1. Difference between IS-A relationship & HAS-A relationship

Whenever one class inherits another class, it is called an IS-A relationship. Has-A relationship: Whenever an instance of one class is used in another class, it is called HAS-A relationship

1. what's super Harvard architecture

-Have Big cache memory

-have I/O controller

1. unstructured DB examples

MongDB

CouchDb

Redis

Key Value

1. What is Hadoop , hive , Apache spark , Apache base , map reduce

* **The Apache Hadoop** software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.
* **Hive** allows users to read, write, and manage petabytes of data using SQL. Hive is built on top of Apache Hadoop, which is an open-source framework used to efficiently store and process large datasets.
* **Apache Spark** is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.
* **Base** is a fully featured desktop database management system, designed to meet the needs of a broad array of users, from tracking a personal CD collections, to producing a corporate monthly departmental sales reports.
* **MapReduce** is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster. As the processing component, MapReduce is the heart of Apache Hadoop. The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform.

1. what's better pip or conda

Conda

conda is a system package manager. pip is a Python package manager.

With conda you can install much more than just Python libraries. You can install entire software stacks such as Python + Django + Celery + PostgreSQL + nginx. You can install R, R libraries, Node.js, Java programs, C and C++ programs and libraries, Perl programs, the list is pretty long and limitless. conda has an env system that allows you to have all of these installed across multiple different environments. Also, conda is able to do all these software and package installations in an isolated, userspace manner. This is critical because it means that you can install complex software stacks on a system (such as your employer's heavily regulated production server) without needing root privileges. In a lot of ways, conda serves as a lightweight userspace alternative to Docker for isolating software stacks. Also important to note that, as of the most recent versions of conda, conda's env is always active. By default when you install conda for the first time, the default env is used (I think its called "base" or something like that), and will remain in use until you create & switch to another env. You do not need a third-party environment management system with conda.

On the other hand, pip can only install Python packages, and it quite often screws up the installations on multi-user systems, breaking global system dependencies and/or the user's dependency stacks. This is why people who rely only on pip MUST use virtualenv, but even then pip sometimes misbehaves and installs to the wrong places. In general, pip is dangerous and a mess to use. Easy to screw up your user Python library stack or even the entire server's installation. Tread carefully any time you use the globally available system-installed pip.

1. what is s.o.l.i.d principles

S - Single-responsiblity Principle

O - Open-closed Principle

L - Liskov Substitution Principle

I - Interface Segregation Principle

D - Dependency Inversion Principle

* Single-responsibility Principle (SRP) states:

A class should have one and only one reason to change, meaning that a class should have only one job.

* Open-closed Principle (OCP) states:

Objects or entities should be open for extension but closed for modification

* Liskov Substitution Principle states:

Let q(x) be a property provable about objects of x of type T. Then q(y) should be provable for objects y of type S where S is a subtype of T.

* Interface segregation principle states:

A client should never be forced to implement an interface that it doesn’t use, or clients shouldn’t be forced to depend on methods they do not use.

* Dependency inversion principle states:

Entities must depend on abstractions, not on concretions. It states that the high-level module must not depend on the low-level module, but they should depend on abstractions.

<https://www.digitalocean.com/community/conceptual_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>

1. What is the topology of a distributed system?

Topology means the shape of a local-area network (LAN) or other communications system. Topologies are either physical or logical. There are four principal topologies used in LANs: bus topology: All devices are connected to a central cable, called the bus or backbone.

1. What is dataOps and what is the cycle for it ?

DataOps is a collaborative data management practice focused on improving the communication, integration and automation of data flows between data managers and data consumers across an organization.

DataOps is an agile approach that aims to reduce cycle time of data analytics. DataOps monitors and automates data life cycle. It improves the integration and automation of data flow between users in the organization. Quality: Large volumes of data can cause data inconsistency problems.

