stored procedure provides a convenient and reusable way to retrieve the maximum death rate value and its year for a specified country code.**

```
delimiter //
CREATE PROCEDURE get_max_obesity_rate_of_country(IN iso_code VARCHAR(32))
BEGIN
    DECLARE death_rate_max FLOAT;
    DECLARE death_rate_year INT;
    DECLARE iso_code_count INT;

    SELECT COUNT(*) INTO iso_code_count FROM obesity_report WHERE C_Code =
iso_code;
    -- give error if iso code with given value is not exist in the obesity_report table
    IF iso_code_count = 0 THEN
        SELECT "Invalid parameter value" AS message;
```

```
   ELSE
        SELECT death_rate,YYear INTO death_rate_max, death_rate_year
        FROM obesity_report where C_code = iso_code
                        and death_rate =(SELECT MAX(death_rate) from obesity_report where
C_code = iso_code);
                SELECT CONCAT('Maximum death rate of ',iso_code,': ', death_rate_max, ',
Year: ', death_rate_year) AS message;
   END IF;
END//
delimiter ;
```