create database healthcare;

use healthcare;

-- ques 1 Display all information in the tables

select \* from healthcare1;

/\*:Observation:

- This query retrieves all data from the healthcare1 table without any filtering, which can be useful for getting a complete view of the table's contents. However, it may lead to inefficiencies by fetching more data than necessary, especially in large tables.

:Insights:

-This query is effective for viewing all the data in the table, but it can be inefficient if only specific information is needed, leading to unnecessary data processing. It's often better to specify the required columns to optimize performance and avoid potential data overload.

-- ques 2 write a query to find out the blood group of patient whose name starts with b

select name, Blood\_type from healthcare1 where name like 'B%';

/\*:Observation:

-This query is efficiently retrieves the names and blood types of patients whose names start with the letter "B." This targeted approach is effective for narrowing down results to only relevant entries, reducing unnecessary data retrieval.

:Insights:

-This query is an efficient way to filter and retrieve specific patient data, focusing only on those whose names start with "B." By limiting the selection to relevant records, it enhances query performance and helps in quickly identifying the desired information.

-- ques 3 write a query to find out how many patienets has cancer and diabetes

select name, medical\_condition from healthcare1 where medical\_condition in ('cancer' and 'diabetes');

/\*:Observation:

-This query attempts to retrieve patients with either cancer or diabetes but contains a logical error in the IN clause. The correct query should separate the conditions with a comma, not an AND, to properly filter for both conditions.

:Insights:

-This query is intended to retrieve patients with either cancer or diabetes but contains an error in its logic. To correctly identify patients with either condition, the IN clause should list the conditions separately as ('cancer', 'diabetes'), ensuring accurate and efficient filtering.

-- ques 4 write a query to find out doctors name who treat obesity

select lower(name), medical\_condition from healthcare1 where medical\_condition = 'obesity';

;Observation:

-This query retrieves the names of individuals treating obesity and converts them to lowercase. However, without filtering for doctors specifically, the query might return patients' names as well if they are listed in the same table.

:Insights:

- This query effectively retrieves names associated with treating obesity and converts them to lowercase for consistency. However, to specifically identify patients, the query should include an additional condition or filter, such as a role or job title, to ensure only patients' names are selected,

-- ques 5 write a query to find out total billing amount generated from each medical condition

select medical\_condition, sum(billing\_amount) from healthcare1 group by medical\_condition;

/\*:Observation:

-This query efficiently calculates the total billing amount for each medical condition. By grouping the results by medical condition, it provides a clear summary of the financial impact associated with each type of treatment.

:Insights:

-This query effectively aggregates the total billing amounts by medical condition, offering valuable insights into the revenue generated from different treatments. This can help in identifying the most financially significant conditions, aiding in resource allocation and financial planning.

-- ques 6 write a query to find out to how much insurance each insurance provider give

select insurance\_provider, count(insurance\_provider) from healthcare1 group by insurance\_provider;

/\*:Observation:

--This query counts the number of entries for each insurance provider but does not directly address the total insurance coverage amounts. While it provides a summary of provider frequency, the query needs modification to sum the actual insurance amounts to meet the intended objective.

:Insights:

-This query provides the number of records associated with each insurance provider but does not capture the total insurance amounts provided. To obtain the total coverage amount, the query should aggregate the sum of insurance amounts rather than just counting entries.

-- ques 7 write a query to find out the urgent admission type

select name, admission\_type from healthcare1 where admission\_type = ('urgent');

/\*:Observation:

-This query accurately retrieves names and admission types for records where the admission type is 'urgent.' This provides a focused view of urgent admissions ensuring relevant data is effectively filtered from the table.

:Insights:

-This query efficiently isolates records with an urgent admission type, helping to quickly identify and review these cases. This targeted approach ensures that only relevant data is retrieved, streamlining analysis and decision-making for urgent admissions.

-- ques 8 write a query to find out patients each medication

select count(medication), medication from healthcare1 group by medication;

/\*:Observation;

-This query provides the count of patients for each medication, summarizing how many patients are prescribed each type. This grouping and counting approach effectively highlights the distribution of medications among patients.

:Insights:

-This query delivers insights into the number of patients receiving each medication by counting occurrences and grouping the results. This helps in understanding the prevalence and distribution of various medications among patients, aiding in inventory and treatment planning.

-- ques 9 write a query to find out the patient who is below 60 years of age

select lower(name), age from healthcare1 where age < 60;

/\*:Observation:

-This query retrieves the names and ages of patients under 60 years old, converting names to lowercase for consistency. This approach effectively filters and standardizes the data, focusing on a specific age group for further analysis.

:Insights:

-This query efficiently identifies patients younger than 60 years and presents their names in lowercase, facilitating uniformity in the output. This targeted query helps in filtering data to focus on a specific age demographic, which can be useful for age-based analysis or targeted interventions.

-- ques 10 write a query to find out how much is each test result

select count(test\_results), test\_results from healthcare1 group by test\_results;

/\*:Observation:

-This query counts the number of occurrences for each test result, providing a frequency distribution. However, if you need the actual amounts or values of test results, a different aggregation function or query structure would be required.

:Insights:

-This query provides a count of how many times each test result appears, offering insights into the frequency of each result type. To analyze the monetary value or other quantitative aspects of the test results, additional columns and aggregation functions would be needed.