**DBT ( DATA BUILD TOOL )**

🡪is an open-source command-line tool that helps organizations build, test, and maintain their data infrastructure. The tool is designed to make it easier for data analysts and engineers to work with data, in their warehouse more effectively, by providing a consistent and standardized approach to data transformation and analysis.

🡪How does dbt data build tool work?

The tool acts as an orchestration layer on top of your data warehouse to improve and accelerate your data transformation and integration process. dbt works by pushing down your code—doing all the calculations at the database level—making the entire transformation process faster, more secure, and easier to maintain.

🡪Is dbt an ETL tool?

dbt is not an ETL tool.

DBT performs the T (Transform) of ETL but it doesn't offer support for Extraction and Load operations.

🡪Does dbt work with SQL Server?

dbt connects to and runs SQL against your database, warehouse, lake, or query engine.

What is dbt ?  
🡪dbt allows you to write SQL queries without having to worry about dependencies  
🡪Write SQL code without having to duplicate sections of queries  
🡪You can split queries into containers (bits of code)  
🡪Built in SQL  
🡪Utilizes templating engines such as Jinja  
🡪Retrieve, rearrange, and organize your data using additional logic in your

SQL  
🡪Can be swiftly coded, tested, and adjusted without having to wait for it to process all your data

Who Should Use dbt ?  
🡪Automate the data transformation process,testing and deployment.  
🡪Keep track of all changes made to the underlying logic  
🡪Use version control to make it simple to trace data and update or modify the pipeline.

How dbt Works?

🡪VERSION CONTROL AND CI/CD  
Deploy safely using dev environments. Git-enabled version control enables collaboration and a return to previous states.  
   
🡪 TEST AND DOCUMENT

Test every model prior to production and share dynamically generated documentation with all the data stakeholders

🡪DEVELOP  
Write modular SQL models with SELECT statements and the ref() function- dbt handles the chore of dependency management

Why is ELT the better choice?  
🡪Agility: all the data is stored in the warehouse and readily available to use.  
🡪Simplicity: Transformations in the data warehouse are generally written in SQL, a language that the entire data team (data engineers, data scientists,  
data analyst) understands.  
🡪Self service analytics: If all of your raw data is within the warehouse, you can use BI tools to drill down from aggregated summary statistics to the raw data underlying them .  
🡪Fixing bugs: If you find errors in your transformation pipeline, you can fix the bug and re-run just the transformations to fix your data