## The Proposed Composite Index

The most efficient index for this query is:

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

## Why this column order?

In a B-Tree composite index, the order of columns follows the "Equality First, Range Second" rule. Here is the breakdown of why doctor_id must come before appointment_date:

### 1. The Equality Column (doctor_id)

The query uses an exact match for the doctor (doctor_id = ?). By putting this first, the database can jump directly to the section of the index tree containing only that doctor's appointments. This narrows the search space immediately and significantly.

### 2. The Range Column (appointment_date)

The query uses a range for the date (>= and <=). Once the database has found the specific doctor_id in the index, all the dates for that doctor are stored in sorted order. The database can then simply scan the starting date and stop at the ending date.