Rosita D'mello
60004200137
Batch: B3

ADBMS Experiment 1

25/25

Date of Performance: 22/9/22
Date of Submission: 29/9/22

Aim: To perform a case study on performance and commercial database

Database: MongoDB used by eBay.

eBay uses MongoDB to carry out multiple tasks handling a great amount of data. Such projects include categorization of merchandise, storage of metadata, cloud management, and search suggestions.

MongoDB is a distributed, document-based database that is well-suited for applications and clouds. It supports JSON documents, while offering flexible and dynamic schemas. It has a strong query language, using which, you can filter, sort, and search data.

Company:

eBay is a multinational e-commerce website that lets consumers sell to other consumers as well as businesses sell to consumers. The eBay website is an online auction and shopping website. Sellers upload their product image and details, and the buyers can buy it or bid for it.

Search suggestions are a key feature of the eBay website. This feature requires storing a large no. of possible suggestions and making it available to

the user speedily. They realized that any database query to get suggestions must make the round trip in less than 60-70 milliseconds. eBay has decided to use MongoDB to achieve this.

They made a MongoDB document out of the search suggestion list. They indexed this document using ~~a prefix~~ keyword prefix. Additionally, they also used pieces of metadata such as product category. Using multiple indices offered not just flexibility but also great speed. They also placed data in the memory, boosting their speed furthermore.

Comparison with proposed database:

The database I propose for eBay is PostgreSQL, (or Postgres). It is a free and open-source RDBMS. Postgres features transactions with the ACID properties. It is designed to handle a range of workloads, from single machines to data warehouses.

1. In the transaction performance testing conducted by OnGres, Postgres measured between 4 and 15 times faster than MongoDB. With 753.3M users a month, multiple transactions may be taking place in a span of a few minutes. A greater transaction speed will only prove to be a boon to eBay.

2. PostgreSQL is relational. One of the reasons

Why companies used to prefer MongoDB is because it was more or less schemaless. JSON format # allowed them to be flexible with the kind of data entered into the website with growing features. But now, PostgreSQL has an amazing native JSON and key/value support. So you can use Hstore/JSON column and leave modelling your domain until you already have a heap of data. You can still use schema on the other tables.

3. Since it is an eCommerce site and the kind of content uploaded is more or less the same - image of product, description in case of sellers, and a cart, payment in case of buyers, they could benefit through a database with a proper schema. Incorrect data is better handled in relational databases.

4. Postgres NoSQL also enables interaction with other sources of data, while MongoDB does not. So, Postgres can extract data from Oracle, MySQL, mongoDB, Redis, Twitter, etc., making the switch easier.

5. Enterprise DB found that Postgres outperforms MongoDB in selecting, loading, and inserting complex document data in key workloads involving 50 million records.
→ Ingestion of high volumes of data was approx 2.1 times faster in Postgres
→ MongoDB consumed 33% more the disk space.

→ Data inserts took almost 3 times longer in MDB
→ Data selections took more than 2.5 times longer

Recommendation:

→ Schemas allow data to be consistent across lots of new versions of the app through its development, serve as a documentation, prevent bugs. You don't have to post process tons of data.

→ A good SQL DB like Postgres is written in a low level language that compiles very fast.

→ Postgres has implemented NoSQL features through the introduction of Hstore and JSONB. Postgres offers validation of data for any JSON field, so entering any incorrect data would eventually lead to an error.

→ PostgreSQL can easily manage the most compelling requests of the biggest companies and institutions. It never stops optimizing its performance.

Conclusion:

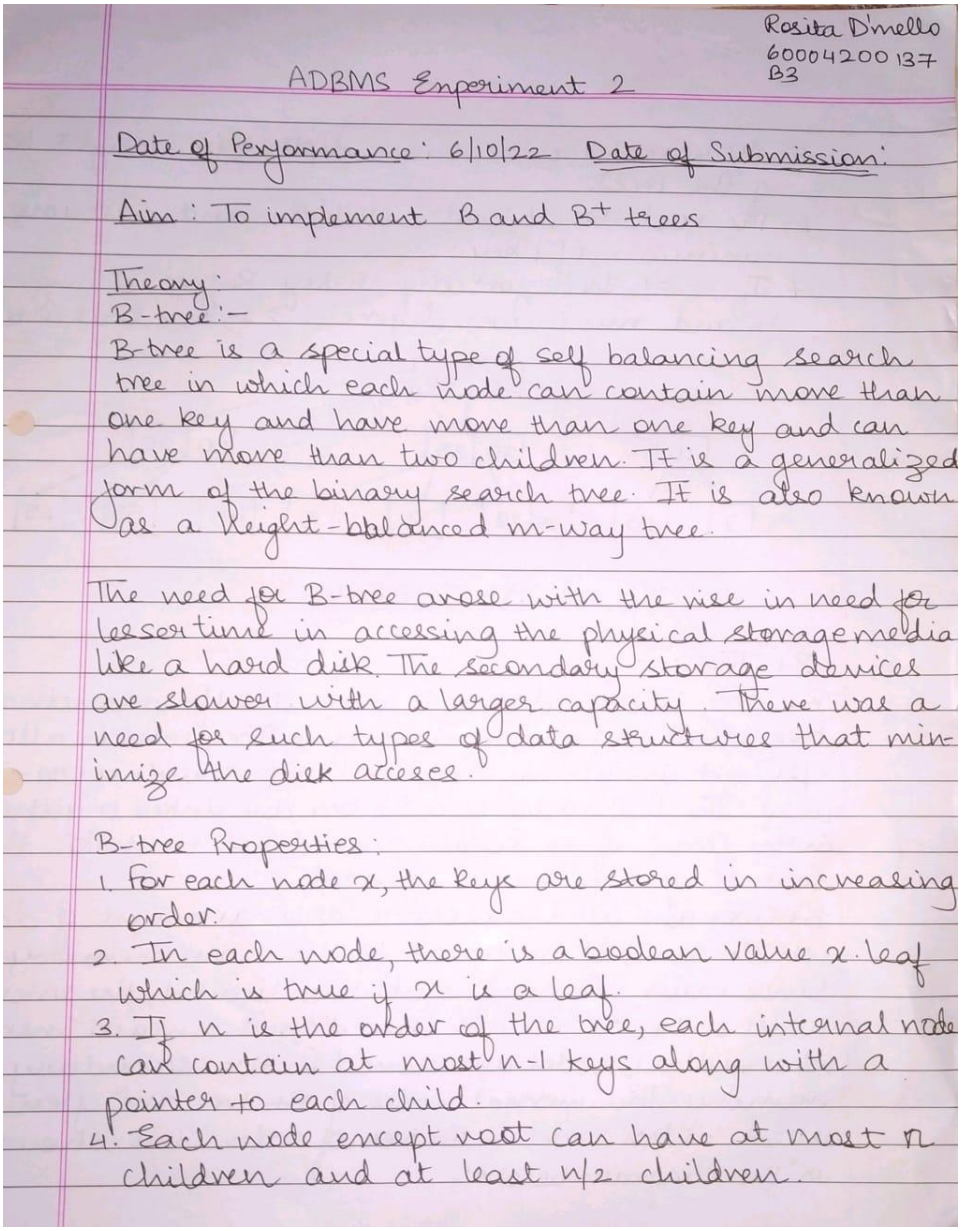
PostgreSQL, an advanced enterprise class open source relational database backed by over 30 years of community development is the backbone to many technologies already, there's no harm in making eBay a part of them as well.

ADBMS Experiment 2

Date of Performance: 6th October 2022
Date of Submission: 22nd October 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Optimize using B and B+ trees
Theory:



Rosita D'mello
60004200137
B3

ADBMS Experiment 2

Date of Performance: 6/10/22 Date of Submission:

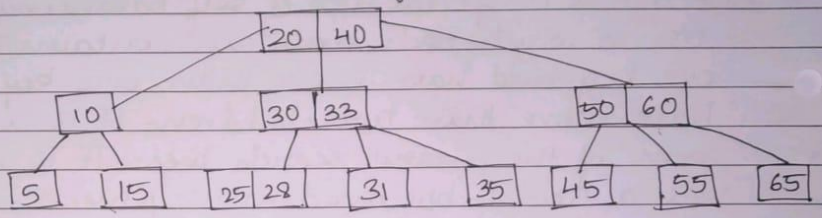
Aim: To implement B and B+ trees

Theory:
B-tree:-
B-tree is a special type of self balancing search tree in which each node can contain more than one key and have more than one key and can have more than two children. It is a generalized form of the binary search tree. It is also known as a Height-balanced m-way tree.

The need for B-tree arose with the rise in need for lesser time in accessing the physical storage media like a hard disk. The secondary storage devices are slower with a larger capacity. There was a need for such types of data structures that minimize the disk accesses.

B-tree Properties:
1. For each node x, the keys are stored in increasing order.
2. In each node, there is a boolean value x.leaf which is true if x is a leaf.
3. If n is the order of the tree, each internal node can contain at most n-1 keys along with a pointer to each child.
4. Each node except root can have at most n children and at least n/2 children.

5. All leaves have the same depth (i.e. height-h of the tree).
6. The root has at least 2 children and contains a minimum of 1 key.
7. If $n \geq 1$ then for any n-key B-tree of height h and minimum degree $t \geq 2$, $h \geq \log_t \frac{(n+1)}{2}$



B-TREE

B+ Tree:—

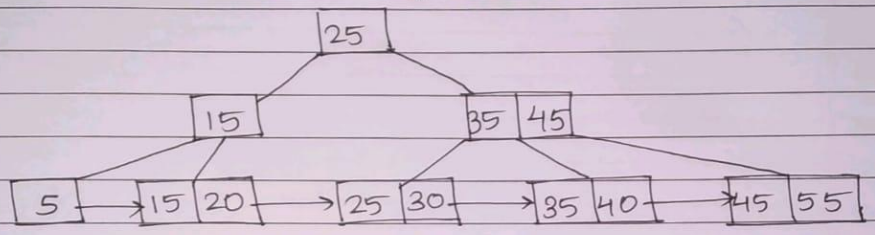
B+ Tree is an advanced form of self-balancing tree which is an extension of B Tree which allows efficient insertion, deletion, and search operations. The leaf nodes of a B+ tree are linked together in the form of a singly linked list to make the search queries more efficient.

B+ Tree are used to store a large amount of data which can not be stored in the main memory. Since main memory is always limited, the internal nodes of the B+ tree are stored in the main memory whereas leaf nodes are stored in the secondary memory. The internal nodes of B+ tree are often called index nodes. A B+ tree of order 3 is shown in the diagram.

~~Advantages:~~

Applications of B+ Trees:

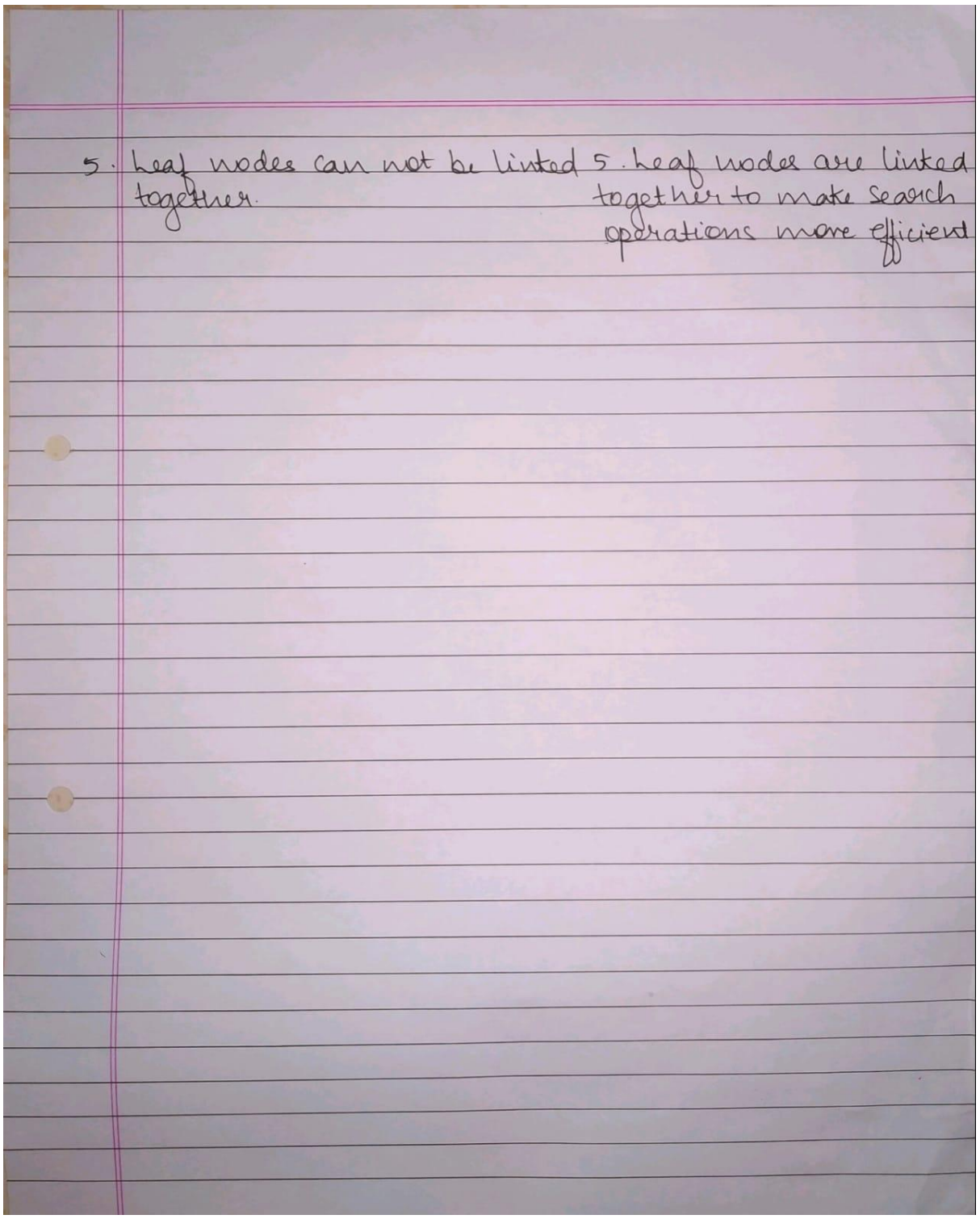
1. Multilevel indexing
2. Faster operations on the tree (insertion, deletion, search)
3. Database indexing.



B+ TREE

Comparison:—

B Tree	B+ Tree.
1. Search keys cannot be repeatedly stored.	1. Redundant search keys can be present.
2. Data can be stored in leaf nodes as well as internal nodes.	2. Data can only be stored on the leaf nodes.
3. Searching for some data is a slower process since data can be found on internal nodes as well as on the leaf nodes.	3. Searching is comparatively faster as data can only be found on the leaf nodes.
4. Deletion of internal nodes are so complicated and time consuming.	4. Deletion will never be a complex process since element will always be deleted from the leaf nodes.



Code:

B tree:

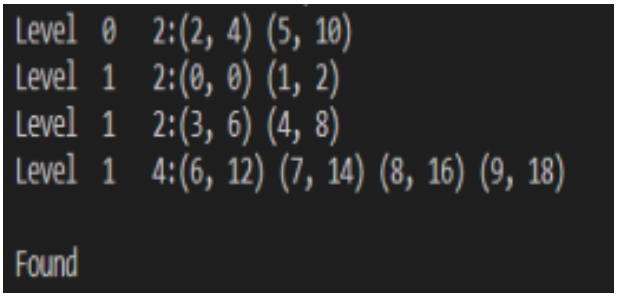
```
class BTreeNode:
 def __init__(self, leaf=False):
 self.leaf = leaf
 self.keys = []
```

```
 self.child = []
class BTree:
 def __init__(self, t):
 self.root = BTreeNode(True)
 self.t = t

 def insert(self, k):
 root = self.root
 if len(root.keys) == (2 * self.t) - 1:
 temp = BTreeNode()
 self.root = temp
 temp.child.insert(0, root)
 self.split_child(temp, 0)
 self.insert_non_full(temp, k)
 else:
 self.insert_non_full(root, k)

 def insert_non_full(self, x, k):
 i = len(x.keys) - 1
 if x.leaf:
 x.keys.append((None, None))
 while i >= 0 and k[0] < x.keys[i][0]:
 x.keys[i + 1] = x.keys[i]
 i -= 1
 x.keys[i + 1] = k
 else:
 while i >= 0 and k[0] < x.keys[i][0]:
 i -= 1
 i += 1
 if len(x.child[i].keys) == (2 * self.t) - 1:
 self.split_child(x, i)
 if k[0] > x.keys[i][0]:
 i += 1
 self.insert_non_full(x.child[i], k)

 def split_child(self, x, i):
 t = self.t
 y = x.child[i]
```

```
 z = BTreeNode(y.leaf)
 x.child.insert(i + 1, z)
 x.keys.insert(i, y.keys[t - 1])
 z.keys = y.keys[t: (2 * t) - 1]
 y.keys = y.keys[0: t - 1]
 if not y.leaf:
 z.child = y.child[t: 2 * t]
 y.child = y.child[0: t - 1]

 def print_tree(self, x, l=0):
 print("Level ", l, " ", len(x.keys), end=":")
 for i in x.keys:
 print(i, end=" ")
 print()
 l += 1
 if len(x.child) > 0:
 for i in x.child:
 self.print_tree(i, l)

 def search_key(self, k, x=None):
 if x is not None:
 i = 0
 while i < len(x.keys) and k > x.keys[i][0]:
 i += 1
 if i < len(x.keys) and k == x.keys[i][0]:
 return (x, i)
 elif x.leaf:
 return None
 else:
 return self.search_key(k, x.child[i])

 else:
 return self.search_key(k, self.root)
def main():
 B = BTree(3)
 for i in range(10):
 B.insert((i, 2 * i))
 B.print_tree(B.root)
```

```
 if B.search_key(8) is not None:
 print("\nFound")
 else:
 print("\nNot Found")
if __name__ == '__main__':
 main()
```

Output:



B+ Tree:

```
mport random
from secrets import choice
splits = 0
parent_splits = 0
fusions = 0
parent_fusions = 0
class Node(object):
def __init__(self, parent=None):
self.keys: list = []
self.values: list[Node] = []
self.parent: Node = parent
def index(self, key):
for i, item in enumerate(self.keys):
if key < item:
return i
return len(self.keys)
def __getitem__(self, item):
return self.values[self.index(item)]
def __setitem__(self, key, value):
i = self.index(key)
self.keys[i:i] = [key]
self.values.pop(i)
self.values[i:i] = value
```

```
def split(self):
global splits, parent_splits
splits += 1
parent_splits += 1
left = Node(self.parent)
mid = len(self.keys) // 2
left.keys = self.keys[:mid]
left.values = self.values[:mid + 1]
for child in left.values:
child.parent = left
key = self.keys[mid]
self.keys = self.keys[mid + 1:]
self.values = self.values[mid + 1:]
return key, [left, self]
class Leaf(Node):
def __init__(self, parent=None, prev_node=None,
next_node=None):
super(Leaf, self).__init__(parent)
self.next: Leaf = next_node
if next_node is not None:
next_node.prev = self
self.prev: Leaf = prev_node
if prev_node is not None:
prev_node.next = self
def __getitem__(self, item):
return self.values[self.keys.index(item)]
def __setitem__(self, key, value):
i = self.index(key)
if key not in self.keys:
self.keys[i:i] = [key]
self.values[i:i] = [value]
else:
self.values[i - 1] = value
def split(self):
global splits
splits += 1
left = Leaf(self.parent, self.prev, self)
mid = len(self.keys) // 2
left.keys = self.keys[:mid]
left.values = self.values[:mid]
self.keys: list = self.keys[mid:]
self.values: list = self.values[mid:]
return self.keys[0], [left, self]
```

```
class BPlusTree(object):
root: Node
def __init__(self, maximum=4):
self.root = Leaf()
self.maximum: int = maximum if maximum > 2 else 2
self.minimum: int = self.maximum // 2
self.depth = 0
def find(self, key) -> Leaf:
node = self.root
while type(node) is not Leaf:
node = node[key]
return node
def __getitem__(self, item):
return self.find(item)[item]
def query(self, key):
leaf = self.find(key)
return leaf[key] if key in leaf.keys else None
def change(self, key, value):
leaf = self.find(key)
if key not in leaf.keys:
return False, leaf
else:
leaf[key] = value
return True, leaf
def __setitem__(self, key, value, leaf=None):
if leaf is None:
leaf = self.find(key)
leaf[key] = value
if len(leaf.keys) > self.maximum:
self.insert_index(*leaf.split())
def insert(self, key, value):
leaf = self.find(key)
if key in leaf.keys:
return False, leaf
else:
self.__setitem__(key, value, leaf)
return True, leaf
def insert_index(self, key, values: list[Node]):
parent = values[1].parent
if parent is None:
values[0].parent = values[1].parent = self.root = Node()
self.depth += 1
self.root.keys = [key]
```

```
self.root.values = values
return
parent[key] = values
if len(parent.keys) > self.maximum:
self.insert_index(*parent.split())
def show(self, node=None, file=None, _prefix="",
_last=True):
if node is None:
node = self.root
print(_prefix, "`- " if _last else "|- ", node.keys,
sep="",file=file)
_prefix += " " if _last else "| "
if type(node) is Node:
for i, child in enumerate(node.values):
_last = (i == len(node.values) - 1)
self.show(child, file, _prefix, _last)
def demo():
bplustree = BPlusTree()
flag = True
while (flag):
print("\n******** B+ Tree ********")
print("1.Insert")
print("2.Search")
print("3.Show Tree")
print("4.Exit")
n = int(input("Enter Your Choice : "))
if n == 1:
value = int(input("Enter Element to Insert : "))
bplustree[value] = value
elif n == 2:
value = int(input("Enter Element to Search : "))
ans = bplustree.find(value)
if ans.values:
print("Value found in the tree")
else:
print("Value not found in the tree")
elif n == 3:
bplustree.show()
elif n == 4:
flag = False
break
else:
flag = False
```

```
break
if __name__ == '__main__':
demo()
```

Output:

```
******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 1
Enter Element to Insert : 4

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 1
Enter Element to Insert : 9

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 1
Enter Element to Insert : 3

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 1
Enter Element to Insert : 8

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 1
Enter Element to Insert : 6
```

```
******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 3
`- [6]
 |- [3, 4]
 `- [6, 8, 9]

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 2
Enter Element to Search : 8
Value found in the tree

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 2
Enter Element to Search : 5
Value found in the tree

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 2
Enter Element to Search : 20
```

```
******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 2
Enter Element to Search : 20

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 3
`- [6]
 |- [3, 4]
 `- [6, 8, 9]

******** B+ Tree ********
1.Insert
2.Search
3.Show Tree
4.Exit
Enter Your Choice : 4
```

Conclusion: Thus, we learnt optimization using B and B+ trees

Rosita D'mello
60004200137
Batch B3.

ADBMS Experiment 3

Date of Performance: 20/10/22
Date of Submission: 22/10/22

Aim: To simulate query optimization by applying SQL query on database.

Theory:

Indexes are used to find rows with specific column values quickly. Without an index, MySQL must begin with the 1st row and then read through the entire table to find the relevant rows. The larger the table, the more this costs.

A SQL query can be written in many different ways. An optimized query also depends on how the data is stored in the file organization. The major purposes of SQL query optimization are:-

1] Reduced Response Time.
2] Reduced CPU Execution Time.
3] Improved Throughput.

There are 2 ways/approaches to Query Optimization:
A) Heuristic Based.

A query tree is a data structure that corresponds to relational algebra expressions. The same query could correspond to many different relational algebraic expressions and hence many different query trees. The task of heuristic optimization of query trees is to find a final query tree that is efficient to execute. The main heuristic is to apply

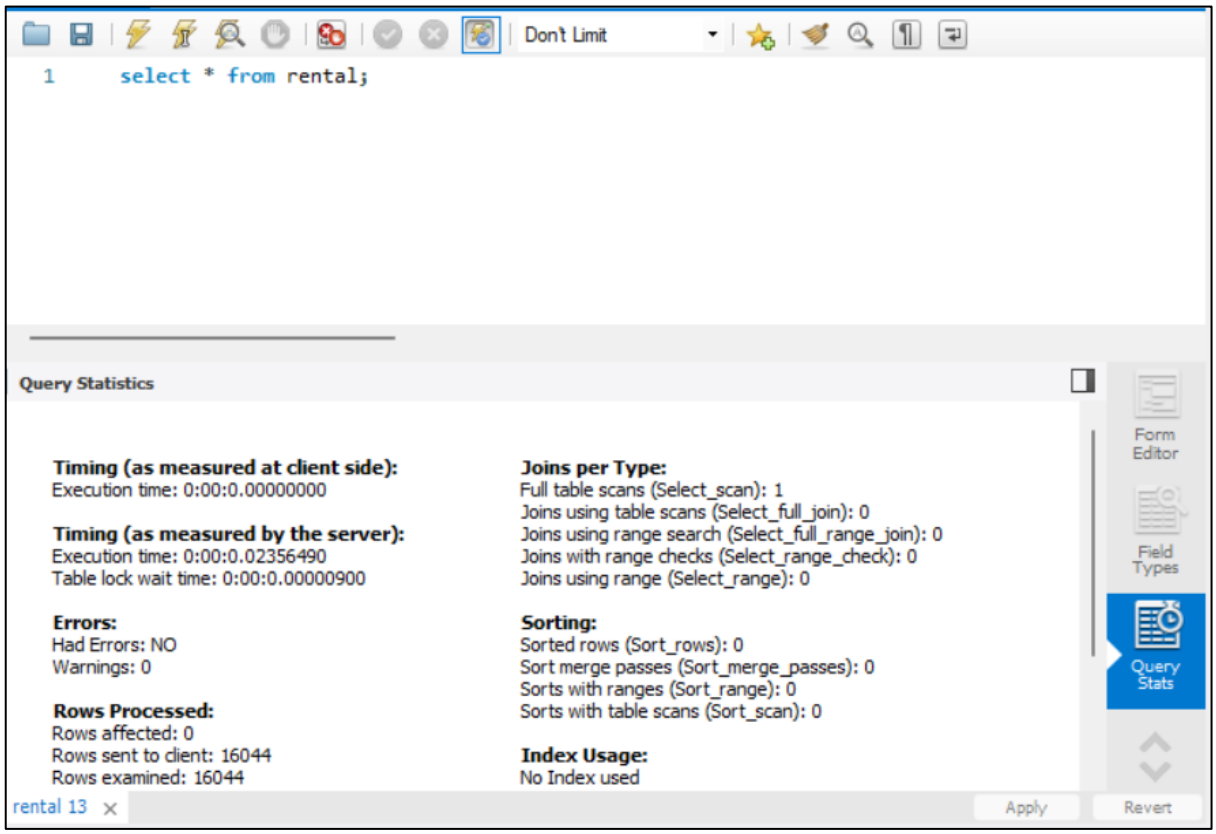
first the operations that reduce size of intermediate results.

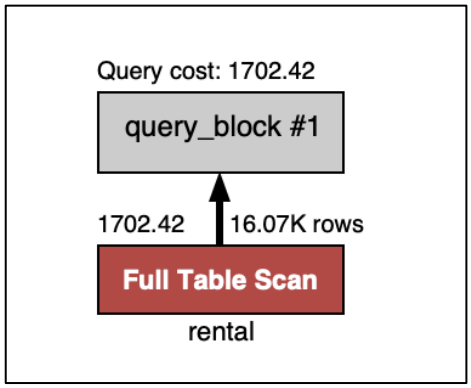
B) Cost Based Optimization.
Estimate and compare costs of executing a query using different executing strategies, choose the one with the lowest cost. The cost of any query includes access cost to secondary memory, storage cost, communication cost, memory usage cost, no. of memory buffers at the time of execution and computation cost.

Conclusion: In this experiment, we studied about different optimization techniques and implemented them. We performed table and index level optimization. A huge difference is noticed between an optimized and non optimized query.

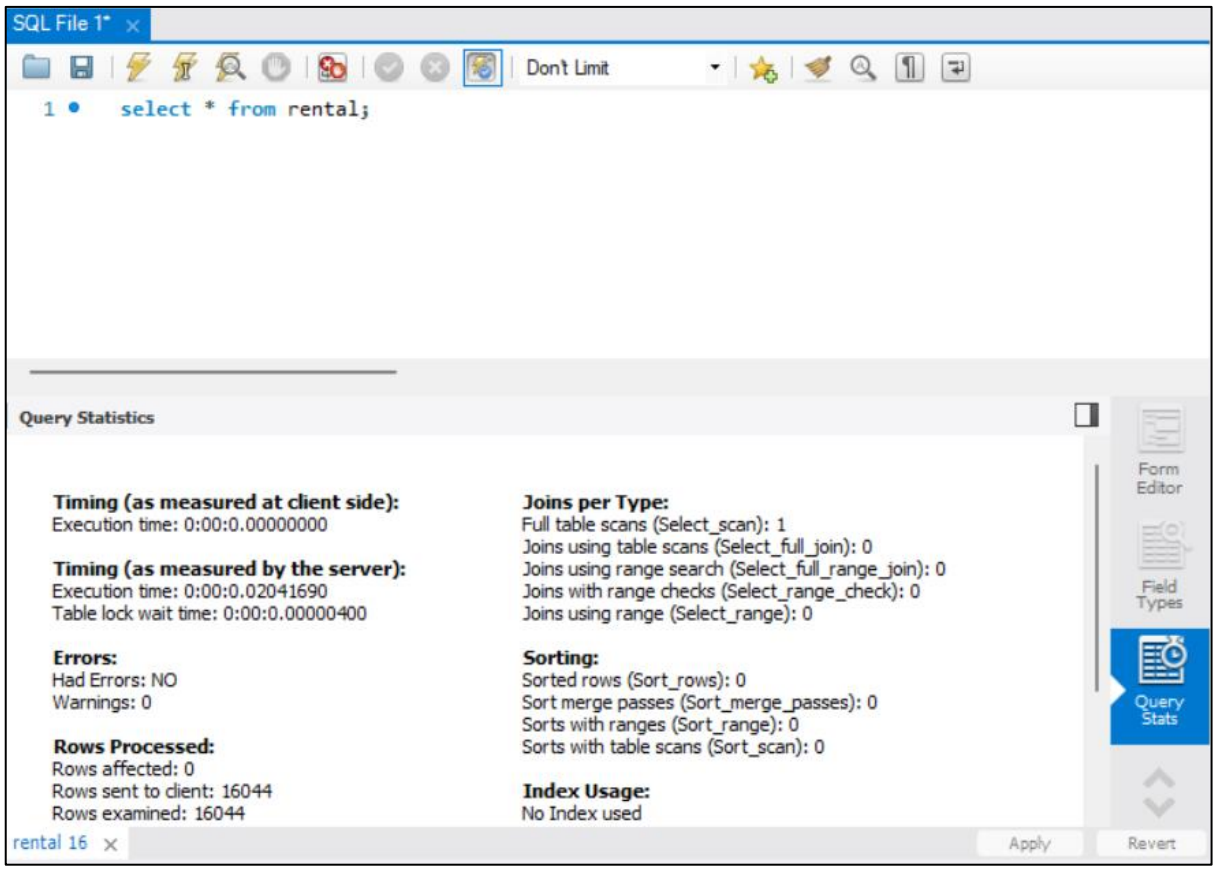
SELECT QUERY

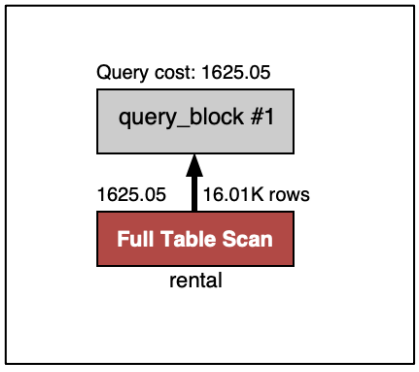
select * from rental;



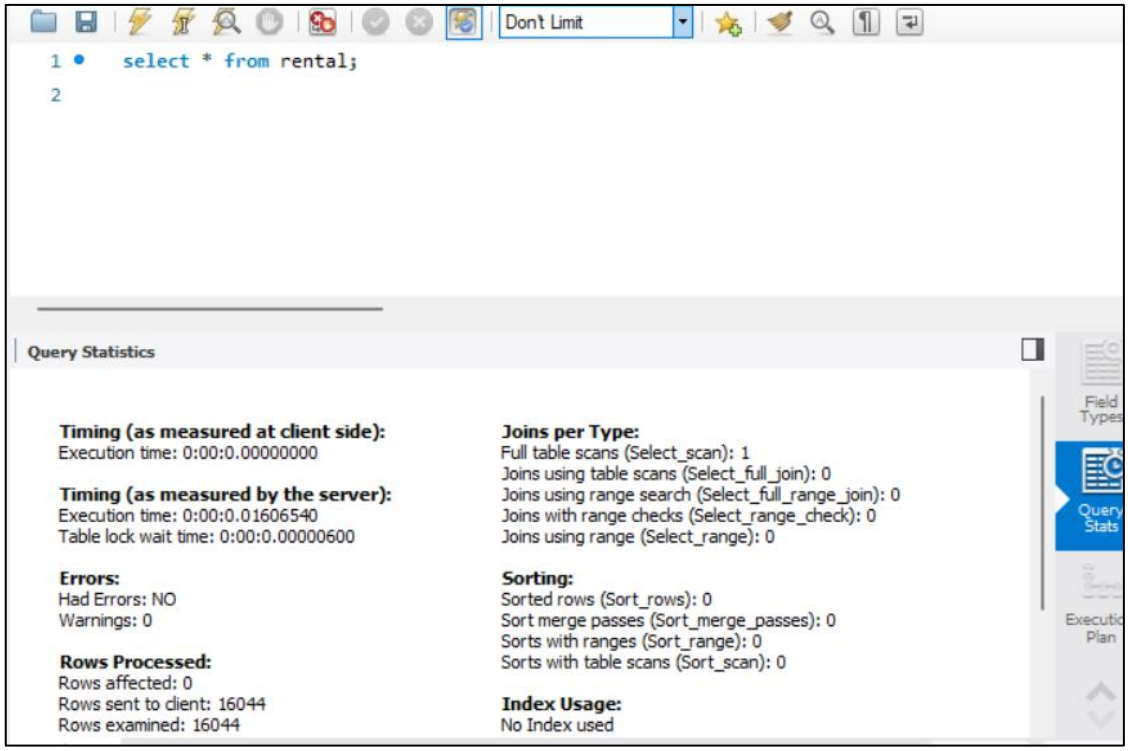


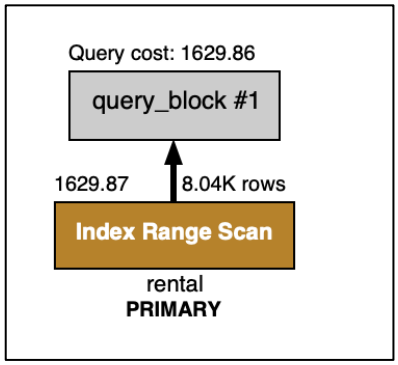
AFTER TABLE OPTIMIZATION:





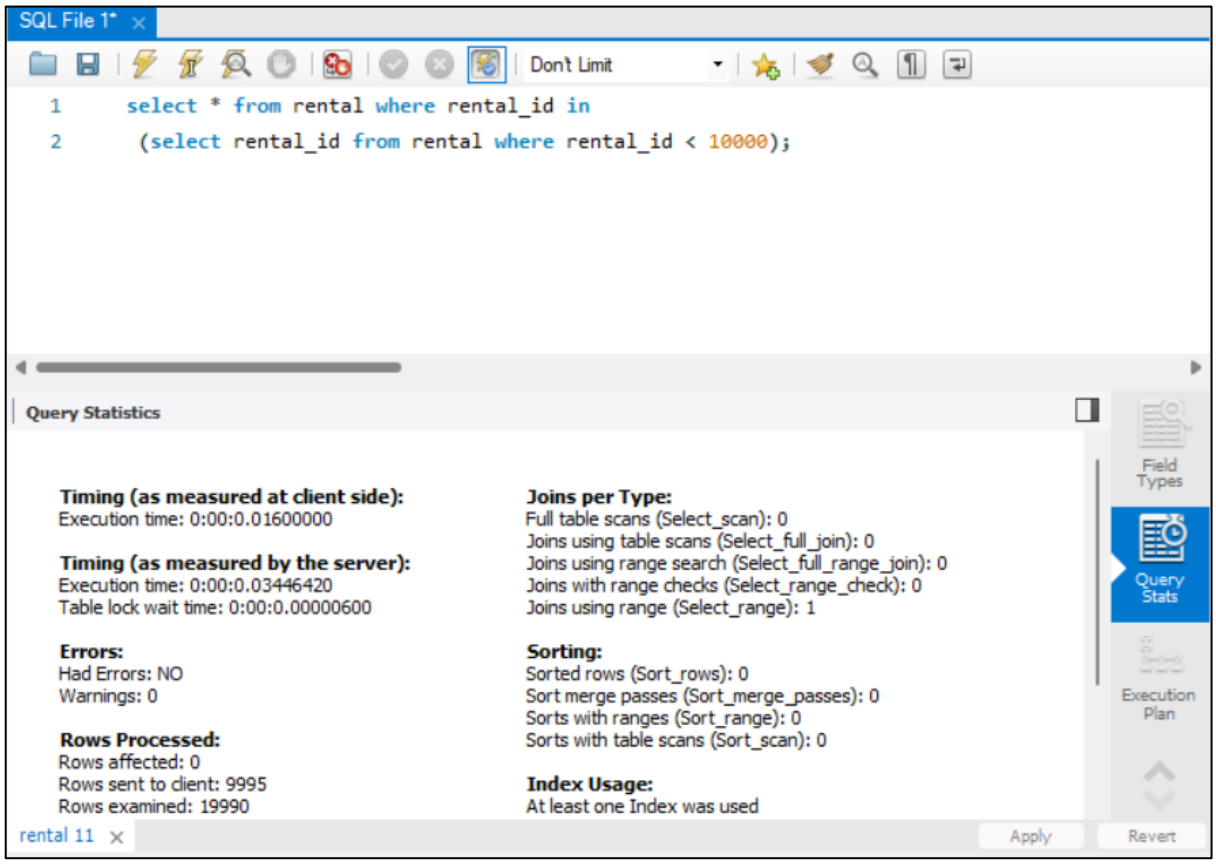
AFTER INDEX OPTIMIZATION:

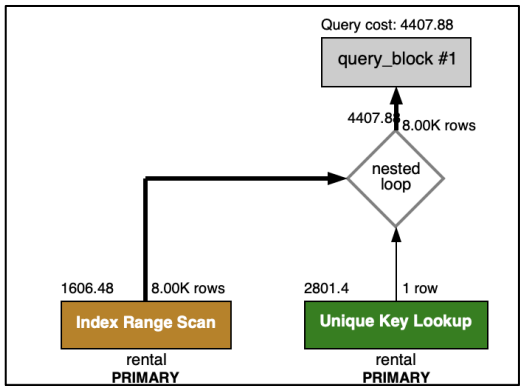




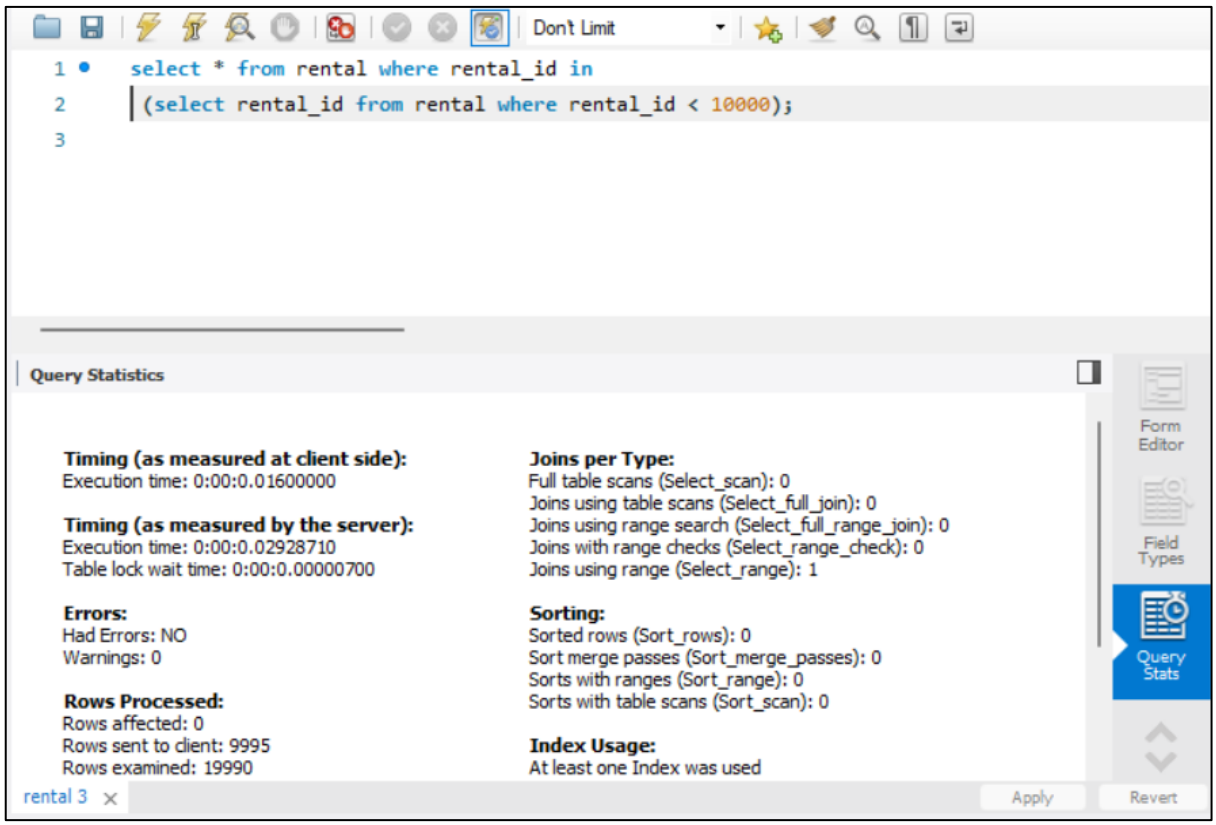
NESTED QUERY

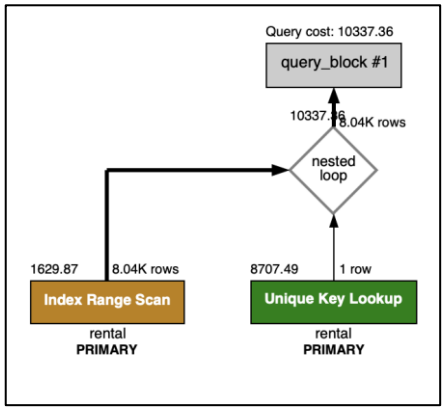
select * from rental where rental_id in (select rental_id from rental where rental_id < 10000);



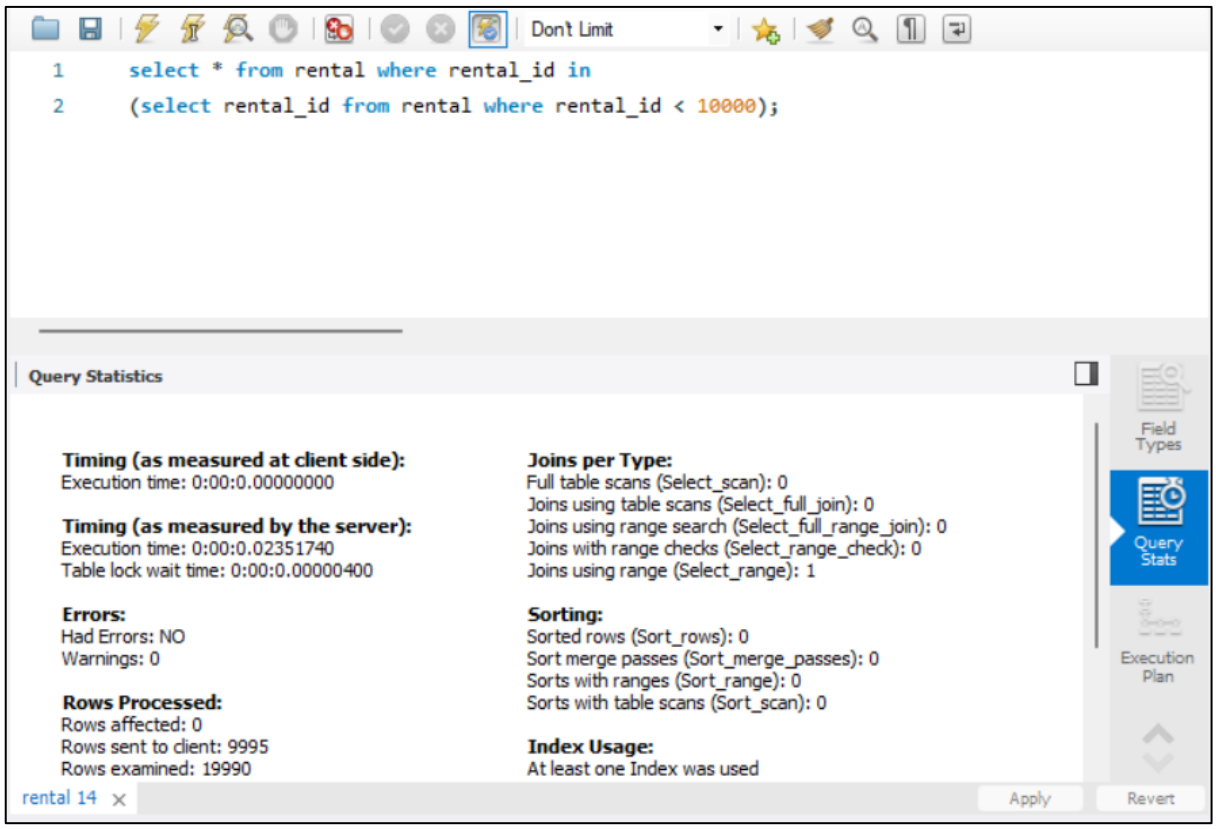


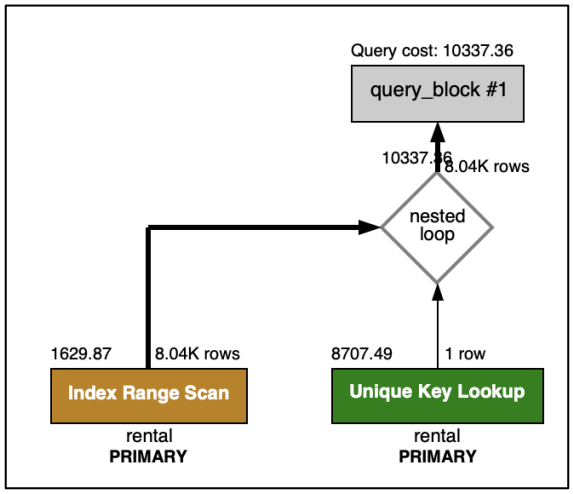
AFTER TABLE OPTIMIZATION:





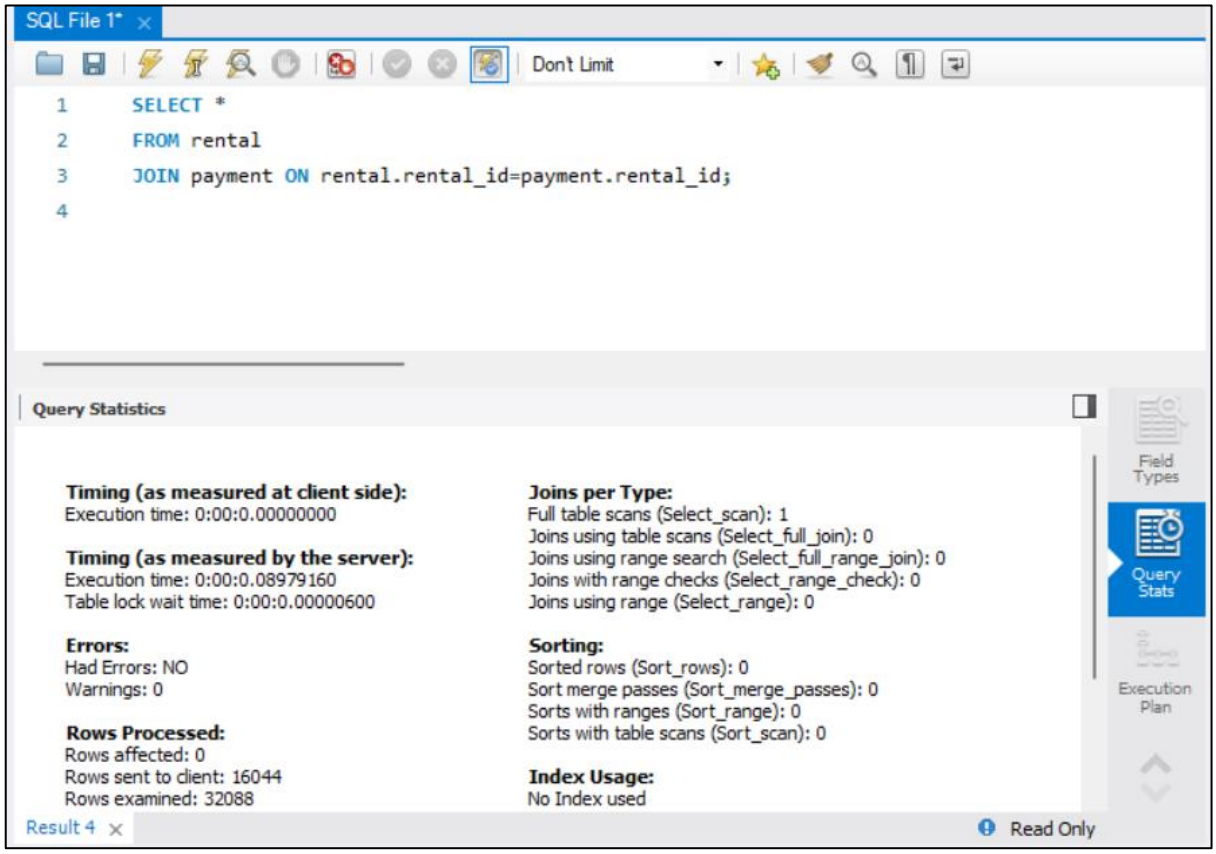
AFTER INDEX OPTIMIZATION:

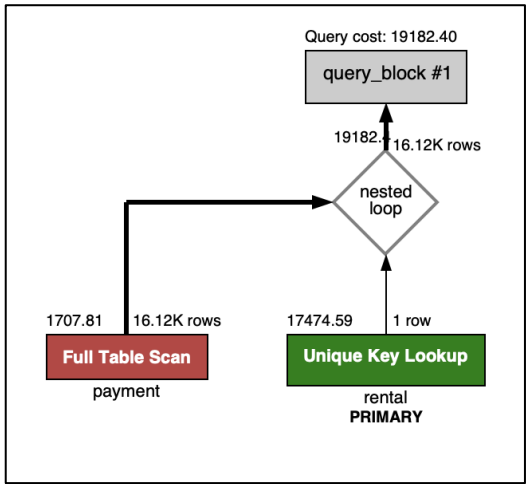




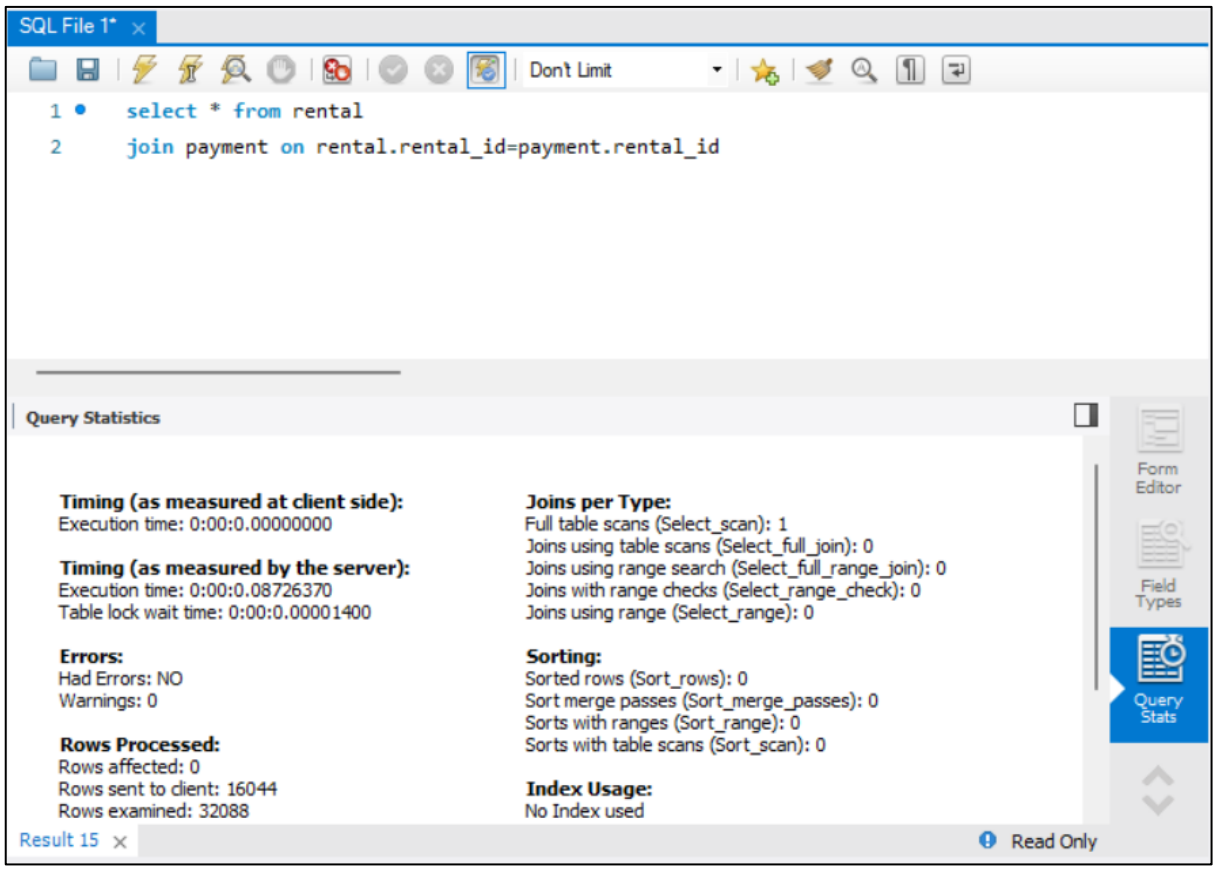
JOIN QUERY

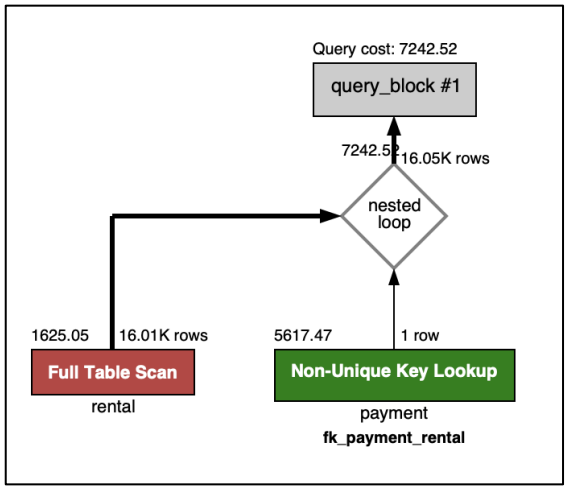
select * from rental join payment on rental.rental_id=payment.rental_id;



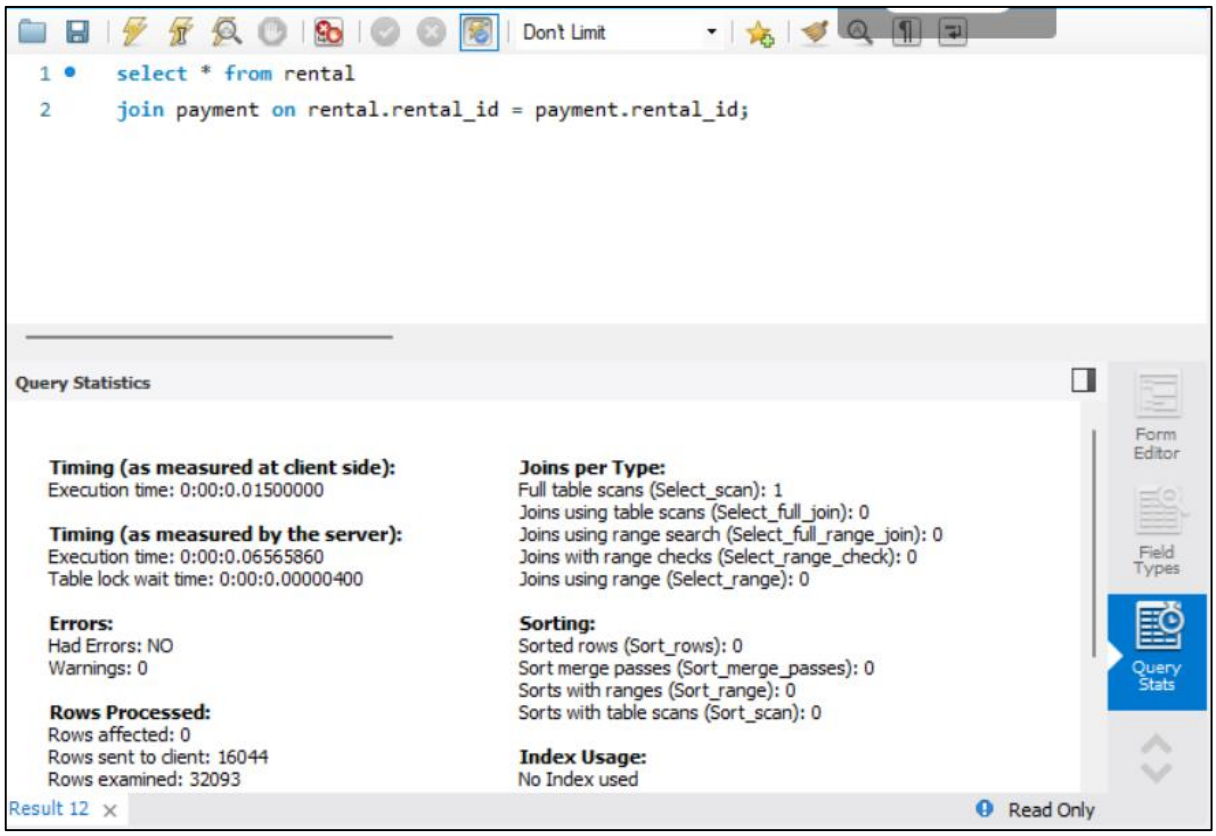


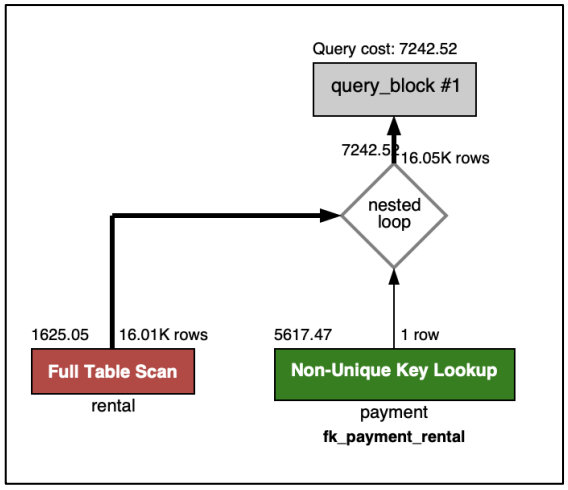
AFTER TABLE OPTIMIZATION:





AFTER INDEX OPTIMIZATION:





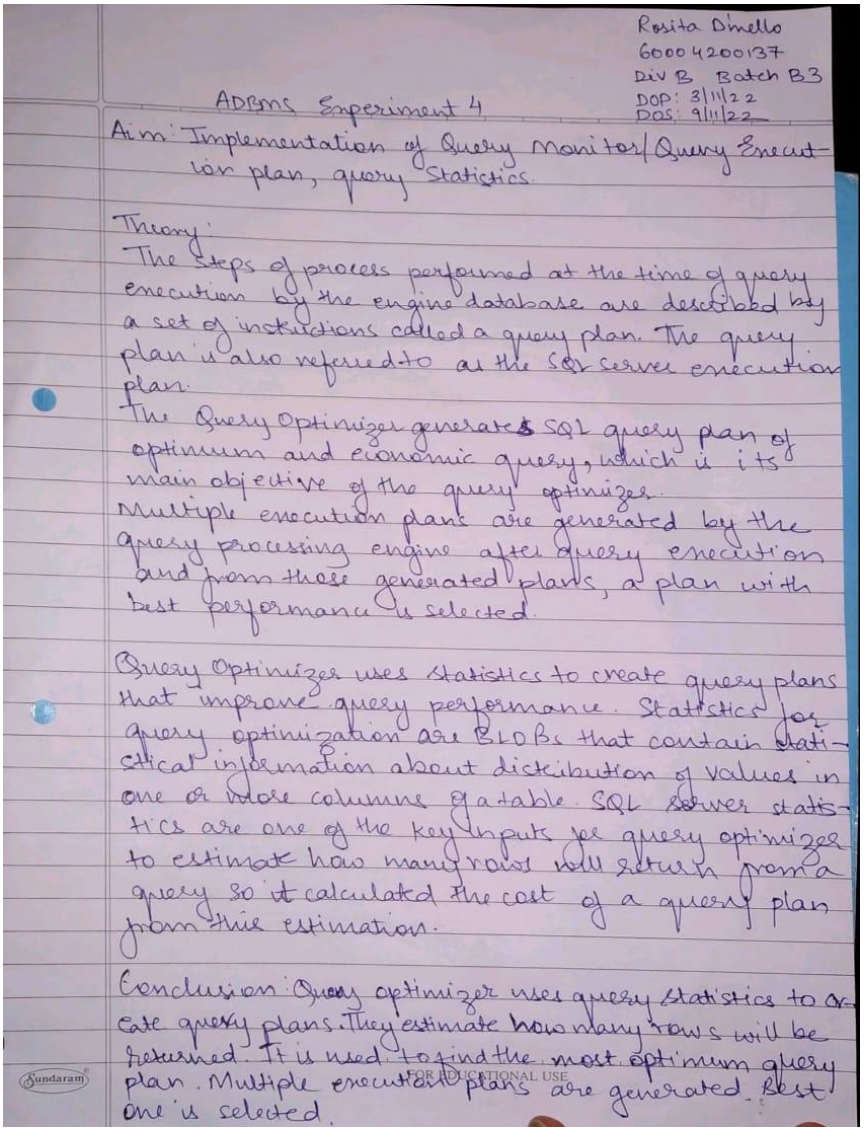
ADBMS Experiment 4

Date of Performance: 3rd November 2022
Date of Submission: 9th November 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Implement Query Monitor (QEP - Query Execution Plan, Query Statistics)

Theory:



Rosita D'mello
60004200137
Div B Batch B3
DOP: 3/11/22
DOS: 9/11/22

ADBMS Experiment 4

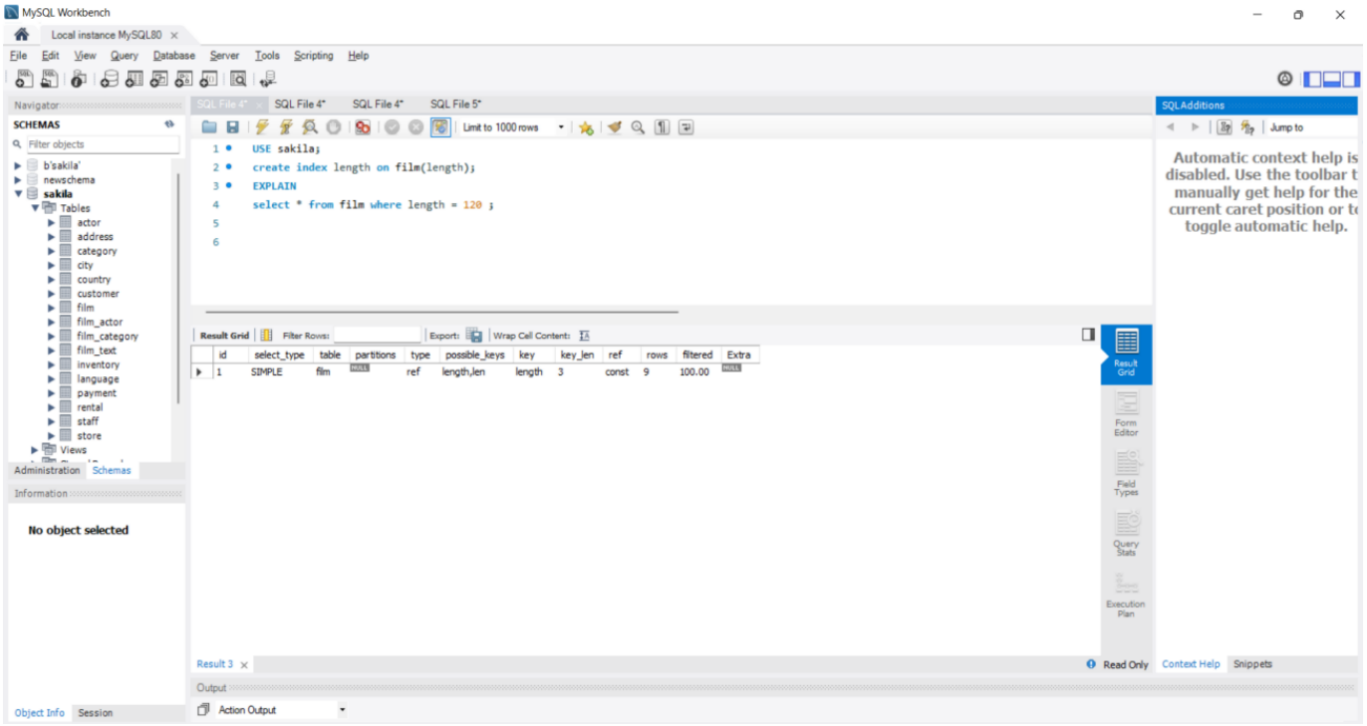
Aim: Implementation of Query monitor/ Query Execution plan, query statistics.

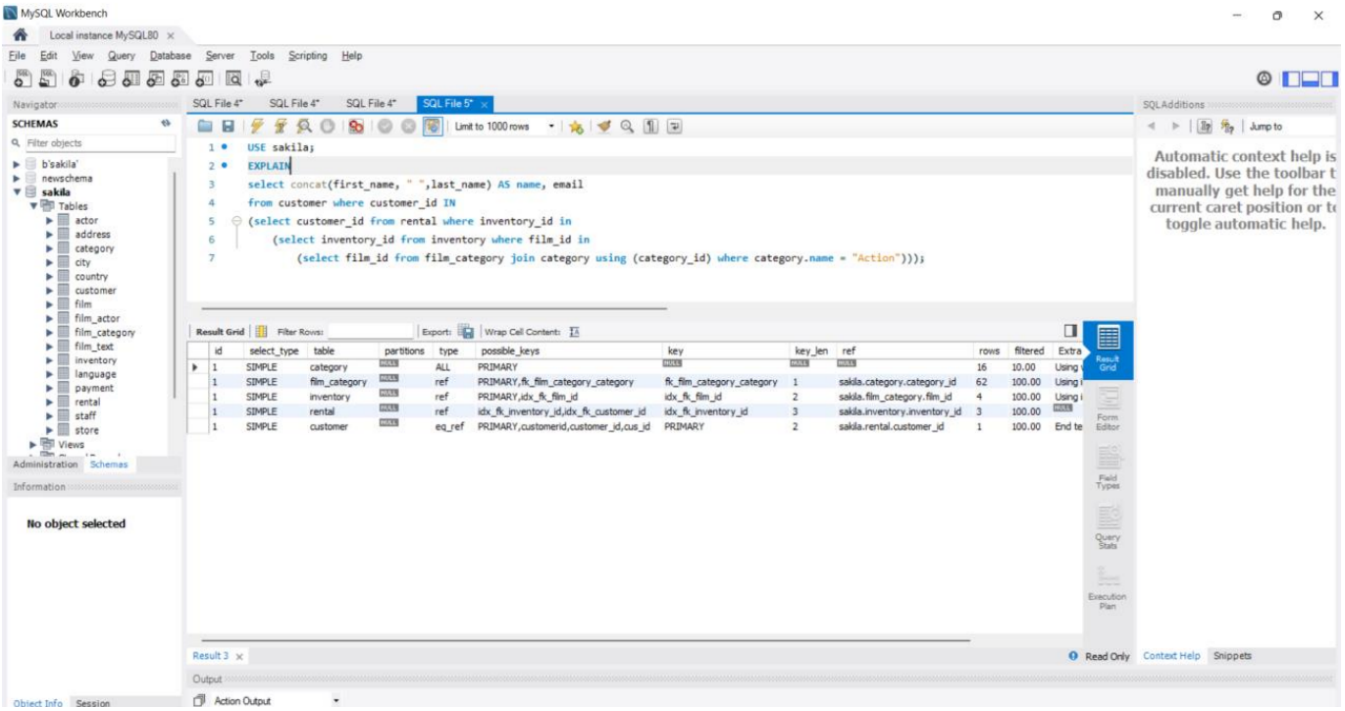
Theory:
The steps of process performed at the time of query execution by the engine database are described by a set of instructions called a query plan. The query plan is also referred to as the SQL server execution plan.
The Query Optimizer generates SQL query plan of optimum and economic query, which is its main objective of the query optimizer.
Multiple execution plans are generated by the query processing engine after query execution and from these generated plans, a plan with best performance is selected.

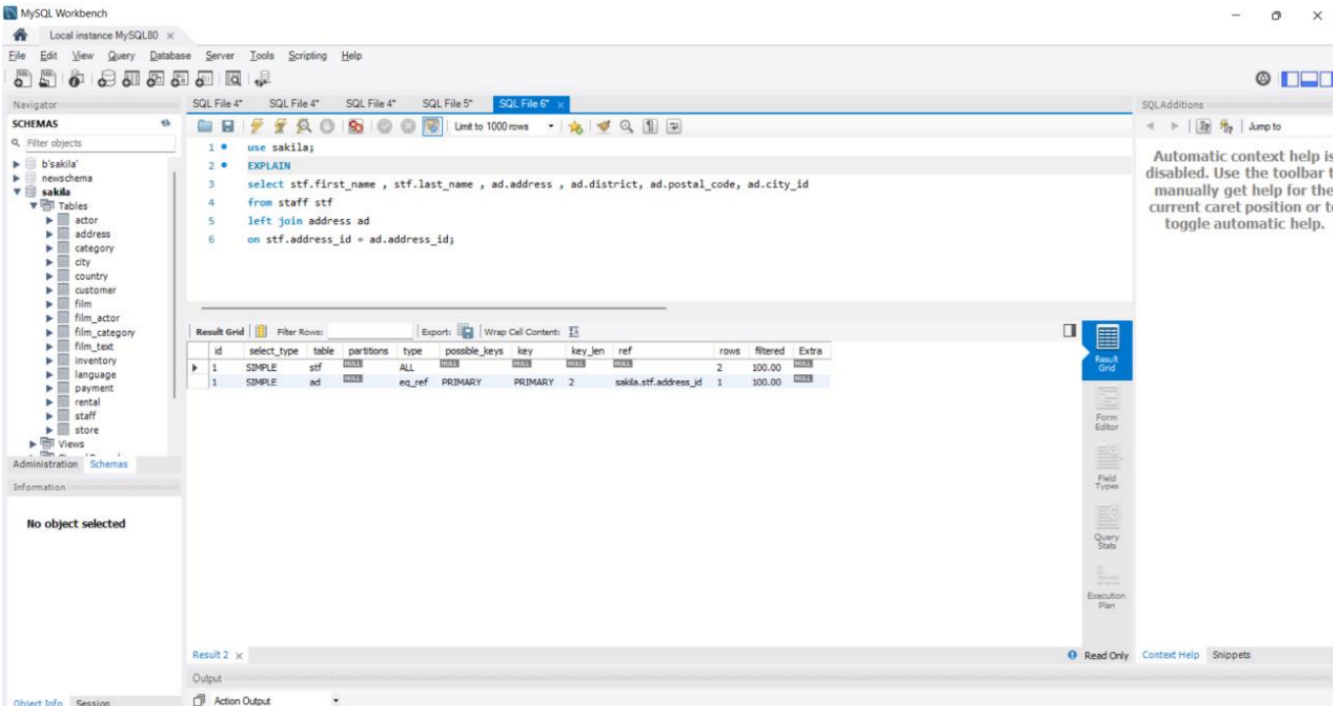
Query Optimizer uses statistics to create query plans that improve query performance. Statistics for query optimization are BLOBs that contain statistical information about distribution of values in one or more columns of a table. SQL server statistics are one of the key inputs for query optimizer to estimate how many rows will return from a query so it calculated the cost of a query plan from this estimation.

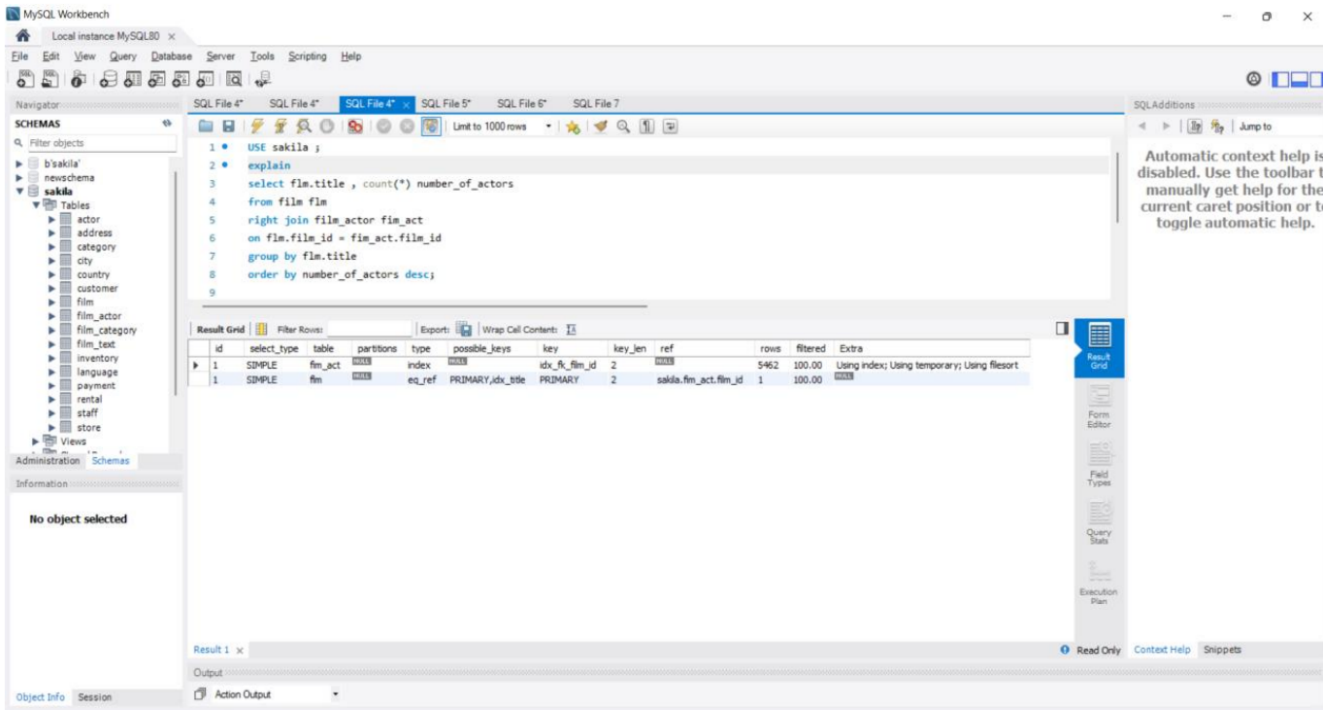
Conclusion: Query optimizer uses query statistics to create query plans. They estimate how many rows will be returned. It is used to find the most optimum query plan. Multiple execution plans are generated. Best one is selected.

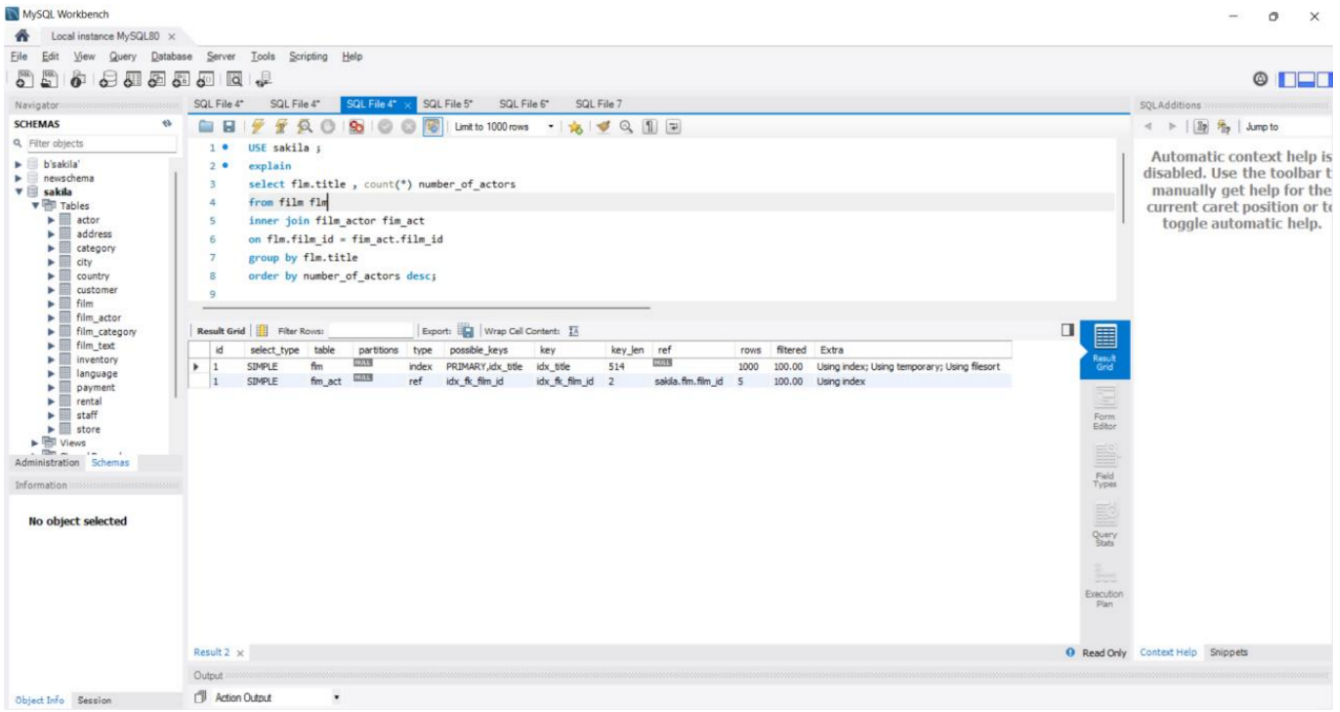
Output:

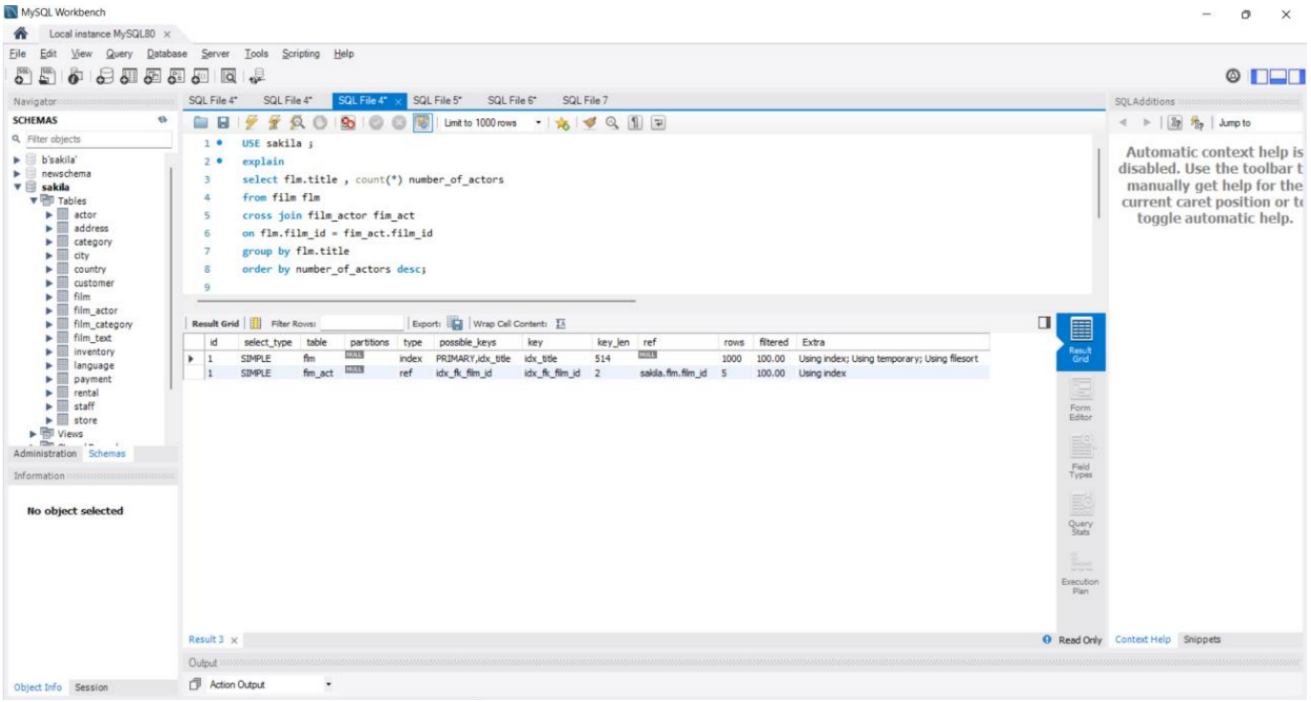

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
store
Views
No object selected
USE sakila;
create index length on film(length);
EXPLAIN
select * from film where length = 120 ;
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE film ref length,len length 3 const 9 100.00
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.


MySQL Workbench
Local instance MySQL80
SCHEMAS
sakila
Tables
No object selected
USE sakila;
EXPLAIN
select concat(first_name, " ",last_name) AS name, email
from customer where customer_id IN
(select customer_id from rental where inventory_id in
(select inventory_id from inventory where film_id in
(select film_id from film_category join category using (category_id) where category.name = "Action")));
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE category ALL PRIMARY 16 10.00
1 SIMPLE film_category ref PRIMARY,fk_film_category_category fk_film_category_category 1 sakila.category.category_id 62 100.00
1 SIMPLE inventory ref PRIMARY,idx_fk_film_id idx_fk_film_id 2 sakila.film_category.film_id 4 100.00
1 SIMPLE rental ref idx_fk_inventory_id,idx_fk_customer_id idx_fk_inventory_id 3 sakila.inventory.inventory_id 3 100.00
1 SIMPLE customer eq_ref PRIMARY,customerid,customer_id,cus_id PRIMARY 2 sakila.rental.customer_id 1 100.00
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 4* SQL File 4* SQL File 5* SQL File 6*
Limit to 1000 rows
1 use sakila;
2 EXPLAIN
3 select stf.first_name , stf.last_name , ad.address , ad.district, ad.postal_code, ad.city_id
4 from staff stf
5 left join address ad
6 on stf.address_id = ad.address_id;
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE stf NULL ALL NULL NULL NULL NULL 2 100.00 NULL
1 SIMPLE ad NULL eq_ref PRIMARY PRIMARY 2 sakila.stf.address_id 1 100.00 NULL
SCHEMAS
b'sakila'
newschema
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
store
Views
No object selected
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result 2
Read Only
Context Help
Snippets
Output
Action Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 4* SQL File 4* SQL File 5* SQL File 6* SQL File 7
Limit to 1000 rows
1 USE sakila ;
2 explain
3 select flm.title , count(*) number_of_actors
4 from film flm
5 right join film_actor fim_act
6 on flm.film_id = fim_act.film_id
7 group by flm.title
8 order by number_of_actors desc;
9
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE fim_act NULL index NULL idx_fk_film_id 2 NULL 5462 100.00 Using index; Using temporary; Using filesort
1 SIMPLE flm NULL eq_ref PRIMARY,idx_title PRIMARY 2 sakila.fim_act.film_id 1 100.00 NULL
SCHEMAS
b'sakila'
newschema
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
store
Views
No object selected
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result 1
Read Only
Context Help
Snippets
Output
Action Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
store
Views
Administration Schemas
Information
No object selected
Object Info Session
SQL File 4* SQL File 4* SQL File 4* SQL File 5* SQL File 6* SQL File 7
Limit to 1000 rows
1 USE sakila ;
2 explain
3 select flm.title , count(*) number_of_actors
4 from film flm
5 inner join film_actor fim_act
6 on flm.film_id = fim_act.film_id
7 group by flm.title
8 order by number_of_actors desc;
9
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE flm NULL index PRIMARY,idx_title idx_title 514 NULL 1000 100.00 Using index; Using temporary; Using filesort
1 SIMPLE fim_act NULL ref idx_fk_film_id idx_fk_film_id 2 sakila.flm.film_id 5 100.00 Using index
Result 2
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output

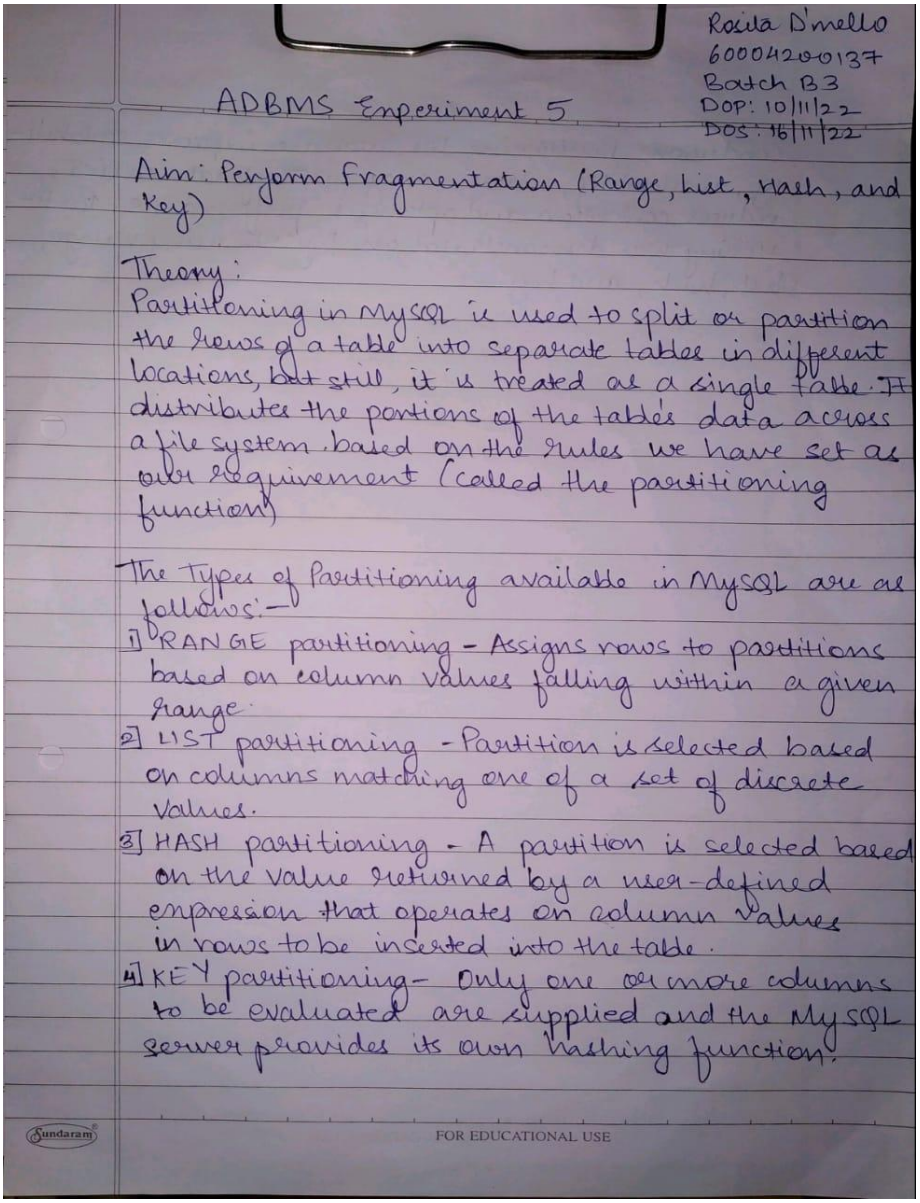

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sakila
Tables
actor
address
category
city
country
customer
film
film_actor
film_category
film_text
inventory
language
payment
rental
staff
store
Views
Administration Schemas
Information
No object selected
Object Info Session
SQL File 4* SQL File 4* SQL File 4* SQL File 5* SQL File 6* SQL File 7
Limit to 1000 rows
1 USE sakila ;
2 explain
3 select flm.title , count(*) number_of_actors
4 from film flm
5 cross join film_actor fim_act
6 on flm.film_id = fim_act.film_id
7 group by flm.title
8 order by number_of_actors desc;
9
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE flm NULL index PRIMARY,idx_title idx_title 514 NULL 1000 100.00 Using index; Using temporary; Using filesort
1 SIMPLE fim_act NULL ref idx_fk_film_id idx_fk_film_id 2 sakila.flm.film_id 5 100.00 Using index
Result 3
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output

ADBMS Experiment 5

Date of Performance: 10th November 2022
Date of Submission: 16th November 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Perform Fragmentation (Range, List, Hash and Key)
Theory:



Rosita D'mello
60004200137
Batch B3
DOP: 10/11/22
DOS: 16/11/22

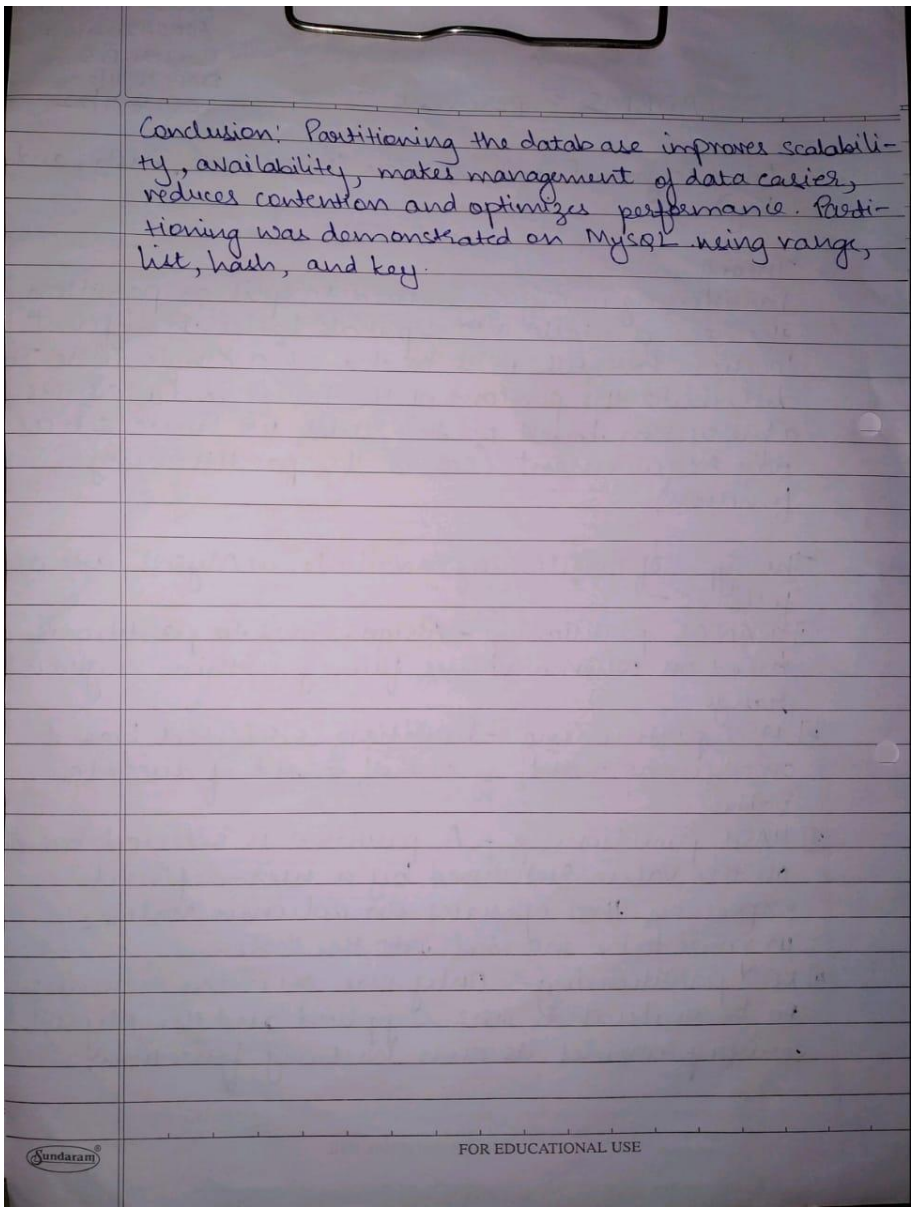
ADBMS Experiment 5

Aim: Perform Fragmentation (Range, List, Hash, and Key)

Theory:
Partitioning in MySQL is used to split or partition the rows of a table into separate tables in different locations, but still, it is treated as a single table. It distributes the portions of the tables data across a file system based on the rules we have set as our requirement (called the partitioning function)

The Types of Partitioning available in MySQL are as follows:-
1] RANGE partitioning - Assigns rows to partitions based on column values falling within a given range.
2] LIST partitioning - Partition is selected based on columns matching one of a set of discrete values.
3] HASH partitioning - A partition is selected based on the value returned by a user-defined expression that operates on column values in rows to be inserted into the table.
4] KEY partitioning - Only one or more columns to be evaluated are supplied and the MySQL server provides its own hashing function.

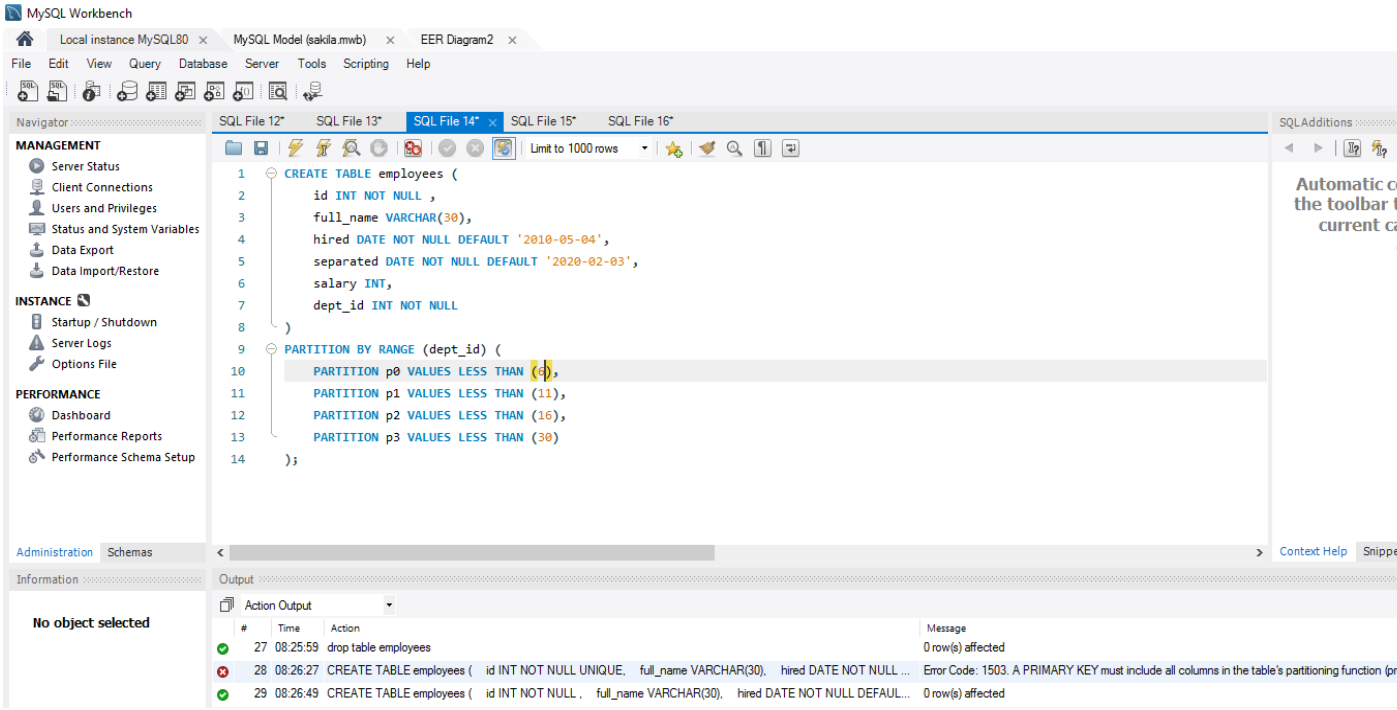
Sundaram
FOR EDUCATIONAL USE

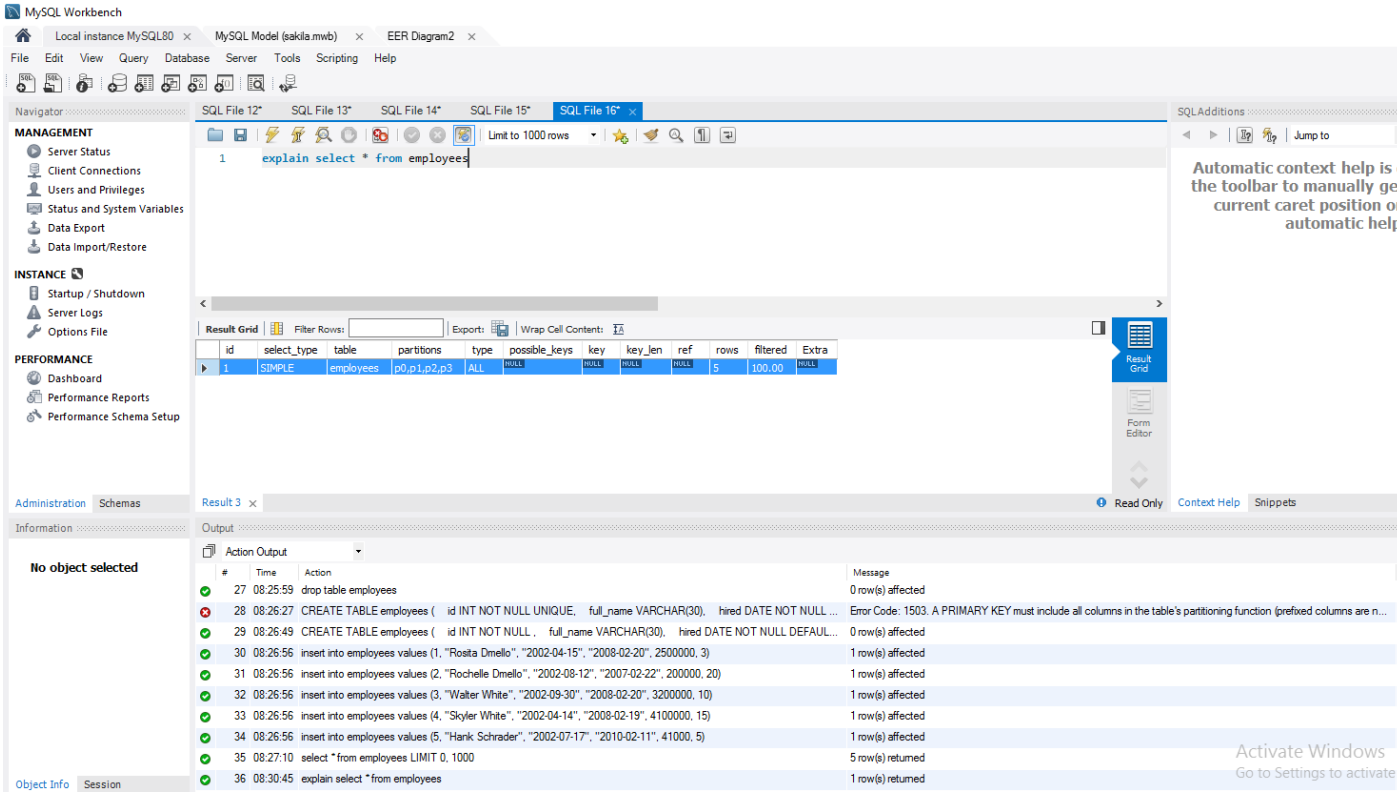

Conclusion: Partitioning the database improves scalability, availability, makes management of data easier, reduces contention and optimizes performance. Partitioning was demonstrated on MySQL using range, list, hash, and key.

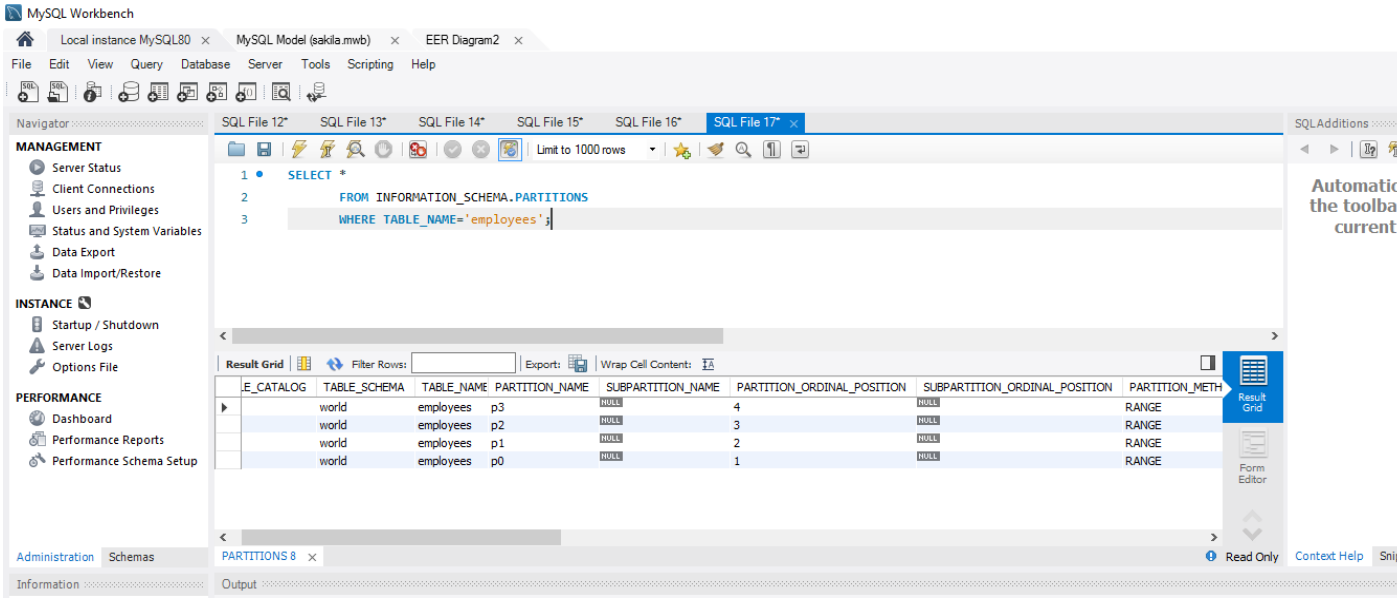
Sundaram

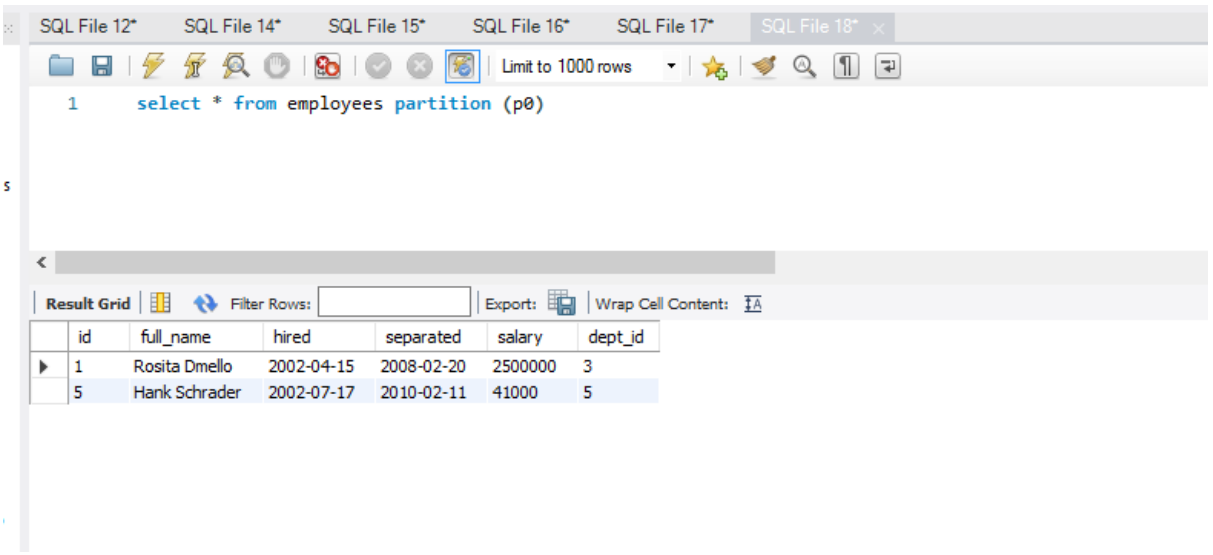
FOR EDUCATIONAL USE

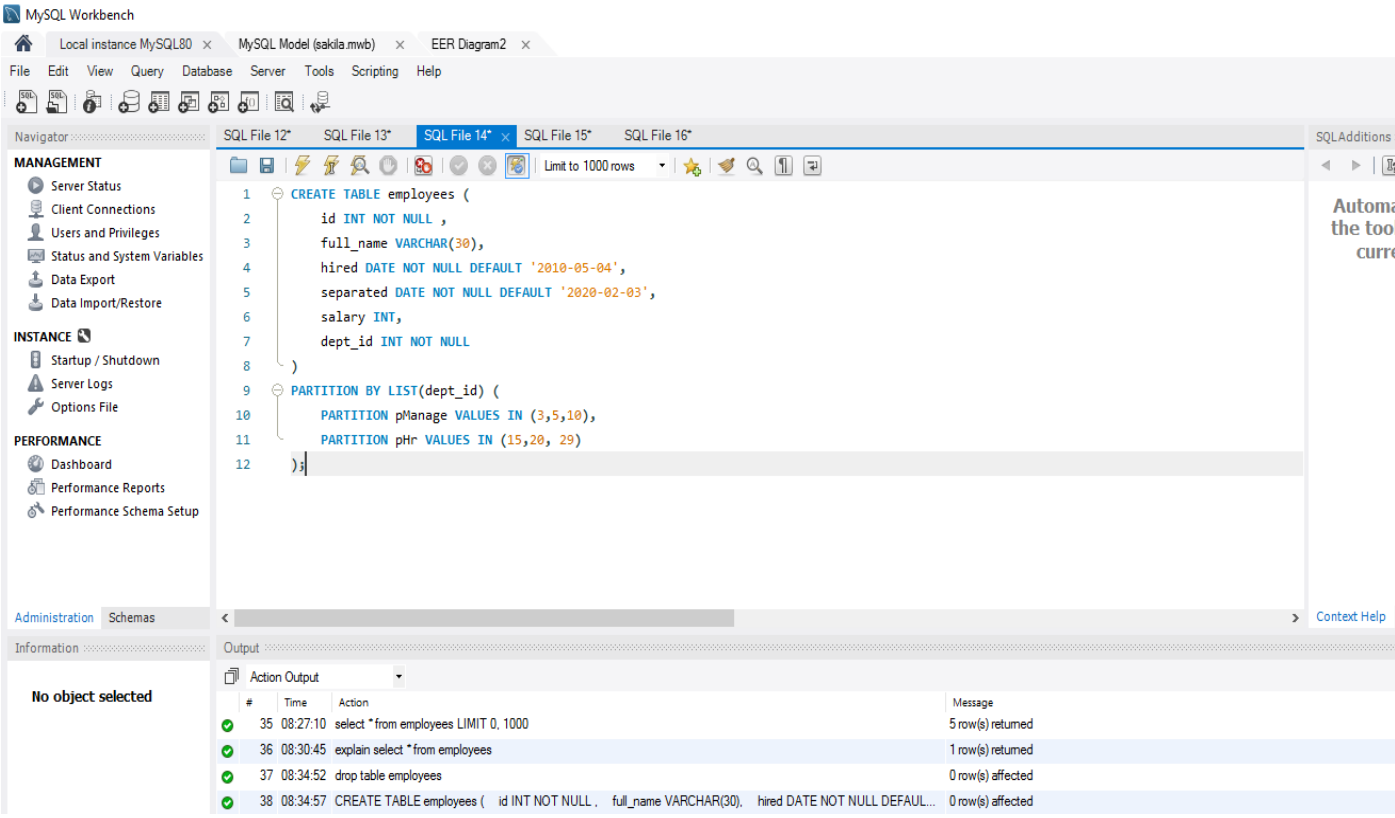
Output:

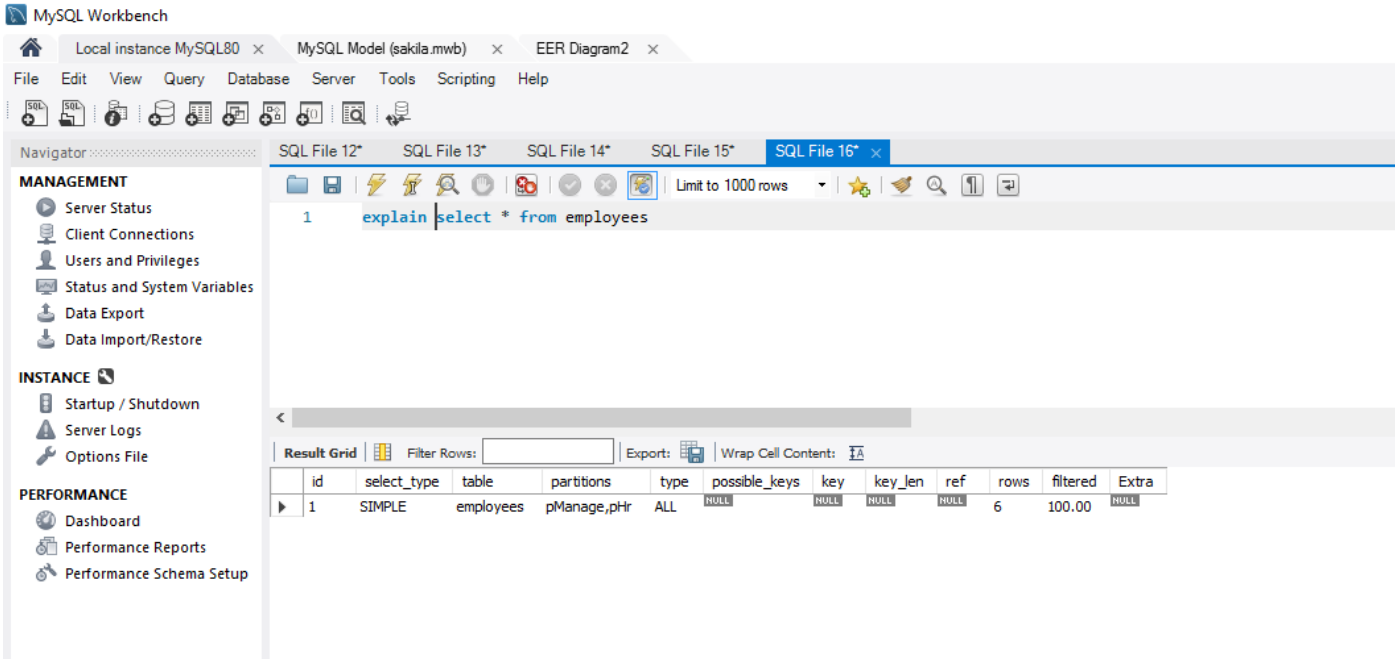

MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16*
Limit to 1000 rows
1 CREATE TABLE employees (
2 id INT NOT NULL ,
3 full_name VARCHAR(30),
4 hired DATE NOT NULL DEFAULT '2010-05-04',
5 separated DATE NOT NULL DEFAULT '2020-02-03',
6 salary INT,
7 dept_id INT NOT NULL
8)
9 PARTITION BY RANGE (dept_id) (
10 PARTITION p0 VALUES LESS THAN (6),
11 PARTITION p1 VALUES LESS THAN (11),
12 PARTITION p2 VALUES LESS THAN (16),
13 PARTITION p3 VALUES LESS THAN (30)
14);
SQLAdditions
Context Help
Output
Action Output
Time Action Message
27 08:25:59 drop table employees 0 row(s) affected
28 08:26:27 CREATE TABLE employees (id INT NOT NULL UNIQUE, full_name VARCHAR(30), hired DATE NOT NULL ... Error Code: 1503. A PRIMARY KEY must include all columns in the table's partitioning function (pr
29 08:26:49 CREATE TABLE employees (id INT NOT NULL , full_name VARCHAR(30), hired DATE NOT NULL DEFAUL... 0 row(s) affected

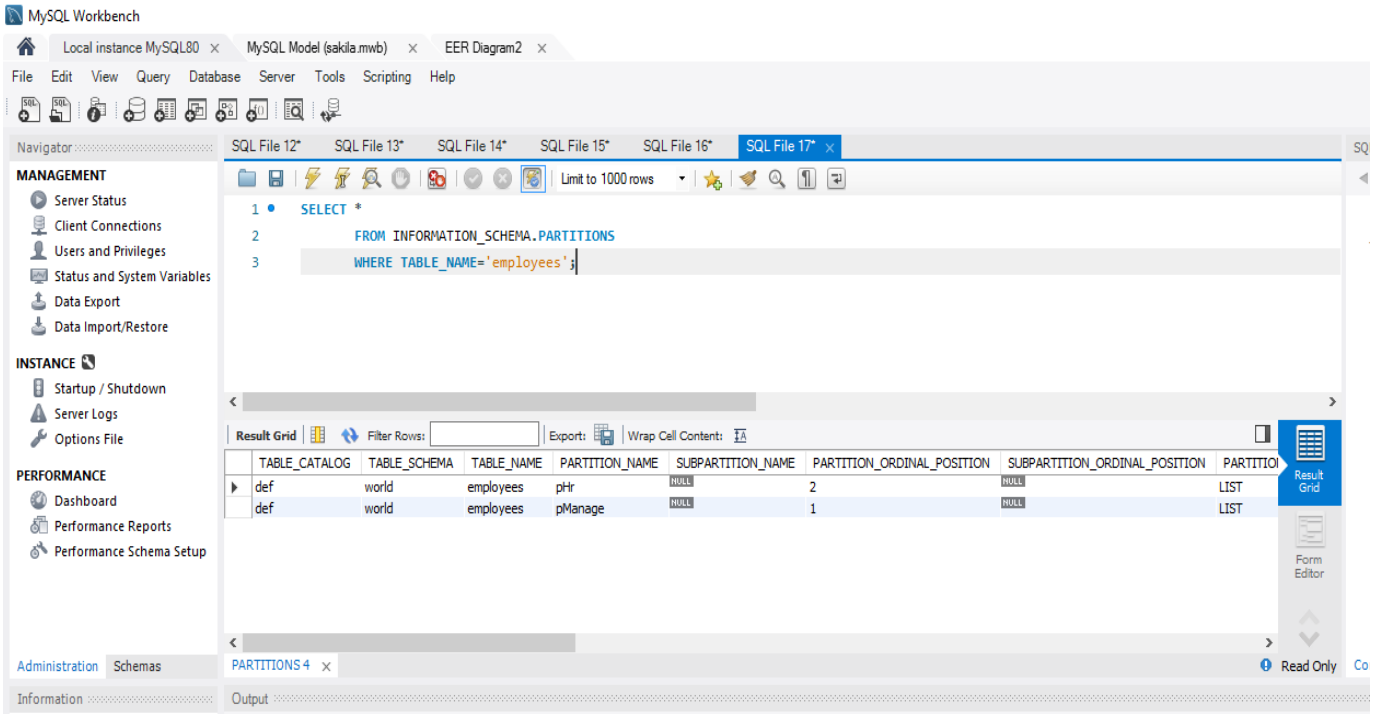

MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16*
Limit to 1000 rows
1 explain select * from employees
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE employees p0,p1,p2,p3 ALL NULL NULL NULL NULL 5 100.00 NULL
Result Grid
Form Editor
Read Only
Result 3
SQLAdditions
Jump to
Context Help
Snippets
Output
Action Output
Time Action Message
27 08:25:59 drop table employees 0 row(s) affected
28 08:26:27 CREATE TABLE employees (id INT NOT NULL UNIQUE, full_name VARCHAR(30), hired DATE NOT NULL ... Error Code: 1503. A PRIMARY KEY must include all columns in the table's partitioning function (prefixed columns are n...
29 08:26:49 CREATE TABLE employees (id INT NOT NULL , full_name VARCHAR(30), hired DATE NOT NULL DEFAUL... 0 row(s) affected
30 08:26:56 insert into employees values (1, "Rosita Dmello", "2002-04-15", "2008-02-20", 2500000, 3) 1 row(s) affected
31 08:26:56 insert into employees values (2, "Rochelle Dmello", "2002-08-12", "2007-02-22", 200000, 20) 1 row(s) affected
32 08:26:56 insert into employees values (3, "Walter White", "2002-09-30", "2008-02-20", 3200000, 10) 1 row(s) affected
33 08:26:56 insert into employees values (4, "Skyler White", "2002-04-14", "2008-02-19", 4100000, 15) 1 row(s) affected
34 08:26:56 insert into employees values (5, "Hank Schrader", "2002-07-17", "2010-02-11", 41000, 5) 1 row(s) affected
35 08:27:10 select * from employees LIMIT 0, 1000 5 row(s) returned
36 08:30:45 explain select * from employees 1 row(s) returned
Object Info Session
Activate Windows
Go to Settings to activate

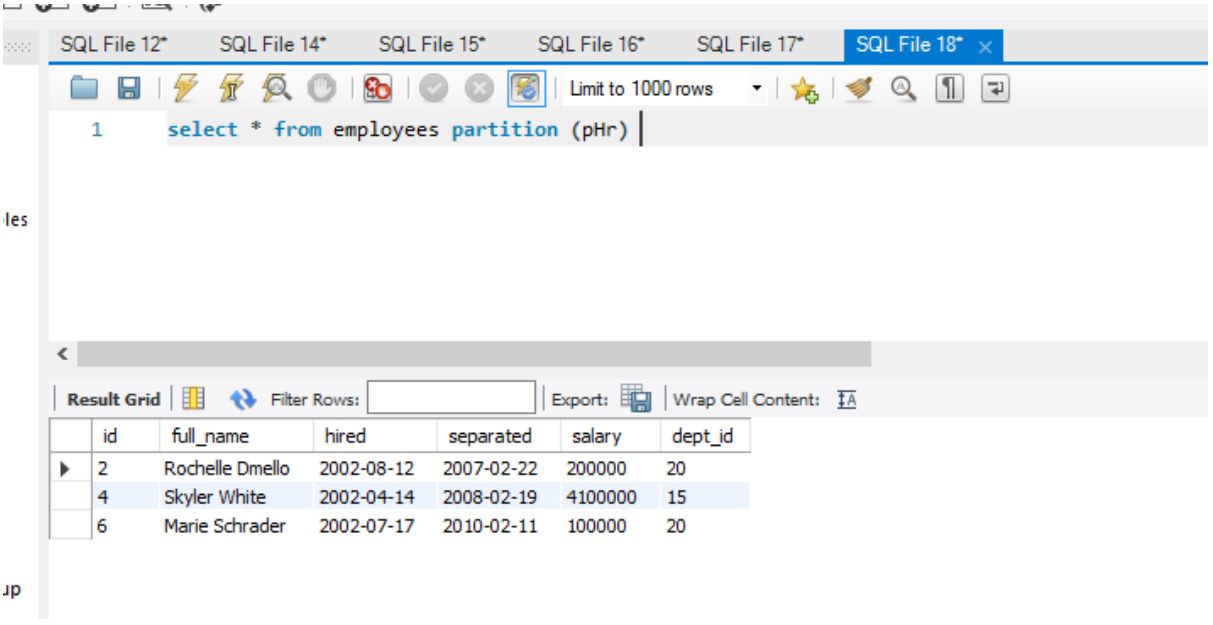

MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*
Limit to 1000 rows
1 SELECT *
2 FROM INFORMATION_SCHEMA.PARTITIONS
3 WHERE TABLE_NAME='employees';
Result Grid Filter Rows: Export: Wrap Cell Content:
E_CATALOG TABLE_SCHEMA TABLE_NAME PARTITION_NAME SUBPARTITION_NAME PARTITION_ORDINAL_POSITION SUBPARTITION_ORDINAL_POSITION PARTITION_METH
world employees p3 NULL 4 NULL RANGE
world employees p2 NULL 3 NULL RANGE
world employees p1 NULL 2 NULL RANGE
world employees p0 NULL 1 NULL RANGE
Result Grid
Form Editor
PARTITIONS 8
Read Only
Context Help
SQLAdditions
Output


SQL File 12* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18*
Limit to 1000 rows
1 select * from employees partition (p0)
Result Grid Filter Rows: Export: Wrap Cell Content:
id full_name hired separated salary dept_id
1 Rosita Dmello 2002-04-15 2008-02-20 2500000 3
5 Hank Schrader 2002-07-17 2010-02-11 41000 5


MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16*
Limit to 1000 rows
1 CREATE TABLE employees (
2 id INT NOT NULL ,
3 full_name VARCHAR(30),
4 hired DATE NOT NULL DEFAULT '2010-05-04',
5 separated DATE NOT NULL DEFAULT '2020-02-03',
6 salary INT,
7 dept_id INT NOT NULL
8)
9 PARTITION BY LIST(dept_id) (
10 PARTITION pManage VALUES IN (3,5,10),
11 PARTITION pHr VALUES IN (15,20, 29)
12);
SQLAdditions
Context Help
Output
Action Output
Time Action Message
35 08:27:10 select * from employees LIMIT 0, 1000 5 row(s) returned
36 08:30:45 explain select * from employees 1 row(s) returned
37 08:34:52 drop table employees 0 row(s) affected
38 08:34:57 CREATE TABLE employees (id INT NOT NULL , full_name VARCHAR(30), hired DATE NOT NULL DEFAUL... 0 row(s) affected


MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16*
Limit to 1000 rows
1 explain select * from employees
Result Grid Filter Rows: Export: Wrap Cell Content:
id select_type table partitions type possible_keys key key_len ref rows filtered Extra
1 SIMPLE employees pManage,pHr ALL NULL NULL NULL NULL 6 100.00 NULL


MySQL Workbench
Local instance MySQL80
MySQL Model (sakila.mwb)
EER Diagram2
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*
Limit to 1000 rows
1 SELECT *
2 FROM INFORMATION_SCHEMA.PARTITIONS
3 WHERE TABLE_NAME='employees';
Result Grid Filter Rows: Export: Wrap Cell Content:
TABLE_CATALOG TABLE_SCHEMA TABLE_NAME PARTITION_NAME SUBPARTITION_NAME PARTITION_ORDINAL_POSITION SUBPARTITION_ORDINAL_POSITION PARTITIO
def world employees pHr NULL 2 NULL LIST
def world employees pManage NULL 1 NULL LIST
Result Grid
Form Editor
Read Only
PARTITIONS 4
Output


SQL File 12* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18*
Limit to 1000 rows
1 select * from employees partition (pHr)
Result Grid Filter Rows: Export: Wrap Cell Content:
id full_name hired separated salary dept_id
2 Rochelle Dmello 2002-08-12 2007-02-22 200000 20
4 Skyler White 2002-04-14 2008-02-19 4100000 15
6 Marie Schrader 2002-07-17 2010-02-11 100000 20

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*

Limit to 1000 rows

```sql
CREATE TABLE employees (
    id INT NOT NULL ,
    full_name VARCHAR(30),
    hired DATE NOT NULL DEFAULT '2010-05-04',
    separated DATE NOT NULL DEFAULT '2020-02-03',
    salary INT,
    dept_id INT NOT NULL
)
PARTITION BY HASH( YEAR(hired) )
PARTITIONS 4;
```

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*

Limit to 1000 rows

```sql
SELECT *
    FROM INFORMATION_SCHEMA.PARTITIONS
    WHERE TABLE_NAME='employees';
```

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

CHEMA	TABLE_NAME	PARTITION_NAME	SUBPARTITION_NAME	PARTITION_ORDINAL_POSITION	SUBPARTITION_ORDINAL_POSITION	PARTITION_METHOD	SUBPARTITION_M
	employees	p3	NULL	4	NULL	HASH	NULL
	employees	p2	NULL	3	NULL	HASH	NULL
	employees	p1	NULL	2	NULL	HASH	NULL
	employees	p0	NULL	1	NULL	HASH	NULL

SQL File 12* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18*

Limit to 1000 rows

```sql
select * from employees partition (p2)
```

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

id	full_name	hired	separated	salary	dept_id
1	Rosita Dmello	2002-04-15	2008-02-20	2500000	3
2	Rochelle Dmello	2002-08-12	2007-02-22	200000	20
3	Walter White	2002-09-30	2008-02-20	3200000	10
4	Skyler White	2002-04-14	2008-02-19	4100000	15
5	Hank Schrader	2002-07-17	2010-02-11	41000	5
6	Marie Schrader	2002-07-17	2010-02-11	100000	20

employees 5

Output

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*

Limit to 1000 rows

```sql
CREATE TABLE employees (
    id INT NOT NULL ,
    full_name VARCHAR(30),
    hired DATE NOT NULL DEFAULT '2010-05-04',
    separated DATE NOT NULL DEFAULT '2020-02-03',
    salary INT,
    dept_id INT NOT NULL,
    UNIQUE KEY (id)
)
PARTITION BY KEY()
PARTITIONS 3;
```

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*

Limit to 1000 rows

```sql
explain select * from employees
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employees	p0,p1,p2	ALL	NULL	NULL	NULL	NULL	6	100.00	NULL

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17*

Limit to 1000 rows

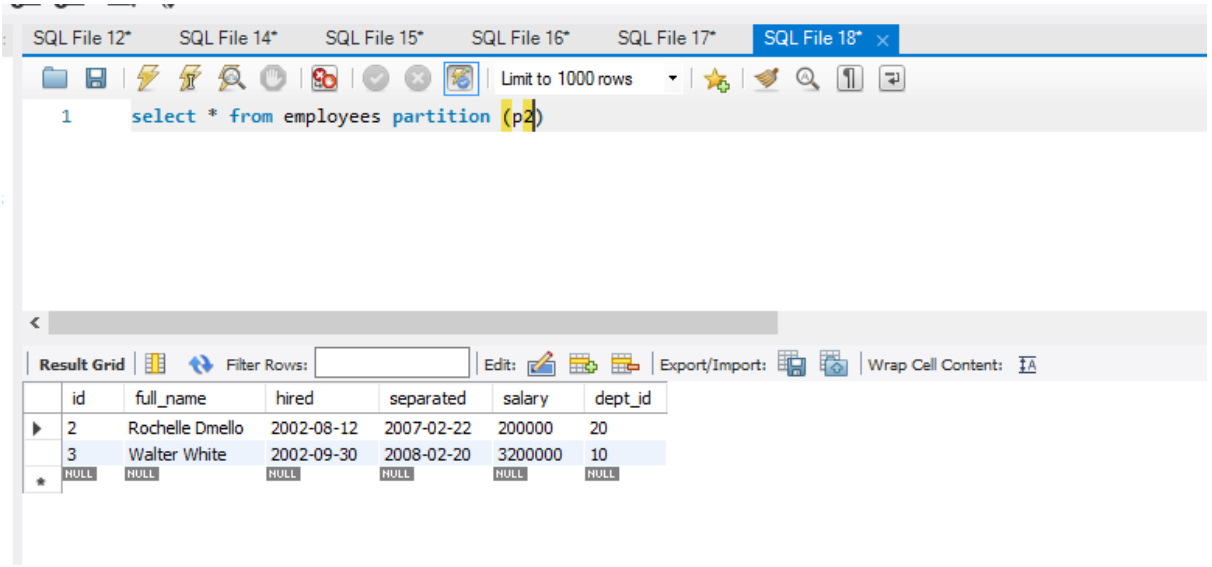
```sql
SELECT *
    FROM INFORMATION_SCHEMA.PARTITIONS
    WHERE TABLE_NAME='employees';
```

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

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	PARTITION_NAME	SUBPARTITION_NAME	PARTITION_ORDINAL_POSITION	SUBPARTITION_ORDINAL_POSITION	PARTITION_
def	world	employees	p2	NULL	3	NULL	KEY
def	world	employees	p1	NULL	2	NULL	KEY
def	world	employees	p0	NULL	1	NULL	KEY

Result Grid

Form Editor


SQL File 12*
SQL File 14*
SQL File 15*
SQL File 16*
SQL File 17*
SQL File 18*
Limit to 1000 rows
1 select * from employees partition (p2)
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id full_name hired separated salary dept_id
2 Rochelle Dmello 2002-08-12 2007-02-22 200000 20
3 Walter White 2002-09-30 2008-02-20 3200000 10
NULL NULL NULL NULL NULL NULL

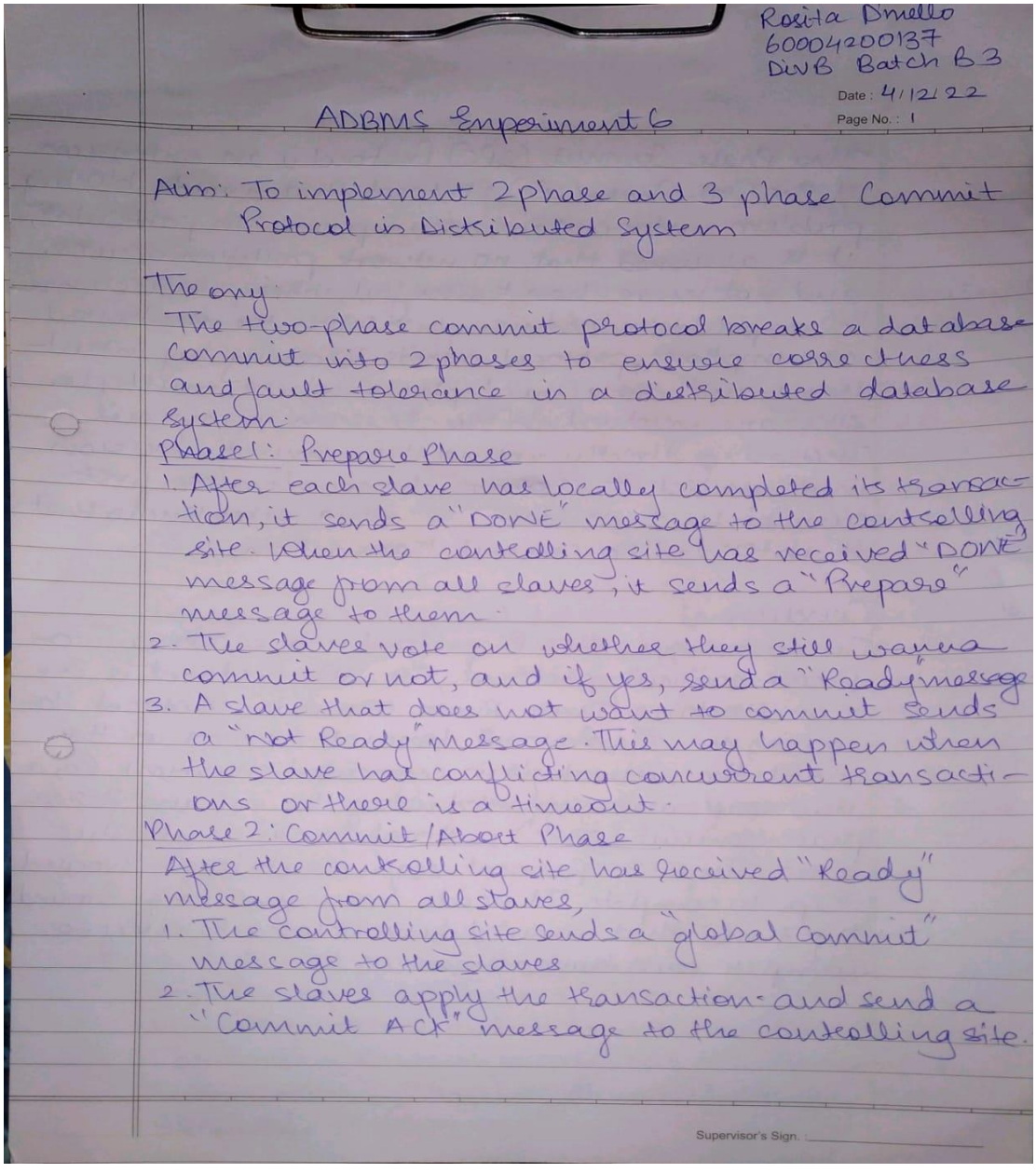
ADBMS Experiment 6

Date of Performance: 17th December 2022
Date of Submission: 4th December 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim : To implement 2 phase and 3phase commit protocol in distributed system

Theory:



Rosita Dmello
60004200137
DivB Batch B3
Date : 4/12/22
Page No. : 1

ADBMS Experiment 6

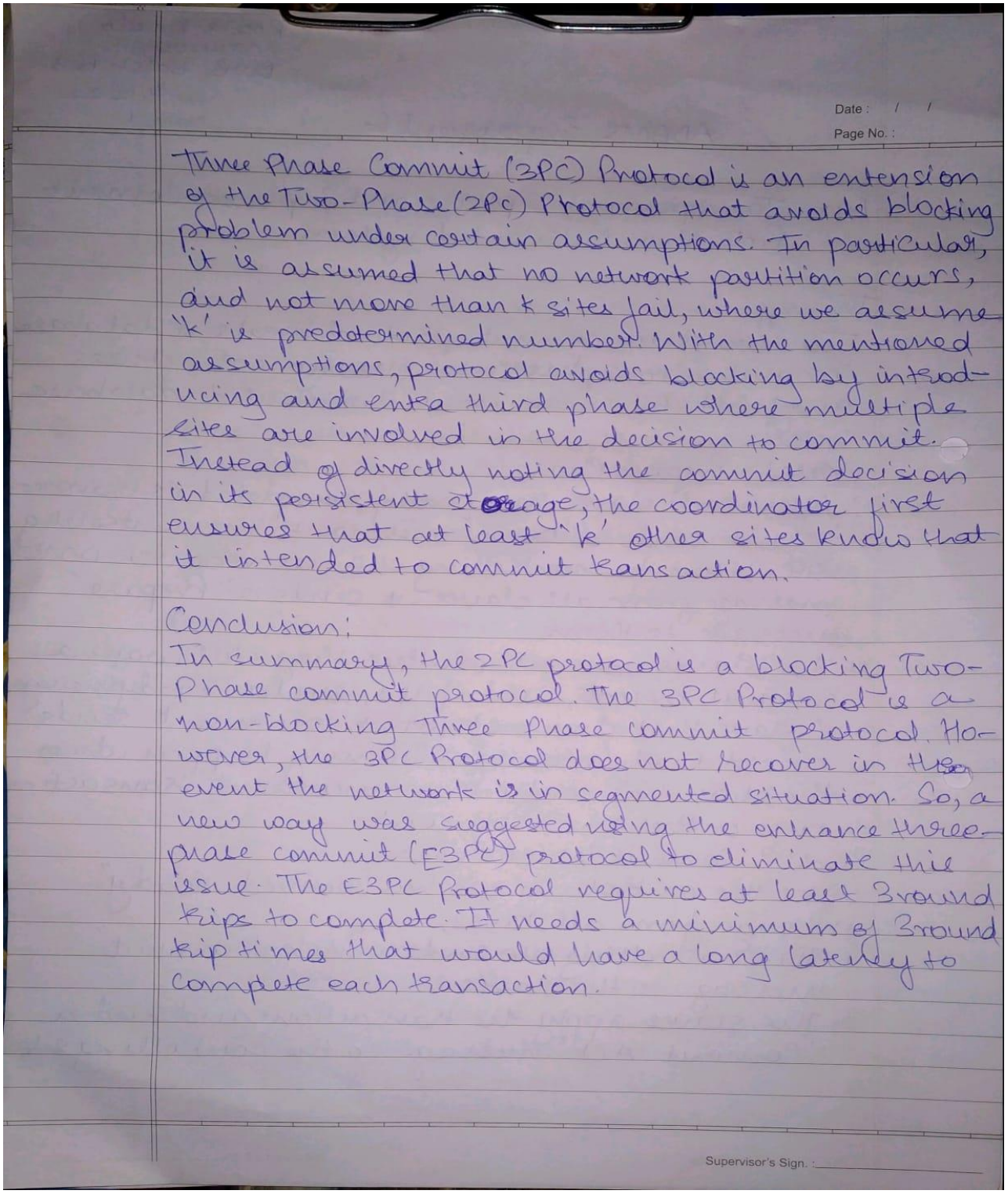
Aim: To implement 2 phase and 3 phase Commit Protocol in Distributed System

Theory:
The two-phase commit protocol breaks a database commit into 2 phases to ensure correctness and fault tolerance in a distributed database system.

Phase1: Prepare Phase
1. After each slave has locally completed its transaction, it sends a "DONE" message to the controlling site. When the controlling site has received "DONE" message from all slaves, it sends a "Prepare" message to them.
2. The slaves vote on whether they still wanna commit or not, and if yes, send a "Ready" message
3. A slave that does not want to commit sends a "not Ready" message. This may happen when the slave has conflicting concurrent transactions or there is a timeout.

Phase 2: Commit/Abort Phase
After the controlling site has received "Ready" message from all slaves,
1. The controlling site sends a "global commit" message to the slaves
2. The slaves apply the transaction and send a "Commit Ack" message to the controlling site.

Supervisor's Sign. :



Date : / /
Page No. :

Three Phase Commit (3PC) Protocol is an extension of the Two-Phase (2PC) Protocol that avoids blocking problem under certain assumptions. In particular, it is assumed that no network partition occurs, and not more than k sites fail, where we assume 'k' is predetermined number. With the mentioned assumptions, protocol avoids blocking by introducing and extra third phase where multiple sites are involved in the decision to commit. Instead of directly noting the commit decision in its persistent storage, the coordinator first ensures that at least 'k' other sites know that it intended to commit transaction.

Conclusion:
In summary, the 2PC protocol is a blocking Two-Phase commit protocol. The 3PC Protocol is a non-blocking Three Phase commit protocol. However, the 3PC Protocol does not recover in the event the network is in segmented situation. So, a new way was suggested using the enhance three-phase commit (E3PC) protocol to eliminate this issue. The E3PC Protocol requires at least 3 round trips to complete. It needs a minimum of 3 round trip times that would have a long latency to complete each transaction.

Supervisor's Sign. :_____

Code:

Client

```
import java.io.*;
import java.net.*;
public class Client implements Runnable
{
```

```
static Socket clientSocket = null;
static PrintStream os = null;
static DataInputStream is = null;
static BufferedReader inputLine = null;
static boolean closed = false;
public static void main(String[] args)
{
   int port_number=1111;
   String host="localhost";
   try {
      clientSocket = new Socket(host, port_number);
      inputLine = new BufferedReader(new InputStreamReader(System.in));
      os = new PrintStream(clientSocket.getOutputStream());
      is = new DataInputStream(clientSocket.getInputStream());
   } catch (Exception e)
   {   System.out.println("Exception occurred : "+e.getMessage()); }


   if (clientSocket != null && os != null && is != null)
   {
      try
      {
         new Thread(new Client()).start();
         while (!closed)
         {
            os.println(inputLine.readLine());
         }
         os.close();
         is.close();
         clientSocket.close();
      } catch (IOException e)
      {
         System.err.println("IOException:  " + e);
      }
   }
}
@SuppressWarnings("deprecation")
public void run()     {
   String responseLine;
   try
   {
      while ((responseLine = is.readLine()) != null)
      {
```

```
            System.out.println("\n"+responseLine);
            if (responseLine.equalsIgnoreCase("GLOBAL_COMMIT")==true ||
responseLine.equalsIgnoreCase("GLOBAL_ABORT")==true )
            {
               break;
            }
         }
         closed=true;
      }
      catch (IOException e)
      {
         System.err.println("IOException:  " + e);
      }
   }
}
```

Server

```
import java.io.*;
import java.net.*;
import java.util.*;

public class Server {
   boolean closed = false, inputFromAll = false;
   List<ClientThread> thread;
   List<String> data;
   List<String> decision;

   Server() {
      thread = new ArrayList<ClientThread>();
      data = new ArrayList<String>();
      decision= new ArrayList<String>();
   }

   public static void main(String args[])
   {
      Socket clientSocket = null;
      ServerSocket serverSocket = null;
      int port_number = 1111;
```

```
        Server server = new Server();
        try
        {
            serverSocket = new ServerSocket(port_number);
        } catch (IOException e) {
            System.out.println(e);
        }
        while (!server.closed)
        {
            try {
                    clientSocket = serverSocket.accept();
                    ClientThread clientThread = new ClientThread(server, clientSocket);
                    (server.thread).add(clientThread);
                    System.out.println("\nNow Total clients are : " + (server.thread).size());
                    (server.data).add("NOT_SENT");
                    (server.decision).add("NOT_SENT");
                    clientThread.start();
            } catch (IOException e) { }
        }
        try {

            serverSocket.close();
        } catch (Exception e1) { }
    }
}

class ClientThread extends Thread
{
    DataInputStream is = null;
    String line;
    String destClient = "";
    String name;
    PrintStream os = null;
    Socket clientSocket = null;
    String clientIdentity;
    Server server;

    public ClientThread(Server server, Socket clientSocket)
    {
        this.clientSocket = clientSocket;
        this.server = server;
    }
```

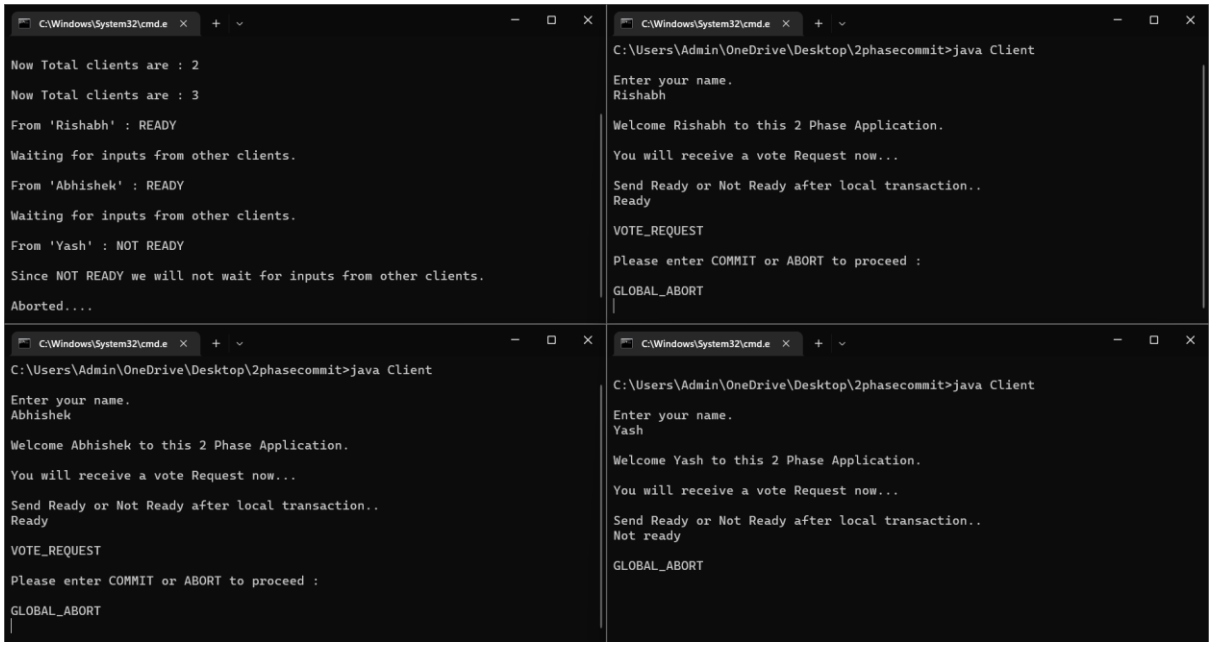
```
    @SuppressWarnings("deprecation")
    public void run()
    {
        try {
            is = new DataInputStream(clientSocket.getInputStream());
            os = new PrintStream(clientSocket.getOutputStream());
            os.println("Enter your name.");
            name = is.readLine();
            clientIdentity = name;
            os.println("Welcome " + name + " to this 2 Phase Application.\nYou will
receive a vote Request now...");
            os.println("Send Ready or Not Ready after local transaction..");
            while (true)
            {
                line = is.readLine();
                if (line.equalsIgnoreCase("NOT READY"))
                {
                    System.out.println("\nFrom ''' + clientIdentity
                            + ''' : NOT READY\n\nSince NOT READY we will not wait for
inputs from other clients.");
                    System.out.println("\nAborted....");

                    for (int i = 0; i < (server.thread).size(); i++) {
                        ((server.thread).get(i)).os.println("GLOBAL_ABORT");
                        ((server.thread).get(i)).os.close();
                        ((server.thread).get(i)).is.close();
                    }
                    break;
                }
                if (line.equalsIgnoreCase("READY"))
                {
                    System.out.println("\nFrom ''' + clientIdentity + ''' : READY");
                    if ((server.thread).contains(this))
                    {
                        (server.data).set((server.thread).indexOf(this), "READY");
                        for (int j = 0; j < (server.data).size(); j++)
                        {
                            if (!(((server.data).get(j)).equalsIgnoreCase("NOT_SENT")))
                            {
                                server.inputFromAll = true;
                                continue;
                            }
                            else{
```

```
                        server.inputFromAll = false;
                        System.out.println("\nWaiting for inputs from other clients.");
                        break;
                    }
                }
                if (server.inputFromAll)
                {
                    System.out.println("All Ready..");
                }
            }
        }

        os.println("VOTE_REQUEST\nPlease enter COMMIT or ABORT to
proceed : ");
        line = is.readLine();
        if (line.equalsIgnoreCase("ABORT"))
        {
            System.out.println("\nFrom ''' + clientIdentity
                    + ''' : ABORT\n\nSince ABORT we will not wait for inputs from
other clients.");
            System.out.println("\nAborted....");

            for (int i = 0; i < (server.thread).size(); i++) {
                ((server.thread).get(i)).os.println("GLOBAL_ABORT");
                ((server.thread).get(i)).os.close();
                ((server.thread).get(i)).is.close();
            }
            break;
        }
        if (line.equalsIgnoreCase("COMMIT")){
            System.out.println("\nFrom ''' + clientIdentity + ''' : COMMIT");
            if ((server.thread).contains(this))
            {
                (server.decision).set((server.thread).indexOf(this), "COMMIT");
                for (int j = 0; j < (server.decision).size(); j++)
                {
                    if (!(((server.decision).get(j)).equalsIgnoreCase("NOT_SENT")))
                    {
                        server.inputFromAll = true;
                        continue;
                    }
                    else{
                        server.inputFromAll = false;
```

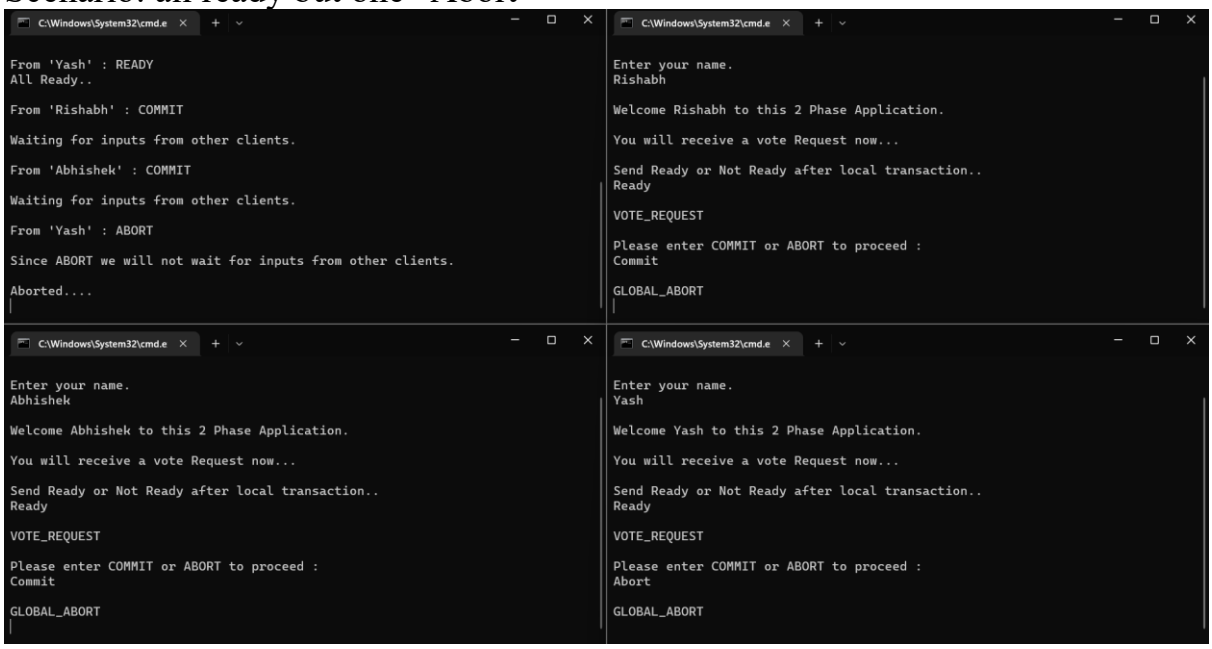
```
                        System.out.println("\nWaiting for inputs from other clients.");
                        break;
                    }
                }
                if (server.inputFromAll){
                    System.out.println("\n\nCommited....");
                    for (int i = 0; i < (server.thread).size(); i++)
                    {
                        ((server.thread).get(i)).os.println("GLOBAL_COMMIT");
                        ((server.thread).get(i)).os.close();
                        ((server.thread).get(i)).is.close();
                    }
                    break;
                }
            }
        }
        server.closed = true;
        clientSocket.close();
    } catch (IOException e) { }

  }
}
```

Output:

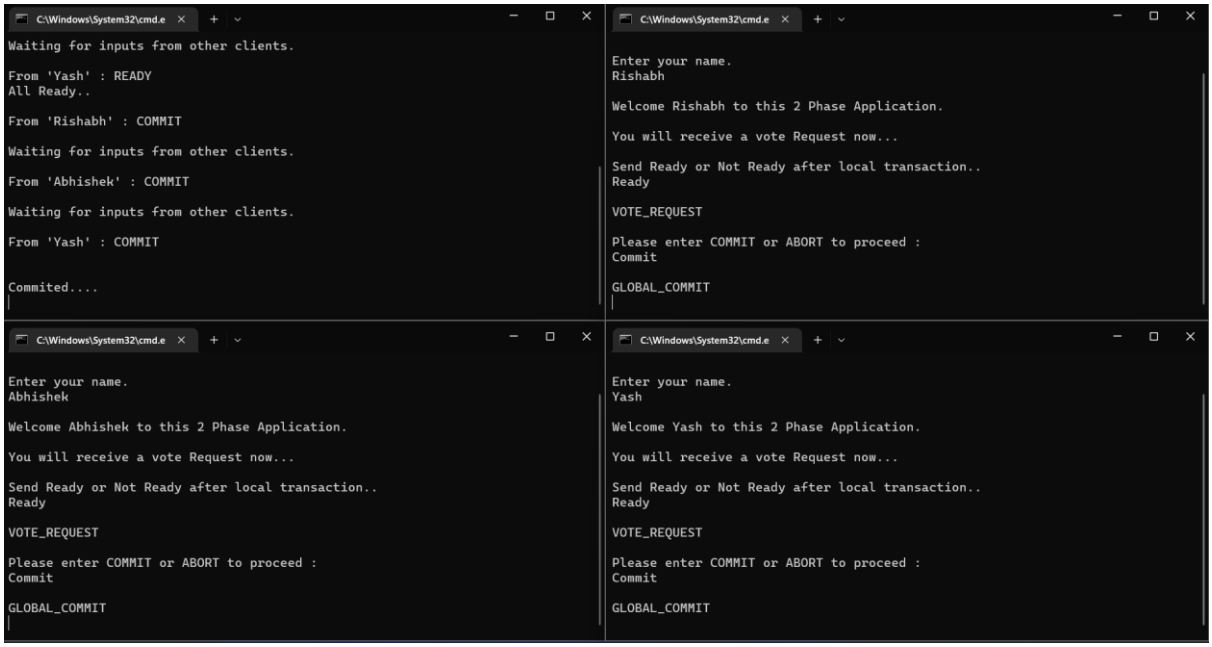
Scenario: One "not ready”



Scenario: all ready but one “Abort”



Scenario: all ready and all commit


C:\Windows\System32\cmd.e
Waiting for inputs from other clients.
From 'Yash' : READY
All Ready..
From 'Rishabh' : COMMIT
Waiting for inputs from other clients.
From 'Abhishek' : COMMIT
Waiting for inputs from other clients.
From 'Yash' : COMMIT
Commited....
C:\Windows\System32\cmd.e
Enter your name.
Rishabh
Welcome Rishabh to this 2 Phase Application.
You will receive a vote Request now...
Send Ready or Not Ready after local transaction..
Ready
VOTE_REQUEST
Please enter COMMIT or ABORT to proceed :
Commit
GLOBAL_COMMIT
C:\Windows\System32\cmd.e
Enter your name.
Abhishek
Welcome Abhishek to this 2 Phase Application.
You will receive a vote Request now...
Send Ready or Not Ready after local transaction..
Ready
VOTE_REQUEST
Please enter COMMIT or ABORT to proceed :
Commit
GLOBAL_COMMIT
C:\Windows\System32\cmd.e
Enter your name.
Yash
Welcome Yash to this 2 Phase Application.
You will receive a vote Request now...
Send Ready or Not Ready after local transaction..
Ready
VOTE_REQUEST
Please enter COMMIT or ABORT to proceed :
Commit
GLOBAL_COMMIT

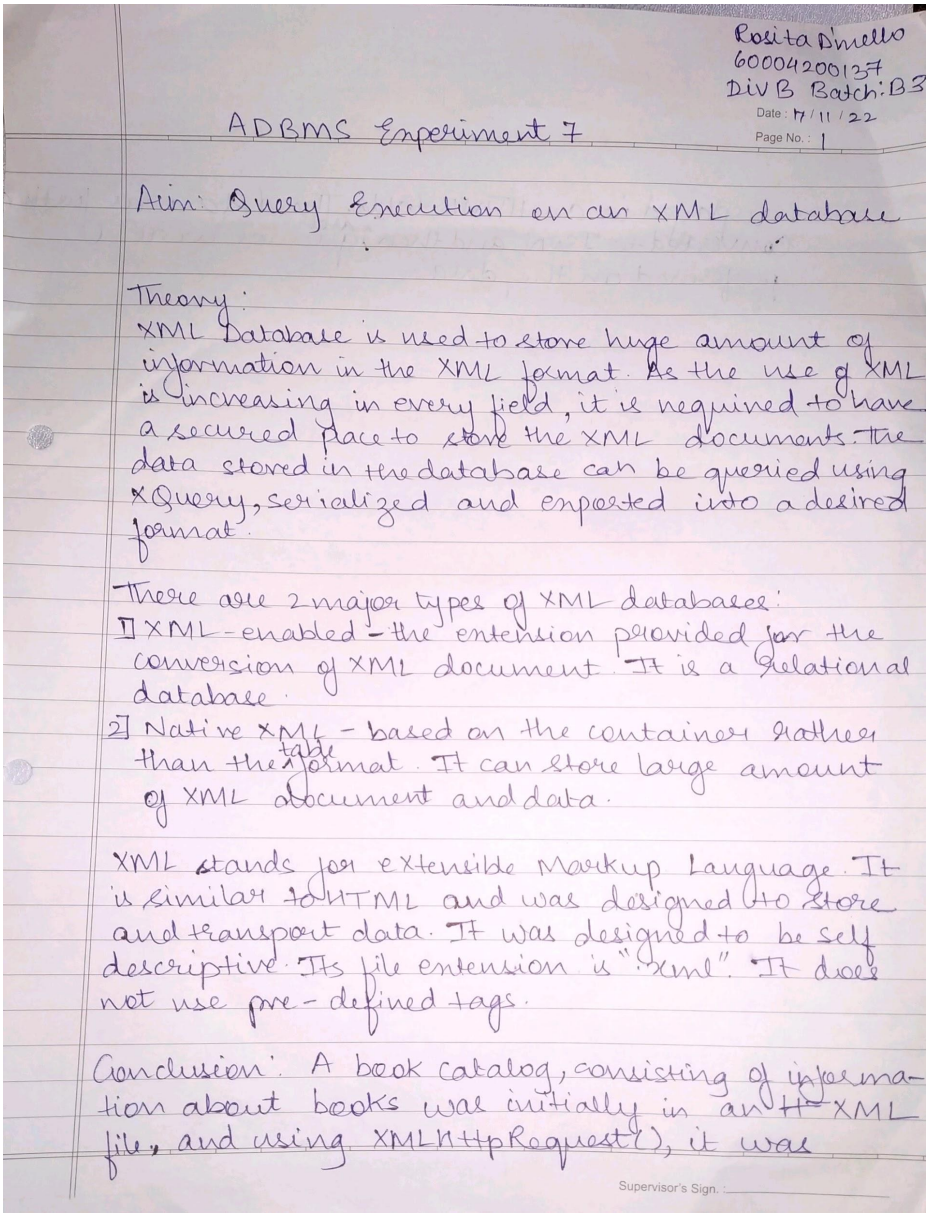
ADBMS Experiment 7

Date of Performance: 17th November 2022
Date of Submission: 17th November 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Query Execution on an XML database

Theory:



Rosita D'mello
60004200137
DivB Batch:B3
Date: 17/11/22
Page No.: 1

ADBMS Experiment 7

Aim: Query Execution on an XML database

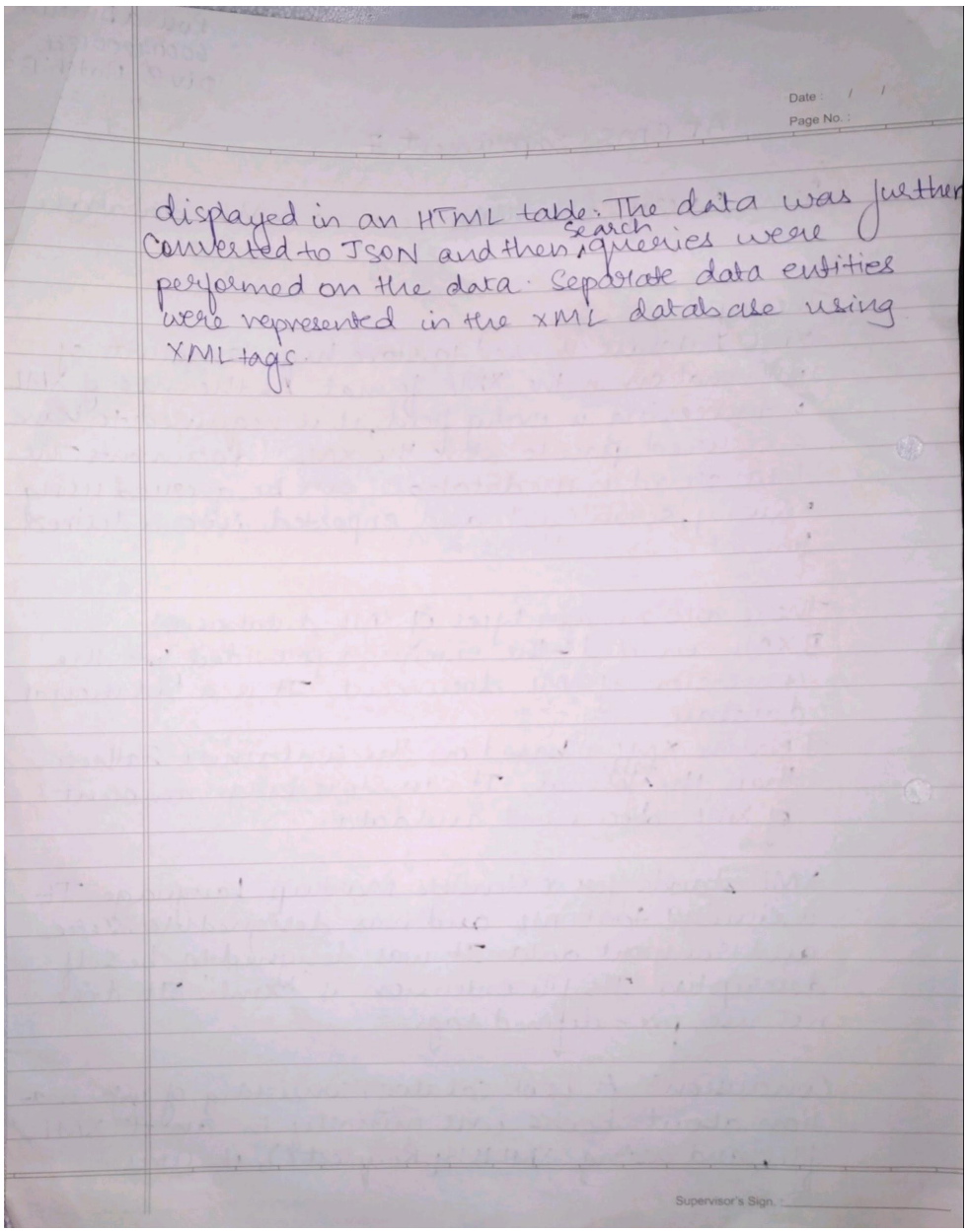
Theory:
XML Database is used to store huge amount of information in the XML format. As the use of XML is increasing in every field, it is required to have a secured place to store the XML documents. The data stored in the database can be queried using XQuery, serialized and exported into a desired format.

There are 2 major types of XML databases:
1] XML-enabled - the extension provided for the conversion of XML document. It is a Relational database.
2] Native XML - based on the container rather than the table format. It can store large amount of XML document and data.

XML stands for eXtensible Markup Language. It is similar to HTML and was designed to store and transport data. It was designed to be self descriptive. Its file extension is ".xml". It does not use pre-defined tags.

Conclusion: A book catalog, consisting of information about books was initially in an XML file, and using XMLHttpRequest(), it was

Supervisor's Sign.: _____



Code:

```
<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>
<script>
```

```
  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.toLowe
rCase()));
  console.log(res);
  if (res)
 { var
queryTable="<tr><th>Title</th><th>Author</th><th>Genre</th><th>Price</t
h><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 used:

```
<?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 midnight

Search an author

Search a genre

Search a price

Search a Publish Date

Title	Author	Genre	Price	Publish Date	
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 he
Maeve Ascendant	Corets, Eva	Fantasy	5.95	2000-11-17	After the collapse of a nanotechnology society in England, the young
Oberon's Legacy	Corets, Eva	Fantasy	5.95	2001-03-10	In post-apocalypse England, the mysterious agent known only as Ober
The Sundered Grail	Corets, Eva	Fantasy	5.95	2001-09-10	The two daughters of Maeve, half-sisters, battle one another for contro
Lover Birds	Randall, Cynthia	Romance	4.95	2000-09-02	When Carla meets Paul at an ornithology conference, tempers fly as fe
Splish Splash	Thurman, Paula	Romance	4.95	2000-11-02	A deep sea diver finds true love twenty thousand leagues beneath the s
Creepy Crawlies	Knorr, Stefan	Horror	4.95	2000-12-06	An anthology of horror stories about roaches, centipedes, scorpions an
Paradox Lost	Kress, Peter	Science Fiction	6.95	2000-11-02	After an inadvertant trip through a Heisenberg Uncertainty Device, Jan
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 programm
MSXML3: A Comprehensive Guide	O'Brien, Tim	Computer	36.95	2000-12-01	The Microsoft MSXML3 parser is covered in detail, with attention to
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

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

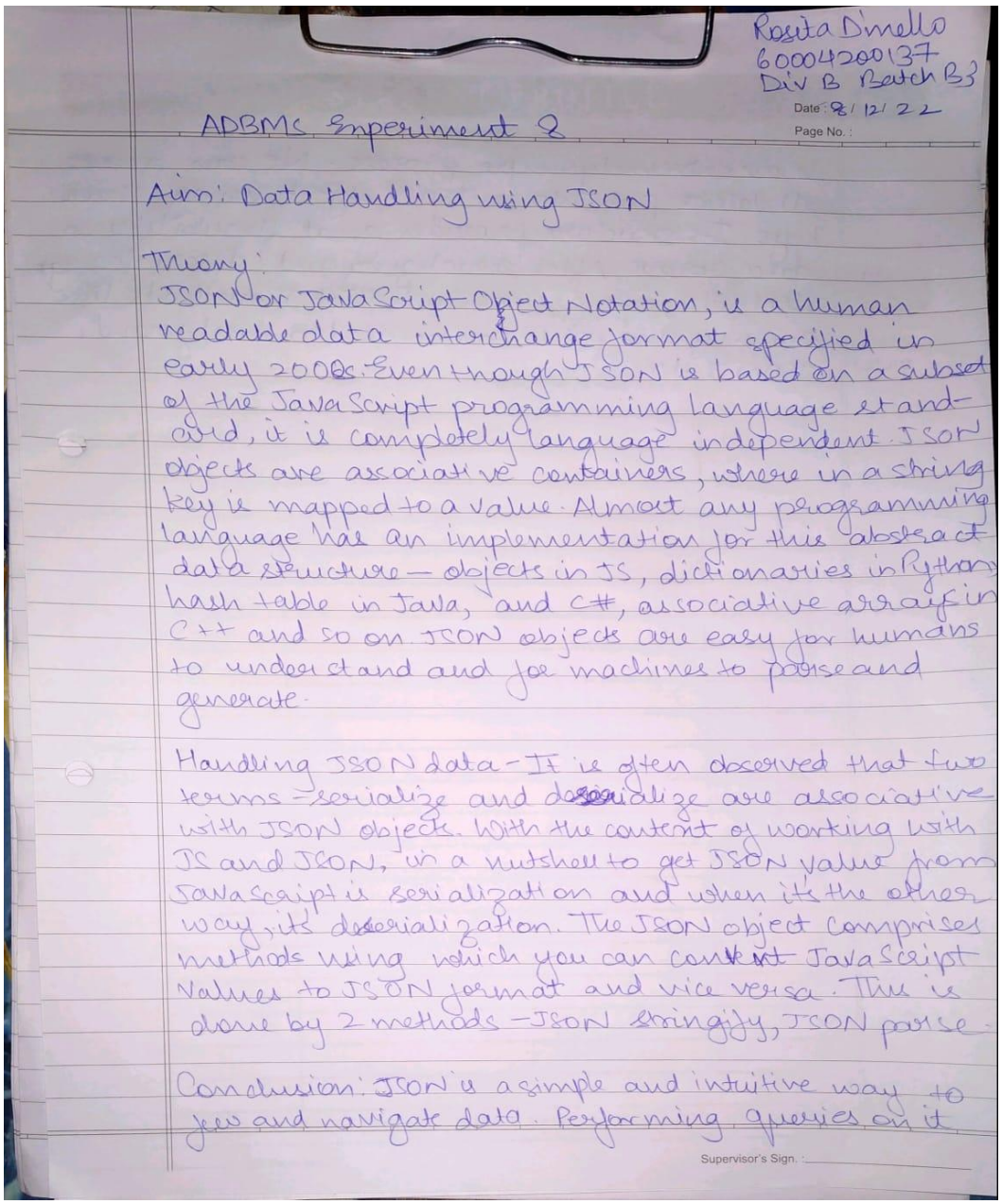
ADBMS Experiment 8

Date of Performance: 17th November 2022
Date of Submission: 4th December 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Data handling using JSON

Theory:



Rosita D'mello
60004200137
Div B Batch B3
Date: 8/12/22
Page No.:

ADBMS Experiment 8

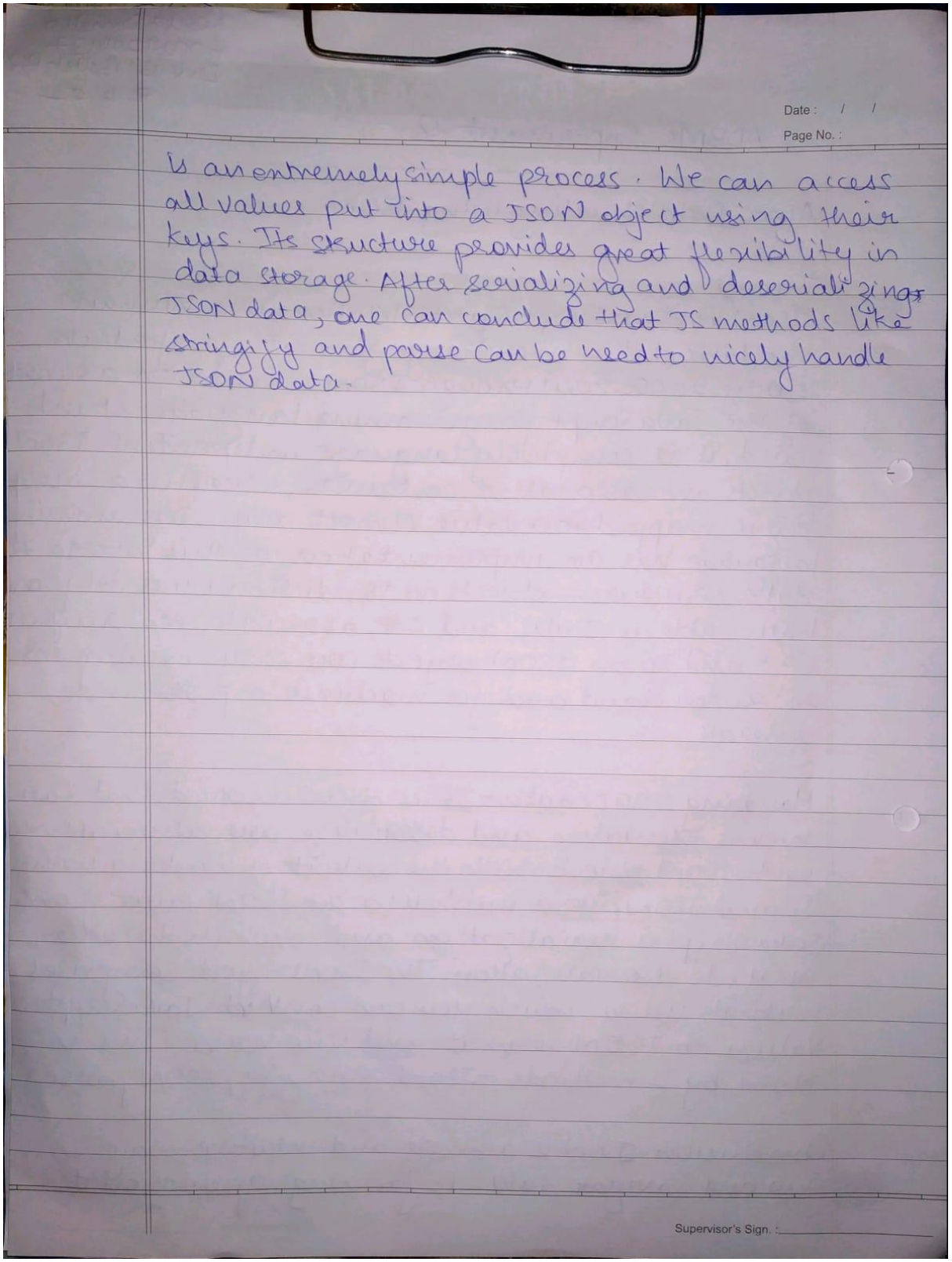
Aim: Data Handling using JSON

Theory:
JSON or JavaScript Object Notation, is a human readable data interchange format specified in early 2000s. Even though JSON is based on a subset of the JavaScript programming language standard, it is completely language independent. JSON objects are associative containers, where in a string key is mapped to a value. Almost any programming language has an implementation for this abstract data structure — objects in JS, dictionaries in Python, hash table in Java, and C#, associative array in C++ and so on. JSON objects are easy for humans to understand and for machines to parse and generate.

Handling JSON data - It is often observed that two terms - serialize and deserialize are associative with JSON objects. With the context of working with JS and JSON, in a nutshell to get JSON value from JavaScript is serialization and when it's the other way, it's deserialization. The JSON object comprises methods using which you can convert JavaScript values to JSON format and vice versa. This is done by 2 methods - JSON stringify, JSON parse.

Conclusion: JSON is a simple and intuitive way to see and navigate data. Performing queries on it

Supervisor's Sign.:


Date : / /
Page No. :

is an extremely simple process. We can access all values put into a JSON object using their keys. Its structure provides great flexibility in data storage. After serializing and deserializing JSON data, one can conclude that JS methods like stringify and parse can be used to nicely handle JSON data.

Supervisor's Sign. :_____

Code:

```
const data = {"type" :
"FeatureCollection",
```

```
"features" :
[{"type" :
"Feature",
"geometry" :
{"type" : "Point",
"coordinates" : [[122.0, 10.0],
[113.0, 2.0],
[14.0, 0.0],
[235.0, 1.0]]},
"properties" : {"Name" : "Redwood City",
"time" : 14.0}},
{"type" : "Feature",
"geometry" : {"type" : "LineString",
"coordinates" : [[102.0, 0.0],
[103.0, 1.0],
[104.0, 0.0],
[105.0, 1.0]]},
"properties" : {"Name" : "USA",
"time" : 24.0}},
{"type" : "Feature",
"geometry" : {"type" : "Polygon",
"coordinates" : [[[100.0, 0.0],
[101.0, 0.0],
[101.0, 1.0],
[100.0, 1.0],
[100.0, 0.0]]]},
"properties" : {"Name" : "Ghatkopar",
"prop1" : {"this" : "that"}}}]}
const features = data["features"]
features.map(feature => {
const coordindates = feature.geometry.coordinates;
if(coordindates[0][0] > 100 && coordindates[0][1] < 102 &&
feature.properties.time > 20){
console.log(feature.properties)
}
})
```

Output:

```
{ Name: 'USA', time: 24 }
```

ADBMS Experiment 9

Date of Performance: 17th November 2022
Date of Submission: 4th December 2022

Name: Rosita D'mello
SAP ID: 60004200137
Batch: B3
Branch: Computer Engineering

Aim: Processing of Spatial and Temporal data

Theory:

Rosita D'mello
60004200137
DivB Batch B3

Date : 4/12/22
Page No. :

ADBMS Experiment 9

Aim: Processing of Spatial and Temporal Data.

Theory:

Temporal Data: A Temporal Database is a database with built-in support for handling time sensitive data. Usually, databases store information only about current state, and not about past states. But for many applications, it is important to maintain the past values, and the time at which the data was updated. That is, the knowledge of evaluation is required. Time dependant data is called temporal data.

Spatial Data: A spatial database is a database that is enhanced to store and access spatial data or data that defines a geometric space. These data are often associated with geographic locations and features, or constructed features like cities. Data on spatial databases handle more complex data like 3D objects, topological coverage and linear networks. Aside from the indexes, spatial databases also offer spatial data types in their data model and query language.

Conclusion: In many cases like medical history, insurance polices' claims, etc., knowledge of history is required. Temporal database queries help in these cases to quickly retrieve evolution of the records. Spatial Databases are used in cases like GIS (Geographic Information Systems) to store and retrieve data associated with location.

Supervisor's Sign. :

Queries:

Temporal:

```
mysql> select * from 2022_july where start_time < "04:00:00";
+------------+------------+----------+----------------------+------------+-------------------------------+
| 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     |                               |
+------------+------------+----------+----------------------+------------+-------------------------------+
26 rows in set (0.00 sec)
```

Spatial:

```
mysql> select * from 2022_july where distance>10000;
+------------+------------+----------+----------------------+------------+----------------------------+
| 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 |
+------------+------------+----------+----------------------+------------+----------------------------+
9 rows in set (0.03 sec)
```

```
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)
```

Data:

```
mysql> select * from 2022_july;
+------------+------------+----------+----------------------+------------+-----------------------------------------------------+
| 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   |     2203 | 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   |     9105 | 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                                         |
```

Rosita D'mello
60004200137
Div B Batch: B3
Date of submission: 10/11/22

ADBMS Experiment 10

Aim: Case Study on Database Security Issues and patents related to it.

Case Study:-

Title: Secure Data Storage System and Method.
US 8 201261 B2 - United States.

Inventors: Chase Barfield, Jason Cornell, Jeff Arbour

Application US12/430,643 events:
2009-04-27 - Application filed by Chase Barfield, Jason Cornell, Jeff Arbour.
2009-04-27 - Priority to US12/430,643
2010-10-28 - Publication of US20100275005A1
2012-06-12 - Publication of US8201261B2
Status - Active
2030-03-14 - Adjusted expiration.

Abstract/Introduction:
A system and method for securely storing data within a network. Data stored on the network connected primary server is initially encrypted. The IP address of the primary server is sent to the secondary server, and a message is received from the secondary server indicating pending orders. If the instructions indicate that the primary server has been attacked, data stored on the primary server is re-encrypted and the IP address of the primary server is sent to the secondary server. Detects unauthorized access to the primary server,

and deleted the data stored on the primary server when there is unauthorized access for a certain number of times.

Claims:

Claim 1 explains a method for securely storing information in a network. It is a 3 step process -

Step 1 - Encrypt data stored on a primary server connected to a network.

Step 2 - Send IP address of the primary server to a second server via the network.

Step 3 - Receive pending instructions from second server

If the second server detects that the data on the primary server has been compromised, it will re-encrypt the data and send the IP address back to server 2

If the primary server detects unauthorized access attempts that cross a predetermined threshold, it will initiate a data erase.

Claim 2 talks about the method of claim 1, which is a method for determining unauthorized access of the primary server. The method includes receiving a password entered via the primary server, repeating the step of receiving a password is not accepted, and receiving a predetermined number of passwords that are not accepted

Claim 3 explains that the method of claim 2 is a method for security in case the wrong password is inputted after a certain number of times. If the predetermined number of passwords are received

but not accepted, then access to the data stored on the primary server is disabled and the data is re-encrypted for more no. of times, after which if the same continues, then the data stored on the primary server is erased.

Claim 4 explains that the method of claim 1 is a method for sending data from a primary server to a secondary server. The primary server sends its IP address to the secondary server, which uses the IP address to determine the primary server's physical location.

Claim 5 tells us that following the method in Claim 1, after erasing the data stored on the primary server, the OS in the Primary Server is erased if a predetermined number of passwords is received but not accepted.

Claim 6 explains a system for secure storage of data in a network.

The following are coupled to the network:

1. A computer (also coupled to at least 1 data storage device)
2. A server, located at the premises on which the computer is situated.
3. A security key device.
4. A second computer.

Data stored on data storage devices is encrypted. Access to the storage device is enabled only once the connection is established. The computer checks for messages from the second computer that indicate pending special instructions. Data stored on the storage device is re-encrypted if certain instructions received in the communication indicate that the computer has been attacked. If no special instructions are received, data stored on the

storage device will be erased after a predetermined number of consecutive unauthorized access attempts.
Claim 7 explains that to implement claim 6, we follow these steps: receive a password from a PC connected to the server. If the password is not accepted, repeat the receiving step.
If a predetermined no. of passwords have been received but not accepted, the system first locks each data storage device and re-encrypts the data stored on it, then after a predetermined no. of additional attempts, the data on it is erased.
Claim 8 explains that in the system of claim 6, the computer is prevented from accessing data stored on the storage device when it is disabled.
Claim 9 tells us that the password is entered via the computer in the system described by claim 6.
Claim 10 is that in the system explained by Claim 6, an IP address is sent to the server and used to determine a physical location of the computer.

Conclusion: Electronic Security for storage devices is generally handled at the network level and does not take into account the possibility of an attack within the network. This patented system ensures that the data in the servers and other storage devices is not compromised or stolen, especially by trying different password combinations. Data is first re-encrypted in risky situations and then erased as an extreme measure.