

PostgreSQL Daten retten

LinuxDay 2026, HTL Dornbirn

Oli Sennhauser

CTO, FromDual GmbH

<https://www.fromdual.com/presentations>

Über FromDual GmbH

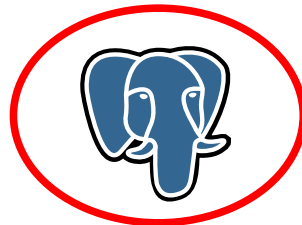


www.fromdual.com

Support



remote-DBA



On-site Beratung



Schulungen



PostgreSQL Daten retten

- Backup / Restore – logisch
- Backup / Restore – physisch
- Point-in-Time-Recovery
- Verzögerte Replikation
- Retten bei korrupten Datenfiles
- Backup testen!!!

Mein Hintergrund



www.fromdual.com



Quelle: firefly.adobe.com / Gemini 3.1 (Nano Banana 2)

Backup – logisch I

- Was passiert?

```
foreach ( schema )  
  foreach ( table )  
    DDL to STDOUT  
    SELECT * FROM table TO STDOUT
```

- PostgreSQL Struktur:

- 1 PostgreSQL Server (postgres Prozess/Cluster)
 - n Datenbanken (database)
 - m Schemata (schema)

Backup – logisch II

- **Wie macht man das?**

EINE Datenbank
KEINE Rollen (≅ User)!

```
$ pg_dump --dbname=<db> > db_dump.sql  
  
$ pg_dump --user=... --host=-... --port=... <db> > db_dump.sql  
  
$ pg_dump <db> --schema=<schema> --table=<test*> > db_schema_table_dump.sql  
  
$ pg_dump <db> | gzip > db_dump.sql.gz  
  
$ pg_dump <db> --inserts --rows-per-insert=1 > db_dump.sql
```

ALLE Datenbanken
Daten + Rollen (≅ User) und Tablespaces

```
$ pg_dumpall --clean --statistics > full_dump.sql  
  
$ pg_dumpall --roles-only > roles_dump.sql
```

Backup – logisch III

- Was erhält man?

```
-- PostgreSQL database dump

-- Dumped from database version 18.4
-- Dumped by pg_dump version 18.4

SET ...

-- Name: company; Type: TABLE; Schema: public; Owner: sitemonitor

CREATE TABLE public.company (
...

ALTER TABLE public.company OWNER TO sitemonitor;

CREATE SEQUENCE public.company_id_seq
...

-- Data for Name: domain; Type: TABLE DATA; Schema: public

COPY public.domain (id, name, tld, created_at) FROM stdin;
94      geberit.ch      ch      2026-09-22 14:39:16.591426
95      givaudan.com      com      2026-09-22 14:39:16.673012
97      logitech.com      com      2026-09-22 14:39:16.871083
...
```

Restore – logisch

- **Wie macht man das?**

EINE Datenbank

User/Rollen selber anlegen!

```
$ createdb --template=template0 <db>
```

```
$ psql --no-psqlrc <db> < db_dump.sql
```

```
$ psql --no-psqlrc --set ON_ERROR_STOP=on <db> < db_dump.sql
```

```
$ psql --no-psqlrc --single-transaction <db> --quiet < db_dump.sql
```

```
$ pg_dump --host=... <db> | psql --no-psqlrc --host=... <db>
```

ALLE Datenbanken

```
$ zcat full_dump.sql.gz | psql --no-psqlrc
```

```
$ psql --no-psqlrc --file=full_dump.sql
```

- **Wann braucht man das?**

- Daten kaputt / Major Upgrade / Migration
- Nicht mehr geeignet bei grossen DB's!
 - 100e Gbyte → RTE (Angepeilte Recovery Zeit)

Backup – physisch I

- **Was passiert?**
 - **Backup-Tool verbindet sich mit Replication Protocol zur Datenbank**
 - **Checkpoint + LSN merken**
 - **Datafiles werden gestreamt DB → Tool**
 - **WALs werden gestreamt (2. Verbindung) DB → Tool**
 - **... bis Backup fertig.**

Backup – physisch II

- Wie macht man das?

Lokal

```
$ pg_basebackup --format=plain --pgdata=/mnt/backup/basebackup
```

Remote (super cool!)

```
postgres=# CREATE ROLE backupuser  
        WITH LOGIN PASSWORD 'secret' REPLICATION;
```

```
# pg_hba.conf  
# TYPE  DATABASE        USER            ADDRESS                 METHOD  
host    replication      backupuser      10.223.125.0/24        scram-sha-256
```

```
$ pg_basebackup --user=backupuser --host=10.223.125.193 \  
    --format=plain --pgdata=/mnt/backup/basebackup
```

Backup – physisch III

- Was erhält man (`$datadir` / `$PGDATA`)?

```
$ cd /mnt/backup/basebackup/
$ ll
-rw----- 1 dba dba      230 Sep 23 17:05 backup_label
-rw----- 1 dba dba 716156 Sep 23 17:05 backup_manifest
drwx----- 14 dba dba    4096 Sep 23 17:05 base
drwx----- 2 dba dba    4096 Sep 23 17:05 global
...
-rw----- 1 dba dba    6011 Sep 23 17:05 pg_hba.conf
-rw----- 1 dba dba    2681 Sep 23 17:05 pg_ident.conf
drwx----- 4 dba dba    4096 Sep 23 17:05 pg_logical
...
-rw----- 1 dba dba        3 Sep 23 17:05 PG_VERSION
drwx----- 4 dba dba    4096 Sep 23 17:05 pg_wal
drwx----- 2 dba dba    4096 Sep 23 17:05 pg_xact
-rw----- 1 dba dba    146 Sep 23 17:05 postgresql.auto.conf
-rw----- 1 dba dba  32693 Sep 23 17:05 postgresql.conf
```

Restore – physisch

- Wie macht man das?

```
$ sudo systemctl stop postgresql
$ rm -rf /var/lib/postgresql/18/main/*
$ (s)cp --archive /mnt/backup/basebackup/* /var/lib/postgresql/18/main/
$ chown postgres: /var/lib/postgresql/18/main/* ?
$ sudo systemctl start postgresql
```

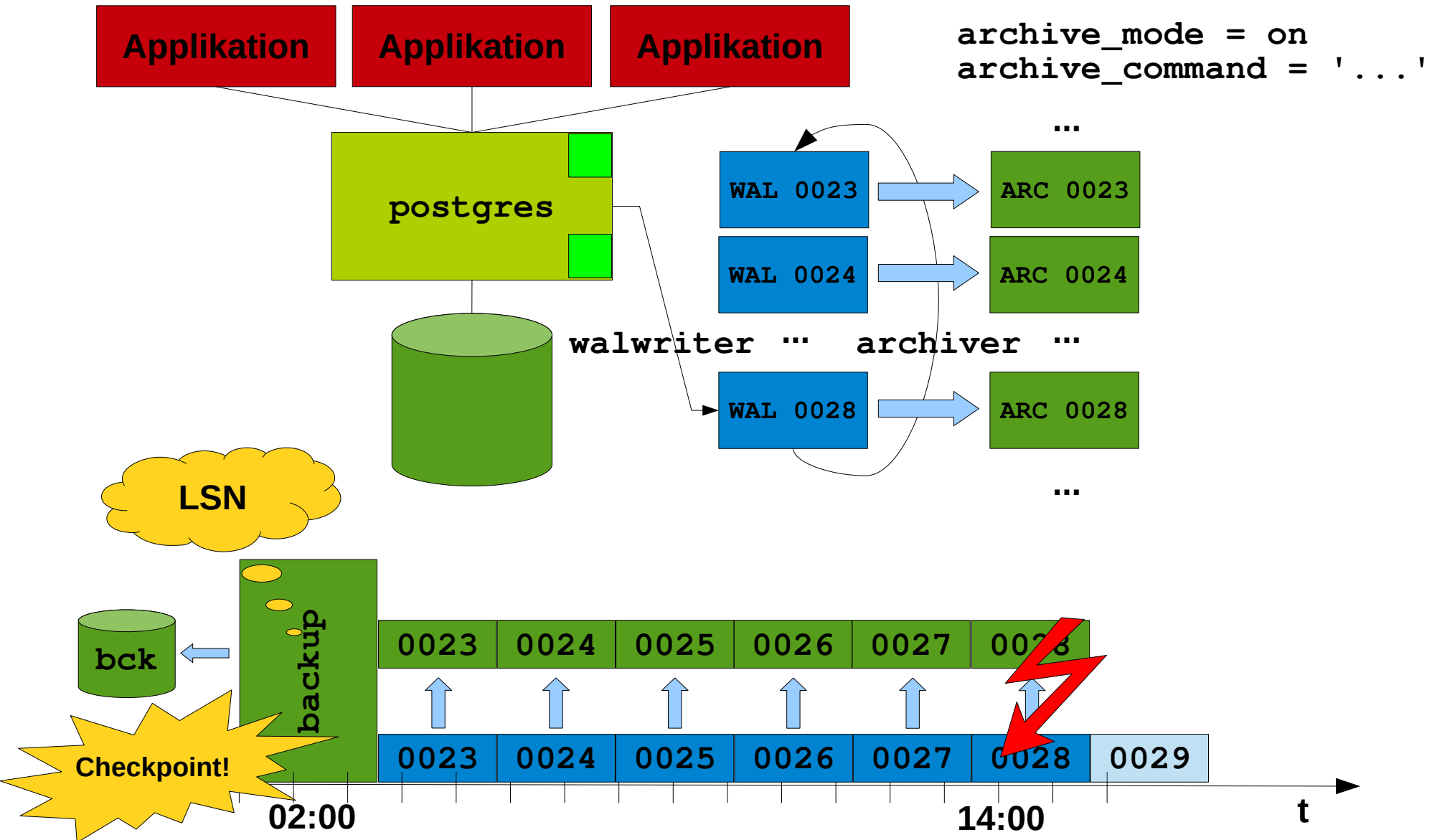
- Wann braucht man das?
 - Wenn Daten kaputt
 - Grundlage für PiTR
 - Grundlage für physische Replikation
- Gut geeignet bei grossen DB's!
 - 1 Tbyte/h → RTE ist realistisch! (bei vernünftiger Infrastruktur...)

Point-in-Time-Recovery (PiTR)

www.fromdual.com

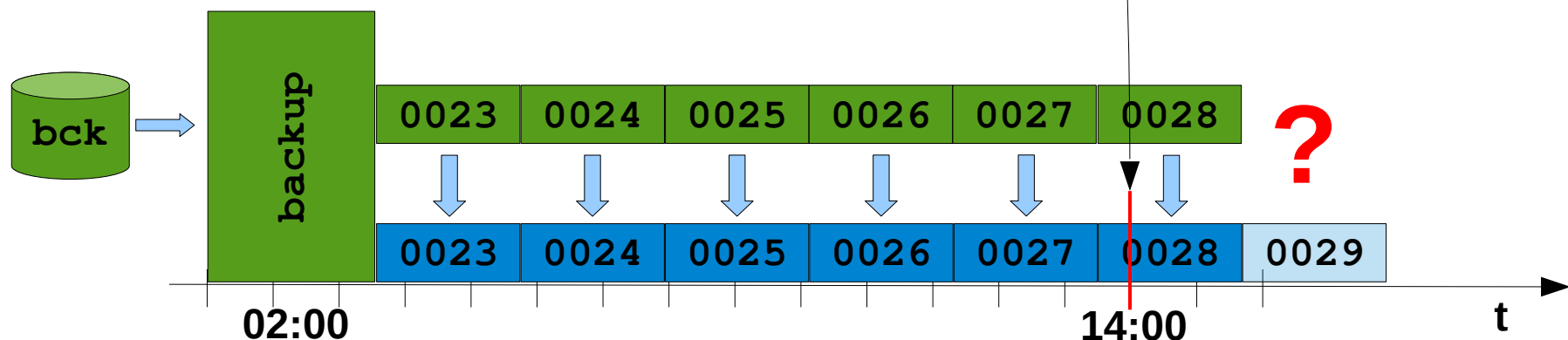
- **Zwischenfall:**
 - **Hardware-Schaden**
 - sofort kaputt, eher selten
 - **Korrupte Blocks oder**
 - **Logische Fehler (Ups-Queries!, häufig?)**
 - merken wir erst "später"
- **Problem**
 - **Backup 1 x pro Tag (02:00?)**
 - Daten von Backup bis jetzt sind futsch! (12 h!)
- **Hier hilft PiTR!**
 - **Vom letzten Backup aus**
 - **Recovery zu einem bestimmten (Zeit-)Punkt oder zum Schluss!!!**

PiTR Schaden



PiTR Reparatur

- Restore vom Backup
- Setze `restore_command`
- und `recovery_target_time`
- Datei `recover.signal` anlegen
- Server starten



PiTR Doing - Backup

- **WAL Archivierung aktivieren**

```
# postgresql.conf
archive_mode = on
archive_command = 'test ! -f /mnt/arch/%f && cp %p /mnt/arch/%f \
&& sync /mnt/arch/%f'
archive_timeout = '15 min'    # max RPO!
wal_level = replica          # default
```

WAL Monitoring:

```
SQL> SELECT * FROM pg_stat_archiver;
```

- **Backup**

```
$ pg_basebackup --format=plain --pgdata=/mnt/backup/basebackup
```

- **WAL und Backup vom Server weg kopieren!**

PiTR Doing - Restore

- Datenbank stoppen / killen
- \$PGDATA sichern (optional)
- Aktuelle WAL weg sichern (pg_wal/)
- Restore wie vorher beschrieben (rm && cp --archive && chown)
- Nicht archivierte WAL ins WAL Archiv kopieren
- Konfiguration anpassen:

```
# postgresql.conf
restore_command = 'cp /mnt/arch/%f %p'
listen_address = ''
recovery_target_time = '2026-09-21 14:25:37' # optional
```

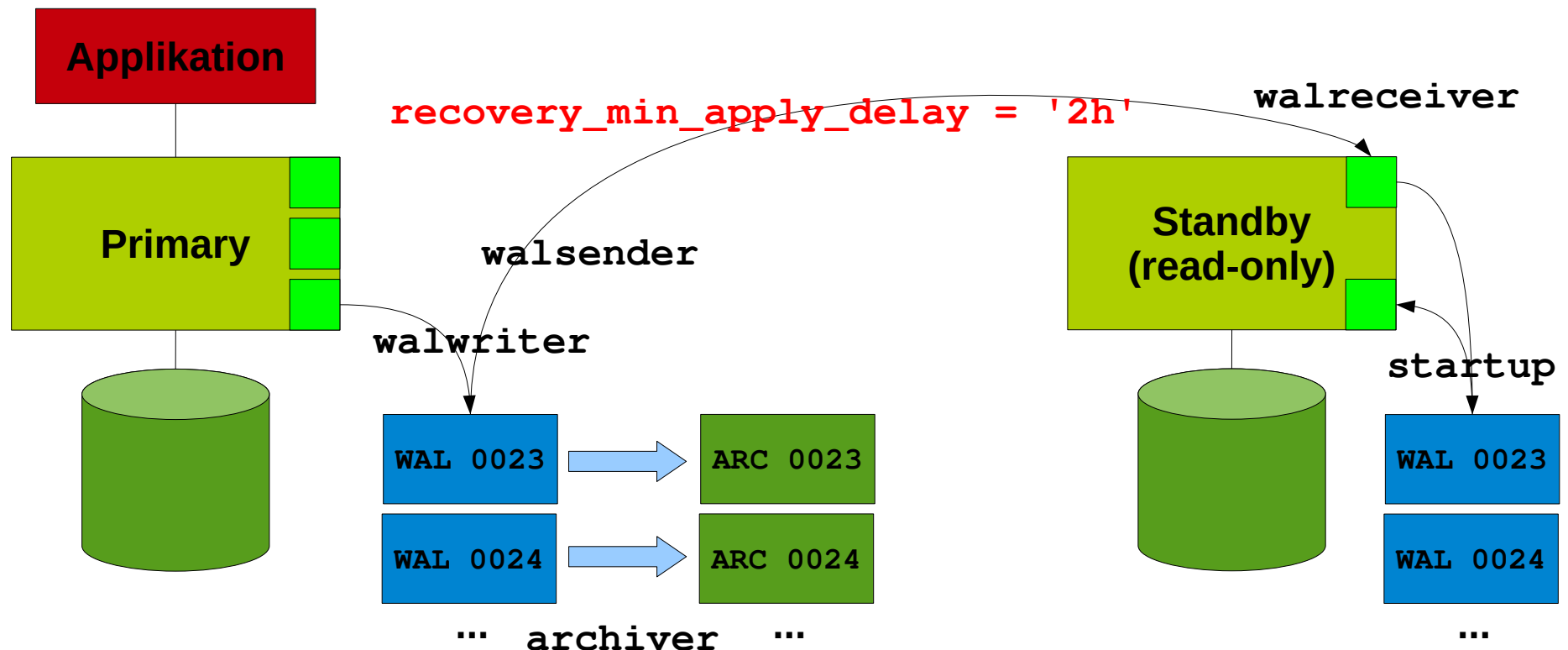
- Datei \$PGDATA/recovery.signal anlegen
- Datenbank starten
- PostgreSQL "promoten" (bei recovery_target_*):

```
SQL> SELECT pg_promote();
pg_promote
-----
t
```

- Konfiguration zurücksetzen (restore*, recovery*)
- Datenbank neu starten (listen_address!)

Verzögerte Replikation I

- PostgreSQL hat:
 - Physical Streaming Replication (default: async)



Verzögerte Replikation II

- **Use cases:**
 - **Ups-Queries!**
 - **Bestimmte Nutzergruppen "ältere" Daten**
 - **"freemium" model:**
 - Online-Trading / Börsenhandel
 - Online-Wetten
 - Sportresultate, ...
 - Online-Spiele
 - Stellenangebote
 - Wetterdaten, etc
 - **Systemtests bei Replikations-Verzögerungen**
 - **Historischer Zustand der Datenbank zeigen**
 - max. 2^{32} Sekunden ($\cong$ 136 Jahre)

Verzg. Replikation einrichten

- **Primary (Neustart erforderlich!)**

```
# postgresql.conf
listen_addresses = '*'
archive_mode      = on
archive_command   = 'test ! -f /mnt/arch/%f && cp %p /mnt/arch/%f && sync /mnt/arch/%f'
```

- **Standby (wie PiTR):**

```
# postgresql.conf
archive_cleanup_command = 'pg_archivecleanup /mnt/arch %r'
recovery_target_timeline = latest # default
primary_conninfo = 'host=10.223.125.65 port=5432 user=replication options=''\'
    -c wal_sender_timeout=5000''\'
recovery_min_apply_delay = '2h'
```

- **Abbruch/Aufholen wie PiTR...**

- **Quellen**

- <https://hexacluster.ai/blog/streaming-replication-internals-for-high-availability-in-postgresql>
- <https://www.fromdual.com/blog/postgresql/artificially-delayed-replication-with-postgresql/>

Korrupte Datenfiles

- **Wie passiert das?**
 - Hardware geht kaputt (SSD!, RAID-Controller, ...)
 - "Jemand" "fummelt" an den Daten rum
 - Bugs (O/S, VM, LVM, F/S, Container, ...)
 - Bit-Flips, Sonnenwind, etc...
- **PostgreSQL Schutzmassnahme:**
 - Seit PostgreSQL v18 (bei neuem Cluster)
- Nachträglich einschalten: pg_checksum (Cluster down!)
- PostgreSQL v19 (Q4 2026):

```
data_checksums = on
```

```
SELECT pg_enable_data_checksums();
```

Was passiert technisch?

- PostgreSQL:
 - Beim Schreiben: Jeden Block mit Checksumme versehen
 - Beim Lesen: Checksumme jedes Blocks kontrollieren...
- Wenn es doch passiert:
 - *"I'm unable to access one of the tables. Even a simple SELECT statement fails with the error below."*

```
prod=# SELECT count(*) FROM schema.tablename;
```

```
WARNING:  page verification failed, calculated checksum 26618  
but expected 52580
```

```
ERROR:   invalid page in block 43197 of relation  
base/24576/24578
```

Interpretation und Massnahmen

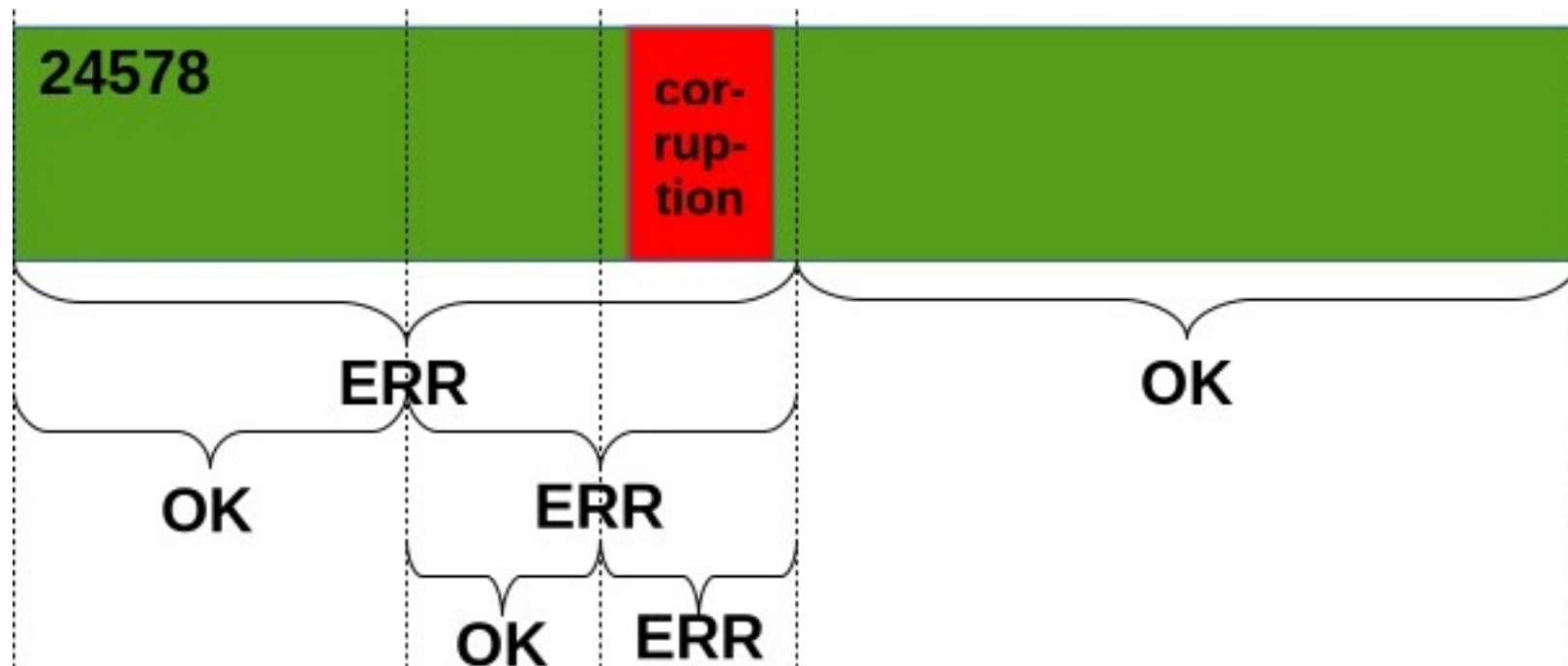
www.fromdual.com

ERROR: invalid page in block 43197 of relation **base/24576/24578**

- **relation = table (kann auch Index sein?)**
- **pg_database: 24576 → Datenbank test**
- **pg_class: 24578 → Tabelle test im Schema public**
- **→ passt zum Query**
- **Massnahme 1: Backup + PiTR!!!**
- **Und sonst?**

Vorgehen I

- Das macht man nur, wenn man KEIN Backup hat!
(Binary search)



Vorgehen II

```
-- Bereich "abtasten"
SELECT * FROM test WHERE id < ...

CREATE TABLE test_copy (LIKE test INCLUDING ALL);

SET enable_seqscan = off;

INSERT INTO test_copy SELECT * FROM test WHERE id < 4622080;
INSERT INTO test_copy SELECT * FROM test WHERE id > 4622186;

-- 106 Rows fehlen noch

SET ignore_checksum_failure = on;

INSERT INTO test_copy SELECT * FROM test WHERE id = ...;

-- 5 Rows Verlust!

DROP TABLE test CASCADE;
ALTER TABLE test_copy RENAME TO test;
-- FOREIGN KEYS und VIEWS wieder anlegen
```

Backup testen!

- **Backups NIE auf der selben Maschine wie die Datenbank speichern. Wenn die Maschine verloren geht, ist alles verloren!**
- **Traue Deinem Backup nicht! Traue nur dem Restore. Du weisst erst, ob Dein Backup gut war, wenn Du es zurückspielen konntest!**
- **Restore kann lange dauern: RTO / RPO!**
- **Regelmässiges Testen gibt Übung!**
- **Quelle:**
 - **<https://www.percona.com/blog/meet-pgstef/>**

Von einander lernen...



www.fromdual.com



Quelle: quillbot.com

Q & A



www.fromdual.com



Fragen ?
Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig an:**
 - **On-site Beratung**
 - **remote-DBA**
 - **Support für MariaDB, Galera und MySQL**
 - **Schulungen**

www.fromdual.com/presentations