

# NoSQL Модель данных

## 1. Графическое представление

Графическое представление модели данных MongoDB показана на рис. 1

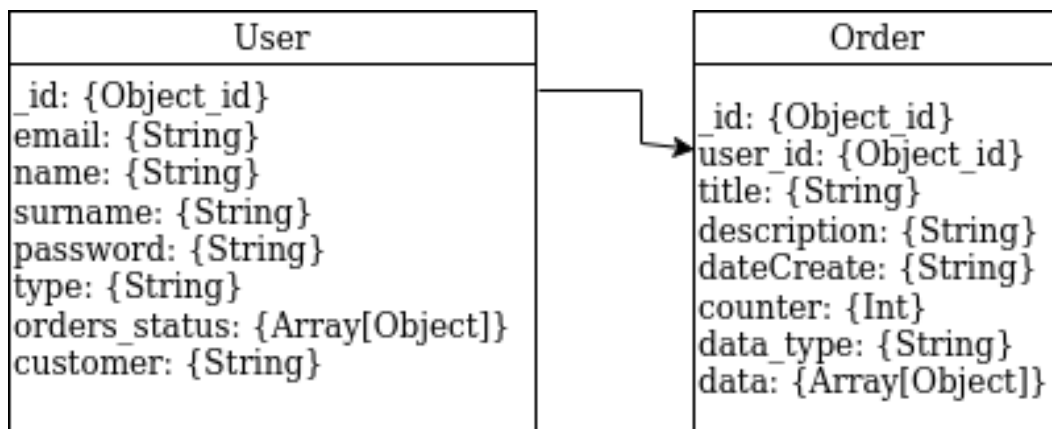


Рис. 1

## 2. Сущности модели данных

Всего было выделено 2 сущности: User, Order.

User — сущность пользователя, содержащая следующие атрибуты:

- «\_id», Object\_id — уникальный идентификатор пользователя. 12B
- «email», String — email пользователя. 50\*2B
- «name», String — имя пользователя. 50\*2B
- «surname», String — фамилия пользователя. 50\*2B
- «password», String — пароль пользователя. 50\*2B
- «type», String — тип пользователя(заказчик/исполнитель). 50\*2B
- «orders\_status», Array[Object] - информация о текущих и выполненных заказах исполнителя.

Object {

title: string, - имя заказа 50\*2B

status: string — состояние выполнения. 50\*2B

}

- «customer», String - имя заказчика. 50\*2B

Order — сущность заказа, содержит следующие атрибуты:

- «\_id», Object\_id — уникальный идентификатор заказа. 12B
- «user\_id», Object\_id — уникальный идентификатор пользователя. 12B
- «title», String — имя заказа. 50\*2B
- «description», String — описание заказа. 50\*2B
- «dateCreate», String — дата создания заказа. 50\*2B
- «counter», int — количество пользователей выполнивших заказ. 4B
- «data\_type», String — тип данных для заказа. 50\*2B
- «data», Array[Object] — данные и результаты выполнения заказа.

```
Object {
  valueObj: string, - ссылка на фото либо текст. 50*2B
  { [key: string(50*2B)]: int(4B) } - словарь для хранения результатов.
}
```

### 3. Оценка объема информации

Пусть А - количество пользователей, С — количество заказов, Е- количество ответов, D - количество объектов в наборе данных.

Тогда хранение чистых данных будет занимать:

$$A*(500 + C*200) + C*(404 + D*(100 + E*104))$$

Фактический объем:

$$A*(512 + C*200) + C*(428 + D*(100 + E*104))$$

$$\text{Избыточность модели: } \frac{A*(512 + C*200) + C*(428 + D*(100 + E*104))}{A*(500 + C*200) + C*(404 + D*(100 + E*104))}$$

Пусть количество пользователей А - 100 , количество заказов С — 1000, среднее количество ответов Е— 3, среднее количество данных в заказе D — 10.

Фактический объем:

$$100*(200512) + 100*(428 + 10*(100 + 312)) = 20506000(\text{B}) \approx 19.5 (\text{MB})$$

### 4. Запросы

Создание нового пользователя:

```
db.users.insertOne(
  '_id': id,
  'email': email,
  'name': name,
  'surname': surname,
  'password': password,
  'type': type,
  'order_status': [],
  'customer': customer
)
```

Подсчет общего числа пользователей:

```
db.users.count()
```

Подсчет общего числа заказов:

```
db.orders.count()
```

# SQL Модель данных

## 1. Графическое представление

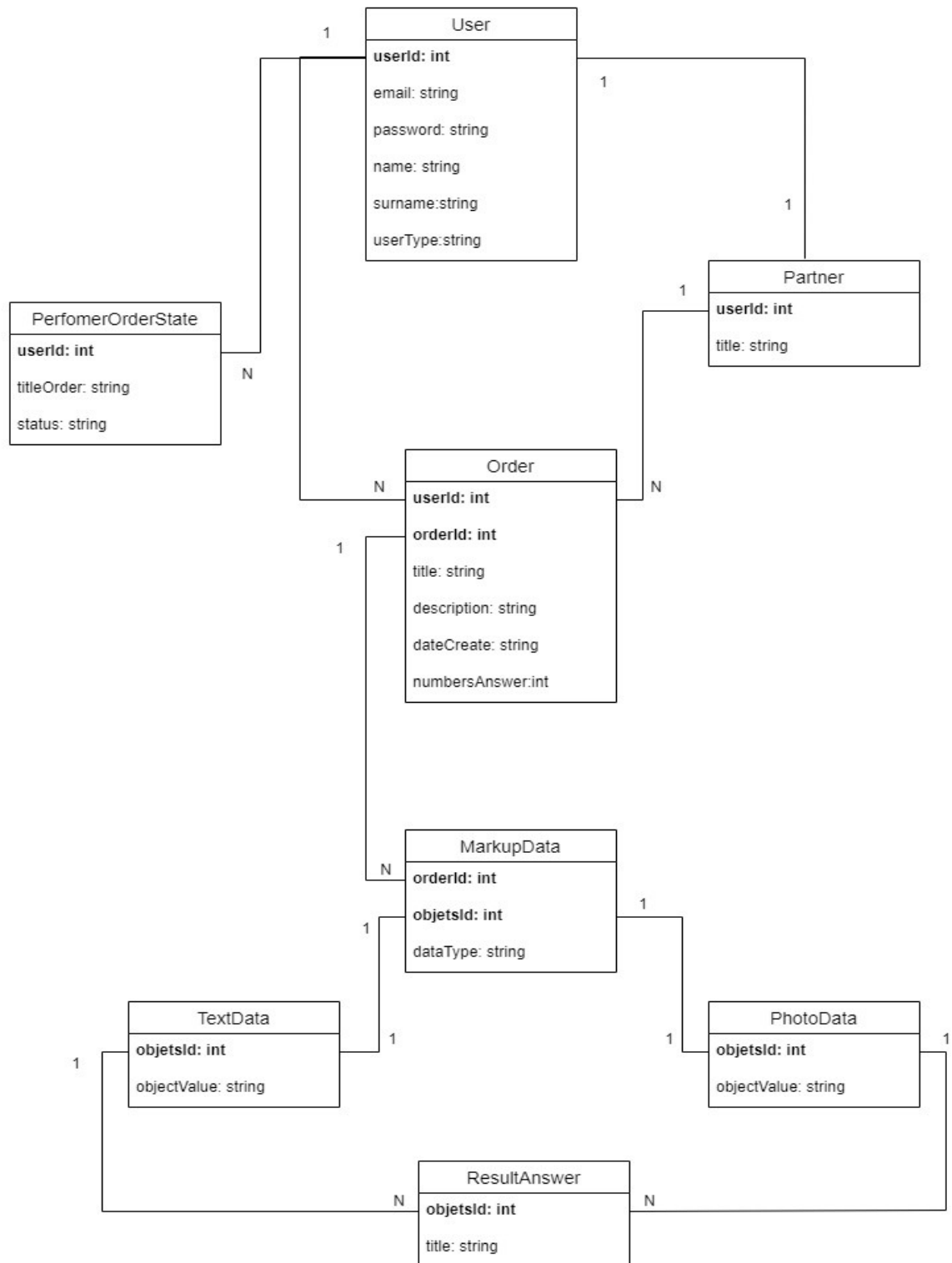


Рис. 2

## 2. Сущности модели данных

В качестве реляционной СУБД использована MySQL, в которой создано 8 таблиц:

- User-хранит информацию о пользователе. Содержит поля:
  - o Userid – уникальный идентификатор пользователя. Тип- int  $V = 4B$
  - o Email – @mail пользователя. Тип – string.  $V = 50 * 2B = 100B$
  - o Password – пароль пользователя. Тип – string.  $V = 50 * 2B = 100B$
  - o Name– имя пользователя. Тип – string.  $V = 50 * 2B = 100B$
  - o Surname– фамилия пользователя. Тип – string.  $V = 50 * 2B = 100B$

Итого: 404B

- PerfomerOrderState – хранит статус заказа Содержит поля:
  - o Userid– уникальный идентификатор статуса заказа. Тип- int  $V = 4B$
  - o titleOrder– название заказа. Тип – string.  $V = 50 * 2B = 100B$
  - o status– статус заказа. Тип – string.  $V = 50 * 2B = 100B$

Итого: 204B

- Order – хранит информацию о заказе Содержит поля:
  - o Userid – уникальный идентификатор пользователя. Тип- int  $V = 4B$
  - o Orderid– уникальный идентификатор заказа. Тип- int  $V = 4B$
  - o Title–название заказа. Тип – string.  $V = 50 * 2B = 100B$
  - o Description–описание заказа. Тип – string.  $V = 50 * 2B = 100B$
  - o dataCreate–дата создания заказа. Тип – string.  $V = 50 * 2B = 100B$
  - o numberAnswer–количество ответов. Тип- int  $V = 4B$

Итого:312B

- Partner – хранит информацию о заказчике Содержит поля:
  - o Userid– уникальный идентификатор заказчика. Тип- int  $V = 4B$
  - o Title – имя заказчика. Тип – string.  $V = 50 * 2B = 100B$

Итого:104B

- MarkupData – хранит информацию о данных для разметки Содержит поля:

o Orderid– уникальный идентификатор заказа. Тип- int  $V = 4B$

o Objectsid– уникальный идентификатор объекта для разметки. Тип- int  $V = 4B$

o dataType– тип данных для разметки. Тип – string.  $V = 50 * 2B = 100B$

Итого:108B

· TextData – хранит текстовые данные для разметки Содержит поля:

o Objectsid– уникальный идентификатор объекта для разметки. Тип- int  $V = 4B$

o objectValue– объект текста. Тип – string.  $V = 50 * 2B = 100B$

Итого:104B

· PhotoData – хранит фото данные для разметки Содержит поля:

o Objectsid– уникальный идентификатор объекта для разметки. Тип- int  $V = 4B$

o objectValue– объект изображения. Тип- string  $V = 100B$

Итого:10B

· ResultAnswer – хранит результаты разметки Содержит поля:

o Objectsid– уникальный идентификатор объекта для разметки. Тип- int  $V = 4B$

o title– значение разметки. Тип – string.  $V = 50 * 2B = 100B$

Итого:104B

### **3. Оценка объема информации**

Есть А пользователей, В из которых заказчики. У каждого заказчика имеется по С заказов. D из которых заказы текстовые. Так же имеется E ответов пользователей

Чистый объем:

·  $A * 404B$  количество пользователей

·  $C * 204$  количество статусов заказа

·  $C * 312$  количество информации о заказах

·  $B * 104$  количество заказчиков

·  $C * 108$  количество информации о данных для разметки

- $D*104$  количество текстовых данных
- $(C-D)*10B$  количество данных изображений
- $C*104B$  количество ответов размеченных данных
- $E*104B$  количество ответов пользователей

Чистый объем БД =  $A*404B + C*204 + C*312 + B*104 + C*108 + D*104 + (C-D)*10B + C*104B + E*104B$

Фактический объем:

- $A*404B$  количество пользователей
- $C*204$  количество статусов заказа
- $C*312$  количество информации о заказах
- $B*104$  количество заказчиков
- $C*108$  количество информации о данных для разметки
- $D*104$  количество текстовых данных
- $(C-D)*10B$  количество данных изображений
- $C*104B$  количество ответов размеченных данных
- $E*104B$  количество ответов пользователей

Фактический объем БД =  $A*404B + C*204 + C*312 + B*104 + C*108 + D*104 + (C-D)*10B + C*104B + E*104B$

Избыточность модели:  $(A*404B + C*204 + C*312 + B*104 + C*108 + D*104 + (C-D)*10B + C*104B + E*104B) \setminus (A*404B + C*204 + C*312 + B*104 + C*108 + D*104 + (C-D)*10B + C*104B + E*104B)$

Пусть 30% пользователей заказчики,  $A$  — количество пользователей, тогда  $B=0.3A$ , 1000 заказов, так же пусть имеется 10000 данных для заказов 7500 из которых картинки, тогда 2500 текстовых. Количество ответов пользователя 3.  $404A + 204*0.3A + 10000*204 + 1000*312 + 0.3A*104 + 1000*108 + 10000*108 + 2500*104 + 7500*100 + 1000*104 + 3000*104 = 22910000 + 207560A$

Если  $A=100$ , Тогда фактический объем занимает 43666000B=43,5MB

## 4. Запросы

1.Добавить пользователя:

```
INSERT INTO User VALUES (...)
```

2.Найти автора заказа с индексом 2

```
SELECT * FROM Order INNER JOIN User ON User.id=2
```

3.Подсчет фото с котом в названии

```
SELECT COUNT(*) as count FROM db.PhotoData WHERE objectsValue='*cat'
```

## Сравнение MongoDB и SQL модели данных

Для реализации модели данных в MongoDB мы создали в разы меньше сущностей и связей между ними, чем при реализации той же модели данных в MySQL. Так же стоит отметить что запросы в MongoDB в разы меньше нежели при использовании SQL модели.

Так же в MongoDB можно отметить небольшое, по сравнению с SQL моделью, количество требуемой памяти для хранения данных. Стоит отметить быстрое выполнение запросов в MongoDB.

Исходя из вышеперечисленного, можно сделать вывод, что для рассматриваемой задачи MongoDB подходит в большей мере.