

Part 1 – Introduction to Data Build Tools

1. Jelaskan apa itu DBT !
 - 1) DBT (Data Build Tool) adalah sebuah alat yang memungkinkan tim data untuk mengubah data yang disimpan dalam data warehouse dengan lebih mudah dan terstruktur.
 - 2) DBT memungkinkan pengguna untuk menulis kode transformasi data menggunakan SQL, dan kemudian menjalankan transformasi tersebut sebagai bagian dari pipeline ETL (Extract, Transform, Load).
 - 3) DBT fokus pada bagian "Transform" dari ETL, dan membantu data engineer atau analyst dalam membangun, menguji, dan memelihara pipeline transformasi data dengan efisien.
 - 4) DBT juga mendukung praktik DevOps dalam pengelolaan data, seperti version control, automated testing, dan documentation generation.
2. Apa keuntungan menggunakan DBT ?
 - 1) Kemudahan Penggunaan :
DBT memungkinkan pengguna untuk menulis transformasi data menggunakan SQL, bahasa yang familiar bagi banyak analis data.
 - 2) Modularitas :
Kode transformasi dalam DBT dapat dipecah menjadi beberapa model, yang memudahkan dalam pemeliharaan dan pengembangan.
 - 3) Testing Otomatis :
DBT mendukung pengujian otomatis terhadap transformasi data, memastikan bahwa data yang dihasilkan akurat dan sesuai dengan spesifikasi.
 - 4) Version Control :
DBT terintegrasi dengan sistem kontrol versi seperti Git, yang memudahkan dalam pelacakan perubahan dan kolaborasi dalam tim.
 - 5) Documentation :
DBT secara otomatis menghasilkan dokumentasi untuk semua model data, termasuk informasi tentang dependensi antar model, sehingga memudahkan dalam memahami pipeline data.
 - 6) Dependency Management :
DBT secara otomatis mengelola urutan eksekusi model berdasarkan dependensi antar model, sehingga tidak perlu dikhawatirkan oleh pengguna.
 - 7) Deployability :
DBT dapat diintegrasikan dengan CI/CD pipeline, memungkinkan untuk otomatisasi dalam pengembangan, testing, dan deployment.

3. Jelaskan dependency tree dan versioning pada DBT !

1) Dependency tree :

I. Dalam DBT adalah representasi dari hubungan antar model data yang ada di dalam proyek DBT. DBT memahami urutan eksekusi model berdasarkan referensi antar model yang ada dalam kode SQL.

Contoh: Jika model B bergantung pada model A, maka DBT akan memastikan bahwa model A dijalankan terlebih dahulu sebelum menjalankan model B.

II. Fungsi: Dependency tree ini memudahkan pengguna dalam memahami urutan dan hubungan antar transformasi data, serta memastikan bahwa setiap model dihasilkan dalam urutan yang benar untuk menghindari kesalahan.

III. DBT secara otomatis membangun dependency tree ini berdasarkan referensi SQL yang digunakan dalam model. Dependency tree ini juga dapat divisualisasikan, sehingga pengguna dapat melihat secara visual bagaimana model-model saling bergantung satu sama lain.

2) Versioning

I. Versioning dalam konteks DBT biasanya merujuk pada penggunaan sistem kontrol versi (seperti Git) untuk mengelola perubahan dalam proyek DBT. Setiap perubahan pada model, dokumentasi, atau konfigurasi dapat dicatat dan dilacak menggunakan Git.

II. Tim dapat bekerja secara kolaboratif tanpa konflik, karena setiap perubahan dicatat dalam repository Git.

III. Setiap perubahan dapat dengan mudah dikembalikan ke versi sebelumnya jika terjadi masalah.

IV. Praktik branching dan merging memudahkan pengelolaan pengembangan fitur baru dan integrasi perubahan ke dalam produksi. Selain itu, DBT juga mendukung konsep "versioning" pada paket DBT (DBT packages), yang memungkinkan pengguna untuk menggunakan dan mengelola dependensi antar proyek DBT, serta memastikan bahwa perubahan pada paket tidak mengganggu pipeline yang sudah ada.