

1. DBT (Data Build Tool) adalah alat transformasi data yang memungkinkan analis dan engineer untuk mengubah data di dalam gudang data (data warehouse) menggunakan bahasa SQL. DBT memungkinkan pengguna untuk menulis transformasi data dalam bentuk skrip SQL dan kemudian menjalankan skrip-skrip dalam urutan yang benar, serta mengelola dependensi antar transformasi. Dengan menggunakan DBT, proses ETL (Extract, Transform, Load) menjadi lebih terstruktur dan mudah dikelola, serta memungkinkan kolaborasi yang lebih baik antara tim analisis data dan tim engineering.
2. Keuntungan menggunakan DBT yaitu :
 - Transformasi Data di Gudang Data (Data Warehouse):
DBT melakukan transformasi data langsung di dalam data warehouse, yang memungkinkan pemanfaatan kekuatan pemrosesan data warehouse dan mengurangi beban pada sistem eksternal.
 - Kemudahan Penulisan dan Pemeliharaan Transformasi:
Dengan DBT, transformasi data ditulis dalam SQL yang merupakan bahasa yang banyak dikenal oleh analis data. Ini memudahkan penulisan dan pemeliharaan transformasi data.
 - Manajemen Dependensi Otomatis:
DBT mengelola dependensi antar transformasi data secara otomatis, memastikan bahwa transformasi dieksekusi dalam urutan yang benar. Ini mengurangi risiko kesalahan dan meningkatkan keandalan proses ETL.
 - Modularisasi dan Reusabilitas:
Transformasi data dipecah menjadi modul-modul kecil yang dapat digunakan kembali, memudahkan pengembangan, pemeliharaan, dan debugging.
 - Pengujian dan Validasi Data:
DBT mendukung penulisan pengujian data untuk memastikan integritas dan kualitas data. Ini membantu dalam mendeteksi dan memperbaiki masalah data secara lebih cepat.
 - Dokumentasi Otomatis:
DBT menghasilkan dokumentasi otomatis untuk semua model dan transformasi data, termasuk deskripsi tabel dan kolom, yang memudahkan pemahaman dan penggunaan data oleh seluruh tim.
 - Kolaborasi yang Lebih Baik:
Dengan menggunakan DBT, analis data dan engineer dapat bekerja bersama dalam platform yang sama, memfasilitasi kolaborasi dan komunikasi yang lebih baik.
 - Integrasi dengan Alat Orkestrasi:
DBT dapat diintegrasikan dengan alat orkestrasi seperti Apache Airflow atau DAGster, memungkinkan otomatisasi dan penjadwalan eksekusi transformasi data.
 - Peningkatan Skalabilitas:
Dengan memanfaatkan kekuatan pemrosesan data warehouse dan melakukan transformasi data di tempat, DBT membantu meningkatkan skalabilitas proses pengolahan data.
 - Open Source dan Komunitas Aktif:
DBT adalah alat open source dengan komunitas pengguna yang aktif, menyediakan dukungan, pembaruan, dan sumber daya yang melimpah untuk pengguna.
3. Dependency tree ialah struktur yang menunjukkan bagaimana berbagai model (tabel atau transformasi data) dalam proyek DBT saling bergantung satu sama lain. Setiap node dalam tree mewakili satu model, dan edges (sisi) menunjukkan dependensi antar model.

Versioning pada DBT merujuk pada praktik mengelola versi dari model, transformasi, dan proyek secara keseluruhan untuk memastikan konsistensi, mengelola perubahan, dan memfasilitasi kolaborasi.

Cara kerja versioning:

-Kontrol Versi dengan Git:

DBT mendukung penggunaan sistem kontrol versi seperti Git. Setiap perubahan pada proyek DBT (kode, konfigurasi, dokumentasi) dapat dicatat dalam repositori Git, memungkinkan pengguna untuk melacak perubahan dari waktu ke waktu.

-Branching dan Merging:

Pengguna dapat membuat cabang (branches) untuk mengembangkan fitur baru atau melakukan perubahan tanpa mengganggu cabang utama (main/master). Setelah perubahan selesai dan diuji, cabang tersebut bisa digabungkan kembali ke cabang utama.

-Tagging:

Versi tertentu dari proyek dapat diberi tag untuk menandai rilis atau checkpoint penting. Tagging memudahkan rollback ke versi sebelumnya jika diperlukan.

-CI/CD Integrasi:

DBT dapat diintegrasikan dengan pipeline CI/CD (Continuous Integration/Continuous Deployment) untuk otomatisasi pengujian, validasi, dan deploy perubahan ke lingkungan produksi.

-Dokumentasi Versi:

Setiap versi proyek DBT bisa disertai dengan dokumentasi yang menjelaskan perubahan, peningkatan, atau perbaikan yang dilakukan. Ini membantu dalam audit dan pelacakan perubahan.

Jadi, dengan dependency tree, DBT memastikan bahwa semua transformasi data dilakukan dalam urutan yang benar dan efisien. Dengan versioning, DBT memungkinkan manajemen perubahan yang aman dan terkontrol, memfasilitasi kolaborasi dan pemeliharaan proyek data.