**1.Hbase is a schema less database, what does it mean?**

-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.

**2.What is the minimum number of column family every Hbase table should have?**

* HBase currently does not do well with anything above two or three column families so keep the number of column families in your schema low (i.e to 1).
* Currently, flushing and compactions are done on a per Region basis so if one column family is carrying the bulk of the data bringing on flushes, the adjacent families will also be flushed though the amount of data they carry is small.
* When many column families the flushing and compaction interaction can make for a bunch of needless i/o loading.

**3.What is the benefit of using connection pool in Hbase?**

* Database connections are expensive to create because of overhead creating a new connection and initialization.
* when connection session initialization requires time for performance processing for user authentication.
* transactional contexts of session will require subsequent database usage.
* Ongoing management connections which are established will act as limiting factor for creating and destroying connections each time.
* When connections were already created
* They will never get disconnected and the previous connections will be given to each client.
* Time will be needed for making new connection will be highly consumed.

By means of connection pooling time can be minimized.