

1.Explain the differences between the cassendra and typical databases.?

Ans: Difference between Cassandra and RDBMSCassandra: Cassandra is a high-performance and highly scalable distributed NoSQL database management system. Cassandra deals with unstructured data and handles a high volume of incoming data velocity. In Cassandra data is written in many locations also data come from many locations this row represents a unit of replication and the column represents a unit of storage. Difference between Cassandra and RDBMS

Cassandra: Cassandra is a high-performance and highly scalable distributed NoSQL database management system. Cassandra deals with unstructured data and handles a high volume of incoming data velocity. In Cassandra data is written in many locations also data come from many locations this row represents a unit of replication and the column represents a unit of storage.

RDBMS: Relational Database Management System (RDBMS) is a Database management system or software that is designed for relational databases and uses Structured Query Language (SQL) for querying and maintaining the database. It deals with structured data and handles moderate incoming data velocity. In RDBMS mainly data is written in one location also data come from one/few locations and a row represents a single record column that represents an attribute.

Difference between Cassandra and RDBMS:

S.No. CASSANDRA RDBMS

1. Cassandra is a high performance and highly scalable distributed NoSQL database management system. RDBMS is a Database management system or software which is designed for relational databases.
 2. Cassandra is a NoSQL database. RDBMS uses SQL for querying and maintaining the database.
 3. It deals with unstructured data. It deals with structured data.
 4. It has a flexible schema. It has fixed schema.
 5. Cassandra has peer-to-peer architecture with no single point of failure. RDBMS has master-slave core architecture means a single point of failure.
 6. Cassandra handles high volume incoming data velocity. RDBMS handles moderate incoming data velocity.
 7. In RDBMS there is limited data source means data come from many locations. In Cassandra there are various data source means data come from one/few location.
 8. It supports simple transactions. It supports complex and nested transactio
- RDBMS: Relational Database Management System (RDBMS) is a Database management system or software that is designed for relational databases and uses Structured Query Language (SQL) for querying and maintaining the database. It deals with structured data and handles moderate incoming data velocity. In RDBMS mainly

data is written in one location also data come from one/few locations and a row represents a single record column that represents an attribute. Difference between Cassandra and RDBMS.

2.What exactly is CQLSH.?

Ans:. By default, Cassandra provides a prompt Cassandra query language shell (cqlsh) that allows users to communicate with it. Using this shell, you can execute Cassandra Query Language (CQL).

Using cqlsh, you can

Define a schema,

Insert data, and

Execute a query.

Starting cqlsh

Start cqlsh using the command cqlsh as shown below. It gives the Cassandra cqlsh prompt as output.

Cqlsh>

Cqlsh – As discussed above, this command is used to start the cqlsh prompt. In addition, it supports a few more options as well. The following table explains all the options of cqlsh and their usage.

Cqlsh –help Shows help topics about the options of cqlsh commands.

Cqlsh –version Provides the version of the cqlsh you are using.

Cqlsh –color Directs the shell to use colored output.

Cqlsh –debug Shows additional debugging information.

Cqlsh –execute

Cql_statement

Directs the shell to accept and execute a CQL command.

Cqlsh –file= “file name” If you use this option, Cassandra executes the command in the given file and exits.

Cqlsh –no-color Directs Cassandra not to use colored output.

Cqlsh -u “user name” Using this option, you can authenticate a user. The default user name is: cassandra.

Cqlsh-p “pass word” Using this option, you can authenticate a user with a password. The default password is: cassandra.

Cqlsh Commands

Cqlsh has a few commands that allow users to interact with it. The commands are listed below.

Documented Shell Commands

Given below are the Cqlsh documented shell commands. These are the commands used to perform tasks such as displaying help topics, exit from cqlsh, describe, etc.

HELP – Displays help topics for all cqlsh commands.

CAPTURE – Captures the output of a command and adds it to a file.

CONSISTENCY – Shows the current consistency level, or sets a new consistency level.

COPY – Copies data to and from Cassandra.

DESCRIBE – Describes the current cluster of Cassandra and its objects.

EXPAND – Expands the output of a query vertically.

EXIT – Using this command, you can terminate cqlsh.

PAGING – Enables or disables query paging.

SHOW – Displays the details of current cqlsh session such as Cassandra version, host, or data type assumptions.

SOURCE – Executes a file that contains CQL statements.

TRACING – Enables or disables request tracing.

3.Explain the cassendra cluster idea.?

Ans:. . Cassandra Cluster

In this Cassandra tutorial, we will go through one of the main parts of the Cassandra database i.e. Cassandra Cluster. Moreover, we will see the meaning of Cluster and different layers in Cluster. This article will guide through the parts of the cluster and the builders associated with it.

So, let's start Cassandra Cluster.2. What is Cluster?

The cluster is a collection of nodes that represents a single system. A cluster in Cassandra is one of the shells in the whole Cassandra database. Many Cassandra Clusters combine together to form the database in Cassandra.

A Cluster is basically the outermost shell or storage unit in a database. The Cassandra Cluster contains many different layers of storage units. Each layer contains the other.

Let's Explore Cassandra Applications

The following are different layers in a Cassandra Cluster:a. Node Cluster

Node is the second layer in a cluster. This layer basically comprises of systems or computers or storage units. Each cluster may contain many nodes or systems. These systems or nodes are connected together.

They collectively share data through the replication in Cassandra and independently as well. The replication factor in the nodes allows the user to have a redundancy for the data stored.

b. Keyspace

The keyspace is the next layer of the storage. In a node, there are many keyspaces. These keyspaces are basically the outermost storage unit in a system. They contain the main data. The data is distributed according to their properties or areas.

The next layer is the column families. The keyspace is further divided into column families. These column families have different areas or headings under which the data is distributed. In a keyspace, these column families are categorized into different headings or titles.

These titles further contain different layers of storage units. These column families can also be characterized by tables. The column families differ from the tables through their APIs.

d. Rows

The next layer in the database is of the Rows as according to column families. The Rows are basically the classification under which the column family is divided. These classifications, in turn, create specific distribution criteria for the entries. Hence, in this Cassandra Cluster tutorial, we went through Cassandra Cluster. Moreover, we saw different layers in Cluster that are a node, keyspace, column families, rows, and column. Also, we discussed the Cluster 5. Use an example to explain the objectBuilder.

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

Ans:.

Object-oriented interface (mapping database tables to Python classes/attributes)

Automatic conversion between internal Notion formats and appropriate Python objects

Local cache of data in a unified data store (Note: disk cache now disabled by default; to enable, add `enable_caching=True` when initializing `NotionClient`)

Real-time reactive two-way data binding (changing Python object -> live updating of Notion UI, and vice-versa) (Note: Notion->Python automatic updating is currently broken and hence disabled by default; call `my_block.refresh()` to update, in the meantime, while monitoring is being fixed)

Callback system for responding to changes in Notion (e.g. for triggering actions, updating another API, etc)

We map tables in the Notion database into Python classes (subclassing `Record`), with each instance of a class representing a particular record. Some fields from the records (like title in the example above) have been mapped to model properties, allowing for easy, instantaneous read/write of the record. Other fields can be read with the `get` method, and written with the `set` method, but then you'll need to make sure to match the internal structures exactly.

The tables we currently support are block (via `Block` class and its subclasses, corresponding to different type of blocks), space (via `Space` class), collection (via `Collection` class), `collection_view` (via `CollectionView` and subclasses), and `notion_user` (via `User` class).

5. Use an example to explain the object.?

Ans:. A python is an object-oriented programming language. Almost everything in Python is considered as an object. An object has its own properties(attributes) and behavior(methods).

A class is a blueprint of the objects or can be termed as object constructor for creating objects.

One class can have many objects and value of properties for different objects can be different.

Example of properties and behavior of an object

Let's take the example of car as an object. Its properties will include its color, company name, year of manufacture, price, mileage etc. The behavior of the car will include the functions it can perform, this will include increase speed, decrease speed, apply brakes etc. Object basically related everything with real life objects. Everything we find around us in real life has some properties and some functions.

Example of class and object

Different objects belonging to same class can have different properties. 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. Everyone will have different names, age and gender.

What is instantiation?

An object is also called an instance of a class. Thus, the process of creating object of a class is known as instantiation.

Defining class in Python

As the function in Python is defined using the keyword 'def'. The keyword 'class' is used to define a class in Python. Since the class is a blueprint of the object, all the common attributes and methods will be declared and defined in the class. Different objects which are created from the class can access those properties and functions. Different objects can hold their own values for properties defined inside the class.