**Tasks before midterm report**

1. Partition

* vertical partition: store data in different tables according to ER Model

A diagram of a beer review

Description automatically generated

* + To be strict, there should be an entity “beer brewer” with attribute “beer brewer id” and relationship “beer brewer produces beer”; entity “consumer” with attribute “profile name” and relationship “consumer makes review”; and relationship “review on beer”. However, in this situation, there will be 8 tables which is too many. So based on “many-to-1 merge on many”, the ER model is reduced to as above.
  + create a unique id for entity “review” as primary key. In MySQL, we can use UUID() function to generate a 36-character unique identifier (which may be too long).
* horizontal partition
  + define a hash function to split data. For example, based on review id and beer id, we can define:
  + 这里可能要先看看数据分布如何，然后特殊设计一个hash算法
    - if starts with odd numbers: store in database 1;
    - if starts with even numbers: store in database 2;
    - if starts with character <= ‘m’: store in database 3;
    - if starts with character <= ‘m’: store in database 4;
    - ……

1. Storage

* Question: can we assume each database is a datanode, and each table is a block? If possible, we can define the write()(create not append) process as follows:
* datanode是hadoop中执行分布式任务的一个任务节点/线程的概念，这里应该不适用这个概念，然后至于把一个表当成是block来处理应该不用，因为block的大小是不变的嘛，但是数据库中表的大小随着用户的使用一定会发生变化，而且block是物理存储的才会有的概念，这个不用我们来操心如何设计
  + create(): create a storage path to the data to be created.
  + addTable(): create a new table in database A and another replicas in database B,C.
  + writeTable(): write data into tables above. Here we can use SQLAlchemy to write a large number of data at one time through Python.
* define replica: default number is 3. Higher replica means higher fault-tolerance ability.
  + If we introduce replica, we must know how to define transaction to meet consistency requirement
  + And we will also want to utilize different locks to avoid dirty read and Phantom read.
  + 并且还要考虑到可能多个用户一起使用的时候的事务隔离等（这个第一版实现的时候先不管，但这个是很重要的）

1. Frontend (not sure about this part)

* demand analysis: basically, our web pages include two parts: one for users to query/update/insert/delete (Question: should users be able to update/insert/delete data?); one to basic analysis result about beers in the form of table and charts (Question: Is it possible to use html to draw pictures with HTML5 Canvas or something?).
* website structure: design website architecture and navigation system.
* User interface design.