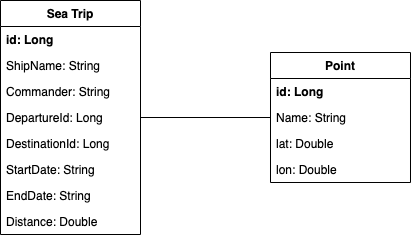
* Схема БД

SQL:



NOSQL:



* Список сущностей

SQL:

Sea Trip – сущность морского путешествия, включает в себя следующие атрибуты:

Id – уникальный идентификатор

ShipName – Название судна

Commander – Имя командира

DepartureId – уникальный идентификатор начальной точки путешествия

DestanationId – уникальный идентификатор конечной точки путешествия

StartDate – дата начала путешествия в формате yyyyMMdd

EndDate – дата окончания путешествия в формате yyyyMMdd

Distance – расстояние пройденное судном за время путешествия

Point – точка на земном шаре (мб. город, порт, остров и тп.):

Id – уникальный идентификатор

Name – название точки

Lat – широта

Lon – долгота

NOSQL(MongoDB):

Каждое путешествие представляет собой отдельный документ. Идентификатором документа является идентификатор путешествия. В документе есть поля “shipName”, “commander”, “startDate”, “endDate”, “distance”, “departure”, “destination”. Поля “departure”, “destination” являются вложенными документами, у которых есть поля “name”, “lat”, “lon”.

* Оценка объема информации

Пусть совокупность полей (“shipName”, “commander”, “startDate”, “endDate”, “distance”, “departure”, destination”) занимает N памяти, M – число путешествий, вложенные поля “name”, “lat”, “lon” – занимают K памяти.

Тогда хранение данных:

В MongoDB будет занимать M\*(N+2\*K) = M\*N + 2\*M\*K

В реляционной БД будет занимать M\*N + 2\*M\*K

Тогда при условии того, что N = (50 байт название судна + 50 байт имя командира + 8 байт дистанция + 16 байта время начала и окончания), K = (32 байта название + 16 байт координаты), пусть число путешествий M = 1200.

Тогда хранение данных будет занимать:

В MongoDB – 1200 \* 124 + 2 \* 1200 \* 48 = 244 800 байт

В реляционной БД - 244 800 байт

Таким образом количество **чистой памяти**, занимаемая реляционной БД, не отличается от MongoDB.

Количество **памяти**, занимаемое MongDB с учетом типов <Timestamp> (4 байта) и <ObjectId> (12 байт) = 1200 \* 116 + 2 \* 1200 \* 60 = 283 200 байт.

Количество **памяти**, занимаемое реляционной БД с учетом полей id (4 байта) = 1200 \* 136 + 2 \* 1200 \* 52 = 288 000 байт.

Таким образом MongoDB выигрывает реляционные БД **при учете всей информации хранимой в модели**.

Формула роста памяти для MongoDB при увеличении количества моделей будет иметь вид: N \* (116 + 2 \* 60) = 236N, для SQL модели: 240N. Следовательно, с увеличением количества записей реляционная БД будет занимать в 60/59 раз больше памяти.

* Запрос на поиск путешествия по id по всей коллекции путешествий (без учета индексов)

В MongoDb – O(M)

В SQL БД – O(2 \* (M\* 2M)) = O(4M2), т.к. необходимо сделать 2 раза операцию join на таблицу Point; размер таблицы Point - 2M, таблицы SeaTrips - M

* Запросы

Создание записи: “db.seaTrips.insertOne(

{

\_id:1,

shipName: "St. Elena",

commander: "David Porter Jr.",

startDate: new Date(41242142),

endDate: new Date(41242142),

distance: 9356.54,

departure: {

"\_id": 21,

"name": "New York",

"lat": 134.2144,

"lon": 95.313

},

destination: {

"\_id": 22,

"name": " Los Angeles",

"lat": 184.2144,

"lon": 75.313

},

})”

Поиск записи: “db.seaTrips.find({\_id:1})”

Удаление записи: “db.seaTrips.remove({\_id:1})”

* Запрос на поиск