Database Incident Root Cause Analysis (RCA) & Remediation Guide

Finding H13 | Severity P1 | Owner Developer (Application SQL / ORM / Migrations) + DBA / Platform Engineer (Cluster GUCs / Vacuum / Pooling / Storage) | Environment cloudrun-showcase (synthetic-demo) | Generated 2026-10-27 17:22:35 UTC

1. Incident Overview & Executive Summary

1.1 Executive Summary

CircuitSQL detected finding H13 (32-bit integer primary key and sequence exhaustion) in environment cloudrun-showcase (synthetic-demo) at 2026-10-27 17:22:35 UTC. The overflow aborts every order insert (SQLSTATE 22003). Migrate the column and sequence to bigint to restore inserts. The correlated report classifies the incident as P1 with triggers H2, H6, H11, H12, pg_stat_statements. CircuitSQL routes remediation to Developer (Application SQL / ORM / Migrations) + DBA / Platform Engineer (Cluster GUCs / Vacuum / Pooling / Storage). Environment cloudrun-showcase (synthetic-demo) reported 4 breached health check(s) across 2 correlated root-cause incident(s) and 2 query offender(s). CircuitSQL routes remediation to Developer (Application SQL / ORM / Migrations) + DBA / Platform Engineer (Cluster GUCs / Vacuum / Pooling / Storage).

1.2 Document Metadata

Field	Value
Incident ID	CSQL-INC-H13-202610271722
Date / Time of Incident	2026-10-27 17:22:35 UTC
Duration	63 min (trigger -25 min to recovery +38 min; see section 4)
Severity Level	P1 - Critical (customer-visible outage or data-risk)
Affected Systems	circuitsql_monitor@db-primary.internal:5432/circuitsql (ssl_mode=require); orders and customers OLTP paths; environment cloudrun-showcase (synthetic-demo)
Database Engine & Version	PostgreSQL 17.2
Lead Investigator(s)	Developer / Marlon Francis Briguera (incident commander); Backend Engineer (query path); DBA / Platform Engineer (cluster controls)
Current Status	Mitigated - monitoring for recurrence; permanent remediation tracked in section 7

2. Technical Environment Context

2.1 System Topography

```
Client / API (FastAPI, app_rw)
        |
        v
[PgBouncer :6432] -- pool_mode=transaction, default_pool_size=4x vCPU
        |
        v
[PostgreSQL primary circuitsql]  << finding H13
        |-- pg_stat_activity / pg_locks / pg_blocking_pids
        |-- pg_stat_statements (offender ranking)
        |-- pg_stat_checkpointer / pg_stat_io (PG 17/18 subsystem counters)
        '-- [Read replica] analytics_ro (reporting, EXPLAIN ANALYZE rehearsals)

[CircuitSQL Monitor] -- H1-H18 probes, CSQL-S01..S07 security scan, RCA export
```

2.2 Privilege & Tool Prerequisites

- Role membership: pg_monitor and pg_read_all_stats (read-only catalog access).
- pg_stat_statements present in shared_preload_libraries and installed in the database.
- PgBouncer admin console reachable on port 6432 for SHOW POOLS and RELOAD.
- psql 17 client plus a shadow database for EXPLAIN (ANALYZE, BUFFERS) rehearsals; never run write DDL against production to inspect a plan.
- ALTER SYSTEM requires a superuser or a role with the ALTER SYSTEM privilege; credentials come from the secret manager, never URL query strings.
- Incident channel and GitLab issue write access for the fix MR linked in section 8.

3. Symptoms & Observed Evidence

3.1 Observed vs. Expected Behavior

Aspect	Observed	Expected
Measured value vs. threshold	H13 evaluated as observed.	Every sequence-backed primary key is bigint, and consumption stays below 50% of the type maximum.
Session and lock state	The finding evaluation detected no unexpected lock queue.	Lock depth <= 1, no idle-in-transaction session, and every statement bounded by lock_timeout and statement_timeout.
Workload impact	orders and customers traffic shows elevated latency or saturation while the condition holds.	p95 within the 250 ms OLTP SLO and connection use below 85% of the pool budget.
Data integrity and compliance	No data loss observed; the control gap maps to SQLSTATE 22003 handling.	Constraints validated, checksums clean, and CIS PostgreSQL 17 controls satisfied.

3.2 System Logs & Error Codes

Application Exception Log

```
2026-10-27 17:10:35.812 UTC [api] ERROR orders.views: request failed route=/orders finding=H13
psycopg.errors.NumericValueOutOfRange: integer out of range
Traceback (most recent call last):
  File "app/routes/orders.py", line 88, in create_order
    session.commit()
psycopg.errors.DatabaseError: ERROR: nextval: reached maximum value of sequence "orders_id_seq" (2147483647) (
SQLSTATE 22003)
```

RDBMS Error Log

```
2026-10-27 17:10:35.101 UTC [41892] LOG: process 41892 still waiting after 1000.042 ms
ERROR: nextval: reached maximum value of sequence "orders_id_seq" (2147483647) (SQLSTATE 22003)
2026-10-27 17:10:36.204 UTC [41892] STATEMENT: /* finding=H13 */ SELECT schemaname, sequencename, data_type::t
ext, last_value, max_value FROM pg_sequences WHERE NOT cycle ORDER BY la...
2026-10-27 17:10:37.318 UTC [41892] DETAIL: causal_tag=[CSQL-M05 | DIRECT] sqlstate=22003
```

4. Timeline of Events

Phase	UTC Timestamp	Event	Evidence
Trigger	2026-10-27 16:57:35 UTC	H13 condition established: 32-bit integer primary key and sequence exhaustion	finding=H13; tag=[CSQL-M05 | DIRECT]
Alerting	2026-10-27 17:04:35 UTC	CircuitSQL monitor evaluated the probe and raised the finding on the console.	environment=cloudrun-showcase (synthetic-demo); breached=true
Degradation	2026-10-27 17:10:35 UTC	orders and customers traffic degraded: latency rose and lock or buffer pressure became visible.	The snapshot captured cloudrun-showcase (synthetic-demo) at 2026-10-27 17:22:35 UTC without a live catalogue row for H13.

Phase	UTC Timestamp	Event	Evidence
Triage	2026-10-27 17:22:35 UTC	On-call opened the per-finding RCA page and confirmed the mechanism against pg_stat_activity, pg_locks, and pg_stat_statements.	RCA page /console/rca/H13
Mitigation	2026-10-27 17:31:35 UTC	On-call terminated blockers and reloaded the connection pool as emergency mitigation (section 6.1).	pg_terminate_backend + PgBouncer RELOAD
Recovery	2026-10-27 18:00:35 UTC	Latency, lock depth, and connection use returned to baseline. On-call monitored the incident for 30 minutes before closing it.	recovery verified at 18:00:35 UTC

5. Technical Root Cause Analysis

Causal Flow Diagram

```
+-----------------------------------------------------------------------------------------------+
| Trigger H13: 32-bit integer primary key and sequence exhaustion                               |
+-----------------------------------------------------------------------------------------------+
                                                  |
                                                  v
+-----------------------------------------------------------------------------------------------+
| Mechanism: The orders primary key still uses a 32-bit integer sequence. After consuming max... |
+-----------------------------------------------------------------------------------------------+
                                                  |
                                                  v
+-----------------------------------------------------------------------------------------------+
| Symptom: The overflow aborts every order insert (SQLSTATE 22003). Migrate the column and se... |
+-----------------------------------------------------------------------------------------------+
                                                  |
                                                  v
+-----------------------------------------------------------------------------------------------+
| Impact: orders/customers OLTP latency and write availability                                  |
+-----------------------------------------------------------------------------------------------+
                                                  v
+-----------------------------------------------------------------------------------------------+
| Blast radius: checkout writes, orders reads, p95 SLO                                          |
+-----------------------------------------------------------------------------------------------+
```

Primary Cause

The orders primary key still uses a 32-bit integer sequence. After consuming max_value, the next nextval call overflows. Correlated verdict: Root Cause (H2 | High): PID 41892 idle in transaction for 312.0s holding RowExclusiveLock on public.orders [CSQL-M05 | DIRECT] -> Proximate Cascading Symptoms (2 suppressed): H11 (Lock blocking chain depth 4 queued behind idle-in-transaction PID 41892 on public.orders [CSQL-M03 | INDIRECT]), H12 (public.orders dead-tuple ratio 34.0% (340,000 dead tuples) amplified by H6 autovacuum stall [CSQL-M01 | INDIRECT]).

Secondary Cause

The original migration chose serial/integer and no capacity review ran before order volume crossed the 2^31 ceiling. Downstream correlation: Root Cause (H6 | Medium): Autovacuum disabled on public.orders while hot updates accumulate dead tuples [CSQL-M01 | DIRECT] (isolated breach, 0 downstream symptoms).

Tertiary Cause

The overflow aborts every order insert (SQLSTATE 22003). Migrate the column and sequence to bigint to restore inserts. Correlated evidence: Check H2 (High, owner DEVELOPER): value=312.0 vs. threshold=60.0 (cited) - PID 41892 idle in transaction for 312.0s holding RowExclusiveLock on public.orders [CSQL-M05 | DIRECT]

Finding SQL

```
SELECT schemaname, sequencename, data_type::text,
       last_value, max_value
  FROM pg_sequences
 WHERE NOT cycle
 ORDER BY last_value::numeric / max_value DESC;
```

Metrics

- PG 17/18 pg_stat_checkpointer: requested ratio 11.2%, write 1840.5 ms, sync 210.2 ms.
- Waiters: LWLock buffer 2, IO 1.
- The snapshot has no catalogue evidence for H13. The exported document records the deterministic probe and the expected threshold.

Causal Tag

[CSQL-M05 | DIRECT]

6. Step-by-Step Resolution & Remediation Procedures

6.1 Emergency Mitigation (Immediate Stabilization)

Step 1 - Capture the blocker set before touching anything

```
SELECT pid, usename, state,
       now() - xact_start AS xact_age,
       wait_event_type, wait_event,
       pg_blocking_pids(pid) AS blockers,
       left(query, 160) AS query
  FROM pg_stat_activity
 WHERE state <> 'idle'
   AND (now() - xact_start > interval '30 seconds'
        OR wait_event_type = 'Lock')
 ORDER BY xact_age DESC;
```

Record the head blocker for H13 in the incident timeline before any termination. The capture is read-only and pg_monitor compatible.

Step 2 - Terminate only the head blocking backend, with a bounded wait

```
-- Replace 41892 with the head blocker captured in Step 1.
SELECT pg_terminate_backend(41892, 5000);

-- Confirm the queue drains instead of re-queuing.
SELECT count(*) AS queue_depth
  FROM pg_stat_activity
 WHERE cardinality(pg_blocking_pids(pid)) > 0;
```

Do not terminate blocked sessions first. They re-queue behind the same head blocker and the incident restarts.

Step 3 - Drain and reload PgBouncer so pooled sessions pick up the new limits

```
-- PgBouncer admin console (port 6432), not the PostgreSQL port.
psql -h 127.0.0.1 -p 6432 -U pgbouncer pgbouncer -c 'SHOW POOLS;'
psql -h 127.0.0.1 -p 6432 -U pgbouncer pgbouncer -c 'RELOAD;'
psql -h 127.0.0.1 -p 6432 -U pgbouncer pgbouncer -c 'SHOW STATS;'
```

RELOAD applies pgbouncer.ini pool_mode and default_pool_size without dropping the listener. Transaction pooling discards session-level state by design.

6.2 Permanent Remediation Steps

6.2.1 CREATE INDEX CONCURRENTLY & EXPLAIN (ANALYZE, BUFFERS)

```
-- 6.2.1 Build the replacement index without blocking writes.
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_orders_customer_created
    ON public.orders (customer_id, created_at DESC);

-- Prove the planner uses it before dropping or shipping anything else.
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, status, total_cents
  FROM public.orders
 WHERE customer_id = $1
 ORDER BY created_at DESC
 LIMIT 50;

-- Finding-specific DDL for H13.
ALTER TABLE public.orders ALTER COLUMN id TYPE bigint;
ALTER SEQUENCE public.orders_id_seq AS bigint;
```

Expand-and-contract: add a bigint identity column, backfill, then swap.

6.2.2 ALTER SYSTEM SET statement_timeout = '5000ms' & idle_in_transaction_session_timeout = '15000ms'

```
-- 6.2.2 Bound every session at the cluster and role level.
ALTER SYSTEM SET statement_timeout = '5000ms';
ALTER SYSTEM SET idle_in_transaction_session_timeout = '15000ms';
SELECT pg_reload_conf();

-- Verify the effective values for the application role.
SELECT rolname, rolconfig
  FROM pg_roles
 WHERE rolname IN ('app_rw', 'migration_runner', 'analytics_ro');
```

Per-role settings override cluster defaults. For H13, the idle-in-transaction ceiling is the control that prevents recurrence.

6.2.3 Before/After @Transactional application refactor

```
# Before: network I/O and retries live inside the transaction.
@transactional
def create_order(payload: OrderIn) -> Order:
    order = Order(**payload.model_dump())
    session.add(order)
    session.flush()
    inventory.reserve(order.id)      # remote call holds the write lock
    payments.authorize(order)        # another remote call
    return order

# After: the transaction only covers database work.
@transactional
def create_order(payload: OrderIn) -> Order:
    order = Order(**payload.model_dump())
    session.add(order)
    session.flush()
    outbox.enqueue("order.created", order.id)  # local insert, no I/O
    return order

# Inventory and payment run after COMMIT, with their own short transactions.
```

Keep only database work inside the transaction boundary. Run remote calls and retries after COMMIT.

7. Preventative Action Items & Monitoring Controls

Action Item	Type	Target Date	Owner	Status
Add a burn-rate alert for H13 with a 5-minute lookback and route it to the Monitor stage dashboard.	Monitoring control	2026-10-30	DBA / Platform Engineer	Open

Action Item	Type	Target Date	Owner	Status
Enforce statement_timeout, lock_timeout, and idle_in_transaction_session_timeout for every application role.	Configuration	2026-10-30	DBA / Platform Engineer	Open
Ship the reviewed remediation for H13 behind Tier 1 static analysis and Tier 2 shadow verification.	Code change	2026-11-10	Backend Engineer	In progress
Add a regression test that fails when the finding reproduces against the showcase snapshot.	Test coverage	2026-11-06	Backend Engineer	Open
Re-audit the control against CIS PostgreSQL 17 Benchmark v1.0.0 and record the evidence in the Govern stage.	Compliance	2027-01-25	Software Architect / Technical Writer	Scheduled

8. Related Resources & References

- Internal Monitoring Dashboard: /console?stage=monitor&view=rca&finding=H13
- Related Merge Request: !0 - fix MR to be opened from section 6.2 (H13)
- Runbook Reference: src/circuitsql/skills/root-cause-analysis/SKILL.md section 4-6; docs/specs/v3-spec.md (finding H13)
- Vendor Documentation: https://www.postgresql.org/docs/17/datatype-numeric.html