



# When Collations Corrupt a PostgreSQL Database

PGDay Belgium 2026

Emrah Becer

5 May 2026

UCLL – Leuven, Belgium

# Today's Task: Upgrade Your Linux Operating System



## Current Configuration

- © Ubuntu Server 18.04
- © PostgreSQL 14.8



## Desired Configuration

- © **Ubuntu Server 22.04**
- © PostgreSQL 14.8

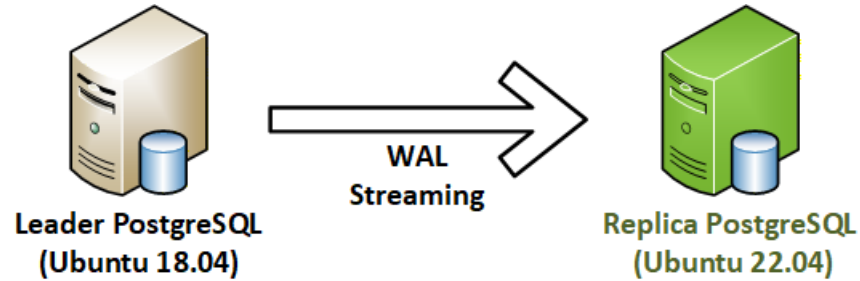
# Hello!

## I am Emrah Becer



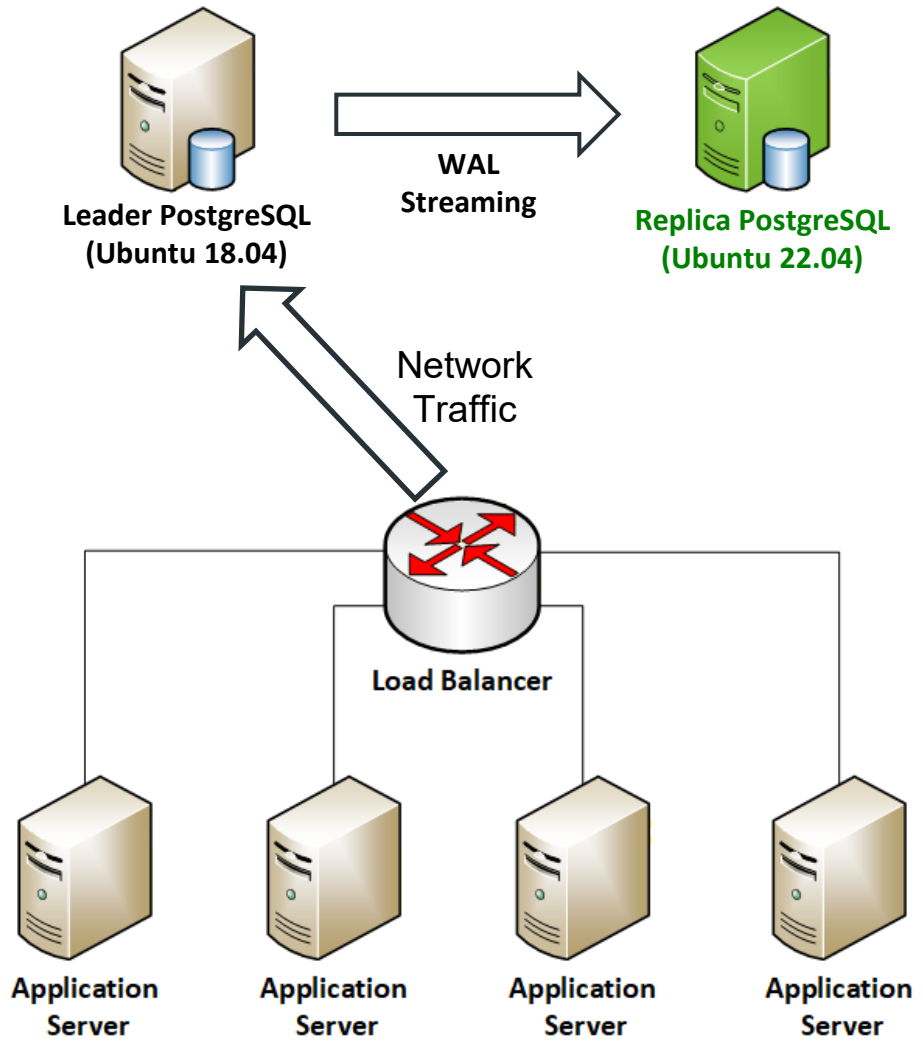
- \* A computer engineer who spent last 20 years administering databases.
- \* Currently working as an independent PostgreSQL consultant at European Patent Office (EPO)
- \* Have special interest in Linux kernel and database internals

# Method: Physical WAL Replication

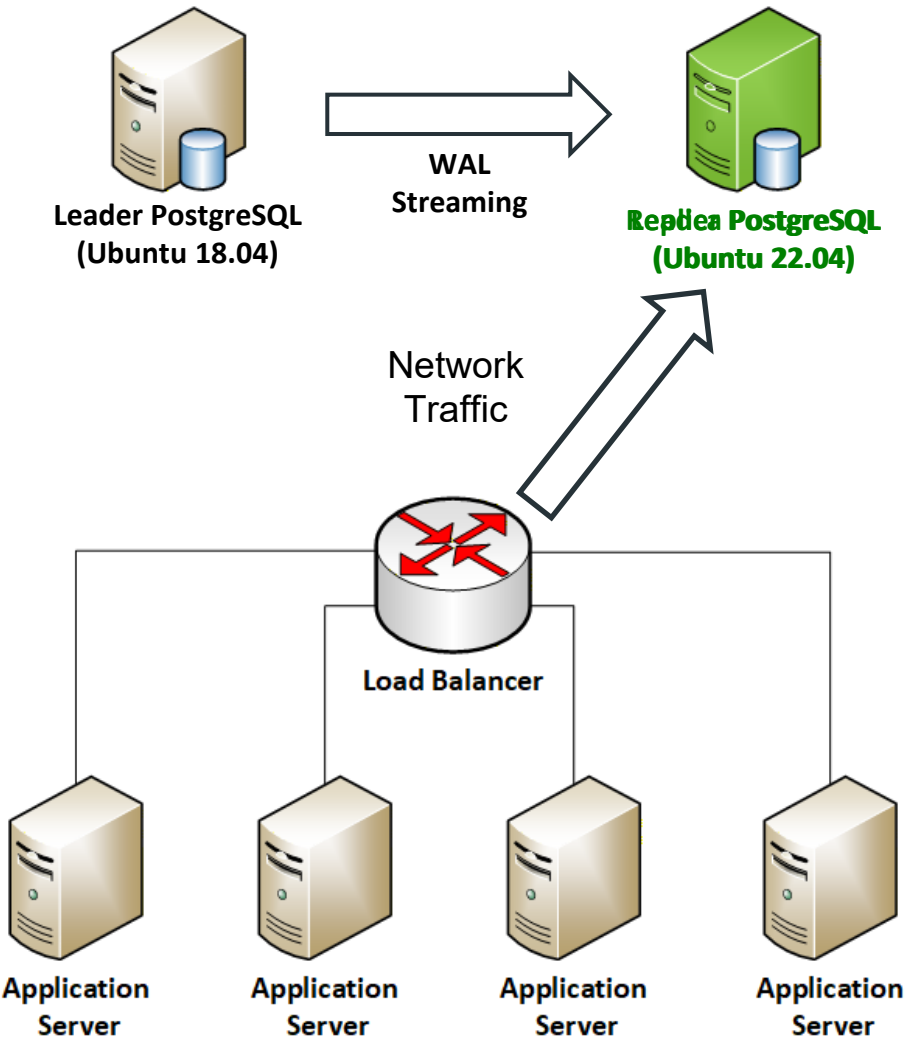


- © Easy to setup
- © Minimum Downtime

# ENTIRE TOPOLOGY



# ENTIRE TOPOLOGY



# Let's Recap



- ◎ We did setup physical replication
- ◎ Row counts were identical
- ◎ We promoted the replica
- ◎ No exception was raised.
- ◎ Life is beautiful



**A FEW  
MOMENTS LATER**

# Users Start Complaining



- ◎ Missing data reported
- ◎ Duplicate data observed



# What Went Wrong?

We'll find out shortly—but first, some theory

# Flashback to Primary School

ö



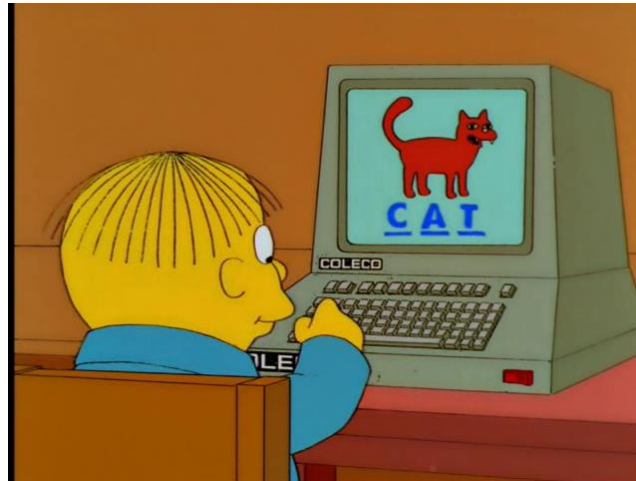
Language	Characteristic	Order
German	“o” with umlaut a variation of “o”	a → ... → o → ö → z
Swedish	distinct letter	a → ... → o → ... z → ö
Turkish	distinct letter	a → ... → o → ö → z

# Takeaway

A letter may have a different characteristic and sort order in different languages



# How Do Computers Store Text?



# Unicode Code Points

ö

- ◎ Code point = U+00F6
- ◎ Hex: 00F6 , Dec: 246
- ◎ Language and OS Independent



# How Do Computers Sort Text?

# Unicode “Weights”

- ◎ Weight = a numeric value
- ◎ Each language assigns a weight to characters.
- ◎ **Computers sort “weights”**

A decorative network diagram in the top-left corner, consisting of various sized nodes (some solid grey, some hollow white) connected by thin grey lines, forming a complex web-like structure.

# So, What's a Collation?



“

**Collation** = Sorting rule  
(character weights) for a given  
language

# Collation Identifier

language = English

encoding = UTF-8

en\_US.UTF-8

region=United States



# Collation in PostgreSQL

# Let's sort a, z, o and ö

```
SELECT * FROM (  
VALUES ('a'), ('z'), ('o'),  
('ö')) AS T (LETTER)  
ORDER BY LETTER  
COLLATE "de_DE"
```

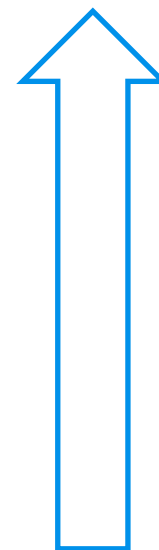
```
SELECT * FROM (  
VALUES ('a'), ('z'), ('o'),  
('ö')) AS T (LETTER)  
ORDER BY LETTER  
COLLATE "sv_SE"
```

```
SELECT * FROM (  
VALUES ('a'), ('z'), ('o'),  
('ö')) AS T (LETTER)  
ORDER BY LETTER  
COLLATE "tr_TR"
```

a	a	a
o	o	o
ö	z	ö
z	ö	z

# Where Collation Can Be Set

Select * from products order by product_name <b>collate "tr_TR"</b>	Statement Level
create table employees (emp_name varchar(100) <b>collate "sv_SE"</b> )	Column/Index Level
create database mydb <b>locale 'de_DE.utf8'</b> template 'template0'	Database Level
\$ initdb <b>--locale=en_US.utf8</b>	Cluster Level



Higher  
Precedence

Lower  
Precedence

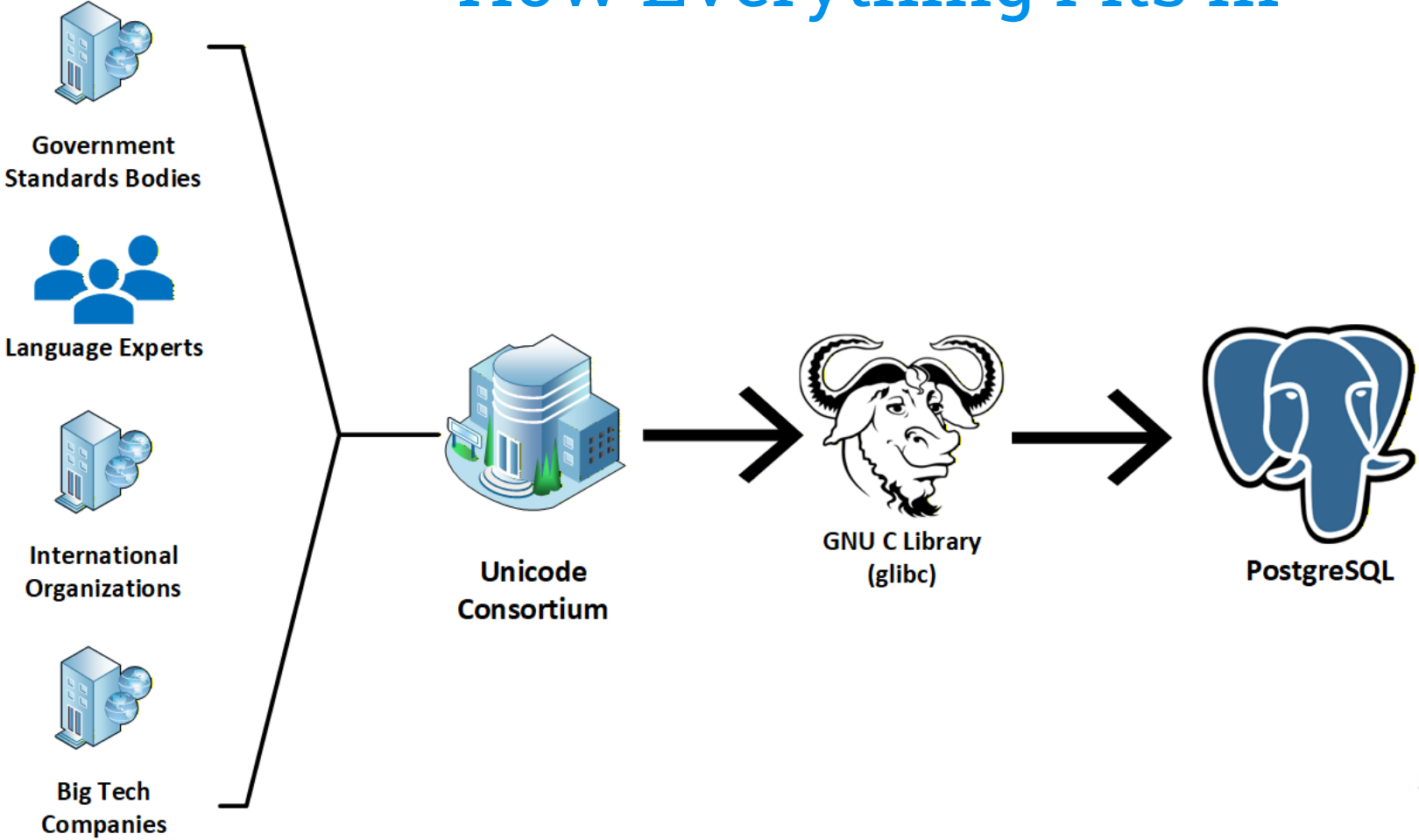
A decorative network diagram in the top-left corner, consisting of various sized grey circles (nodes) connected by thin grey lines (edges). Some nodes are solid, while others are hollow with a dashed border. The network is dense and irregular, extending from the top-left towards the center.

# Collation in Linux

# Locale Files in Linux

- ◎ Maintained by GNU Project
- ◎ Files reside under “/usr/lib/locale/”
- ◎ Exposed to applications via GNU C Library (glibc)

# How Everything Fits In



# PostgreSQL's Interaction with GNU C Library



Can you compare  
“ö” against “z” in  
sv\_SE please

**strcoll\_l**  
function call



“ö” is greater than  
“z” in sv\_SE

# Takeaway

PostgreSQL is UNAWARE of  
language sorting rules.

It depends on glibc



# So, why we messed up during our upgrade?



# What changed between Ubuntu 18 and 22 ?

## 18.04

- ◎ glibc 2.27
- ◎ “a” → “\$a”
- ◎ “p1” → “p-1”



## 22.04

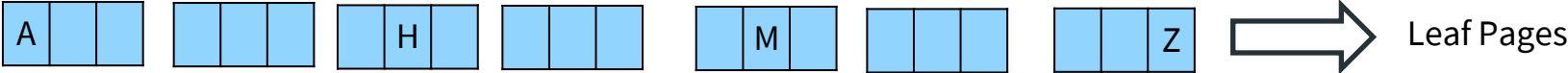
- ◎ glibc 2.35
- ◎ “\$a” → “a”
- ◎ “p-1” → “p1”

# Takeaway

Grammar rules can change  
over time.  
They are NOT immutable.



# B-Tree Index Structure In Postgres



Keys are stored in pages as **sorted**

# Let's Recap



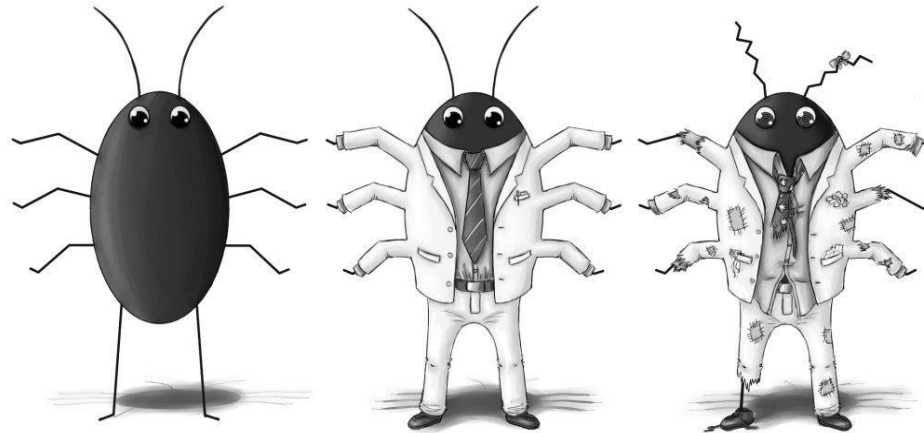
- ◎ After OS upgrade, glibc version changed
- ◎ Collation (sorting rules) changed
- ◎ Index was build in old collation
- ◎ Index lookups used the new collation
- ◎ Index got logically corrupted



# How to Avoid Collation Corruptions

# Current Situation

- ◎ By PostgreSQL 18, still no ultimate solution.
- ◎ Different approaches with different trade-offs



**BUG**

**FEATURE**

**BY DESIGN**

# 1. Reindex All Text Based Indexes

## Pros:

- ⦿ Guaranteed fix. Most popular method

## Cons:

- ⦿ Expensive operation
- ⦿ If done online:
  - Degrade user experience
  - Risk of unique index violation
- ⦿ Will result in downtime if done offline

## 2. Logical Replication

### Pros:

- ⦿ Guaranteed fix.
- ⦿ Minimal downtime

### Cons:

- ⦿ High setup cost.
- ⦿ Error prone. Ex: replication of sequences, DDL statements.



# 3. Compile Your Own PostgreSQL

- ⦿ Compile your own postgresql with static ICU (International Components for Unicode) library

## Pros:

- ⦿ Guaranteed fix
- ⦿ No need to worry about glibc version.

## Cons:

- ⦿ Overkill
- ⦿ Compatibility?

# AMCHECK, A Diagnostic Tool

- ⦿ create extension amcheck;
- ⦿ `select bt_index_check('<index_name>');`
- ⦿ Will report any corruption detected

# Methods That Won't Help



# Methods That WON'T Help

- ⦿ Physical backup/restore (pg\_basebackup, pgBackRest)
- ⦿ Upgrading Postgres Version (pg\_upgrade)
- ⦿ Using ICU libraries without compiling your own postgresql

A decorative network diagram in the top-left corner, consisting of various sized grey circles (nodes) connected by thin grey lines (edges). Some nodes are solid grey, while others are hollow with a dashed border. The network is dense and irregular, extending from the top-left towards the center of the slide.

# Final Words

# Final Words

- ◎ By today, there is still no ultimate solution.
- ◎ Rebuilding all text based indexes is the most suitable solution for majority of cases.
- ◎ For edge cases, consider :
  - logical replication or
  - selective re-indexing.

# Thank You!

## Any questions?

You can find me at:



Emrah Becer



[info@denizdata.com](mailto:info@denizdata.com)



<https://www.linkedin.com/in/emrah-becer/>