Composite Index for Faster Booking Queries

Submit the GIT repo link as a submission on Openverse

1. A query frequently runs:

```
 SELECT * FROM bookings

WHERE doctor_id = ?

AND appointment_date >= ?

AND appointment_date <= ?;
```

Propose a composite index
Explain column order choice.

The Query That Runs Frequently

```
SELECT *
FROM bookings
WHERE doctor_id = ?
  AND appointment_date >= ?
  AND appointment_date <= ?;
```

---

1. What is a Composite Index?

A **composite index** is an index created on **more than one column**.

Instead of searching the whole table row by row, the database uses this index to **find data quickly**, just like using an index page in a book.

---

2. Proposed Composite Index

```
CREATE INDEX idx_bookings_doctor_date
ON bookings (doctor_id, appointment_date);
```

---

3. Where is this Used?

This is used in:

- **Hospital appointment systems**
- **Clinic booking apps**
- **Doctor scheduling software**

Whenever the system needs to:

- Show all appointments of a doctor
- Within a selected date range

---

4. Why This Column Order?

Order used:

(doctor_id, appointment_date)

Reason (very simple):

1. **doctor_id comes first**
   - Because we always search for **one specific doctor**
   - This sharply reduces the number of rows to look at
2. **appointment_date comes second**
   - Because we search for a **range of dates**
   - The database can scan dates efficiently **only after** finding the doctor

---

5. Why NOT Reverse the Order?

❌ `(appointment_date, doctor_id)` is not ideal because:

- The database would first scan **many dates**
- Then filter doctors later

- This is slower when many doctors exist

---

6. How This Index Helps

Without index:
Database checks **every booking** (slow)

With composite index:
Database:

1. Finds the **doctor**
2. Looks only at that doctor’s **appointments**
3. Filters by **date range**

Result: ⚡ **Much faster**

---

7. In Very Simple Words

- Composite index = **shortcut for database**
- Correct order = **doctor first, date next**
- Why = because we **always search by doctor**, then by date
