Ira A. Fulton Schools of Engineering
Arizona State University

Term: Fall 2025 **Subject:** Computer Science & Engineering (CSE) **Number:** 412

Course Title: Database Management (CSE 412)

Assignment 04
Demonstrating the Effect of Indexes on Search Performance

Objective: To design and implement a database-backed web application that demonstrates the performance difference between search operations executed with and without indexing.

Task Description

1. Database Creation

- Use PostgreSQL through PgAdmin or psql.
- Create a new database with two relations (tables) of your choice.
  - Each table must have at least 10 attributes.
  - Define appropriate data types and primary/foreign key relationships as needed.
- Keep your SQL schema creation in a file named schema.sql.

2. Data Population (Backend Logic)

- In your Flask backend, connect to the created database and populate data in one of the following ways:
  - Option A: Import data from a CSV file.
  - Option B: Generate random data programmatically.
- Each table should have at least 10,000 tuples.
- Save your data insertion logic in data_generation.py or include it in your Flask backend.

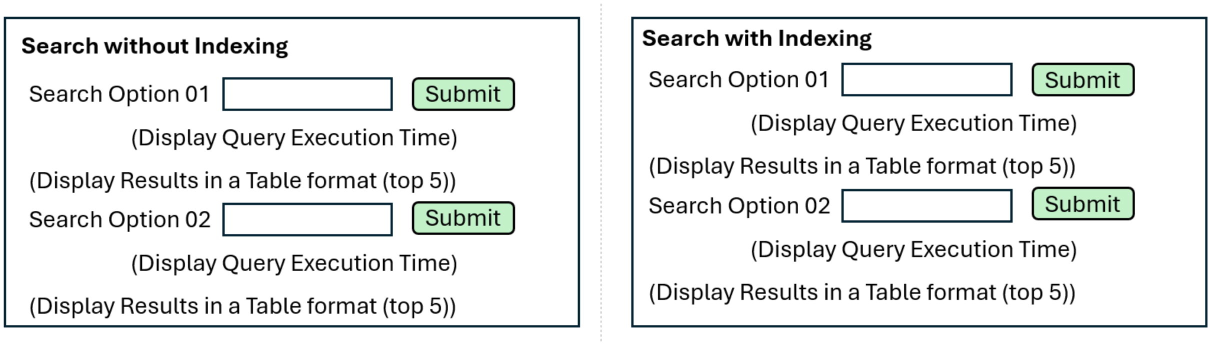
3. Flask Web Application

Develop a Flask-based web application with the following structure and functionality:

Frontend (index.html)

- The webpage should have two sections:
  - Search without Indexing
  - Search with Indexing
- Each section must include:
  - Search Option 01: Query using a single table.
  - Search Option 02: Query using both tables (a join or combined search).
- For each search:
  - Display the query execution time.
  - Display the top 5 results in a table format. (Search can be text-based or use any suitable HTML widget.)

Example (not limited to and the front end is free form)


Search without Indexing
Search Option 01
Submit
(Display Query Execution Time)
(Display Results in a Table format (top 5))
Search Option 02
Submit
(Display Query Execution Time)
(Display Results in a Table format (top 5))
Search with Indexing
Search Option 01
Submit
(Display Query Execution Time)
(Display Results in a Table format (top 5))
Search Option 02
Submit
(Display Query Execution Time)
(Display Results in a Table format (top 5))

*Search Option 01 – Results using a single table

*Search Option 02 – Results using two tables

4. Backend Logic

- Establish a connection to your PostgreSQL database using Flask (psycopg2 or equivalent).
- Implement the following:
    - Without Index: Execute the search queries before creating any indexes.
- With Index: Execute the same queries after creating suitable indexes.
- Each query should be executed twice - once in each mode.
- Measure query execution time using one of the following:
    - EXPLAIN ANALYZE or
    - Python's time.time() or any equivalent timer.

5. Deliverables

Submit the following components:

Code Files (ZIP)

- app.py - Flask backend logic
- templates/index.html - Frontend interface
- schema.sql - Database creation script
- data_generation.py - Data population script

Documentation (.md/.txt)

- README.md or .txt - setup and execution instructions.

Report (PDF)

- report.pdf containing:
    - Screenshots of the web interface for both modes with query execution times
    - A summary comparing performance with and without indexing

Submit the following three files:

1. **ZIP file** containing all source code files
2. **README.md** (text or markdown) setup & run instructions
3. **PDF report** — screenshots and observations

Note: Do not include database passwords or credentials in your submission. You may use the Week 12 demo code as starter code.