

[◀ Return to Classroom](#)

Deforestation Exploration

REVIEW

HISTORY

Meets Specifications

YOU MADE IT

Hello Dorothy,

Congratulations on completing your **Deforestation Exploration** project consisting of a great forestation **VIEW** with correct queries, you should be proud of yourself for making this project a success and we encourage you to continue with this momentum, zeal, and great achievements await you in this field. Wish you all the best as you continue learning new skills.

Regards!!!

STAY SAFE.

Building A View

The create a **forestation** view query that the student writes prior to answering the questions joins all three tables on the columns indicated, and creates a new column by performing a calculation that compares two columns.

Awesome

Great job creating your forestation **view** query using the CREATE concept which *guides you through the rest of your queries*.

- You can [Refer View](#) for more insights on creating **VIEW**.

```
CREATE VIEW forestation AS
  SELECT fa.country_code,
         fa.country_name,
         fa.year,
         fa.forest_area_sqkm,
         la.total_area_sq_mi,
         (la.total_area_sq_mi * 2.59) total_area_sqkm,
         r.region,
         r.income_group,
         ROUND(CAST(fa.forest_area_sqkm / (la.total_area_sq_mi * 2.59) * 100 AS
numeric), 2) forest_to_land_percentage
  FROM forest_area fa
  JOIN land_area la
  ON fa.country_code = la.country_code
  AND fa.year = la.year
  JOIN regions r ON r.country_code = la.country_code;
```

Basic SQL Queries

Each query is included in the Appendix and executes properly. A reviewer should be able to execute this same query and get the correct output.

Awesome

Terrific work understanding basic SQL concepts and providing queries which all executed properly giving the right outputs. This will help you in the future in understanding *big data analysis*.

`SELECT` queries return results consistent with the question being asked.

Awesome

The `SELECT` clause is the most fundamental concept to master in SQL and so always keep it concise and simple. Your `SELECT` queries are used correctly.

- [Here](#) is a resource where you can learn more about using the **SELECT** statement in queries.

`WHERE` clauses used in `SELECT` statements filter tables according to the questions being asked

The `WHERE` clause is your **go-to filter clause** and is used in almost every query. You have used the clauses in the `SELECT` statement appropriately and well formatted.

`ORDER BY` clauses used in `SELECT` statements sort query results according to the questions being asked, and specify `ASC` for ascending or `DESC` for descending where appropriate

`ORDER BY` is often used when **prioritizing problems**.

Well done, you used them correctly

`GROUP BY` clauses aggregate results by chosen categorical variables

Great job in mastering `GROUP BY` which is important when **synthesizing information**.

You have used it to sort the right queries for the Regions.

Queries make use of operators such as =, < and/or > to qualify `WHERE` clauses and `JOIN` statements, as well as conditions AND and OR to link conditional clauses.

Boolean operators are used when **slicing and dicing the data** to solve a specific problem at hand.

Good work in using these operators to qualify your `WHERE` clauses and `JOIN` statements.

Windows Functions

Queries make use of Windows Functions such as SUM, COUNT, ROUND and/or ABS as needed to perform the appropriate calculation in order to answer the questions posed.

Awesome

Good job creating windows functions which are very tricky and similar to aggregate functions but **retain their number of rows in the output**.

- Check out [this additional resource](#) for more on Windows functions

```
WITH source_table AS (  
  SELECT a.region,  
         ROUND(CAST((SUM(a.forest_area_sqkm) / SUM(a.total_area_sqkm) * 100) AS numeric),  
2) forest_percentage_1990,  
         ROUND(CAST((SUM(b.forest_area_sqkm) / SUM(b.total_area_sqkm) * 100) AS numeric),  
2) forest_percentage_2016  
  FROM forestation a  
  JOIN forestation b  
  ON a.region=b.region  
  WHERE b.year=2016 AND a.year=1990  
  GROUP BY 1, a.year  
)  
  
SELECT region, forest_percentage_2016  
FROM source_table  
WHERE region = 'World';
```

Join Command

Queries include the appropriate form of Join (Inner, Left, Right, Outer) clause to ensure that no necessary rows are left out.

JOINS help **stitch tables together that have complementary information**

Impressive job mastering how to join tables using a *common key*.

This indeed shows that you know what is required. 🙌

The student creates a query that joins a table to itself in order to compare values in two different rows.

Queries include Join clauses that match appropriate columns together using the ON command and the appropriate Boolean operator.

Good job using JOIN clauses and appropriate Boolean operators in your queries as required.

Case Command

The query the student writes for question 3(c) includes a `CASE` statement that addresses the question.

Awesome

You have done terrific work using the `CASE` statement which is **advanced SQL used when creating a column based on an existing column's value.**

- [Here](#) is a resource you can consult to learn more on using CASE statements.

```
WITH quartiles_table AS (SELECT country_name,  
                             forest_to_land_percentage,  
                             CASE WHEN forest_to_land_percentage BETWEEN 0 AND 25 THEN '0-25%'  
                             WHEN forest_to_land_percentage BETWEEN 25 and 50 THEN '25-50%'  
                             WHEN forest_to_land_percentage BETWEEN 50 AND 75 THEN '50-75%'  
                             ELSE '75-100%' END AS quartiles  
                             FROM forestation  
                             WHERE year = 2016 AND forest_to_land_percentage IS NOT NULL  
                             GROUP BY 1, 2)
```

```
SELECT quartiles, COUNT(*)  
FROM quartiles_table  
GROUP BY 1  
ORDER BY 1;
```

quartiles	count
0-25%	85
25-50%	73
50-75%	38
75-100%	9

Report Formatting

All five elements of the rubric are present in the report.

1. GLOBAL SITUATION
2. REGIONAL OUTLOOK
3. COUNTRY-LEVEL DETAIL
4. RECOMMENDATIONS
5. APPENDIX: SQL queries used



All five elements of the report are provided 🙌

- All queries captured in the Appendix follow SQL formatting guidelines, including those for indentation, capitalization.
- All queries are properly formatted using best practices syntax
- All queries run without errors



Well done with your queries having *SQL formatting*

- You can however visit [this resource](#) to improve your formatting skills.

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)