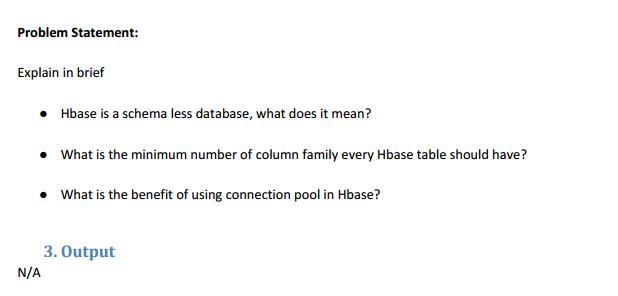
**ASSIGNMENT 31.3**



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

* In case of traditional RDBMS database.
* Table schema will be predefined and one cannot modify those easily.
* schema should defined mainly while making the table .
* Datatype of the column should be defined mainly.
* For the case of HBASE the schema can mentioned while loading of data.
* Data which is stored in HBASE will be mainly by making use of bytes .
* So it will have a independent datatype.
* Whenever a new data enters the database then HBASE can handle it smoothly.
* When HBase do not have any concept for fixed columns schema.
* It will act as a wide column store will comprise of column families equivalent to tables.
* Column names can be variable and number of columns can vary by means of rows
* So there is probability of having tables with billions of rows
* It is possible to have rows with 5 or 5 million columns.
* Without using HBase one cannot do table joins
* So it is a “schema-less” nature.
* HBase schema will have less database.
* “schema-less” doesn’t mean data will not have any structure
* Data model will help us for designing, rationing, and communicating about data.

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

* HBase currently will not have anything above two or three column families
* keeping the column number families in your schema will be as low as 1.
* Currently flushing and compactions will be done on a per Region basis .
* When one column family carry bulk data will bring on flushes.
* Adjacent families can also be flushed though the data amount is small.
* When a column having many families have the flushing and compaction interaction will make a bunch of input/output loading.

1. **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.