

## **Project 2**

**By Fayzulla Abdurakhimov, Charu Mangla,  
Edgar Martinez, Garima Gupta and Akshitha Morusu**

**BUAN 6320 Database Foundations**

**Professor Hossein Kamalzadeh**

**University of Texas at Dallas**

# TABLE OF CONTENTS

1. Part 1- Indexing and Query Timing.....	2
1.1 List of current indexes and columns.....	2
1.2 Commonalities between columns.....	3
1.3 Copy of database without indexes.....	4
1.4 Five queries to test database.....	6
1.5 Execute times between both databases.....	6
1.6 Creating indexes for database 1.....	10
1.7 Queries to test new indexes.....	11
1.8 Execute times between both databases.....	11
1.9 Conclusion of our findings.....	14
2. Part 2- MongoDB and MQL.....	15
2.1 Explore dataset.....	15
2.2 Why it is better to use non-relational databases.....	15
2.3 Importing the dataset.....	16
2.4 List attributes of your database common among all documents.....	16
2.4.1 Provide some of the values of these fields.....	17
2.5 List attributes of your database not common among all documents.....	19
2.5.1 Provide some of the values of these fields.....	20
2.6 Five queries using key-value pairs found.....	23
2.7 Five update queries to update values in database.....	25
2.8 Five queries to insert new documents into the database.....	27
2.9 Five delete queries to remove documents for database.....	29

## Part 1 Indexing and query timing

### 1.1 List of current indexes and columns

Incident table:

Indexes in Table				
Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	INCIDENT_ID
<input checked="" type="checkbox"/>	of_fk	BTREE	NO	OFFENCE_ID
<input checked="" type="checkbox"/>	d_fk	BTREE	NO	Date_ID
<input checked="" type="checkbox"/>	l_fk	BTREE	NO	LOCATION_ID

Index Details				
Key Name:	PRIMARY	Index Type:	BTREE (clustered)	Packed:
Allows NULL:		Unique:	YES	
Cardinality:	10208	Comment:		User Comment:

  

Columns in table				
Column	Type	Nullable	Indexes	
INCIDENT_NUMBER	varchar(45)	YES		
SHOOTING	varchar(45)	YES		
OFFENCE_ID	int	YES	of_fk	
Date_ID	int	YES	d_fk	
LOCATION_ID	int	YES	l_fk	
INCIDENT_ID	int	NO	PRIMARY	

Date table:

Indexes in Table				
Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	Date_ID

Index Details				
Key Name:	PRIMARY	Index Type:	BTREE (clustered)	Packed:
Allows NULL:		Unique:	YES	
Cardinality:	9970	Comment:		User Comment:

  

Columns in table				
Column	Type	Nullable	Indexes	
OCCURRED_ON_DATE	varchar(45)	YES		
DAY_OF_WEEK	varchar(45)	YES		
Date_ID	int	NO	PRIMARY	

## Offense table:

The screenshot shows the 'Indexes in Table' and 'Columns in table' sections for the Offense table.

**Indexes in Table:**

Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	OFFENCE_ID
<input checked="" type="checkbox"/>	OFFENCE_ID_UNIQUE	BTREE	YES	OFFENCE_ID

**Index Details:**

- Key Name: OFFENCE\_ID\_UNIQUE
- Index Type: BTREE
- Packed:
- Allows NULL:
- Cardinality: 263
- Comment:
- User Comment:

**Columns in table:**

Column	Type	Nullable	Indexes
OFFENSE_CODE	int	YES	
OFFENSE_CODE_GR...	varchar(100)	YES	
OFFENSE_DESCRIPTI...	varchar(100)	YES	
OFFENCE_ID	int	NO	PRIMARY, OFFENCE_ID_UNIQUE

**Create Index for Selected Columns...**

## Location table:

The screenshot shows the 'Indexes in Table' and 'Columns in table' sections for the Location table.

**Indexes in Table:**

Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE (clustered)	YES	LOCATION_ID

**Index Details:**

- Key Name: PRIMARY
- Index Type: BTREE (clustered)
- Packed:
- Allows NULL:
- Cardinality: 10142
- Comment:
- User Comment:

**Columns in table:**

Column	Type	Nullable	Indexes
Location	varchar(45)	YES	
DISTRICT	varchar(45)	YES	
REPORTING_AREA	varchar(45)	YES	
STREET	varchar(45)	YES	
LOCATION_ID	int	NO	PRIMARY

### 1.2 Commonalities between columns:

Between the four tables in the database, there are 8 columns that are automatically indexed by the DBMS. On the incident table (which we consider to be the most important as it aggregates all of the values from the other tables), we have 4 indexes for the incident ID, offense ID, date ID, and location ID columns. The remaining tables have indexes for their

respective ID column. For instance, the date table has an index for its date ID column and the location table has an index for its location ID column. However, the location has two indexes for the offense ID, likely because it represents both a unique and primary key. Whereas the other tables were only primary keys that may have duplicate rows. Therefore, the indexes have in common that they are all primary key columns. These primary key columns are indexed to better optimize data storage for the primary key.

### **1.3 Copy of database without index**

**Please refer question 1.1 to see Database 1**

Database 2:

Incident table:

## Date table:

**Indexes in Table**

Visible	Key	Type	Uni...	Columns

**Index Details**

Key Name:  
Index Type:  
Allows NULL:  
Cardinality:  
Comment:  
User Comment:

**Columns in table**

Column	Type	Nullable	Indexes
OCCURRED_ON_DATE	varchar(45)	YES	
DAY_OF_WEEK	varchar(45)	YES	
Date_ID	int	NO	

## Offense table:

**Indexes in Table**

Visible	Key	Type	Uni...	Columns

**Index Details**

Key Name:  
Index Type:  
Allows NULL:  
Cardinality:  
Comment:  
User Comment:

**Columns in table**

Column	Type	Nullable	Indexes
OFFENSE_CODE	int	YES	
OFFENSE_CODE_GRP	varchar(100)	YES	
OFFENSE_DESCRIPTION	varchar(100)	YES	
OFFENCE_ID	int	NO	

## Location table:

**Indexes in Table**

Visible	Key	Type	Uni...	Columns

**Index Details**

Key Name:  
Index Type:  
Allows NULL:  
Cardinality:  
Comment:  
User Comment:

**Columns in table**

Column	Type	Nullable	Indexes
Location	varchar(45)	YES	
DISTRICT	varchar(45)	YES	
REPORTING_AREA	varchar(45)	YES	
STREET	varchar(45)	YES	
LOCATION_ID	int	NO	

#### **1.4 and 1.5 Five queries to test database**

Query 1

```
Select I.DISTRICT, I.REPORTING_AREA, I.STREET, i.SHOOTING, d.OCCURRED_ON_DATE,  
o.OFFENSE_DESCRIPTION  
from incident2 i  
left join location2 l  
on I.LOCATION_ID = i.LOCATION_ID  
left join Date1 d  
on i.Date_id = d.Date_id  
left join offence1 o  
on i.OFFENCE_ID = o.OFFENCE_ID;
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.016	0.032
2	0.000	0.000
3	0.000	0.016
4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000
8	0.000	0.015
9	0.000	0.000
10	0.000	0.015
Average	0.0016	0.0078

Query 2

```
Select I.DISTRICT, I.SHOOTING
from incident2 i
left join location2 l
on I.LOCATION_ID = l.LOCATION_ID;
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.000	0.016
2	0.000	0.000
3	0.000	0.000
4	0.000	0.000
5	0.000	0.016
6	0.000	0.000
7	0.000	0.016
8	0.000	0.015
9	0.000	0.000
10	0.000	0.016
Average	0.000	0.0079

Query 3

```
Select i.INCIDENT_NUMBER, o.OFFENSE_CODE_GROUP, d.OCCURRED_ON_DATE
from incident2 i
left join Date1 d
on i.Date_id = d.Date_id
left join offence1 o
on i.OFFENCE_ID = o.OFFENCE_ID;
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.000	0.000
2	0.000	0.000
3	0.000	0.000
4	0.015	0.000
5	0.000	0.015
6	0.000	0.000
7	0.000	0.000
8	0.000	0.016
9	0.000	0.000
10	0.000	0.000
Average	0.0015	0.0031

Query 4

```
Select i.INCIDENT_NUMBER, i.REPORTING_AREA, o.OFFENSE_CODE,  
from incident2 i  
left join location2 l  
on l.LOCATION_ID = i.LOCATION_ID  
left join offence1 o  
on i.OFFENCE_ID = o.OFFENCE_ID;
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.000	0.000
2	0.016	0.015
3	0.000	0.000

4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.015	0.016
8	0.000	0.000
9	0.000	0.016
10	0.000	0.015
Average	0.0031	0.0062

### Query 5

```
Select i INCIDENT_NUMBER, i STREET, o OFFENSE_DESCRIPTION, d DAY_OF_WEEK
from incident2 i
left join location2 l
on l LOCATION_ID = i LOCATION_ID
left join Date1 d
on i Date_id = d Date_id
left join offence1 o
on i OFFENCE_ID = o OFFENCE_ID;
```

Time:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.000	0.016
2	0.000	0.016
3	0.000	0.000
4	0.000	0.000
5	0.000	0.015
6	0.000	0.016

7	0.000	0.000
8	0.000	0.000
9	0.016	0.000
10	0.000	0.000
Average	0.0016	0.0063

## 1.6 Creating indexes for database 1

Date table: created index for “occured on date”

Table: date1

Incident table: created index for “incident number”

Location table: created index for “reporting area”

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the Navigator with various schemas and tables. The main window is focused on the 'location2' table under the 'project' schema. The 'Indexes' tab is selected. A red box highlights the second index entry, 'idx\_location2\_REPORTING\_AREA', which is defined as a BTREE index on the 'REPORTING\_AREA' column.

Offense table: created index for “offense code group”

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the Navigator with various schemas and tables. The main window is focused on the 'offence1' table under the 'project' schema. The 'Indexes' tab is selected. A red box highlights the third index entry, 'idx\_offence1\_OFFENSE\_CODE\_GROUP', which is defined as a BTREE index on the 'OFFENSE\_CODE\_GROUP' column.

### 1.7 and 1.8 Queries to test new indexes

Query 1

*SELECT \* FROM date1*

*where OCCURRED\_ON\_DATE between '2018-01-01' and '2018-12-12';*

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.015	0.000
2	0.000	0.015
3	0.000	0.000

4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000
8	0.000	0.0
9	0.000	0.000
10	0.000	0.016
Average	0.0015	0.0031

## Query 2

*SELECT \* FROM incident2*

*WHERE (left(INCIDENT\_NUMBER,2) % 2) = 0;*

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.000	0.000
2	0.016	0.016
3	0.000	0.000
4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000
8	0.000	0.000
9	0.000	0.000
10	0.016	0.000
Average	0.0032	0.0016

### Query 3

```
SELECT * FROM location2  
where REPORTING_AREA between '100' and 300";
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.016	0.016
2	0.000	0.000
3	0.000	0.000
4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000
8	0.000	0.016
9	0.000	0.000
10	0.000	0.000
Average	0.0016	0.0032

### Query 4

```
SELECT * FROM offence1  
where OFFENSE_CODE_GROUP = 'Auto Theft' or OFFENSE_CODE_GROUP = 'Property  
Lost';
```

Times:

Instance	Database 1 (with index)	Database 2 (no index)
1	0.016	0.015
2	0.000	0.000

3	0.000	0.000
4	0.000	0.000
5	0.000	0.000
6	0.000	0.016
7	0.000	0.000
8	0.000	0.000
9	0.000	0.000
10	0.000	0.000
Average	0.0016	0.0031

### ***1.9 Conclusion of our findings***

Based on our findings from the parts above, we have found that on average the execute times of the queries were shorter for the database with indexes. The database without indexes exhibited longer execute times likely because the index served to optimize the DBMS.

We also noticed that the more complex queries with more join statements resulted in a larger variance between the index and no index databases (as seen by query 1 in part 1.4). This is likely where efficiency becomes a larger factor. Comparatively, the simpler queries with less joins or less stringent where statements resulted in less difference between the index and no index databases.

## Part 2 MongoDB and MQL

### 2.1 and 2.2 Explore dataset and explain why it is better to use non-relational databases

With the evolving IoT ecosystem where the data is not anymore only in tabular form, it already has taken a form of images, text and is dynamic in nature where we adding or modifying information every microsecond, we require a flexible language/tool which can easily adapt and give us result that fast and plays with high volume of data too.

We chose the Bikez dataset which has 38624 documents with a lot of fields, and are not common between various modules. For updating, sorting, understanding the uniqueness of this dataset, and mapping a relation between them, it is ideal for us to use such a tool which can understand its limitations and expand and emphasize the anomalies of the dataset and can project the true essence of the dataset. While working with 38K documents, MongoDB was flexible enough to address the expansion and concurrent use of the dataset without any downtime.

Some of the queries we executed, where we actually felt that it was better to use a non-relational database are following :

- Distinct value : finding that after scanning all the documents, the result had unique values of some of the key fields like year, rating, etc.
- Common value - finding that we were able to scan and return the values which are common among all the documents
- Modification of the documents - we were able to insert, update and delete all the documents at the same time with a single line of query

The non relational database gave us such freedom through which we were able to make sense of our data with ease. This dataset, which was disorganized in nature, had multiple documents with values which were unique in nature and had a high volume of documents. Using MongoDB we were actually able to draw sense out of this unstructured data.

### **2.3 and 2.4 Importing the dataset and finding common attributes among documents**

In order to import the dataset we used MongoDB compass by creating a database and then assigning collection for Bikez.com and PakWheels

**// Total Number of Documents**

```
> db.getCollection("temp").find({}).count()
```

```
38624
```

**Problem 2.4**

```
> db.getCollection("temp").find( {_id:{$exists:true} }).count()
```

```
38624
```

```
> db.getCollection("temp").find( {Model:{$exists:true} }).count()
```

```
38624
```

```
> db.getCollection("temp").find( {"Rating":{$exists:true} }).count()
```

```
38624
```

```
> db.getCollection("temp").find( {"Year":{$exists:true} }).count()
```

```
38624
```

```
> db.getCollection("temp").find( {"Category":{$exists:true} }).count()
```

```
38624
```

```
38624
> db.getCollection("temp").find( {_id:{$exists:true} }).count()
38624
> db.getCollection("temp").find( {Model:{$exists:true} }).count()
38624
> db.getCollection("temp").find( {"Rating":{$exists:true} }).count()
38624
> db.getCollection("temp").find( {"Year":{$exists:true} }).count()
38624
> db.getCollection("temp").find( {"Category":{$exists:true} }).count()
38624
> |
```

\_id, Model, Rating, Year, and Category are some of the database attributes (fields/properties) that are shared by all documents. There are 38624 documents in total. The count of the

characteristics (\_id, Model, Rating, Year, Category) after running the query is 38624, indicating that these attributes are present in all documents.

#### 2.4.1 Provide some of values in these fields

##### \_id

```
> db.getCollection("temp").distinct('_id')
```

```
ObjectId("627d488ae2cbbbc0d76c74c7"),
ObjectId("627d488ae2cbbbc0d76c74c8"),
ObjectId("627d488ae2cbbbc0d76c74c9"),
ObjectId("627d488ae2cbbbc0d76c74ca"),
ObjectId("627d488ae2cbbbc0d76c74cb"),
ObjectId("627d488ae2cbbbc0d76c74cc"),
ObjectId("627d488ae2cbbbc0d76c74cd"),
ObjectId("627d488ae2cbbbc0d76c74ce"),
ObjectId("627d488ae2cbbbc0d76c74cf"),
ObjectId("627d488ae2cbbbc0d76c74d0"),
ObjectId("627d488ae2cbbbc0d76c74d1"),
ObjectId("627d488ae2cbbbc0d76c74d2"),
ObjectId("627d488ae2cbbbc0d76c74d3"),
ObjectId("627d488ae2cbbbc0d76c74d4"),
ObjectId("627d488ae2cbbbc0d76c74d5"),
ObjectId("627d488ae2cbbbc0d76c74d6"),
ObjectId("627d488ae2cbbbc0d76c74d7"),
ObjectId("627d488ae2cbbbc0d76c74d8"),
ObjectId("627d488ae2cbbbc0d76c74d9"),
ObjectId("627d488ae2cbbbc0d76c74da"),
ObjectId("627d488ae2cbbbc0d76c74db"),
ObjectId("627d488ae2cbbbc0d76c74dc"),
ObjectId("627d488ae2cbbbc0d76c74dd"),
ObjectId("627d488ae2cbbbc0d76c74de"),
ObjectId("627d488ae2cbbbc0d76c74df"),
ObjectId("627d488ae2cbbbc0d76c74e0"),
ObjectId("627d488ae2cbbbc0d76c74e1"),
ObjectId("627d488ae2cbbbc0d76c74e2"),
ObjectId("627d488ae2cbbbc0d76c74e3")
```

##### **Model**

```
> db.getCollection("temp").distinct('Model')
```

```
"Zündapp Elastic 198",
"Zündapp GS 100 Scrambler",
"Zündapp GS 125",
"Zündapp K 200 Kardan",
"Zündapp K 211",
"Zündapp K 400 Kardan",
"Zündapp K 500",
"Zündapp K 600",
"Zündapp K 800",
"Zündapp KKS 500 Kardan Sport",
"Zündapp KS 100",
"Zündapp KS 125 Sport",
"Zündapp KS 175",
"Zündapp KS 350",
"Zündapp KS 50 Super Sport type 517-20",
"Zündapp KS 500 Kardan Sport",
"Zündapp KS 601",
"Zündapp KS 601 Kardan Sport",
"Zündapp KS 750",
"Zündapp KS 750 Kardan Sport",
"Zündapp Norma Luxsus Db234",
"Zündapp RS 50 Roller Super",
"Zündapp Roller Super",
"Zündapp Z 2 G",
```

## Rating

```
> db.getCollection("temp").distinct('Rating')
```

```
"4.1 Check out the detailed rating of racing track capabilities, engine performance, accident risk, etc. Compare with any other bike.",  
"4.2 Check out the detailed rating of reliability, maintenance costs, value for money, etc. Compare with any other motorcycle.",  
"4.2 See the detailed rating of design and look, engine performance, reliability, fun-factor, etc. Compare with any other bike.",  
"4.2 See the detailed rating of design and look, maintenance cost, engine performance, etc. Compare with any other bike.",  
"4.2 See the detailed rating of engine performance, design and look, accident risk, etc. Compare with any other motorcycle.",  
"4.2 View the detailed rating of value for money, design and look, reliability, etc. Compare with any other bike.",  
"4.3 Check out the detailed rating of racing track capabilities, engine performance, accident risk, etc. Compare with any other bike.",  
"4.3 Check out the detailed rating of reliability, maintenance costs, value for money, etc. Compare with any other motorcycle.",  
"4.3 See the detailed rating of design and look, engine performance, reliability, fun-factor, etc. Compare with any other bike.",  
"4.3 See the detailed rating of engine performance, design and look, accident risk, etc. Compare with any other motorcycle.",  
"4.3 See the detailed rating of touring capabilities, reliability, accident risk, etc. Compare with any other motorbike.",  
"4.4 Check out the detailed rating of off-road capabilities, engine performance, maintenance cost, etc. Compare with any other bike.",  
"4.4 Check out the detailed rating of reliability, maintenance costs, value for money, etc. Compare with any other motorcycle.",  
"4.4 See the detailed rating of design and look, engine performance, reliability, fun-factor, etc. Compare with any other bike.",  
"4.5 Check out the detailed rating of racing track capabilities, engine performance, accident risk, etc. Compare with any other bike.",  
"4.6 See the detailed rating of touring capabilities, reliability, accident risk, etc. Compare with any other motorbike."  
"Do you know this bike?Click here to rate it. We miss 1 vote to show the rating.",  
"Do you know this bike?Click here to rate it. We miss 2 votes to show the rating."
```

## Year

```
> db.getCollection("temp").distinct('Year')
```

```
"2003",  
"2004",  
"2005",  
"2006",  
"2007",  
"2008",  
"2009",  
"2010",  
"2011",  
"2012",  
"2013",  
"2014",  
"2015",  
"2016",  
"2017",  
"2018",  
"2019",  
"2020",  
"2021"
```

## Category

```
> db.getCollection("temp").distinct('Category')
```

```
"ATV",  
"Allround",  
"Classic",  
"Cross / motocross",  
"Custom / cruiser",  
"Enduro / offroad",  
"Minibike, cross",  
"Minibike, sport",  
"Naked bike",  
"Prototype / concept model",  
"Scooter",  
"Speedway",  
"Sport",  
"Sport touring",  
"Super motard",  
"Touring",  
"Trial",  
"Unspecified category"
```

## **2.5 List attributes of the database not common among all documents**

```
> db.getCollection("temp").find( {"Trail":{$exists:true} }).count()  
7238  
  
> db.getCollection("temp").find( {"Displacement":{$exists:true} }).count()  
37777  
  
> db.getCollection("temp").find( {"Torque":{$exists:true} }).count()  
16089  
  
> db.getCollection("temp").find( {"Gearbox":{$exists:true} }).count()  
32331  
  
> db.getCollection("temp").find( {"Comments":{$exists:true} }).count()  
12580  
  
> db.getCollection("temp").find( {"Engine type":{$exists:true} }).count()  
38614  
  
> db.getCollection("temp").find( {"Fuel system":{$exists:true} }).count()  
27308
```

```
> db.getCollection("temp").find( {"Trail":{$exists:true} }).count()  
7238  
> db.getCollection("temp").find( {"Displacement":{$exists:true} }).count()  
37777  
> db.getCollection("temp").find( {"Torque":{$exists:true} }).count()  
16089  
> db.getCollection("temp").find( {"Gearbox":{$exists:true} }).count()  
32331  
> db.getCollection("temp").find( {"Comments":{$exists:true} }).count()  
12580  
> db.getCollection("temp").find( {"Fuel system":{$exists:true} }).count()  
27308  
> db.getCollection("temp").find( {"Engine type":{$exists:true} }).count()  
38614  
> |
```

Trail, Displacement, Torque, Gearbox, Comments, Fuel system, and Engine type are some of the elements (fields/properties) of our database that are not common across all documents. There are 38624 documents in all. We get a count of less than 38624 for these attributes,

indicating that they are not present in all documents, i.e., they are not common across all documents.

### **2.5.1 Provide some of values in these fields**

#### **Trail**

```
> db.getCollection("temp").di
```

```
stinct('Trail')
```

```
"92 mm (3.6 inches)",  
"93 mm (3.6 inches)",  
"93 mm (3.7 inches)",  
"94 mm (3.7 inches)",  
"95 mm (3.7 inches)",  
"95 mm (3.8 inches)",  
"96 mm (3.8 inches)",  
"97 mm (3.8 inches)",  
"98 mm (3.8 inches)",  
"98 mm (3.9 inches)",  
"99 mm (3.9 inches)"
```

#### **Displacement**

```
> db.getCollection("temp").distinct('Displacement')
```

```
"996.0 ccm (60.78 cubic inches)",  
"996.1 ccm (60.78 cubic inches)",  
"996.4 ccm (60.80 cubic inches)",  
"997.0 ccm (60.84 cubic inches)",  
"997.6 ccm (60.87 cubic inches)",  
"998.0 ccm (60.90 cubic inches)",  
"998.2 ccm (60.91 cubic inches)",  
"998.6 ccm (60.93 cubic inches)",  
"999.0 ccm (60.96 cubic inches)",  
"999.6 ccm (61.00 cubic inches)",  
"999.7 ccm (61.00 cubic inches)",  
"999.8 ccm (61.01 cubic inches)"
```

## Torque

```
> db.getCollection("temp").distinct('Torque')
```

```
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs)" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 2350 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 3500 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 3750 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 5800 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 6000 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 7000 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 7500 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 7700 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 9000 RPM" ,  
"99.0 Nm (10.1 kgf-m or 73.0 ft.lbs) @ 9500 RPM" ,  
"99.1 Nm (10.1 kgf-m or 73.1 ft.lbs) @ 7700 RPM" ,  
"99.1 Nm (10.1 kgf-m or 73.1 ft.lbs) @ 8200 RPM"
```

## Gearbox

```
> db.getCollection("temp").distinct('Gearbox')
```

```
"1-speed" ,  
"10-speed" ,  
"100-speed" ,  
"2-speed" ,  
"2-speed automatic" ,  
"2000-speed" ,  
"3-speed" ,  
"3-speed automatic" ,  
"4-speed" ,  
"4-speed with reverse" ,  
"400-speed" ,  
"5-speed" ,  
"5-speed with reverse" ,  
"6-speed" ,  
"6-speed with reverse" ,  
"7-speed" ,  
"8-speed" ,  
"Automatic"
```

## Comments

```
> db.getCollection("temp").distinct('Comments')
```

```
ce of the 125cc ohv machines was improved, they proved very popular. Three variants were offered: Turismo, Lusso, and Special.",  
"info taken from Honda Service Manual 61MM201",  
"insanely fast, lots of excess power.",  
"it's a really quick bike for the size of his engine",  
"limited two years production. Automatic transmission.",  
"made by Yamaha Thailand model code 4XC",  
"manufactured 1200 pcs",  
"manufactured in singapore, popular engine for bucket racers, digital display",  
"miles per gallon is 190 @ 20mph (optimum)\r\nThe Rockford Motors Tora series was offered in two models. The Standard and the Deluxe which was the street legal version; the above bike is standard. This bike was manufactured only from 1971-1974. The year on this bike above is unspecified. This bike is absolutely rare and hard to find.\r\n\r\n",  
"not sure why I have to submit this bike again but here it is.",  
"old work horse",  
"sexy motorcycle",  
"the only place on the bike that says what it is.... is on the exhaust and it claims to be a HS 250; and i dont know the year!",  
"top speed is what I've read somewhere; 0-100 Km/h en 1/4 mile are the specs of the XJ 700 X and might differ somewhat",  
"unleaded 98 Octane\r\n",  
""Rain", "Road", "Dynamic", "Race"" and "Race Pro 1-3" riding modes. Dynamic Traction Control. Cruise control. Heated grips."  
]
```

## Fuel System

```
> db.getCollection("temp").distinct('Fuel system')
```

```
"Turbo",  
"Turbo. BING",  
"Turbo. Bosch common rail injection. Turbo charger with intercooler.",  
"Turbo. Common rail injection. Turbo charger with intercooler.",  
"Turbo. KKK Acabion Extended",  
"Turbo. Motronic-controlled injection",  
"Turbo. Rotrex radial compressor C-series, detatchable",  
"Turbo. Supercharged",  
"Turbo. Supercharger technology.",  
"Turbo. Turbo 1.25 bar ",  
"Turbo. Turbo 1.25 bar, intercooler ",  
"Turbo. Turbo charger with intercooler."
```

## Engine type

```
> db.getCollection("temp").distinct('Engine type')
```

```
"Diesel",  
"Dual disk Wankel",  
"Electric",  
"Four cylinder boxer, four-stroke",  
"Four cylinder boxer, two-stroke",  
"Gas turbine",  
"In-line four, four-stroke",  
"In-line four, two-stroke",  
"In-line six, four-stroke",  
"In-line six, two-stroke",  
"In-line three, four-stroke",  
"In-line three, two-stroke",  
"Radial",  
"Single cylinder, four-stroke",  
"Single cylinder, two-stroke",  
"Single disk Wankel",  
"Six cylinder boxer, four-stroke",  
"Square four cylinder",  
"Twin, four-stroke",  
"Twin, two-stroke",  
"Two cylinder boxer, four-stroke",  
"Two cylinder boxer, two-stroke",  
"V10, four-stroke",  
"V2, four-stroke",  
"V2, two-stroke",  
"V3, two-stroke",  
"V4, four-stroke",  
"V4, two-stroke",  
"V6, four-stroke",
```

## 2.6 Five queries using key-value pairs found

### QUERY 1

```
> db.getCollection("temp").find({Model: "proEco Emma"})
```

```
> db.getCollection("temp").find({Model: "proEco Emma"})
{ "_id" : ObjectId("627d4888e2cbbcd97dc12be"), "Model" : "proEco Emma", "Year" : "2013", "Category" : "Scooter", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Engine type" : "Electric", "Engine details" : "Wheel hub motor", "Power" : "2.7 HP (2.0 kW)", "Top speed" : "45.0 km/h (28.0 mph)", "Cooling system" : "Air", "Front brakes" : "Single disc", "Seat" : "Dual", "Weight incl. oil, gas, etc" : "145.0 kg (319.7 pounds)", "Seat height" : "780 mm (30.7 inches) If adjustable, lowest setting.", "Overall height" : "11130 mm (44.5 inches)", "Overall length" : "1935 mm (76.2 inches)", "Overall width" : "710 mm (28.0 inches)", "Ground clearance" : "100 mm (3.9 inches)", "Color options" : "Red, blue, black, silver, beige, pink, yellow, olive, brown", "Electrical" : "60 V, 40 Ah Silicon battery.", "Comments" : "Range 70 km. Charge time 5-8 hours. More than 450 charge cycles. German brand.", "Insurance costs" : "Compare US insurance quotes from the nation's top providers.", "Finance options" : "Compare US motorcycle loan quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 13 proEco Emma discussion group or the general proEco discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
> |
```

### QUERY 2

```
> db.getCollection("temp").find({Year: "2021"})
```

```
> db.getCollection("temp").find({Year: "2021"})
{ "_id" : ObjectId("627d4888e2cbbcd97dc0e05"), "Model" : "AJP PR7", "Year" : "2021", "Category" : "Sport", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "600.0 cm (36.5 cubic inches)", "Engine type" : "Single cylinder, four-stroke", "Power" : "40.0 HP (30.0 kW)", "Top speed" : "180 km/h (111.8 mph)", "Cooling system" : "Liquid", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Frame type" : "Aluminum/steel", "Rake (fork angle)" : "16.5°", "Front suspension" : "ZF Sachs 010 mm close-circuit", "Rear suspension" : "ZF Sachs proc. system", "Rear wheel travel" : "920 mm (36.2 inches) If adjustable, lowest setting.", "Ground clearance" : "310 mm (12.2 inches)", "Wheelbase" : "1540 mm (60.6 inches)", "Fuel capacity" : "17.00 litres (4.49 gallons)", "Color options" : "White, black", "Starter" : "Electric & kick", "Comments" : "Portuguese made bike.", "Insurance costs" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 13 AJP PR7 discussion group or the general AJP discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0e05"), "Model" : "Aprilia RS 125", "Year" : "2021", "Category" : "Sport", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "124.2 cm (7.5 cubic inches)", "Engine type" : "Single cylinder, four-stroke", "Power" : "19.0 HP (14.0 kW)", "Top speed" : "190 km/h (118.0 mph)", "Cooling system" : "Water", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Frame type" : "Aluminum/steel", "Rake (fork angle)" : "22.0°", "Front suspension" : "Upside down hydraulic fork, 41 mm.", "Front wheel travel" : "110 mm (4.3 inches)", "Rear suspension" : "Swingarm. Hydraulic monoshock with adjustable spring preload.", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "17.00 litres (4.49 gallons)", "Color options" : "Black, silver, beige, pink, yellow, olive, brown", "Starter" : "Electric", "Comments" : "Compare US motorcycle loan quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 21 Aprilia RS 125 discussion group or the general Aprilia discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0e05"), "Model" : "Aprilia dorsoduro 900", "Year" : "2021", "Category" : "Supermoto", "Rating" : "Price as new: Euro 9990. MSRP depend on country, taxes, accessories, etc.", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "990 longitudinal 60° V twin.", "Power" : "95.2 HP (69.5 kW) @ 9500 RPM", "Torque" : "99.0 Nm (9.2 kgf-m or 66.4 ft.lbs) @ 6500 RPM", "Compression" : "11.0:1", "Bore x stroke" : "92.0 x 67.0 mm (3.6 x 2.7 inches)", "Valves per cylinder" : "4", "Fuel system" : "Injection. Integrated engine management system. Electronic fuel injection with ride-by-wire electronic throttle control.", "Front control" : "Double Overhead Cams/Twin Cam (DOHC)", "Ignition" : "Digital electronic, integrated in the fuel injection system.", "Lubrication system" : "Oil pump", "Gearbox" : "6-speed", "Transmission type, final drive" : "Chain", "Frame type" : "Hydroformed multi-platé wet clutch.", "Rake (fork angle)" : "22.0°", "Front suspension" : "Telescopic fork", "Front wheel travel" : "120 mm (4.7 inches)", "Rear suspension" : "Swingarm. Hydraulic monoshock with adjustable spring preload.", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "19.00 litres (4.99 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 21 Aprilia dorsoduro 900 discussion group or the general Aprilia discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0e05"), "Model" : "Aprilia RSV4", "Year" : "2021", "Category" : "Sport", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "1099.0 cc
```

### QUERY 3

```
> db.getCollection("temp").find({"Engine type": "Diesel"})
```

```
> db.getCollection("temp").find({"Engine type": "Diesel"})
{ "_id" : ObjectId("627d4888e2cbbcd97dc0effc"), "Model" : "Neander Turbo Diesel 1400", "Year" : "2019", "Category" : "Custom / cruiser", "Rating" : "Do you know this bike?Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "1400.0 cm (85.43 cubic inches)", "Engine type" : "Diesel", "Power" : "11.0 HP (8.1 kW) @ 2000 RPM", "Torque" : "210.0 Nm (18.0 kgf-m or 137.4 ft.lbs) @ 1600 RPM", "Top speed" : "160 km/h (102 mph)", "Fuel consumption" : "16.00 litres/100km (40.6 km/l or 94.09 mpg)", "Gearbox" : "5-speed", "Transmission type, final drive" : "Belt", "Clutch" : "Multidisc", "Front suspension" : "Telescopic fork", "Rear suspension" : "White Powder shocks, adjustable", "Front brakes" : "Single disc. Brembo", "Rear brakes" : "Single disc. Brembo", "Rake (fork angle)" : "25.0°", "Front wheel travel" : "150 mm (5.9 inches)", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "19.00 litres (4.99 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 21 Neander Diesel 1400 discussion group or the general Neander discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0effc"), "Model" : "Sommer Diesel S10", "Year" : "2019", "Category" : "Allround", "Rating" : "Price as new: Euro 9800. MSRP depend on country, taxes, accessories, etc.", "Rating" : "Do you know this bike?Click here to rate it. We miss 1 vote to show the rating.", "Displacement" : "516.0 cm (31.49 cubic inches)", "Engine type" : "Diesel", "Engine details" : "Hatz 100HP engine", "Power" : "100.0 HP (74.6 kW) @ 2000 RPM", "Torque" : "185.0 Nm (16.0 kgf-m or 117.4 ft.lbs) @ 1800 RPM", "Top speed" : "180 km/h (111.8 mph)", "Fuel consumption" : "18.00 litres/100km (56.5 km/l or 120.0 mpg)", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Clutch" : "Multiplate", "Front suspension" : "Telescopic fork", "Rear suspension" : "Swing arm", "Front wheel travel" : "120 mm (4.7 inches)", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "12.00 litres (3.17 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 21 Sommer Diesel S10 discussion group or the general Sommer discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0f033"), "Model" : "Sommer Diesel S10", "Year" : "2019", "Category" : "Allround", "Rating" : "Price as new: Euro 9800. MSRP depend on country, taxes, accessories, etc.", "Rating" : "Do you know this bike?Click here to rate it. We miss 1 vote to show the rating.", "Displacement" : "516.0 cm (31.49 cubic inches)", "Engine type" : "Diesel", "Engine details" : "Hatz 100HP engine", "Power" : "100.0 HP (74.6 kW) @ 2000 RPM", "Torque" : "185.0 Nm (16.0 kgf-m or 117.4 ft.lbs) @ 1800 RPM", "Top speed" : "180 km/h (111.8 mph)", "Fuel consumption" : "18.00 litres/100km (56.5 km/l or 120.0 mpg)", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Clutch" : "Multiplate", "Front suspension" : "Telescopic fork", "Rear suspension" : "Swing arm", "Front wheel travel" : "120 mm (4.7 inches)", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "12.00 litres (3.17 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 21 Sommer Diesel S10 discussion group or the general Sommer discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0f035"), "Model" : "Sommer Hatz 1850 Diesel", "Year" : "2019", "Category" : "Allround", "Rating" : "Do you know this bike?Click here to rate it. We miss 1 vote to show the rating.", "Displacement" : "516.0 cm (31.49 cubic inches)", "Engine type" : "Diesel", "Engine details" : "Hatz 100HP engine", "Power" : "12.0 HP (8.8 kW) @ 3500 RPM", "Torque" : "27.5 NM (2.8 kgf-m or 20 ft.lbs) @ 2000 RPM", "Top speed" : "180 km/h (111.8 mph)", "Fuel consumption" : "20.00 litres/100km (43.7 km/l or 13.5 mpg)", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Clutch" : "Multiplate", "Front suspension" : "Telescopic fork", "Rear suspension" : "Swing arm", "Front wheel travel" : "120 mm (4.7 inches)", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "12.00 litres (3.17 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 18 Sommer Hatz 1850 Diesel discussion group or the general Sommer discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
| _id : ObjectId("627d4888e2cbbcd97dc0f035"), "Model" : "Sommer Hatz 1850 Diesel", "Year" : "2019", "Category" : "Allround", "Rating" : "Do you know this bike?Click here to rate it. We miss 1 vote to show the rating.", "Displacement" : "516.0 cm (31.49 cubic inches)", "Engine type" : "Diesel", "Engine details" : "Hatz 100HP engine", "Power" : "12.0 HP (8.8 kW) @ 3500 RPM", "Torque" : "27.5 NM (2.8 kgf-m or 20 ft.lbs) @ 2000 RPM", "Top speed" : "180 km/h (111.8 mph)", "Fuel consumption" : "20.00 litres/100km (43.7 km/l or 13.5 mpg)", "Gearbox" : "5-speed", "Transmission type, final drive" : "Chain", "Clutch" : "Multiplate", "Front suspension" : "Telescopic fork", "Rear suspension" : "Swing arm", "Front wheel travel" : "120 mm (4.7 inches)", "Rear wheel travel" : "120 mm (4.7 inches)", "Fuel capacity" : "12.00 litres (3.17 gallons)", "Color options" : "Black, silver, red, white, blue", "Starter" : "Electric", "Comments" : "Compare US insurance quotes from the nation's top providers.", "Parts finder" : "\r\n\r\nChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries. \r\n\r\nAlso check out our overview of motorcycle webshops at Biker.info.", "Ask questions" : "Join the 18 Sommer Hatz 1850 Diesel discussion group or the general Sommer discussion group.", "Related bikes" : "List related bikes for comparison of specs." }
```

### QUERY 4

```
> db.getCollection("temp").find({"Gearbox": "4-speed"})
```

## **QUERY 5**

```
> db.getCollection("temp").find({"Category": "Scooter"})
```

From the previous steps, we found key-value pairs to narrow down the result.mFor example, for key value pairs Category: Scooter, Gearbox: 4-speed, Engine Type: Diesel, Year: 2021 we found more than 3 documents. For the key value pair Model: "proEco Emma" we got 1 document.

## Problem 2.7

We used command Update for the first two queries. By using query Update only the first document of the total documents is updated.

For query 3, 4 and 5 we used the query UpdateMany which will update all the documents.

### QUERY 1

```
> db.getCollection("temp").update({Model: "proEco Emma"}, {$set: {Model: "Ducati panigale V4"}})
```

```
> db.getCollection("temp").update({model: "proEco Emma"}, {$set: {model: "Ducati panigale V4"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.getCollection("temp").find({Model: "proEco Emma"})
> db.getCollection("temp").find({Model: "Ducati panigale V4"})
[{"_id": ObjectId("627d4889e2cbbcc0d76c12be"), "Model": "Ducati panigale V4", "Year": "2013", "Category": "Scooter", "Rating": "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Engine type": "Electric", "Engine details": "Wheel hub motor", "Power": "2.7 HP (2.0 kW)", "Top speed": "45.6 km/h (28.0 mph)", "Cooling system": "Air", "Front brakes": "Single disc", "Rear brakes": "Single disc", "Seat": "Dual", "Weight incl. oil, gas, etc": "115.0 kg (319.7 pounds)", "Seat height": "788 mm (30.7 inches) If adjustable, lowest setting.", "Overall height": "1130 mm (44.5 inches)", "Overall length": "1935 mm (76.2 inches)", "Overall width": "718 mm (28.0 inches)", "Ground clearance": "108 mm (3.9 inches)", "Color options": "Red, blue, black, silver, beige, pink, yellow, olive, brown, Electrical", "Battery": "12V 10 Ah Silicon battery.", "Comments": "Range 70 km. Charge time 5-8 hours. More than 450 charge cycles. German brand.", "Insurance costs": "Compare US insurance quotes from the nation's top providers.", "Finance options": "Compare US motorcycle loan quotes from the nation's top providers.", "Parts finder": "\r\n\tChapparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n\tAlso check out our overview of motorcycle webshops at Bikez.info.", "Ask questions": "Join the 13 proEco Emma discussion group or the general proEco discussion group.", "Related bikes": "List related bikes for comparison of specs."}]
```

### QUERY 2

```
> db.getCollection("temp").update({Year: "2018"}, {$set: {Year: "2030"}})
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.getCollection("temp").update({Year: "2018"}, {$set: {Year: "2030"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> |
```

```
> db.getCollection("temp").update({Year: "2018"}, {$set: {Year: "2030"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.getCollection("temp").find({Year: "2030"})
[{"_id": ObjectId("627d4889e2cbbcc0d76f515d"), "Model": "AJP PR3 Enduro 240", "Year": "2030", "Category": "Enduro / offroad", "Rating": "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement": "233.0 ccm (14.22 cubic inches)", "Engine type": "Single cylinder, four stroke", "Power": "20.0 HP (14.6 kW) @ 8000 RPM", "Torque": "18.0 Nm (1.8 kgf-m or 13.3 ft.lbs) @ 7000 RPM", "Bore x stroke": "69.0 x 62.5 mm (2.7 x 2.5 inches)", "Valves per cylinder": "2", "Fuel system": "Carburettor, 30 mm", "Fuel control": "Overhead Cams (OHC)", "Cooling system": "Oil & air", "Gearbox": "5-speed", "Transmission type,final drive": "Chain", "Clutch": "Oil bath, multidisc", "Front suspension": "AJP Ø38mm, fully adjustable", "Front wheel travel": "240 mm (9.4 inches)", "Rear suspension": "Öhlé progressive system", "Rear wheel travel": "280 mm (11.0 inches)", "Front tyre": "70/100-19", "Rear tyre": "100/100-19", "Front brakes": "Single disc, 2-piston calipers", "Diameter": "220 mm (8.7 inches)", "Rear brakes": "Single disc", "Weight incl. oil, gas, etc": "196.0 kg (433.7 pounds)", "Seat height": "800 mm (31.1 inches) If adjustable, lowest setting.", "Ground clearance": "305 mm (12.0 inches)", "Wheelbase": "1320 mm (52.0 inches)", "Fuel capacity": "7.50 litres (1.98 gallons)", "Color options": "White/red/black", "Starter": "Electric kick", "Comments": "AJP is a Portuguese brand.", "Insurance costs": "Compare US insurance quotes from the nation's top providers.", "Finance options": "Compare US motorcycle loan quotes from the nation's top providers.", "Parts finder": "\r\n\tChapparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n\tAlso check out our overview of motorcycle webshops at Bikez.info.", "Dirt-bike parts": "Dirt-bike parts and gear available from Mega Motor Madness. Ships to most countries.", "Ask questions": "Join the 18 AJP PR3 Enduro 240 discussion group or the general AJP discussion group.", "Related bikes": "List related bikes for comparison of specs."}]
```

### QUERY 3

```
> db.getCollection("temp").updateMany({"Engine type": "Diesel"}, {$set: {"Engine type": "Petrol"}})
```

```
{ "acknowledged" : true, "matchedCount" : 27, "modifiedCount" : 27 }
```

```
> db.getCollection("temp").updateMany({"Engine type": "Diesel"}, {$set: {"Engine type": "Petrol"}})
{ "acknowledged" : true, "matchedCount" : 27, "modifiedCount" : 27 }
> |
```

```
> db.getCollection("temp").updateMany({"Engine type": "Diesel"}, {$set: {"Engine type": "Petrol"}})
uncaught exception: TypeError: db.getCollection(...).updateMany is not a function :
@(shell):1:1
> db.getCollection("temp").updateMany({"Engine type": "Diesel"}, {$set: {"Engine type": "Petrol"}})
{ "acknowledged" : true, "matchedCount" : 27, "modifiedCount" : 27 }
> db.getCollection("temp").find({"Engine type": "Petrol"})
{ "_id" : ObjectId("627d4888e2cbbcc0d76beefcf"), "Model" : "Neander Turbo Diesel 1400", "Year" : "2019", "Category" : "Custom / cruiser", "Rating" : "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "1400.0 ccm (85.43 cubic inches)", "Engine type" : "Petrol", "Power" : "112.0 HP (81.8 kW) @ 4200 RPM", "Torque" : "214.0 Nm (21.8 kgf-m or 157.8 ft.lbs) @ 2600 RPM", "0-100 km/h (0-62 mph)" : "4.00 seconds", "Bore x stroke" : "105.0 x 77.6 mm (4.1 x 3.1 inches)", "Valves per cylinder" : "4", "Fuel system" : "Turbo", "Cooling system" : "Oil & air", "Gearbox" : "6-speed", }
```

## QUERY 4

```
> db.getCollection("temp").updateMany({"Gearbox": "4-speed"}, {$set: {"Gearbox": "13-speed"}})
{ "acknowledged" : true, "matchedCount" : 2931, "modifiedCount" : 2931 }
```

```
> db.getCollection("temp").updateMany({"Gearbox": "4-speed"}, {$set: {"Gearbox": "13-speed"}})
{ "acknowledged" : true, "matchedCount" : 2931, "modifiedCount" : 2931 }
> |
```

```
> db.getCollection("temp").updateMany({"Gearbox": "4-speed"}, {$set: {"Gearbox": "13-speed"}})
{ "acknowledged" : true, "matchedCount" : 2931, "modifiedCount" : 2931 }
> db.getCollection("temp").find({"Gearbox": "13-speed"})
{ "_id" : ObjectId("627d4888e2cbbcc0d76be3e"), "Model" : "Bajaj CT 100", "Year" : "2021", "Category" : "Allround", "Rating" : "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "102.0 ccm (6.22 cubic inches)", "Engine type" : "Single cylinder, four-stroke", "Power" : "7.8 HP (5.7 kW) @ 7500 RPM", "Torque" : "8.3 Nm (0.9 kgf-m or 6.2 ft.lbs) @ 5500 RPM", "Top speed" : "90.0 km/h (55.9 mph)", "Bore x stroke" : "47.0 x 58.8 mm (1.9 x 2.3 inches)", "Valves per cylinder" : "2", "Fuel system" : "Injection", "Cooling system" : "Air", "Gearbox" : "13-speed", "Transmission type, final drive" : "Chain", "Driveline" : "Constant mesh", "Frame type" : "Tubular single down tube with lower cradle frame", "Front suspension" : "Hydraulic, Telescopic Type", "Front wheel travel" : "125 mm (4.9 inches)", "Rear suspension" : "Spring-in-Spring (SNS)", "Rear wheel travel" : "100 mm (3.9 inches)", "Front tyre" : "2.75-17", "Rear tyre" : "3.00-17", "Front brakes" : "Expanding brake (drum brake)", "Diameter" : "110 mm (4.3 inches)", "Rear brakes" : "Expanding brake (drum brake)", "Wheels" : "Wire spoke or alloy wheels", "Weight incl. oil, gas, etc" : "115.0 kg (253.5 pounds)", "Overall height" : "1072 mm (42.2 inches)", "Overall length" : "1945 mm (76.6 inches)", "Overall width" : "752 mm (29.6 inches)", "Ground clearance" : "170 mm (6.7 inches)", "Wheel
```

## QUERY 5

```
> db.getCollection("temp").updateMany({"Transmission type, final drive": "Chain"}, {$set: {"Transmission type, final drive": "Automate"}})
{ "acknowledged" : true, "matchedCount" : 21507, "modifiedCount" : 21507 }
```

```
> db.getCollection("temp").updateMany({"Transmission type, final drive": "Chain"}, {$set: {"Transmission type, final drive": "Automate"}})
{ "acknowledged" : true, "matchedCount" : 21507, "modifiedCount" : 21507 }
> |
```

```
> db.getCollection("temp").find({"Transmission type, final drive": "Automate"})
{ "_id" : ObjectId("627d4888e2cbbcc0d76be04"), "Model" : "AJP PR7", "Year" : "2022", "Category" : "Sport", "Rating" : "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement" : "600.0 ccm (36.61 cubic inches)", "Engine type" : "Single cylinder, four-stroke", "Torque" : "58.0 Nm (5.9 kgf-m or 42.8 ft.lbs)", "Bore x stroke" : "100.0 x 76.4 mm (3.9 x 3.0 inches)", "Fuel system" : "Injection, Mikuni", "Fuel control" : "Double Overhead Cams/Twin Cam (DOHC)", "Cooling system" : "Liquid", "Gearbox" : "5-speed", "Transmission type, final drive" : "Automate", "Frame type" : "Aluminum/steel", "Rake (fork angle)" : "26.5°", "Trail" : "300 mm (11.8 inches)", "Front suspension" : "ZF Sachs Ø48 mm close-circuit", "Rear suspension" : "ZF Sachs progs. system", "Rear wheel travel" : "280 mm (11.0 inches)", "Front tyre" : "90/90-21", "Rear tyre" : "140/80-18", "Front brakes" : "Single disc, Brembo 2-piston calipers", "Diameter" : "240 mm (9.4 inches)", "Rear brakes" : "Single disc, Brembo single-piston", "Seat height" : "920 mm (36.2 inches) If adjustable, lowest setting.", "Ground clearance" : "310 mm (12.2 inches)", "Wheelbase" : "1540 mm (60.6 inches)", "Fuel capacity" : "17.00 litres (4.49 gallons)
```

## Problem 2.8

### QUERY 1

```
> db.getCollection("temp").insert({Model: "Mercedez Benz", Year : "2099", Category: "Luxury", Price: "10000", Color : "Black"})
```

```

WriteResult({ "nInserted" : 1 })

> db.getCollection("temp").find({"Model": "Mercedez Benz"})

{ "_id" : ObjectId("627dc2a11c9c4328c1f54dfe"), "Model" : "Mercedez Benz", "Year" : "2099",
"Category" : "Luxury", "Price" : "10000", "Color" : "Black" }

> db.getCollection("temp").insert({Model: "Mercedez Benz", Year : "2099" , Category: "Luxury" , Price: "10000" , Color : "Black"})
WriteResult({ "nInserted" : 1 })
> db.getCollection("temp").find({"Model": "Mercedez Benz"})

{ "_id" : ObjectId("627dc2a11c9c4328c1f54dfe"), "Model" : "Mercedez Benz", "Year" : "2099", "Category" : "Luxury", "Price" : "10000", "Color" : "Black" }
>

```

## QUERY 2

```

> db.getCollection("temp").insert({Model: "Tesla", Year : "2011" , Category: "Eco friendly" , Price:
"100000" , Color : "White", "Fuel Capacity": "0 litres"})

WriteResult({ "nInserted" : 1 })

> db.getCollection("temp").insert({Model: "Tesla", Year : "2011" , Category: "Eco friendly" , Price: "100000" , Color : "White", "Fuel Capacity": "0 litres"})
WriteResult({ "nInserted" : 1 })
>

```

```

> db.getCollection("temp").find({{"Model": "Tesla"})
[ { "_id" : ObjectId("627dc62aca44516bf86e683b"), "Model" : "Tesla", "Year" : "2011", "Category" : "Eco friendly", "Price" : "100000", "Color" : "White", "Fuel Capacity" : "0 litres" }
, { "_id" : ObjectId("627dc65bca44516bf86e683c"), "Model" : "Tesla", "Year" : "2011", "Category" : "Eco friendly", "Price" : "100000", "Color" : "White", "Fuel Capacity" : "0 litres" }
]

```

## QUERY 3

```

> db.getCollection("temp").insert({Model: "KTM", Year : "2014" , Category: "Eco friendly" , Price:
"300000" , "Fuel Capacity": "20 litres"})

WriteResult({ "nInserted" : 1 })

> db.getCollection("temp").find({{"Model": "KTM"})
[ { "_id" : ObjectId("627dca4df042ac95a9929c39"), "Model" : "KTM", "Year" : "2014", "Category" : "Eco friendly", "Price" : "300000", "Fuel Capacity" : "20 litres" }
, { "_id" : ObjectId("627dca6af042ac95a9929c3a"), "Model" : "KTM", "Year" : "2014", "Category" : "Eco friendly", "Price" : "300000", "Fuel Capacity" : "20 litres" }
]

```

```
> db.getCollection("temp").find({"Model": "KTM"})
```

```

> db.getCollection("temp").find({{"Model": "KTM"})
[ { "_id" : ObjectId("627dca4df042ac95a9929c39"), "Model" : "KTM", "Year" : "2014", "Category" : "Eco friendly", "Price" : "300000", "Fuel Capacity" : "20 litres" }
, { "_id" : ObjectId("627dca6af042ac95a9929c3a"), "Model" : "KTM", "Year" : "2014", "Category" : "Eco friendly", "Price" : "300000", "Fuel Capacity" : "20 litres" }
]

```

## QUERY 4

```

> db.getCollection("temp").insert({Model: "BMW S1000RR", Year : "2017" , Category: "Eco
friendly" , Price: "300000" , "Fuel Capacity": "30 litres", Gearbox: "4-speed"})


```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.getCollection("temp").insert({Model: "BMW S1000RR", Year : "2017" , Category: "Eco friendly" , Price: "300000" , Fuel Capacity": "30 litres", Gearbox: "4-speed"})
WriteResult({ "nInserted" : 1 })
```

```
> db.getCollection("temp").find({"Model": "BMW S1000RR"})
```

```
> db.getCollection("temp").find({"Model": "BMW S1000RR"})
{ "_id" : ObjectId("627dc18f042ac95a9929c3b"), "Model" : "BMW S1000RR", "Year" : "2017", "Category" : "Eco friendly", "Price" : "300000", "Fuel Capacity" : "30 litres", "Gearbox" : "4-speed" }
> |
```

## QUERY 5

```
> db.getCollection("temp").insert({Model: "Royal Enfield", Year : "2018", Price: "500000" , "Fuel Capacity": "40 litres", Gearbox: "16-speed"})
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.getCollection("temp").insert({Model: "Royal Enfield", Year : "2018", Price: "500000" , "Fuel Capacity": "40 litres"
, Gearbox: "16-speed"})
WriteResult({ "nInserted" : 1 })
```

```
> db.getCollection("temp").find({"Model": "Royal Enfield"})
```

```
> db.getCollection("temp").find({"Model": "Royal Enfield"})
{ "_id" : ObjectId("627dcbaef042ac95a9929c3c"), "Model" : "Royal Enfield", "Year" : "2018", "Price" : "500000", "Fuel Capacity" : "40 litres", "Gearbox" : "16-speed" }
> |
```

## Problem 2.9

### QUERY 1

```
> db.getCollection("temp").deleteMany({Model: "Aprilia Dorsoduro 900"})
```

```
{ "acknowledged" : true, "deletedCount" : 9 }
```

```
> db.getCollection("temp").deleteMany({Model: "Aprilia Dorsoduro 900"})
{ "acknowledged" : true, "deletedCount" : 9 }
> |
```

### QUERY 2

```
> db.getCollection("temp").deleteMany({Year: "2013"})
```

```
{ "acknowledged" : true, "deletedCount" : 1407 }
```

```
> db.getCollection("temp").deleteMany({Year: "2013"})
{ "acknowledged" : true, "deletedCount" : 1407 }
> |
```

### QUERY 3

```
> db.getCollection("temp").deleteMany({Category: "Super motard"})
{ "acknowledged" : true, "deletedCount" : 1548 }

> db.getCollection("temp").deleteMany({Category: "Super motard"})
{ "acknowledged" : true, "deletedCount" : 1548 }
> |
```

### QUERY 4

```
> db.getCollection("temp").deleteMany({"Cooling system": "Liquid"})
{ "acknowledged" : true, "deletedCount" : 13605 }

> db.getCollection("temp").deleteMany({"Cooling system": "Liquid"})
{ "acknowledged" : true, "deletedCount" : 13605 }
> |
```

### QUERY 5

```
> db.getCollection("temp").deleteMany({Compression: "11.0:1"})
{ "acknowledged" : true, "deletedCount" : 263 }

> db.getCollection("temp").deleteMany({Compression: "11.0:1"})
{ "acknowledged" : true, "deletedCount" : 263 }
> ^C
```