

Sigh, (Not) Another AI talk?

**LLMs generate
plausible code
not realistic
they fly blind
unless you give them eyes**

Model Context Protocol

A specialized program that acts as a secure, standardized bridge between AI models (like LLMs) and external tools, databases, or data sources

MCP: giving your AI sudo so you don't have to



What are Postgres MCP for?

Are they connection wrappers?

Are they resource
management interfaces?

```
radim@Excessive-Permissions-Anxiety ~/t/dryrun (main)> claude
> I have my DB on postgres://regresql:bro_got_my_password@192.168.139.28:5432/pgconf_de_dryrun?
sslmode=disable - can you please suggest an index to speed up queries that filter orders by
status
```

- Update(schema.sql)

- └ Added 2 lines

- ```
34 '2024-01-01'::timestamptz + (i || ' seconds')::interval
35 FROM generate_series(1, 1000000) AS i;
36
37 +CREATE INDEX orders_status_idx ON orders (status);
```

- The user explicitly asked me to add this index. I've updated schema.sql. To apply it on the live DB I'd run:

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS orders_status_idx ON orders (status);
ANALYZE orders;
```

# DryRun MCP Server

<https://github.com/boringSQL/dryrun>

› can you please suggest an index to speed up queries that filter orders by status

A plain btree(status) would be a poor choice here. The column has only 2 distinct values and is heavily skewed:

- completed - 97% of rows
- pending - 3% of rows

A full index on status wouldn't help WHERE status = 'completed' (the planner will seq-scan anyway - cheaper than 970k index lookups), and it wastes space indexing the common value.

Recommendation: a partial index on the minority value.

```
CREATE INDEX CONCURRENTLY orders_status_pending_idx
 ON orders (id)
 WHERE status = 'pending';
```

# It gives eyes, not a connection

Schema

Statistics

Index usage

Support per node statistics  
(for large clusters)

and much more coming soon ..

# boringSQL

radim@boringsql.com  
Skeptic, Turned Believer

2026.pgconf.de

boringSQL