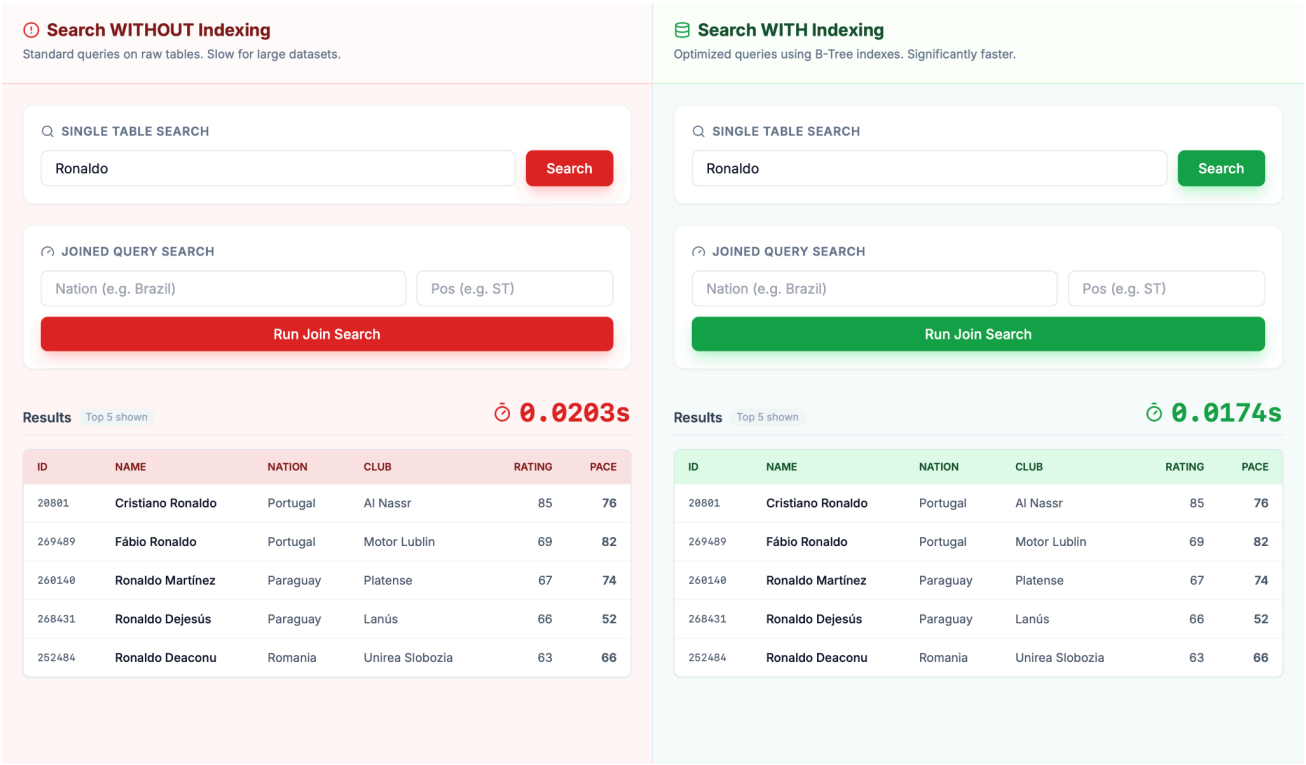
HW4 - CSE 412

Bhavya Patel - ASU ID: 1225740997

Single Table Search:



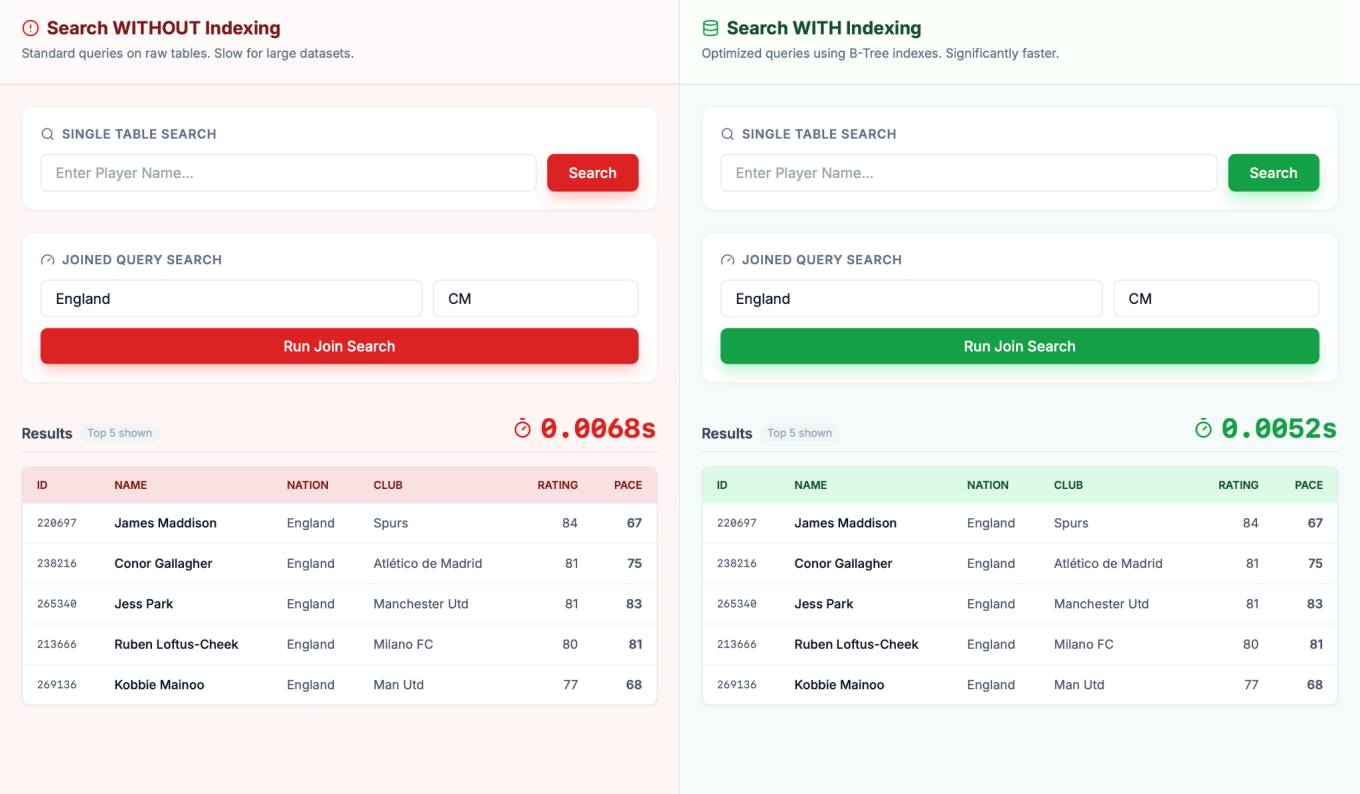
SQL Query For Search Without Indexing:

```
SET LOCAL enable_indexscan = OFF;
EXPLAIN ANALYZE SELECT * FROM players WHERE name ILIKE '%Ronaldo%';
```

SQL Query For Search With Indexing:

```
SET LOCAL enable_indexscan = ON;
EXPLAIN ANALYZE SELECT * FROM players WHERE name ILIKE '%Ronaldo%';
```

Join Table Search:



SQL Query For Search Without Indexing:

```
SET LOCAL enable_indexscan = OFF;
SET LOCAL enable_hashjoin = OFF;
SET LOCAL enable_mergejoin = OFF;
EXPLAIN ANALYZE
SELECT p.id, p.name, p.nation, p.club, ps.ovr, ps.pac FROM players p
JOIN player_stats ps ON p.player_id = ps.player_id
WHERE p.nation = 'England' AND p.position = 'CM';
```

SQL Query For Search Without Indexing:

```
SET LOCAL enable_indexscan = ON;
SET LOCAL enable_hashjoin = ON;
SET LOCAL enable_mergejoin = ON;
EXPLAIN ANALYZE
SELECT p.id, p.name, p.nation, p.club, ps.ovr, ps.pac FROM players p
JOIN player_stats ps ON p.player_id = ps.player_id
WHERE p.nation = 'England' AND p.position = 'CM';
```

Explanation:

Without Indexing:

When I search for "Ronaldo" without using an index, my database behaves like someone reading a book from cover to cover to find a specific word. It has to check every single row in the players table (all 17,000+) to see if the name matches. It requires a lot of reading and processing power because the database cannot skip any data. As the dataset grows, this approach gets significantly slower. Similarly, in the Join Search, without the index, the database had to brutally compare rows between the Players and Stats tables.

With Indexing:

When I create an index on the name column, the database builds a separate, organized structure (like a B-Tree). This is exactly like the index at the back of a textbook. It tells the database exactly which rows contain "Ronaldo". Instead of reading 17,000 rows, the database might only read a few "branches" of the index tree to find the exact location of the data. It jumps straight to the result. Similarly, in the Join Search, with the index, it instantly located the English players and their positions, snapping the data together efficiently (likely using a Hash Join).