Introduction to Indexing: Takeaways ⤴

Syntax

- Listing what SQLite is doing to return our results:

```
EXPLAIN QUERY PLAN SELECT * FROM facts;
```

- Creating an index:

```
CREATE INDEX index_name ON table_name(column_name);
```

- Creating an index if it does not exist:

```
CREATE INDEX IF NOT EXISTS area_idx ON facts(area);
```

Concepts

- Your query in SQLite is tokenized and parsed to look for any syntax errors before returning the results to you. If there are any syntax errors, the query execution halts and an error message is returned to you.

- You should minimize the amount of disk reads necessary when working with a database stored on disk.

- The query optimizer generates cost estimates for the various ways to access the underlying data, factoring in the schema of the tables and the operations the query requires. The optimizer quickly assesses the various ways to access the data and generate a best guess for the fastest query plan.

- SQLite still scans the entire table. A full table scan has time complexity   where n is the number of total rows in the table.

- Binary search of a table using the primary key would be   where   is the number of total rows in the table. Binary search on a primary key would be over a million times faster when working on a database with millions of rows compared to doing a full table scan.

- Either `SCAN` or `SEARCH` will always appear at the start of the query explanation for `SELECT` queries.

- An index table is optimized for lookups by the primary key.

Resources

- [What is an index?](#)
- [Query Plan](#)
- [Time Complexity](#)