Proposed Composite Index

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

Why this column order?

The query is:

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

1️⃣ Equality condition comes first

- `doctor_id = ?` is an **equality filter**
- Equality conditions should be placed **first** in a composite index
- This allows the database to directly narrow down to rows for one doctor

2️⃣ Range condition comes next

- `appointment_date` is used with a **range condition** (`>=` and `<=`)
- Range filters should come **after equality filters**
- Once the doctor's rows are identified, the index efficiently scans only the required date range

---

Why not reverse the order?

- The database would first scan a **large date range**
- Then filter by `doctor_id`, which is less efficient
- This leads to more index scans and higher I/O cost