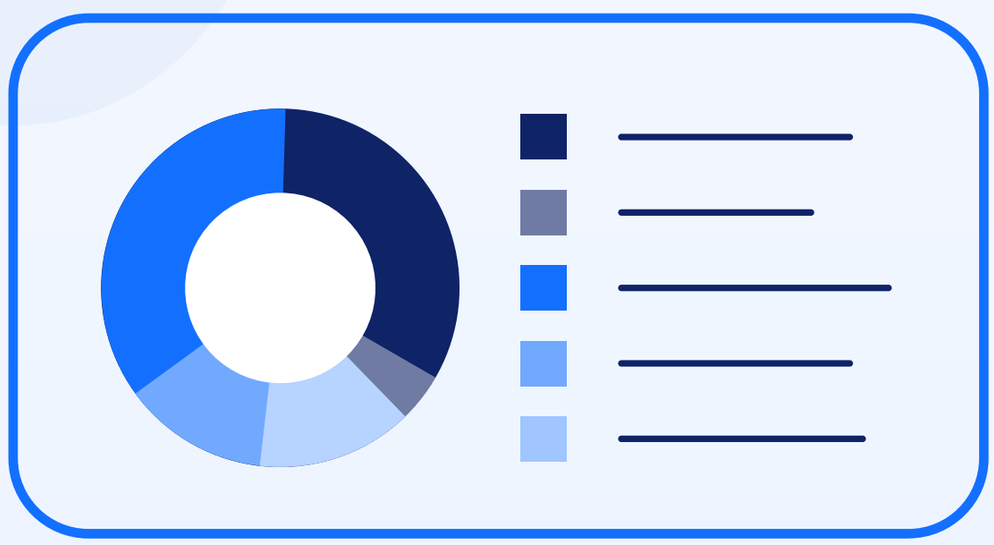
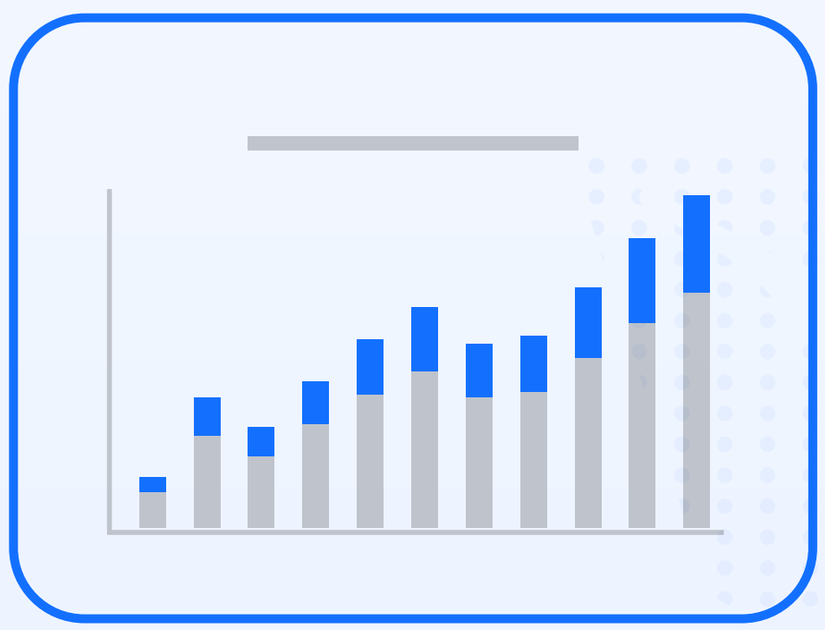
TOP 30

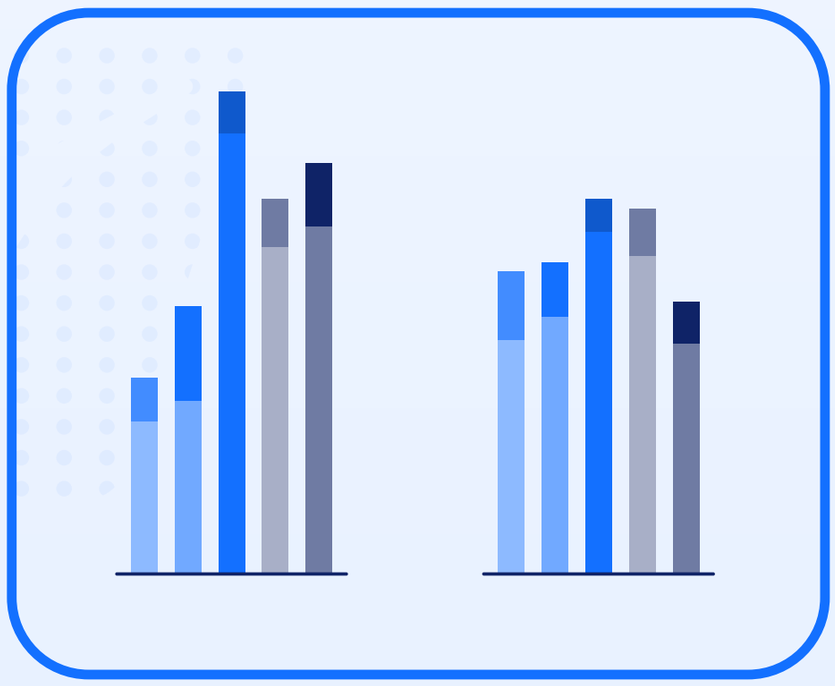
Frequently Asked

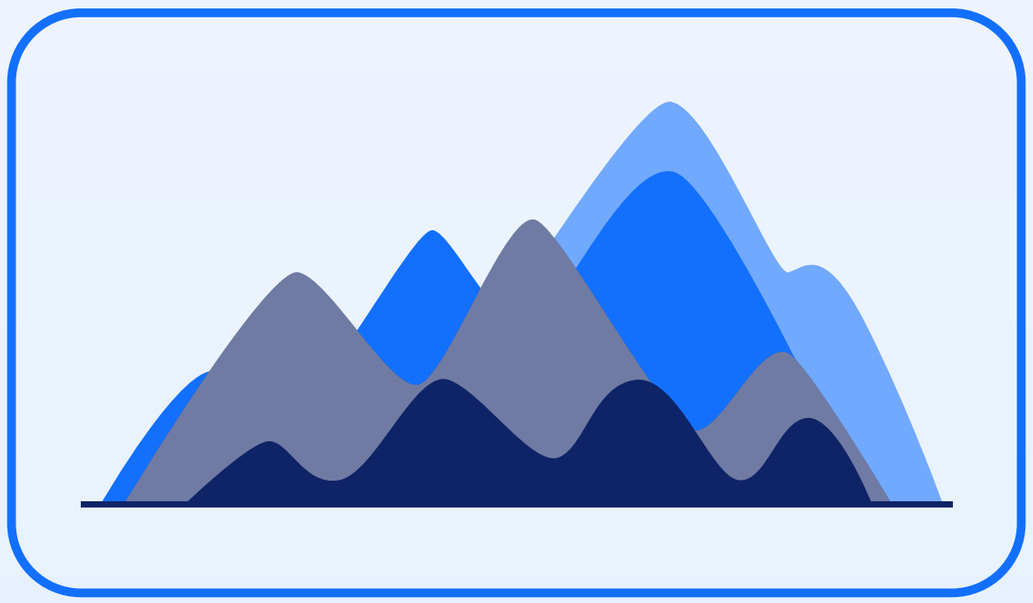
DATA ANALYST

Questions









By MAANG Companies

Disclaimer

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.

Question - 1

What makes OLTP different from OLAP?

- **OLTP (Online Transaction Processing)** deals with day to day transactions which makes real time entry and retrieval of data fast.
- **OLAP (Online Analytical Processing)** is for the analysis of huge amount for data and is more concentrating on the high integrity of the queries made and the reports developed for decision making.

In short: OLTP is good for processing data transactions; OLAP good for data analysis.

Question - 2

Explain how you would approach cleaning a dataset with 10% missing values.

To clean a dataset with 10% missing values, I would

1. **Assess the missing data :** Which columns have got missing values and how many records are there with these missing values?
2. **Decide on handling methods :**
    - If there is a significant number of missing values for numerical data columns, use imputation (mean, median model-based), or delete rows/columns alike.
    - For categorical variables it is better to use the mode imputation or create a new category for instance “Unknown”.
3. **Decide on handling methods :** Ensure while scrubbing the data it does not have loss of patterns and doesn’t have elements of biasness.

Question - 3

How do you design an ETL pipeline for real-time analytics?

To design an ETL (Extract, Transform, Load) pipeline for real-time analytics:

1. **Extract :** Some of the common real-time data source are the use of message queues such as Kafka or APIs for real-time data access.
2. **Transform :** Do operations such as filtering, aggregation, or enrichment on the fly, typically employing stream processing engines (Apache Flink, Spark Streaming ...).
3. **Load :** Load transformed data to real-time data storage such as a real-time data-warehouse like AWS Redshift, google big query etc.

Question - 4

How do you ensure data quality in a project?

To ensure data quality in a project, I focus on:

1. **Clear Data Collection Standards :** Develop guidelines in terms of the adherence of the process used in data collection.
2. **Data Validation :** This is in several cases done with the use of software, but it is important to perform this check frequently.
3. **Data Cleaning :** It usually involves eliminating, for example, duplicate or unnecessary results of database queries.
4. **Timely Updates :** Update the data and keep the same up-to-date.
5. **Regular Audits :** The reviews should be done periodically in order to ensure that its contents are accurate and complete.

Question - 5

What is the importance of p-values in hypothesis testing?

A P-value aids in adding whether or not the outcomes of a hypothesis test are statistically different. If the results are statistically significant, a small p-value ($p < 0.05$, or the researcher's chosen alpha level) reject null hypothesis in favour of the stand with the consequent supporting the research alternative hypothesis. A large p-value means the data we present in our study does not allow us to reject the null hypothesis.

Question - 6

Can you describe the difference between normalization and standardization?

Normalization brings data to a scale within a particularly predefined range, which is often between 0 and 1 while Standardization adjusts data to have a zero mean and unit standard deviation. Normalization is helpful when the features have different units of measurement but **standardization** is helpful when features have different scales which must be measured in one form.

Question - 7

Explain how you would optimize a SQL query for large datasets.

To optimize a SQL query for large datasets:

1. **Use indexes** : Index a column on fields to be searched on and fields used in a `JOIN` statement as well as in an `ORDER BY` statement.
2. **Limit result set** : To limit the number of results returned use `LIMIT` or `TOP` .
3. **Avoid** `SELECT` : The learner should only choose the desirable columns to enhance the efficiency of data processing.
4. **Use efficient joins** : Use `INNER JOIN` rather than `OUTER JOIN` and avoid unnecessary ones.
5. **Use** `WHERE` **filters early** : Filters need to be applied right from the start as to minimize the number of rows that are going to be dealt with.
6. **Optimize subqueries** : Generally it is better to use joins rather than subqueries.
7. **Analyze query execution plan** : Look at the execution plan to recognize some problems concerning time-consuming.

Question - 8

How do you handle skewed data distributions?

To handle skewed data distributions, you can use techniques like:

1. **Log Transformation :** If the data are skewed, and the variable is continuous, use log or square root transformation.
2. **Winsorization :** Maximum and minimum numbers for variables to limit effect of outliers.
3. **Resampling :** Undersampling or oversampling is the method used when there is imbalanced data.
4. **Model Selection :** This one should be done using algorithms that are less likely to be substantially affected by skewed data, such as tree based models.

Question - 9

What is a Type I error and Type II error? Give examples

To handle skewed data distributions, you can use techniques like:

- Type I error on the other hand is made when the null hypothesis is rejected when in actual sense it is true.
  Example: An X-ray, for example, improperly suggests a person free of a certain disease actually has the disease.

- In Type II error also known as false negative, you do not reject a false null hypothesis or fail to find them to be not true.
  Example: An m-test fails to 'screen out' a person who, in reality, has a disease.

Question - 10

What is the difference between LEFT JOIN and FULL OUTER JOIN in SQL?

In SQL:

1. **LEFT JOIN** : Brings back the all of the records of the left table with the matching record from the right table. In case where there is no match, NULL values are returned from the columns in the right table.
2. **FULL OUTER JOIN** : Brings all records if there is a match in any of the two tables. It contains combination of unmatched rows of both left and right tables, where unmatched column has NULL values.

Question - 11

How would you design a dashboard to track product performance?

Design a product performance dashboard with the following key features:

1. **Overview Section :** The major ideas such as the sales, revenues or customers feedback can be easily observed at a glance.
2. **Graphs/Charts :** Categorize performance by means of charts such as bar, line or pie.
3. **Filters :** Users should be able to sort the data by time, geographical area, and/ or by product type.
4. **Comparisons :** Build direct comparisons with related values (for example, product to target comparison).
5. **Alerts/Notifications :** They underline some important change or problem in the company (for example, weak product/s).
6. **Real-time Data :** It is also important to make sure the latest data is put in the dashboard frequently.

Question - 12

How do you decide between RDBMS and NoSQL for a project?

1. **RDBMS (e.g., MySQL, PostgreSQL) :** Suits well when dealing with huge volumes of structured data, where the assemblies of data are intricate and for situations where reliable transactions are preferred.
2. **NoSQL (e.g., MongoDB, Cassandra) :** Suitable to handle data that are ill defined or partially defined, excellent for growing businesses, and when the structure may evolve over some time.

Question - 13

Explain the concept of data normalization in databases.

Data normalization in databases can be defined as the action of arranging data to decrease the problems of repetition as well as to increase the consistency of data. It is necessary for breaking a site into more discreet tables to reduce a number of replicates, and establishing associations between them. This makes the database much more flexible as well as easy to manage than using other complicated structures.

Question - 14

How do you detect and handle outliers in a dataset?

1. **Detect Outliers** :

    - It is better to use such graphical representation such as box plot or scatter plot.
    - Use statistical approaches such as the IQR rule or the Z-scores to so doing.

2. **Handle Outliers** :

    - Remove : In particular, it may happen that an outlier results from data entry errors or it is not useful for the analysis
    - Transform : Reduce the impact by using some of the methods like the log transformation.
    - Cap/Impute: It is suggested replacing outliers with maximum or median value.

Question - 15

How do you approach A/B testing?

My conception of the A/B testing is based on beginning with a specific objective, for example, increasing conversion. Then, I split the audience into two groups: one gets to see the first test (control) and the other gets an opportunity to look at the second test (variation). I make sure that the test is run for long enough to collect adequate data then statistics must be used to find out the version that performed well.

Question - 16

What is the difference between batch processing and stream processing?

1. **Batch Processing** : Analyses a large amount of data that has been gathered over the period. It is a batch process which means it is not done interactively (for instance preparing daily, weekly or monthly reports).
2. **Stream Processing** : Data gets analyzed in real-time, a moment when the data is being produced, and you can work with it immediately (for instance, analyzing traffic on the website in the process of its functioning).

Question - 17

How do you optimize joins in SQL queries?

To optimize joins in SQL queries:

1. **Use Proper Indexing :** Make it possible that only indexed columns are used in the join conditions.
2. **Filter Early :** By adding the filters in the `WHERE` or `ON` clause, always try to reduce the dataset before joining it.
3. **Choose the Right Join Type :** Always opt for `INNER JOIN` since it has been stated to be faster than using an `OUTER JOIN` .
4. **Avoid Joining Unnecessary Tables :** Thus, only those tables and columns required for the query only should be included.
5. **Check Execution Plan :** Optimize query execution plan so as to remove any obstacles or modify the methods used in query optimization.

Question - 18

How do you design a data warehouse with a star schema?

To design a data warehouse with a star schema :

1. **Identify the Business Process :** Identify out the type of flow you wish to model, for instance; sale flow, inventory flow and so on.
2. **Define the Fact Table :** Make special table that will contain numeric values like amount of sales, number of pieces sold and etc.
3. **Define Dimension Tables :** Generate other tables for the descriptive attributes (time, product, customer, location etc.).
4. **Establish Relationships :** Relation the fact table to each of the dimension tables using the primary key and foreign key.
5. **Optimize for Queries :** Make sure that the schema is decomposed and kept as plain as possible for needed queries.

Question - 19

How would you calculate the 90th percentile of sales in SQL?

Calculating of the 90th percentile of sales in SQL is easier if using built-in function named *PERCENTILE_CONT*, which computes a percentile within a given set of values arranged according to the specified order.

```sql
SELECT PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY sales) AS percentile_90
FROM sales_table;
```

This will return the 90th percentile of the sales column from the *sales_table*.

Question - 20

What is the difference between a Snowflake schema and a Star schema?

- **A Star schema** is a basic data model characterized by a fact table and one or more dimension tables. Every dimension table is connected with the fact table creating a star-like structure of organization.
- **A Snowflake schema** is slightly more difficult to understand. This is somewhat like the Star schema but the dimension tables are normalized into several related tables that create a ‘snowflake’ structure.

Question - 21

What is the role of indexing in databases?

Database indexing enhances the efficiency of data searching through formation of a structure for data rows called table or tree, such that the database does not need to go through the entire table in order to find a particular row. Like in books, it is an index helping to search faster.

Question - 22

How would you calculate churn rate in SQL

To calculate churn rate in SQL:

1. Total customers at the beginning of a period : *total_customers_start*
2. *Quantify churned or the number of customers who left during the period (churned_customers).*

Use the formula :

```sql
SELECT
    (CAST(churned_customeers AS FLOAT) /
total_customers_start) * 100 AS churn_rate
FROM (
    SELECT
        COUNT (*) AS churned_customers
    FROM customers
    WHERE status = ‘churned’ AND churn_date BETWEEN
    ‘start_date’ AND ‘end_date’
) AS churned;
(
    SELECT
```

```
        COUNT (*) AS total_customers_start
    FROM customers
    WHERE join_date <= 'start_date'
) AS start;
```

Replace '*start_date*' and '*end_date*' with your period dates.

Question - 23

How do you decide between using Python or SQL for a data task?

Use Python where you need to perform rather heavy calculations, carry out analytics or machine learning, or format free form data. SQL should be used preferably when it is directly required to operate on relational databases through queries for filter, aggregate function or joining of big frames.

Question - 24

Explain the differences between supervised and unsupervised learning

1. **Supervised Learning** :

    - Incorporates labeled data which are input-output pairs.
    - The model learns with the aim at predicting outputs from the given inputs.
    - Example: House price prediction based on historical data.

2. **Unsupervised Learning** :

    - Requires no predetermined outcomes to be applied to the data it processes.
    - The model classifies data that is available by looking for patterns or grouping them.
    - Example: Customer classification in marketing.

Question - 25

How do you prioritize tasks in a data analytics project?

Prioritize tasks in a data analytics project using these steps :

1. **Define Objectives :** Learn what the project is about and what the major questions to answer are.
2. **Assess Impact :** Concentrate on what you think is most critical or useful to your work.
3. **Sequence Dependencies :** Complete basic activities before deploying, for example, analytical procedures.
4. **Allocate Resources :** Fit the tasks to the characteristics of the team members and resources in their disposal.
5. **Set Timelines :** Divide work on the project into particular stages with corresponding dates.
6. **Iterate :** Carry out investigation based on the results and update the plan according to the development in the project.

Question - 26

How do you decide which visualization to use for a given dataset?

To decide on a visualization :

1. **Understand Your Data :** Consider the type of data (categorical and numerical) as well as the type of relationship to establish which is appropriate namely comparison, distribution, trends, or composition.
2. **Define Your Goal :** Be clear about what you need to portray for example, temporal changes, relative sizes, relationships.
3. **Choose the Right Chart :**
    - Comparison : Bar chart, line chart.
    - Distribution : Histogram, box plot.
    - Trends : Line chart.
    - Composition : Pie chart, stacked bar chart.
    - Relationships : Scatter plot, bubble chart.

Question - 27

What is the difference between UNION and UNION ALL in SQL?

The key difference between `UNION` and `UNION ALL` in SQL is :

1. **`UNION` :** After executing two queries, the command combines the results and erases all the rows that are similar. This means there is additional activity that includes sorting and checking for duplicates of similar records.

2. **`UNION ALL` :** Joins two columns/trials on same parameters and retrieves all the rows, including any rows that are duplicated. It is faster, especially due to the lack of needful check that are generally performed to confirm a record was indeed successfully added.

When you are looking for unique output then you should go for `UNION` while for duplicate output and when performance is also a concern you should go for `UNION ALL`.

Question - 28

How do you ensure the scalability of a data pipeline?

To ensure scalability in a data pipeline :

1. **Distributed Processing :** Organize large datasets through the utilization of special platforms such as Apache Spark, or Kafka.
2. **Horizontal Scaling :** Invest in more machines, or nodes for handling increased workload.
3. **Modular Design :** Make pipelines in standalone and composable steps to scale up the process more efficiently.
4. **Auto-scaling :** Use service models that are connected with the cloud scenario and which are able to increase on their own.
5. **Optimized Storage :** Some general and cheap storage solutions are cloud object storage which includes S3 and GCS.
6. **Monitoring and Load Balancing :** Be consistent in analyzing frequent performance and avoid large variations in the distribution of load.

Question - 29

How do you handle correlated variables in predictive modeling?

To handle correlated variables in predictive modeling :

1. **Identify Correlation :** When using the correlation matrix select variables with high correlation, for instance Pearson correlation coefficient greater than 0.8.
2. **Remove Redundancy :** There should be one measure retained while other measures that can offer similar information should be removed.
3. **Use Regularization :** The correlation is managed by methods like Lasso or Ridge regression because it punishes less important features.
4. **Dimensionality Reduction :** Use methods such as PCA in order to replace several related variables by several orthogonal components.
5. **Domain Knowledge :** Choose the variable that best fits what you consider to be the problem with the organization.

Question - 30

Explain the difference between rank() and dense_rank() in SQL.

The key difference between *RANK()* and *DENSE_RANK()* in SQL lies in how they handle ranking when there are ties :

1. **RANK() :** May leave gaps in the ranking depending on the number of ties present for the ranking. For example, where two rows have the same rank of 1, then the next row would have a rank of 3 (1, 1, 3).
2. **DENSE_RANK() :** It does not create gaps in ranking. If there are equal two rows as the highest rank, the next rank will be the second rank (1st rank = 1, 2nd rank = 1).

Both are used to order the rows in order to give each row a rank according to the order given.

BOSSCODER ACADEMY

WHY BOSSCODER?

2200+ Alumni placed at Top Product-based companies.

More than **136% hike** for every 2 out of 3 Working Professional.

Average Package of **24LPA**.

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company .

Rahul
Google

EXPLORE MORE