PART 2 – Fundamental DE

TASK 1

1. Kapan kita harus menggunakan relational database atau nosql database ?

   Jawaban : Pemilihan antara menggunakan database relasional atau database NoSQL sebagian besar tergantung pada kebutuhan dan karakteristik dari proyek atau aplikasi yang akan di bangun. Berikut adalah beberapa pertimbangan umum:

   - Menggunakan Database Relasional (SQL):

     Struktur Data yang Terstruktur dan Terhubung: Jika data memiliki struktur yang jelas dan relasinya kompleks, seperti data transaksional, data yang sering berubah, atau data yang memerlukan konsistensi tinggi antar tabel, maka database relasional sering kali lebih cocok.

     Konsistensi dan Integritas Data: Database relasional menawarkan dukungan yang kuat untuk transaksi ACID (Atomicity, Consistency, Isolation, Durability) yang memastikan konsistensi dan integritas data yang tinggi.

     Kemampuan Query Kompleks: Jika aplikasi kita memerlukan kemampuan untuk melakukan query kompleks yang melibatkan join antar tabel, agregasi data, dan operasi SQL lainnya, database relasional biasanya lebih mudah dikelola dalam hal ini.

     Skalabilitas Vertikal: Database relasional lebih cocok untuk skalabilitas vertikal (menambah kapasitas pada satu node server) daripada horizontal (menambah node server baru).

   - Menggunakan Database NoSQL:

     Skema Fleksibel dan Dinamis: Jika kita memiliki data yang tidak memiliki struktur yang tetap, atau struktur data yang cenderung berubah seiring waktu, NoSQL database seperti MongoDB, Cassandra, atau Redis bisa lebih cocok karena mereka menawarkan skema yang fleksibel.

     Skalabilitas Horizontal: NoSQL database umumnya dirancang untuk skalabilitas horizontal dengan baik, artinya lebih mudah menambahkan node server baru untuk meningkatkan kapasitas.

     Kinerja Tinggi dan Skala Besar: NoSQL database sering kali dapat menangani volume data yang besar dengan performa yang lebih baik, terutama untuk aplikasi dengan kebutuhan latensi rendah dan throughput tinggi.

     Model Data yang Berbeda: NoSQL database menyediakan model data yang berbeda-beda seperti dokumen (document-based), key-value, column-family, atau graph. Pemilihan tergantung pada kebutuhan aplikasi.

   Pertimbangan Tambahan:

   Ketersediaan dan Toleransi Terhadap Kesalahan: NoSQL database umumnya dirancang untuk toleransi terhadap kesalahan (fault tolerance) dan dapat menangani kegagalan node secara lebih baik daripada beberapa database relasional.

Biaya dan Kebutuhan Teknis: Selain kebutuhan fungsional, pertimbangan juga untuk biaya pengembangan, pemeliharaan, dan ketersediaan keterampilan teknis dalam tim.

Hybrid atau Polyglot Approach: Kadang-kadang kombinasi dari kedua jenis database (relasional dan NoSQL) dapat memberikan solusi terbaik untuk aplikasi yang kompleks.

Pemilihan antara database relasional atau NoSQL harus dipertimbangkan dengan cermat berdasarkan kebutuhan spesifik proyek, skala operasi, dan karakteristik data yang akan dielola.

2. Apa perbedaan antara database, data lake, data warehouse, dan data mart ?

   Jawaban : Database, data lake, data warehouse, dan data mart adalah konsep yang berbeda dalam pengelolaan dan analisis data. Berikut adalah perbedaan mendasar antara mereka:

   - ✓ Database adalah koleksi terstruktur dari data yang terorganisir dalam tabel dengan relasi yang jelas antara entitas dan atributnya.
     Karakteristik Utama: Data disimpan dalam bentuk yang terstruktur dengan skema yang telah ditentukan sebelumnya, menggunakan bahasa query (seperti SQL) untuk mengambil dan memanipulasi data.
     Tujuan: Biasanya digunakan untuk aplikasi operasional dan transaksional di mana konsistensi dan integritas data sangat penting.

   - ✓ Data lake adalah penyimpanan besar dan terdistribusi yang menyimpan data dalam format mentah (raw) atau semi-struktur tanpa memerlukan skema atau transformasi sebelum penyimpanan.
     Karakteristik Utama: Memungkinkan penyimpanan data dari berbagai sumber dalam format aslinya, termasuk data semi-struktur dan tidak terstruktur seperti teks, gambar, dan file log.
     Tujuan: Digunakan untuk analisis data mendalam, pembelajaran mesin (machine learning), dan eksplorasi data yang memerlukan akses ke berbagai jenis data dengan cepat.

   - ✓ Definisi: Data warehouse adalah database besar yang dikhususkan untuk menganalisis data bisnis dari berbagai sumber operasional.
     Karakteristik Utama: Data dalam data warehouse diambil dari sumber-sumber yang berbeda, diubah menjadi format yang lebih terstruktur, dan disimpan untuk analisis dan pelaporan bisnis.
     Tujuan: Digunakan untuk menganalisis tren bisnis, mendukung pengambilan keputusan strategis, dan melaporkan kinerja bisnis dengan menggunakan teknik OLAP (Online Analytical Processing).

   - ✓ Data mart adalah subset atau potongan dari data warehouse yang berfokus pada kumpulan data yang spesifik untuk departemen atau fungsi bisnis tertentu.
     Karakteristik Utama: Data mart memiliki skema yang sudah ditentukan dan diatur untuk mendukung kebutuhan analisis spesifik dari suatu bagian atau tim dalam organisasi.

Tujuan: Digunakan untuk memberikan akses cepat dan efisien ke data yang relevan bagi departemen atau unit bisnis tertentu tanpa perlu mengakses seluruh data warehouse.

Perbandingan Singkat:

- ✓ Database adalah koleksi data terstruktur untuk aplikasi operasional.
- ✓ Data lake adalah penyimpanan besar untuk data mentah, semi-struktur, dan tidak terstruktur.
- ✓ Data warehouse adalah database besar untuk analisis data bisnis yang terstruktur.
- ✓ Data mart adalah subset dari data warehouse yang dikhususkan untuk kebutuhan analisis departemen atau fungsi bisnis tertentu.

Pemilihan antara database, data lake, data warehouse, atau data mart tergantung pada kebutuhan analisis dan kebutuhan spesifik organisasi terhadap data yang tersedia dan jenis analisis yang ingin dilakukan.

3. Jelaskan apa itu normalisasi database, dan normalisasikan tabel dibawah !

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 Fransisco	307	California
8	Laura Taylor	102	Data Analyst	208	Seattle	308	Washington

Jawaban : Normalisasi database adalah proses desain database untuk mengurangi redudansi data dan memastikan integritas data. Tujuannya adalah meminimalkan duplikasi data, meningkatkan efisiensi penyimpanan, dan mengoptimalkan kinerja query.

Untuk normalisasi tabel di atas, kita akan mengidentifikasi entitas dan hubungan antar entitas untuk memastikan setiap tabel memiliki struktur yang terstruktur dan efisien.

Tabel Employee

Tabel ini akan berisi informasi yang berkaitan langsung dengan karyawan.

employee_id	employee_name	job_code	city_code
1	John Smith	101	201
2	Alice Johnson	102	202
3	Bob Davis	103	203
4	Emily Wilson	101	204
5	Michael Lee	102	205
6	Sarah Brown	103	206

7	James Clark	101	207
8	Laura Taylor	102	208
9	Daniel White	103	209
10	Olivia Martin	101	210

Tabel Job

Tabel ini akan berisi informasi mengenai kode pekerjaan (job_code) dan nama pekerjaan (job).

job_code	job
101	Software Engineer
102	Data Analyst
103	Bob Davis

Tabel City

Tabel ini akan berisi informasi mengenai kode kota (city_code), nama kota (city_name), kode provinsi (province_code), dan nama provinsi (province_name).

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

Dengan melakukan normalisasi seperti di atas, kita memisahkan data menjadi beberapa tabel terkait yang mengurangi redundansi informasi dan memastikan data tersimpan dengan cara yang lebih efisien dan terstruktur.