1. **Explain the features of Cassandra database. What are the advantages of Cassandra DB?**
2. **Peer-to-peer Network:**

A node in Cassandra is structurally identical to any other node.

* It has a peer-to-peer distributed system across its nodes, and data is distributed among all the nodes in a cluster.
* In Cassandra, each node is independent and at the same time interconnected to other nodes.
* Every node in a cluster can accept read and write requests, regardless of where the data is actually located in the cluster.
* In the case of failure of one node, Read/Write requests can be served from other nodes in the network.

1. **Gossip Protocol and Failure detection:**

The communication between nodes is often like peer-to-peer communication, where every node talks to the other.

* In Cassandra, when one node talks to another, the node which is expected to respond, not only provides information about its status, but also provides information about the nodes that it had communicated with before.
* When two nodes communicate with one another; for instance, Node A to Node B,

then Node A sends a message ‘gossipdigestsynmessage’, which is very similar to TCP protocol to Node B.

* Here, Node B, once receives the message, sends an acknowledgement message ‘ack’,

and then Node A responds with an acknowledgement message to Node B’s ‘ack’ message. This is known as the 3-way handshake.

1. **Partitioner**

* A partitioner takes a call on how data is distributed on the various nodes in a cluster.
* It also determines the node on which the first copy of data is placed.

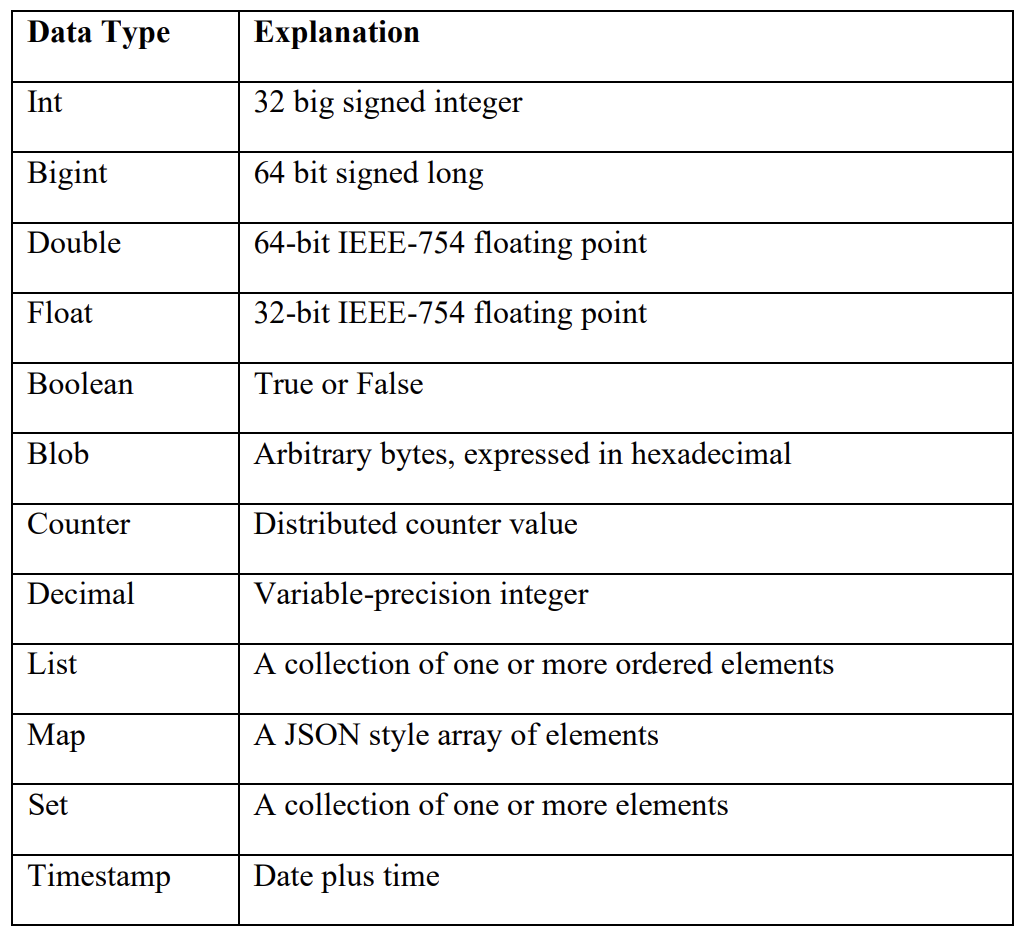
1. **Replication factor**

Replication factor determines the number of copies of data that will be stored across nodes in a cluster.

If one wishes to store only one copy of data of each row on one node then the replication factor is set to 1. The number of copies of each row of data determines the replication factor. However, the replication factor should be more than one but less than the number of nodes in a cluster.

1. **Explain CQL data types**

CQL provides a rich set of built-in data types, including collection types. Along with these data types, users can also create their own custom data types. The following table provides a list of built-in data types available in CQL.

****

1. **Explain types of collections in Cassandra database.**

**Set Collection:** A column of type set consists of unordered unique values. However, when the column is queried, it returns the value in sorted order. For example, for text values, it sorts in alphabetical order.

**Eg:**

CREATE TABLE data2 (name text PRIMARY KEY, phone set);

INSERT INTO data2(name, phone)VALUES ('rahman', {9848022338,9848022339});

**List Collection:** When the order of elements matter, one should go for a list collection. For example, when you store the preferences of places to visit by a user, you would like to respect his preferences and retrieve the values in the order in which he has entered rather than in sorted order. A list also allows one to store the same value multiple times.

**Eg:**

CREATE TABLE data(name text PRIMARY KEY, email list);

INSERT INTO data(name, email) VALUES ('ramu', ['abc@gmail.com','cba@yahoo.com'])

**Map Collection:** As the name implies, a map is used to map one thing to another. A map is a pair of typed values. It is used to store timestamp related information. Each element of the map is stored as a Cassandra column. Each element can be individually queried, modified, and deleted.

**Eg:**

CREATE TABLE data (name text PRIMARY KEY, address map);

INSERT INTO data (name, address) VALUES ('robin', {'home': 'Hyderabad', 'office': 'Delhi'});

1. **What are CRUD operations in Cassandra databases? Explain with suitable examples**

Cassandra supports basic database operations, which can be categorized into - Create, Read, Update, and Delete (CRUD) operations, as well as some specialized operations due to its distributed nature.

**CREATE (INSERT):**

To create data in Cassandra, use the INSERT statement. An insert statement does not return an output. One is not required to place values in all the columns; however, it is mandatory to specify all the columns that make up the primary key.

INSERT INTO student\_info (RollNo,StudName,DateofJoining,LastExamPercent) VALUES (l,'Michael Storm','2012-03-29', 69.6)

**READ:**

The SELECT statement is used for reading data from Cassandra tables. You can perform various types of queries, such as selecting specific rows, filtering based on conditions, and retrieving aggregated data.

To view the data from the table "student\_info"

SELECT \* FROM student\_info;

**UPDATE:**

To modify existing data, Cassandra supports the UPDATE statement. You can update one or more columns in a row. Cassandra allows conditional updates, where you can specify conditions that must be met for the update to occur.

Command: UPDATE student\_info SET StudName = 'Dileep' WHERE RollNo = 2

**DELETE:**

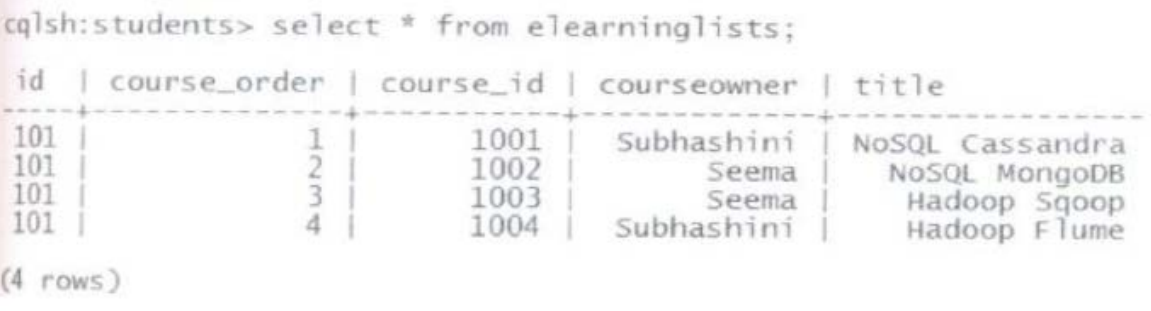
Delete statement removes one or more columns from one or more rows of a Cassandra table or removes entire rows if no columns are specified.

Command: DELETE emp\_sal FROM empWHERE emp\_id=3;

1. **Explain import and export commands with suitable examples in Cassandra.**

Objective: Export the contents of the table/column family "elearninglists" present in the "students" database to a CSV file (d:\elearninglists.csv).

**Step 1: Check the records of the table "elearninglists" present in the "students" database.** SELECT \* FROM elearninglists;

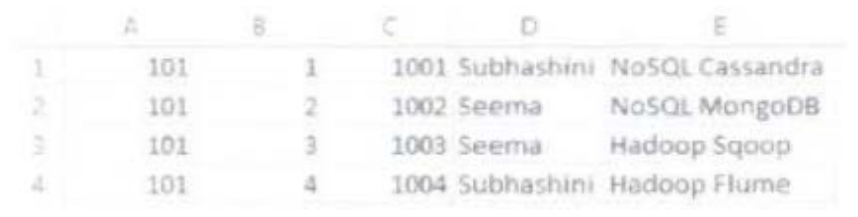


**Step 2: Execute the below command at the cqlsh prompt:**

COPY e1earninglists (id, course\_order, course\_id, courseowner, title) TO 'd:\elearninglists.csv';

**Step 3: Check the existence of the "elearninglists.csv" file in "D:\".**

Given below is the content of the "d:\elearninglists.csv" file.

****

**Or**

CREATE TABLE airplanes (name text PRIMARY KEY, manufacturer ascii, year int, mach float);

INSERT INTO airplanes (name, manufacturer, year, mach) VALUES ('P38- Lightning', 'Lockheed', 1937, 0.7);

COPY airplanes (name, manufacturer, year, mach) TO 'temp.csv';

COPY airplanes (name, manufacturer, year, mach) FROM 'temp.csv';

1. **What is TTL in Cassandra? Explain with suitable examples**

**TIME TO LIVE (TTL)**

Data in a column, other than a counter column, can have an optional expiration period called TTL (Time to live).

Functionality by which data can be automatically expired.

▪ During data insertion, you have to specify ‘ttl’ value in seconds. ‘ttl’ value is the time to live value for the data.

▪ After that particular amount of time, data will be automatically removed.

▪ For example, specify ttl value 100 seconds during insertion. Data will be automatically deleted after 100 seconds.

The client request may specify a TTL value for the data. The TTL is specified in seconds.

CREATE TABLE userlogin (userid int primary key, password text);

INSERT INTO userlogin (userid, password) VALUES (l,'infy') USING TTL 30;

SELECT TTL (password) from userlogin where userid=1;