**Foreman: Building a tailored data assistant using dbt metadata**

**Slide 1**

(Cover)

**Slide 2**

To kick off our session, we'd like to introduce you to a role that might seem a bit old-fashioned but is incredibly relevant to our discussion.

This is our friend (**name**?) and he is a warehouse foreman.

Imagine a warehouse that is filled with goods, machinery, and workers. In this environment, **X** is the figure who ensures everything runs smoothly. He oversees operations, manages inventory, coordinates tasks, and ensures that every item is in its rightful place. As a warehouse foreman, **X**’s intimate knowledge of the warehouse layout, processes, and personnel is crucial for maintaining efficiency and order.

But of course, we’re not here to talk about physical warehouses.

Just like a physical warehouse, a data warehouse is a repository where vast amounts of data are stored, organized, and managed. The data warehouse is the backbone of data operations in many organizations, housing critical information that drives business decisions. However, managing a data warehouse can be just as complex as managing a physical warehouse. This is where the concept of a “foreman" can becomes incredibly valuable for data.

**Slide 3** (start of 1. The revival of the DWH with the cloud)

In the past, data warehouses were often on-premises systems, characterized by:

* Rigid Structures: Traditional data warehouses had fixed schemas and were not easily adaptable to changing business needs.
* High Maintenance Costs: Managing hardware, software, and storage was expensive and required significant IT resources.
* Limited Scalability: Scaling up often meant investing in more hardware, which was both costly and time-consuming.
* Batch Processing: Data was typically processed in batches, leading to delays in data availability and analysis.

**Slide 4**

The advent of cloud technologies has revolutionized the landscape of data warehousing, breathing new life into what was once considered a rigid and costly infrastructure. Cloud-based data warehouses have brought about a revival by addressing the limitations of traditional systems and introducing a host of new capabilities.

Firstly, cloud-based data warehouses offer unparalleled flexibility and scalability. Unlike their on-premises counterparts, these modern systems can scale elastically, adapting to varying workloads and data volumes. This means organizations can easily scale up during peak times and scale down during quieter periods, optimizing both performance and cost.

Real-time processing is another significant advancement. Traditional data warehouses often relied on batch processing, which could delay data availability and analysis. In contrast, modern cloud-based warehouses support real-time data ingestion and processing, enabling organizations to gain immediate insights and make timely decisions.

Cost-effectiveness is also a major benefit. The pay-as-you-go pricing model of cloud services eliminates the need for hefty upfront investments in hardware and software. Organizations only pay for the resources they use, making it easier to manage budgets and reduce financial risk.

Moreover, the seamless integration with advanced analytics and machine learning tools allows organizations to extract deeper insights and build predictive models, driving innovation and competitive advantage.

**Slide 5** (start of 2. The role of dbt and “Analytics Engineering”)

The transformation of data warehousing through cloud technologies has paved the way for a new era in data management and analytics: analytics engineering. This emerging discipline bridges the gap between data engineering and data analysis, focusing on creating well-structured, reliable, and scalable data pipelines that empower data analysts and business users. At the heart of this transformation is the Data Build Tool (dbt), which plays a pivotal role in enabling analytics engineering.

Analytics engineering is about applying software engineering best practices to data workflows. It emphasizes the importance of version control, testing, documentation, and modularity in building data pipelines. This approach ensures that data is not only accurate and reliable but also easily accessible and understandable by end-users. The goal is to create a robust data infrastructure that supports advanced analytics and decision-making.

**Slide 6**

dbt is a powerful tool that has become the cornerstone of analytics engineering. It allows data teams to transform raw data into meaningful insights by writing modular SQL queries and managing them as code. Here’s how dbt contributes to the modern data landscape:

* Version Control and Collaboration: dbt integrates seamlessly with version control systems like Git, enabling teams to collaborate on data transformations in a controlled and transparent manner. This ensures that changes are tracked, reviewed, and approved, reducing the risk of errors and inconsistencies.
* Testing and Validation: dbt allows data engineers to write tests for their data transformations, ensuring data quality and integrity. Automated testing helps catch issues early, preventing faulty data from propagating through the pipeline.
* Modularity and Reusability: dbt encourages modularity by allowing data transformations to be broken down into reusable components. This modular approach simplifies the development process and promotes consistency across the data pipeline.

The combination of cloud-based data warehouses and dbt creates a powerful synergy that enhances the capabilities of analytics engineering. Modern data warehouses provide the scalable and flexible infrastructure needed to store and process large volumes of data, while dbt offers the tools to transform and manage this data efficiently.

Together, they enable organizations to build robust data pipelines that deliver high-quality, reliable, and actionable insights. This integrated approach not only improves data accessibility and usability but also accelerates the development of advanced analytics and machine learning models.

**Slide 7**

Among the many features that dbt offers, its documentation capabilities stand out as a critical component for the concept of a data warehouse foreman. Documentation in dbt is not just an add-on; it is a fundamental feature that provides context, transparency, and trust in the data.

Here’s why documentation is key to our discussion.

Dbt automatically generates comprehensive documentation for every model, test, and transformation within the data pipeline. This metadata includes descriptions of data sources, transformation logic, and dependencies, providing a clear and detailed map of the data landscape. This level of documentation ensures that every stakeholder, from data engineers to business analysts, can understand the data’s journey and its current state.

By leveraging dbt’s documentation capabilities, organizations can create a data foreman that ensures data is well-organized, accessible, and useful. This data foreman can oversee data operations, manage data inventory, coordinate tasks, and ensure data quality and compliance, much like a traditional warehouse foreman.

**Slide 8** (start of 3. Usage of GenAI for Data Analytics & Data Literacy)

In data warehousing, two prevalent issues frequently stand out. First, source providers often fail to document their data adequately, leading to confusion and errors. Second, data analysts frequently struggle to identify the specific information needed to answer business questions. These problems result in reduced data quality, inefficiency, and increased risk. Without proper documentation and clear data understanding, organizations face significant challenges in data literacy.

**Slide 9**

Data literacy is defined as the ability to read, understand, create, and communicate data as information. Enhancing data literacy is crucial for making informed decisions and fostering a data-centric culture. Strategies for improvement include implementing comprehensive training programs, establishing robust documentation standards, and providing accessible resources. By investing in these areas, organizations can empower their teams to better understand and utilize data, leading to more effective and efficient operations.

**Slide 10**

In the Data and AI Trends Report 2024 published by Google, data literacy was identified as one of the key trends in Generative AI applications, mentioning that “as tools become more accessible, even non-technical team members will benefit from data insights. This democratization of data means better data literacy across your organization, smarter decisions being made, and ultimately greater success in the market”. The increasing accessibility of data tools is a key driver for organizational success. Ensuring that all team members can understand and use data effectively is critical for staying competitive.

**Slide 11**

But if we look at where the money is being invested, it is difficult to identify explicit data literacy. In this bar chart by Gartner with the most GenAI investment in 2023, even taking into account that a lot is changing in this field, data literacy is not on the list, but it should be. Investment in data literacy is crucial for maximizing the benefits of GenAI technologies. By prioritizing data literacy, organizations can ensure that all team members are equipped to leverage GenAI tools effectively, leading to smarter decisions and greater overall success.

**Slide 12** (start of 4. the “Foreman”)

Just as we saw in the beginning, a warehouse foreman oversees operations and ensures everything runs smoothly, a data warehouse foreman manages and optimizes data workflows. By combining traditional data warehousing practices with the power of Generative AI, it is possible to create robust systems that ensure data quality, accessibility, and efficiency. This role leverages can and should leverage automated documentation capabilities such as the ones provide by dbt to serve as a comprehensive knowledge base, enhancing the data management capabilities.

**Slide 13**

Nowadays, when we talk about Generative AI, the Retrieval Augmented Generation, or RAG, is a common buzzword. The idea is that we can have a simple prompt to an LLM but the response is generated solely based on the model's existing knowledge. While this can be useful, it often lacks the context and specificity needed for accurate and actionable insights. This is where RAG comes into play, enhancing the response by incorporating relevant external information.

**Slide 14**

Now, let's look at a prompt with RAG. RAG consists of three key components: Retrieval, Augmentation, and Generation. Retrieval involves obtaining pertinent passages of text from an external knowledge source. Augmentation adds more context, boosts the response's accuracy, or improves its fluency. Finally, Generation produces a new answer based on the enhanced data. This process ensures that the responses are more accurate, context-aware, and useful.

**Slide 15**

Here, we emphasize the knowledge source in the RAG process. This is a critical component as it provides the necessary context and information to enhance the generated responses. By integrating a reliable and comprehensive knowledge source, such as dbt's automated documentation, we can significantly improve the quality and relevance of the AI-generated outputs.

**Slide 16**

In a 'Foreman' RAG, the only change is that the knowledge source is the dbt documentation output. This integration allows us to leverage the rich metadata and detailed data lineage provided by dbt, ensuring that the AI has access to accurate and up-to-date information. This setup enhances the AI's ability to generate precise and contextually relevant responses, making the data warehouse foreman a powerful tool for data management.

The common components of a RAG architecture include a specific type of database (the vector store), of course the Large Language Model (the LLM) and an orchestrator.

**Slide 17**

Let's zoom in on the LLM component. This is where the GPTs, the Claudes, and the Llamas enter. Each of these models has its strengths and can be chosen based on specific needs and requirements. The LLM is responsible for generating the final responses, leveraging the context and information retrieved from the knowledge source.

**Slide 18**

Next, we focus on the Vector Store component. Vector stores like pgvector, Quadrant, and Chroma are used to store and retrieve vector representations of the data. These representations are crucial for efficient and accurate retrieval of relevant information. By selecting the appropriate vector store, we can optimize the retrieval process, ensuring that the most relevant data is used for augmentation.

**Slide 19**

Finally, let's look at the Orchestrator component. Tools like Langchain and LlamaIndex are useful for managing the RAG process, but they can sometimes be redundant and overly complex for production applications. It's important to choose an orchestrator that balances functionality with simplicity, ensuring that the RAG process is efficient and manageable. This will help streamline operations and maintain system performance.

**Slide 20**

To illustrate the practical application of the 'Data Warehouse Foreman' concept, we played with a simple example. We created a simple dbt project inspired on Data Makers Fest, specifically in the information we can obtain on the website. We created our very simple yaml and 5 models or tables: attractions, partners, sessions, speakers and tutorials.

**Slide 21**

When he asked the base ‘LLM’ without any added knowledge, it answered us that Data Makers Fest is an event that celebrates data science and analytics. Ok, it makes sense, but it sounds a little generic.

**Slide 22**

If we try something a bit more specific, for example, where is it hosted, it tells us that it can vary each year, which is a bit odd, but we can give the benefit of the doubt.

**Slide 23**

If we ask, ok, so tell me the host of one of the previous editions, it tells us San Francisco, and it’s here that it starts to hallucinate. It does not seem to have specific knowledge about Data Makers Fest but it can generate answers that sound human and logical.

**Slide 24**

If we ask what type of sessions do we have, it says a lot of generic info about what almost every conference has.

**Slide 25**

In an approach like this, the Foreman would be a RAG approach where the data warehouse is being managed by dbt and the generated documentation is automatically being uploaded to the RAG knowledge base. In this case, if we ask what is Data Makers Fest we get a more specific and correct answer, also with the location, so we don’t even need to ask that follow up question.

**Slide 26**

If we ask what type of sessions, we get a more concrete answer, short and long presentation, which was the definition present in the call for speakers form that we adapted into dbt. Also, there are the tutorials which we materialized in a different table in purpose.

**Slide 27**

Something that we can gain from having the data warehouse documentation automatically used to Generative AI applications is that extremely specific questions like where are the tutorials taking place can be answered with less probability of hallucinations, and we know that they were held on the 23rd at Porto Business School.

**Slide 28**

Instead of normal question and answer chatbots, another interesting possibility to leverage the natura of the dbt docs are the text-to-sql use cases. Since every table, and column has a name, description and data type, we can ask questions and convert them into SQL.

**Slide 29**

Like, an easy one, how many sessions are there, the answer is a count from the sessions tables.

**Slide 30**

If we ask how many partners of each type we have, the LLM will search for a ‘type’ column in the partners’ table. We don’t have a ‘type’ column but we have ‘tier’, so the answer is a count and group by tier.

**Slide 31**

For more specific queries, the metadata will provide info about foreign keys. In this case if we ask if there are any speakers with more than one participation, it does a union between sessions and tutorials because it knows they are tracked separately. This is an interesting approach and a great way to leverage having a good documentation.

**Slide 32** (start of 5. Conclusions & Next Steps)

Even though what we brought to this session was a very simple example, we already learned some lessons while transitioning from these demos to production environments. And the truth is that, when we are managing a warehouse with dozens of schemas and hundreds of tables, things are not as simple as just connecting dbt documentation to our RAG.

There are multiple teams and companies that report drastic performance drops once the real-world data complexities are introduced. Databricks highlights that "Text2SQL is Not Enough," indicating that simple text-to-SQL solutions often fall short in production scenarios. Snowflake emphasizes the "Need for a semantic model" when talking about building a self-service analytics product. Numbers Station AI, a Text-to-SQL start-up, underscores the "difficulties of building enterprise-grade Text-to-SQL solutions".

**Slide 22**

However, as the science fiction writer C.J. Cherryh once wrote “Goods sit in the warehouse until information moves them.”. Just as goods remain idle in a warehouse without the right information to mobilize them, data is also in our databases. A Data Warehouse Foreman, empowered by automated documentation and generative AI, can act as a another powerful catalyst to moves data from storage to insights. With the Data Warehouse Foreman we can make our data and our documentation work harder and smarter for us.

**Slide 23**

Thank you!