

JONATHAN DSOUZA
60004220118

COMPS A

C12

EXP 2 - Stimulate query optimization

The screenshot shows the MySQL Workbench interface with a query window open. The query is:

```
1 • SELECT username, password FROM ga3.user_details where gender='Male'
```

The Query Statistics pane displays the following details:

- Timing (as measured at client side):** 0 rows, 0 bytes, 0.000 sec.
- Timing (as measured by the server):** Execution time: 0:00:0.0119000, Table lock wait time: 0:00:0.0000000
- Errors:** Errors: 0, Warnings: 0
- Rows Processed:** Rows affected: 0, Rows inserted: 0, Rows updated: 0, Rows examined: 2045
- Temporary Tables:** Temporary disk tables created: 0, Temporary tables created: 0

The Results pane shows the output of the query:

Action	Time	Action	Message	Duration / Fetch
4	09:09:37	SELECT username, password FROM ga3.user_details where gender='Male' LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
5	09:09:35	SELECT username, password FROM ga3.user_details where gender='Male' LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec
6	09:10:08	select COUNT(username) FROM ga3.user_details LIMIT 0, 1000	1 row(s) returned	0.500 sec / 0.000 sec
7	09:11:27	optimize table user_details	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.	0.000 sec
8	09:11:38	optimize table ga3.user_details	1 row(s) returned	0.047 sec / 0.000 sec
9	09:14:36	SELECT username, password FROM ga3.user_details where gender='Male' LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec

First query

MySQL Workbench

Query 1

```

1 •  SELECT username, password FROM `ga1`.`user_details` where gender='Male';
2
3 •  select sum(user_id) FROM `ga1`.`user_details`;
4
5 •  select * from `ga1`.`user_details` order by user_id desc;

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:00.080000000
Table lock wait time: 0:00:00.000000000

Timing (as measured by the server):
Execution time: 0:00:00.4919240
Table lock wait time: 0:00:00.000000000

Errors:
None found.
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 48741
Rows examined: 100000

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Index Usage:
No Index used

Other Info:
Event Id: 40
Thread Id: 362

user_details 6

Action Output

#	Time	Action	Message	Duration / Fetch
1	29.31.33	select COUNT(username) FROM `ga1`.`user_details` LIMIT 0, 1000	1 row(s) returned	0.453 sec / 0.000 sec
2	3 09.22.25	select sum(user_id) FROM `ga1`.`user_details` LIMIT 0, 50000	1 row(s) returned	0.453 sec / 0.000 sec
3	4 09.24.12	select * from `ga1`.`user_details` order by user_id desc LIMIT 0, 50000	Error Code: 1045: No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.	0.500 sec / 0.016 sec
4	5 09.34.21	select * from `ga1`.`user_details` order by user_id desc LIMIT 0, 50000	50000 rows(s) returned	0.500 sec / 0.016 sec
5	6 09.35.02	select sum(user_id) FROM `ga1`.`user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
6	7 09.35.35	SELECT username, password FROM `ga1`.`user_details` where gender='Male' LIMIT 0, 50000	48741 row(s) returned	0.000 sec / 0.484 sec

Object Info **Session**

Second query

MySQL Workbench

Query 1

```

1 •  SELECT username, password FROM `ga1`.`user_details` where gender='Male';
2
3 •  select sum(user_id) FROM `ga1`.`user_details`;
4
5 •  select * from `ga1`.`user_details` order by user_id desc;

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:00.035000000
Table lock wait time: 0:00:00.000000000

Timing (as measured by the server):
Execution time: 0:00:00.01320430
Table lock wait time: 0:00:00.000000000

Errors:
None found.
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 100000

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Index Usage:
At least one index was used

Other Info:
Event Id: 42
Thread Id: 362

Result 7

Action Output

#	Time	Action	Message	Duration / Fetch
1	3 09.32.25	select sum(user_id) FROM `ga1`.`user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
2	4 09.34.12	select * from `ga1`.`user_details` order by user_id desc LIMIT 0, 50000	Error Code: 1045: No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.	0.000 sec
3	5 09.34.21	select * from `ga1`.`user_details` order by user_id desc LIMIT 0, 50000	50000 rows(s) returned	0.500 sec / 0.016 sec
4	6 09.35.02	select sum(user_id) FROM `ga1`.`user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
5	7 09.35.35	SELECT username, password FROM `ga1`.`user_details` where gender='Male' LIMIT 0, 50000	48741 row(s) returned	0.000 sec / 0.484 sec
6	8 09.36.11	select sum(user_id) FROM `ga1`.`user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec

Object Info **Session**

Third query

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

- Schemas:** pg12
- Query Editor:**

```

1 • SELECT username, password FROM `gaj.user_details` where gender='Male';
2
3 • select sum(user_id) FROM `gaj.user_details`;
4
5 • select * from `gaj.user_details` order by user_id desc;
    
```
- Query Statistics:**
 - Timing (as measured at client side): Execution time: 0:00:0.5000000
 - Timing (as measured by the server): Execution time: 0:00:0.4950000
 - Table lock wait time: 0:00:0.0000000
 - Errors: Red error icon: No Warnings: 0
 - Rows Processed: Rows affected: 0 Rows inserted: 0 Rows updated: 30000 Rows examined: 150000
 - Temporary Tables: Temporary disk tables created: 0 Temporary tables created: 0
- Output Tab:**

Action	Time	Message	Duration / Fetch
4 09:35:12	select * from `user_details` order by user_id desc LIMIT 0, 50000	Error Code: 1045: No database selected	0.000 sec / 0.032 sec
5 09:35:21	select * from `pg.user_details` order by user_id desc LIMIT 0, 50000	50000 rows returned	0.500 sec / 0.016 sec
6 09:35:02	select sum(user_id) FROM `pg.user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
7 09:35:35	SELECT `username, password` FROM `pg.user_details` where gender='Male' LIMIT 0, 50000	40741 rows returned	0.000 sec / 0.494 sec
8 09:35:11	select * from `gaj.user_details` FROM `gaj.user_details` LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
9 09:35:55	select * from `pg.user_details` order by user_id desc LIMIT 0, 50000	50000 rows returned	0.500 sec / 0.032 sec

Fourth query

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

- Schemas:** pg12
- Query Editor:**

```

1 • SELECT username, password FROM `gaj.user_details` where gender='Male';
2
3 • select sum(user_id) FROM `gaj.user_details`;
4
5 • select * from `gaj.user_details` order by user_id desc;
6
7 • select * from `gaj.user_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male');
    
```
- Query Statistics:**
 - Timing (as measured at client side): Execution time: 0:00:0.1200000
 - Timing (as measured by the server): Execution time: 0:00:0.4950000
 - Table lock wait time: 0:00:0.0000000
 - Errors: Red error icon: No Warnings: 0
 - Rows Processed: Rows affected: 0 Rows inserted: 1264 Rows updated: 0 Rows examined: 101264
 - Temporary Tables: Temporary disk tables created: 0 Temporary tables created: 0
- Output Tab:**

Action	Time	Message	Duration / Fetch
9 09:36:55	select * from `pg.user_details` order by user_id desc LIMIT 0, 50000	50000 rows returned	0.500 sec / 0.032 sec
10 09:39:30	select * from `pg.users_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000	Error Code: 1045: No database selected	0.000 sec
11 09:39:41	select * from `pg.users_details` where user_id in (select user_id from pg.user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000	Error Code: 1146: Table 'pg.users_details' doesn't exist	0.000 sec
12 09:40:07	select * from `pg.users_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000	Error Code: 1045: No database selected	0.000 sec
13 09:40:26	select * from `pg.users_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000	Error Code: 1045: No database selected	0.000 sec
14 09:41:34	select * from `pg.user_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000	1264 rows returned	0.110 sec / 0.390 sec

Fifth query

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

- Query Editor:** Contains the following SQL code:

```
1 •  SELECT username, password FROM `gaj`.user_details where gender='Male'
2
3 •  select sum(user_id) FROM `gaj`.user_details;
4
5 •  select * from `gaj`.user_details order by user_id desc;
6
7 •  select * from `gaj`.user_details where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male');
8
9 •  select u1.user_id,u1.first_name from `gaj`.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
```
- Query Statistics:** Displays performance metrics for the query execution.
- Output:** Shows the results of the query, which is empty (0 rows).

Optimize

First query

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas: gaj

Query 1: user_details user_details user_details user_details user_details user_details

```

5 • select * from gaj.user_details order by user_id desc;
6
7 • select * from gaj.user_details where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male');
8
9 • select u1.user_id,u1.first_name from gaj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
10
11 • optimize table user_details;
12
13 • SELECT username, password FROM gaj.user_details where gender='Male';
14
15

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.8380000
Timing (as measured by the server):
Execution time: 0:00:0.477e-005
Table lock wait time: 0:00:0.0000000300

Errors:
No Errors

Rows Processed:
Rows affected: 0
Rows inserted: 48741
Rows examined: 200000

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Object Info Session

Result Grid Form Editor Field Types Query Stats Execution Plan

Read Only Context Help Snippets

Duration / Fetch

0.000 sec 0.000 sec 0.000 sec 0.110 sec / 0.390 sec 0.016 sec / 0.531 sec 0.016 sec / 0.459 sec 0.016 sec / 0.465 sec

Type here to search

09:46 19-10-2023 ENG

Second query

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas: gaj

Query 1: user_details user_details user_details user_details user_details user_details

```

7 • select * from gaj.user_details where user_id in (select user_id from gaj.user_details where last_name = 'rogers' and gender='male') LIMIT 0, 50000;
8
9 • select u1.user_id,u1.first_name from gaj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
10
11 • optimize table user_details;
12
13 • SELECT username, password FROM gaj.user_details where gender='Male';
14
15 • select sum(user_id) FROM gaj.user_details;
16
17

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.8380000
Timing (as measured by the server):
Execution time: 0:00:0.477e-005
Table lock wait time: 0:00:0.0000000300

Errors:
No Errors

Rows Processed:
Rows affected: 0
Rows inserted: 1
Rows examined: 200000

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Object Info Session

Result Grid Form Editor Field Types Query Stats Execution Plan

Read Only Context Help Snippets

Duration / Fetch

0.000 sec 0.000 sec 0.000 sec 0.110 sec / 0.390 sec 0.016 sec / 0.531 sec 0.016 sec / 0.459 sec 0.016 sec / 0.465 sec

Type here to search

09:46 19-10-2023 ENG

Third query

MySQL Workbench

Query 1 user_details user_details user_details user_details user_details user_details

```

9 • select u1.user_id,u1.first_name from gaj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
10 • optimize table user_details;
11 •
12 • SELECT username, password FROM gaj.user_details where gender='Male';
13 •
14 •
15 • select sum(user_id) FROM gaj.user_details;
16 •
17 • select * from gaj.user_details order by user_id desc;
18 •
19 •

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.8380000
Timing (as measured by the server):
Execution time: 0:00:0.3350000
Table lock wait time: 0:00:0.0000000

Errors:
No Errors / Warnings: 0

Rows Processed:
Rows affected: 0
Rows inserted: 0
Rows updated: 0
Rows examined: 150000

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

user_details 13 x

Output:

#	Time	Action	Message	Duration / Fetch
13	09:41:26	select * from gaj.user_details where user_id in (select user_id from user_details where last_name='rogers' and gender='male') LIMIT 0, 50000	Error Code: 1045: No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.	0.000 sec
14	09:41:34	select * from gaj.user_details where user_id in (select user_id from user_details where last_name='rogers' and gender='male') LIMIT 0, 50000	1264 rows) returned	0.110 sec / 0.390 sec
15	09:41:11	select u1.user_id,u1.first_name from gaj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status LIMIT 0, 50000	50000 rows) returned	0.016 sec / 0.531 sec
16	09:46:35	SELECT username, password FROM gaj.user_details where gender='Male' LIMIT 0, 50000	48741 rows) returned	0.016 sec / 0.469 sec
17	09:47:56	select sum(user_id) FROM gaj.user_details LIMIT 0, 50000	1 rows) returned	0.016 sec / 0.000 sec
18	09:48:39	select * from gaj.user_details order by user_id desc LIMIT 0, 50000	50000 rows) returned	0.500 sec / 0.016 sec

Object Info Session

Fourth query

MySQL Workbench

Query 1 user_details user_details user_details user_details user_details user_details

```

9 • optimize table user_details;
10 •
11 •
12 •
13 • SELECT username, password FROM gaj.user_details where gender='Male';
14 •
15 • select sum(user_id) FROM gaj.user_details;
16 •
17 • select * from gaj.user_details order by user_id desc;
18 •
19 • select * from gaj.user_details where user_id in (select user_id from user_details where last_name='rogers' and gender='male');
20 •
21 •

```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.9400000
Timing (as measured by the server):
Execution time: 0:00:0.3400000
Table lock wait time: 0:00:0.0000000

Errors:
No Errors / Warnings: 0

Rows Processed:
Rows affected: 0
Rows inserted: 0
Rows updated: 0
Rows examined: 10234

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

user_details 14 x

Output:

#	Time	Action	Message	Duration / Fetch
14	09:41:34	select * from gaj.user_details where user_id in (select user_id from user_details where last_name='rogers' and gender='male') LIMIT 0, 50000	1264 rows) returned	0.110 sec / 0.390 sec
15	09:41:11	select u1.user_id,u1.first_name from gaj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status LIMIT 0, 50000	50000 rows) returned	0.016 sec / 0.531 sec
16	09:46:35	SELECT username, password FROM gaj.user_details where gender='Male' LIMIT 0, 50000	48741 rows) returned	0.016 sec / 0.469 sec
17	09:47:56	select sum(user_id) FROM gaj.user_details LIMIT 0, 50000	1 rows) returned	0.016 sec / 0.000 sec
18	09:48:39	select * from gaj.user_details order by user_id desc LIMIT 0, 50000	50000 rows) returned	0.500 sec / 0.016 sec
19	09:49:11	select * from gaj.user_details where user_id in (select user_id from user_details where last_name='rogers' and gender='male') LIMIT 0, 50000	1264 rows) returned	0.016 sec / 0.391 sec

Object Info Session

Fifth query

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
13 •    SELECT username, password FROM `gaj`.user_details WHERE gender='Male';
14
15 •    select sum(user_id) FROM `gaj`.user_details;
16
17 •    select * from `gaj`.user_details ORDER BY user_id DESC;
18
19 •    select * from `gaj`.user_details WHERE user_id IN (SELECT user_id FROM user_details WHERE last_name='rogers' AND gender='Male');
20
21 •    SELECT u1.user_id,u1.first_name FROM `gaj`.user_details AS u1,user_details AS u2 WHERE u1.user_id = u2.user_id AND u1.status=u2.status;
```

Below the code, the "Query Statistics" section provides performance details:

- Timing (as measured at client side): Execution time: 0:00:0.360000
- Timing (as measured by the server):
- Errors: 0 errors, 0 warnings, 0
- Rows Processed: Rows affected: 0 Rows examined: 100000 Rows read: 30000
- Temporary Tables: Temporary disk tables created: 0 Temporary tables created: 0

The "Result 15" tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
15	09:45:11	select u1.user_id,u1.first_name FROM `gaj`.user_details AS u1 WHERE u1.user_id = u2.user_id AND u1.status=u2.status LIMIT 0, 50000	50000 rows(s) returned	0.016 sec / 0.531 sec
16	09:45:35	SELECT `username`, `password` FROM `gaj`.`user_details` WHERE `gender`='Male' LIMIT 0, 50000	48741 rows(s) returned	0.019 sec / 0.489 sec
17	09:45:56	select sum(user_id) FROM `gaj`.user_details LIMIT 0, 50000	1 row(s) returned	0.016 sec / 0.000 sec
18	09:46:29	select * from `gaj`.user_details ORDER BY user_id DESC LIMIT 0, 50000	50000 rows(s) returned	0.500 sec / 0.016 sec
19	09:49:11	select * from `gaj`.user_details WHERE user_id IN (SELECT user_id FROM user_details WHERE last_name='rogers' AND gender='Male') LIMIT 0, 50000	1264 rows(s) returned	0.054 sec / 0.331 sec
20	09:50:07	SELECT u1.user_id,u1.first_name FROM `gaj`.user_details AS u1,user_details AS u2 WHERE u1.user_id = u2.user_id AND u1.status=u2.status LIMIT 0, 50000	50000 rows(s) returned	0.016 sec / 0.531 sec

Index

First query

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

- Query Editor:** Contains the following SQL code:

```
18 •    select * from pgj.user_details order by user_id desc;
19 •    select * from pgj.user_details where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male');
20
21 •    select u1.user_id,u1.first_name from pgj.user_details as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
22
23 •    create index ind_username on user_details(username);
24
25 •    SELECT username, password FROM pgj.user_details where gender='Male';
26
27 •    select sum(user_id) FROM pgj.user_details;
```
- Query Statistics:** Shows execution time, errors, rows processed, temporary tables, and other metrics.
- Result Grid:** Displays the results of the query, showing columns like Action, Time, and Action.
- System Bar:** Shows the Windows taskbar with various icons and system status (31°C, Haze, ENG, 09:54, 19-10-2023).

Second query

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

- Query Editor:** Contains the same SQL code as the first query, including the creation of the 'ind_username' index and the 'SELECT' statement.
- Query Statistics:** Shows execution time, errors, rows processed, temporary tables, and other metrics.
- Result Grid:** Displays the results of the query, showing columns like Action, Time, and Action.
- System Bar:** Shows the Windows taskbar with various icons and system status (31°C, Haze, ENG, 09:54, 19-10-2023).

Third query

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

- Query Editor:** Contains the following SQL code:

```
15 • select ul.user_id,ul.first_name from ga1.user_details as ul,user_details as u2 where ul.user_id = u2.user_id and ul.status=u2.status;
22
23 • create index ind_username on user_details(username);
24
25 • SELECT username, password FROM ga1.user_details where gender='Male';
26
27 • select sum(user_id) FROM ga1.user_details;
28
29 • select * from ga1.user_details order by user_id desc;
30
31 • select * from ga1.user_details where user_id in (select user_id from user_details where last_name ='rogers' and gender='male');


```
- Query Statistics:** Shows execution time, errors, rows processed, temporary tables, and other metrics.
- Output Tab:** Displays the results of the query, showing 1 row(s) returned.
- System Bar:** Shows the date and time as 19-10-2023 09:55.

Fourth query

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

- Query Editor:** Contains the same SQL code as the third query, including the creation of an index and the selection of male users.
- Query Statistics:** Shows execution time, errors, rows processed, temporary tables, and other metrics.
- Output Tab:** Displays the results of the query, showing 1 row(s) returned.
- System Bar:** Shows the date and time as 19-10-2023 09:55.

Fifth query

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

- Query Editor:** Contains the following SQL code:

```
25 •    SELECT username, password FROM `gaj`.`user_details` where gender='Male'
26
27 •    select sum(user_id) FROM `gaj`.`user_details`;
28 •    select * from `gaj`.`user_details` order by user_id desc;
29 •    select * from `gaj`.`user_details` where user_id in (select user_id from user_details where last_name = 'rogers' and gender='male');
30
31 •    select u1.user_id,u1.first_name from `gaj`.`user_details` as u1,user_details as u2 where u1.user_id = u2.user_id and u1.status=u2.status;
```
- Information Panel:** Shows Query Statistics, Errors, Row Processing, Temporary Tables, and Index Usage.
- Result Grid:** Displays the execution log with columns: Action, Time, Action, Message, Duration / Fetch.
- System Bar:** Shows the date (19-10-2023), time (09:56), and weather (31°C Haze).

CONCLUSION : Thus we have implemented query optimization.

Jonathan Dsouza
60004220118
Batch - C12

Exp 3 - Implement Query Monitor

SELECT * FROM user_details

ORDER BY first_name;

(0.50212500)

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

- Query Editor:** Contains the SQL statement:

```
1 • SELECT * FROM user_details
2   ORDER BY first_name;
```
- Result Grid:** Displays the results of the query, showing 100000 rows. The columns are: user_id, username, first_name, last_name, gender, password, and status.
- Action Output:** Shows the execution log with 9 entries, including the query itself and its execution time (0.50212500). The log includes messages like "0 row(s) affected" and "100000 row(s) returned".
- System Status:** Shows the system status with various metrics like CPU usage, memory, and disk activity.
- Performance Metrics:** Shows CPU, Memory, and Disk usage over time.

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

- Query Editor:** Contains the SQL statement:

```
1 • SELECT * FROM user_details
2   ORDER BY first_name;
3
```
- Query Statistics:** Provides detailed timing information at the client side and server side, along with error logs, rows processed, temporary tables, and index usage.
- Execution Plan:** Shows the execution plan with 10 steps, including joins, table scans, and sorts.
- Action Output:** Shows the execution log with 9 entries, including the query itself and its execution time (0.50212500). The log includes messages like "39 row(s) returned" and "100000 row(s) returned".
- System Status:** Shows the system status with various metrics like CPU usage, memory, and disk activity.
- Performance Metrics:** Shows CPU, Memory, and Disk usage over time.

MySQL Workbench

unconnected localhost

File Edit View Query Database Server Tools Scripting Help

Navigator Administration Schemas Information

No object selected

Visual Explain Display Info Read + Eval cost Overview View Source

Query cost: 118442.29

```

    graph TD
        A[query_block #1] --> B[ORDER]
        B --> C[118442.29 rows Full Table Scan user_details]
    
```

user_details 1

Action Output

#	Time	Action	Message	Duration / Fetch
4	15:18:45	show tables	39 row(s) returned	0.000 sec / 0.000 sec
5	15:18:04	select * from user_details	100000 row(s) returned	0.015 sec / 0.516 sec
6	15:18:50	SELECT user_id, first_name FROM user_details where gender='Male'	48741 row(s) returned	0.000 sec / 0.453 sec
7	15:22:55	select COUNT(user_id) FROM user_details	1 row(s) returned	0.016 sec / 0.000 sec
8	15:22:17	SELECT * FROM user_details ORDER BY first_name	100000 row(s) returned	0.531 sec / 0.078 sec
9	15:24:47	EXPLAIN SELECT * FROM user_details ORDER BY first_name	OK	0.000 sec

Object Info Session

32°C Haze 15:26 10-10-2023

MySQL Workbench

unconnected localhost

File Edit View Query Database Server Tools Scripting Help

Navigator Administration Schemas Information

No object selected

Form Editor Navigator: 1 / 100001

User_id:	99681
Username:	chrshaydon2
First_name:	bel
Last_name:	david
Gender:	Female

user_details 1

Action Output

#	Time	Action	Message	Duration / Fetch
4	15:18:45	show tables	39 row(s) returned	0.000 sec / 0.000 sec
5	15:18:04	select * from user_details	100000 row(s) returned	0.015 sec / 0.516 sec
6	15:18:50	SELECT user_id, first_name FROM user_details where gender='Male'	48741 row(s) returned	0.000 sec / 0.453 sec
7	15:22:55	select COUNT(user_id) FROM user_details	1 row(s) returned	0.016 sec / 0.000 sec
8	15:22:17	SELECT * FROM user_details ORDER BY first_name	100000 row(s) returned	0.531 sec / 0.078 sec
9	15:24:47	EXPLAIN SELECT * FROM user_details ORDER BY first_name	OK	0.000 sec

Object Info Session

32°C Haze 15:26 10-10-2023

Conclusion - Thus we have successfully implemented Query Monitor.

JONATHAN DSOUZA
60004220118
COMPS A
C12

Exp - 4 - To implement operations like searching, inserting and deleting in B trees and B+ trees.

B trees

```
from BTrees.IIBTree import
IIBTreeimport time
bt = IIBTree()
insertion_start_time =
time.time()for i in
range(1000):
    bt.update({i: 2*i})
insertion_end_time =
time.time()print(
    f"Insertion time:
{round((insertion_end_time-
insertion_start_time)*1000,3)}")
milliseconds") key = int(input("enter key:
"))
search_start_time =
time.time()if
bt.has_key(key):
    print(bt[key])
search_end_time =
time.time()print(
    f"Search time: {round((search_end_time-
```

```
PS C:\Users\devan\OneDrive\Desktop\College\Sem 5 Docs>
Insertion time: 1.56 milliseconds
enter key: 56
112
Search time: 1.008 milliseconds
```

B+ Trees:

```
from bplustree import
BPlusTreeimport time

tree = BPlusTree('D:/b1.db',
order=50)for i in range(1000):
    data = (2*i).to_bytes(10,
    'big')tree[i] = data
data = int(input("Enter key :
"))start_time = time.time()
byte_data = tree.get(data)
end_time = time.time()
int_data = int.from_bytes(byte_data,
'big')print("Value : ", int_data)
```

```
PS C:\Users\devan\OneDrive\Desktop\College\Sem 5 Docs> python -u "c:\Us
le.py"
Found an existing WAL file, the B+Tree was not closed properly
Enter key : 103
Value : 206
Time taken : 1.1188983917236328 ms
PS C:\Users\devan\OneDrive\Desktop\College\Sem 5 Docs> █
```

CONCLUSION - Thus we have implemented B and B+ trees.

JONATHAN

DSOUZA

60004220118

COMPS -A

C12

EXP - 5 - Perform Fragmentation (Range, List, Hash and Key)

```
CREATE TABLE Employees_range
```

```
    (ID INT,  
     Name VARCHAR(50),  
     Position VARCHAR(50),  
     Department  
     VARCHAR(50),Salary INT
```

```
);
```

```
INSERT INTO Employees_range (ID, Name, Position, Department, Salary)
```

```
VALUES
```

```
(1, 'John Doe', 'Manager', 'Sales', 70000),  
(2, 'Jane Smith', 'Assistant Manager', 'Sales', 60000),  
(3, 'Mike Johnson', 'Sales Representative', 'Sales',  
50000),(4, 'Alice Williams', 'Manager', 'Marketing',  
80000),  
(5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000),  
(6, 'Charlie Davis', 'Marketing Specialist', 'Marketing', 60000),  
(7, 'David Miller', 'Manager', 'HR', 75000),  
(8, 'Eva Wilson', 'Assistant Manager', 'HR',  
65000),(9, 'Frank Moore', 'HR Specialist', 'HR',  
55000), (10, 'Grace Taylor', 'Manager', 'Finance',  
85000),  
(11, 'Henry Anderson', 'Assistant Manager', 'Finance',  
75000),(12, 'Irene Thomas', 'Finance Specialist', 'Finance',  
65000), (13, 'Jack Jackson', 'Manager', 'IT', 90000),  
(14, 'Kelly White', 'Assistant Manager', 'IT',  
80000),(15, 'Larry Harris', 'IT Specialist', 'IT',  
70000),  
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
```

```

(17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 70000),
(18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
(19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
(20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);

ALTER TABLE Employees_range PARTITION BY RANGE (ID) (
    PARTITION p0 VALUES LESS THAN (6),
    PARTITION p1 VALUES LESS THAN (11),
    PARTITION p2 VALUES LESS THAN (16),
    PARTITION p3 VALUES LESS THAN MAXVALUE
);

```

```

SELECT * FROM Employees_range PARTITION (p0);
SELECT * FROM Employees_range PARTITION (p1);
SELECT * FROM Employees_range PARTITION (p2);
SELECT * FROM Employees_range PARTITION (p3);

```

A screenshot of a database result grid titled "Result Grid". The grid has columns: ID, Name, Position, Department, and Salary. There are five rows of data. Row 1: ID 1, Name John Doe, Position Manager, Department Sales, Salary 70000. Row 2: ID 2, Name Jane Smith, Position Assistant Manager, Department Sales, Salary 60000. Row 3: ID 3, Name Mike Johnson, Position Sales Representative, Department Sales, Salary 50000. Row 4: ID 4, Name Alice Williams, Position Manager, Department Marketing, Salary 80000. Row 5: ID 5, Name Bob Brown, Position Assistant Manager, Department Marketing, Salary 70000.

	ID	Name	Position	Department	Salary
▶	1	John Doe	Manager	Sales	70000
	2	Jane Smith	Assistant Manager	Sales	60000
	3	Mike Johnson	Sales Representative	Sales	50000
	4	Alice Williams	Manager	Marketing	80000
	5	Bob Brown	Assistant Manager	Marketing	70000

```
CREATE TABLE Employees_key2
(
    ID INT,
    Name VARCHAR(50),
    Position VARCHAR(50),
    Department
    VARCHAR(50),
    Salary INT,
    PRIMARY KEY(ID)
);

INSERT INTO Employees_key2 (ID, Name, Position, Department, Salary)
VALUES
    (1, 'John Doe', 'Manager', 'Sales', 70000),
    (2, 'Jane Smith', 'Assistant Manager', 'Sales', 60000),
    (3, 'Mike Johnson', 'Sales Representative', 'Sales',
    50000),
    (4, 'Alice Williams', 'Manager', 'Marketing',
    80000),
    (5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000),
    (6, 'Charlie Davis', 'Marketing Specialist', 'Marketing', 60000),
    (7, 'David Miller', 'Manager', 'HR', 75000),
    (8, 'Eva Wilson', 'Assistant Manager', 'HR',
    65000),
    (9, 'Frank Moore', 'HR Specialist', 'HR',
    55000),
    (10, 'Grace Taylor', 'Manager', 'Finance',
    85000),
    (11, 'Henry Anderson', 'Assistant Manager', 'Finance',
    75000),
    (12, 'Irene Thomas', 'Finance Specialist', 'Finance',
    65000),
    (13, 'Jack Jackson', 'Manager', 'IT', 90000),
    (14, 'Kelly White', 'Assistant Manager', 'IT',
    80000),
    (15, 'Larry Harris', 'IT Specialist', 'IT',
    70000),
    (16, 'Molly Clark', 'Manager', 'Operations', 80000),
    (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 70000),
```

```

(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE Employees_key2 PARTITION BY KEY()
PARTITIONS 4;

```

```

SELECT * FROM Employees_key2 PARTITION (p0);
SELECT * FROM Employees_key2 PARTITION (p1);
SELECT * FROM Employees_key2 PARTITION (p2);
SELECT * FROM Employees_key2 PARTITION (p3);

```

	ID	Name	Position	Department	Salary
▶	1	John Doe	Manager	Sales	70000
	5	Bob Brown	Assistant Manager	Marketing	70000
	9	Frank Moore	HR Specialist	HR	55000
	13	Jack Jackson	Manager	IT	90000
*	17	Nancy Thompson	Assistant Manager	Operations	70000
	NULL	NULL	NULL	NULL	NULL

```

CREATE TABLE Employees_hash
(
ID INT,
Name VARCHAR(50),
Position VARCHAR(50),
Department
VARCHAR(50),Salary INT
);
INSERT INTO Employees_hash (ID, Name, Position, Department,
Salary)VALUES
(1, 'John Doe', 'Manager', 'Sales', 70000),
(2, 'Jane Smith', 'Assistant Manager', 'Sales', 60000),
(3, 'Mike Johnson', 'Sales Representative', 'Sales',
50000),(4, 'Alice Williams', 'Manager', 'Marketing',
80000),

```

```

(5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000),
(6, 'Charlie Davis', 'Marketing Specialist', 'Marketing', 60000),
(7, 'David Miller', 'Manager', 'HR', 75000),
(8, 'Eva Wilson', 'Assistant Manager', 'HR',
65000),(9, 'Frank Moore', 'HR Specialist', 'HR',
55000), (10, 'Grace Taylor', 'Manager', 'Finance',
85000),
(11, 'Henry Anderson', 'Assistant Manager', 'Finance',
75000),(12, 'Irene Thomas', 'Finance Specialist', 'Finance',
65000), (13, 'Jack Jackson', 'Manager', 'IT', 90000),
(14, 'Kelly White', 'Assistant Manager', 'IT',
80000),(15, 'Larry Harris', 'IT Specialist', 'IT',
70000),
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
(17, 'Nancy Thompson', 'Assistant Manager','Operations' ,70000),
(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE Employees_hash PARTITION BY HASH(ID)
PARTITIONS 4;

```

```

SELECT * FROM Employees_hash PARTITION
(p0);SELECT * FROM Employees_hash PARTITION
(p1);SELECT * FROM Employees_hash PARTITION
(p2);SELECT * FROM Employees_hash PARTITION
(p3);

```



The screenshot shows a database query results grid with the following data:

	ID	Name	Position	Department	Salary
▶	4	Alice Williams	Manager	Marketing	80000
	8	Eva Wilson	Assistant Manager	HR	65000
	12	Irene Thomas	Finance Specialist	Finance	65000
	16	Molly Clark	Manager	Operations	80000
	20	Robert Robinson	Customer Service Representative	Customer Service	50000

```

CREATE TABLE Employees_list4
(
    ID INT,
    Name VARCHAR(50),
    Position VARCHAR(50),
    Department
    VARCHAR(50),
    Salary INT
);

INSERT INTO Employees_list4 (ID, Name, Position, Department, Salary)
VALUES
    (1, 'John Doe', 'Manager', 'Sales', 70000),
    (2, 'Jane Smith', 'Assistant Manager', 'Sales', 60000),
    (3, 'Mike Johnson', 'Sales Representative', 'Sales',
    50000),
    (4, 'Alice Williams', 'Manager', 'Marketing',
    80000),
    (5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000),
    (6, 'Charlie Davis', 'Marketing Specialist', 'Marketing', 60000),
    (7, 'David Miller', 'Manager', 'HR', 75000),
    (8, 'Eva Wilson', 'Assistant Manager', 'HR',
    65000),
    (9, 'Frank Moore', 'HR Specialist', 'HR',
    55000),
    (10, 'Grace Taylor', 'Manager', 'Finance',
    85000),
    (11, 'Henry Anderson', 'Assistant Manager', 'Finance',
    75000),
    (12, 'Irene Thomas', 'Finance Specialist', 'Finance',
    65000),
    (13, 'Jack Jackson', 'Manager', 'IT', 90000),
    (14, 'Kelly White', 'Assistant Manager', 'IT',
    80000),
    (15, 'Larry Harris', 'IT Specialist', 'IT',
    70000),
    (16, 'Molly Clark', 'Manager', 'Operations', 80000),
    (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 70000),
    (18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
    (19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
    (20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);
ALTER TABLE Employees_list4 PARTITION BY LIST (ID) (

```

```

PARTITION p0 VALUES IN (1,3,5,7,17),
PARTITION p1 VALUES IN (2,4,6,8,18),
PARTITION p2 VALUES IN (9,11,13,15,19),
PARTITION p3 VALUES IN (10,12,14,16,20)

);

```

```

SELECT * FROM Employees_list4 PARTITION
(p0);SELECT * FROM Employees_list4 PARTITION
(p1);SELECT * FROM Employees_list4 PARTITION
(p2);SELECT * FROM Employees_list4 PARTITION
(p3);

```

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

	ID	Name	Position	Department	Salary
▶	9	Frank Moore	HR Specialist	HR	55000
	11	Henry Anderson	Assistant Manager	Finance	75000
	13	Jack Jackson	Manager	IT	90000
	15	Larry Harris	IT Specialist	IT	70000
	19	Patricia Martinez	Manager	Customer Service	75000

Conclusion - Thus we have implemented fragmentation.

JONATHAN DSOUZA
60004220118
Comps A
C12

Exp - 6 - Implementation of two phase protocol.

Code. -

```
import time
import random

class Coordinator:
    def __init__(self, participants):
        self.participants = participants

    def send_prepare(self):
        # Phase 1: Prepare
        votes = []
        for participant in self.participants:
            try:
                vote = participant.prepare()
                votes.append(vote)
            except Exception as e:
                print(f"Error during prepare phase: {str(e)}")
                self.send_abort()

        # If all participants agree to commit, proceed to Phase 2: Commit
        if all(votes):
            self.send_commit()
        else:
            self.send_abort()

    def send_commit(self):
        # Phase 2: Commit
        for participant in self.participants:
            try:
                participant.commit()
            except Exception as e:
                print(f"Error during commit phase: {str(e)}")
                self.send_abort()

    def send_abort(self):
        # Abort in case of any failure during prepare or commit phases
        for participant in self.participants:
            try:
                participant.abort()
            except Exception as e:
                print(f"Error during abort phase: {str(e)}")

class Participant:
    def __init__(self, name):
        self.name = name

    def prepare(self):
        print(f"{self.name}: Prepare phase")
```

```
# Simulate prepare actions
time.sleep(random.uniform(0.5, 1.5))
# For simplicity, let's assume the prepare phase always succeeds
return True

def commit(self):
    print(f"{self.name}: Commit phase")
    # Simulate commit actions
    time.sleep(random.uniform(0.5, 1.5))
    # For simplicity, let's assume the commit phase always succeeds
    pass

def abort(self):
    print(f"{self.name}: Abort phase")
    # Simulate abort actions
    time.sleep(random.uniform(0.5, 1.5))
    # For simplicity, let's assume the abort phase always succeeds
    pass

if __name__ == "__main__":
    # Example usage
    participant1 = Participant("Participant 1")
    participant2 = Participant("Participant 2")

    coordinator = Coordinator([participant1, participant2])

    try:
        # Phase 1: Prepare
        coordinator.send_prepare()

        print("Transaction committed successfully!")

    except Exception as e:
        # If any participant aborts or an error occurs, abort the transaction
        print(f"Transaction aborted: {str(e)}")
```

OUTPUT -

```
see.py > Coordinator > send_prepare
1 import time
2 import random
3
4
5 class Coordinator:
6     def __init__(self, participants):
7         self.participants = participants
8
9     def send_prepare(self):
10        # Phase 1: Prepare
11        votes = []
12        for participant in self.participants:
13            try:
14                vote = participant.prepare()
15                votes.append(vote)
16            except Exception as e:
17                print(f"Error during prepare phase: {str(e)}")
18                self.send_abort()
19
20        # If all participants agree to commit, proceed to Phase 2: Commit
21        if all(votes):
22            ...
23
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● jonathandsouza@Jonathans-MacBook-Air vscode % /usr/bin/python3 /Users/jonathandsouza/Documents/vscode/se
e.py
Participant 1: Prepare phase
Participant 2: Prepare phase
Participant 1: Commit phase
Participant 2: Commit phase
Transaction committed successfully!
○ jonathandsouza@Jonathans-MacBook-Air vscode %
```

CONCLUSION - Thus we have implemented two phase protocol.

JONATHAN DSOUZA

60004220118

COMPS A

C12

EXP 7 - Query Execution on an XML database

```
<html>
<head>
<style>
table, th, td {
border: 1px solid black;
border-collapse: collapse;
margin: 5px;
}
th, td {
padding: 5px;
}
input {
margin-bottom: 5px;
}
</style>
</head>
<body>
<button type="button" onclick="loadXMLDoc()">View Information
about the Books</button>
<br><br>
<div>
<label>
Search a title
</label>
<input type="text" name="title"
onchange="performQuery(event)"/>
</div>
<div>
<label>
Search an author
</label>
<input type="text" name="author"
onchange="performQuery(event)"/>
</div>
<div>
<label>
Search a genre
</label>
<input type="text" name="genre"
onchange="performQuery(event)"/>
</div>
```

```
<div>
<label>
Search a price
</label>
<input type="text" name="price"
onchange="performQuery(event)"/>
</div>
<div>
<label>
Search a Publish Date
</label>
<input type="text" name="publish_date"
onchange="performQuery(event)"/>
</div>
<table id="data-table"></table>
<h2>Result</h2>
<table id="query-table"></table>
    var arrayObj = [];
function
loadXMLDoc() {
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
myFunction(this);
}
};
xmlhttp.open("GET", "books.xml", true); xmlhttp.send();
}
function
myFunction(xml) {
var i;
var xmlDoc =
xml.responseXML; var
table=<tr><th>Title</th><th>Author</th><th>Genre</th><th>Price</th><th> Publish
Date</th><th>Description</th></tr>";
var x = xmlDoc.getElementsByTagName("book");
for (i = 0; i <x.length; i++) {
table += "<tr><td>" + x[i].getElementsByTagName("title")
[0].childNodes[0].nodeValue + "</td><td>" + x[i].getElementsByTagName("author")
[0].childNodes[0].nodeValue + "</td><td>" +
x[i].getElementsByTagName("genre")
[0].childNodes[0].nodeValue + "</td><td>" +
x[i].getElementsByTagName("price") [0].childNodes[0].nodeValue +
"</td><td>" +
x[i].getElementsByTagName("publish_date")
[0].childNodes[0].nodeValue + "</td><td>" +
```

```
x[i].getElementsByTagName("description")
[0].childNodes[0].nodeValue
+
"</td></tr>";
arrayObj.push({ title:
x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue,
author:
x[i].getElementsByTagName("author")
[0].childNodes[0].nodeValue, genre:
x[i].getElementsByTagName("genre")[0].childNodes[0].nodeValue,
price:
x[i].getElementsByTagName("price")[0].childNodes[0].nodeValue,
publish_date:
x[i].getElementsByTagName("publish_date")
[0].childNodes[0].nodeValue, description:
x[i].getElementsByTagName("description")
[0].childNodes[0].nodeValue
});
}
document.getElementById("data-table").innerHTML = table;
console.log(arrayObj);
}

function performQuery(event) {
var res = arrayObj.find((obj) =>
obj[event.target.name].toLowerCase().includes(event.target.value.toLowerCase()));
console.log(res); if (res)
{ var queryTable=<tr><th>Title</th><th>Author</th><th>Genre</th><th>Price</th><th>Publish Date</th><th>Description</th></tr>;
queryTable += "<tr><td>" + res.title +
"</td><td>" + res.author +
"</td><td>" + res.genre +
"</td><td>" + res.price +
"</td><td>" + res.publish_date +
"</td><td>" + res.description +
"</td></tr>";}
document.getElementById("query-table").innerHTML = queryTable;
} else {
alert("No result found!")
}
</script>
</body>
</html>
```

XML -

```
<?xml version="1.0"?>
<catalog>
<book id="bk101">
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<genre>Computer</genre>
<price>44.95</price>
<publish_date>2000-10-01</publish_date>
<description>An in-depth look at creating applications with
XML.</description>
</book>
<book id="bk102">
<author>Ralls, Kim</author>
<title>Midnight Rain</title>
<genre>Fantasy</genre>
<price>5.95</price>
<publish_date>2000-12-16</publish_date>
<description>A former architect battles corporate zombies, an
evil sorceress, and her
own childhood to become queen of the world.</description>
</book>
<book id="bk103">
<author>Corets, Eva</author>
<title>Maeve Ascendant</title>
<genre>Fantasy</genre>
<price>5.95</price>
<publish_date>2000-11-17</publish_date>
<description>After the collapse of a nanotechnology society in
England, the young
survivors lay the foundation for a new society.</description>
</book>
<book id="bk104">
<author>Corets, Eva</author>
<title>Oberon's Legacy</title>
<genre>Fantasy</genre>
<price>5.95</price>
<publish_date>2001-03-10</publish_date>
<description>In post-apocalypse England, the mysterious agent
known only as
Oberon helps to create a new life for the inhabitants of
London. Sequel to Maeve
Ascendant.</description>
</book>
<book id="bk105">
<author>Corets, Eva</author>
<title>The Sundered Grail</title>
<genre>Fantasy</genre>
```

```
<price>5.95</price>
<publish_date>2001-09-10</publish_date>
<description>The two daughters of Maeve, half-sisters, battle
one another for control
of England. Sequel to Oberon's Legacy.</description>
</book>
<book id="bk106">
<author>Randall, Cynthia</author>
<title>Lover Birds</title>
<genre>Romance</genre>
<price>4.95</price>
<publish_date>2000-09-02</publish_date>
<description>When Carla meets Paul at an ornithology
conference, tempers fly as feathers get
ruffled.</description>
</book>
<book id="bk107">
<author>Thurman, Paula</author>
<title>Splish Splash</title>
<genre>Romance</genre>
<price>4.95</price>
<publish_date>2000-11-02</publish_date>
<description>A deep sea diver finds true love twenty thousand
leagues beneath the
sea.</description>
</book>
<book id="bk108">
<author>Knorr, Stefan</author>
<title>Creepy Crawlies</title>
<genre>Horror</genre>
<price>4.95</price>
<publish_date>2000-12-06</publish_date>
<description>An anthology of horror stories about roaches,
centipedes, scorpions
and other insects.</description>
</book>
<book id="bk109">
<author>Kress, Peter</author>
<title>Paradox Lost</title>
<genre>Science Fiction</genre>
<price>6.95</price>
<publish_date>2000-11-02</publish_date>
<description>After an inadvertant trip through a Heisenberg
Uncertainty Device, James Salway
discovers the problems
of being quantum.</description>
</book>
<book id="bk110">
<author>O'Brien, Tim</author>
```

```
<title>Microsoft .NET: The Programming Bible</title>
<genre>Computer</genre>
<price>36.95</price>
<publish_date>2000-12-09</publish_date>
<description>Microsoft's .NET initiative is explored in detail
in this deep
programmer's reference.</description>
</book>
<book id="bk111">
<author>O'Brien, Tim</author>
<title>MSXML3: A Comprehensive Guide</title>
<genre>Computer</genre>
<price>36.95</price>
<publish_date>2000-12-01</publish_date>
<description>The Microsoft MSXML3 parser is covered in
detail, with attention to XML DOM interfaces, XSLT processing,
SAX and more.</description>
</book>
<book id="bk112">
<author>Galos, Mike</author>
<title>Visual Studio 7: A Comprehensive Guide</title>
<genre>Computer</genre>
<price>49.95</price>
<publish_date>2001-04-16</publish_date>
<description>Microsoft Visual Studio 7 is explored in depth,
looking at how Visual Basic,
Visual C++, C#, and ASP+ are integrated into a comprehensive
development
environment.</description>
</book>
</catalog>
```

Output -

[View Information about the Books](#)

Search a title

Search an author

Search a genre

Search a price

Search a Publish Date

Title	Author	Genre	Price	Publish Date	Description
XML Developer's Guide	Gambardella, Matthew	Computer	44.95	2000-10-01	An in-depth look at creating applications with XML.
Midnight Rain	Ralls, Kim	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.
Maeve Ascendant	Corets, Eva	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology society in England, the young Maeve must travel to the North to begin her quest for life and freedom.
Oberon's Legacy	Corets, Eva	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious agent known only as Oberon must travel to the North to begin her quest for life and freedom.
The Sundered Grail	Corets, Eva	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, battle one another for control of England.
Lover Birds	Randall, Cynthia	Romance	4.95	2000-09-02	When Carla meets Paul at an ornithology conference, tempers fly as frantically as the birds they study.
Splish Splash	Thurman, Paula	Romance	4.95	2000-11-02	A deep sea diver finds true love twenty thousand leagues beneath the surface.
Creepy Crawlies	Knorr, Stefan	Horror	4.95	2000-12-06	An anthology of horror stories about roaches, centipedes, scorpions and other insects.
Paradox Lost	Kress, Peter	Science Fiction	6.95	2000-11-02	After an inadvertant trip through a Heisenberg Uncertainty Device, James Kress' Peter has to figure out how to get back.
Microsoft .NET: The Programming Bible	O'Brien, Tim	Computer	36.95	2000-12-09	Microsoft's .NET initiative is explored in detail in this deep programming guide.
MSXML3: A Comprehensive Guide	O'Brien, Tim	Computer	36.95	2000-12-01	The Microsoft MSXML3 parser is covered in detail, with attention to the XML DOM.
Visual Studio 7: A Comprehensive Guide	Galos, Mike	Computer	49.95	2001-04-16	Microsoft Visual Studio 7 is explored in depth, looking at how Visual Studio can be used to build Windows and Web-based applications.

Result

Title	Author	Genre	Price	Publish Date	Description
Midnight Rain	Ralls, Kim	Fantasy	5.95	2000-12-16	A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Conclusion – Thus we have successfully implemented queries on a XML database.

JONATHAN DSOUZA
60004220118

COMPS A
C12

EXP 8 - Data Handling using JSON

Code:

```
const url = 'https://dummyjson.com/users';

fetch(url)
  .then(response => {
    if (response.ok) {
      return response.json();
    } else {
      throw new Error(`Error: ${response.status}`);
    }
  })
  .then(apiResponse => {
    const data = apiResponse.users;
    //Print all users
    console.log('All users:', data);

    const filteredUsers = data.filter(user => user.age > 25);
    const usernamesOver25 = filteredUsers.map(user => user.firstName);
    console.log('Users over 25:', usernamesOver25);

    const usersFromCity = data.filter(user => user.address.state === 'CA');
    const usersFromCA = usersFromCity.map(user => user.firstName);
    console.log('Users from California:', usersFromCA);

    const averageAge = data.reduce((sum, user) => sum + user.age, 0) / data.length;
    console.log('Average age of users:', averageAge);

    const oldestUser = data.reduce((oldest, user) => (user.age > oldest.age ? user : oldest), data[0]);
    console.log('Oldest user:', oldestUser.firstName);
```

```
    })
  .catch(error => {
    console.error(error.message);
});
```

Output -

```
Users over 25: [
  'Terry',    'Sheldon',  'Terrill',
  'Miles',    'Mavis',    'Oleta',
  'Ewell',    'Eleanora', 'Marcel',
  'Assunta',  'Trace',   'Jeanne',
  'Trycia',   'Bradford', 'Arely',
  'Gust',     'Lenna',    'Tressa',
  'Felicity', 'Griffin',  'Piper',
  'Kody',     'Macy',    'Maurine'
]
Users from California: [ 'Terrill', 'Oleta', 'Lenna',
  'Macy' ]
Average age of users: 34.86666666666667
Oldest user: Terry
```

Conclusion -

Thus we have successfully handled data using JSON.

JONATHAN DSOUZA
60004220118

COMPS A

C12

EXP 9 - Processing of Spatial and Temporal data.

Code:

Data

start_date	start_time	distance	mode	confidence	Places
02-07-2022		18716	IN_PASSENGER_VEHICLE	MEDIUM	
06-07-2022	03:23:42	1669	MOTORCYCLING	LOW	
06-07-2022	03:35:46	11483	IN_TRAIN	LOW	
06-07-2022	03:58:13	1069	MOTORCYCLING	MEDIUM	
08-07-2022	03:11:39	461	MOTORCYCLING	LOW	
08-07-2022	03:26:15	1099	IN_PASSENGER_VEHICLE	LOW	
08-07-2022	03:42:42	7548	IN_TRAIN	MEDIUM	Anhad Misal
11-07-2022	03:48:22	1609	IN_BUS	MEDIUM	Kandivali Bus Station (W)
11-07-2022	04:11:17	9363	IN_TRAIN	MEDIUM	Agarkar Chowk
11-07-2022	04:39:15	2125	IN_SUBWAY	LOW	SAMMOHI BY MOKSHA & HIRAL
11-07-2022	04:48:18	896	IN_PASSENGER_VEHICLE	LOW	Trumpet Sky Lounge
11-07-2022	12:22:26	2972	IN_PASSENGER_VEHICLE	LOW	McDonald's
11-07-2022	12:48:32	9467	IN_TRAIN	MEDIUM	Kandivali Bus Depot
11-07-2022	13:10:00	2283	IN_PASSENGER_VEHICLE	LOW	State Bank of India
13-07-2022	03:19:59	905	WALKING	LOW	
13-07-2022	03:48:02	1896	IN_BUS	LOW	Kandivali Railway Station (W)
13-07-2022	04:04:12	9348	IN_TRAIN	MEDIUM	Agarkar Chowk / Pinky Cinema
13-07-2022	04:33:49	3270	IN_PASSENGER_VEHICLE	LOW	Lotus Business Park
13-07-2022	13:06:26	5297	IN_BUS	LOW	Villa Decor - Premium Bed, Bath & Mattresses Store
13-07-2022	13:44:17	11855	IN_TRAIN	LOW	Kandivali Station (W)
13-07-2022	14:15:27	1969	IN_PASSENGER_VEHICLE	MEDIUM	State Bank of India
14-07-2022	03:54:54	8949	IN_BUS	HIGH	
14-07-2022	04:39:02	578	WALKING	MEDIUM	UK Realty
14-07-2022	14:12:01	6005	IN_BUS	LOW	Yoko Sizzlers
14-07-2022	14:58:50	4087	UNKNOWN_ACTIVITY_TYPE	LOW	Xth Central Mall
15-07-2022	03:48:49	8500	IN_PASSENGER_VEHICLE	LOW	
15-07-2022	04:41:11	9185	IN_BUS	LOW	RikÄ? - Terrace Bar & Grill
15-07-2022	14:15:09	5691	IN_BUS	MEDIUM	D Mart
15-07-2022	15:08:26	4933	IN_BUS	MEDIUM	La Pino'z Pizza Kandivali
16-07-2022	03:44:35	9838	IN_BUS	MEDIUM	Laxmi Industrial Colony
16-07-2022	13:08:07	2965	MOTORCYCLING	LOW	Andheri Station (W)
16-07-2022	13:32:44	9292	IN_TRAIN	MEDIUM	Kandivali West
16-07-2022	13:53:02	2074	IN_PASSENGER_VEHICLE	LOW	La Pino'z Pizza Kandivali
17-07-2022	08:38:08	880	MOTORCYCLING	LOW	Malad Industrial Estate
17-07-2022	10:23:08	1492	IN_PASSENGER_VEHICLE	LOW	Aadhar Center
17-07-2022	10:57:46	1294	MOTORCYCLING	MEDIUM	La Pino'z Pizza Kandivali
17-07-2022	11:38:53	5829	MOTORCYCLING	LOW	State Bank Of India ATM
18-07-2022	03:51:31	1292	IN_BUS	LOW	
18-07-2022	03:57:47	9400	IN_BUS	LOW	Anhad Misal

Queries -

- a. select * from 2022_july where start_time < "04:00:00";
- b. select * from 2022_july where distance > 10000;
- c. select * from 2022_july where Places = "Doolally Taproom - Andheri";

Output:

Temporal

start_date	start_time	distance	mode	confidence	Places
02-07-2022		18716	IN_PASSENGER_VEHICLE	MEDIUM	
06-07-2022	03:23:42	1669	MOTORCYCLING	LOW	
06-07-2022	03:35:46	11483	IN_TRAIN	LOW	
06-07-2022	03:58:13	1069	MOTORCYCLING	MEDIUM	
08-07-2022	03:11:39	461	MOTORCYCLING	LOW	
08-07-2022	03:26:15	1099	IN_PASSENGER_VEHICLE	LOW	
08-07-2022	03:42:42	7548	IN_TRAIN	MEDIUM	Anhad Misal
11-07-2022	03:48:22	1609	IN_BUS	MEDIUM	Kandivali Bus Station (W)
13-07-2022	03:19:59	905	WALKING	LOW	
13-07-2022	03:48:02	1896	IN_BUS	LOW	Kandivali Railway Station (W)
14-07-2022	03:54:54	8949	IN_BUS	HIGH	
15-07-2022	03:48:49	8500	IN_PASSENGER_VEHICLE	LOW	
16-07-2022	03:44:35	9838	IN_BUS	MEDIUM	Laxmi Industrial Colony
18-07-2022	03:51:31	1292	IN_BUS	LOW	
18-07-2022	03:57:47	9400	IN_BUS	LOW	Anhad Misal
19-07-2022	03:54:30	10085	IN_BUS	MEDIUM	Lotus Business Park
20-07-2022	03:32:29	1721	IN_BUS	MEDIUM	Bajaj Municipal School
20-07-2022	03:47:40	9487	IN_TRAIN	MEDIUM	Andheri Police Station
21-07-2022	03:46:02	8698	UNKNOWN_ACTIVITY_TYPE	LOW	Sun-n-Sand Hotel
23-07-2022	03:45:05	11590	IN_PASSENGER_VEHICLE	LOW	
25-07-2022	03:34:52	10629	IN_BUS	MEDIUM	Doolally Taproom - Andheri
26-07-2022	03:52:48	10072	IN_BUS	MEDIUM	Doolally Taproom - Andheri
27-07-2022	03:48:55	9483	IN_BUS	LOW	Doolally Taproom - Andheri
28-07-2022	03:29:48	9599	IN_BUS	MEDIUM	RikÄ? - Terrace Bar & Grill
29-07-2022	03:31:35	1948	MOTORCYCLING	LOW	Kandivali Bus Station (W)
29-07-2022	03:47:34	9579	IN_TRAIN	MEDIUM	

Spatial

start_date	start_time	distance	mode	confidence	Places
02-07-2022		18716	IN_PASSENGER_VEHICLE	MEDIUM	
06-07-2022	03:35:46	11483	IN_TRAIN	LOW	
13-07-2022	13:44:17	11855	IN_TRAIN	LOW	Kandivali Station (W)
19-07-2022	03:54:30	10085	IN_BUS	MEDIUM	Lotus Business Park
21-07-2022	18:59:35	41236	UNKNOWN_ACTIVITY_TYPE	LOW	Jay's Dosa & Paratha Center
23-07-2022	03:45:05	11590	IN_PASSENGER_VEHICLE	LOW	
24-07-2022	06:56:09	10674	MOTORCYCLING	LOW	Axis Bank ATM
25-07-2022	03:34:52	10629	IN_BUS	MEDIUM	Doolally Taproom - Andheri
26-07-2022	03:52:48	10072	IN_BUS	MEDIUM	Doolally Taproom - Andheri

```
mysql> select * from 2022_july where Places="Doolally Taproom - Andheri";
+-----+-----+-----+-----+-----+
| start_date | start_time | distance | mode   | confidence | Places
+-----+-----+-----+-----+-----+
| 25-07-2022 | 03:34:52  |    10629 | IN_BUS | MEDIUM    | Doolally Taproom - Andheri |
| 26-07-2022 | 03:52:48  |    10072 | IN_BUS | MEDIUM    | Doolally Taproom - Andheri |
| 27-07-2022 | 03:48:55  |     9483 | IN_BUS | LOW       | Doolally Taproom - Andheri |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Conclusion – Thus we have successfully implemented the processing of spatial and temporal data.