NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

a) 1.a Database Preparation for apartment management system.

```
import random
import sqlite3
conn = sqlite3.connect('DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db')
c = conn.cursor()
c.execute('''CREATE TABLE IF NOT EXISTS Tenants
             (tenant_id INTEGER PRIMARY KEY,
             name TEXT,
             apartment_no TEXT,
             contact_number TEXT)''')

c.execute('''CREATE TABLE IF NOT EXISTS Apartments
             (apartment_no TEXT PRIMARY KEY,
             num_bedrooms INTEGER,
             rent_amount REAL)''')

c.execute('''CREATE TABLE IF NOT EXISTS MaintenanceRequests
             (request_id INTEGER PRIMARY KEY,
             apartment_no TEXT,
             issue_description TEXT,
             request_date DATE,
             status TEXT)''')

c.execute('''CREATE TABLE IF NOT EXISTS Payments
             (payment_id INTEGER PRIMARY KEY,
             tenant_id INTEGER,
             amount REAL,
             payment_date DATE)''')

c.execute('''CREATE TABLE IF NOT EXISTS Employees
             (employee_id INTEGER PRIMARY KEY,
             name TEXT,
             position TEXT,
             contact_number TEXT)''')

conn.commit()
conn.close()
```

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Inserting the records to the table

Projects > Database > Apartment > Insert_data.py

```python
import sqlite3
import random
conn = sqlite3.connect('DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db')
c = conn.cursor()

def generate_indian_name():
    first_names = ["Aditya", "Aarav", "Aanya", "Arjun", "Aisha", "Amit", "Deepika", "Dev", "Esha", "Gautam", "Ishaan", "Kavya", "Krish",
    "Meera", "Neha", "Nikhil", "Priya", "Rahul", "Riya", "Sahil", "Shreya", "Tanvi", "Varun", "Vidya", "Yash"]
    last_names = ["Agarwal", "Bhatt", "Chopra", "Desai", "Gupta", "Joshi", "Kumar", "Mishra", "Patel", "Sharma", "Singh", "Verma"]
    return random.choice(first_names) + " " + random.choice(last_names)

for i in range(1, 10001):
    name = generate_indian_name()
    apartment_no = "A" + str(random.randint(1, 1000))
    contact_number = "+91" + str(random.randint(6000000000, 9999999999))
    c.execute("INSERT INTO Tenants (name, apartment_no, contact_number) VALUES (?, ?, ?)", (name, apartment_no, contact_number))

for i in range(1, 1001):
    apartment_no = "A" + str(i)
    num_bedrooms = random.randint(1, 4)
    rent_amount = random.uniform(10000, 50000)
    c.execute("INSERT INTO Apartments (apartment_no, num_bedrooms, rent_amount) VALUES (?, ?, ?)", (apartment_no, num_bedrooms, rent_amount))

for i in range(1, 10001):
    apartment_no = "A" + str(random.randint(1, 1000))
    issue_description = "Issue #" + str(i)
    request_date = "2024-03-18"  # Example date
    status = random.choice(["Pending", "In Progress", "Completed"])
    c.execute("INSERT INTO MaintenanceRequests (apartment_no, issue_description, request_date, status) VALUES (?, ?, ?, ?)", (apartment_no, issue_description, request_date, status))

for i in range(1, 5001):
    tenant_id = random.randint(1, 10000)
    amount = random.uniform(5000, 20000)
    payment_date = "2024-03-18"  # Example date
    c.execute("INSERT INTO Payments (tenant_id, amount, payment_date) VALUES (?, ?, ?)", (tenant_id, amount, payment_date))

for i in range(1, 101):
    name = generate_indian_name()
    position = random.choice(["Manager", "Maintenance Staff", "Security Guard"])
    contact_number = "+91" + str(random.randint(6000000000, 9999999999))
    c.execute("INSERT INTO Employees (name, position, contact_number) VALUES (?, ?, ?)", (name, position, contact_number))
conn.commit()
conn.close()
```

Inserted Records:

Tenants:

Projects > Database > Apartment > DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db

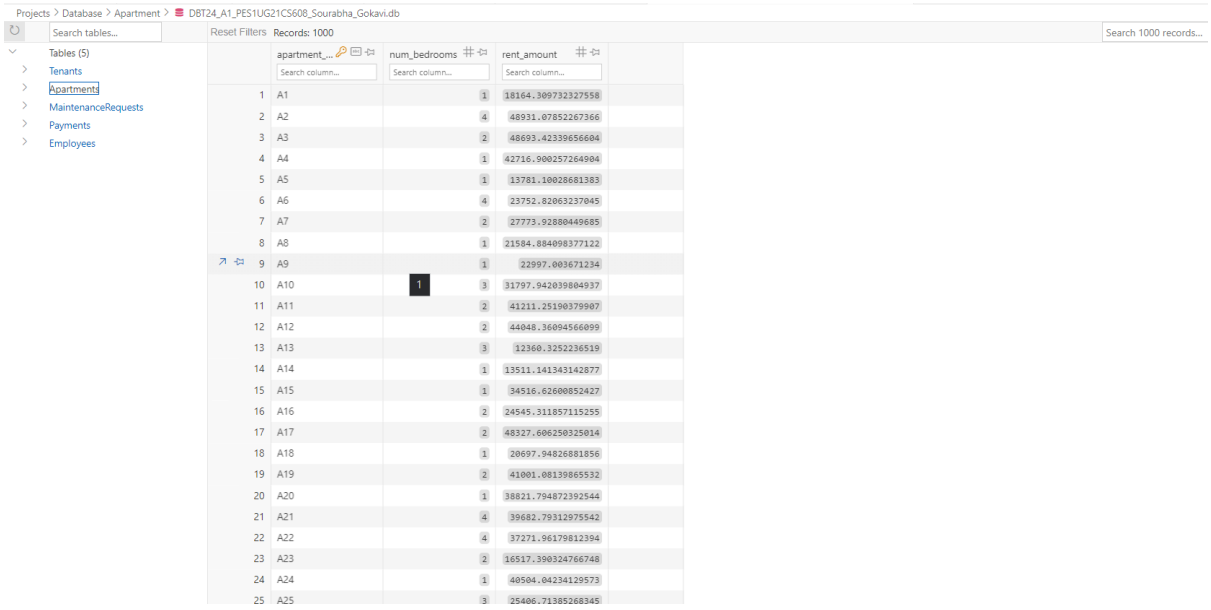
Search tables... | Reset Filters | Records: 10000 | Search 10000 records...

Tables (5)
- Tenants
- Apartments
- MaintenanceRequests
- Payments
- Employees

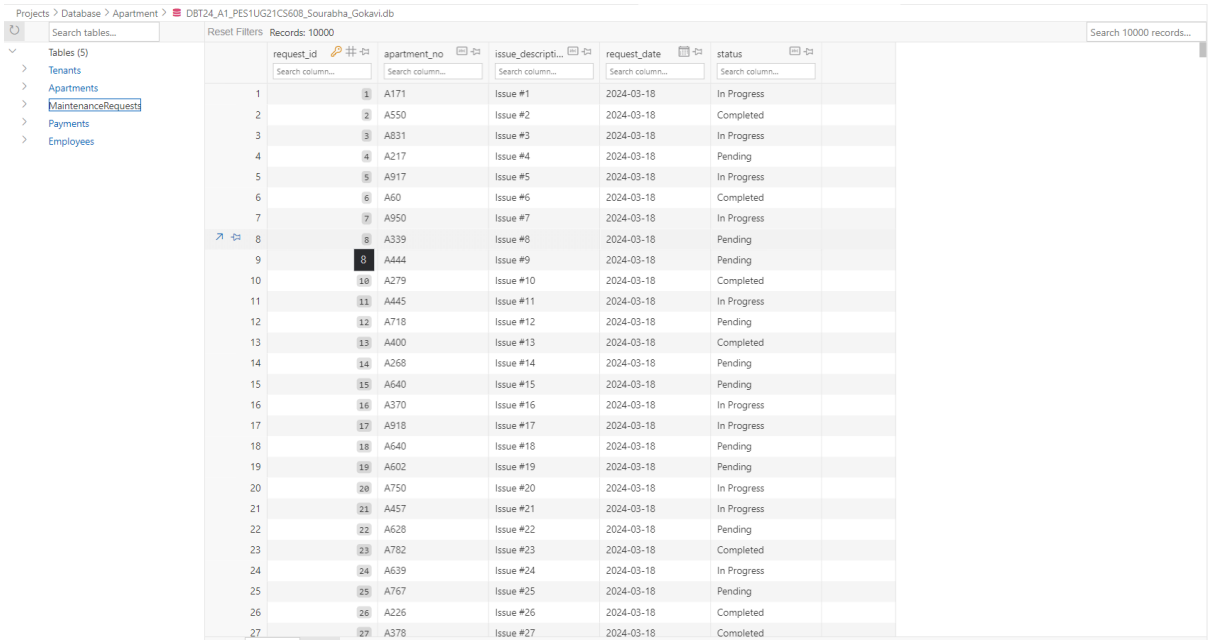
	tenant_id	name	apartment_no	contact_number
1	1	Varun Desai	A677	+916570104785
2	2	Aanya Joshi	A71	+916027495518
3	3	Riya Chopra	A651	+917536385773
4	4	Dev Verma	A985	+917987109762
5	5	Amit Bhatt	A753	+918244585213
6	6	Priya Bhatt	A992	+917018629407
7	7	Deepika Agarwal	A104	+919228410944
8	8	Esha Bhatt	A835	+917986498861
9	9	Arjun Sharma	A466	+916292149074
10	10	Yash Verma	A980	+916241022802
11	11	Nikhil Verma	A846	+916541183490
12	12	Neha Agarwal	A417	+917788981538
13	13	Deepika Singh	A521	+917361150144
14	14	Aarav Verma	A381	+919823063155
15	15	Aditya Kumar	A448	+916295850722
16	16	Aditya Patel	A904	+919399171763
17	17	Kavya Sharma	A916	+919372057947
18	18	Dev Chopra	A243	+918932112831
19	19	Amit Sharma	A865	+917879543990
20	20	Deepika Sharma	A918	+916236657310
21	21	Shreya Kumar	A356	+919485386251

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Apartment:

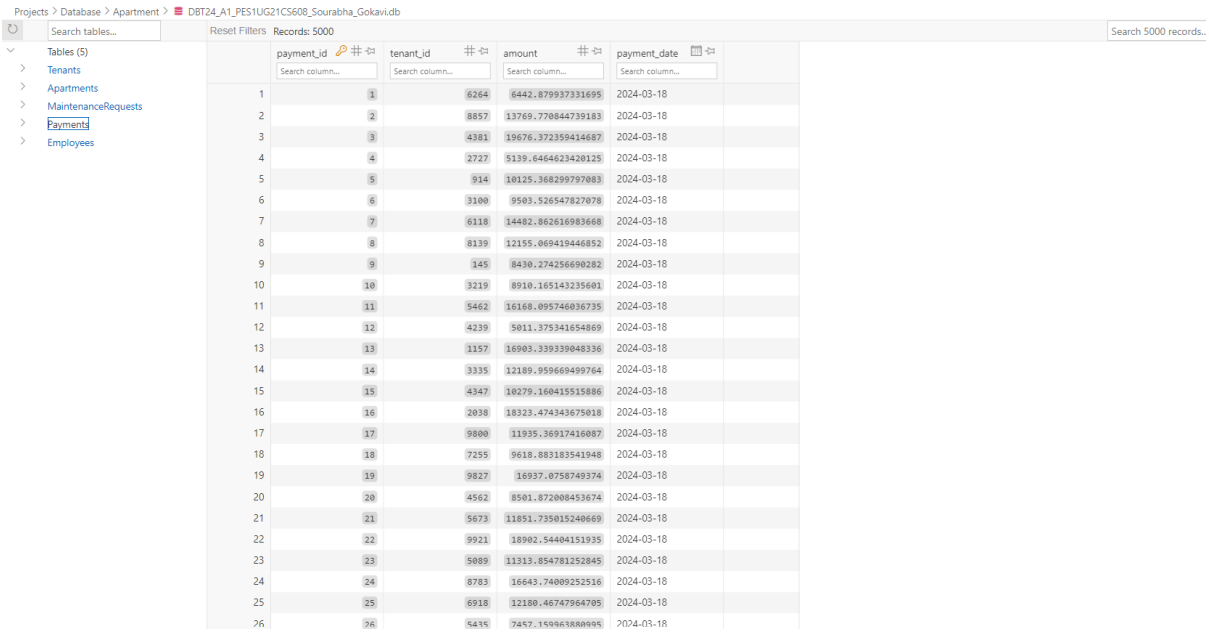


MaintainanceRequest:



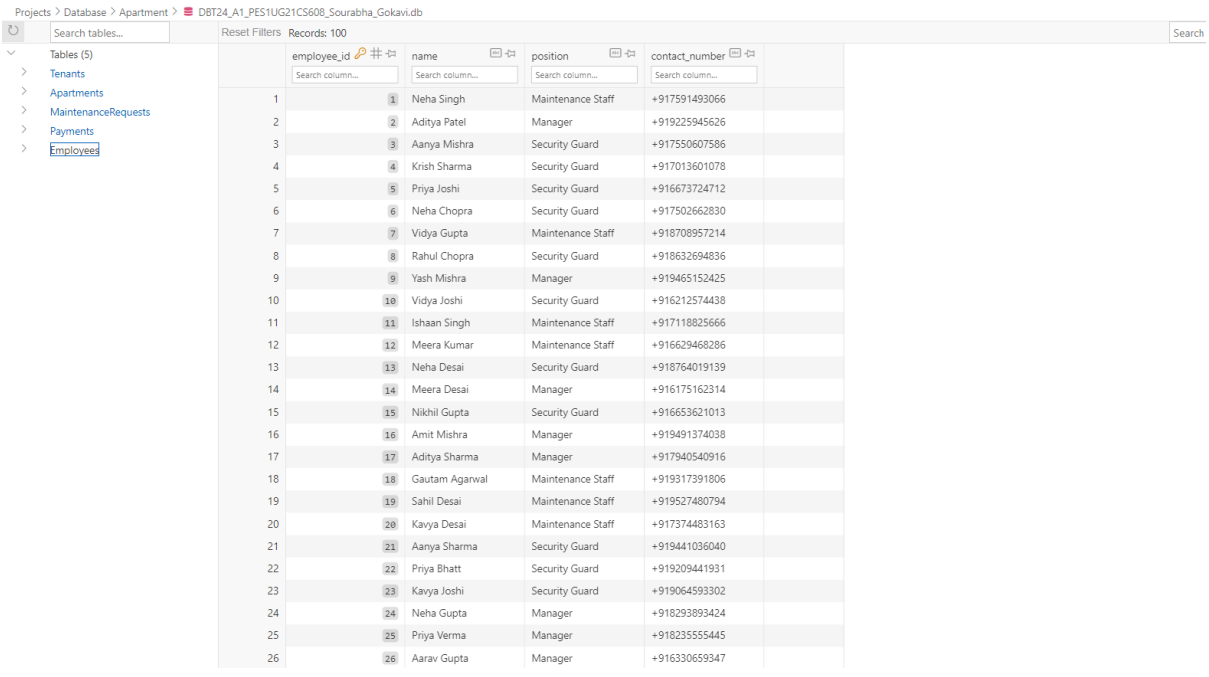
NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Payments:



	payment_id	tenant_id	amount	payment_date
1	1	6264	6442.879937331695	2024-03-18
2	2	8857	13769.770844739183	2024-03-18
3	3	4381	19676.372359414687	2024-03-18
4	4	2727	5139.646462342012 5	2024-03-18
5	5	914	10125.368299797083	2024-03-18
6	6	3100	9503.526547827078	2024-03-18
7	7	6118	14482.862616983668	2024-03-18
8	8	8139	12155.069419446852	2024-03-18
9	9	145	8430.274256690282	2024-03-18
10	10	3219	8910.165143235601	2024-03-18
11	11	5462	16168.095746036735	2024-03-18
12	12	4239	5011.375341654869	2024-03-18
13	13	1157	16903.339339048336	2024-03-18
14	14	3335	12189.959669499764	2024-03-18
15	15	4347	10279.160415515886	2024-03-18
16	16	2038	18323.474343675018	2024-03-18
17	17	9800	11935.36917416087	2024-03-18
18	18	7255	9618.883183541948	2024-03-18
19	19	9827	16937.0758749374	2024-03-18
20	20	4562	8501.872008453674	2024-03-18
21	21	5673	11851.735015240669	2024-03-18
22	22	9921	18902.54404151935	2024-03-18
23	23	5089	11313.854781252845	2024-03-18
24	24	8783	16643.74009252516	2024-03-18
25	25	6918	12180.46747964705	2024-03-18
26	26	5435	7457.159963880995	2024-03-18

Employees:



	employee_id	name	position	contact_number
1	1	Neha Singh	Maintenance Staff	+917591493066
2	2	Aditya Patel	Manager	+919225945626
3	3	Aanya Mishra	Security Guard	+917550607586
4	4	Krish Sharma	Security Guard	+917013601078
5	5	Priya Joshi	Security Guard	+916673724712
6	6	Neha Chopra	Security Guard	+917502662830
7	7	Vidya Gupta	Maintenance Staff	+918708957214
8	8	Rahul Chopra	Security Guard	+918632694836
9	9	Yash Mishra	Manager	+919465152425
10	10	Vidya Joshi	Security Guard	+916212574438
11	11	Ishaan Singh	Maintenance Staff	+917118825666
12	12	Meera Kumar	Maintenance Staff	+916629468286
13	13	Neha Desai	Security Guard	+918764019139
14	14	Meera Desai	Manager	+916175162314
15	15	Nikhil Gupta	Security Guard	+916653621013
16	16	Amit Mishra	Manager	+919491374038
17	17	Aditya Sharma	Manager	+917940540916
18	18	Gautam Agarwal	Maintenance Staff	+919317391806
19	19	Sahil Desai	Maintenance Staff	+919527480794
20	20	Kavya Desai	Maintenance Staff	+917374483163
21	21	Aanya Sharma	Security Guard	+919441036040
22	22	Priya Bhatt	Security Guard	+919209441931
23	23	Kavya Joshi	Security Guard	+919064593302
24	24	Neha Gupta	Manager	+918293893424
25	25	Priya Verma	Manager	+918235555445
26	26	Aarav Gupta	Manager	+916330659347

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

b) Queries Creation and Performance Measurement.

Projects > Database > Apartment > b_Performance.py

```python
import sqlite3
conn = sqlite3.connect('DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db')
c = conn.cursor()

def select_all_and_count(table_name):
    c.execute(f"SELECT * FROM {table_name}")
    rows = c.fetchall()
    row_count = len(rows)
    print(f"Table: {table_name} ({row_count} rows)")
    for row in rows:
        print(row)
    print("\n")

table_names = ["Tenants", "Apartments", "MaintenanceRequests", "Payments", "Employees"]

for table_name in table_names:
    select_all_and_count(table_name)

# Index scan
c.execute("SELECT * FROM Tenants WHERE tenant_id = 100")  # Assuming tenant_id is indexed
print("Index Scan Example:")
print(c.fetchone())

# Table scan
c.execute("SELECT * FROM Apartments WHERE num_bedrooms = 3")  # Assuming num_bedrooms is not indexed
print("\nTable Scan Example:")
print(c.fetchone())

# Multi-table join
c.execute('''SELECT Tenants.name, Apartments.apartment_no, MaintenanceRequests.issue_description
             FROM Tenants
             JOIN Apartments ON Tenants.apartment_no = Apartments.apartment_no
             JOIN MaintenanceRequests ON MaintenanceRequests.apartment_no = Apartments.apartment_no
             WHERE MaintenanceRequests.status = 'Pending' ''')
print("\nMulti-table Join Example:")
print(c.fetchall())
conn.close()
```

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Output :

```
(7436, 'Esha Mishra', 'A940', '+917006612544')
(7437, 'Rahul Chopra', 'A219', '+919218701960')
(7438, 'Vidya Sharma', 'A811', '+916357302665')
(7439, 'Ishaan Mishra', 'A64', '+918544425699')
(7440, 'Riya Desai', 'A813', '+918227068915')
(7441, 'Aarav Joshi', 'A696', '+917045291549')
(7442, 'Aisha Patel', 'A847', '+919988178859')
(7443, 'Neha Joshi', 'A549', '+919791973936')
(7444, 'Aditya Desai', 'A113', '+917589827129')
(7445, 'Nikhil Singh', 'A132', '+916738315964')
(7446, 'Esha Patel', 'A545', '+917595922358')
(7447, 'Nikhil Agarwal', 'A366', '+918626716307')
(7448, 'Aanya Kumar', 'A951', '+918442719286')
(7449, 'Tanvi Chopra', 'A66', '+917891147924')
(7450, 'Esha Sharma', 'A370', '+918825900574')
(7451, 'Yash Mishra', 'A10', '+917008523467')
(7452, 'Aarav Gupta', 'A726', '+917660316427')
(7453, 'Aarav Desai', 'A983', '+919268912019')
(7454, 'Riya Joshi', 'A588', '+919609910039')
(7455, 'Neha Agarwal', 'A237', '+916046249724')
(7456, 'Vidya Sharma', 'A728', '+918053200305')
(7457, 'Yash Gupta', 'A875', '+916466457134')
(7458, 'Aisha Chopra', 'A14', '+916317455679')
(7459, 'Aanya Kumar', 'A243', '+919128864698')
```

c) Indexing for Query Performance Improvement.

Projects > Database > Apartment > Index_creation.py

```python
import sqlite3
conn = sqlite3.connect('DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db')
c = conn.cursor()
# Creatig the indexe on the Tenants table
c.execute("CREATE INDEX IF NOT EXISTS idx_tenant_apartment_no ON Tenants (apartment_no)")
c.execute("CREATE INDEX IF NOT EXISTS idx_tenant_contact_number ON Tenants (contact_number)")

# Creating the indexe on the Apartments table
c.execute("CREATE INDEX IF NOT EXISTS idx_apartment_num_bedrooms ON Apartments (num_bedrooms)")

# Creating the indexe on the MaintenanceRequests table
c.execute("CREATE INDEX IF NOT EXISTS idx_maintenance_apartment_no ON MaintenanceRequests (apartment_no)")
c.execute("CREATE INDEX IF NOT EXISTS idx_maintenance_status ON MaintenanceRequests (status)")
conn.commit()

#query part
c.execute('''EXPLAIN QUERY PLAN
            SELECT * FROM Tenants WHERE apartment_no = "A456" ''')
print("Query Plan without Index:")
print(c.fetchall())

# Runnig the query with index
c.execute('''EXPLAIN QUERY PLAN
            SELECT * FROM Tenants WHERE apartment_no = 'A456' ''')
print("\nQuery Plan with Index:")
print(c.fetchall())

conn.close()
```

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Output:

```
C:\PES\26\Projects\Database\Apartment>python Index_creation.py
Query Plan without Index:
[(3, 0, 0, 'SEARCH Tenants USING INDEX idx_tenant_apartment_no (apartment_no=?)')]

Query Plan with Index:
[(3, 0, 0, 'SEARCH Tenants USING INDEX idx_tenant_apartment_no (apartment_no=?)')]
```

d) Query Optimization with Varied Join Orders and Types

Projects > Database > Apartment > Query_optimization.py

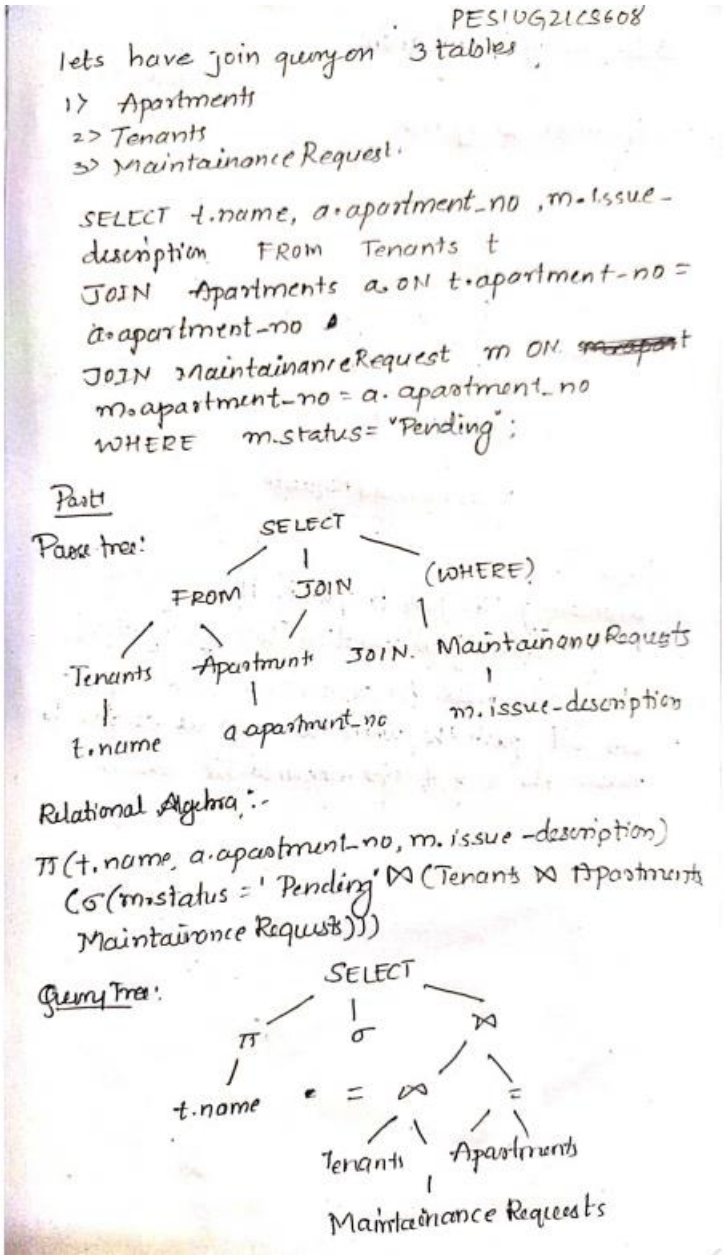
```python
# query optimization with varied join order
import sqlite3
import time
# Connect to the SQLite database
conn = sqlite3.connect('DBT24_A1_PES1UG21CS608_Sourabha_Gokavi.db')
c = conn.cursor()

# query
query = '''
        SELECT t.name, a.apartment_no, m.issue_description
        FROM Tenants t
        JOIN Apartments a ON t.apartment_no = a.apartment_no
        JOIN MaintenanceRequests m ON m.apartment_no = a.apartment_no
        WHERE m.status = 'Pending'
        '''
# calculating the time
# query execution before optimization
start_time = time.time()
c.execute(query)
result_before = c.fetchall()
before = time.time() - start_time
print("Time(Before):", before)

# changing the join order
# Changing join order and using LEFT OUTER JOIN
new_qurey = '''
            SELECT t.name, a.apartment_no, m.issue_description
            FROM MaintenanceRequests m
            LEFT OUTER JOIN Apartments a ON m.apartment_no = a.apartment_no
            JOIN Tenants t ON t.apartment_no = a.apartment_no
            WHERE m.status = 'Pending'
            '''
            # calculatin the time
start_time = time.time()
c.execute(new_qurey)
result_after = c.fetchall()
new_time = time.time() - start_time
print("Time(After):", new_time)

print("Optimized or not (Trur / False)---->", new_time<before)

conn.close()
```

Output:

```
C:\PES\26\Projects\Database\Apartment>python Query_optimization.py
Time(Before): 0.09489083290100098
Time(After): 0.09401369094848633
Optimized or not (Trur / False)----> True
```

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

e) Query Analysis and Optimization:

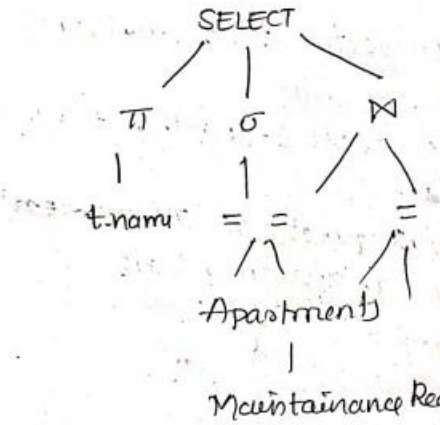


NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System

Part 2: Query Optimization:

PES1UG21CS608
Sourabha Gokavi

Optimized Query Tree



Steps

1) Reordering the join to optimize it.
2) Using correct join operation (like left, right --)
3) Using the index for performance optimization
4) we will push the where as down as possible to reduce the size of the intermediate results.

Relational Schema:

NAME: SOURABHA G GOKAVI
SUB: Data-Base Technologies (Assignment 1)
SRN: PES1UG21CS608
Topic: Apartment Management System