Contributing (code) to postgres

29.06.2026

SENSATIONAL

About me

Alberto Piai

Sensational AG, Zürich

apiai@sensational.ch

alberto.piai@gmail.com

SENSATIONAL

The problem I'm trying to solve

```
create table t (a int);

alter table t
add column b int
generated always as (a * 2) stored;
```

SENSATIONAL

My proposal

SENSATIONAL

My proposal

```
alter table t add column b int;
```

SENSATIONAL

My proposal

```
alter table t add column b int;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check clause
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check clause
  check ( b is not distinct from (a * 2) ) not valid;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;

-- new command: switch without rewrite
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;

-- new command: switch without rewrite
alter table t alter b
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;

-- new command: switch without rewrite
alter table t alter b
    add generated always stored
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;

-- new command: switch without rewrite
alter table t alter b
    add generated always stored
    using constraint check_clause;
```

SENSATIONAL

My proposal

```
alter table t add column b int;

-- backfill at the appropriate pace
update t set b = (a * 2) where b is null;

-- prove that the data satisfies the gen expression
alter table t add constraint check_clause
    check ( b is not distinct from (a * 2) ) not valid;
alter table t validate constraint check_clause;

-- new command: switch without rewrite
alter table t alter b
    add generated always stored
    using constraint check_clause;
```

SENSATIONAL

Timeline

Timeline

- v1 posted to pgsql-hackers: 2026-03-17

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07
- v3,v4 minor fixes and rebases in the meantime

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07
- v3,v4 minor fixes and rebases in the meantime
- first discussions in person: pgconf.dev in May

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07
- v3,v4 minor fixes and rebases in the meantime
- first discussions in person: pgconf.dev in May
- first review on pgsql-hackers: 2026-05-26

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07
- v3,v4 minor fixes and rebases in the meantime
- first discussions in person: pgconf.dev in May
- first review on pgsql-hackers: 2026-05-26
- new proposal, improved design: 2026-05-27

Timeline

- v1 posted to pgsql-hackers: 2026-03-17
- v2 posted to pgsql-hackers: 2026-04-07
- v3,v4 minor fixes and rebases in the meantime
- first discussions in person: pgconf.dev in May
- first review on pgsql-hackers: 2026-05-26
- new proposal, improved design: 2026-05-27
- v5 hopefully this weekend :)