

# WHAT I DON'T WANT TO EXIST IN THE DATA WORLD IN 5 YEARS

Provided by Seattle Data Guy

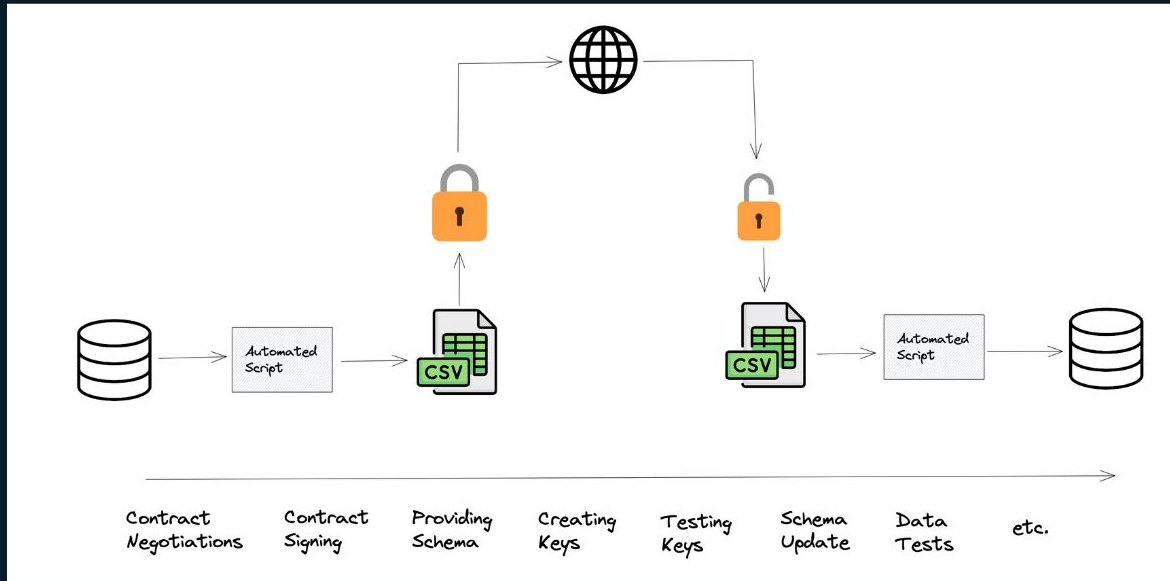
# Agenda

01 SFTP

02 Data Team Standards

03 Data Engineers

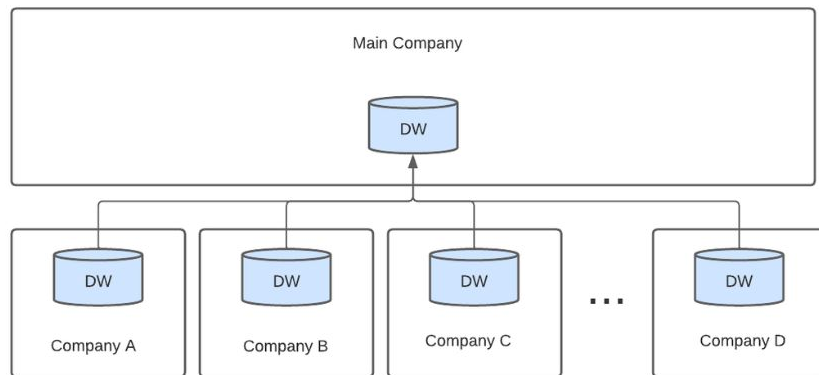
# Sharing Data Current State



## Problems

- Slow to develop/process
- Heavily impacted by change
- Duplicated data
- Risk of data being forgotten

# Sharing Data Future State



## Improvements

- No more duplicate data
- Removes data engineering busy work
- Changes can still break pipelines but are far more transparent

Contract  
Negotiations

Contract  
Signing

~~Providing  
Schema~~

~~Creating  
Keys~~

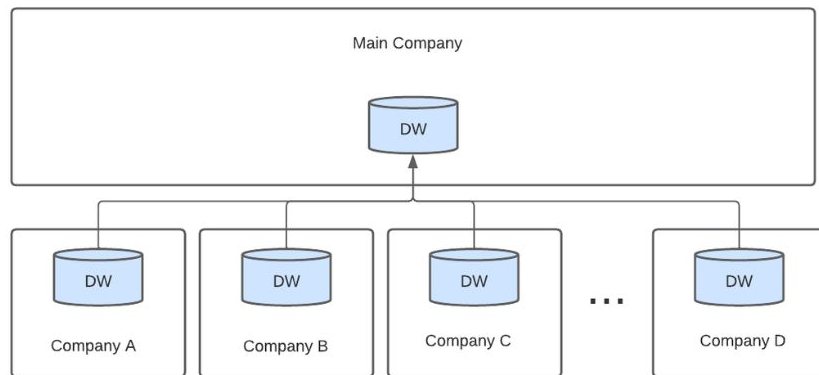
~~Testing  
Keys~~

~~Schema  
Update~~

~~Data  
Tests~~

~~etc.~~

# Sharing Data Future State



## Improvements

- No more duplicate data
- Removes data engineering busy work
- Changes can still break pipelines but are far more transparent

Contract  
Negotiations

Contract  
Signing

~~Providing  
Schema~~

~~Creating  
Keys~~

~~Testing  
Keys~~

~~Schema  
Update~~

~~Data  
Tests~~

~~etc.~~

# Current Data Team Standards

01

Many teams lack clear SQL and coding standards

02

Many teams lack onboarding documentation

03

Many teams lack design standards

04

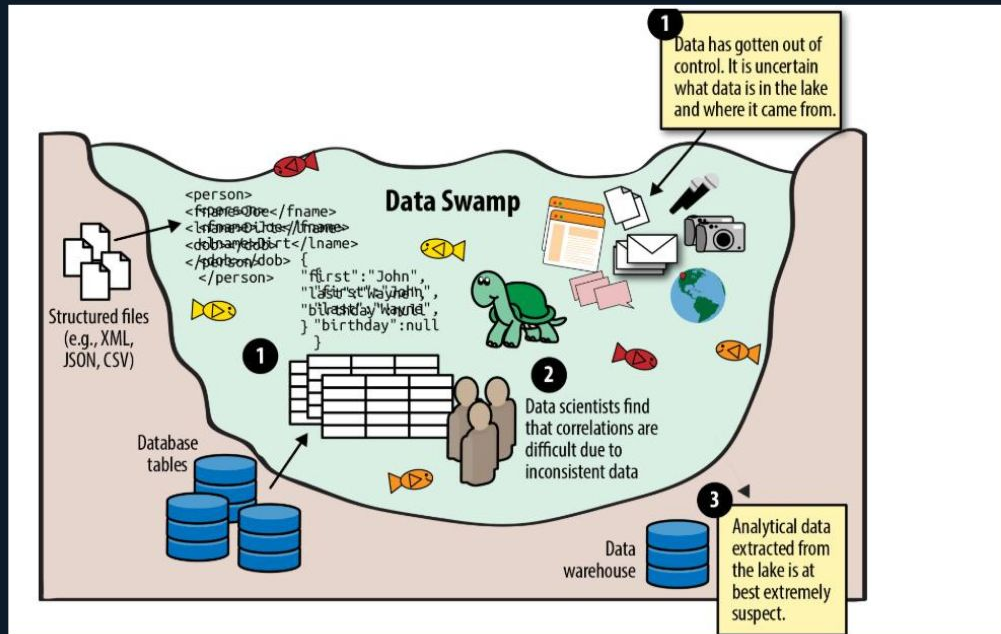
Many teams lack design standards





**What Does This  
Lead To?**

# Sharing Data Future State



Source: [r/dataengineering/](#) - u/DrRedmondNYC

## Improvements

- Data swamps
- Key-person dependency issues
- Constant rework every few years
- Debugging and maintaining slow-downs
- Chaos



## Best Practices

- No tabs should be used - only spaces. Your editor should support [tab settings](#) for more details.
- Wrap long lines of code, between 80 and 100, to a new line.
- Do not use the `USING` command in joins because it prevents you from [viewing the forum discussion on this topic](#).
- Understand the difference between the following relational operators:
  - `UNION ALL` and `UNION`
  - `LIKE` and `ILIKE`
  - `NOT` and `!` and `<>`
  - `DATE_PART()` and `DATE_TRUNC()`
- Use the `AS` operator when aliasing a column or table.
- Prefer `DATEDIFF` to inline additions `date_column + 1` to work for a wider variety of date parts.
- Prefer `!=` to `<>`. This is because `!=` is more common in natural language, which is how we're more likely to speak.
- Prefer `LOWER(column) LIKE '%match%'` to `column LIKE '%match%'` to avoid case letters leading to an unexpected result.
- Prefer `WHERE` to `HAVING` when either would suffice.



# Future State Of Data Team Standards

Coding, SQL and design standards are everywhere. There is no longer any need to join a team that has no form of onboarding documentation.

Increased  
Transparency of  
Standards

Clear Walk  
Throughs On What  
Is Required For  
Onboarding

Increasing Ownership  
By Involving New Hires

# Will Data Engineers Continue To Exist



**Zach Wilson** ✓

@EcZachly

...

Will data engineering still exist in 10 years?

My prediction is DE will explode over the next 5 years and then a good chunk of DE roles will morph / fade away.

The DE roles that do large-scale integrations of many different types of data sources will have secure jobs.

DE that work vertically with software engineers and analysts are a little more in trouble.

These DEs will experience pressure to either own more of the infrastructure like SWEs or own more of the metrics like analysts.

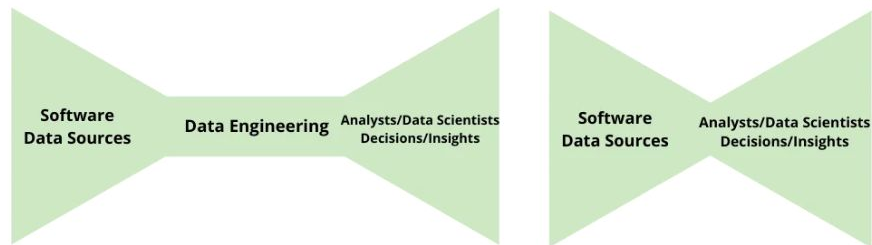
This type of pressure will push DEs to analytics engineering over the long term.

Where do you think DE is going?

[#dataengineering](#)

7:59 PM · Mar 12, 2023 · **58.2K** Views

# Current State Of Data Engineers



Current State

What Companies Want

- ☑ Everyone has always wanted to replace data engineers (in all their current and previous forms)
- ☑ Let's talk about why it likely won't happen
- ☑ At least not yet...



# Thank You