Explain in brief

● Hbase is a schema less database, what does it mean?

**Solution:**

-HBase is a wide column store and has column families which are roughly equivalent to tables.  The column names can be completely variable and the number of columns can vary by row – so you could have a table with billions of rows and could have rows with 5 or 5 million columns.   Without HBase you can’t do table joins – and **so is definitely more of a “schema-less” nature.**

-HCatalog is a recent addition to the Hadoop stack to serve as a metadata repository about all the data stored in Pig & Hive tables and HBase column families.  This could roughly equate with a relational database schema as expressed in the system catalog.

-Even in the case of HBase, Data Modeling is still a valuable undertaking in order to design, rationalize, and communicate about data in Hadoop, even though database DDL would not be generated. With HBase, the data needs to be 100% denormalized, so a logical star schema could be used as it is easier for business people and data managers to comprehend, knowing that the star schema would, at the physical level, be converted into a single column family.   Even if the names of columns are going to be wildly different, a data model would still be beneficial. For example, key columns can be identified, columns which would require indexing can be identified, and the granularity and relationships of the data can be understood.

-In summary, “schema-less” doesn’t mean the data doesn’t have structure and data models can help us to design, rationalize, and communicate about data.

● What is the minimum number of column family every Hbase table should have?

**Solution:**

-Minimum number of column family every HBASE table should is 1 because HBASE stores the name of the column under the column family only. Also HBASE has schema less structure and hence for creating a table it is created with only a column family and while inserting the values the column name is mentioned.

-The general flow is that a Client contacts the Zookeeper first to find a particular row key. It does so by retrieving the server name from Zookeeper. With this information it can now query that server to get the server that holds the metatable. Both these details are cached and only looked up once. Lastly, it can query the metaserver and retrieve the server that has the row the client is looking for.

-Column family make the above procedure very simple in finding the particular record. Also for each created data in hbase creates the timestamp, value length and other details so it is necessary to create at least the column family.

-Hbase creates a spate directory for each and every column so one should not create many column families else the namenode will get overwhelmed so many copies of the directories due to replication factor. Hence column families must be carefully engineered.

● What is the benefit of using connection pool in Hbase?

**Solution:**

-A cluster connection encapsulating lower level individual connections to actual servers and a connection to zookeeper. Connections are instantiated through the [ConnectionFactory](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/ConnectionFactory.html" \o "class in org.apache.hadoop.hbase.client) class. The lifecycle of the connection is managed by the caller, who has to [close()](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Connection.html#close--) the connection to release the resources.

-The connection object contains logic to find the master, locate regions out on the cluster, keeps a cache of locations and then knows how to re-calibrate after they move. The individual connections to servers, meta cache, zookeeper connection, etc. are all shared by the [Table](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Table.html) and [Admin](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Admin.html) instances obtained from this connection.

-Connection creation is a heavy-weight operation. Connection implementations are thread-safe, so that the client can create a connection once, and share it with different threads. [Table](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Table.html) and [Admin](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Admin.html)instances, on the other hand, are light-weight and are not thread-safe. Typically, a single connection per client application is instantiated and every thread will obtain its own Table instance. Caching or pooling of [Table](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Table.html) and [Admin](https://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Admin.html) is not recommended.