

# **УРОК 15.1. MONGODB COMPASS AND MONGOSH. ФОРМИРОВАНИЕ ВЫБОРКИ: ЗНАКОМСТВО, ПРОЕКТИРОВАНИЕ БД**

ОСОБЕННОСТИ MONGODB COMPASS	2
НОВЫЕ ОПЕРАТОРЫ	6
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	7

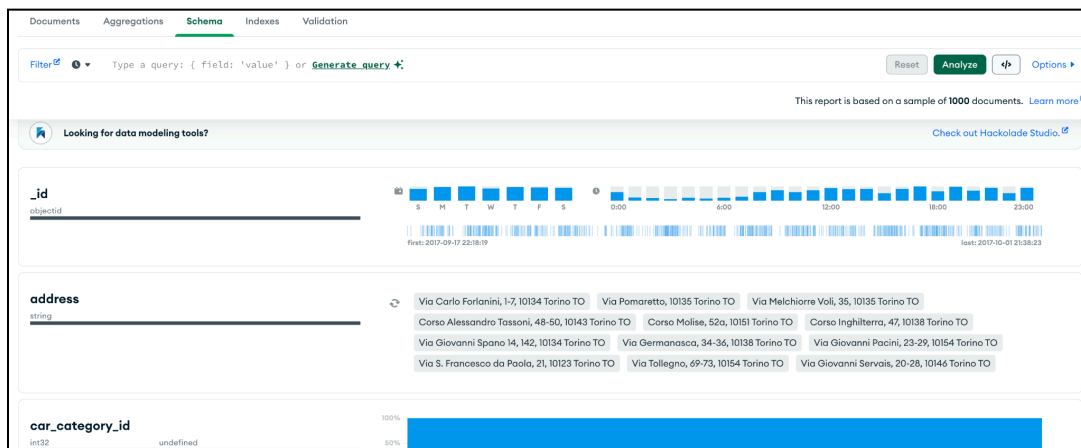


# ОСОБЕННОСТИ MONGODB COMPASS

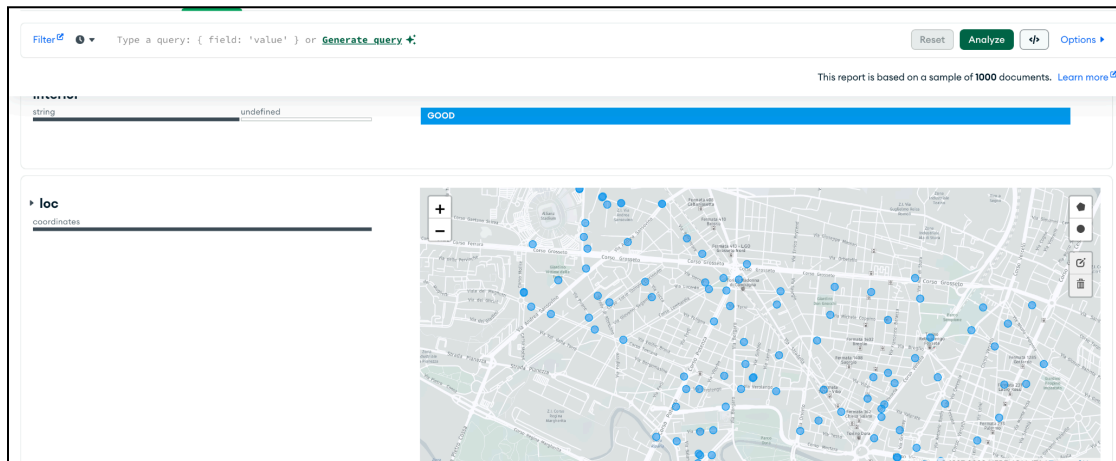
1. Разные виды отображения коллекций.  
Например тут мы видим табличный вид.

#	bookings							
	_id ObjectId	init_fuel Int32	city String	walking Object	init_address String	vender String	final_time Int32	
1	ObjectId('59bef0cd2ad8532c2...')	96	"Torino"	{ } 2 fields	"Via Chambery 93, 110, 1014."	"car2go"	1505686342	
2	ObjectId('59bef0cd2ad8532c2...')	90	"Torino"	{ } 2 fields	"Strada Altesano, 140, 101."	"car2go"	1505686489	
3	ObjectId('59bef0cd2ad8532c2...')	75	"Torino"	{ } 2 fields	"Via Guido Reni, 26A, 10136."	"car2go"	1505686390	
4	ObjectId('59bef12f2ad8532c2...')	31	"Torino"	{ } 2 fields	"Via Silvio Pellico, 14, 10."	"car2go"	1505688989	
5	ObjectId('59bef1442ad853511...')	44	"Torino"	{ } 2 fields	"Via Luigi Capriolo, 31, 10."	"enjoy"	1505686322	
6	ObjectId('59bef1442ad853511...')	45	"Torino"	{ } 2 fields	"Via Amilcare Ponchielli, 2."	"enjoy"	1505687287	
7	ObjectId('59bef1602ad8532c2...')	37	"Torino"	{ } 2 fields	"Via Andrea Sansovino, 35, ..."	"car2go"	1505688313	
8	ObjectId('59bef1952ad8532c2...')	84	"Torino"	{ } 2 fields	"Via Belfiore, 32, 10125 To."	"car2go"	1505687131	
9	ObjectId('59bef1952ad8532c2...')	100	"Torino"	{ } 2 fields	"Corso Massimo d'Azeglio, 8."	"car2go"	1505687864	
10	ObjectId('59bef1952ad8532c2...')	100	"Torino"	{ } 2 fields	"Via Luigi Tarino, 18A, 101."	"car2go"	1505686835	
11	ObjectId('59bef122ad8532c2...')	68	"Torino"	{ } 2 fields	"Corso Duca degli Abruzzi, ..."	"car2go"	1505689232	
12	ObjectId('59bef122ad853511...')	95	"Torino"	{ } 2 fields	"Corso Cosenza, 64, 10137 T."	"enjoy"	1505687287	
13	ObjectId('59bef122ad853511...')	90	"Torino"	{ } 2 fields	"Via Vicenza, 27, 10144 Tor."	"enjoy"	1505687771	
14	ObjectId('59bef122ad853511...')	52	"Torino"	{ } 2 fields	"Corso Unione Sovietica, 60."	"enjoy"	1505689218	
15	ObjectId('59bef122ad853511...')	75	"Torino"	{ } 2 fields	"Via Paolo Sacchi, 29-31, 1."	"enjoy"	1505687287	
16	ObjectId('59bef122ad853511...')	76	"Torino"	{ } 2 fields	"Corso Luigi Settembrini, 2."	"enjoy"	1505686794	
17	ObjectId('59bef122ad853511...')	91	"Torino"	{ } 2 fields	"Corso Regina Margherita, 5."	"enjoy"	1505688234	
18	ObjectId('59bef122ad853511...')	40	"Torino"	{ } 2 fields	"Corso Galileo Galilei, 8, ..."	"enjoy"	1505687823	
19	ObjectId('59bef122ad853511...')	80	"Torino"	{ } 2 fields	"Corso Galileo Ferraris, 57."	"enjoy"	1505691872	

2. Анализ документов и их полей.



Поддержка отображения координат.



### 3. Автоподстановка и подсказки.

Filter  Generate query Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 20 of 353

```
{
  "_id": "ObjectId('59bef9c02ad853511fb67e9b')",
  "city": "Torino",
  "vendor": "enjoy",
  "final_time": 1505692096,
  "plate": "EZ808DD",
  "car_category_id": 8,
  "init_time": 1505687994,
  "car_category_type_id": 1,
  "virtual_rental_type_id": 2,
  "carModelData": Array (3)
    car_name: "Fiat 500"
    init_date: 2017-09-18T00:39:54.000+00:00
    onClick_disabled: false
  virtual_rental_id: 382886,
  fuel: 35,
  final_date: 2017-09-18T01:48:16.000+00:00,
  loc: Object
    address: "Via Nizza, 8, 10125 Torino TO"
}
```

### 4. Удобное формирование запроса шаг за шагом.

Filter  Generate query Explain Reset Find Options

**Project**

**Sort**  MaxTimeMS

**Collation**  Skip  Limit

EXPORT DATA

1 - 20 of 78557

```
car_name: "Fiat 500"
fuel: 0

car_name: "Fiat 500"
fuel: 0

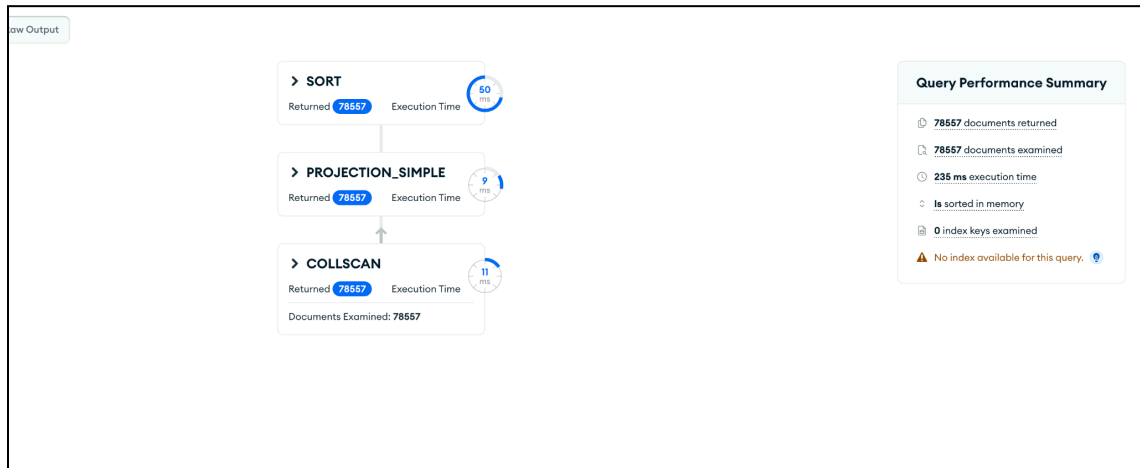
car_name: "Fiat 500"
fuel: 0

car_name: "Fiat 500"
fuel: 0

car_name: "Fiat 500"
fuel: 0
```



## 5. План выполнения запросов.



## 6. Валидация данных и поиск не соответствующих документов

Documents Aggregations Schema Indexes **Validation**

Validation Action: Error Validation Level: Strict

```
1 {
2   "$jsonSchema": {
3     "bsonType": "object"
4   }
5 }
```

Validation modified [Cancel] [Update]

**Sample documents**

Passed validation Failed validation

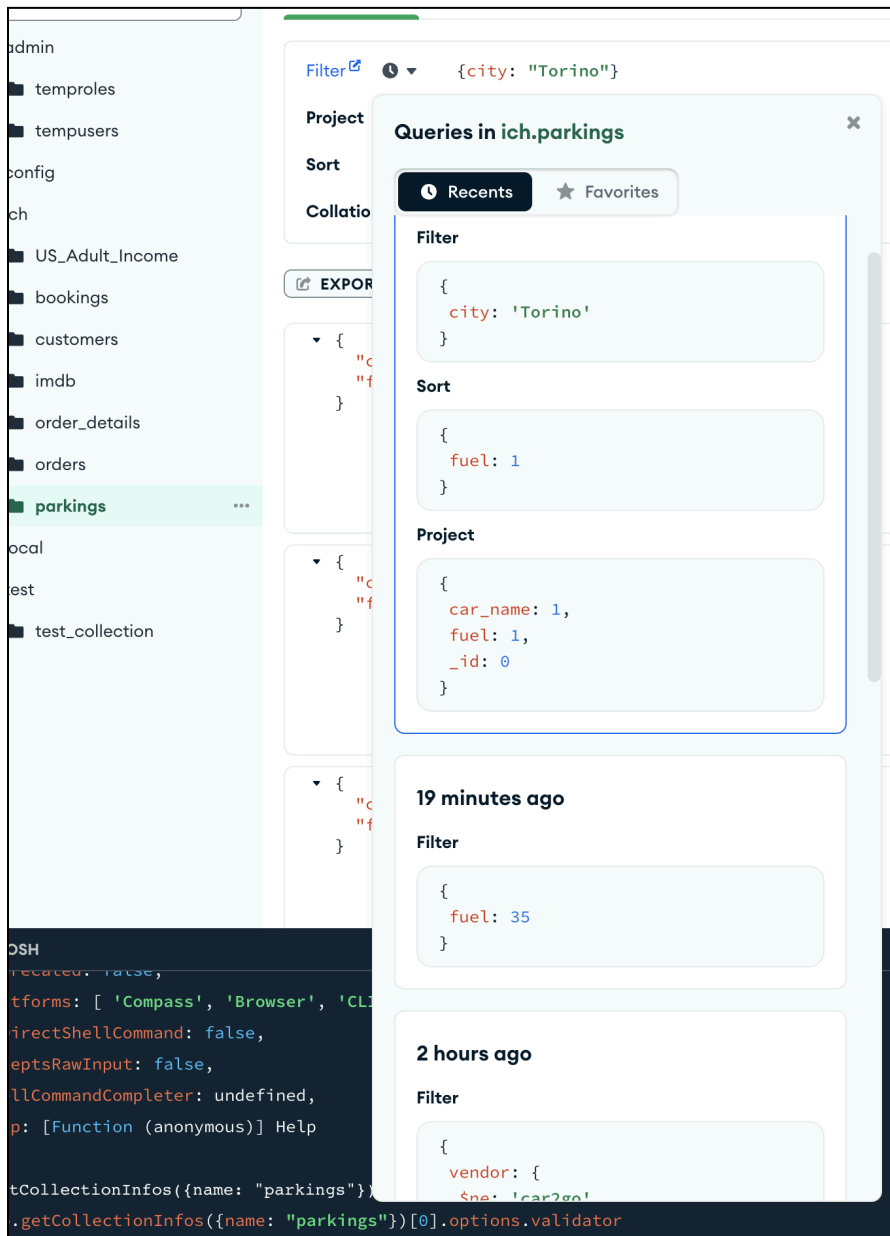
```
{
  "_id": ObjectId("59c8d2872ad8532c2a7b0f46"),
  "plate": "152/FF177NT",
  "fuel": 90,
  "vendor": "car2go",
  "final_time": 1506341696,
  "loc": Object,
  "init_time": 1506333307,
  "vin": "WME4533421K152192",
  "smartPhoneRequired": true,
  "init_date": 2017-09-25T11:55:07.000+06:00
}
```

No Preview Documents

Unset

```
{
  $jsonSchema: {
    required: ['vendor'], // the vendor field is required
    properties: {
      vendor: {
        enum: ['car2go', 'enjoy'],
        description: "can only be either 'car2go' or 'enjoy'"
      }
    }
  }
}
```

## 7. Сохранение истории запросов



The screenshot displays the MongoDB Compass interface. On the left, a sidebar shows a project structure with folders like 'temproles', 'tempusers', 'config', 'ch', 'US\_Adult\_Income', 'bookings', 'customers', 'imdb', 'order\_details', 'orders', 'parkings' (highlighted), 'local', 'test', and 'test\_collection'. The main area shows a filter for '{city: "Torino"}'. A 'Queries in ich.parkings' dialog box is open, showing a list of queries with tabs for 'Recents' and 'Favorites'. The 'Recents' tab is active, showing three queries:

- Filter:** { city: 'Torino' }
- Sort:** { fuel: 1 }
- Project:** { car\_name: 1, fuel: 1, \_id: 0 }

Below the queries, the dialog shows the timestamp '19 minutes ago' and the filter { fuel: 35 }. At the bottom, it shows '2 hours ago' and the filter { vendor: { \$ne: 'car2go' } }. The bottom of the screen shows a terminal window with a command prompt and some code snippets.



## НОВЫЕ ОПЕРАТОРЫ



**\$all** - это оператор, который выбирает документы, в которых значением поля является массив, содержащий все указанные элементы.



**\$exists** - это оператор, который соответствует документам, содержащим это поле, включая документы, в которых значение поля равно нулю, если **<boolean>** имеет значение **true**. Если **<boolean>** имеет значение **false**, запрос возвращает только те документы, которые не содержат это поле.

Пример: `{awards: {$exists :true}}`



**\$in** - это оператор, который выбирает документы, в которых значение поля равно любому значению в указанном массиве.

Пример: `{ field: { $in: [<value1>, <value2>, ... <valueN> ] } }`



## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Подключение к базе:

```
mongodb://ich1:password@mongo.itcareerhub.de/?readPreference=primary&ssl=false&authMechanism=DEFAULT&authSource=ich
```

1. Найдите все фильмы, получившие на Rotten Tomatoes оценку выше 4,5. Рецензии на Rotten Tomatoes содержатся во вложенном документе Tomatoes. Отсортируйте результаты в порядке возрастания даты выпуска.

Unset

```
FILTER: {"tomatoes.viewer.rating": {$gt: 4.5}}
SORT: {released: -1}
```

2. Для фильмов жанра «Драма» и из США, покажите их сюжет (plot), продолжительность (runtime) и название (title). Упорядочите результаты по убыванию продолжительности.

Unset

```
FILTER: {genres: 'Drama', countries: 'USA'}
PROJECT: {runtime: 1, plot: 1}
SORT: {runtime: -1}
```

3. Для фильмов жанра «Драма» и «Фэнтези» и относящихся к США, выведите их сюжет, продолжительность (время выполнения) и название. Упорядочите результаты по убыванию продолжительности.

Unset

```
FILTER: {genres: {$all: ["Drama", "Fantasy"]}, countries: "USA" }
PROJECT: {runtime: 1, plot: 1}
SORT: {runtime: -1}
```



4. Найдите фильмы, удовлетворяющие всем следующим условиям:

- были опубликованы между 1900 и 1910 годами.
- рейтинг imdb выше 9.0
- имеют награды или номинации

Выведите название, год выхода и длительность и не показывайте идентификатор документа .

Отсортируйте результаты по возрастанию рейтинга IMDB.

Unset

```
FILTER {year: {$in: ["1900", "1901", "1902", "1903", "1904", "1905", "1906", "1907",  
"1907", "1908", "1909", "1910"]}, "imdb.rating": {$gte: 7.0}}
```

```
PROJECT {runtime:1, title: 1, year:1, _id:0}
```

```
SORT {"imdb.rating": 1}
```