Practice technical questions

Thomas Dickson

November 17, 2023

This document contains questions to test knowledge on subject areas to do with working as a data engineer, MLOps engineer and software engineer.

This knowledge is just as important as being able to pass logic and coding questions. Got to Section 1 for a list of resources to use to test logic, brainteasers and coding questions.

Contents

1 Practical questions

These are some references for practical problems to work through. Aim to spent 80% of interview prep time working on practical problems described in the references below.

- [5] is a good reference for brainteasers, logic problems and statistical questions.

- Use [3] as a reference for software engineering problems as well as general interview prep.

Recommended way of working:

1. Review question answers from the day before and double check them.
2. Iterate over questions from a given problem domain.
3. Log the problem domain, questions passed and failed.
4. Iterate over problem domains but do always review the failed questions from the previous session. Practice recall is the name of the game.

2 Software engineering

2.1 Management

1. What does **agile** mean in the context of software development?

   Solution:

   - A process for discovering requirements and solutions improvement through the collaborative effort of self-organizing and cross-functional teams with their end users.
   - Popularised in the Agile manifesto
   - Underpins Kanban and Scrum.

2. (4 points) What are lean management practices?

   Solution:

   - Limit Work in Progress (WIP)
   - Visual management
   - Feedback from production
   - Lightweight change approvals

   Reference [1, p. 76]

3. (4 points) What are the components of Lean Product Management?

Solution:

- Work in small batches
- Make flow of work visible
- Gather and implement customer Feedback
- Team experimentation

Reference [1, p. 85]

4. (6 points) What are common factors that lead to burnout?

Solution:

- Work overload: job demands exceed human limits.
- Lack of control: inability to influence decisions that affect your job.
- Insufficient rewards: insufficient financial, institutional, or social rewards.
- Breakdown of community: unsupportive workplace environment.
- Absence of fairness: lack of fairness in decision-making processes.
- Value conflicts: mismatch in organizational values and the individuals values.

Reference [1, p. 96]

5. (5 points) How can you reduce or fight burnout?

Solution:

- Organizational culture. Managers are responsible for fostering a supportive and respectful work enviroment, and they can do so by creating a blame free environment, striving to learn from failures, and communicating a shared sense of purpose. Human error is never the root cause of failure in systems.
- Deployment pain. Managers and leaders should ask their teams how painful their deployments are and fix the things that hurt the most.
- Effectiveness of leaders. Responsibilities of a team leader include limiting work in process and eliminating roadblocks for the team so they can get their work done.
- Organizational investment in DevOps.

- Organizational performance. Lean management and continuous delivery practices help improve software delivering performance, which in turn improves organizational perforamcne.

6. What factors can you use to select software?

Solution: [6, p. 119]

2.2 Principles

1. (2 points) How can you classify tests?

Solution: You can classify tests on two main dimensions:

- Size. Size refers to the resources consumed by a test and what it is allowed to do.
- Scope. Scope refers to how much code a test is intended to validate.

2. (5 points) What is a unit test? What are some properties of unit testing?

Solution: A unit test can refer to a test of narrow scope, such as of a single class or method. Unit tests are usually small in size, but that's not always the case.

Some properties of unit tests are that:

- They tend to be small, which helps them be fast and deterministic.
- They can be easy to write at the same time as the code they're testing.
- They promote high levels of test coverage as a consequence of the previous two factors.
- They tend to make it easy to understand what's wrong when they fail.
- They can serve as documentation and examples.

Reference [4, Chapter 17]

3 ML and ML Ops

1. What are the key distance metrics you can use when comparing vectors?

Solution:

- Minkowski distance: $(\sum_{i=1}^{n} |X_i - Y_i|^p)^{(1/p)}$
  - When $p = 1$ this becomes the Manhattan distance - i.e. the distance you'd travel if you had to walk the blocks to get there
  - When $p = 2$ it becomes the Euclidean distance which would be the straight line distance/distance as the crow flies.
- Cosine distance: $S_C(A, B) := cos(\Theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$
  - This is a measure of similarity between two non-zero vectors defined in an inner product space.
  - It doesn't depend on the magnitude of the vectors.
  - In informational retrieval and text mining, each document could be represented by the vector of the numbers of occurences of each work therefore cosine similarity would give a useful measure of how similar two documents are likely to be independent of the length of the documents. Presumably that would depend on how the documents/sentences/words would have been vectorized.

4 Data engineering

4.1 Principles and concepts

1. What are the differences between a column orientated database (DBMS) and a row orientated database?

Solution: Describe properties of both and give examples.

2. What are the differences between a relational and document databases?

Solution: Relational database and document database definitions:

- A relational database is based on a relational document model where data is organised into relations, also known as tables, and each relation is an unordered collection of tuples. The data is normalised in order to minimise the number of locations different data might need to be updated. An example is Postgres.
- Document database stores nested records. An example is Elasticsearch.

Here are some ways they can be compared:

- Fault tolerance
- Concurrency.
- Which data model will lead to simpler code? Document databases may have issues with joins or referring to deeply nested objects.
- How can schema flexibility in the document model enable functionality? What if one of the fields needed to be updated? In a document data model new documents would have to be written whereas a relational data model would require a migration to add a column in the right table.
- Data locality for queries. If the application often needed to access the entire document then a document data model would make sense.

[2, p. 38]

3. What properties does an ACID transaction have?

Solution:

- Atomicity. Transactions are often composed of multiple statements. Atomicity guarantees that each statement is treated as a single unit, which either succeeds or fails completedly. If any of the statements in a transaction fail then the entire transaction fails. This prevents partial updates.
- Consistency. Consistentcy ensures that a transaction can only bring the database from one consistent state to another, preserving database invariants.
- Isolation. Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.
- Durability. Durability guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure. This usually means that only completed transactions (or effects) are recorded in non-volatile memory.

4. What are the differences between an OLTP and an OLAP system?

Solution:

- An Online Transaction Processing (OLTP) system is a database that reads and writes records at a high rate.
  - These systems are typically referred to as transactional databases, but that does not imply that the system in question supports atomic transactions.

  - Generally speaking these systems support low latency and high concurrency.
  - OLTP databases work well as application backends when thousands or millions of users can be interacting with the application simultaneously.
- An Online Analytical Processing (OLAP) system is built to run large analytics queries and is typically inefficient at handling lookups of individual records.
  - The Online part of OLAP implies that the system is continually scanning for incoming queries, making the system suitable for interactive analytics.
  - High latency queries with low latency lookups.
  - Snowflake/BigQuery -¿ columnar databases?

5. How should you evaluate which storage abstraction should be used?

Solution:

- Purpose and use case: What purpose and use case the data will be used for?
- Update pattern: Is the abstraction optimized for bulk updates, streaming inserts, or upserts?
- Cost: What are the direct and indirect financial costs? Time to value? The opportunity costs?
- Seperate storage and compute: The trend is toward seperating storage and compute but most systems hybridize seperation and colocation. This means that the lines between OLAP databases and data lakes are blurring.

Reference [6, p. 219]

6. What is a data warehouse?

Solution:

- Data warehouses are standard OLAP data architecture.
- Data warehouses can refer to:
  1. Technology platforms such as Google BigQuery and Teradata
  2. An architecture for data centralization
  3. An organisational pattern within a company
- In practice, data warehouses are used to organize data into a data lake.
- Cloud data warehouses can be coupled with object storage to provide a complete data-lake solution.

7. What is a data lake?

Solution:

- The *data lake* was concieved of as a massive store where data was retained in raw, unprocessed form.
- The last 5 years has seen two major developments:
    1. A migration towards sepeartion of compute and storage.
    2. Discovery that functionality such as schema management dismissed in the move to data lakes was, in face, extremely useful.

Reference [6, p. 220].

8. What is a data lakehouse?

Solution: A data lakehouse is an architecture that combines aspects of the data warehouse and the data lake. A lakehouse stores data in object storage but also adds arrangement features designed to streamline data management and enhance engineering experience, e.g. schema management and features for managing updates/deletes. Reference [6, p. 220].

4.2 Tooling

1. (1 point) What is docker?

Solution:

2. (1 point) What is terraform?

Solution:

5 Python

Questions on Python programming all the way to niche Python behaviour.

5.1 General knowledge

1. Can you hash a set?

Solution:

- No you can't because a set, as well as a list and a dict are all mutable and therefore unhashable.
- If you want to make sure they're hashable then you need to make them immutable REF.

2. What is monkey patching?

Solution:

- Monkey patching is dynamically updating a method at runtime.
- It means that it's possible to modify the behaviour of some source code without editing that source code directly.

3. How does Python work under the hood?

Solution: It uses Cython. Here's a guide.

4. What are Python decorators?

Solution: A specific change made in Python syntax to alter functions easily.

5. What is the difference between a list and a tuple?

Solution: A tuple is not mutable but can be hashed. Lists are mutable.

6. What is the Global Interpreter Lock?

Solution: The GIL is a construct that ensures that only one thread is executed at any given time. A thread acquires the GIL and then performs work before passing it to the next thread.

7. What is the difference between range and xrange?

Solution:

- Both return sequences of numbers.
- range returns a list.
- xrange returns a generator

8. What is a generator and why is it useful?

Solution:

- A generator is a function that returns an iterator that produces a sequence of values when iterated over.
- Generators are useful when you want to generate a sequence but you don't want to store it all in memory.

9. (4 points) Describe the different types of inheritance.

Solution:

- Single inheritance, where a derived or child class inherits properties from a single parent class.
- Multiple inheritance, when a class can be derived from more than one base class.
- Multilevel inheritance, where features of the base class and the derived class are further inherited into the new derviced class.
- Hierarchical inheritance, when more than one derived class are created from a single base.

10. (3 points) What is a deep copy and what is a shallow copy and when is the difference relevant?

Solution:

- A shallow copy constructs a new compound object and then inserts references into it to the objects found in the original.
- A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original.
- The difference between shallow and deep copying is only relevant for compound objects - objects that contain other objects.

11. (2 points) What problems can exist with deep copy operations that don't exist with shallow copy operations?

Solution:

- Recursive objects (compound objects that directly or indirectly contain a reference to themselves) may cause a recursive loop.
- Because deep copy copies everything it may copy too much, such as data which is intended to be shared between copies.

Python copy documentation.

12. (2 points) What are local and global namespaces?

Solution:

- Local namespaces are defined inside a block of code and are only accessible inside the block.
- Global namespaces includes names from various imported modules that are being used in a project. It lasts until the script ends.

13. What is the difference between a module and a package?

Solution:

- A module is a single file containing python code.
- A package is a collection of modules that are organised in a directory hierarchy.

14. What are abstract base classes and why are they useful?

Solution: Python ABC Docs.

References for these questions include my head, actual interview questions and [7].

5.2 Pandas

1. What types of merge exist and how are they used?

Solution:

6 SQL

1. What types of join exist and how are they used?

Solution:

7 Maths

7.1 Useful algorithms

1. Write a function to convert an arabic number to roman numerals.

Solution: https://medium.com/@tomas.langkaas/eight-algorithms-for-roman-nume

7.2 Big O notation

https://github.com/Devinterview-io/big-o-notation-interview-questions

7.3 Useful numbers

https://www.techinterviewhandbook.org/algorithms/math/
Double check cracking the coding interview.

References

1. Nicole Forsgren, J. H. & Kim, G. *Accelerate: Building and Scaling High Performing Technology Organiztions* 1st ed. (IT Revolution, 2018).
2. Kleppmann, M. *Designing Data-Intensive Applications* 7th ed. (O'Reilly, 2019).
3. McDowell, G. L. *Cracking the Coding Interview* 6th ed. (CareerCup, 2019).
4. Titus Winters, T. M. & Wright, H. *Software Engineering at Google* 1st ed. (O'Reilly, 2020).
5. Crack, T. F. *Heard on The Street: Quantitative Questions from Wall Street Job Interviews* 22nd ed. (O'Reilly, 2021).
6. Reiss, J. & Housley, M. *Fundamentals of Data Engineering* 1st ed. (O'Reilly, 2022).
7. Hackr.IO. *Python Interview Questions* https://hackr.io/blog/python-interview-questions.