1.) Explain the differences between Cassandra and Typical databases.

Sl.no	CASSANDRA	RDBMS
1.	1.Cassandra is a high performance and highly scalable distributed NoSQL database management system.	1.RDBMS is a Database management system or software which is designed for relational databases.
2.	2. Cassandra is a NoSQL database.	2.RDBMS uses SQL for querying and maintaining the database.
3.	3. It deals with unstructured data.	3. It deals with structured data.
4.	4. It has a flexible schema.	4. It has fixed schema.
5.	5. Cassandra has peer-to-peer architecture with no single point of failure.	5.RDBMS has master-slave core architecture means a single point of failure.
6.	6.Cassandra handles high volume incoming data velocity.	6. RDBMS handles moderate incoming data velocity.
7.	7.In Cassandra there are various data source means data come from one/few location.	7. In RDBMS there is limited data source means data come from many locations.
8.	8. It supports simple transactions.	8.It supports complex and nested Transactions.
9.	9.In Cassandra the outermost container is Keyspace.	9. In RDBMS the outermost container is database.
10.	10. Cassandra follows decentralized deployments.	10.RDBMS follows centralized deployments.
11.	11.In Cassandra data written in many locations.	11.In RDBMS mainly data are written in one location

2.) What exactly is CQLSH?

- cqlsh is a command-line interface for interacting with Cassandra using CQL (the Cassandra Query Language). It is shipped with every Cassandra package, and can be found in the bin/ directory alongside the cassandra executable.

3.) Explain the Cassandra cluster idea.

- A Cassandra cluster does not have a single point of failure as a result of the peer-to-peer distributed architecture. Nodes in a cluster communicate with each other for various purposes. There are various components used in this process: Seeds: Each node configures a list of seeds which is simply a list of other nodes.

4.) Give an example to demonstrate the class notion.

- Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects.
- EXAMPLE:-

  ```
  • class MyClass:
      x = 5
  ```

5.) Use an example to explain the object.

- Example of class and object For example, Person(Human) can be treated as a class which has properties such as name, age,gender etc. Every individual can be treated as an object of the class human or Person. Each individual will have different values of the properties of class Person.