

Experiment No 2 - Query Optimization

Simple Query

The screenshot displays three separate sessions in MySQL Workbench, each showing the execution of the same SQL query. The query is:

```

1 USE kreenashah;
2 SELECT * FROM user_details;
3
4 SELECT COUNT(*) FROM user_details;
5 SELECT AVG(user_id) FROM user_details;
6
7 SELECT * FROM user_details ORDER BY last_name DESC;
8
9 SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '%2%');
10
11 SELECT u1.username AS user1_username, u2.username AS user2_username
12 FROM user_details u1
13 JOIN user_details u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id;
    
```

Session 1 (Top):

- Time (as measured at client side): Execution time: 0.0000000000000000
- Time (as measured by the server): Execution time: 0.0000000000000000
- Table lock wait time: 0.0000000000000000
- Errors: 0 Warnings: 0
- Rows processed: 0 Rows affected: 0 Rows inserted: 0 Rows updated: 286533
- Temporary Tables: 0 Temporary tables created: 0
- Other Info: Thread Id: 53

Session 2 (Middle):

- Time (as measured at client side): Execution time: 0.0000000000000000
- Time (as measured by the server): Execution time: 0.0000000000000000
- Table lock wait time: 0.0000000000000000
- Errors: 0 Warnings: 0
- Rows processed: 0 Rows affected: 0 Rows inserted: 0 Rows updated: 0
- Temporary Tables: 0 Temporary tables created: 0
- Other Info: Thread Id: 53

Session 3 (Bottom):

- Time (as measured at client side): Execution time: 0.0000000000000000
- Time (as measured by the server): Execution time: 0.0000000000000000
- Table lock wait time: 0.0000000000000000
- Errors: 0 Warnings: 0
- Rows processed: 0 Rows affected: 0 Rows inserted: 0 Rows updated: 0
- Temporary Tables: 0 Temporary tables created: 0
- Other Info: Thread Id: 53

The results show identical execution times and statistics across all three sessions, indicating consistent performance.

The screenshot shows the MySQL Workbench interface. The main area displays a complex SQL query:

```
SELECT COUNT(*) FROM user_details;
SELECT AVG(user_id) FROM user_details;
SELECT COUNT(*) FROM user_details WHERE user_id > last_user_id;
SELECT COUNT(*) FROM user_details WHERE user_id < first_user_id;
SELECT COUNT(*) FROM user_details WHERE user_id > last_user_id AND user_id < first_user_id;
SELECT user_id AS user_id, user_name AS user_name, user_email AS user_email
FROM user_details
WHERE user_id > 20 AND user_id < first_user_id AND user_email LIKE '%@%'.
```

The 'Schemas' tab is selected, showing the database structure. Other tabs include 'Administrators', 'Session', and 'Object Info'. A 'SQl' tab is also visible at the top.

Optimized Query

The screenshot shows a complex SQL query being executed in MyPQ Local Workbench. The query involves multiple joins and conditions, including a self-join on the 'user_details' table. The execution plan is displayed as a tree structure with nodes representing different parts of the query. The interface includes tabs for SQL, Grid, and File, and a sidebar with navigation, schema browser, and object info sections. A status bar at the bottom provides information about the connection and session.

The screenshot shows the MySQL Workbench interface with a query editor containing a multi-step SQL script. The script uses various functions like COUNT, GROUP_CONCAT, and SUBSTR to extract user details from a database. It includes several WHERE clauses involving LIKE and IN clauses to filter data. The results of the query are displayed in a table with columns for id, user_name, first_name, last_name, and user_id.

Query Results

| id | user_name | first_name | last_name | user_id |
|----|------------|------------|-----------|---------|
| 1 | root | MySQL | Workbench | 1 |
| 2 | kreenashal | Kreenash | Shah | 2 |
| 3 | root | MySQL | Workbench | 1 |
| 4 | root | MySQL | Workbench | 1 |
| 5 | root | MySQL | Workbench | 1 |
| 6 | root | MySQL | Workbench | 1 |
| 7 | root | MySQL | Workbench | 1 |
| 8 | root | MySQL | Workbench | 1 |
| 9 | root | MySQL | Workbench | 1 |
| 10 | root | MySQL | Workbench | 1 |
| 11 | root | MySQL | Workbench | 1 |

Query Statistics

Timing (as measured at client side):
Execution time: 0.0000000000000000
Timing (as measured by the server):
Execution time: 0.0000000000000000
Table lock wait time: 0.0000000000000000

Errors:
Warnings: 0
Warnings: 0

Rows Processed:
Rows read: 10000
Rows sent to client: 10000
Rows selected: 10000

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

Index Usage:
No index used

Other Info:
Events 16 93
Threads 12 12

The screenshot shows the MySQL Workbench interface with a query editor and a results pane. The query editor contains a multi-line SQL script for selecting user details from a database named 'krennabah'. The results pane displays the execution log for each statement, including timing, errors, and index usage.

| Statement | Time (ms) | Timing (as measured by the client) | Timing (as measured by the server) | Errors | Warnings | Rows Processed | Temporary Tables | Index Usage |
|---|-----------|------------------------------------|------------------------------------|--------|----------|----------------|-----------------------------|-------------|
| 1 USE krennabah; | 0 | Execution time: 0.000 < 0.000000 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 2 SELECT * FROM user_details; | 1 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 3 OPTIMIZE TABLE user_details; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 4 SELECT COUNT(*) FROM user_details; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 5 SELECT AVG(user_id) FROM user_details; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 6 SELECT SUM(user_id) FROM user_details; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 7 SELECT * FROM user_details ORDER BY last_name DESC; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 8 SELECT * FROM user_details WHERE password IN ('SECRET password FROM user_details WHERE password LIKE '%2%') | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 9 SELECT * FROM user_details WHERE password = 'SECRET password FROM user_details WHERE password LIKE '%2%' | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 10 FROM user_DETAILS u1 | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |
| 11 JOIN user_DETAILS u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id; | 0 | 0 | 0 | None | None | 0 | 0 temporary tables created. | None |

The screenshot shows the MySQL Workbench interface with a query editor and a timeline panel.

Query Editor:

```
1 USE kreenashah;
2 SELECT * FROM user_details;
3 UPDATE TABLE user_details;
4 UPDATE TABLE user_details;
5 SELECT COUNT(*) FROM user_details;
6 SELECT ROW(user_id) FROM user_details;
7 SELECT * FROM user_details ORDER BY last_name DESC;
8 SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '2%');
9 SELECT username AS user_username, u2.username AS user2_username
10 FROM user_details u1
11 JOIN user_details u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id;
```

Timeline Panel:

| Event | Time | Duration |
|------------------|---------------------|-----------|
| 1. Query start | 2023-07-20 10:51:47 | 0.000 sec |
| 2. Rows Read | 2023-07-20 10:51:47 | 0.000 sec |
| 3. Rows Inserted | 2023-07-20 10:51:47 | 0.000 sec |
| 4. Rows Updated | 2023-07-20 10:51:47 | 0.000 sec |
| 5. Rows Deleted | 2023-07-20 10:51:47 | 0.000 sec |
| 6. Rows Merged | 2023-07-20 10:51:47 | 0.000 sec |
| 7. Rows Fetched | 2023-07-20 10:51:47 | 0.000 sec |
| 8. Other Info | 2023-07-20 10:51:47 | 0.000 sec |
| 9. Total | 2023-07-20 10:51:47 | 0.000 sec |

Information Panels:

- Statistics:** Shows query statistics like timing and row counts.
- Joins per Type:** Details the types of joins used in the query.
- Sorting:** Details the sorting operations.
- Index Usage:** Details the use of indexes.
- Temporary Tables:** Details temporary tables created.
- Action Order:** Details the execution order of statements.
- Output:** Details the output of each statement.

Indexed Query

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- 6050421090
- bview
- courses
- husain
- kreenashah
- kreena
- kreena243
- kreena245
- new
- newschema
- saikia
- sys
- university_239
- world

Query 1 SQL File 3* SQL File 4* SQL File 7* SQL File 6* SQL File 8*

```

1 • USE kreenashah;
2 • SELECT * FROM user_details;
3
4 • OPTIMIZE TABLE user_details;
5 • CREATE INDEX IND ON user_details(user_id, username, first_name, last_name, gender, password, status);
6 • SELECT COUNT(*) FROM user_details;
7 • SELECT AVG(user_id) FROM user_details;
8 • SELECT * FROM user_details ORDER BY last_name desc;
9 • SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '%2%');
10 • SELECT u1.username AS user1_username, u2.username AS user2_username
11 FROM USER_DETAILS u1
12 JOIN USER_DETAILS u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id;

```

Line to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.125000000
Timing (as measured by the server):
Execution time: 0:00:0.125000000
Table lock wait time: 0:00:0.000000000

Errors:
No Errors / NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 10000
Rows read from disk: 10000

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

Schema: kreenashah

user_details 21

Action Output

| Time | Action | Message | Duration / Fetch |
|-------------|---|--|-----------------------|
| 25 07:53:13 | SELECT AVG(user_id) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.031 sec / 0.000 sec |
| 26 07:54:08 | SELECT * FROM user_details ORDER BY last_name desc LIMIT 0, 10000 | 10000 row(s) returned | 0.500 sec / 0.000 sec |
| 27 07:54:26 | SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '%2%') LIMIT 0, 10000 | 5175 row(s) returned | 0.469 sec / 0.484 sec |
| 28 07:54:43 | SELECT u1.username AS user1_username, u2.username AS user2_username FROM USER_DETAILS u1 JOIN USER_DETAILS u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id; | 10000 row(s) returned | 0.078 sec / 0.985 sec |
| 29 07:55:46 | CREATE INDEX IND ON user_details(user_id, username, first_name, last_name, gender, password, status) | 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0 | 1.266 sec |
| 30 07:55:53 | SELECT COUNT(*) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.000 sec / 0.000 sec |
| 31 07:56:09 | SELECT AVG(user_id) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.015 sec / 0.000 sec |
| 32 07:56:33 | SELECT * FROM user_details ORDER BY last_name desc LIMIT 0, 10000 | 10000 row(s) returned | 0.125 sec / 0.000 sec |

Object Info Session

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- 6050421090
- bview
- courses
- husain
- kreenashah
- kreena
- kreena243
- kreena245
- new
- newschema
- saikia
- sys
- university_239
- world

Query 1 SQL File 3* SQL File 4* SQL File 7* SQL File 6* SQL File 8*

```

1 • USE kreenashah;
2 • SELECT * FROM user_details;
3
4 • OPTIMIZE TABLE user_details;
5 • CREATE INDEX IND ON user_details(user_id, username, first_name, last_name, gender, password, status);
6 • SELECT COUNT(*) FROM user_details;
7 • SELECT AVG(user_id) FROM user_details;
8 • SELECT * FROM user_details ORDER BY last_name desc;
9 • SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '%2%');
10 • SELECT u1.username AS user1_username, u2.username AS user2_username
11 FROM USER_DETAILS u1
12 JOIN USER_DETAILS u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id;

```

Line to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.000000000
Timing (as measured by the server):
Execution time: 0:00:0.000157900
Table lock wait time: 0:00:0.000000000

Errors:
No Errors / NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows read from disk: 1

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

Schema: kreenashah

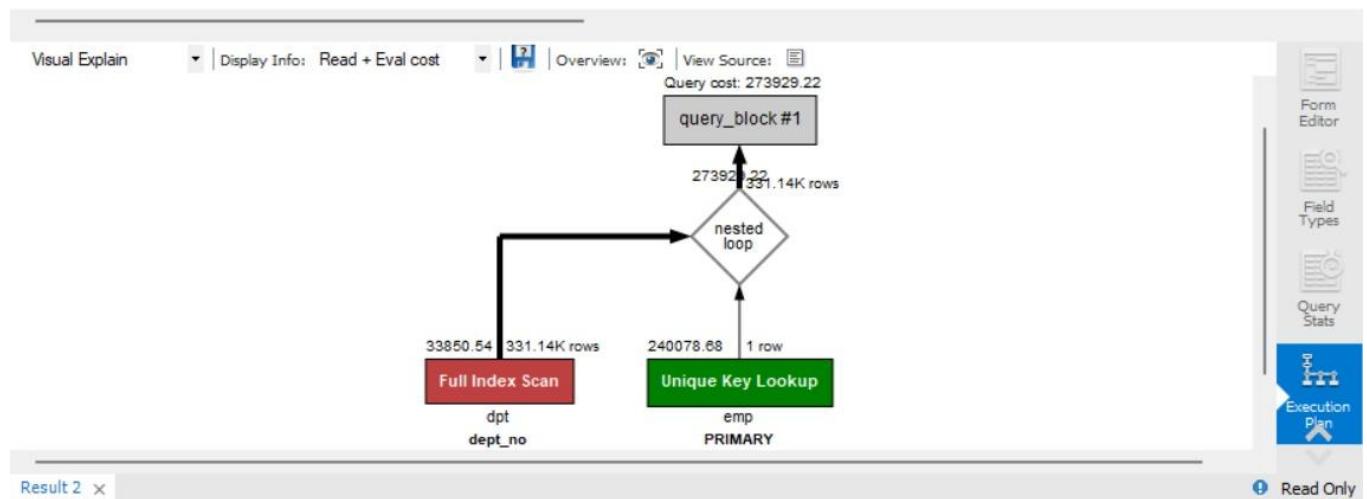
user_details 19

Action Output

| Time | Action | Message | Duration / Fetch |
|-------------|---|--|-----------------------|
| 23 07:53:15 | OPTIMIZE TABLE user_details | | 0.047 sec / 0.000 sec |
| 24 07:53:25 | SELECT COUNT(*) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.000 sec / 0.000 sec |
| 25 07:53:51 | SELECT AVG(user_id) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.031 sec / 0.000 sec |
| 26 07:54:03 | SELECT * FROM user_details ORDER BY last_name desc LIMIT 0, 10000 | 10000 row(s) returned | 0.500 sec / 0.000 sec |
| 27 07:54:20 | SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '%2%') LIMIT 0, 10000 | 5175 row(s) returned | 0.469 sec / 0.484 sec |
| 28 07:54:46 | SELECT u1.username AS user1_username, u2.username AS user2_username FROM USER_DETAILS u1 JOIN USER_DETAILS u2 ON u1.first_name = u2.first_name AND u1.user_id < u2.user_id; | 10000 row(s) returned | 0.078 sec / 0.985 sec |
| 29 07:55:46 | CREATE INDEX IND ON user_details(user_id, username, first_name, last_name, gender, password, status) | 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0 | 1.266 sec |
| 30 07:55:53 | SELECT COUNT(*) FROM user_details LIMIT 0, 10000 | 1row(s) returned | 0.000 sec / 0.000 sec |

Object Info Session

Experiment No 3 - Query Monitoring



Experiment No 4 - B/B+ Trees

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "B&BPLUS" containing "BplusTree.py" and "BTree.py".
- Code Editor:** Displays the content of "BTree.py".
- Terminal:** Shows the command "python .\BTree.py" being run, followed by its output:

```
Requirement already satisfied: cffi in c:\python310\lib\site-packages (from persistent>=4.1.0->BTrees) (1.15.1)
Requirement already satisfied: setuptools in c:\python310\lib\site-packages (from zope.interface>=5.0.0->BTrees) (65.5.0)
Requirement already satisfied: pycparser in c:\python310\lib\site-packages (from cffi->persistent>=4.1.0->BTrees) (2.21)
Installing collected packages: zope.interface, persistent, BTrees
Successfully installed BTrees-5.1 persistent-5.1 zope.interface-6.1

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
● PS C:\Users\Admin\OneDrive\Desktop\sem5\sem5_pracs\VADBMS\B&Bplus> python .\BTree.py
Insertion time: 7123.004 milliseconds
enter key: 33333
66666
Search time: 7123.004 milliseconds
● PS C:\Users\Admin\OneDrive\Desktop\sem5\sem5_pracs\VADBMS\B&Bplus>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "B&BPLUS" containing "BplusTree.py" and "BTree.py".
- Code Editor:** Displays the content of "BplusTree.py".
- Terminal:** Shows the command "python .\BplusTree.py" being run, followed by its output:

```
Successfully installed BTrees-5.1 persistent-5.1 zope.interface-6.1

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
● PS C:\Users\Admin\OneDrive\Desktop\sem5\sem5_pracs\VADBMS\B&Bplus> python .\BplusTree.py
Insertion time: 7123.004 milliseconds
enter key: 33333
66666
Search time: 7123.004 milliseconds
● PS C:\Users\Admin\OneDrive\Desktop\sem5\sem5_pracs\VADBMS\B&Bplus>
```

Experiment No 5 - Distributed Database Design

MySQL Workbench - Local instance MySQL80

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content: | SQL Grid | Form Editor | Context Help | Snippets

Output

```

27 (14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
28 (15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
29 (16, 'Molly Clark', 'Manager', 'Operations', 80000),
30 (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 70000),
31 (18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
32 (19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
33 (20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);

34 • ALTER TABLE usingRange PARTITION BY RANGE (ID) (
35   PARTITION p0 VALUES LESS THAN ($1),
36   PARTITION p1 VALUES LESS THAN ($1),
37   PARTITION p2 VALUES LESS THAN ($1),
38   PARTITION p3 VALUES LESS THAN MAXVALUE
39 );
40 )
41 
42 • SELECT * FROM usingRange PARTITION (p0);
43 • SELECT * FROM usingRange PARTITION (p1);
44 • SELECT * FROM usingRange PARTITION (p2);
45 • SELECT * FROM usingRange PARTITION (p3);

```

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content: | SQL Grid | Form Editor | Context Help | Snippets

| ID | Name | Position | Department | Salary |
|----|----------------|----------------------|------------|--------|
| 1 | Kreena Shah | Manager | Sales | 70000 |
| 2 | Arya Smith | Assistant Manager | Sales | 60000 |
| 3 | Sakshi Johnson | Sales Representative | Sales | 50000 |
| 4 | Alice Williams | Manager | Marketing | 80000 |
| 5 | Bob Brown | Assistant Manager | Marketing | 70000 |

Action Output

- 1 07:42:05 CREATE DATABASE kreenshah243 1 rows affected 0.001 sec
- 2 07:42:05 USE kreenshah243 0 rows affected 0.015 sec
- 3 07:42:05 CREATE TABLE usingRange (ID INT PRIMARY KEY , Name VARCHAR(50) , Position VARCHAR(50) , Department VARCHAR(50) , Salary INT) 0 rows affected 0.002 sec
- 4 07:42:12 INSERT INTO usingRange (ID, Name, Position, Department, Salary) VALUES (1, 'Kreena Shah', 'Manager', 'Sales', 70000), (2, 'Arya Smith', 'Assistant Manager', 'Sales', 60000), (3, 'Sakshi Johnson', 'Sales Representative', 'Sales', 50000), (4, 'Alice Williams', 'Manager', 'Marketing', 80000), (5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.047 sec
- 5 07:42:25 ALTER TABLE usingRange PARTITION BY RANGE (ID) (PARTITION p0 VALUES LESS THAN (\$1), PARTITION p1 VALUES LESS THAN (\$1), PARTITION p2 VALUES LESS THAN (\$1), PARTITION p3 VALUES LESS THAN MAXVALUE) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.224 sec
- 6 07:42:28 SELECT * FROM usingRange PARTITION (p0) LIMIT 0, 50000 5 rows returned 0.000 sec / 0.000 sec

Object Info | Session

MySQL Workbench - Local instance MySQL80

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content: | SQL Grid | Form Editor | Context Help | Snippets

Output

```

27 (15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
28 (16, 'Molly Clark', 'Manager', 'Operations', 80000),
29 (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 70000),
30 (18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
31 (19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
32 (20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);

33 • ALTER TABLE usingList PARTITION BY LIST (ID) (
34   PARTITION p0 VALUES IN (1,2,3,4,5),
35   PARTITION p1 VALUES IN (6,7,8,9,10),
36   PARTITION p2 VALUES IN (11,12,13,14,15),
37   PARTITION p3 VALUES IN (16,17,18,19,20)
38 );
39 )
40 
41 • SELECT * FROM usingList PARTITION (p0);
42 • SELECT * FROM usingList PARTITION (p1);
43 • SELECT * FROM usingList PARTITION (p2);
44 • SELECT * FROM usingList PARTITION (p3);

```

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content: | SQL Grid | Form Editor | Context Help | Snippets

| ID | Name | Position | Department | Salary |
|----|----------------|----------------------|------------|--------|
| 1 | Kreena Shah | Manager | Sales | 70000 |
| 2 | Arya Smith | Assistant Manager | Sales | 60000 |
| 3 | Sakshi Johnson | Sales Representative | Sales | 50000 |
| 4 | Alice Williams | Manager | Marketing | 80000 |
| 5 | Bob Brown | Assistant Manager | Marketing | 70000 |

Action Output

- 2 07:42:05 USE kreenshah243 0 rows affected 0.016 sec
- 3 07:42:05 CREATE TABLE usingList (ID INT PRIMARY KEY , Name VARCHAR(50) , Position VARCHAR(50) , Department VARCHAR(50) , Salary INT) 0 rows affected 0.062 sec
- 4 07:42:12 INSERT INTO usingList (ID, Name, Position, Department, Salary) VALUES (1, 'Kreena Shah', 'Manager', 'Sales', 70000), (2, 'Arya Smith', 'Assistant Manager', 'Sales', 60000), (3, 'Sakshi Johnson', 'Sales Representative', 'Sales', 50000), (4, 'Alice Williams', 'Manager', 'Marketing', 80000), (5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.047 sec
- 5 07:42:25 ALTER TABLE usingList PARTITION BY RANGE (ID) (PARTITION p0 VALUES LESS THAN (\$1), PARTITION p1 VALUES LESS THAN (\$1), PARTITION p2 VALUES LESS THAN (\$1), PARTITION p3 VALUES LESS THAN (\$1)) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.234 sec
- 6 07:42:28 SELECT * FROM usingList PARTITION (p0) LIMIT 0, 50000 5 rows returned 0.000 sec / 0.000 sec
- 7 07:43:57 USE kreenshah243 0 rows affected 0.000 sec
- 8 07:44:00 CREATE TABLE usingList (ID INT PRIMARY KEY , Name VARCHAR(50) , Position VARCHAR(50) , Department VARCHAR(50) , Salary INT) 0 rows affected 0.047 sec
- 9 07:44:02 INSERT INTO usingList (ID, Name, Position, Department, Salary) VALUES (1, 'Kreena Shah', 'Manager', 'Sales', 70000), (2, 'Arya Smith', 'Assistant Manager', 'Sales', 60000), (3, 'Sakshi Johnson', 'Sales Representative', 'Sales', 50000), (4, 'Alice Williams', 'Manager', 'Marketing', 80000), (5, 'Bob Brown', 'Assistant Manager', 'Marketing', 70000) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.015 sec
- 10 07:44:07 ALTER TABLE usingList PARTITION BY LIST (ID) (PARTITION p0 VALUES IN (1,2,3,4,5), PARTITION p1 VALUES IN (6,7,8,9,10), PARTITION p2 VALUES IN (11,12,13,14,15), PARTITION p3 VALUES IN (16,17,18,19,20)) 20 rows affected Records: 20 Duplicates: 0 Warnings: 0 0.234 sec
- 11 07:44:11 SELECT * FROM usingList PARTITION (p0) LIMIT 0, 50000 5 rows returned 0.000 sec / 0.000 sec

Object Info | Session

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- 6004210090
- brew
- coursera
- employedb
- husain
- husain9
- hussein
- kreena243
- kreena_243
- kreenashah
- newschema
- parandb
- sakila
- sonia
- sys
- university_239
- world

SQL File 6* SQL File 3* SQL File 4* SQL File 5* SQL File 7* SQL File 8*

```

20 (6, 'Eva Wilson', 'Assistant Manager', 'HR', 65000),
21 (9, 'Frank Moore', 'HR Specialist', 'HR', 55000),
22 (10, 'Grace Taylor', 'Manager', 'Finance', 85000),
23 (11, 'Henry Anderson', 'Assistant Manager', 'Finance', 75000),
24 (12, 'Irene Thomas', 'Finance Specialist', 'Finance', 65000),
25 (13, 'Jack Jackson', 'Manager', 'IT', 90000),
26 (14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
27 (15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
28 (16, 'Molly Clark', 'Manager', 'Operations', 90000),
29 (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 80000),
30 (18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
31 (19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
32 (20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);
33
34 • ALTER TABLE usingshash PARTITION BY HASH(ID)
35
36
37 • SELECT * FROM usingshash PARTITION (p0);
38
39 • SELECT * FROM usingshash PARTITION (n1);

```

Result Grid

| ID | Name | Position | Department | Salary |
|----|-----------------|---------------------------------|------------------|--------|
| 4 | Alice Wilkins | Manager | Marketing | 80000 |
| 8 | Eva Wilson | Assistant Manager | HR | 65000 |
| 12 | Irene Thomas | Finance Specialist | Finance | 65000 |
| 13 | Jack Jackson | Manager | Operations | 90000 |
| 20 | Robert Robinson | Customer Service Representative | Customer Service | 50000 |

Administration Schemas Information

No object selected

usinghash 1 ×

Action Output

| Time | Action | Message | Duration / Fetch |
|-------------|---|---|-----------------------|
| 7 07:43:57 | USE `keenshash` | 0 rows(s) affected | 0.000 sec |
| 8 07:44:00 | CREATE TABLE `usingshash` (`ID` INT PRIMARY KEY, `Name` VARCHAR(50), `Position` VARCHAR(50), `Department` VARCHAR(50), `Salary` INT) | 0 rows(s) affected | 0.047 sec |
| 9 07:44:02 | INSERT INTO `usingshash` (ID, Name, Position, Department, Salary) VALUES (1, 'Keena Shah', 'Manager', 'Sales', 70000), (2, 'Aya Smith', 'Assistant Manager', 'Marketing', 60000) | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.015 sec |
| 10 07:44:07 | ALTER TABLE `usingshash` PARTITION BY LIST (ID) (PARTITION p0 VALUES IN (1,2,3,4), PARTITION p1 VALUES IN (6,7,8,9,10), PARTITION p2 VALUES IN (11,12,13,14,15,16,17,18,19,20)) | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.234 sec |
| 11 07:44:11 | SELECT * FROM `usingshash` PARTITION p0 LIMIT 0, 50000 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 12 07:44:20 | USE `keenshash` | 0 rows(s) affected | 0.000 sec |
| 13 07:44:32 | CREATE TABLE `usingshash` (`ID` INT PRIMARY KEY, `Name` VARCHAR(50), `Position` VARCHAR(50), `Department` VARCHAR(50), `Salary` INT) | 0 rows(s) affected | 0.047 sec |
| 14 07:44:35 | INSERT INTO `usingshash` (ID, Name, Position, Department, Salary) VALUES (1, 'Keena Shah', 'Manager', 'Sales', 70000), (2, 'Aya Smith', 'Assistant Manager', 'Marketing', 60000) | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 15 07:44:38 | ALTER TABLE `usingshash` PARTITION BY HASH(ID) PARTITIONS 4 | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.218 sec |
| 16 07:44:43 | SELECT * FROM `usingshash` PARTITION p0 LIMIT 0, 50000 | 5 row(s) returned | 0.000 sec / 0.000 sec |

Object Info Session

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- 6004210090
- brew
- coursera
- employedb
- husain
- husain9
- hussein
- kreena243
- kreena_243
- kreenashah
- new
- newschema
- parandb
- sakila
- sonia
- sys
- university_239
- world

SQL File 6* SQL File 3* SQL File 4* SQL File 5* SQL File 7* SQL File 8*

```

20 (6, 'Eva Wilson', 'Assistant Manager', 'HR', 65000),
21 (9, 'Frank Moore', 'HR Specialist', 'HR', 55000),
22 (10, 'Grace Taylor', 'Manager', 'Finance', 85000),
23 (11, 'Henry Anderson', 'Assistant Manager', 'Finance', 75000),
24 (12, 'Irene Thomas', 'Finance Specialist', 'Finance', 65000),
25 (13, 'Jack Jackson', 'Manager', 'IT', 90000),
26 (14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
27 (15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
28 (16, 'Molly Clark', 'Manager', 'Operations', 90000),
29 (17, 'Nancy Thompson', 'Assistant Manager', 'Operations', 80000),
30 (18, 'Oscar Garcia', 'Operations Specialist', 'Operations', 60000),
31 (19, 'Patricia Martinez', 'Manager', 'Customer Service', 75000),
32 (20, 'Robert Robinson', 'Customer Service Representative', 'Customer Service', 50000);
33
34 • ALTER TABLE usingskey PARTITION BY KEY()
35
36
37 • SELECT * FROM usingskey PARTITION (p0);
38
39 • SELECT * FROM usingskey PARTITION (n1);

```

Result Grid

| ID | Name | Position | Department | Salary |
|----|----------------|-------------------|------------|--------|
| 1 | Keena Shah | Manager | Sales | 70000 |
| 5 | Bob Brown | Assistant Manager | Marketing | 60000 |
| 9 | Frank Moore | HR Specialist | HR | 55000 |
| 13 | Jack Jackson | Manager | IT | 90000 |
| 17 | Nancy Thompson | Assistant Manager | Operations | 70000 |

Administration Schemas Information

No object selected

usingkey 1 ×

Action Output

| Time | Action | Message | Duration / Fetch |
|-------------|---|---|-----------------------|
| 12 07:45:39 | USE `keenshash` | 0 rows(s) affected | 0.000 sec |
| 13 07:45:42 | CREATE TABLE `usingskey` (`ID` INT PRIMARY KEY, `Name` VARCHAR(50), `Position` VARCHAR(50), `Department` VARCHAR(50), `Salary` INT) | 0 rows(s) affected | 0.047 sec |
| 14 07:45:45 | INSERT INTO `usingskey` (ID, Name, Position, Department, Salary) VALUES (1, 'Keena Shah', 'Manager', 'Sales', 70000), (2, 'Aya Smith', 'Assistant Manager', 'Marketing', 60000) | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 15 07:45:49 | ALTER TABLE `usingskey` PARTITION BY HASH(ID) PARTITIONS 4 | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.218 sec |
| 16 07:45:51 | SELECT * FROM `usingskey` PARTITION p0 LIMIT 0, 50000 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 17 07:45:52 | USE `keenshash` | 0 rows(s) affected | 0.000 sec |
| 18 07:45:55 | CREATE TABLE `usingskey` (`ID` INT PRIMARY KEY, `Name` VARCHAR(50), `Position` VARCHAR(50), `Department` VARCHAR(50), `Salary` INT) | 0 rows(s) affected | 0.047 sec |
| 19 07:45:58 | INSERT INTO `usingskey` (ID, Name, Position, Department, Salary) VALUES (1, 'Keena Shah', 'Manager', 'Sales', 70000), (2, 'Aya Smith', 'Assistant Manager', 'Marketing', 60000) | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.016 sec |
| 20 07:45:02 | ALTER TABLE `usingskey` PARTITION BY KEY() PARTITIONS 4 | 20 rows(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.250 sec |
| 21 07:45:05 | SELECT * FROM `usingskey` PARTITION p0 LIMIT 0, 50000 | 5 row(s) returned | 0.000 sec / 0.000 sec |

Object Info Session

Experiment No 6 - Two Phase Protocol

Server Code

```
import socket def ServerSoc():
host = "127.0.0.1"
port = 8000
print("Server is running!") msg = "PREPARE"
log = msg over = 0
s_soc = socket.socket()
s_soc.bind((host,port))
s_soc.listen(2) while(1):
replies = []
print(f"Coordinator : {msg.upper()}") for i in range(3):
conn,add = s_soc.accept() conn.send(msg.encode())
data = conn.recv(1024).decode() replies.append(data.upper()) print(f"Subordinator {i} {add} :
{data.upper()}")
if over == 1: break
if ("ABORT" in replies) or (len(replies)<3) :
print(f'at the abort stage the replies are {replies},and the length is
{len(replies)}')
msg = "ABORT"
print("TRANSACTION ABORTED!\nThe final log is: ", log+" "+msg) over = 1
elif "SUCCESS" in replies: msg = "COMPLETE"
print("TRANSACTION COMPLETED!\nThe final log is: ", log+" "+msg) over = 1
else:
msg = "COMMIT"
log += " "+msg ServerSoc()
```

Client Code

```
import socket def ClientSoc():
host = "127.0.0.1"
port = 8000 log = "" over = 0 while(1):
try:
s_soc = socket.socket() s_soc.connect((host,port))
rec_data = s_soc.recv(1024).decode() print("Coordinator: ", rec_data.upper()) if
rec_data.upper() == "ABORT":
msg = "OK" print("TRANSACTION ABORTED!")
over = 1
elif rec_data.upper() == "SUCCESS": msg = "OK"
print("TRANSACTION COMPLETED!")
over = 1
else:
msg = input("System Status: ").upper() log += " "+msg s_soc.send(msg.encode())
if over == 1: break s_soc.close()
except:
print("END OF TRANSACTION\n final log is: ",log+" "+rec_data.upper()) break
ClientSoc()
```

Output:

All in the ready phase:

```
Windows PowerShell x + - x Windows PowerShell x + - x Windows PowerShell x + - x Windows PowerShell x + - x

Server is running!
Coordinator : PREPARE
Subordinator 0 ('127.0.0.1', 61780) : READY
Subordinator 1 ('127.0.0.1', 61781) : READY
Subordinator 2 ('127.0.0.1', 61782) : READY
Coordinator : COMMIT
Subordinator 0 ('127.0.0.1', 61785) : COMPLETE
Subordinator 1 ('127.0.0.1', 61786) : COMPLETE
Subordinator 2 ('127.0.0.1', 61787) : COMPLETE
Coordinator : COMMIT
Subordinator 0 ('127.0.0.1', 61792) : OK
Subordinator 1 ('127.0.0.1', 61793) : OK
Subordinator 2 ('127.0.0.1', 61794) : OK
Coordinator : COMMIT
Subordinator 0 ('127.0.0.1', 61796) : SUCCESS
Subordinator 1 ('127.0.0.1', 61801) : SUCCESS
Subordinator 2 ('127.0.0.1', 61802) : SUCCESS
TRANSACTION COMPLETED!
The final log is: PREPARE COMMIT COMMIT COMMIT COMPLETE
Coordinator : COMPLETE
Subordinator 0 ('127.0.0.1', 61803) : OK
Subordinator 1 ('127.0.0.1', 61805) : OK
Subordinator 2 ('127.0.0.1', 61811) : OK
PS C:\Users\almart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```



```
PS C:\Users\almart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> python client1.py
Coordinator: PREPARE
System Status: READY
Coordinator: COMMIT
System Status: COMPLETE
Coordinator: COMMIT
System Status: OK
Coordinator: COMMIT
System Status: SUCCESS
Coordinator: COMPLETE
System Status: OK
END OF TRANSACTION
final log is: READY COMPLETE OK SUCCESS OK COMPLETE
PS C:\Users\almart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```



```
PS C:\Users\almart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> python client2.py
Coordinator: PREPARE
System Status: Ready
Coordinator: COMMIT
System Status: COMPLETE
Coordinator: COMMIT
System Status: OK
Coordinator: COMMIT
System Status: SUCCESS
Coordinator: COMPLETE
System Status: OK
END OF TRANSACTION
Final log is: READY COMPLETE OK SUCCESS OK COMPLETE
PS C:\Users\almart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```

One of the subordinators aborts:

```
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> python server.py
Server Started
Coordinator : PREPARE
Subordinator 0 ('127.0.0.1', 61601) : READY
Subordinator 1 ('127.0.0.1', 61604) : READY
Subordinator 2 ('127.0.0.1', 61605) : ABORT
at the abort stage the replies are ['READY', 'READY', 'ABORT'],and the length is 3
TRANSACTION ABORTED!
The final log is: PREPARE ABORT
Coordinator : ABORT
Subordinator 0 ('127.0.0.1', 61606) :
Subordinator 1 ('127.0.0.1', 61607) :
Subordinator 2 ('127.0.0.1', 61608) :
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```



```
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> python client.py
Coordinator : PREPARE
System Status: READY
Coordinator: ABORT
TRANSACTION ABORTED!
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```



```
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> python client.py
Coordinator : PREPARE
System Status: ABORT
Coordinator: ABORT
TRANSACTION ABORTED!
PS C:\Users\amart\OneDrive\Documents\Sem5\LAB-Sem5\ADBSMS-lab\labcode> |
```

Experiment No 7 - XML Programming

Code :

```
<!DOCTYPE html>
<html>
<head>
<title>XML Catalog</title>
</head>
<body>
<h1>Book Catalog</h1>
<input type="text" id="titleInput" placeholder="Search by Title">
<input type="text" id="authorInput" placeholder="Search by Author">
<input type="text" id="genreInput" placeholder="Search by Genre">
<input type="number" id="priceInput" placeholder="Search by Price">
<input type="date" id="dateInput" placeholder="Search by Publish Date">
<button onclick="searchBooks()">Search</button>
<ul id="bookList"></ul>

<script>
var catalogData; // Store the XML data for searching

// Function to load and parse the XML file
function loadXML() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState === 4 && this.status === 200) {
            catalogData = this.responseXML; // Store the XML data for searching
            displayCatalog(catalogData);
        }
    };
    xhttp.open("GET", "book.xml", true);
    xhttp.send();
}

// Function to display the catalog data
function displayCatalog(xml) {
    var books = xml.getElementsByTagName("book");
    var list = document.getElementById("bookList");

    for (var i = 0; i < books.length; i++) {
        var book = books[i];
        var author = book.getElementsByTagName("author")[0].textContent;
        var title = book.getElementsByTagName("title")[0].textContent;
        var genre = book.getElementsByTagName("genre")[0].textContent;
        var price = parseFloat(book.getElementsByTagName("price")[0].textContent);
        var publishDate = new Date(book.getElementsByTagName("publish_date")[0].textContent);
        var description = book.getElementsByTagName("description")[0].textContent;

        var listItem = document.createElement("li");
        listItem.innerHTML = "<strong>Title:</strong> " + title + "<br>" +
            "<strong>Author:</strong> " + author + "<br>" +
            "<strong>Genre:</strong> " + genre + "<br>" +
            "<strong>Price:</strong> " + price + "<br>" +
            "<strong>Publish Date:</strong> " + publishDate.toDateString() + "<br>" +
            "<strong>Description:</strong> " + description;
        list.appendChild(listItem);
    }
}
```

```

        "<strong>Price:</strong> $" + price.toFixed(2) + "<br>" +
        "<strong>Publish Date:</strong> " + publishDate.toDateString() + "<br>" +
        "<strong>Description:</strong> " + description + "<br>";

    list.appendChild(listItem);
}
}

// Function to search books based on user input
function searchBooks() {
    var title = document.getElementById("titleInput").value.toLowerCase();
    var author = document.getElementById("authorInput").value.toLowerCase();
    var genre = document.getElementById("genreInput").value.toLowerCase();
    var price = parseFloat(document.getElementById("priceInput").value);
    var date = new Date(document.getElementById("dateInput").value);

    var list = document.getElementById("bookList");
    list.innerHTML = ""; // Clear the previous list

    var books = catalogData.getElementsByTagName("book");

    for (var i = 0; i < books.length; i++) {
        var book = books[i];
        var bookTitle = book.getElementsByTagName("title")[0].textContent.toLowerCase();
        var bookAuthor = book.getElementsByTagName("author")[0].textContent.toLowerCase();
        var bookGenre = book.getElementsByTagName("genre")[0].textContent.toLowerCase();
        var bookPrice = parseFloat(book.getElementsByTagName("price")[0].textContent);
        var bookDate = new Date(book.getElementsByTagName("publish_date")[0].textContent);

        if (
            (title === "" || bookTitle.includes(title)) &&
            (author === "" || bookAuthor.includes(author)) &&
            (genre === "" || bookGenre.includes(genre)) &&
            (isNaN(price) || price === bookPrice) &&
            (isNaN(date) || date.toDateString() === bookDate.toDateString())
        ) {
            var description = book.getElementsByTagName("description")[0].textContent;

            var listItem = document.createElement("li");
            listItem.innerHTML = "<strong>Title:</strong> " + bookTitle + "<br>" +
                "<strong>Author:</strong> " + bookAuthor + "<br>" +
                "<strong>Genre:</strong> " + bookGenre + "<br>" +
                "<strong>Price:</strong> $" + bookPrice.toFixed(2) + "<br>" +
                "<strong>Publish Date:</strong> " + bookDate.toDateString() + "<br>" +
                "<strong>Description:</strong> " + description + "<br>";

            list.appendChild(listItem);
        }
    }
}

```

```

// Load and display the XML data when the page loads
window.onload = function() {
    loadXML();
};

</script>
</body>
</html>

```

Database (sample book added here)

```

<?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>
</catalog>

```

Screenshots :

Search by Title Search by Author Search by Genre Search by Price dd-mm-yyyy Search

- Title:** XML Developer's Guide
Author: Gambardella, Matthew
Genre: Computer
Price: \$44.95
Publish Date: Sun Oct 01 2000
Description: An in-depth look at creating applications with XML.
- Title:** Midnight Rain
Author: Ralls, Kim
Genre: Fantasy
Price: \$5.95
Publish Date: Sat Dec 16 2000
Description: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.
- Title:** Maeve Ascendant
Author: Corets, Eva
Genre: Fantasy
Price: \$5.95
Publish Date: Fri Nov 17 2000
Description: After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.
- Title:** Oberon's Legacy
Author: Corets, Eva
Genre: Fantasy
Price: \$5.95
Publish Date: Sat Mar 10 2001
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.
- Title:** The Sundered Grail
Author: Corets, Eva
Genre: Fantasy
Price: \$5.95
Publish Date: Mon Sep 10 2001
Description: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Search by Title ralls, kim Search by Genre 5.95 dd-mm-yyyy Search

- Title:** midnight rain
Author: ralls, kim
Genre: fantasy
Price: \$5.95
Publish Date: Sat Dec 16 2000
Description: A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.

Experiment No 8 - Document Database

The image displays three separate browser windows, each showing a code editor interface for displaying data from a JSON file named `2022_JUNE.json`. The code in all three windows is identical, demonstrating how to handle JSON data using JavaScript and HTML.

```
<!DOCTYPE html>
<html>
<body>
<h1>Data Handling using JSON</h1>
<button onclick="Func()">Show Details</button>
<script>
function Func() {
    var data_loc;
    fetch("./2022_JUNE.json")
        .then((res) => {
            return res.json();
        })
        .then((data) =>
            console.log(data)
        );
}
</script>
</body>
</html>
```

The second window shows the expanded JSON object `2022_JUNE.json` in the code editor, revealing a complex nested structure of location history data. The third window shows the same code as the first, but with a different line number (Ln 38, Col 1) and a different status bar message (Spaces: 4, UTF-8, CRLF, HTML).

localhost:8000/displaydata.html

How to plot a ROC... GitHub - labrijisaad... Twitter Sentiment A... Twitter Sentiment A... Home - Netflix

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights

▲ DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfrchfhceifdheapeoicaho/sourceMap/chrome/scripts/iframe_form_check.map: System error:

```
displaydata.html:14
var {activitySegments: (...) } = data;
  ↳ activitySegments
    ↳ activities: (15) [(), {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}]
      ↳ activityType: "IN_PASSENGER_VEHICLE"
      ↳ confidence: "LOW"
      ↳ distance: 1665
      ↳ duration:
        ↳ endTimeStamp: "2022-06-03T13:37:18.241Z"
        ↳ startTimeStamp: "2022-06-03T12:45:39.015Z"
      ↳ [[Prototype]]: Object
    ↳ [[Prototype]]: Object
  ↳ [[Prototype]]: Object
  ↳ parkingEvent:
    ↳ location: (latitudeE7: 191739130, longitudeE7: 728600935, accuracyMetres: 77)
      ↳ locationSource: "NOT_FROM_RAW_LOCATION"
      ↳ method: "END_OF_ACTIVITY_SEGMENT"
      ↳ timestamp: "2022-06-03T13:38:35.646Z"
      ↳ [[Prototype]]: Object
    ↳ startLocation:
      ↳ latitudeE7: 191740278
      ↳ longitudeE7: 728600095
      ↳ sourceInfo: (deviceTag: 912782204)
      ↳ [[Prototype]]: Object
    ↳ [[Prototype]]: Object
  ↳ waypointPath:
    ↳ confidence: 1
    ↳ distanceMetres: 10958.0746174180279
    ↳ roadSegments: (28) [(), {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}]
      ↳ source: "INFERRED"
      ↳ travelMode: "DRIVING"
      ↳ waypoints: (3) [(), {}, {}]
      ↳ [[Prototype]]: Object
    ↳ [[Prototype]]: Object
  ↳ [[Prototype]]: Object
```

Show Details

File Edit Selection View Go Run Terminal Help

displaydata.html ● { } 2022_JUNE.json

EXPLORER

- TAKEOUT
 - Location History
 - Semantic Location History
 - 2018
 - 2019
 - 2020
 - 2021
 - 2022
 - Records.json
 - Settings.json
 - 2022_JUNE.json
 - displaydata.html

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Data Handling using JSON</h1>
    <button onclick="Func()">Show Details</button>
    <script>
      function Func() {
        var data_loc;
        fetch("./2022_JUNE.json")
          .then((res) => {
            return res.json();
          })
          .then((data) =>
            console.log(
              data.timelineObjects[1]
            )
          );
      }
    </script>
  </body>
</html>
```

localhost:8000/displaydata.html

How to plot a ROC... GitHub - labrijisaad... Twitter Sentiment A... Twitter Sentiment A... Home - Netflix

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights

▲ DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfrchfhceifdheapeoicaho/sourceMap/chrome/scripts/iframe_form_check.map: System error:

```
displaydata.html:14
var {placeVisits: (...) } = data;
  ↳ placeVisits
    ↳ centerLatE7: 191740471
    ↳ centerLangE7: 728606154
    ↳ childVisits: Array(4)
      ↳ 0: {centerLatE7: 191740018, centerLangE7: 728606383}
      ↳ 1: {duration: (startTimestamp: "2022-06-03T13:53:08.028Z", endTimeStamp: "2022-06-03T14:02:35.766Z"), editConfirmationStatus: "NOT_CONFIRMED", location: (latitudeE7: 191737554, longitudeE7: 728608371, placeId: "ChIJ0SLSusa35zsR9q77F0mNg0", address: "General Arun Kumar Vaidya Marg Shop R1, Third Floor,dham, Goregaon, Mumbai, Maharashtra 400063, India", locationConfidence: 77, otherCandidateLocations: (8) [(), {}, {}, {}, {}, {}, {}, {}], placeConfidence: "MEDIUM_CONFIDENCE", placeVisitImportance: "MAIN", placeVisitType: "SINGLE_PLACE", visitConfidence: 96}, [[Prototype]]: Object
      ↳ 1: {location: (lat: 191737554, lon: 728608371), duration: (startTimestamp: "2022-06-03T13:53:08.028Z", endTimeStamp: "2022-06-03T14:02:35.766Z"), editConfirmationStatus: "NOT_CONFIRMED", placeId: "ChIJ0SLSusa35zsR9q77F0mNg0", address: "General Arun Kumar Vaidya Marg Shop R1, Third Floor,dham, Goregaon, Mumbai, Maharashtra 400063, India", locationConfidence: 77, otherCandidateLocations: (8) [(), {}, {}, {}, {}, {}, {}, {}], placeConfidence: "MEDIUM_CONFIDENCE", placeVisitImportance: "MAIN", placeVisitType: "SINGLE_PLACE", visitConfidence: 96}, [[Prototype]]: Object
      ↳ 2: {location: (lat: 191739383, lon: 7286084974), duration: (startTimestamp: "2022-06-03T13:53:08.028Z", endTimeStamp: "2022-06-03T14:02:35.766Z"), editConfirmationStatus: "NOT_CONFIRMED", placeId: "ChIJ0SLSusa35zsR9q77F0mNg0", address: "General Arun Kumar Vaidya Marg Shop R1, Third Floor,dham, Goregaon, Mumbai, Maharashtra 400063, India", locationConfidence: 77, otherCandidateLocations: (8) [(), {}, {}, {}, {}, {}, {}, {}], placeConfidence: "MEDIUM_CONFIDENCE", placeVisitImportance: "MAIN", placeVisitType: "SINGLE_PLACE", visitConfidence: 96}, [[Prototype]]: Object
      ↳ 3: {centerLatE7: 191737554, centerLangE7: 7286085168, placeId: "ChIJ__vwe35zsRa7ReftEyWo", address: "Western Express Highway Goregaon East, Oberoi Garden City, Goregaon, Mumbai, Maharashtra 400063, India", locationConfidence: 81, otherCandidateLocations: (4) [(), {}, {}, {}], placeConfidence: "HIGH_CONFIDENCE", placeVisitImportance: "MAIN", placeVisitType: "SINGLE_PLACE", visitConfidence: 96}, [[Prototype]]: Object
    ↳ length: 4
  ↳ [[Prototype]]: Array(0)
  ↳ editConfirmationStatus: "NOT_CONFIRMED"
  ↳ location: (latitudeE7: 191737554, longitudeE7: 728608371, placeId: "ChIJ0SLSusa35zsR9q77F0mNg0", address: "General Arun Kumar Vaidya Marg Shop R1, Third Floor,dham, Goregaon, Mumbai, Maharashtra 400063, India", name: "Oberoi mall", locationConfidence: 81, otherCandidateLocations: (4) [(), {}, {}, {}], placeConfidence: "HIGH_CONFIDENCE", placeVisitImportance: "MAIN", placeVisitType: "SINGLE_PLACE", simplifiedRoadPath: (points: Array(14), source: "INFERRED", distanceMetres: 314.0213031589879)}, [[Prototype]]: Object
```

Show Details

Experiment No. 9 - Spatial and Temporal Data Handling

| START_DATE | START_TIME | DISTANCE | MODE | CONFIDENCE | PLACES |
|------------|------------|----------|----------------------|------------|--|
| 2022-06-03 | 04:26:00 | 1665 | IN_PASSENGER_VEHICLE | LOW | Oberoi Mall |
| 2022-06-03 | 03:02:00 | 1149 | IN_PASSENGER_VEHICLE | LOW | Malad |
| 2022-06-06 | 03:12:00 | 1041 | IN_PASSENGER_VEHICLE | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 03:31:00 | 9688 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 07:19:00 | 1146 | IN_PASSENGER_VEHICLE | LOW | Malad Station |
| 2022-06-06 | 07:29:00 | 1338 | IN_PASSENGER_VEHICLE | LOW | Vile Parle |
| 2022-06-06 | 07:49:00 | 9571 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 02:38:00 | 911 | MOTORCYCLING | LOW | Vile Parle Station |
| 2022-06-07 | 02:49:00 | 1034 | MOTORCYCLING | LOW | McDonald's |
| 2022-06-07 | 03:03:00 | 9641 | IN_TRAIN | LOW | Union Bank Of India |
| 2022-06-07 | 07:21:00 | 1305 | MOTORCYCLING | LOW | Rekha Steel Centre |
| 2022-06-07 | 07:35:00 | 1231 | IN_PASSENGER_VEHICLE | LOW | Bikaji Retail Store Malad East |
| 2022-06-07 | 07:53:00 | 9442 | IN_TRAIN | LOW | Surbhi Sweets and Snacks |
| 2022-06-07 | 02:18:00 | 928 | MOTORCYCLING | LOW | Union Bank Of India |
| 2022-06-08 | 02:43:00 | 9966 | IN_TRAIN | LOW | Bikaji Retail Store Malad East |
| 2022-06-08 | 06:30:00 | 1141 | MOTORCYCLING | LOW | M R Boutique |
| 2022-06-08 | 06:54:00 | 1159 | WALKING | HIGH | Ashoka General Hospital |
| 2022-06-08 | 07:11:00 | 9111 | IN_TRAIN | MEDIUM | Union Bank Of India |
| 2022-06-08 | 11:56:00 | 874 | MOTORCYCLING | LOW | Monica Hall |
| 2022-06-09 | 11:56:00 | 230 | MOTORCYCLING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-09 | 12:17:00 | 140 | MOTORCYCLING | LOW | Bikaji Retail Store Malad East |
| 2022-06-09 | 12:48:00 | 483 | MOTORCYCLING | LOW | "Navjeevan School |
| 2022-06-09 | 13:06:00 | 250 | MOTORCYCLING | MEDIUM | Rekha Steel Centre |
| 2022-06-09 | 08:50:00 | 544 | MOTORCYCLING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-10 | 09:30:00 | 359 | MOTORCYCLING | LOW | Ashoka General Hospital |
| 2022-06-10 | 09:42:00 | 530 | IN_PASSENGER_VEHICLE | LOW | Malad |
| 2022-06-10 | 10:18:00 | 450 | WALKING | LOW | Surbhi Sweets and Snacks |
| 2022-06-10 | 11:20:00 | 501 | WALKING | LOW | Vile Parle |
| 2022-06-11 | 11:38:00 | 322 | WALKING | MEDIUM | McDonald's |
| 2022-06-11 | 11:50:00 | 176 | WALKING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-11 | 11:59:00 | 4413 | MOTORCYCLING | LOW | Surbhi Sweets and Snacks |
| 2022-06-14 | 12:38:00 | 806 | MOTORCYCLING | MEDIUM | McDonald's |
| 2022-06-14 | 12:51:00 | 557 | MOTORCYCLING | MEDIUM | Union Bank Of India |
| 2022-06-14 | 04:03:00 | 287 | MOTORCYCLING | LOW | Vile Parle |
| 2022-06-15 | 04:16:00 | 1030 | CYLCLING | LOW | Monica Hall |
| 2022-06-15 | 04:39:00 | 9661 | IN_TRAIN | HIGH | Bandra Station |
| 2022-06-15 | 08:10:00 | 1133 | MOTORCYCLING | LOW | Malad Station |
| 2022-06-15 | 08:27:00 | 838 | IN_PASSENGER_VEHICLE | LOW | Oberoi Mall |
| 2022-06-15 | 08:38:00 | 122 | WALKING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-15 | 09:00:00 | 5003 | IN_TRAIN | LOW | Malad Station |
| 2022-06-15 | 10:11:00 | 660 | IN_PASSENGER_VEHICLE | LOW | Vile Parle Station |
| 2022-06-15 | 10:22:00 | 770 | WALKING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-15 | 05:25:00 | 15342 | IN_TRAIN | LOW | Ashoka General Hospital |

```
MySQL 8.0 Command Line Client

mysql> select *from 2022_JUNE WHERE START_TIME < "10:00:00";
+-----+-----+-----+-----+-----+-----+-----+
| START_DATE | START_TIME | DISTANCE | MODE | CONFIDENCE | PLACES |
+-----+-----+-----+-----+-----+-----+
| 2022-06-03 | 04:26:00 | 1665 | IN_PASSENGER_VEHICLE | LOW | Oberoi Mall |
| 2022-06-03 | 03:02:00 | 1149 | IN_PASSENGER_VEHICLE | LOW | Malad |
| 2022-06-06 | 03:12:00 | 1041 | IN_PASSENGER_VEHICLE | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 03:31:00 | 9680 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 07:19:00 | 1146 | IN_PASSENGER_VEHICLE | LOW | Malad Station |
| 2022-06-06 | 07:29:00 | 1338 | IN_PASSENGER_VEHICLE | LOW | Vile Parle |
| 2022-06-06 | 07:49:00 | 9571 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 02:38:00 | 911 | MOTORCYCLING | LOW | Vile Parle Station |
| 2022-06-07 | 02:49:00 | 1034 | MOTORCYCLING | LOW | McDonald's |
| 2022-06-07 | 03:03:00 | 9641 | IN_TRAIN | LOW | Union Bank Of India |
| 2022-06-07 | 07:21:00 | 1305 | MOTORCYCLING | LOW | Rekha Steel Centre |
| 2022-06-07 | 07:35:00 | 1231 | IN_PASSENGER_VEHICLE | LOW | Bikaji Retail Store Malad East |
| 2022-06-07 | 07:53:00 | 9442 | IN_TRAIN | LOW | Surbhi Sweets and Snacks |
| 2022-06-07 | 02:18:00 | 928 | MOTORCYCLING | LOW | Union Bank Of India |
| 2022-06-08 | 02:43:00 | 9966 | IN_TRAIN | LOW | Bikaji Retail Store Malad East |
| 2022-06-08 | 06:30:00 | 1141 | MOTORCYCLING | LOW | M R Boutique |
| 2022-06-08 | 06:54:00 | 1159 | WALKING | HIGH | Ashoka General Hospital |
| 2022-06-08 | 07:11:00 | 9111 | IN_TRAIN | MEDIUM | Union Bank Of India |
| 2022-06-09 | 08:50:00 | 544 | MOTORCYCLING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-10 | 09:30:00 | 359 | MOTORCYCLING | LOW | Ashoka General Hospital |
| 2022-06-10 | 09:42:00 | 530 | IN_PASSENGER_VEHICLE | LOW | Malad |
| 2022-06-14 | 04:03:00 | 287 | MOTORCYCLING | LOW | Vile Parle |
| 2022-06-15 | 04:16:00 | 1030 | CYCLING | LOW | Monica Hall |
| 2022-06-15 | 04:39:00 | 9661 | IN_TRAIN | HIGH | Bandra Station |
| 2022-06-15 | 08:10:00 | 1133 | MOTORCYCLING | LOW | Malad Station |
| 2022-06-15 | 08:27:00 | 838 | IN_PASSENGER_VEHICLE | LOW | Oberoi Mall |
| 2022-06-15 | 08:38:00 | 122 | WALKING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-15 | 09:00:00 | 5003 | IN_TRAIN | LOW | Malad Station |
| 2022-06-15 | 05:25:00 | 15342 | IN_TRAIN | LOW | Ashoka General Hospital |
+-----+-----+-----+-----+-----+-----+
29 rows in set (0.00 sec)
```

```
mysql> select *from 2022_JUNE WHERE CONFIDENCE="MEDIUM";
+-----+-----+-----+-----+-----+-----+
| START_DATE | START_TIME | DISTANCE | MODE | CONFIDENCE | PLACES |
+-----+-----+-----+-----+-----+-----+
| 2022-06-06 | 03:31:00 | 9680 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 07:49:00 | 9571 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-08 | 07:11:00 | 9111 | IN_TRAIN | MEDIUM | Union Bank Of India |
| 2022-06-09 | 13:06:00 | 250 | MOTORCYCLING | MEDIUM | Rekha Steel Centre |
| 2022-06-09 | 08:50:00 | 544 | MOTORCYCLING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-11 | 11:38:00 | 322 | WALKING | MEDIUM | McDonald's |
| 2022-06-11 | 11:50:00 | 176 | WALKING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-14 | 12:38:00 | 806 | MOTORCYCLING | MEDIUM | McDonald's |
| 2022-06-14 | 12:51:00 | 557 | MOTORCYCLING | MEDIUM | Union Bank Of India |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

```
mysql> select *from 2022_JUNE WHERE PLACES = "SVKM's Dwarkadas J. Sanghvi College of Engineering";
+-----+-----+-----+-----+-----+-----+
| START_DATE | START_TIME | DISTANCE | MODE | CONFIDENCE | PLACES |
+-----+-----+-----+-----+-----+-----+
| 2022-06-06 | 03:12:00 | 1041 | IN_PASSENGER_VEHICLE | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 03:31:00 | 9680 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 07:49:00 | 9571 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-09 | 11:56:00 | 230 | MOTORCYCLING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-09 | 08:50:00 | 544 | MOTORCYCLING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-11 | 11:50:00 | 176 | WALKING | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-15 | 08:38:00 | 122 | WALKING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-15 | 10:22:00 | 770 | WALKING | LOW | SVKM's Dwarkadas J. Sanghvi College of Engineering |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> select *from 2022_JUNE WHERE DISTANCE > 2000;
+-----+-----+-----+-----+-----+-----+
| START_DATE | START_TIME | DISTANCE | MODE | CONFIDENCE | PLACES |
+-----+-----+-----+-----+-----+-----+
| 2022-06-06 | 03:31:00 | 9680 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-06 | 07:49:00 | 9571 | IN_TRAIN | MEDIUM | SVKM's Dwarkadas J. Sanghvi College of Engineering |
| 2022-06-07 | 03:03:00 | 9641 | IN_TRAIN | LOW | Union Bank Of India |
| 2022-06-07 | 07:53:00 | 9442 | IN_TRAIN | LOW | Surbhi Sweets and Snacks |
| 2022-06-08 | 02:43:00 | 9966 | IN_TRAIN | LOW | Bikaji Retail Store Malad East |
| 2022-06-08 | 07:11:00 | 9111 | IN_TRAIN | MEDIUM | Union Bank Of India |
| 2022-06-11 | 11:50:00 | 4413 | MOTORCYCLING | LOW | Surbhi Sweets and Snacks |
| 2022-06-15 | 04:39:00 | 9661 | IN_TRAIN | HIGH | Bandra Station |
| 2022-06-15 | 09:00:00 | 5003 | IN_TRAIN | LOW | Malad Station |
| 2022-06-15 | 05:25:00 | 15342 | IN_TRAIN | LOW | Ashoka General Hospital |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```