1

SELECT place, longitiude, depth

FROM earthquakes\_table

WHERE depth > 10;

SELECT place, longitude, depth: This line specifies the columns you want to retrieve from the table. In this case, you want to retrieve the "place," "longitude," and "depth" columns.

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table."

WHERE depth > 10: This line is a condition or filter that narrows down the results based on the specified criteria. Here, you are retrieving rows where the value in the "depth" column is greater than 10.

Overall, this query retrieves the "place," "longitude," and "depth" values from the "earthquakes\_table" for rows where the depth is greater than 10.

A screenshot of a computer

Description automatically generated with medium confidence

2

select

longitiude - latitude as Diff,

depth - deptherror as DepthDist

FROM earthquakes\_table ;

SELECT longitude - latitude AS Diff, depth - deptherror AS DepthDist: This line specifies the columns you want to retrieve from the table. In this case, you are performing calculations on the columns "longitude," "latitude," "depth," and "deptherror." The calculated difference between "longitude" and "latitude" is aliased as "Diff," and the difference between "depth" and "deptherror" is aliased as "DepthDist."

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table."

Overall, this query retrieves the calculated differences between "longitude" and "latitude" as "Diff" and the differences between "depth" and "deptherror" as "DepthDist" from the "earthquakes\_table."

A screenshot of a data output

Description automatically generated with medium confidence

3

SELECT place, COUNT(\*) as earthquakes\_no , AVG(mag) as avg\_mag

FROM earthquakes\_table

GROUP BY place

HAVING COUNT(\*) = 10

ORDER BY earthquakes\_no DESC;

SELECT place, COUNT(\*) as earthquakes\_no, AVG(mag) as avg\_mag: This line specifies the columns you want to retrieve from the table. You are selecting the "place" column, counting the occurrences of each place and aliasing it as "earthquakes\_no," and calculating the average value of the "mag" column and aliasing it as "avg\_mag."

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table."

GROUP BY place: This clause groups the rows based on the "place" column. This allows the subsequent aggregation functions (COUNT and AVG) to be applied per distinct "place."

HAVING COUNT(\*) = 10: This clause filters the grouped rows and includes only those groups where the count of earthquakes for each "place" is exactly 10. It acts as a condition on the grouped results.

ORDER BY earthquakes\_no DESC: This line sorts the result set in descending order based on the "earthquakes\_no" column. This means that places with the highest count of earthquakes (equal to 10) will appear first in the result.

Overall, this query retrieves the "place," the count of earthquakes for each place, and the average magnitude of earthquakes for places where the count is exactly 10. The result is then sorted in descending order based on the count of earthquakes.

A screenshot of a computer

Description automatically generated with medium confidence

4

BEGIN;

EXPLAIN

select \* from public.earthquakes\_table ;

The BEGIN; statement starts a transaction block, which is typically used when you want to execute multiple SQL statements as part of a transaction. Transactions ensure that a set of related database operations is treated as a single unit and can be committed or rolled back as a whole.

The EXPLAIN command is then used to analyze the execution plan of the subsequent SELECT statement. It provides information about how the database intends to execute the query, including the operations, indexes, join methods, and estimated costs involved.

The SELECT \* FROM public.earthquakes\_table; statement retrieves all columns from the "earthquakes\_table" in the "public" schema. By running the EXPLAIN command before it, you can see the execution plan without actually fetching the result set.

A screenshot of a computer

Description automatically generated with medium confidence

5

SELECT \*

FROM earthquakes\_table

ORDER BY mag ASC

LIMIT 10;

SELECT \*: This line specifies that you want to retrieve all columns from the "earthquakes\_table". The asterisk (\*) is a wildcard that represents all columns.

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table".

ORDER BY mag ASC: This line specifies the ordering of the result set. The "mag" column is used for sorting in ascending order (ASC), meaning the smallest values will appear first.

LIMIT 10: This line limits the number of rows returned to 10. Only the first 10 rows, according to the specified ordering, will be included in the result set.

Overall, this query retrieves the first 10 rows from the "earthquakes\_table", sorted in ascending order based on the "mag" column (magnitude). By using the LIMIT clause, you ensure that only 10 rows are returned, starting from the smallest magnitude values.

A screenshot of a computer

Description automatically generated with medium confidence

6

SELECT \*

FROM earthquakes\_table

WHERE latitude < 30 AND longitiude < -100

ORDER BY mag ASC

LIMIT 12;

SELECT \*: This line specifies that you want to retrieve all columns from the "earthquakes\_table". The asterisk (\*) is a wildcard that represents all columns.

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table".

WHERE latitude < 30 AND longitude < -100: This line specifies a condition or filter for the query. It retrieves rows where the latitude is less than 30 and the longitude is less than -100. This condition narrows down the results to earthquakes located in a specific geographical area.

ORDER BY mag ASC: This line specifies the ordering of the result set. The "mag" column is used for sorting in ascending order (ASC), meaning the smallest magnitude values will appear first.

LIMIT 12: This line limits the number of rows returned to 12. Only the first 12 rows, according to the specified ordering, will be included in the result set.

Overall, this query retrieves the first 12 rows from the "earthquakes\_table" that meet the condition of having a latitude less than 30 and a longitude less than -100. The result set is then sorted in ascending order based on the "mag" column (magnitude).

A screenshot of a computer

Description automatically generated with medium confidence

7

SELECT place, mag

FROM earthquakes\_table

WHERE latitude > 30 AND longitiude < -100

AND mag > 3.0

ORDER BY mag ASC

LIMIT 13;

SELECT place, mag: This line specifies that you want to retrieve the "place" and "mag" columns from the "earthquakes\_table".

FROM earthquakes\_table: This line indicates the table from which you want to retrieve the data. In this case, it is the "earthquakes\_table".

WHERE latitude > 30 AND longitude < -100 AND mag > 3.0: This line specifies multiple conditions or filters for the query. It retrieves rows where the latitude is greater than 30, the longitude is less than -100, and the magnitude (mag) is greater than 3.0. This condition narrows down the results to earthquakes located in a specific geographical area with a certain magnitude threshold.

ORDER BY mag ASC: This line specifies the ordering of the result set. The "mag" column is used for sorting in ascending order (ASC), meaning the smallest magnitude values will appear first.

LIMIT 13: This line limits the number of rows returned to 13. Only the first 13 rows, according to the specified ordering, will be included in the result set.

Overall, this query retrieves the first 13 rows from the "earthquakes\_table" that meet the conditions of having a latitude greater than 30, a longitude less than -100, and a magnitude greater than 3.0. The result set includes the "place" and "mag" columns and is sorted in ascending order based on the "mag" column.

A screenshot of a data output

Description automatically generated with medium confidence