### **1. Apa itu DBT?**

DBT (Data Build Tool) adalah alat yang memungkinkan Data Engineer untuk mengubah data di Data Warehouse dengan menulis skrip SQL. DBT dapat mengatur, menguji, dan mengelola transformasi data secara lebih efisien dan konsisten.

Beberapa komponen utama dalam DBT meliputi:

* **Models**: Skrip SQL yang mengubah data.
* **Sources**: Definisi sumber data mentah yang ada dalam gudang data.
* **Seeds**: Data statis yang dapat dimuat ke dalam gudang data untuk digunakan dalam model.
* **Snapshots**: Mengambil gambar dari data dalam waktu tertentu untuk melacak perubahan dari waktu ke waktu.
* **Tests**: Memastikan data Anda memenuhi kriteria tertentu, seperti validasi skema dan integritas data.

### **2. Apa keuntungan menggunakan DBT?**

DBT menawarkan beberapa keuntungan penting:

* **Modularitas dan Reusabilitas**: Struktur DBT memungkinkan Anda untuk membuat transformasi data yang modular dan dapat digunakan kembali.
* **Version Control**: Integrasi dengan sistem kontrol versi seperti Git memungkinkan pengelolaan perubahan dengan mudah dan melacak evolusi proyek Anda.
* **Testing**: DBT memungkinkan Anda untuk menulis tes yang memastikan data yang dihasilkan sesuai dengan harapan.
* **Dokumentasi Otomatis**: DBT dapat menghasilkan dokumentasi dari kode transformasi data Anda, membuatnya lebih mudah dipahami oleh anggota tim lain.
* **Transparansi dan Kejelasan**: Dengan menulis transformasi dalam SQL dan menggunakan kontrol versi, Anda dapat melihat dengan jelas bagaimana data diubah.
* **Kolaborasi yang Lebih Baik**: DBT mendukung pengembangan kolaboratif melalui pengelolaan versi dan dokumentasi otomatis.

### **3. Jelaskan dependency tree dan versioning pada DBT!**

#### **Dependency Tree**

Dependency tree dalam DBT adalah struktur yang menunjukkan bagaimana model saling bergantung satu sama lain. Ketika membuat model dalam DBT, membutuhkan data dari model lain sebagai input. Dependency tree memetakan hubungan ini untuk melihat urutan eksekusi model dan memastikan semua dependensi terpenuhi sebelum menjalankan suatu model.

Contoh sederhana:

* **Model A**: Mengambil data langsung dari sumber data mentah / raw data.
* **Model B**: Menggunakan data dari Model A untuk membuat transformasi lebih lanjut.
* **Model C**: Menggunakan data dari Model B.

Dalam hal ini, dependency tree akan menunjukkan bahwa Model B bergantung pada Model A, dan Model C bergantung pada Model B.

#### **Versioning**

Versioning dalam DBT dilakukan melalui integrasi dengan sistem kontrol versi seperti Git. Dengan menggunakan Git, Anda dapat:

* **Melacak perubahan**: Setiap perubahan pada skrip DBT Anda dicatat, memungkinkan Anda untuk melihat siapa yang mengubah apa dan kapan.
* **Revert ke versi sebelumnya**: Jika ada masalah dengan perubahan terbaru, Anda dapat dengan mudah kembali ke versi sebelumnya dari kode Anda.
* **Branching dan Merging**: Anda dapat bekerja pada fitur baru atau perbaikan bug di cabang terpisah tanpa mengganggu alur kerja utama. Setelah selesai, perubahan tersebut dapat digabungkan kembali ke cabang utama.
* **Kolaborasi**: Beberapa anggota tim dapat bekerja pada proyek yang sama, dan kontribusi mereka dapat digabungkan melalui Git.

Versioning membantu menjaga integritas dan stabilitas proyek DBT Anda, memungkinkan pengembangan yang lebih aman dan terstruktur.

Jika Anda memiliki pertanyaan lebih lanjut atau butuh penjelasan lebih detail, jangan ragu untuk bertanya!