

**CS 306 DATABASE SYSTEMS – PROJECT STEP 3**  
**COUNTRIES, POLLUTIONS AND SOLUTIONS**  
**GROUP 22**

- Ahmet Alperen Güngör – 28847
- Anıl Arslan – 29468
- Batuhan İşildak - 29181
- Dediş Atakan Öz – 29414
- Eren Altın – 28986

[https://github.com/batuhanisildak-malwation/CS306\\_Group22](https://github.com/batuhanisildak-malwation/CS306_Group22)

**Views:**

They are in the views directory in github.

**GetCountryAirPollutionByYear**

- Finding the countries' amount of air pollution categorized by years, by joining Pollution, Country and Year tables (with same Country Ids) where Pollution type is Air Pollution

```
create definer = root@`%` view GetCountryAirPollutionByYear as
select `C`.`name` AS `Country Name`, `P`.`amount` AS `Amount`, `P`.`amountUnit` AS
`Amount Unit`, `Y`.`value` AS `Year`
from ((`test-db`.`Pollution` `P` join `test-db`.`Country` `C` on ((`P`.`countryId` = `C`.`id`))) join
`test-db`.`Year` `Y` on ((`P`.`yearId` = `Y`.`id`)))
where (`P`.`type` = 'Air Pollution');
```

**GetCountryPlasticPollutionByYear**

- Finding the countries' amount of plastic pollution categorized by years, by joining Pollution, Country and Year tables (with same Country Ids) where Pollution type is Plastic Pollution

```
create definer = root@`%` view GetCountryPlasticPollutionByYear as
select `C`.`name` AS `Country Name`, `P`.`amount` AS `Amount`, `P`.`amountUnit` AS
`Amount Unit`, `Y`.`value` AS `Year`
from ((`test-db`.`Pollution` `P` join `test-db`.`Country` `C`
on ((`P`.`countryId` = `C`.`id`))) join `test-db`.`Year` `Y` on ((`P`.`yearId` = `Y`.`id`)))
where (`P`.`type` = 'Plastic Pollution');
```

**GetCountryTotalDeathByYear**

- Finding the countries' total death count categorized by years, by joining Country and Death tables (with same Country Ids) and using SUM of different death cause types.

```
create view GetCountryTotalDeathByYear as
SELECT C.name AS 'Country', SUM(D.count) AS 'Total Death' FROM Death D
INNER JOIN Country C ON C.id = D.countryID
GROUP BY D.countryId;
```

### **GetDeathCountByDeathCause**

- Finding number of deaths for different death causes, using SUM of different years and countries' death count under each Death Causes.

```
create view GetDeathCountByDeathCause as
SELECT DC.name AS 'Death Cause', SUM(D.count) AS 'Total Death' FROM Death D
JOIN DeathCause DC on DC.id = D.deathCauseId
GROUP BY D.deathCauseId
ORDER BY 2 DESC
```

### **GetMinDeathCountByDeathCause**

- Finding which death cause caused least number of deaths, by using a subquery that has the total death count (using SUM, like the above view), then using MIN of total death count (rows with the lowest number of deaths).

```
create view GetMinDeathCountByDeathCause as
Select * from (SELECT DC.name AS 'Death Cause', SUM(D.count) AS 'Total Death' FROM
Death D
JOIN DeathCause DC on DC.id = D.deathCauseId
GROUP BY D.deathCauseId
ORDER BY 2 DESC) as subq where subq.`Total Death`=(SELECT MIN(`Total Death`) as cnt
FROM
(SELECT DC.name AS 'Death Cause', SUM(D.count) AS 'Total Death' FROM Death D
JOIN DeathCause DC on DC.id = D.deathCauseId
GROUP BY D.deathCauseId
ORDER BY 2 DESC) as SubQ);
```

### **GetMostCommonDeathCauseByCountry**

- Finding the most common cause of death of each country, by using Death, Death Cause and Country tables. Subqueries and MAX is used in order to filter the data (select rows with the highest number of deaths) to show most common death cause.

```
create view GetMostCommonDeathCauseByCountry as
SELECT SQ.Country AS 'Country',
       SQ.Cause AS 'Cause',
       MAX(SQ.Count) AS 'Max Count'
FROM (
  SELECT C.name AS 'Country',
         DC.name AS 'Cause',
```

```

        SUM(D.count) AS 'Count'
    FROM Death D
    INNER JOIN DeathCause DC ON D.deathCauseId = DC.id
    INNER JOIN Country C ON D.countryId = C.id
    GROUP BY C.name, DC.name
) AS SQ
GROUP BY SQ.Country, SQ.Cause
HAVING MAX(SQ.Count) = (
    SELECT MAX(SQ2.Count)
    FROM (
        SELECT C.name AS 'Country',
              DC.name AS 'Cause',
              SUM(D.count) AS 'Count'
        FROM Death D
        INNER JOIN DeathCause DC ON D.deathCauseId = DC.id
        INNER JOIN Country C ON D.countryId = C.id
        GROUP BY C.name, DC.name
    ) AS SQ2
    WHERE SQ2.Country = SQ.Country
) ORDER BY Country ASC;

```

#### **GetCountriesWithAvgSolarEnergyAmountMoreThanTenByYear**

- Finding countries that has average solar energy amount more than 10 kwh, using COUNT (number of countries) and AVG (average of solar energy).

```

create view GetCountriesWithAvgSolarEnergyAmountMoreThanTenByYear as
SELECT COUNT(*) as CountryCount FROM Country C
WHERE C.Id IN (SELECT SubQ.countryId FROM (SELECT AVG(amount) AS AverageAmount,
countryId FROM CountryRenewableEnergySource CRES
WHERE CRES.renewableEnergySourceId = 2
GROUP BY CRES.countryId) AS SubQ WHERE SubQ.AverageAmount > 10
);

```

#### **Aggregate Operators:**

Different operators are used in the following views:

**SUM:** GetCountryTotalDeathByYear, GetDeathCountByDeathCause,  
GetMinDeathCountByDeathCause, GetMostCommonDeathCauseByCountry

**MIN:** GetMinDeathCountByDeathCause

**MAX:** GetMostCommonDeathCauseByCountry

**AVG:** GetCountriesWithAvgSolarEnergyAmountMoreThanTenByYear

**COUNT:** GetCountriesWithAvgSolarEnergyAmountMoreThanTenByYear

#### In and Exists:

- To find countries that use more than 10 kwh solar energy, both in and exist are used in different codes, and outputs are compared – they have the same output: 44.

##### Exists:

```
SELECT COUNT(*) AS CountryCount FROM Country C
WHERE EXISTS (SELECT SubQ.countryId FROM (SELECT AVG(amount) AS AverageAmount,
countryId FROM CountryRenewableEnergySource CRES
WHERE CRES.renewableEnergySourceId = 2 AND C.Id = CRES.countryId
GROUP BY CRES.countryId
HAVING AverageAmount > 10
) AS SubQ)
```

##### In:

```
SELECT COUNT(*) as CountryCount FROM Country C
WHERE C.Id IN (SELECT SubQ.countryId FROM (SELECT AVG(amount) AS AverageAmount,
countryId FROM CountryRenewableEnergySource CRES
WHERE CRES.renewableEnergySourceId = 2
GROUP BY CRES.countryId) AS SubQ WHERE SubQ.AverageAmount > 10
);
```

They are in the in\_exists\_test directory in repository.  
Screenshots of the results are also included in the repository.

#### Constraints and Triggers:

Constraints enforce data integrity and rules, while triggers can manipulate – change the data in multiple ways like updating.

- **Trigger:**  
CREATE TRIGGER year\_check\_insert BEFORE INSERT ON Year  
FOR EACH ROW  
BEGIN  
IF NEW.value < 1900 THEN  
SET NEW.value = 1900;  
ELSEIF NEW.value > 2100 THEN  
SET NEW.value = 2100;  
END IF;  
END;  
  
CREATE TRIGGER year\_check\_update BEFORE UPDATE ON Year

```

FOR EACH ROW
BEGIN
  IF NEW.value < 1900 THEN
    SET NEW.value = 1900;
  ELSEIF NEW.value > 2100 THEN
    SET NEW.value = 2100;
  END IF;
END;/

```

- **Constraints:**

```
ALTER TABLE Year ADD CONSTRAINT check_year_unique UNIQUE (value);
```

They are in constraint\_and\_triggers directory in repository.

Action logs are included in the github repository.

In addition, screenshots of the results are also included in the repository.

### Procedure:

Views that can be changed by parameters are created, so that outputs can be changed by different user inputs.

- **Input year values can be changed**

```

CREATE PROCEDURE getYearValue
  (IN yearId INT)
BEGIN
  SELECT value
  FROM Year
  WHERE Id = yearId;
END

```

```
CALL getYearValue(1);
```

- **Input country values can be changed**

```

CREATE PROCEDURE GetMostCommonDeathCauseByCountryNameProcedure
  (IN countryName VARCHAR(255))
BEGIN
  SELECT SQ.Country AS 'Country',
         SQ.Cause AS 'Cause',
         MAX(SQ.Count) AS 'Max Count'
  FROM (
    SELECT C.name AS 'Country',
           DC.name AS 'Cause',
           SUM(D.count) AS 'Count'
    FROM Death D
    INNER JOIN DeathCause DC ON D.deathCauseId = DC.id
    INNER JOIN Country C ON D.countryId = C.id
  )

```

```

        GROUP BY C.name, DC.name
    ) AS SQ
    GROUP BY SQ.Country, SQ.Cause
    HAVING MAX(SQ.Count) = (
        SELECT MAX(SQ2.Count)
        FROM (
            SELECT C.name AS 'Country',
                   DC.name AS 'Cause',
                   SUM(D.count) AS 'Count'
            FROM Death D
            INNER JOIN DeathCause DC ON D.deathCauseId = DC.id
            INNER JOIN Country C ON D.countryId = C.id
            GROUP BY C.name, DC.name
        ) AS SQ2
        WHERE SQ2.Country = SQ.Country
    ) AND Country = countryName;
END

```

They are in procedures directory in repository.

Screenshots of different input values (for country, Uganda and Turkey; for year, 1950 and 1951) are included in github