Neha Rajamani

CS 348

Index Document

Index on **book_id** in *Transactions* table:

```
    "CREATE INDEX IF NOT EXISTS idx_transactions_book_id "
  "ON transactions(book_id)"
```

The purpose of this index is to speed up any operation (lookup or join) that needs to filter by a specific book.

Queries that it could benefit:

- **SELECT t.transaction_id, t.book_id, b.title, t.customer_id, c.first_name, c.last_name, t.date_borrowed, t.date_returned**

  FROM transactions t

  JOIN books b ON t.book_id = b.book_id

  JOIN customers c ON t.customer_id = c.customer_id

This query is used in get_transactions() to display all the transaction within the database. The index would benefit this query because it will help avoid scanning every single row within transactions when executing the JOIN on book_id clause

- **SELECT ***

  FROM transactions t

  WHERE book_id = t.book_id

This query would help when trying to display all the history for every book and since it filters by book_id, the lookup would be very fast.

Index on **customer_id** in *Transactions* table:

```
    "CREATE INDEX IF NOT EXISTS idx_transactions_customer_id "

  "ON transactions(customer_id)"
```

The purpose of this index is to speed up any operation (lookup or join) that needs to filter by a specific customer.

Queries that it could benefit:

- **SELECT t.transaction_id, t.book_id, b.title, t.customer_id, c.first_name, c.last_name, t.date_borrowed, t.date_returned**

  FROM transactions t

  JOIN books b ON t.book_id = b.book_id

  JOIN customers c ON t.customer_id = c.customer_id

This query is used in get_transactions() to display all the transaction within the database. The index would benefit this query because it will help avoid scanning every single row within transactions when executing the JOIN on customer_id clause

- **SELECT ***

  FROM transactions t

  WHERE customer_id = t.customer_id

This query would help when trying to display all the history for every customer and since it filters by customer_id, the lookup would be very fast.