

WEBINAR SERIES · DATABASE RELIABILITY

When Replication Breaks



Quest

Your Recovery Playbook

SharePlex Advanced Operations Part 2 - Skills 101
March 2026

YOUR PRESENTER

Bilel SFAXI



Entrepreneur, engineer, passionate, tech enthusiast, with more than 20 years of experience working in IT for various international software vendors, and in daily contact with new technologies (such as AI recently).

A bit of a geek at heart and fascinated by Japanese culture!

Bilel SFAXI

Sr. Presales Engineer

bilel.sfaxi@quest.com

<https://www.linkedin.com/in/bilelsfaxi/>

What we'll cover

01



The Stakes – Why Replication Failures Hurt

5 min

02



Anatomy of a Failure – Root Causes & Signals

5 min

03



The Recovery Playbook – Step-by-Step

5 min

04



Live Demo – Diagnosing & Recovering in Shareplex

12 min

05



Prevention & Best Practices

3 min

When replication fails, business stops.

\$9,000

average cost per minute
of unplanned downtime

68%

of DBAs report replication
incidents monthly

4+ hrs

average time to recover
without a playbook



Replication gaps, out-of-sync data, and silent failures can corrupt downstream systems before anyone notices.

References:

- Atlassian's summary of Ponemon & Gartner: <https://www.atlassian.com/incident-management/kpis/cost-of-downtime>
- A Quest/DBTA survey: <https://www.quest.com/dbta-survey-replication-solves-database-downtime>
- The DORA State of DevOps reports: <https://dora.dev/research/>

Know the enemy before you fight it

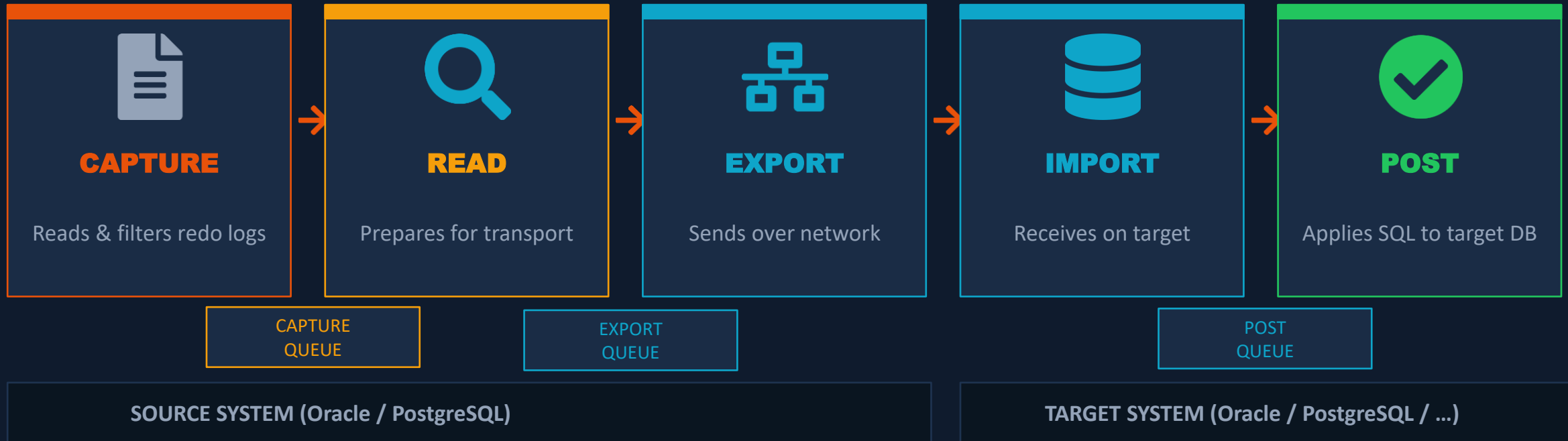
ROOT CAUSES

- Redo log switch / archive gap
- DDL changes not propagated
- Network interruption mid-transaction
- Target database constraint violations
- Shareplex process crash (Capture, Read, Export)
- Disk/space exhaustion in queue
- Permission or auth changes on target

WARNING SIGNALS

- Latency spike in sp_ctrl status
- POST queue depth growing
- Error messages in event.log
- Compare reports showing divergence
- Capture process in 'stopped' state
- Replication lag > SLA threshold
- Target row count mismatch

Understanding the pipeline you're recovering

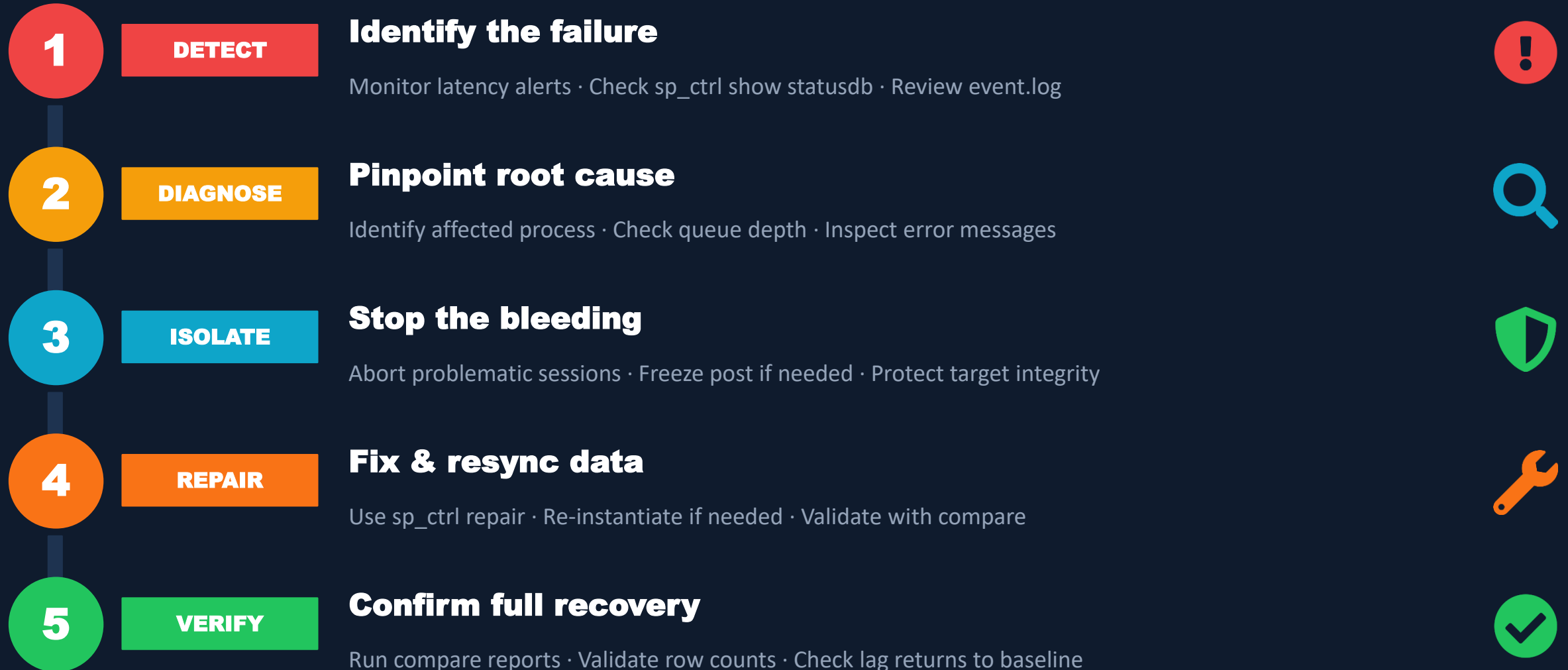


sp_ctrl — The Command Center

show statusdb · show log · compare · repair · start / stop / abort

Every recovery action flows through sp_ctrl — master it and you control recovery speed.

A systematic approach, every time





LIVE DEMO

Diagnosing & Recovering in Shareplex

1

Simulating a replication failure

2

Reading signals in sp_ctrl

3

Executing repair commands

4

Validating full recovery

~12 minutes

The Environment



Scenario: Oracle-to-Oracle replication across two hosts. A network interruption causes the Export process to halt mid-stream, creating a queue backlog. The target falls behind and begins diverging.

Step 1	Observe the failure	<code>sp_ctrl> show statusdb</code>	<i>Note STOPPED status on Export</i>
Step 2	Read the logs	<code>sp_ctrl> show log 20</code>	<i>Identify network error in event.log</i>
Step 3	Check queue depth	<code>sp_ctrl> show post detail</code>	<i>Confirm backlog accumulating</i>
Step 4	Restart the process	<code>sp_ctrl> start post on host2</code>	<i>Resume replication flow</i>
Step 5	Measure lag	<code>sp_ctrl> show statusdb detail</code>	<i>Watch latency return to baseline</i>
Step 6	Validate data integrity	<code>sp_ctrl> compare splex.demo_src</code>	<i>Confirm zero row differences</i>

Reading the signals in sp_ctrl

```
sp_ctrl - show statusdb

sp_ctrl> show statusdb

Process      Status      PID      Running Since
Capture      Running     12345    2026-03-16 06:00
Read         Running     12346    2026-03-16 06:00
Export       Running     12347    2026-03-16 06:00
Import       Running     12350    2026-03-16 06:00
Post         DELAYED     12351    2026-03-16 06:00

sp_ctrl> show post detail

Queue depth: 245,830 messages ← CRITICAL
Oldest message: 2026-03-16 08:43:02
Lag: 00:03:47 (and growing)
```

Key Observations



Post WAITING

No data pushed to target



Lag Growing

3m 47s and increasing



Queue Backlog

245K messages pending



Post DELAYED

Target falling behind

Executing the repair

sp_ctrl – recovery sequence

Step 1: Restart Post process

```
sp_ctrl> start post on host2
```

```
Post started successfully.
```

Step 2: Monitor queue drain

```
sp_ctrl> qstatus
```

```
Queue depth: 89,240 messages (draining...)
```

Step 3: Check lag reduction

```
sp_ctrl> show post detail
```

```
Lag: 00:01:12 ↓ (improving)
```

Step 4: Validate data integrity

```
sp_ctrl> compare splex.demo_src
```

```
Differences: 0 ✓ All rows match
```

Recovery Timeline



Don't wait for it to break

MONITOR

- Set up `sp_ctrl compare` on a schedule
- Alert on queue depth thresholds
- Monitor latency SLA breaches
- Track log switch frequency
- Dashboard: latency + error rate




PREPARE


- Maintain a runbook per failure type
- Test recovery drills quarterly
- Document your topology fully
- Assign DBA on-call rotation
- Pre-stage repair scripts

HARDEN

- Enable Shareplex HA / failover
- Archive logs retention policy
- Network redundancy (dedicated replication NIC)
- Regular compare & repair cycles
- Version-controlled config backups


Your 3-minute summary

1  Replication failures are inevitable — your response time determines business impact.

2  Most failures follow predictable patterns: log gaps, process crashes, network issues, schema drift.

3  `sp_ctrl` is your single source of truth — master 6 commands and you can handle 80% of scenarios.

4  Follow the 5-step playbook: Detect → Diagnose → Isolate → Repair → Verify. Never skip steps.

5  Prevention costs 10× less than recovery. Monitor, document, and drill before failures happen.

Q & A

Questions & Discussion

Shareplex Documentation

quest.com/products/shareplex

SharePlex Support

support.quest.com/shareplex

Community & Support Forum

community.quest.com

Thank you for attending!

Don't Miss What's Next!

Introducing SharePlex Insights:

AI-Driven Visibility into Your SharePlex Environment

May 28th @ 11:00a.m. ET



 Quest

Thank You!