AI-Assisted PostgreSQL Engineering for NASA Near-Earth Object Surveyor Mission

neos.epss.ucla.edu

Tatiana Mosheva / IPAC/Caltech

Practical experience with AI-assisted PostgreSQL and Python engineering for the NEO Surveyor ground system.

AI proposals depend on the context provided and require independent verification.

Engineering context

Projected daily growth

~1M new rows / 12–16 new partitions

100+ database functions
Python applications and scripts

AI assists: code review, library migrations, SQL design and API refactoring.

Engineer-Controlled Workflow



1

1 Context

The engineer defines the task and selects schema, code and constraints.
What context does the AI have?



3

3 Schema and query design

Schema and semantics guide the answer.
Valid SQL can still be the wrong answer.

EXAMPLE: MISSING SCHEMA CONTEXT

AI PROPOSAL	ACTUAL SCHEMA
`objectid`	`"ObjectID"`

The prompt omitted exact spelling and quoting.
AI assumed lowercase identifiers.
No DDL or database access? Unverified.



4

4 Assumptions and unknowns

What assumptions did the AI make?
Which rules or context are missing?
The engineer makes both explicit.



5

5 Verify

Sandbox, tests, query plans and results.
AI found problems in application code.
Its own proposals also required verification.



2

2 AI proposal

Code review and migrations.
SQL, patches, assumptions.

Revise proposal



6

6 Engineer

Accept, revise or reject.
Tests provide evidence.
The engineer decides what to implement.

ACCEPT



AI proposes. Evidence decides.
If we can't verify it, we shouldn't use it.

tmosheva@ipac.caltech.edu