FUNDAMENTAL DATA ENGINEERING
PART 1 & 2

STUDENT : WARTADI
MENTOR : BILAL BENEFIT

PART 1 - Fundamental DE

No	Soal	Perintah/Output
1	**Apa peran utama seorang Data Engineer dalam ekosistem data? Bagaimana peran ini berbeda dari Data Scientist dan Data Analyst? Berikan beberapa contoh peran dari seorang Data Engineer yang mungkin bersinggungan atau bahkan sama dengan peran Data Scientist dan Data Analyst!**	**Peran Utama Seorang Data Engineer:** **Membangun Infrastruktur Data:** Mengembangkan dan memelihara sistem yang memungkinkan pengumpulan, penyimpanan, dan pemrosesan data. **ETL (Extract, Transform, Load) dan ELT (Extract, Load, Transform):** Menyiapkan dan pipeline data untuk memastikan data dari berbagai sumber yang dapat dikonsolidasikan dan diolah dengan efisien. **Pengolahan Data dalam Skala Besar:** Menggunakan alat dan teknologi big data untuk memproses data dalam volume besar secara efisien. **Perbedaan dengan Data Scientist dan Data Analyst:** **Data Engineer vs Data Scientist:** **Data Engineer** fokus pada pembangunan dan pemeliharaan infrastruktur data, sedangkan **Data Scientist** fokus pada analisis data dan pengembangan model prediktif atau algoritma machine learning. **Data Engineer vs Data Analyst:** **Data Engineer** bekerja lebih pada sisi teknis dan infrastruktur, sementara **Data Analyst** berfokus pada analisis data dan pelaporan untuk memberikan wawasan bisnis.

PART 1 - Fundamental DE

No	Soal	Perintah/Output
2	**Berikan beberapa contoh peran dari seorang Data Engineer yang mungkin bersinggungan atau bahkan sama dengan peran Data Scientist dan Data Analyst!**	**Contoh Peran yang Bersinggungan:** **Pengolahan Data:** **Data Engineer:** Menyiapkan raw data dari berbagai sumber untuk diolah. **Data Scientist/Data Analyst:** Mengolah dan menganalisis data yang disediakan oleh Data Engineer untuk mendapatkan insight. **Penyimpanan Data:** **Data Engineer:** Mengelola database dan sistem penyimpanan data. **Data Analyst:** Menggunakan database yang dikelola oleh Data Engineer untuk mengakses dan menganalisis data.

PART 1 - Fundamental DE

No	Soal	Perintah/Output
3	**Langkah-langkah Proses ETL dan ELT:**	**ETL (Extract, Transform, Load):** **Extract:** Mengambil data dari berbagai sumber. **Transform:** Mengolah data agar sesuai dengan kebutuhan analisis atau sistem penyimpanan. **Load:** Memuat data yang sudah diolah ke dalam sistem penyimpanan. **ELT (Extract, Load, Transform):** **Extract:** Mengambil data dari berbagai sumber. **Load:** Memuat data mentah langsung ke dalam sistem penyimpanan. **Transform:** Mengolah data dalam sistem penyimpanan sesuai kebutuhan analisis.

PART 2 - Fundamental DE

No	Soal	Perintah/Output
1	**Kapan kita harus menggunakan relational database atau NoSQL database?**	**Relational Database:** Digunakan ketika data memiliki struktur yang jelas dan relasi yang kompleks antar tabel. Contoh: Sistem manajemen pelanggan (CRM), sistem keuangan. **NoSQL Database:** Digunakan ketika data bersifat tidak terstruktur atau semi terstruktur, atau ketika kebutuhan skala dan performa tinggi diperlukan. Contoh: Media sosial, aplikasi IoT, dan big data.

PART 2 - Fundamental DE

No	Soal	Perintah/Output
2	**Perbedaan antara database, data lake, data warehouse, dan data mart:**	**Database:** Sistem penyimpanan data yang biasanya digunakan untuk aplikasi sehari-hari. Struktur data: Terstruktur (relasional atau NoSQL). **Data Lake:** Tempat penyimpanan yang menampung data dalam bentuk aslinya (mentah) dari berbagai sumber. Struktur data: Tidak terstruktur, semi-terstruktur, dan terstruktur. **Data Warehouse:** Sistem penyimpanan data yang dioptimalkan untuk query dan analisis. Struktur data: Terstruktur. **Data Mart:** Subset dari data warehouse yang berfokus pada area bisnis tertentu. Struktur data: Terstruktur.

PART 2 - Fundamental DE

No	Soal	Perintah/Output
3	**Jelaskan Normalisasi Data**	**Normalisasi database** adalah proses pengorganisasian tabel dalam sebuah database untuk mengurangi redundansi data dan memastikan integritas data. Tujuan utama dari normalisasi adalah untuk meminimalkan duplikasi data dan menghindari masalah seperti *anomaly* saat melakukan operasi *insert*, *update*, atau *delete*.

PART 2 - Fundamental DE

Normalisasi Tabel:

Tabel Asli:

employee_id	employee_name	job_code	job	city_code	city_name	province_code	province_name
1	John Smith	101	Software Engineer	201	New York	301	New York
2	Alice Johnson	102	Data Analyst	202	Los Angeles	302	California
3	Bob Davis	103	Data Engineer	203	Chicago	303	Illinois
4	Emily Wilson	101	Software Engineer	204	Houston	304	Texas
5	Michael Lee	102	Data Analyst	205	Miami	305	Florida
6	Sarah Brown	103	Data Engineer	206	Boston	306	Massachusetts
7	James Clark	101	Software Engineer	207	San Francisco	307	California
8	Laura Taylor	102	Data Analyst	208	Seattle	308	Washington
9	Daniel White	103	Data Engineer	209	Denver	309	Colorado
10	Olivia Martin	101	Software Engineer	210	Atlanta	310	Georgia

Tabel Setelah dinormalisasi

Tabel Employee:

employee_id	employee_name	job_code
1	John Smith	101
2	Alice Johnson	102
3	Bob Davis	103
4	Emily Wilson	101
5	Michael Lee	102
6	Sarah Brown	103
7	James Clark	101
8	Laura Taylor	102
9	Daniel White	103
10	Olivia Martin	101

Tabel Job:

job_code	job
101	Software Engineer
102	Data Analyst
103	Data Engineer

Tabel City:

city_code	city_name	province_code
201	New York	301
202	Los Angeles	302
203	Chicago	303
204	Houston	304
205	Miami	305
206	Boston	306
207	San Francisco	307
208	Seattle	308
209	Denver	309
210	Atlanta	310

Tabel Province:

province_code	province_name
301	New York
302	California
303	Illinois
304	Texas
305	Florida
306	Massachusetts
307	California
308	Washington
309	Colorado
310	Georgia

PART 2 - Fundamental DE

Penjelasan Normalisasi Tabel

- **Tabel Employee:**
  - Berisi informasi dasar karyawan beserta kode pekerjaan (job_code). Dengan memisahkan informasi pekerjaan ke dalam tabel Job, kita menghindari pengulangan deskripsi pekerjaan.
- **Tabel Job:**
  - Berisi kode pekerjaan dan deskripsinya. Memisahkan informasi ini membantu menghindari duplikasi dan memudahkan pemeliharaan jika ada perubahan pada deskripsi pekerjaan.
- **Tabel City:**
  - Berisi kode kota dan nama kota, serta kode provinsi (province_code). Dengan memisahkan informasi kota ke dalam tabel City, kita menghindari duplikasi nama kota.
- **Tabel Province:**
  - Berisi kode provinsi dan nama provinsi. Memisahkan informasi ini membantu menghindari duplikasi dan memudahkan pemeliharaan jika ada perubahan pada nama provinsi.

Dengan struktur tabel yang dinormalisasi, kita memastikan bahwa setiap data disimpan secara konsisten dan efisien, yang pada gilirannya meningkatkan integritas data dan memudahkan pemeliharaan serta pengelolaan data.

THANK YOU ☺