## **Meets Specifications**

## **Congratulations !!**

Congratulations on completing the Deforestation Exploration project, your efforts have paid off   
Fantastic work on this submission! I was delighted to see the results of all your learnings come to play in this project.

.  
*What stands out in your submission is the logic you used in writing your queries, most especially the solution you came up with for question 3C. You properly partitioned the countries into their respective quartiles based on their forest percentage value, this was impressive. Also, you made some brilliant recommendations. Keep this up *

.

## **Suggestion**

Check this unique way of coming up with the solution for question 3c below

SELECT FLOOR(forest\_percentage/25) quartile, count(\*)

FROM forestation

WHERE year=2016

AND forest\_percentage IS NOT NULL

AND country<>'World'

GROUP BY 1;

.

## **Extra Materials**

Here are some resources you might find useful for more insight and further learning.

* [Database normalization](https://en.wikipedia.org/wiki/Database_normalization)
* [SQL Interview Questions](https://www.interviewbit.com/sql-interview-questions/)
* [46 Questions on SQL to test a data science professional (Skill test Solution)](https://www.analyticsvidhya.com/blog/2017/01/46-questions-on-sql-to-test-a-data-science-professional-skilltest-solution/)

.  
**You put in a lot of effort to meet all the requirements for this project.  
It was a pleasure reviewing your work, all the best with your next project, and have a great day!**

## **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.

Great job! You have applied the concept of CREATE by joining the existing tables and creating a single view. This will now serve as your single source of truth throughout the rest of the project.

* You could refer to [Microsoft Documentation](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15() For more info on creating views.

.

#### THE FOLLOWING A SOME OF THE USE CASES OF VIEWS

* As a security mechanism by allowing users to access data through the view, without granting the users permission to directly access the underlying base tables.
* To provide a backward compatible interface to emulate a table whose schema has changed.

## **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!**

I tested your SQL queries on the Udacity workspace they are running as expected and give the appropriate results.

SELECT queries return results consistent with the question being asked.

## **Awesome !!**

I like the fact that you have properly limited and ordered the output of your queries based on the question it is trying to answer

## **TIPS**

The SELECT clause specifies the columns from which data values are to be retrieved by the query. Data retrieval is limited to the columns specified. When selecting from two or more tables having duplicate column names, it may be necessary to qualify column names with table or view names.

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

The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition. In your current submission of the project, you had perfectly taken advantage of the WHERE clause to filter the results. 

## **TIPS**

The SQL WHERE clause is something you must master if you wish to use SQL for working with data. It is arguably one of the most basic and must-learn constructs of SQL. For more insight into learning, refer to the below

* [Write a WHERE Clause in SQL](https://learnsql.com/blog/where-clause-in-sql/)

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 clauses are used when we want to prioritize the problem. You had perfectly used the ORDER BY clause to solve the REGION database problems and provide accurate results as per the question asked. 

## **TIPS**

The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns. Some databases sort the query results in ascending order by default.

GROUP BY clauses aggregate results by chosen categorical variables

Awesome! GROUP BY is important when you are synthesizing information and comes up very often in real-world applications. Great job in mastering it

.

## **TIPS**

The GROUP BY statement in SQL is used to arrange identical data into groups with the help of some functions. i.e if a particular column has the same values in different rows then it will arrange these rows in a group.

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.

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

## **TIPS**

Operators are the foundation of any programming language. We can define operators as symbols that help us to perform specific mathematical and logical computations on operands. In other words, we can say that an operator operates the operands. SQL operators have three different categories.

* Arithmetic operator
* Comparison operator
* Logical operator

## **Join Command**

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

Though you correctly used JOIN in your queries, you didn't specify the kind of JOIN (INNER/LEFT/RIGHT/OUTER) in the queries. It is good practice to use the appropriate type of Join.

.

## **TIPS**

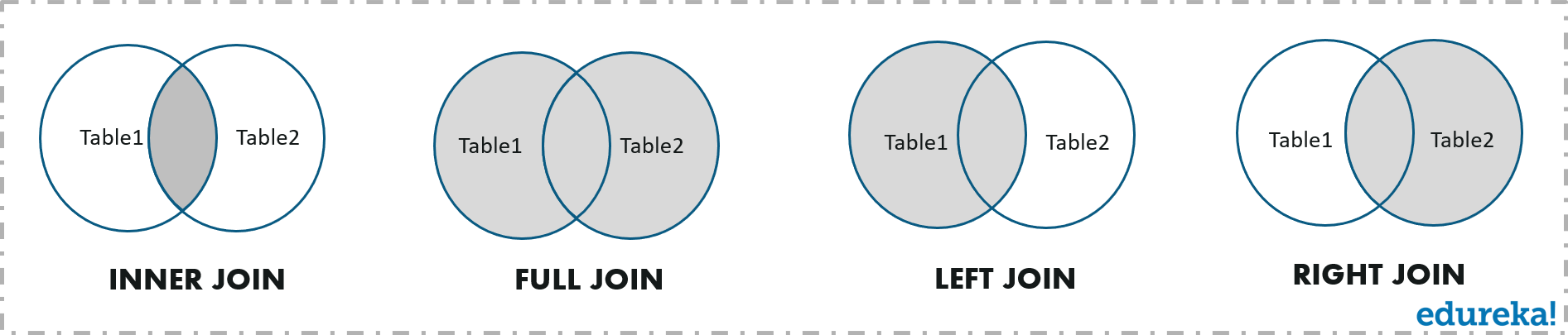
Make it a part of you to always indicate the appropriate type of join in all your queries.

There are mainly four types of joins that you need to understand. They are:

* [INNER JOIN](https://www.edureka.co/blog/sql-joins-types#INNER%20%20Join)
* [FULL JOIN](https://www.edureka.co/blog/sql-joins-types#FULL%20%20Join)
* [LEFT JOIN](https://www.edureka.co/blog/sql-joins-types#LEFT%20%20Join)
* [RIGHT JOIN](https://www.edureka.co/blog/sql-joins-types#RIGHT%20%20Join)

.

#### YOU CAN REFER TO THE BELOW IMAGE.



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

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

A self-join is a query that joins a table to itself in order to compare values in two different rows. For example, you could join the forestation VIEW you created to itself like this below

.

#### TO LEARN MORE ABOUT SELF JOINS, YOU SHOULD REFER TO THE LINK BELOW

* [What Is a Self Join in SQL? An Explanation With Seven Examples](https://learnsql.com/blog/what-is-self-join-sql/)

## **Case Command**

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

Really impressive! CASE WHEN is an advanced SQL topic. It’s typically used when creating a column based on an existing column’s value. Great work mastering this advanced topic!

## **TIPS**

The CASE statement is SQL's way of handling if/then logic. The CASE statement is followed by at least one pair of WHEN and THEN statements—SQL's equivalent of IF/THEN in Excel.

.

#### REFER TO THE BELOW TO LEARN MORE ABOUT THE CASE STATEMENT

* [The SQL CASE statement](https://mode.com/sql-tutorial/sql-case/#the-sql-case-statement)

## **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

Wonderful work! You have completed all parts of the project, covering basic and advanced SQL concepts.

.

#### THE FOLLOWING ARE ALL INCLUDED IN THE REPORT

* GLOBAL SITUATION 
* REGIONAL OUTLOOK 
* COUNTRY-LEVEL DETAIL 
* RECOMMENDATIONS 
* APPENDIX: SQL queries used 
* 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

It is also good practice that you properly format your code with an appropriate capitalization of the SQL clauses (SELECT, FROM, WHERE, CASE, JOIN, GROUP BY, etc), and proper indentations.