

# My Journey into PostgreSQL Development

Nazır Bilal Yavuz · Microsoft

PGConf.dev 2026 · Vancouver

*byavuz81@gmail.com*

## Nazir Bilal Yavuz

Software Engineer @ Microsoft

**~7 years**

software engineering experience

**~4.5 years**

joined Open Source Postgres Contributors team

**Zero database experience**

prior database experience when I started

# Agenda

*Each topic follows the same structure:*

## My experience

*What actually happened*

## What helped

*Things that made a difference*

## Looking back

*What I would do differently*



01

**Finding items to work on**



02

**Mailing lists**



03

**Working on patches**



04

**Review & Feedback**

# #1 Finding Items to Work On

## My Experience

### My team

Chance to ask if they have something for me

### Conference

Future work, hallway talks

### Diverse subsystems

Many different areas that you could be interested in

## What Helped

### Reading mailing list daily

Discussions have ideas

### Reviews → code understanding

Finding problems, optimizations, ideas...

## Looking Back

### Maintain a backlog

Postgres doesn't have an issue tracker

# #1 Finding Items to Work On

## My Experience

### My team

Chance to ask if they have something for me

### Conference

Future work, hallway talks

### Diverse subsystems

Many different areas that you could be interested in

## What Helped

### Reading mailing list daily

Discussions have ideas

### Reviews → code understanding

Finding problems, optimizations, ideas...

## Looking Back

### Maintain a backlog

Postgres doesn't have an issue tracker

# #1 Finding Items to Work On

## My Experience

### My team

Chance to ask if they have something for me

### Conference

Future work, hallway talks

### Diverse subsystems

Many different areas that you could be interested in

## What Helped

### Reading mailing list daily

Discussions have ideas

### Reviews → code understanding

Finding problems, optimizations, ideas...

## Looking Back

### Maintain a backlog

Postgres doesn't have an issue tracker

## #2 Mailing Lists

### My Experience

#### Easy to follow

Send & Read

#### Archives are gold

Search before posting

#### Anxiety

Public + non-native English speaker  
=  
over-editing every email

### What Helped

#### Conferences

Meet with people

#### Read how others write

- \* Understand the culture
- \* Everyone makes mistakes

#### Discord channel

More casual conversation

### Looking Back

#### Timebox your email

Perfect is the enemy of posted

## #2 Mailing Lists

### My Experience

#### Easy to follow

Send & Read

#### Archives are gold

Search before posting

#### Anxiety

Public + non-native English speaker  
=  
over-editing every email

### What Helped

#### Conferences

Meet with people

#### Read how others write

- \* Understand the culture
- \* Everyone makes mistakes

#### Discord channel

More casual conversation

### Looking Back

#### Timebox your email

Perfect is the enemy of posted

## #2 Mailing Lists

### My Experience

#### Easy to follow

Send & Read

#### Archives are gold

Search before posting

#### Anxiety

Public + non-native English speaker  
=  
over-editing every email

### What Helped

#### Conferences

Meet with people

#### Read how others write

- \* Understand the culture
- \* Everyone makes mistakes

#### Discord channel

More casual conversation

### Looking Back

#### Timebox your email

Perfect is the enemy of posted

# #3 Working on Patches

## My Experience

### Standards are very high

Nice for improving but could be discouraging

### My team

Get unblocked & get pre-reviews

### No reply

Silence doesn't mean rejection

## What Helped

### Doing reviews

Understanding other patches made writing my own much easier

### My team & Mentoring program

### No bad feedback exists

Shared goal: Better Postgres

## Looking Back

### Ask before coding

A quick 'does this make sense?' email saves weeks of wasted work

### WIP patches are fine

Feedback shapes the right direction

# #3 Working on Patches

## My Experience

### Standards are very high

Nice for improving but could be discouraging

### My team

Get unblocked & get pre-reviews

### No reply

Silence doesn't mean rejection

## What Helped

### Doing reviews

Understanding other patches made writing my own much easier

### My team & Mentoring program

### No bad feedback exists

Shared goal: Better Postgres

## Looking Back

### Ask before coding

A quick 'does this make sense?' email saves weeks of wasted work

### WIP patches are fine

Feedback shapes the right direction

# #3 Working on Patches

## My Experience

### Standards are very high

Nice for improving but could be discouraging

### My team

Get unblocked & get pre-reviews

### No reply

Silence doesn't mean rejection

## What Helped

### Doing reviews

Understanding other patches made writing my own much easier

### My team & Mentoring program

### No bad feedback exists

Shared goal: Better Postgres

## Looking Back

### Ask before coding

A quick 'does this make sense?' email saves weeks of wasted work

### WIP patches are fine

Feedback shapes the right direction

# #4 Review & Feedback

## My Experience

### Highly experienced community

Hard to object

### Very welcoming culture

Feeling part of the team

## What Helped

### Doing reviews myself

The fastest way to learn both the code and the culture

### Reading others' reviews

Watching experienced people review taught me what to look for

## Looking Back

### Review aim & method first

Before diving into code, ask: what is this patch trying to do?

### Say LGTM or +1

People tend to say something when something is wrong

Patch review workshops

### CommitFest application

Status -> 'Needs review'

Tags -> 'Good First Review'

# #4 Review & Feedback

## My Experience

**Highly experienced community**

Hard to object

**Very welcoming culture**

Feeling part of the team

## What Helped

**Doing reviews myself**

The fastest way to learn both the code and the culture

**Reading others' reviews**

Learn what reviewers are looking for

## Looking Back

**Review aim & method first**

Before diving into code, ask: what is this patch trying to do?

**Say LGTM or +1**

People tend to say something when something is wrong

**Patch review workshops**

**CommitFest application**

Status -> 'Needs review'

Tags -> 'Good First Review'

# #4 Review & Feedback

## My Experience

### Highly experienced community

Hard to object

### Very welcoming culture

Feeling part of the team

## What Helped

### Doing reviews myself

The fastest way to learn both the code and the culture

### Reading others' reviews

Watching experienced people review taught me what to look for

## Looking Back

### Review aim & method first

Before diving into code, ask: what is this patch trying to do?

### Say LGTM or +1

People tend to say something when something is wrong

Patch review workshops

### CommitFest application

Status -> 'Needs review'

Tags -> 'Good First Review'

# Conclusion



## Finding items to work on

Read discussions daily · Conferences · Keep a backlog



## Mailing lists

Straightforward · Anxiety -> Timebox your emails · Shared goal: better Postgres



## Working on patches

Standards are high · Ask before coding · WIP patches are welcome



## Reviews

Best way to learn Postgres · Patch review workshops · CommitFest

# Thank You!

## Questions?

---

Nazır Bilal Yavuz · Microsoft

PGConf.dev 2026 · Vancouver

*byavuz81@gmail.com*