EXPERIMENT 1:

Structured Query Execution for Information Retrieval

Zoya Momin

Department of Computer Engineering

M.H Saboo Siddik College of Engineering

Mumbai, India

zoya.221257.co@mhssce.ac.in

1. INTRODUCTION

Information Retrieval (IR) focuses on obtaining relevant data from vast collections using query-based mechanisms. Structured queries like SQL provide precise control, whereas unstructured queries mirror natural language interactions. This experiment aims to explore structured queries, their execution, and performance evaluation while contrasting them with unstructured approaches.Applications include search engines, digital libraries, biomedical databases, and enterprise data systems. By combining structured queries with advanced IR techniques, systems can offer faster, more accurate, and context-aware search capabilities.

QUERIES:

1. Creating PERSON table

Schema SQL

```sql
CREATE TABLE coder (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    rating DECIMAL(3,1), -- rating out of 10
    dob DATE,
    language VARCHAR(30) -- main programming language
);


INSERT INTO coder (name, age, rating, dob, language) VALUES
('Alice Johnson', 25, 8.5, '2000-05-12', 'Python'),
('Brian Smith', 30, 9.2, '1995-03-28', 'JavaScript'),
('Catherine Lee', 27, 7.8, '1998-11-15', 'Java'),
('David Brown', 22, 9.0, '2003-01-04', 'C++'),
('Emily Davis', 29, 8.1, '1996-07-20', 'Ruby'),
('Frank Wilson', 26, 9.5, '1999-09-13', 'Go'),
('Grace Miller', 24, 8.3, '2001-02-18', 'C#'),
('Henry Thompson', 28, 7.9, '1997-08-30', 'PHP'),
('Isabella White', 23, 8.7, '2002-04-25', 'Swift'),
('Jack Harris', 31, 9.4, '1994-06-09', 'TypeScript');
```

1. Retrieve all records from the "employees" table

SELECT * FROM coder;

Query #1 Execution time: 0.27ms

id	name	age	rating	dob	language
1	Alice Johnson	25	8.5	2000-05-12	Python
2	Brian Smith	30	9.2	1995-03-28	JavaScript
3	Catherine Lee	27	7.8	1998-11-15	Java
4	David Brown	22	9.0	2003-01-04	C++
5	Emily Davis	29	8.1	1996-07-20	Ruby
6	Frank Wilson	26	9.5	1999-09-13	Go
7	Grace Miller	24	8.3	2001-02-18	C#
8	Henry Thompson	28	7.9	1997-08-30	PHP
9	Isabella White	23	8.7	2002-04-25	Swift
10	Jack Harris	31	9.4	1994-06-09	TypeScript

2. Retrieve names from coder where language is java
SELECT name FROM coder WHERE language = “Python”;

name
Alice Johnson

3. Retrieve the average age of the coder
SELECT AVG(age) AS avg_age FROM coder;

avg_age
26.5000

4. Sort coder by age in ascending order
SELECT name, age FROM coder ORDER BY age ASC;

Query #1 Execution time: 0.28ms

name	age
David Brown	22
Isabella White	23
Grace Miller	24
Alice Johnson	25
Frank Wilson	26
Catherine Lee	27
Henry Thompson	28
Emily Davis	29
Brian Smith	30
Jack Harris	31

5. Retrieve total avg rating per language category.
SELECT language, AVG(rating) AS avg_rating FROM coder GROUP BY language;

Query #2 Execution time: 0.32ms

language	avg_rating
C++	8.93333
JavaScript	8.90000
Python	8.06667
Swift	8.70000

6. Retrieve coder names with rating above avg
SELECT name FROM coder WHERE rating > (SELECT AVG(rating) FROM coder);

Query #3 Execution time: 0.21ms

name
Brian Smith
David Brown
Frank Wilson
Isabella White
Jack Harris

7. List coders who code in Python or Java

SELECT name FROM coder WHERE language = ‘Python' UNION SELECT name FROM coder WHERE language = ‘Java';

Query #1 Execution time: 0.45ms

name
Alice Johnson
Catherine Lee
Henry Thompson

8. Coders born between Jan 1, 1998 and Jan 1, 2001

SELECT name FROM coder WHERE dob BETWEEN "1998-01-01" AND “2001-01-01";

Query #1 Execution time: 0.33ms

name
Alice Johnson
Catherine Lee
Frank Wilson

Insert for join

```sql
CREATE TABLE coder (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    rating DECIMAL(3,1),
    dob DATE,
    language VARCHAR(30)
);

INSERT INTO coder (name, age, rating, dob, language) VALUES
('Alice Johnson', 25, 9.2, '1999-04-15', 'Python'),
('Brian Smith', 30, 8.5, '1994-07-22', 'Java'),
('Charlotte Brown', 27, 7.8, '1997-03-10', 'C++'),
('David Wilson', 22, 8.0, '2002-11-05', 'Python'),
('Ella Thompson', 29, 9.5, '1995-09-18', 'JavaScript'),
('Frank Harris', 35, 6.9, '1989-01-25', 'Java'),
('Grace Miller', 26, 8.8, '1998-06-30', 'Go'),
('Henry Davis', 28, 7.5, '1996-12-12', 'C#'),
('Isabella Moore', 24, 9.0, '2000-05-14', 'Python'),
('Jack Taylor', 31, 7.2, '1993-08-03', 'Ruby');

CREATE TABLE projects (
    id SERIAL PRIMARY KEY,
    project_name VARCHAR(100),
    language VARCHAR(30)
);

INSERT INTO projects (project_name, language) VALUES
('E-Commerce Website', 'Python'),
('Banking System', 'Java'),
('Game Engine', 'C++'),
('Chatbot AI', 'Python'),
('Weather App', 'JavaScript'),
('Inventory Management', 'Java'),
('Cloud Monitoring Tool', 'Go'),
('Hospital Management System', 'C#'),
('Data Analysis Tool', 'Python'),
('Blog Platform', 'Ruby');

```

9. Coders working on language-specific projects

SELECT c.name AS coder_name, p.project_name FROM coder c JOIN projects p ON c.language = p.language;

Query #1 Execution time: 0.27ms

coder_name	project_name
Alice Johnson	E-Commerce Website
David Wilson	E-Commerce Website
Isabella Moore	E-Commerce Website
Brian Smith	Banking System
Frank Harris	Banking System
Charlotte Brown	Game Engine
Alice Johnson	Chatbot AI
David Wilson	Chatbot AI
Isabella Moore	Chatbot AI
Ella Thompson	Weather App
Brian Smith	Inventory Management
Frank Harris	Inventory Management
Grace Miller	Cloud Monitoring Tool

2. CONCLUSION

This experiment bridges theoretical and practical understanding of structured query execution for information retrieval. Participants learned to convert real-world needs into SQL queries, evaluate their efficiency, and explore the contrast with unstructured, NLP-driven approaches. This foundational knowledge is essential for building efficient retrieval systems across various domains.