Nama : Qorina M. H. Mumtaza

Kelas : DE – 1

1. Jelaskan apa itu DBT!

Jawab :

DBT (Data Build Tool) adalah command line tool yang memungkinkan data analyst dan engineer untuk membangun organisasi, menguji, dan memelihara infrastuktur data mereka di warehouse dengan efektif.

1. Apa keuntungan menggunakan DBT?

Jawab:

* Modularity and Reusability: DBT memungkinkan user menggunakan kembali logika untuk meningkatkan konsistensi dan mengurangi redundansi data.
* Version Control: DBT memungkinkan control versi transformasi data sehingga memudahkan dalam hal melacak perubahan, berkolaborasi dengan team, maupun degrade ke versi sebelumnya apabila diperlukan.
* Testing Capabilities: DBT menyediakan fungsionalitas pengujian untuk memastikan kualitas dan akurasi data.
* Incremental Builds: DBT dapat melakukan pembangunan incremental, memungkinkan pengguna untuk memperbaiki bagian tertentu dari suatu model tanpa harus membangun ulang dari awal.
* Documentation Generation: DBT secara otomatis menyimpan dokumentasi untuk transformasi data yang dilakukan pada suatu proyek.
* Collaboration and Workflow Management: DBT menyederhanakan alur kerja dan mendorong pendekatan terstruktur terhadap transformasi data.
* Cloud-Native Integration: DBT terintegrasi baik dengan gudang data cloud seperti BigQuery, Snowflake, dan Redshift, sehingga meningkatkan skalabilitas dan performanya.
* Community and Ecosystem: DBT memiliki komunitas yang berkembang, hal ini memungkinkan user untuk memanfaatkan berbagai solusi dari kontribusi komunitas.

Keunggulan tersebut secara kolektif menjadikan DBT tools yang ampuh untuk mengelola dan mentransformasikan data secara terstruktur, kolaboratif, dan efisien.

1. Jelaskan dependency tree dan versioning pada DBT!

Dependency Tree: Mewakili hubungan antara model data yang berbeda, memvisualisasikan bagaimana model bergantung satu sama lain, dengan kata lain menggambarkan aliran transformasi data. Struktur tree ini membantu pengguna memahami urutan model yang harus dibangun atau dijalankan untuk memastikan bahwa model hilir menggunakan data terbaru dari dependensinya.

Misal, Model A bergantung pada Model B dan Model B bergantung pada Model C, Dependency Tree akan menggambarkan struktur hierarki hubungan tersebut, memastikan bahwa Model C dibangun sebelum Model B, dan Model B dibangun sebelum Model A.

Versioning: Versioning di DBT mirip dengan Control Version dalam pengembangan perangkat lunak. Hal ini memungkinkan user melacak perubahan yang dilakukan pada model data dari waktu ke waktu, memudahkan user mengelola berbagai versi kode dan kembali ke versi sebelumnya apabila diperlukan. Hal ini sangat penting dalam menjaga silsilah data, memastikan reproduktifitas, dan mengaudit perubahan yang dilakukan pada jalur data.

Dengan menggabungkan Dependency Tree dan Versioning, user dapat memahami hubungan antar model, mengelola perubahan secara sistematis, dan memastikan bahwa transformasi dijalankan dalam urutan yang benar berdasarkan ketergantungan sembari mempertahankan riwayat perubahan yang terperinci untuk akuntabilitas dan reproduktifitas.