Data Science & Data Analyst Interview Preparation Guide

SQL • Python • Statistics • Machine Learning • Case Studies • Behavioral

---

Prepared for Ashutosh — Code With Purpose

Tip: Don't just memorize answers — understand the 'why'. Interviewers probe follow-ups far more than they ask the first question itself.

Contents

1. SQL Interview Questions
2. Python for Data Science
3. Statistics & Probability
4. Machine Learning
5. Data Analysis & Business Case Questions
6. Behavioral & HR Questions
7. Quick Revision Cheat-Sheet

1. SQL Interview Questions

Q1. What is the difference between WHERE and HAVING?

WHERE filters individual rows before grouping/aggregation happens. HAVING filters groups after a GROUP BY / aggregate function has been applied. You can't use aggregate functions like COUNT() or SUM() in WHERE, but you can in HAVING.

Q2. Explain the different types of JOINs.

INNER JOIN returns only matching rows in both tables. LEFT JOIN returns all rows from the left table plus matches from the right (NULL if no match). RIGHT JOIN is the mirror of LEFT. FULL OUTER JOIN returns all rows from both sides, matched where possible. CROSS JOIN returns the Cartesian product.

Q3. What is the difference between RANK(), DENSE_RANK(), and ROW_NUMBER()?

ROW_NUMBER() assigns a unique sequential number to every row, even for ties. RANK() gives the same rank to ties but skips the next rank(s) (1,1,3). DENSE_RANK() gives the same rank to ties but does not skip (1,1,2).

Q4. How do you find duplicate rows in a table?

GROUP BY the columns you consider a duplicate key, then use HAVING COUNT(*) > 1. Example: SELECT email, COUNT(*) FROM users GROUP BY email HAVING COUNT(*) > 1.

Q5. What is a self-join and when would you use one?

A self-join joins a table to itself, typically used for hierarchical or comparative data — e.g., finding employees and their managers within the same employee table.

Q6. Explain window functions and give an example.

Window functions perform calculations across a set of rows related to the current row without collapsing them into a single output row (unlike GROUP BY). Example: SELECT name, salary, AVG(salary) OVER (PARTITION BY department) AS dept_avg FROM employees.

Q7. What's the difference between UNION and UNION ALL?

UNION combines result sets and removes duplicate rows (requires a sort/dedup step, so it's slower). UNION ALL combines result sets and keeps duplicates, making it faster when you know there won't be overlaps or duplicates are acceptable.

Q8. How do indexes improve query performance, and what's the trade-off?

Indexes let the database find rows without scanning the whole table (like a book's index), speeding up SELECTs and WHERE/JOIN lookups. The trade-off is slower INSERT/UPDATE/DELETE operations (the index must be updated too) and extra storage space.

Q9. What is normalization? Name the first three normal forms briefly.

Normalization organizes data to reduce redundancy. 1NF: atomic column values, no repeating groups. 2NF: 1NF + no partial dependency on a composite key. 3NF: 2NF + no transitive dependency (non-key columns depend only on the primary key).

Q10. Write a query to find the second-highest salary from an employee table.

SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees); -- Alternative: use DENSE_RANK() and filter rank = 2.

Q11. What is a CTE (Common Table Expression) and why use one over a subquery?

A CTE (WITH clause) defines a named temporary result set that can be referenced within the main query. It improves readability, allows recursion, and can be reused multiple times in the same query, unlike a subquery which is repeated inline.

Q12. How would you find the Nth highest value in a column?

Using window functions: SELECT * FROM (SELECT *, DENSE_RANK() OVER (ORDER BY col DESC) AS rnk FROM table) t WHERE rnk = N.

Q13. What is the difference between a clustered and a non-clustered index?

A clustered index determines the physical order of data in the table (only one per table). A non-clustered index is a separate structure that points back to the data rows (a table can have many).

Q14. Explain ACID properties in databases.

Atomicity: a transaction fully completes or fully fails. Consistency: the database moves from one valid state to another. Isolation: concurrent transactions don't interfere with each other. Durability: once committed, changes persist even after a crash.

Q15. How do you handle NULL values in SQL aggregations?

Most aggregate functions (SUM, AVG, COUNT(column)) ignore NULLs automatically. COUNT(*) counts all rows including NULLs. Use COALESCE() or IS NULL/IS NOT NULL to explicitly handle or filter NULLs when the default behavior isn't what you want.

2. Python for Data Science

Q1. What is the difference between a list, tuple, and set in Python?

List: ordered, mutable, allows duplicates. Tuple: ordered, immutable, allows duplicates (faster, hashable if contents are hashable). Set: unordered, mutable, no duplicates, fast membership testing (O(1) average).

Q2. Explain the difference between deep copy and shallow copy.

A shallow copy (copy.copy()) creates a new object but inserts references to the same nested objects. A deep copy (copy.deepcopy()) recursively copies every nested object, so changes to the copy never affect the original.

Q3. What are Python decorators and why are they useful?

A decorator is a function that wraps another function to extend its behavior without modifying its source code, often used for logging, timing, caching, or access control. Applied using the @decorator_name syntax above a function definition.

Q4. Difference between loc[] and iloc[] in Pandas.

loc[] selects rows/columns by label (index name or column name). iloc[] selects by integer position, regardless of the actual index labels.

Q5. How do you handle missing data in a Pandas DataFrame?

Detect with df.isnull().sum(). Handle by dropping (df.dropna()), filling with a statistic (df.fillna(df.mean())), forward/backward fill (ffill/bfill), or using model-based imputation (e.g., KNNImputer) depending on how much data is missing and why.

Q6. What is the difference between apply(), map(), and applymap() in Pandas?

map() works element-wise on a Series. apply() works on a Series or DataFrame, row-wise or column-wise (axis parameter). applymap() applies a function element-wise across an entire DataFrame.

Q7. Explain list comprehension with an example, and why it's often preferred over loops.

A list comprehension builds a list in a single readable line: [x**2 for x in range(10) if x % 2 == 0]. It's typically faster than an equivalent for-loop with .append() because of optimized internal looping in CPython, and it's more concise.

Q8. What is the Global Interpreter Lock (GIL) and how does it affect multithreading?

The GIL is a mutex in CPython that allows only one thread to execute Python bytecode at a time, even on multi-core machines. This means threading doesn't speed up CPU-bound tasks, but it's fine for I/O-bound tasks; for true parallel CPU work, use multiprocessing instead.

Q9. How would you merge two DataFrames in Pandas, and what are the join types?

pd.merge(df1, df2, on='key', how='inner') — how can be 'inner', 'left', 'right', or 'outer', mirroring SQL join semantics.

Q10. What's the difference between NumPy arrays and Python lists?

NumPy arrays are homogeneous (single data type), stored in contiguous memory, and support fast vectorized operations. Python lists are heterogeneous and slower for numerical operations since they store references to objects rather than raw values.

Q11. How do you handle outliers in a dataset using Python?

Common approaches: IQR method (flag points outside Q1 - 1.5*IQR to Q3 + 1.5*IQR), Z-score thresholding (|z| > 3), visual inspection with boxplots, or domain-specific capping/winsorizing — choice depends on whether outliers are errors or genuine extreme values.

Q12. What is broadcasting in NumPy?

Broadcasting lets NumPy perform arithmetic on arrays of different shapes by automatically expanding the smaller array's dimensions without copying data, as long as their shapes are compatible.

Q13. Explain the difference between *args and **kwargs.

*args collects extra positional arguments into a tuple. **kwargs collects extra keyword arguments into a dictionary. Both let a function accept a variable number of arguments.

Q14. How would you optimize a slow Pandas operation on a large dataset?

Vectorize instead of using .apply()/loops, use appropriate dtypes (category, downcast numerics), read only needed columns, use chunking for very large files, or switch to a faster engine like Polars/Dask/Vaex for out-of-core processing.

Q15. What is a generator in Python and why use one?

A generator is a function using 'yield' that produces values lazily, one at a time, instead of building a full list in memory. Useful for processing large or infinite data streams memory-efficiently.

3. Statistics & Probability

Q1. What is the Central Limit Theorem and why does it matter?

It states that the sampling distribution of the mean of a sufficiently large number of independent samples will approximate a normal distribution, regardless of the population's original distribution. It underpins most hypothesis testing and confidence interval methods.

Q2. Explain Type I and Type II errors.

Type I error (false positive): rejecting a true null hypothesis (alpha level). Type II error (false negative): failing to reject a false null hypothesis (beta). There's typically a trade-off between the two, controlled by significance level and sample size/power.

Q3. What is a p-value? What does $p < 0.05$ actually mean?

A p-value is the probability of observing data at least as extreme as what was seen, assuming the null hypothesis is true. $p < 0.05$ means there's less than a 5% chance of seeing this result (or more extreme) if the null hypothesis were true — it does NOT mean a 95% chance the alternative hypothesis is true.

Q4. Difference between correlation and causation.

Correlation measures how two variables move together, but doesn't imply that one causes the other — a third confounding variable or reverse causality could explain the relationship. Causation requires controlled experiments or careful causal inference methods to establish.

Q5. What is the difference between population and sample, and why do we use sample statistics?

A population is the entire group of interest; a sample is a subset drawn from it. We use samples because studying an entire population is often impractical or impossible, and sample statistics (with appropriate methods) allow us to estimate population parameters within a known margin of error.

Q6. Explain the difference between a Type of Bias: Selection bias vs Survivorship bias.

Selection bias occurs when the sample isn't representative of the population due to how it was collected. Survivorship bias is a specific form where you only analyze entities that 'survived' a process, ignoring those that failed or dropped out, leading to overly optimistic conclusions.

Q7. What is the difference between variance and standard deviation?

Variance is the average squared deviation from the mean (units are squared). Standard deviation is the square root of variance, bringing it back to the original units, which makes it more interpretable.

Q8. Explain A/B testing and how you'd determine statistical significance.

A/B testing randomly splits users into control and treatment groups to compare a metric. You'd run a hypothesis test (e.g., t-test or z-test for proportions), check the p-value against a significance threshold, and also consider effect size, sample size/power, and practical significance before concluding a real difference exists.

Q9. What is multicollinearity and why is it a problem in regression?

Multicollinearity occurs when independent variables are highly correlated with each other. It inflates the variance of coefficient estimates, making them unstable and hard to interpret, though it doesn't necessarily hurt overall prediction accuracy.

Q10. Explain the difference between a confidence interval and a prediction interval.

A confidence interval estimates the range for a population parameter (like the mean). A prediction interval estimates the range for a single future observation, and is always wider because it accounts for both estimation uncertainty and individual variability.

Q11. What is Bayes' Theorem? Give a real-world example.

Bayes' Theorem updates the probability of a hypothesis given new evidence: P(A|B) = P(B|A)*P(A) / P(B). Example: updating the probability someone has a disease after a positive test result, factoring in the test's false positive rate and the disease's base rate (prevalence).

Q12. What's the difference between parametric and non-parametric tests?

Parametric tests (t-test, ANOVA) assume the data follows a specific distribution (usually normal) and rely on parameters like mean/variance. Non-parametric tests (Mann-Whitney U, Kruskal-Wallis) make fewer assumptions about the underlying distribution and are used when those assumptions are violated.

Q13. Explain skewness and kurtosis.

Skewness measures the asymmetry of a distribution (positive skew = long right tail, negative = long left tail). Kurtosis measures the 'tailedness' or how much data is in the tails versus the center compared to a normal distribution.

4. Machine Learning

Q1. Explain the bias-variance tradeoff.

Bias is error from overly simplistic assumptions (underfitting); variance is error from being too sensitive to training data fluctuations (overfitting). Reducing one often increases the other, so the goal is finding the sweet spot that minimizes total error on unseen data.

Q2. What is the difference between bagging and boosting?

Bagging (e.g., Random Forest) trains multiple models independently in parallel on bootstrapped samples and averages/votes results, primarily reducing variance. Boosting (e.g., XGBoost, AdaBoost) trains models sequentially, each correcting the errors of the previous one, primarily reducing bias.

Q3. How do you handle class imbalance in a classification problem?

Techniques include resampling (SMOTE/oversampling the minority class, undersampling the majority), class-weighting in the loss function, using appropriate metrics (F1, precision-recall AUC instead of accuracy), and ensemble methods designed for imbalance like Balanced Random Forest.

Q4. Explain precision, recall, and F1-score. When would you prioritize one over another?

Precision = TP/(TP+FP) — how many predicted positives are actually correct. Recall = TP/(TP+FN) — how many actual positives were caught. F1 is their harmonic mean. Prioritize precision when false positives are costly (e.g., spam filters); prioritize recall when false negatives are costly (e.g., cancer/fraud detection).

Q5. What is regularization and why do we need L1 vs L2?

Regularization adds a penalty term to the loss function to discourage overly complex models and reduce overfitting. L1 (Lasso) adds the absolute value of coefficients, which can shrink some to exactly zero (feature selection). L2 (Ridge) adds the squared value, shrinking coefficients smoothly but rarely to zero.

Q6. How does a Random Forest work?

It builds many decision trees, each trained on a random bootstrap sample of the data and a random subset of features at each split (reducing correlation between trees), then aggregates their predictions via majority vote (classification) or averaging (regression).

Q7. What is the difference between supervised, unsupervised, and reinforcement learning?

Supervised learning trains on labeled data to predict outcomes (classification/regression). Unsupervised learning finds patterns/structure in unlabeled data (clustering, dimensionality reduction). Reinforcement learning trains an agent to take actions in an environment to maximize cumulative reward through trial and error.

Q8. Explain how gradient descent works.

Gradient descent iteratively updates model parameters in the direction that reduces the loss function, moving opposite to the gradient (slope) of the loss with respect to the parameters, scaled by a learning rate, until it converges to a minimum.

Q9. What is cross-validation and why use k-fold instead of a single train/test split?

Cross-validation splits data into k folds, training on k-1 folds and validating on the remaining fold, rotating through all folds. This gives a more robust estimate of model performance than a single split, since every data point is used for both training and validation, reducing variance in the performance estimate.

Q10. How would you detect and prevent overfitting?

Detect via a large gap between training and validation performance. Prevent through regularization, more training data, simpler models, dropout (in neural nets), early stopping, cross-validation, and feature selection to reduce noise.

Q11. Explain the difference between a decision tree and logistic regression.

Logistic regression models the log-odds of the outcome as a linear combination of features, producing a smooth probability estimate and being highly interpretable via coefficients. Decision trees split data recursively based on feature thresholds, capturing non-linear relationships and interactions naturally, but are more prone to overfitting without pruning.

Q12. What is the curse of dimensionality?

As the number of features grows, data becomes increasingly sparse in the feature space, distances between points become less meaningful, and models require exponentially more data to generalize well — motivating dimensionality reduction techniques like PCA.

Q13. How does PCA (Principal Component Analysis) work?

PCA finds new orthogonal axes (principal components) that capture the maximum variance in the data, ordered by how much variance they explain. Projecting data onto the top few components reduces dimensionality while preserving as much information as possible.

Q14. What is the difference between a generative and a discriminative model?

Generative models learn the joint probability distribution P(X, Y) and can generate new data (e.g., Naive Bayes, GANs). Discriminative models learn the conditional probability P(Y|X) directly to distinguish between classes (e.g., logistic regression, SVM), typically performing better on classification tasks alone.

Q15. Explain how you would evaluate a regression model.

Common metrics: MAE (average absolute error, robust to outliers), MSE/RMSE (penalizes larger errors more), and R-squared (proportion of variance explained). Choice depends on whether large errors should be penalized more heavily and whether interpretability in the original units matters.

Q16. What is feature engineering and can you give examples?

Feature engineering is creating new input variables from raw data to improve model performance, e.g., extracting day-of-week from a timestamp, creating ratio features, one-hot encoding categoricals, binning continuous variables, or aggregating transaction history into summary statistics per user.

5. Data Analysis & Business Case Questions

Q1. Walk me through how you would analyze a drop in a company's weekly active users.

Structure the approach: (1) confirm the data/metric definition hasn't changed, (2) segment the drop by platform, geography, user cohort, and acquisition channel to isolate where it's concentrated, (3) check for external factors (seasonality, outages, competitor launches, marketing spend changes), (4) check for internal factors (recent releases, bugs, pricing changes), (5) quantify magnitude and recommend next steps or A/B tests to validate a hypothesis.

Q2. How would you measure the success of a new feature launch?

Define the primary success metric tied to the feature's goal (e.g., engagement, retention, revenue) before launch, run an A/B test against a control group, monitor guardrail metrics to catch unintended negative effects, and give it enough time/sample size to reach statistical significance before concluding impact.

Q3. A stakeholder asks you to build a dashboard. What questions would you ask first?

Who is the audience and what decisions will they make with it? What's the required refresh frequency? What are the 2-3 key metrics that matter most? What historical context or benchmarks should be shown? What's the current source of truth for the data, and are there known data quality issues?

Q4. How would you explain a complex analysis or model to a non-technical stakeholder?

Lead with the business implication and recommendation first, use analogies and visuals instead of jargon and formulas, quantify impact in terms they care about (revenue, cost, time saved), and be transparent about key assumptions and limitations without over-explaining the technical mechanics.

Q5. How do you decide which metrics matter for a given business problem?

Start from the business objective, work backward to a primary metric (North Star) that reflects it, add supporting/diagnostic metrics that explain movement in the primary metric, and add guardrail metrics to catch negative side effects — avoiding vanity metrics that look good but don't tie to real outcomes.

Q6. Describe a time you found an insight in data that changed a decision.

This is a behavioral/STAR-format question — answer with a real example structured as: the Situation and business context, the specific Task or question you were investigating, the Action you took (your analysis approach and tools), and the measurable Result or decision it influenced.

Q7. How would you approach analyzing customer churn for a subscription business?

Define churn precisely (voluntary vs involuntary, time window), build a cohort-based retention curve to understand the pattern, identify leading indicators via exploratory analysis or a classification model (usage decline, support tickets, pricing tier), segment by customer type, and translate findings into specific, testable retention interventions.

6. Behavioral & HR Questions

Q1. Tell me about yourself.

Keep it to ~90 seconds: current focus/skills, a highlight project or achievement with quantifiable impact, and why you're excited about this specific role/company. Practice this out loud — it sets the tone for the whole interview.

Q2. Why do you want to work here / in this role?

Connect something specific about the company's product, mission, or data challenges to your own skills and goals — avoid generic answers like 'growth opportunities' without specifics.

Q3. Tell me about a time you disagreed with a team member or manager.

Use the STAR format: Situation, Task, Action, Result. Emphasize how you communicated respectfully, used data to support your view, and reached a resolution — the outcome and what you learned matter more than winning the disagreement.

Q4. Describe a challenging project and how you overcame obstacles.

Pick a project with a genuine obstacle (data quality issues, unclear requirements, technical constraints), be specific about the actions you personally took, and quantify the outcome where possible.

Q5. How do you prioritize when you have multiple deadlines?

Explain your framework: assessing business impact and urgency, communicating trade-offs early with stakeholders, and being transparent about what will slip rather than silently missing deadlines.

Q6. What's a time you made a mistake in an analysis? What did you do?

Own the mistake honestly, explain how you caught it (or how it was caught), the corrective action taken, and — importantly — what process change you made afterward to prevent it from recurring.

Q7. Where do you see yourself in 3-5 years?

Tie your answer to realistic growth within the data field (e.g., deepening ML expertise, moving toward more ownership of business-facing analytics) rather than an unrelated or overly ambitious leap — show you've thought about it honestly.

7. Quick Revision Cheat-Sheet

Topic	Key Formula / Concept
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F1-Score	2 * (Precision * Recall) / (Precision + Recall)
Variance	Average of squared deviations from the mean
Standard Deviation	Square root of variance
Bayes' Theorem	P(A|B) = P(B|A) * P(A) / P(B)
Z-score	(x - mean) / standard deviation
R-squared	1 - (SS_residual / SS_total)
IQR (outlier rule)	Outlier if < Q1 - 1.5*IQR or > Q3 + 1.5*IQR
Normalization (1NF/2NF/3NF)	Atomic values -> no partial dependency -> no transitive dependency

Final Tips Before Your Interview

- Practice explaining every project on your resume out loud in under 2 minutes — interviewers almost always start here.
- For SQL/coding rounds, practice writing queries on paper or a plain editor (no autocomplete) to simulate real conditions.
- For case questions, always structure your answer out loud first ('I'd break this into 3 parts...') before diving in.
- Prepare 2-3 thoughtful questions to ask the interviewer about the team's data stack, current challenges, or success metrics.
- Get a good night's sleep — clear thinking under pressure matters more than last-minute cramming.