

**BUAN 6320.502**

**Project 2**

**Submitted by**

Raghothama Rao Pranesha – rxr220019

Afsaruddin Mohammed - axm210415

Koti Reddy Gangasani - kxg220019

Sri Mahalakshmi Pendyala – sxp220096

Vikrant Sagar Remoddula - vxr220005

# Part 1

1.1. List all the current indexes in your database and the columns they are associated with along with the index type

Answer:

```
5 •   SELECT DISTINCT
6       TABLE_NAME,
7       INDEX_NAME,
8       INDEX_TYPE
9   FROM INFORMATION_SCHEMA.STATISTICS
10  WHERE TABLE_SCHEMA = 'mydb';
```

TABLE_NAME	INDEX_NAME	INDEX_TYPE
category	PRIMARY	BTREE
county	PRIMARY	BTREE
item	fk_Item_Category1_idx	BTREE
item	fk_Item_Vendor1_idx	BTREE
item	PRIMARY	BTREE
sales	PRIMARY	BTREE
store	fk_Store_County1_idx	BTREE
store	PRIMARY	BTREE
supply	PRIMARY	BTREE
supply	vendor_id_idx	BTREE
vendor	PRIMARY	BTREE

There are 7 tables in our database and the indexes for each table is shown below: (name the table for each snippet)

Table 1 vendor:

```
8 •   show index
9     from vendor
10    from mydb;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
vendor	0	PRIMARY	1	Vendor_id	A	345	NULL	NULL	NULL	BTREE			YES	NULL

Table 2 supply:

```
8 •   show index
9     from supply
10    from mydb;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
supply	0	PRIMARY	1	Store_id	A	2405	NULL	NULL	NULL	BTREE			YES	NULL
supply	0	PRIMARY	2	Vendor_id	A	94970	NULL	NULL	NULL	BTREE			YES	NULL
supply	1	vendor_id_idx	1	Vendor_id	A	319	NULL	NULL	NULL	BTREE			YES	NULL

Table 3 store:

Result Grid   Filter Rows: Export: Wrap Cell Content: <input type="checkbox"/>														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
store	0	PRIMARY	1	Store_id	A	2447	NULL	NULL	BTREE		YES	NULL		
store	1	fk_Store_County1_idx	1	County_County_id	A	99	NULL	NULL	BTREE		YES	NULL		

Table 4 sales:

Result Grid   Filter Rows: Export: Wrap Cell Content: <input type="checkbox"/>														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
sales	0	PRIMARY	1	Invoice_id	A	20798139	NULL	NULL	BTREE		YES	NULL		

Table 5 item:

Result Grid   Filter Rows: Export: Wrap Cell Content: <input type="checkbox"/>														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
item	0	PRIMARY	1	Item_id	A	95	NULL	NULL	BTREE		YES	NULL		
item	1	fk_Item_Category1_idx	1	Category_Category_id	A	11	NULL	NULL	BTREE		YES	NULL		
item	1	fk_Item_Vendor1_idx	1	Vendor_Vendor_id	A	14	NULL	NULL	BTREE		YES	NULL		

Table 6 county:

Result Grid   Filter Rows: Export: Wrap Cell Content: <input type="checkbox"/>														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
county	0	PRIMARY	1	County_id	A	99	NULL	NULL	BTREE		YES	NULL		

Table 7 category:

Result Grid   Filter Rows: Export: Wrap Cell Content: <input type="checkbox"/>														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
category	0	PRIMARY	1	Category_id	A	109	NULL	NULL	BTREE		YES	NULL		

1.2. Explain what is in common between these columns (why these columns are indexed automatically by the database management system)

**Answer:** When a database table is accessed, data can be found fast by using indexes rather than by manually searching through all the rows. Utilizing one or more fields from a database table, indexes can be formed, laying the groundwork for both quick random lookups and effective access to ordered items.

All these columns have primary keys as common, and each table is given a primary key when the table is created. When a primary key is given to a table the primary key column is automatically indexed by the

database management system. Since the primary key, index, etc. are kept in B-trees, primary key is implicitly indexed in MySQL.

1.3. Make a copy of your database and delete all the indexes there (you might need to delete foreign keys before you can delete some of the indexes) – now you have two databases: database A with indexes and database B without any indexes.

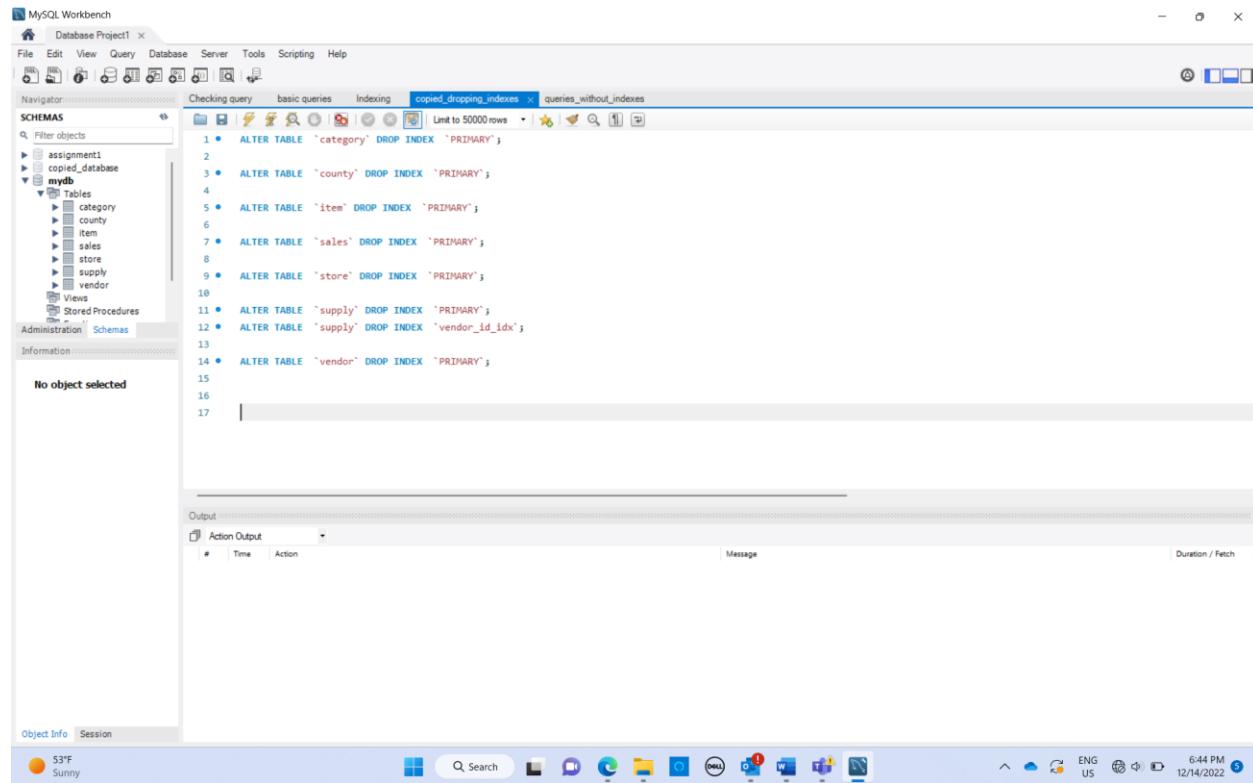
**Answer:** Created a database called copied\_database. Exported everything from the original database as a self-contained file(.sql) using:

*Server->Data Export->Select database to export-> Export to self-contained file*

Further, imported that file into the copied database using:

*Server->Data Import->Import from a self-contained file-> Select Default Target Schema*

Further, all the foreign keys (from tables: supply, store and item) are deleted so that we can delete all the indexes.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, which includes 'assignment1', 'copied\_database', and 'mydb'. Under 'mydb', there are several tables: category, county, item, sales, store, supply, and vendor. The central area contains a query editor with the following SQL code:

```
1 • ALTER TABLE `category` DROP INDEX `PRIMARY`;
2
3 • ALTER TABLE `county` DROP INDEX `PRIMARY`;
4
5 • ALTER TABLE `item` DROP INDEX `PRIMARY`;
6
7 • ALTER TABLE `sales` DROP INDEX `PRIMARY`;
8
9 • ALTER TABLE `store` DROP INDEX `PRIMARY`;
10
11 • ALTER TABLE `supply` DROP INDEX `PRIMARY`;
12 • ALTER TABLE `supply` DROP INDEX `vendor_Id_idx`;
13
14 • ALTER TABLE `vendor` DROP INDEX `PRIMARY`;
15
16
17
```

The bottom status bar shows the weather as 53°F Sunny, the date and time as 12/14/2022 6:44 PM, and various system icons.

1.4. Write at least 5 queries (with JOINs between your tables)

**Answer:** Based on logical connections between the tables, the SQL clause JOIN is used to query and access data from various tables. In other words, JOINS specify how SQL Server need to choose entries from a different table using information from a different source.

The following queries are done in the **copied\_database** where all the **indexes are dropped**:

- Query 1

```

1 -- Top 10 Dates with maximum amount of sales in a single transaction
2 • select s.date, s.amount, s.Store_Id from
3 sales s inner join store st on st.Store_Id=s.Store_Id
4 order by amount desc;

```

date	amount	Store_Id
2018-10-02	279557	2663
2012-12-03	254100	2633
2012-06-04	254100	2633
2021-07-05	250932	2633
2021-04-19	250932	2633
.....	.....	.....

Result 10 x

Action Output

#	Time	Action	Message	Duration / Fetch
50	13:32:42	show index from supply from copied_database	0 row(s) returned	0.000 sec / 0.000 sec
51	13:35:27	select * from category LIMIT 0, 50000	100 row(s) returned	0.000 sec / 0.000 sec
52	13:40:19	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	69.485 sec / 0.109 sec
53	13:42:10	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	63.359 sec / 0.157 sec
54	13:43:26	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	63.430 sec / 0.172 sec
55	13:44:33	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	71.594 sec / 0.125 sec
56	13:47:00	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	74.891 sec / 0.094 sec
57	13:48:21	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	63.032 sec / 0.156 sec
58	13:49:46	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	70.219 sec / 0.125 sec
59	13:53:13	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	61.031 sec / 0.140 sec
60	13:55:22	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	70.968 sec / 0.235 sec
61	13:56:57	select s.date, s.amount, s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id order by amount desc;	50000 row(s) returned	68.265 sec / 0.188 sec

Object Info Session

Query Completed

Cloudy 63°F 12/13/2022 1:58 PM ENG US

Average timing (Duration/Fetch) of this query run 10 times without index – 67.6282 sec/0.1501 sec

- Query 2

```

5
6 -- Top 10 maximum amount of sales from a particular store on a particular date
7 • select s.date, sum(s.amount), s.Store_Id from
8 sales s inner join store st on st.Store_Id=s.Store_Id
9 group by Date
10 order by amount desc;

```

date	sum(s.amount)	Store_Id
2015-11-09	152264.9008488655	3814
2012-09-12	1192733.7340180874	3420
2012-06-11	1362906.9898854944	4677
2020-11-05	1245238.0801087618	2633
2012-03-14	897131.7493860722	3420
2013-03-12	120090.6554756165	3944
2014-09-16	1003678.1096148491	3420
2014-09-17	1375136.2081427574	3447
.....	.....	.....

Result 24 x

Action Output

#	Time	Action	Message	Duration / Fetch
68	14:16:24	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	73.593 sec / 0.000 sec
1	14:16:31	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	61.141 sec / 0.000 sec
2	14:18:59	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	70.391 sec / 0.015 sec
3	14:28:18	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	52.285 sec / 0.000 sec
4	14:35:23	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	54.016 sec / 0.000 sec
5	14:40:08	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	50.406 sec / 0.016 sec
6	14:45:47	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	48.076 sec / 0.016 sec
7	14:47:37	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	55.344 sec / 0.000 sec
8	14:50:39	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	55.766 sec / 0.000 sec
9	14:53:37	select s.date, sum(s.amount), s.Store_Id from sales s inner join store st on st.Store_Id=s.Store_Id group by Date	2418 row(s) returned	63.156 sec / 0.000 sec

Object Info Session

Average timing (Duration/Fetch) of this query run 10 times without index – 58.8156 sec/0.0047 sec

### • Query 3

```

11
12 -- Top cities with maximum amount of sales
13 • select st.city as City, sum(s.amount) as Revenue
14 from sales s join store st on st.Store_id=s.Store_id
15 group by st.city
16 order by s.amount desc;

```

City	Revenue
WEST UNION	2575987.670363188
EARLING	183144.1995229258
LECLAIRE	3792136.92171546
DENISON	11096468.74485302
FONDA	178866.669978714
CALDWELL LAKE	33925.260709479
DOON	186220.51658107
ROCK VALLEY	1846697.1524736881
WILLOW RANCH	877144.1177476742

Result 34 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:57:57	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	252.344 sec / 0.000 sec
2	15:02:23	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	293.265 sec / 0.000 sec
3	15:07:21	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	207.812 sec / 0.000 sec
4	15:14:04	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	216.125 sec / 0.000 sec
5	15:20:39	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	205.594 sec / 0.000 sec
6	15:24:40	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	280.407 sec / 0.000 sec
7	15:43:29	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	291.921 sec / 0.000 sec
8	15:48:47	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	277.578 sec / 0.000 sec
9	15:58:13	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	285.453 sec / 0.000 sec
10	16:12:07	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	295.437 sec / 0.000 sec

Object Info Session

Query Completed

63°F Mostly cloudy

ENG US 417 PM 12/13/2022

Average timing(Duration/Fetch) of this query run 10 times without index – 260.5936 sec/0.000 sec

### • Query 4

```

18 -- Top 10 category with maximum amount of sales
19 • select c.Category_name as Category, sum(s.amount) as Revenue
20 from sales s
21 join item i on s.Item_id=i.Item_id
22 join category c on i.Category_Category_id=c.Category_id
23 group by c.Category_name
24 order by s.amount desc
25 limit 10;

```

Category	Revenue
AMERICAN SLOE GINS	4126432.993275404
IMPORTED VODKA	56903490.804596305
VODKA 80 PROOF	179868950.3359518
IMPORTED VODKA - MSC	16679793.7840566
IMPORTED DRY GIN	50037263.68372011
BLENDDED WHISKIES	65574554.7193277
TENNESSEE WHISKIES	90433422.87413204
STRAIGHT BOURBON WHISKIES	433362044.187109
FLAVORED GINS	104724867.15592885
AMERICAN DRY GINS	965819649.0091205

Result 44 x

Action Output

#	Time	Action	Message	Duration / Fetch
11	16:24:08	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	98.875 sec / 0.000 sec
12	16:26:01	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.281 sec / 0.000 sec
13	16:28:48	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.326 sec / 0.000 sec
14	16:30:33	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.625 sec / 0.000 sec
15	16:35:32	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.015 sec / 0.000 sec
16	16:56:50	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	101.438 sec / 0.000 sec
17	17:01:24	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	99.750 sec / 0.000 sec
18	17:08:46	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.093 sec / 0.000 sec
19	17:16:56	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	100.063 sec / 0.000 sec
20	17:23:14	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	66.266 sec / 0.000 sec

Object Info Session

Query Completed

60°F Mostly sunny

ENG US 5:25 PM 12/13/2022

Average timing(Duration/Fetch) of this query run 10 times without index – 96.7734 sec/0.000 sec

- **Query 5**

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** world, Views, Stored Procedures, Functions, com\_database (Tables: category, county, item, sales, store, supply, vendor).
- Current Database:** com\_database.
- Query Editor:** Contains the following SQL code:

```
23 -- group by c.Category_name
24 -- order by s.amount desc
25 -- limit 10;
26
27
28 * select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v
      where v.vendor_id = i.Vendor_Vendor_id and v.Vendor_id = 259
29
30
```
- Result Grid:** Displays the results of the query:

Vendor_id	Item_id	Item_description
259	84	Evan Williams Str Bourbon
259	52	Burnett's Gin London Dry
259	2	Evan Williams Str Bourbon
- Action Output:** Shows the execution history of the query:

Action	Time	Message	Duration / Fetch
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	21 17:27:45	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.031 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	22 17:27:50	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.016 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	23 17:27:52	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	24 17:27:53	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	25 17:27:54	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	26 17:27:56	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.015 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	27 17:27:57	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.015 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	28 17:27:58	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	29 17:27:59	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v	30 17:28:01	where v.vendor_id = i.Vendor.... 3 row(s) returned	0.000 sec / 0.000 sec
- Session Info:** Shows the session status and environment information (60°F, Mostly sunny).

Average timing(Duration/Fetch) of this query run 10 times without index – 0.0061 sec/0.016 sec

1.5. Execute and time these queries on both databases and report your findings (repeat timing for each query at least 10 times and average the times)

**Answer:** Now, the same queries that are mentioned above are queried in the original database which has indexes to time the queries:

- **Query 1**

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Checking query basic queries Indexing copied\_dropping\_indexes queries\_without\_indexes

```

11 • Select distinct(Vendor_name) from vendor
12
13
14 -- Queries for basic analysis of the dataset;
15
16 -- Top 10 Dates with maximum amount of sales in a single transaction
17 • select s.date, s.amount, s.Store_id from
18 sales s inner join store st on st.Store_id=s.Store_id
19 order by amount desc;
20
21
22 -- Top 10 maximum amount of sales from a particular store on a particular date

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Read Only

Table: store

Columns:

- Store\_id** int PK
- Location varchar
- Store\_name varchar
- Street\_address varchar
- City varchar
- Zip\_code varchar
- County\_County\_id int

Result 10 x

date	amount	Store_id
2018-10-02	279557	2663
2012-12-03	254100	2633
2012-06-04	254100	2633
2020-10-08	250932	2633
2021-07-05	250932	2633

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:38:01	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	70,546 sec / 0.188 sec
2	17:39:29	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	73,859 sec / 0.297 sec
3	17:40:47	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	70,515 sec / 0.219 sec
4	17:42:31	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	66,703 sec / 0.235 sec
5	17:43:44	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	67,453 sec / 0.250 sec
6	17:44:56	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	65,063 sec / 0.219 sec
7	17:46:07	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	65,031 sec / 0.234 sec
8	17:47:15	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	61,953 sec / 0.219 sec
9	17:48:21	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	64,468 sec / 0.235 sec
10	17:49:33	select a.date, a.amount, a.Store_id from sales a inner join store at on at.Store_id=a.Store_id order by amount des...	50000 row(s) returned	65,375 sec / 0.219 sec

Object Info Session

Query Completed

60°F Mostly sunny

5:50 PM 12/13/2022

Average timing (Duration/Fetch) of this query run 10 times with index – 67.0966 sec/ 0.2315 sec

- Query 2:

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Checking query basic queries Indexing copied\_dropping\_indexes queries\_without\_indexes

```

17 • select s.date, s.amount, s.Store_id from
18 sales s inner join store st on st.Store_id=s.Store_id
19 order by amount desc;
20
21
22 -- Top 10 maximum amount of sales from a particular store on a particular date
23 • select s.date, sum(s.amount), s.Store_id from
24 sales s inner join store st on st.Store_id=s.Store_id
25 group by Date
26 order by amount desc;
27
28 -- Top cities with maximum amount of sales

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Read Only

Table: store

Columns:

- Store\_id** int PK
- Location varchar
- Store\_name varchar
- Street\_address varchar
- City varchar
- Zip\_code varchar
- County\_County\_id int

Result 20 x

date	sum(amount)	Store_id
2015-11-09	1522644.9008488655	3814
2012-09-12	1192733.7340180874	3420
2012-06-11	1362906.9898854944	4677
2020-11-05	1245258.0891087618	2633
2012-03-14	897131.7493860722	3420

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:51:56	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	27,453 sec / 0.000 sec
2	17:52:27	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	26,391 sec / 0.000 sec
3	17:52:56	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	26,500 sec / 0.000 sec
4	17:53:30	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	26,797 sec / 0.000 sec
5	17:53:59	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	26,578 sec / 0.000 sec
6	17:54:27	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	27,781 sec / 0.000 sec
7	17:54:56	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	32,594 sec / 0.000 sec
8	17:55:38	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	35,203 sec / 0.000 sec
9	17:56:16	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	27,844 sec / 0.000 sec
10	17:56:47	select a.date, sum(a.amount), a.Store_id from sales a inner join store at on at.Store_id=a.Store_id group by Date ...	2418 row(s) returned	38,593 sec / 0.016 sec

Object Info Session

Query Completed

59°F Mostly cloudy

6:09 PM 12/13/2022

Average timing (Duration/Fetch) of this query run 10 times with index – 29.5734 sec/ 0.016 sec

- **Query 3:**

MySQL Workbench - Database Project1 - MySQL Model (DB-Project1.mwb) - EER Diagram

**Checking query:** basic\_queries

```

25 group by Date
26 order by amount desc;
27
28 -- Top cities with maximum amount of sales
29 • select st.City as City, sum(s.amount) as Revenue
30 from sales s join store st on st.Store_id=s.Store_id
31 group by st.City
32 order by s.amount desc;
33
34 -- Top 10 category with maximum amount of sales
35 • select c.Category_name as Category, sum(s.amount) as Revenue

```

**Result Grid:**

City	Revenue
WEST UNION	2575987.670363188
EARLING	183144.199522958
LECLAIRE	3792136.921715498
DENISON	11096468.74485302
FONDA	178866.069978714

**Action Output:**

#	Time	Action	Message	Duration / Fetch
1	18:14:19	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	190.375 sec / 0.000 sec
2	18:30:34	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	189.953 sec / 0.015 sec
3	18:33:52	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	186.671 sec / 0.000 sec
4	19:34:12	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	196.047 sec / 0.000 sec
5	19:41:36	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	192.593 sec / 0.000 sec
6	19:44:54	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	202.343 sec / 0.000 sec
7	19:52:30	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	212.593 sec / 0.000 sec
8	19:56:09	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	205.125 sec / 0.000 sec
9	20:01:49	select st.City as City, sum(s.amount) as Revenue from sales s join store st on st.Store_id=s.Store_id group by st...	450 row(s) returned	196.516 sec / 0.000 sec

Average timing (Duration/Fetch) of this query run 10 times with index – 197.8434 sec / 0.015 sec

- **Query 4:**

MySQL Workbench - Database Project1 - MySQL Model (DB-Project1.mwb) - EER Diagram

**Checking query:** basic\_queries\*

```

33
34 -- Top 10 category with maximum amount of sales
35 • select c.Category_name as Category, sum(s.amount) as Revenue
36 from sales s
37 join item i on s.Item_id=i.Item_id
38 join category c on i.Category_Category_id=c.Category_id
39 group by c.Category_name
40 order by s.amount desc
41 limit 10;
42
43

```

**Result Grid:**

Category	Revenue
AMERICAN SLOE GINS	4126432.99327944
IMPORTED VODKA	56903490.804996305
VODKA 80 PROOF	179868930.3395918
IMPORTED VODKA - MISC	166750793.7840566
IMPORTED DRY GINS	50037263.68372011

**Action Output:**

#	Time	Action	Message	Duration / Fetch
1	20:06:08	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	51.250 sec / 0.000 sec
2	20:08:34	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	52.515 sec / 0.000 sec
3	20:09:33	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	52.344 sec / 0.000 sec
4	20:10:28	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	52.250 sec / 0.000 sec
5	20:11:23	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	51.545 sec / 0.000 sec
6	20:12:47	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	51.875 sec / 0.000 sec
7	20:13:42	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	53.281 sec / 0.000 sec
8	20:14:39	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	52.828 sec / 0.000 sec
9	20:15:36	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	49.063 sec / 0.000 sec
10	20:16:30	select c.Category_name as Category, sum(s.amount) as Revenue from sales s join item i on s.Item_id=i.Item_id...	10 row(s) returned	57.140 sec / 0.000 sec

Average timing (Duration/Fetch) of this query run 10 times with index – 52.4092 sec/ 0.000 sec

- **Query 5:**

The screenshot shows the MySQL Workbench interface. In the central SQL editor, there are two queries. The first query selects the category name and sum of amount from the sales table, grouped by category. The second query selects vendor id, item id, and item description from the item and vendor tables where vendor id is 259. The Result Grid pane shows the output of the first query, which contains three rows of data from the store table.

Vendor_id	Item_id	Item_description
259	2	Evan Williams Str Bourbon
259	52	Burnett's Gin London Dry
259	84	Evan Williams Yr Str Bourbon

The Action Output pane shows 10 rows of activity log, all completed in 0.000 sec / 0.000 sec.

Average timing (Duration/Fetch) of this query run 10 times with index – 0.000 sec/ 0.000 sec

1.6. Select some columns from database A (columns that are not already indexed) and create index on them

**Answer:** We have selected and created index for the columns shown below:

The screenshot shows the MySQL Workbench interface. In the central SQL editor, a CREATE INDEX statement is run on the sales table for columns Store\_id, Item\_id, Amount, and Date. The output pane shows the execution of the statement and the creation of the index.

Action	Time	Message	Duration / Fetch
11 21:04:15	select	Amount from sales LIMIT 0, 50000	0.032 sec / 0.140 sec
12 21:10:31	CREATE INDEX	custom_index_sales ON sales (Store_id, Item_id, Amount, Date)	135.204 sec

1.7. Write a query for each column – the query should include the column in the WHERE clause in a condition

**Answer:** Each column is queried in both the databases using the WHERE clause condition to time them.

The snapshots below show the queries done in the original database **mydb** using custom index **custom\_index\_sales**.

- **Query 1:**

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: mydb Tables: sales

Checking query: basic queries Indexing copied\_dropping\_indexes queries\_without\_indexes

55  
56 • CREATE INDEX custom\_index\_sales  
57 ON sales (Store\_id, Item\_id, Amount, Date);  
58  
59 • select Store\_id from sales  
60 where Store\_id>3800;

Result Grid: Filter Rows: Export: Wrap Cell Content: Fetch Rows: Read Only

sales 53 x

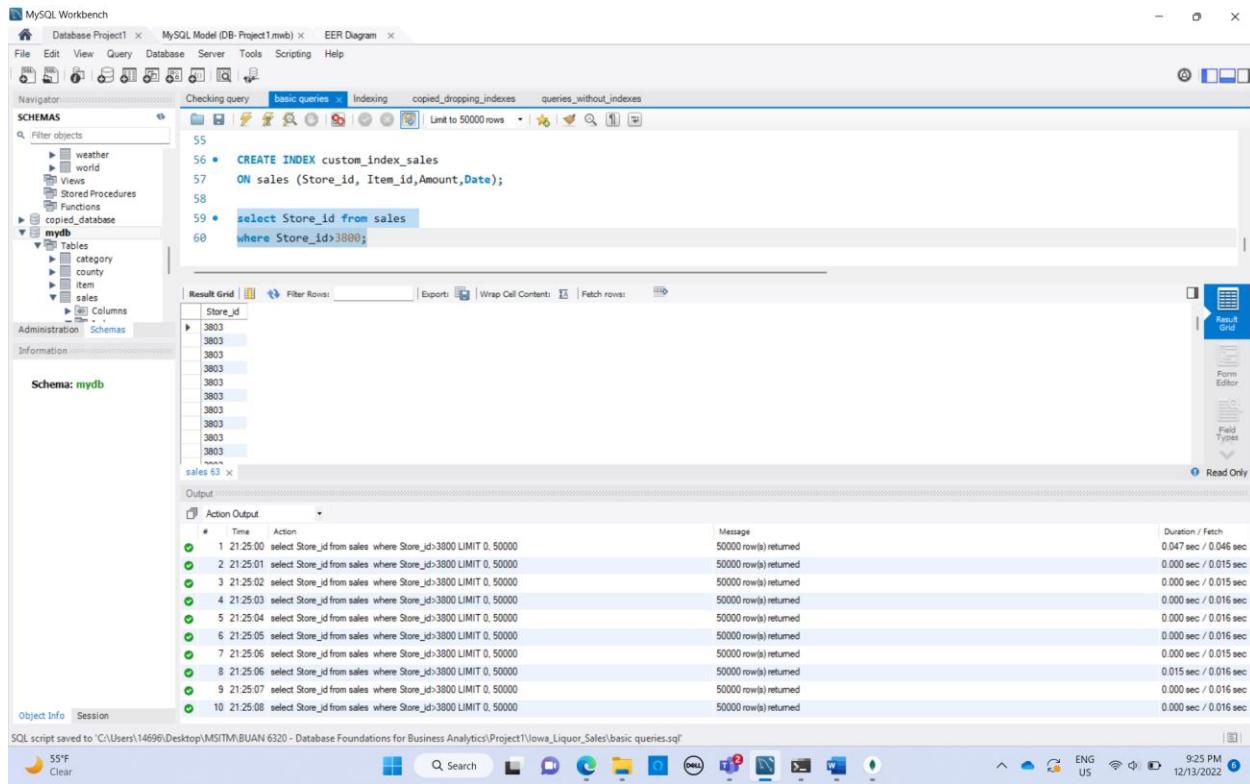
Action Output

#	Time	Action	Message	Duration / Fetch
1	21:25:00	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.047 sec / 0.046 sec
2	21:25:01	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
3	21:25:02	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
4	21:25:03	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
5	21:25:04	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
6	21:25:05	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
7	21:25:06	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
8	21:25:06	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.015 sec / 0.016 sec
9	21:25:07	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
10	21:25:08	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec

Object Info Session

SQL script saved to 'C:\Users\14696\Desktop\MSITM\BUAN 6320 - Database Foundations for Business Analytics\Project1\Iowa\_Liquor\_Sales\basic\_queries.sql'

55°F Clear 9:25 PM 12/13/2022



Average timing(Duration/Fetch) of this query run 10 times with index – 0.0062sec/ 0.0187 sec

- Query 2:

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: mydb Tables: sales

Checking query: basic queries Indexing copied\_dropping\_indexes queries\_without\_indexes

57 ON sales (Store\_id, Item\_id, Amount, Date);  
58  
59 • select Store\_id from sales  
60 where Store\_id>3800;  
61  
62 • select Item\_id from sales  
63 where Item\_id=7;

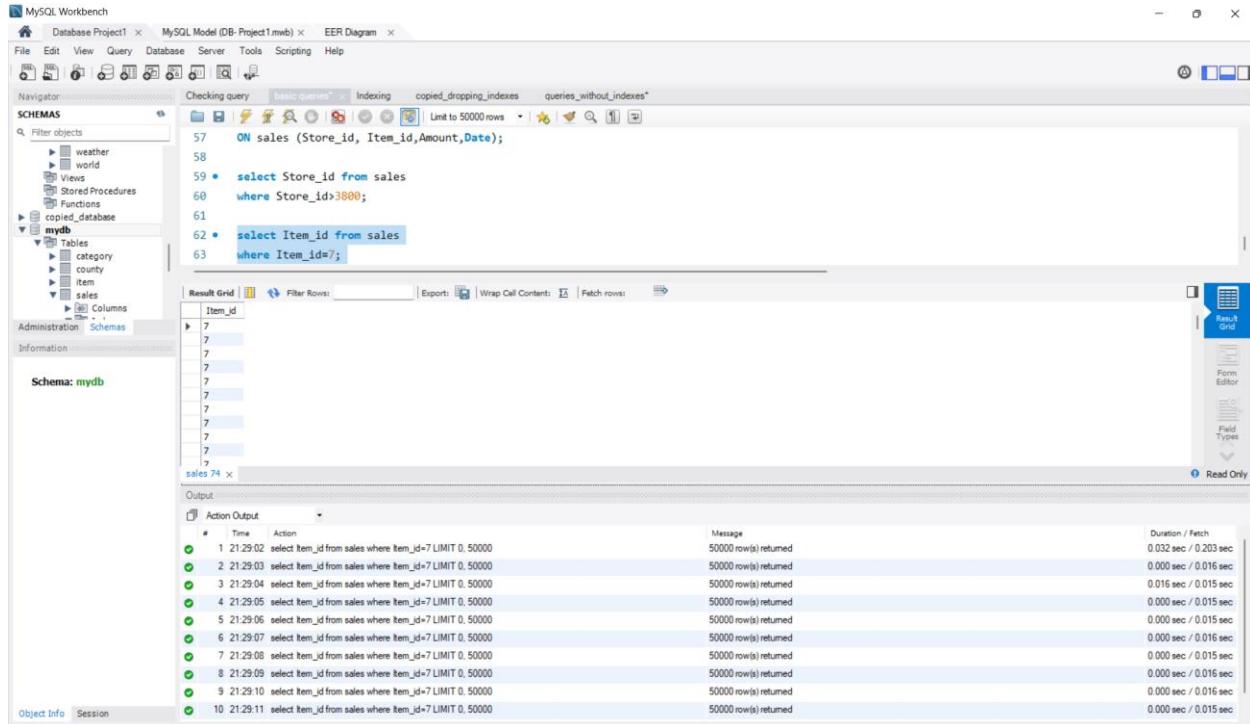
Result Grid: Filter Rows: Export: Wrap Cell Content: Fetch Rows: Read Only

sales 74 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:29:02	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.032 sec / 0.203 sec
2	21:29:03	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
3	21:29:04	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.015 sec
4	21:29:05	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
5	21:29:06	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
6	21:29:07	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
7	21:29:08	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
8	21:29:09	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
9	21:29:10	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.016 sec
10	21:29:11	select Item_id from sales where Item_id=7 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec

Object Info Session



Average timing (Duration/Fetch) of this query run 10 times with index – 0.0048sec/ 0.0342 sec

- Query 3:

```

60    where Store_id>300;
61
62 •   select Item_id from sales
63     where Item_id=7;
64
65 •   select Amount from sales
66     where Amount>=300;

```

Amount	Count
300	50000
300.06	50000
300.06	50000
300.48	50000
301.32	50000
301.32	50000
301.32	50000
301.32	50000
301.32	50000

Output:

#	Time	Action	Message	Duration / Fetch
11	21:35:06	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.172 sec
12	21:35:08	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.031 sec
13	21:35:09	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.031 sec
14	21:35:10	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.032 sec
15	21:35:11	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.031 sec
16	21:35:12	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.031 sec
17	21:35:13	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.031 sec
18	21:35:14	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.015 sec / 0.016 sec
19	21:35:15	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.015 sec
20	21:35:16	select Amount from sales where Amount>=300 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.031 sec

Average timing(Duration/Fetch) of this query run 10 times with index – 0.031sec/ 0.0421 sec

#### • Query 4:

```

62 •   select Item_id from sales
63     where Item_id=7;
64
65 •   select Amount from sales
66     where Amount>=300;
67
68 •   select Date from sales
69     where Date='2015-01-01';

```

Date	Count
2018-01-03	50000
2018-05-22	50000
2017-11-02	50000
2015-01-22	50000
2015-02-26	50000
2015-04-16	50000
2015-04-23	50000
2015-06-11	50000
2015-10-22	50000
2016-06-09	50000
2015-02-12	50000
2015-02-26	50000

Output:

#	Time	Action	Message	Duration / Fetch
1	20:18:11	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.078 sec
2	20:18:14	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.047 sec
3	20:18:15	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.047 sec
4	20:18:17	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
5	20:18:18	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
6	20:18:19	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
7	20:18:21	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
8	20:18:22	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.031 sec
9	20:18:23	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
10	20:18:25	select Date from sales where Date='2015-01-01' LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec

Average timing(Duration/Fetch) of this query run 10 times with index – 0.01567 sec/0.0485 sec

1.8. Execute and time these queries on both databases and report your findings (repeat timing for each query at least 10 times and average the times)

**Answer:** The same queries on each column using WHERE clause as a condition and their average timings done in the **copied\_database** which **has no index** are shown below

Without index:

- **Query 1:**

The screenshot shows the SSMS interface with the following details:

- Navigator:** Shows the schema **copied\_database** with tables **category**, **county**, **item**, and **sales**.
- Query Editor:** Displays the following T-SQL code:

```

22    -- join category c on i.Category_Catgeory_Id=c.Category_Id
23    -- group by c.Category_name
24    -- order by s.amount desc
25    -- limit 10;
26
27 •   select Store_id from sales
28   where Store_id>3800;
29

```
- Result Grid:** Shows the results of the query, displaying the value **3869** repeated 85 times.
- Action Output:** A table showing the execution history of the query over 10 runs, including time, action, message, and duration/fetch time.

#	Time	Action	Message	Duration / Fetch
15	21:21:23	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.015 sec / 0.360 sec
16	21:22:00	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.110 sec
17	21:22:02	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.093 sec
18	21:22:03	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.047 sec
19	21:22:04	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.093 sec
20	21:22:05	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
21	21:22:06	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
22	21:22:07	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.078 sec
23	21:22:08	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.047 sec
24	21:22:09	select Store_id from sales where Store_id>3800 LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.062 sec

Average timing(Duration/Fetch) of this query run 10 times without index – 0.0047 sec/ 0.1078 sec

- **Query 2:**

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is `copied_database`.
- Tables:** The `sales` table is selected.
- Code Editor:** The query window contains the following code:

```

24 -- order by s.amount desc
25 -- limit 10;
26
27 * select Store_id from sales
28 where Store_id>3800;
29
30 * select Item_id from sales
31 where Item_id=7;
    
```
- Result Grid:** The results for the first query (Store\_id) show all rows (Item\_id = 7).
- Action Output:** A log of 10 executions of the second query (select Item\_id from sales where Item\_id=7) is shown, with each execution taking approximately 0.031 seconds.

Average timing(Duration/Fetch) of this query run 10 times without index – 0.036sec/ 1.0828 sec

- **Query 3:**

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is `copied_database`.
- Tables:** The `sales` table is selected.
- Code Editor:** The query window contains the following code:

```

27 * select Store_id from sales
28 where Store_id>3800;
29
30 * select Item_id from sales
31 where Item_id=7;
32
33 * select Amount from sales
34 where Amount>300;
    
```
- Result Grid:** The results for the third query (Amount) show multiple rows.
- Action Output:** A log of 10 executions of the fourth query (select Amount from sales where Amount>300) is shown, with each execution taking approximately 0.031 seconds.

Average timing (Duration/Fetch) of this query run 10 times without index – 0.0359sec/ 0.9233 sec

- **Query 4:**

The screenshot shows the MySQL Workbench environment. In the top navigation bar, the 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', and 'Help' menus are visible. The 'Navigator' pane on the left displays the database schema, specifically the 'copied\_database' schema which contains tables like 'category', 'county', 'item', 'sales', and 'store'. The main workspace shows a 'Checking query' tab with the following SQL code:

```

34 where Amount>=300;
35
36 • select v.Vendor_id,i.Item_id,i.Item_description from item as i, vendor as v
37      where v.vendor_id = i.Vendor_Vendor_id and v.Vendor_id = 259;
38
39 • select Date from sales
40 where Date>="2015-01-01";

```

The 'Result Grid' below the code shows the results of the query, with 21 rows of data. The 'Action Output' pane at the bottom shows the execution details for 10 runs of the query, including the time taken for each run and the duration of the fetch operation.

#	Time	Action	Message	Duration / Fetch
1	20:22:20	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.172 sec
2	20:22:22	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
3	20:22:24	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.016 sec / 0.063 sec
4	20:22:25	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.063 sec
5	20:22:27	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.109 sec
6	20:22:29	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.078 sec
7	20:22:30	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
8	20:22:32	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
9	20:22:34	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.094 sec
10	20:22:36	select Date from sales where Date>="2015-01-01" LIMIT 0, 50000	50000 row(s) returned	0.000 sec / 0.079 sec

Average timing (Duration/Fetch) of this query run 10 times without index - 0.0032 sec/0.094 sec

## 1.9. Make a conclusion based on your findings in this part

**Answer:** Instead of having to search through every row inside a database table each time a database table is visited, indexes are utilized to locate data quickly. A database table's columns can be used to generate indexes, which serve as the foundation for both quick random lookups and effective access to ordered items. From the findings, we can observe that the average querying time (Duration/Fetch) for **copied\_database** which has no indexes is more when compared to the **mydb** which has indexes(automatic/custom). The querying process too longer in copied\_database due to the absence of indexes. Indexes in a database facilitates for faster querying. Hence, index is the best approach to locate the data quickly.

## PART 2

2.1. Explore your dataset and familiarize yourself with the dataset and its content

**Answer:** We have used **Bikez** and **Pakwheels** datasets.

The **Bikez** dataset contains information about the bike models, old and new as well as stats about each model. There are fields such as Model, Year, Category, Engine type, Torque, Rear suspension, Front tyre, Rear tyre, & Insurance costs etc. where all the fields are of the datatype **string**

The PakWheels dataset contains information about The largest online marketplace in Pakistan for buying and selling cars. It gathers a large number of brand-new, used, and certified pre-owned vehicles from a large number of dealers and individual sellers. The dataset has fields such as brand of datatype **object**, model of datatype **string**, modelDate of datatype **integer** and so on.

2.2. Explain why it is better to use non-relational databases such as MongoDB to work with such a dataset.

**Answer:**

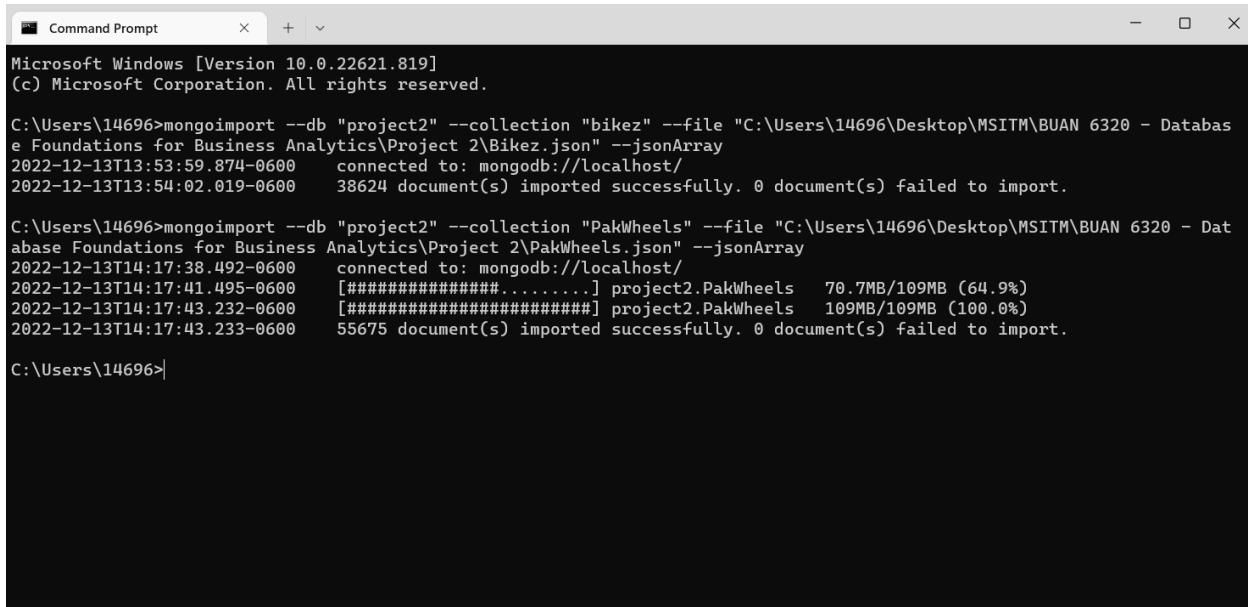
It is better to use non-relational databases such as MongoDB to work with such datasets because a query doesn't need to view multiple tables to provide an answer, as it frequently does with relational datasets, non-relational databases frequently operate more quickly. Therefore, non-relational databases are perfect for applications that manage a wide variety of data types or for storing data that could be modified often. NoSQL databases frequently offer better performance and are more scalable than relational databases. It is best suited for storing data which are often changed.

Ex: The objects in the dataset PakWheels cannot be written in a single structured table. The fields of datatype Objects need to be stored in a separate table. Hence it is best to use non-relational database such as MongoDB so that there is no need to view multiple tables to deliver an answer.

2.3. Import your dataset into MongoDB executing the following commands in the system terminal:

**Answer:**

- mongoimport --db "project2" --collection "bikez" --file "C:\Users\14696\Desktop\MSITM\BUAN 6320 - Database Foundations for Business Analytics\Project 2\Bikez.json" --jsonArray
- mongoimport --db "project2" --collection "bikez" --file "C:\Users\14696\Desktop\MSITM\BUAN 6320 - Database Foundations for Business Analytics\Project 2\PakWheels.json" --jsonArray



```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\14696>mongoimport --db "project2" --collection "bikez" --file "C:\Users\14696\Desktop\MSITM\BUAN 6320 - Database Foundations for Business Analytics\Project 2\Bikez.json" --jsonArray
2022-12-13T13:53:59.874-0600      connected to: mongodb://localhost/
2022-12-13T13:54:02.019-0600      38624 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\14696>mongoimport --db "project2" --collection "PakWheels" --file "C:\Users\14696\Desktop\MSITM\BUAN 6320 - Database Foundations for Business Analytics\Project 2\PakWheels.json" --jsonArray
2022-12-13T14:17:38.492-0600      connected to: mongodb://localhost/
2022-12-13T14:17:41.495-0600      [#####] project2.PakWheels    70.7MB/109MB (64.9%)
2022-12-13T14:17:43.232-0600      [#####] project2.PakWheels    109MB/109MB (100.0%)
2022-12-13T14:17:43.233-0600      55675 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\14696>
```

2.4. List some of the attributes (field/properties) of your database which are common among all documents.

**Answer:** “\_id”, “Model”, “Year”, “Category”, “Displacement”, “Engine type”, “Torque”, “Bore x stroke”, “Rear suspension”, “Front tyre”, “Rear tyre”, & “Insurance costs”.

2.4.1. For these fields, provide some of the values they contain in the database

**Answer:**

```
"_id": "ObjectId("639a086dee940f5793e24976")"

"Model": "AJP PR7", "Aprilia Dorsoduro 900",

"Year": "2021", "2021"

"Category": "Sport", "Super motard"

"Displacement": "600.0 ccm (36.61 cubic inches)", "896.1 ccm (54.68 cubic inches)"

"Engine type": "Single cylinder, four-stroke", "V2, four-stroke"

"Torque": "58.0 Nm (5.9 kgf-m or 42.8 ft.lbs)", "90.0 Nm (9.2 kgf-m or 66.4 ft.lbs) @ 6500 RPM"

"Bore x stroke": "100.0 x 76.4 mm (3.9 x 3.0 inches)", "92.0 x 67.4 mm (3.6 x 2.7 inches)"

"Rear suspension": "ZF Sachs prog. system ", "Aluminium alloy swingarm Sachs monoshock absorber adjustable in preload and hydraulic rebound damping.",

"Front tyre": "90/90-21 ", "120/70-ZR17 "

"Rear tyre": "140/80-18 ", "180/55-ZR17 "

"Insurance costs": "Compare US insurance quotes from the nation's top providers."
```

2.5. List some of the attributes (field/properties) of your database which are not common among all the documents

**Answer:** "Fuel system", "Fuel control", "Cooling system" "Gearbox" "Transmission type, "final drive", "Frame type", "Rake (fork angle)", "Trail", "Front suspension", "Price as new", "Engine details", "Power", "Compression"

2.5.1. For these fields, provide some of the values they contain in the database

**Answer:** "Fuel system": "Injection. Mikuni"

"Fuel control": "Double Overhead Cams/Twin Cam (DOHC)"

"Cooling system": "Liquid"

"Gearbox": "5-speed"

"Transmission type,final drive": "Chain"

"Frame type": "Aluminum/steel"

"Rake (fork angle)": "26.5°"

"Trail": "300 mm (11.8 inches)"

"Front suspension": "ZF Sachs Ø48 mm close-circuit"

"Price as new": " Euro 9990. MSRP depend on country, taxes, accessories, etc."

"Engine details": "V90 longitudinal 90° V twin."

"Power": "95.2 HP (69.5 kW) @ 8750 RPM"

"Compression": "11.0:1"

2.6. Write at least 5 queries using the key-value pairs you found in the previous steps to narrow down the result (provide the queries and results in your report)

**Answer:**

**Query 1:** To find the bikes which are made in 2020

```
db.getCollection('bikez').find({"Year":"2020"})
```

```

project2> db.getCollection('bikez').find({year:"2020"})
[{"_id": ObjectId("6398d857217846d578ce3573"), "Model": "AJS PRM Enduro 125", "Year": "2020", "Category": "Enduro / offroad", "Rating": "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement": "124.0 ccm (7.57 cubic inches)", "Engine type": "Single cylinder, four-stroke", "Power": "12.0 HP (8.9 kW) @ 12000 RPM", "Torque": "1.5 Nm (1.1 kgf-m or 8.3 ft.lbs) @ 8000 RPM", "Bore x stroke": "56.5 x 49.5 mm (2.2 x 1.9 inches)", "Valves per cylinder": "2", "Fuel system": "Carburetor, 38 mm", "Fuel control": "Single overhead cam (SOHC)", "Cooling system": "Air", "Gearbox": "5-speed", "Transmission type, final drive": "Chain", "Frame type": "Composite aluminum and steel", "Front suspension": "AJP Ø38mm fork, fully adjustable", "Front wheel travel": "+200 mm (9.4 inches)", "Rear suspension": "OHLV progressive system", "Rear wheel travel": "+200 mm (11.8 inches)", "Front tyre": "2.75-19", "Rear tyre": "3.25-17", "Front brakes": "240 mm (9.4 inches), 2 piston disc, 2 piston caliper", "Diameter": "200 mm (8 inches)", "Rear brakes": "Single disc", "Weight incl. oil, gas, etc": "104.0 kg (229.3 pounds)", "Seat height": "840 mm (33.1 inches) If adjustable, lowest setting.", "Ground clearance": "310 mm (12.2 inches)", "Wheelbase": "1360 mm (53.5 inches)", "Fuel capacity": "7.00 litres (1.85 gallons)", "Color options": "White/red/grey", "Starter": "Electric", "Comments": "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": "n/a", "\tChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n", "\t\tAlso check out our overview of motorcycle webshops at Bikez.info.", "Dirt-Bike parts": "Dirt-Bike parts gear available from Mega Motor Madness. Ships to most countries.", "Ask questions": "Join the 21 AJS PRM Enduro 125 discussion group or the general AJS discussion group.", "Related bikes": "List related bikes for comparison of specs"}, {"_id": ObjectId("6398d857217846d578ce3574"), "Model": "AJS PRM Enduro 200", "Year": "2020", "Category": "Enduro / offroad", "Rating": "Do you know this bike? Click here to rate it. We miss 1 vote to show the rating.", "Displacement": "231.0 ccm (14.32 cubic inches)", "Engine type": "Single cylinder, four-stroke", "Power": "20.0 HP (14.6 kW) @ 8000 RPM", "Bore x stroke": "58.0 x 50.0 mm (2.3 x 1.97 inches)", "Valve head size": "44.0 mm (1.73 inches)", "Fuel system": "Digital Fuel Injection", "Fuel control": "Digital Fuel Injection", "Cooling system": "Air", "Gearbox": "5-speed", "Transmission type, final drive": "Chain", "Frame type": "Aluminum/steel", "Front suspension": "AJP 43mm fork, fully adjustable", "Front wheel travel": "+200 mm (9.4 inches)", "Rear suspension": "OHLV progressive system", "Rear wheel travel": "+200 mm (11.8 inches)", "Front tyre": "2.75-19", "Rear tyre": "3.25-17", "Front brakes": "240 mm (9.4 inches), 2 piston disc, 2 piston caliper", "Diameter": "200 mm (8 inches)", "Rear brakes": "Single disc", "Weight incl. oil, gas, etc": "110.0 kg (242.5 pounds)", "Seat height": "840 mm (33.1 inches) If adjustable, lowest setting.", "Ground clearance": "310 mm (12.2 inches)", "Wheelbase": "1360 mm (53.5 inches)", "Fuel capacity": "7.00 litres (1.85 gallons)", "Color options": "White/black", "Starter": "Electric & kick", "Comments": "Portuguese made bike.", "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": "n/a", "\tChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n", "\t\tAlso check out our overview of motorcycle webshops at Bikez.info.", "Ask questions": "Join the 21 AJS PRM Enduro 125 discussion group or the general AJS discussion group.", "Related bikes": "List related bikes for comparison of specs"}]

```

## Query 2- To find all the Sport bikes, Super motards and Enduro/offroad bikes

```
db.getCollection('bikez').find({"Category": {$in:["Sport", "Super motard","Enduro / offroad"]}})
```

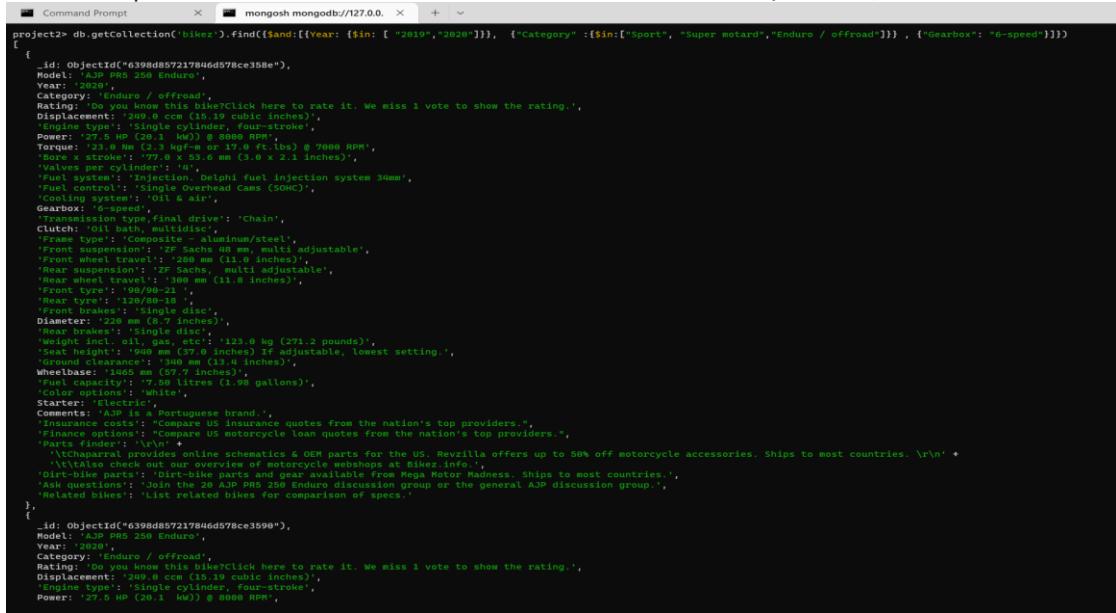
```

project2> db.getCollection('bikez').find({year:"2021"})
[{"_id": ObjectId("6398d857217846d578ce32d"), "Model": "Aprilia RS 125 GP Replica", "Year": "2021", "Category": "Sport", "Rating": "Do you know this bike? Click here to rate it. We miss 2 votes to show the rating.", "Displacement": "124.0 ccm (7.57 cubic inches)", "Engine type": "Single cylinder, four-stroke", "Power": "15.8 HP (11.9 kW) @ 10500 RPM", "Torque": "10.9 Nm (1.1 kgf-m or 8.0 ft.lbs) @ 8250 RPM", "Compression": "12.5:1", "Bore x stroke": "58.0 x 47.0 mm (2.3 x 1.9 inches)", "Valves per cylinder": "4", "Fuel system": "Digital Fuel Injection", "Fuel control": "Digital Fuel Injection", "Cooling system": "Air", "Gearbox": "5-speed", "Transmission type, final drive": "Chain", "Frame type": "Aluminum/steel", "Front suspension": "Aprilia 43mm fork, fully adjustable", "Front wheel travel": "+200 mm (9.4 inches)", "Rear suspension": "OHLV progressive system", "Rear wheel travel": "+200 mm (11.8 inches)", "Front tyre": "2.75-19", "Rear tyre": "3.25-17", "Front brakes": "240 mm (9.4 inches), 2 piston disc, 2 piston caliper", "Diameter": "200 mm (8 inches)", "Rear brakes": "Single disc", "Weight incl. oil, gas, etc": "110.0 kg (242.5 pounds)", "Seat height": "820 mm (36.2 inches) If adjustable, lowest setting.", "Ground clearance": "310 mm (12.2 inches)", "Wheelbase": "1350 mm (53.1 inches)", "Fuel capacity": "7.00 litres (1.85 gallons)", "Color options": "White/black", "Starter": "Electric", "Comments": "Portuguese made bike.", "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": "n/a", "\tChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n", "\t\tAlso check out our overview of motorcycle webshops at Bikez.info.", "Ask questions": "Join the 21 Aprilia RS 125 GP Replica discussion group or the general Aprilia discussion group.", "Related bikes": "List related bikes for comparison of specs"}]

```

**Query 3-** To find the Sport bikes, Super motards and Enduro/offroad bikes made in 2019 or 2020 which have a 6 speed gear box transmission.

```
db.getCollection('bikez').find({$and:[{Year: {$in: [ '2019','2020']}}, {"Category" :{$in:["Sport", "Super motard","Enduro / offroad"]}}, {"Gearbox": "6-speed"]})}
```



```
project> db.getCollection('bikez').find({$and:[{Year: {$in: [ '2019','2020']}}, {"Category" :{$in:["Sport", "Super motard","Enduro / offroad"]}}, {"Gearbox": "6-speed"]})
```

```
[{"_id": ObjectId("6398d857217846d578ce359e"), "Model": "AJP PRS 250 Enduro", "Year": "2020", "Category": "Sport", "Rating": "Do you know this bike? Click here to rate it. We miss 1 vote to show the rating.", "Displacement": "249.0 ccm (16.19 cubic inches)", "Power": "27.5 HP (20.1 kW) @ 8000 RPM", "Torque": "21.0 Nm (2.3 kgf-m or 17.8 ft-lbs) @ 7000 RPM", "Bore x stroke": "77.0 x 53.0 mm (3.0 x 2.1 inches)", "Valves per cylinder": "4", "Fuel system": "Electronic Delphi Fuel injection system 34mm", "Fuel control": "Single Overhead Cams (SOHC)", "Cooling system": "Oil & air", "Gearbox": "6-speed", "Transmission type": "final drive": "Chain", "Clutch": "Oil bath, multidiaphragm", "Frame type": "Composite - aluminum/steel", "Front suspension": "Telescopic, multi adjustable", "Front wheel travel": "200 mm (11.8 inches)", "Rear suspension": "ZF Sachs, multi adjustable", "Rear wheel travel": "180 mm (11.8 inches)", "Front brakes": "2x 300 mm disc", "Rear brake": "Single disc", "Disc brakes": "Front and rear", "Weight incl. oil, gas, etc": "123.0 kg (271.2 pounds)", "Seat height": "800 mm (37.0 inches) If adjustable, lowest setting.", "Wheelbase": "1460 mm (57.7 inches)", "Fuel capacity": "7.50 litres (1.98 gallons)", "Fuel options": "White", "Starter": "Electric", "Comments": "AJP is a Portuguese brand.", "Insurance costs": "Compare US insurance quotes from the nation's top providers.", "Parts availability": "Compare US motorcycle loan quotes from the nation's top providers.", "Parts Finder": "\n\n\tChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n\t\n\tAlso check out our overview of motorcycle webshops at Biker.info.", "Disclaimer": "This is a general overview of motorcycle types. See our specific model pages for more details.", "Ask questions": "Join the 20 AJP PRS 250 Enduro discussion group or the general AJP discussion group.", "Related bikes": "List related bikes for comparison of specs."}, {"_id": ObjectId("6398d857217846d578ce3599"), "Model": "AJP PRS 250 Enduro", "Year": "2019", "Category": "Enduro / offroad", "Rating": "Do you know this bike? Click here to rate it. We miss 1 vote to show the rating.", "Displacement": "249.0 ccm (16.19 cubic inches)", "Power": "27.5 HP (20.1 kW) @ 8000 RPM", "Torque": "21.0 Nm (2.3 kgf-m or 17.8 ft-lbs) @ 7000 RPM", "Bore x stroke": "77.0 x 53.0 mm (3.0 x 2.1 inches)", "Valves per cylinder": "4", "Fuel system": "Electronic Delphi Fuel injection system 34mm", "Fuel control": "Single Overhead Cams (SOHC)", "Cooling system": "Oil & air", "Gearbox": "6-speed", "Transmission type": "final drive": "Chain", "Clutch": "Oil bath, multidiaphragm", "Frame type": "Composite - aluminum/steel", "Front suspension": "Telescopic, multi adjustable", "Front wheel travel": "200 mm (11.8 inches)", "Rear suspension": "ZF Sachs, multi adjustable", "Rear wheel travel": "180 mm (11.8 inches)", "Front brakes": "2x 300 mm disc", "Rear brake": "Single disc", "Disc brakes": "Front and rear", "Weight incl. oil, gas, etc": "123.0 kg (271.2 pounds)", "Seat height": "800 mm (37.0 inches) If adjustable, lowest setting.", "Wheelbase": "1460 mm (57.7 inches)", "Fuel capacity": "7.50 litres (1.98 gallons)", "Fuel options": "White", "Starter": "Electric", "Comments": "AJP is a Portuguese brand.", "Insurance costs": "Compare US insurance quotes from the nation's top providers.", "Parts availability": "Compare US motorcycle loan quotes from the nation's top providers.", "Parts Finder": "\n\n\tChaparral provides online schematics & OEM parts for the US. Revzilla offers up to 50% off motorcycle accessories. Ships to most countries.\r\n\t\n\tAlso check out our overview of motorcycle webshops at Biker.info.", "Disclaimer": "This is a general overview of motorcycle types. See our specific model pages for more details.", "Ask questions": "Join the 20 AJP PRS 250 Enduro discussion group or the general AJP discussion group.", "Related bikes": "List related bikes for comparison of specs."}]
```

**Query 4-** To find all the used cars manufactured after 2010.

```
db.getCollection('PakWheels').find({$and: [{"itemCondition": "used"}, {"modelDate": { $gt: 2010} } ] })
```

```

{
  "_id": ObjectId("6398dde251af991c5d4c2bcc"),
  "type": "Car",
  "brand": {"type": "Brand", name: "Suzuki"},
  "model": "Mehran",
  "description": "I'm 100% original. Alloy Rims. New tires installed recently. Driven on petrol throughout. Never been into any accident. All original documents are complete. All token taxes are paid for life. Looking to sell the car urgently.",
  "itemCondition": "used",
  "modelDate": "2013",
  "manufacturer": "Suzuki",
  "fuelType": "Petrol",
  "name": "Suzuki Mehran 2013 for sale in Rawalpindi",
  "image": "https://cache1.pakwheels.com/ad_pictures/5534/suzuki-mehran-vx-euro-ii-2013-55345059.jpg",
  "vehicleTransmission": "Manual",
  "color": "White",
  "bodyType": "Hatchback",
  "vehicleEngine": {"type": "EngineSpecification", engineDisplacement: "800cc"},
  "mileageFromOdometer": "02,000 km",
  "sellerLocation": "Rawalpindi Punjab",
  "postedFrom": "Added via Website",
  "keywords": "suzuki mehran 2013 for sale, suzuki mehran 2013, suzuki mehran 2013 rawalpindi, 2013 suzuki mehran, suzuki mehran for sale, used suzuki mehran 2013",
  "extraFeatures": [
    {"RegisteredIn": "Islamabad",
     "Color": "White",
     "Assembly": "Local",
     "EngineCapacity": "800 cc",
     "BodyType": "Hatchback",
     "LastUpdated": "Aug 27, 2021",
     "Adref#": "5546013"
   },
   "features": [
     "AM/FM Radio",
     "Alloy Rims",
     "Cassette Player",
     "Immobilizer Key"
   ],
   "adLastUpdated": "Aug 27, 2021",
   "price": 799999,
   "priceCurrency": "PKR"
 },
 {
  "_id": ObjectId("6398dde251af991c5d4c2bcc"),
  "type": "Car",
  "brand": {"type": "Brand", name: "Suzuki"},
  "model": "Cultus",
  "description": "I piece touch first owner new tyre urgent sale need cash",
  "itemCondition": "used",
  "modelDate": "2016",
  "manufacturer": "Suzuki",
  "fuelType": "Petrol",
  "name": "Suzuki Cultus 2016 for sale in Lahore",
  "image": "https://cache1.pakwheels.com/ad_pictures/5533/suzuki-cultus-limited-edition-2-2016-55334978.jpg",
  "vehicleTransmission": "Manual",
  "color": "Grey",
  "bodyType": "Hatchback",
  "vehicleEngine": {"type": "EngineSpecification", engineDisplacement: "1000cc"},
  "adLastUpdated": "Aug 27, 2021",
  "price": 799999,
  "priceCurrency": "PKR"
 }
}

64°F Cloudy Search ENG US 4:01 PM 12/13/2022

```

**Query 5-** To find all the used 3.5 liter petrol cars manufactured after 2010 of price less than 3.5 million PKR(Pakistani Rupees)

```
db.getCollection('PakWheels').find({$and: [{"itemCondition": "used"}, {"modelDate": { $gt: 2010} }, {"fuelType": "Petrol"}, {"extraFeatures.EngineCapacity": "3500 cc"}, {"price": {$lt:3500000}}]})
```

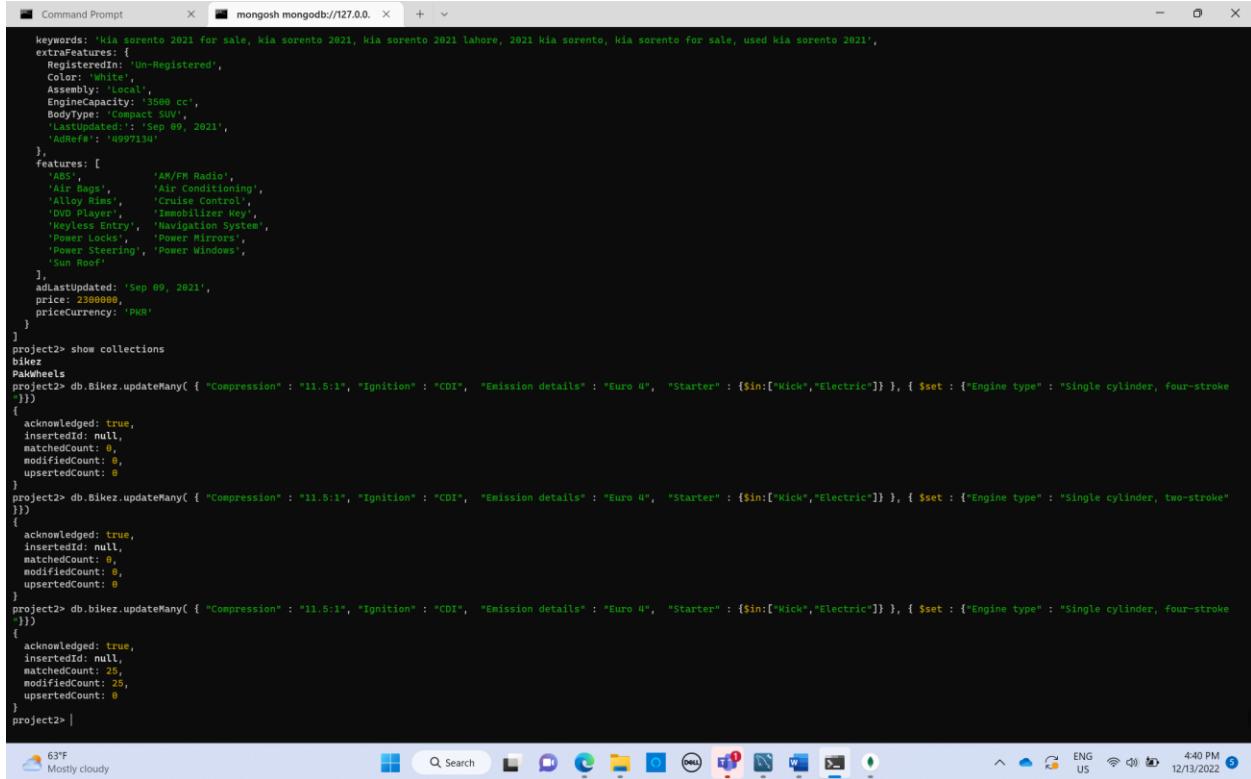
```

project2> db.getCollection('PakWheels').find({$and: [{"itemCondition": "used"}, {"modelDate": { $gt: 2010} }, {"fuelType": "Petrol"}, {"extraFeatures.EngineCapacity": "3500 cc"}, {"price": {$lt:3500000}}]}) 
UncaughtError: Unexpected character ``'. (1:11)
> 1 > db.getCollection('PakWheels').find({$and: [{"itemCondition": "used"}, {"modelDate": { $gt: 2010} }, {"fuelType": "Petrol"}, {"extraFeatures.EngineCapacity": "3500 cc"}, {"price": {$lt:3500000}}]}) 
2 | 
project2> db.getCollection('PakWheels').find({$and: [{"itemCondition": "used"}, {"modelDate": { $gt: 2010} }, {"fuelType": "Petrol"}, {"extraFeatures.EngineCapacity": "3500 cc"}, {"price": {$lt:3500000}}]}) 
[
  {
    "_id": ObjectId("6398dde451af991c5d4c6913"),
    "type": "Car",
    "brand": {"type": "Brand", name: "Toyota"},
    "model": "Corolla",
    "description": "full ok",
    "itemCondition": "used",
    "modelDate": "2016",
    "manufacturer": "Toyota",
    "fuelType": "Petrol",
    "name": "Toyota Corolla 2016 for sale in Faisalabad",
    "image": "https://cache2.pakwheels.com/ad_pictures/5598/toyota-corolla-xli-vvti-2016-55980116.jpg",
    "vehicleTransmission": "Manual",
    "color": "Black",
    "bodyType": "Sedan",
    "vehicleEngines": {"type": "EngineSpecification", engineDisplacement: "1300cc"},
    "mileageFromOdometer": "02,000 km",
    "sellerLocation": "Lahore Punjab",
    "postedFrom": "Added via Website",
    "keywords": "kia sorento 2021 for sale, kia sorento 2021 lahore, 2021 kia sorento, kia sorento for sale, used kia sorento 2021",
    "extraFeatures": [
      {"RegisteredIn": "Un-Registered",
       "Color": "White",
       "Assembly": "Local",
       "EngineCapacity": "1500 cc",
       "BodyType": "SUV/Crossover SUV",
       "LastUpdated": "Sep 09, 2021",
       "Adref#": "4099713#"
     },
     "features": [
       "ABS",
       "AM/FM Radio",
       "Air Bags",
       "Air Conditioning",
       "Alloy Rims",
       "Cruise Control",
       "CD Player",
       "Central Locking",
       "Keyless Entry",
       "Navigation System",
       "Power Locks",
       "Power Mirrors",
       "Power Steering",
       "Power Windows",
       "Sun Roof"
     ],
     "adLastUpdated": "Sep 09, 2021",
     "price": 2300000,
     "priceCurrency": "PKR"
   }
]
```

2.7. Write at least 5 update queries to update some of the values in your database

**Answer:**

- db.bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { \$in:[ "Kick", "Electric" ] } }, { \$set : { "Engine type" : "Single cylinder, **four**-stroke" } } )

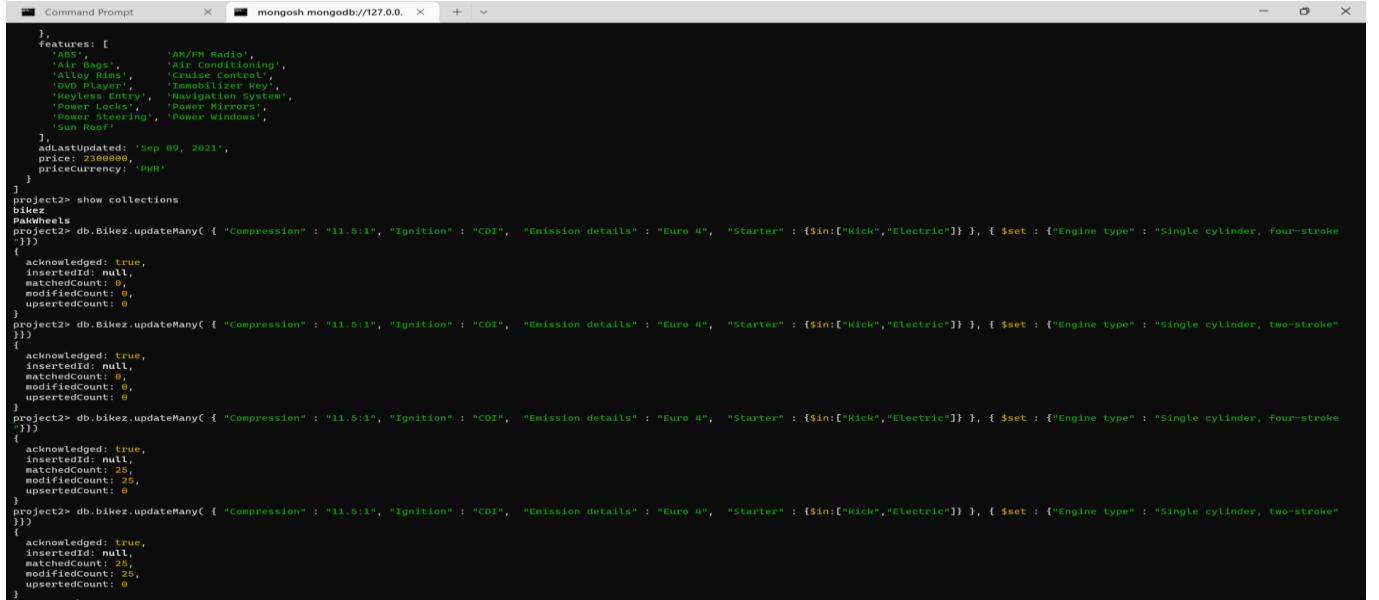


```

Keywords: 'kia sorento 2021 for sale', 'kia sorento 2021', 'kia sorento 2021 lahore', '2021 kia sorento', 'kia sorento for sale', 'used kia sorento 2021',
extraFeatures: [
  'Un-Registered',
  'Color: white',
  'Assembly: local',
  'EngineCapacity: <1500 cc',
  'BodyType: Compact SUV',
  'LastUpdated: Sep 09, 2021',
  'AdRef#: 499713H'
],
features: [
  'ABS', 'AM/FM Radio',
  'Air Bags', 'Air Conditioning',
  'Alloy Rims', 'Cruise Control',
  'DVD Player', 'Immobilizer Key',
  'Keyless Entry', 'Navigation System',
  'Power Locks', 'Power Mirrors',
  'Power Steering', 'Power Windows',
  'Sun Roof'
],
adLastUpdated: 'Sep 09, 2021',
price: 2300000,
priceCurrency: 'PKR'
]
]
project2> show collections
bikez
PakWheels
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "Single cylinder, four-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "Single cylinder, two-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "Single cylinder, four-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 25,
  modifiedCount: 25,
  upsertedCount: 0
}
project2> |

```

- db.bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { \$in:[ "Kick", "Electric" ] } }, { \$set : { "Engine type" : "Single cylinder, **two**-stroke" } } )



```

},
features: [
  'ABS', 'AM/FM Radio',
  'Air Bags', 'Air Conditioning',
  'Alloy Rims', 'Cruise Control',
  'DVD Player', 'Immobilizer Key',
  'Keyless Entry', 'Navigation System',
  'Power Locks', 'Power Mirrors',
  'Power Steering', 'Power Windows',
  'Sun Roof'
],
adLastUpdated: 'Sep 09, 2021',
price: 2300000,
priceCurrency: 'PKR'
]
]
project2> show collections
bikez
PakWheels
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "Single cylinder, four-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "single cylinder, two-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "single cylinder, four-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 25,
  modifiedCount: 25,
  upsertedCount: 0
}
project2> db.Bikez.updateMany( { "Compression" : "11.5:1", "Ignition" : "CDI", "Emission details" : "Euro 4", "Starter" : { $in:[ "Kick", "Electric" ] } }, { $set : { "Engine type" : "single cylinder, two-stroke" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 25,
  modifiedCount: 25,
  upsertedCount: 0
}

```

- db.bikez.updateMany( { "Year" : "2021", "Category" : { \$in:[ "Sport", "Super motard", "Enduro / offroad" ] } }, { \$set : { "Gearbox" : "7-speed" } } )

```

project2> db.bikez.updateMany( { "Year" : "2021", "Category" : { $in:["Sport","Super motard","Enduro / offroad"]} }, { $set : {"Gearbox" : "7-speed"} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 234,
  modifiedCount: 234,
  upsertedCount: 0
}
project2> |

```

- db.PakWheels.updateMany( { "extraFeatures.EngineCapacity" : "3500 cc"}, { \$set : {"price" : 4500000}})

```

project2> db.bikez.updateMany( { "Year" : "2021", "Category" : { $in:["Sport","Super motard","Enduro / offroad"]} }, { $set : {"Gearbox" : "7-speed"} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 234,
  modifiedCount: 234,
  upsertedCount: 0
}
project2> db.PakWheels.updateMany( { "extraFeatures.EngineCapacity" : "3500 cc"}, { $set : {"price" : 4500000}} )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 165,
  modifiedCount: 165,
  upsertedCount: 0
}
project2> |

```

- db.PakWheels.updateMany( { "modelDate" : {\$gt: 2015}, "bodyType" : "SUV"}, { \$set : {"price" : 3500000}})

```

project2> db.PakWheels.updateMany( { "modelDate" : {$gt: 2015}, "bodyType" : "SUV"}, { $set : {"price" : 3500000}} )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1624,
  modifiedCount: 1624,
  upsertedCount: 0
}
project2> |

```

2.8. Write at least 5 queries to insert new documents into your database (provide the queries and results in your report)

**Answer:**

- db.bikez.insertOne({Model:'Aprilia SX 500',Year:'2022', Category:'Super motard', Price:'\$5000',Gearbox:'7-speed manual', 'Color options':[ 'Black,silver,red and white', 'Blue,gold,red and white']})
- db.bikez.insertOne({Model:'Ducati Superleggera',Year:'2023', Category:'Sport', Price:'\$35000',Gearbox:'7-speed manual', 'Color options':[ 'red', 'white']})
- db.bikez.insertOne({Model:'KTM Super Duke 1290',Year:'2023', Category:'Naked', Price:'\$5500',Gearbox:'7-speed manual', 'Color options':[ 'orange', 'white', 'black']})
- db.PakWheels.insertOne({model:'Supra', modelDate:2022, itemCondition: 'new'})
- db.PakWheels.insertOne({Model:'Nissan GTR35',Year:2015, Category: 'used'})

```

project2> db.PakWheels.updateMany( { "modelDate" : { $gt: 2015}, "bodyType" : "SUV"}, { $set : {"price" : 3500000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1624,
  modifiedCount: 1624,
  upsertedCount: 0
}
project2> db.bikez.insertOne({Model:'Aprilia SX 500',Year:'2022', Category:'Super mard', Price:'$5000',Gearbox:'7-speed manual', 'Color options':[ 'Black,silver,red and white', 'Blue,gold,red and white']})

{
  acknowledged: true,
  insertedId: ObjectId("639906d825c81bf7e81c7401")
}
project2> db.bikez.insertOne({Model:'Ducati Superleggera',Year:'2023', Category:'Sport', Price:'$35000',Gearbox:'7-speed manual', 'Color options':[ 'red', 'white']})
{
  acknowledged: true,
  insertedId: ObjectId("639907cf25c81bf7e81c7402")
}
project2> db.bikez.insertOne({Model:'KTM Super Duke 1290',Year:'2023', Category:'Naked', Price:'$5500',Gearbox:'7-speed manual', 'Color options':[ 'orange', 'white', 'black']})
{
  acknowledged: true,
  insertedId: ObjectId("639908f225c81bf7e81c7403")
}
project2> db.PakWheels.insertOne({model: 'Supra', modelDate:2022, itemCondition: 'new'}
...
...
project2> db.PakWheels.insertOne({model: 'Supra', modelDate:2022, itemCondition: 'new'})

{
  acknowledged: true,
  insertedId: ObjectId("639916a725c81bf7e81c7404")
}
project2> db.PakWheels.insertOne({Model: 'Nissan GTR35',Year:2015, Category: 'used'})
{
  acknowledged: true,
  insertedId: ObjectId("6399172425c81bf7e81c7405")
}
project2>

```

2.9. Write at least 5 delete queries to remove documents from your database (provide the queries and results in your report)

**Answer:** Deleted the records added in the above step.

- db.bikez.remove ( { \_id : { \$in: [ ObjectId("639906d825c81bf7e81c7401"), ObjectId("639907cf25c81bf7e81c7402"), ObjectId("639908f225c81bf7e81c7403") ] } })
- db.PakWheels.remove ( { \_id : { \$in: [ ObjectId("639916a725c81bf7e81c7404"), ObjectId("6399172425c81bf7e81c7405") ] } })

```

project2> db.PakWheels.updateMany( { "modelDate" : { $gt: 2015}, "bodyType" : "SUV"}, { $set : {"price" : 3500000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1624,
  modifiedCount: 1624,
  upsertedCount: 0
}
project2> db.bikez.insertOne({Model:'Aprilia SX 500',Year:'2022', Category:'Super mard', Price:'$5000',Gearbox:'7-speed manual', 'Color options':[ 'Black,silver,red and white', 'Blue,gold,red and white']})

{
  acknowledged: true,
  insertedId: ObjectId("639906d825c81bf7e81c7401")
}
project2> db.bikez.insertOne({Model:'Ducati Superleggera',Year:'2023', Category:'Sport', Price:'$35000',Gearbox:'7-speed manual', 'Color options':[ 'red', 'white']})
{
  acknowledged: true,
  insertedId: ObjectId("639907cf25c81bf7e81c7402")
}
project2> db.bikez.insertOne({Model:'KTM Super Duke 1290',Year:'2023', Category:'Naked', Price:'$5500',Gearbox:'7-speed manual', 'Color options':[ 'orange', 'white', 'black']})
{
  acknowledged: true,
  insertedId: ObjectId("639908f225c81bf7e81c7403")
}
project2> db.PakWheels.insertOne({model: 'Supra', modelDate:2022, itemCondition: 'new'}
...
...
project2> db.PakWheels.insertOne({model: 'Supra', modelDate:2022, itemCondition: 'new'})

{
  acknowledged: true,
  insertedId: ObjectId("639916a725c81bf7e81c7404")
}
project2> db.PakWheels.insertOne({Model: 'Nissan GTR35',Year:2015, Category: 'used'})
{
  acknowledged: true,
  insertedId: ObjectId("6399172425c81bf7e81c7405")
}
project2> db.bikez.remove ( { _id : { $in: [ ObjectId("639906d825c81bf7e81c7401"), ObjectId("639907cf25c81bf7e81c7402"), ObjectId("639908f225c81bf7e81c7403") ] } })
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{
  acknowledged: true, deletedCount: 3
}
project2> db.PakWheels.remove ( { _id : { $in: [ ObjectId("639916a725c81bf7e81c7404"), ObjectId("6399172425c81bf7e81c7405") ] } })
{
  acknowledged: true, deletedCount: 2
}
project2>

```

## Deleting other records

- db.PakWheels.remove({ \_id: ObjectId("6398dde251af991c5d4c2bca") })  
project2> db.PakWheels.remove({ \_id: ObjectId("6398dde251af991c5d4c2bca") })  
{ acknowledged: true, deletedCount: 1 }  
project2> |
  - db.bikez.remove ( {"Model":"AJP PR7"})

```
project2> db.bikez.remove ( {"Model":"AJP PR7"} )  
{ acknowledged: true, deletedCount: 2 }
```

- db.bikez.remove ( {"Model":"Aprilia RS 125 GP Replica "})

```
project2> db.bikez.remove ( {"Model":"Aprilia RS 125 GP Replica "} )  
{ acknowledged: true, deletedCount: 3 }
```