Database Design

Introduction

- Perancangan basis data adalah proses menciptakan solusi penyimpanan data yang terstruktur dan efisien.
- Basis data yang dirancang dengan baik sangat penting untuk integritas data dan kinerja aplikasi.
- Perancangan basis yang baik adalah memperhatikan prinsip-prinsip desain basis data yang efektif

Langkah dalam Perancangan Basis Data

1. **Analisis Persyaratan**: Memahami data dan hubungannya.
2. **Diagram Entitas-Hubungan (ERD)**: Membuat representasi visual entitas dan hubungannya.
3. **Normalisasi**: Mengorganisir data ke dalam tabel yang efisien.
4. **Hubungan Tabel**: Mendefinisikan bagaimana tabel saling berhubungan.
5. **Indeks dan Kunci**: Memilih primary key dan foreign key.
6. **Integritas Data**: Menerapkan aturan untuk menjaga kualitas data.
7. **Optimisasi Kinerja**: Memastikan pengambilan dan penyimpanan data yang efisien.

Analisis Persyaratan

- Memahami data: Data apa yang perlu disimpan?
- Mengidentifikasi entitas: Objek atau konsep utama apa yang ada?
- Mendefinisikan hubungan: Bagaimana entitas-entitas tersebut terhubung?
- Pertimbangkan volume data: Seberapa banyak data yang akan disimpan?
- Pikirkan pola penggunaan: Bagaimana data akan diambil dan diperbarui?

Diagram Entitas-Hubungan (ERD)

- ERD adalah representasi visual dari model data.
- Entitas direpresentasikan sebagai tabel, dan hubungan ditampilkan sebagai garis yang menghubungkan tabel.
- Kunci primer dan foreign key diindikasikan.
- Membantu memvisualisasikan struktur basis data.

Normalisasi

- Normalisasi adalah proses mengorganisir data secara efisien ke dalam tabel.
- Mengurangi redundansi data dan memastikan integritas data.
- Mengikuti bentuk normal, seperti 1NF, 2NF, dan 3NF, untuk menghilangkan anomali data.
- **Tujuan:** Meningkatkan efisiensi penyimpanan dan memudahkan pemeliharaan database

Normalisasi: Bentuk Normal Pertama (1NF)

Definisi: Tabel harus memiliki:

- Bentuk relasional
- Nilai atomik (tidak bisa dibagi lagi)

Normalisasi: Bentuk Normal Pertama (1NF)

Contoh:

ID_Pesanan	Nama_Pelanggan	Produk
1	Andi	Pensil, Penghapus

Menjadi

ID_Pesanan	Nama_Pelanggan	Produk
1	Andi	Pensil
1	Andi	Penghapus

Normalisasi: Bentuk Normal Kedua (2NF)

Definisi: Harus dalam 1NF dan semua atribut non-kunci harus sepenuhnya bergantung pada primary key.

Contoh:

ID_Pesanan	Nama_Pelanggan	Produk	Harga
1	Andi	Pensil	5000
1	Andi	Penghapus	3000

Normalisasi: Bentuk Normal Kedua (2NF)

Tabel Pesanan

ID_Pesanan	Nama_Pelanggan
1	Andi

Tabel Detail Pesanan

ID_DetailPesanan	ID_Pesanan	Produk	Harga
1	1	Pensil	5000
2	1	Penghapus	3000

Normalisasi: Bentuk Normal Ketiga (3NF)

Definisi: Harus dalam 2NF dan tidak ada atribut non-kunci yang bergantung transitif pada primary key.

Normalisasi: Bentuk Normal Ketiga (3NF)

Tabel Pesanan

ID_Pesanan	Nama_Pelanggan	Alamat
1	Andi	Jl. Merdeka No.5

Tabel Pelanggan

Nama_Pelanggan	Alamat
Andi	Jl. Merdeka No.5

Menghapus **Alamat** dari **Tabel Pesanan** karena bergantung pada **Nama_Pelanggan**, bukan pada **ID_Pesanan**.

Hubungan Tabel

- Tentukan bagaimana tabel berhubungan satu sama lain.
- Gunakan primary key dan foreign key untuk mendirikan koneksi.
- Hubungan umum meliputi satu-satu, satu-ke-banyak, dan banyak-ke-banyak.
- Hubungan memastikan konsistensi data.

Indeks dan Kunci

- Pilih primary key untuk setiap tabel.
- Gunakan foreign key untuk menghubungkan tabel.
- Indeks meningkatkan kinerja kueri dengan membuat struktur pencarian cepat.
- Pilih kolom-kolom yang akan diindeks berdasarkan pola kueri.

Integritas Data

- Terapkan aturan integritas data dengan `constraints`.
- Gunakan `constraints` untuk mencegah masukan data yang tidak valid.
- `Constraints` umum meliputi `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, dan `CHECK`.

Optimisasi Kinerja

- Perancangan untuk efisiensi dalam pengambilan dan penyimpanan data.
- Pertimbangkan penggunaan denormalisasi untuk operasi yang sering dibaca.
- Optimisasi kueri SQL untuk kecepatan.

Jenis Data dalam Desain Basis Data

- Jenis data menentukan tipe nilai yang dapat disimpan dalam kolom.
- Beberapa jenis data umum dalam desain basis data termasuk:
  - Teks (VARCHAR, CHAR)
  - Angka (INT, FLOAT)
  - Tanggal (DATE, TIME)
  - Boolean (BOOL)
  - Binary (BLOB)

Hubungan Antar Entitas

- Selain jenis data, hubungan antara entitas dalam desain basis data juga sangat penting.
- Populer disebut sebagai Entity Relationship.
- Hubungan antara entitas dapat berupa:
  - one-to-one
  - one-to-many
  - many-to-many

Contoh Hubungan Antar Entitas

- Contoh hubungan antar entitas one-to-one:
  - Satu pengguna memiliki satu profil.
  - Satu profil memiliki satu pengguna.
- Contoh hubungan antar entitas one-to-many:
  - Satu penulis memiliki banyak buku.
  - Satu buku memiliki satu penulis.
- Contoh hubungan antar entitas many-to-many:
  - Satu produk memiliki banyak kategori.
  - Satu kategori memiliki banyak produk.

Latihan

- Buatlah desain basis data untuk aplikasi toko buku online.
- Entitas yang perlu dibuat meliputi:
  - Buku
  - Penulis
  - Penerbit
  - Kategori
- Tentukan hubungan antar entitas.
- Tentukan atribut untuk setiap entitas.
- Tentukan tipe data untuk setiap atribut.

Latihan

- Buat desain basis data untuk aplikasi pemutar musik.
- Entitas yang perlu dibuat meliputi:
  - Lagu
  - Album
  - Artis
  - Genre
- Tentukan hubungan antar entitas.
- Tentukan atribut untuk setiap entitas.
- Tentukan tipe data untuk setiap atribut.

Latihan

- Buat desain basis data untuk aplikasi belanja.
- Entitas yang perlu dibuat meliputi:
  - Produk
  - Keranjang Belanja
  - Pesanan
- Tentukan hubungan antar entitas.
- Tentukan atribut untuk setiap entitas.
- Tentukan tipe data untuk setiap atribut.
- Tentukan kunci utama dan kunci asing.
- Tentukan aturan integritas data (constraint).

Database naming best practices

- Use plural nouns for table names, e.g. `users`, `books`, `categories`
- Using `snake_case` for table names
- Using `snake_case` for column names
- Using `id` as the primary key for all tables
- Using `{table_name}_id` as the foreign key for all tables
- Using `snake_case` for index names
- Keep table and column names concise