Contoh Tabel yang Tidak Ter-Normalisasi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_buku | judul | pengarang | penerbit | harga | stock | Tanggal terbit | katagori |
| 1 | Pemrograman Python | John Doe | PT. Maju Mundur | 100000 | 50 | 2023-01-01 | Programming |
| 2 | Novel Romantis | Jane Smith | PT. Cahaya Ilmu | 80000 | 30 | 2022-12-15 | Fiction |
| 3 | Algoritma dan Struktur Data | John Doe | PT. Maju Mundur | 120000 | 20 | 2023-02-10 | Programming |
| 4 | Novel Misteri | Mark Brown | PT. Cahaya Ilmu | 90000 | 15 | 2023-03-05 | Fiction |

Dari tabel di atas, kita bisa melihat beberapa masalah yang muncul akibat tidak diterapkannya normalisasi:

* **Data Pengarang dan Penerbit Berulang:** John Doe dan PT. Maju Mundur muncul dua kali, begitu juga dengan Jane Smith dan PT. Cahaya Ilmu. Hal ini menyebabkan redundansi data dan boros penyimpanan.
* **Kesulitan dalam Memperbarui Data:** Jika ingin mengubah alamat kantor PT. Maju Mundur, kita harus mengubah data tersebut pada setiap baris yang terkait dengan penerbit tersebut.
* **Risiko Ketidakkonsistenan Data:** Jika ada kesalahan saat memperbarui data, misalnya salah mengetik nama pengarang, kesalahan tersebut akan menyebar ke semua baris yang terkait.

**Contoh Tabel Setelah Dinormalisasi:**

Untuk mengatasi masalah di atas, kita bagi tabel menjadi empat tabel yang lebih kecil:

**Tabel buku**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_buku | judul | Id\_pengarang | Id\_penerbit | harga | stock | Tanggal terbit | Id\_katagori |
| 1 | Pemrograman Python | 1 | 1 | 100000 | 50 | 2023-01-01 | 1 |
| 2 | Novel Romantis | 2 | 2 | 80000 | 30 | 2022-12-15 | 2 |
| 3 | Algoritma dan Struktur Data | 1 | 1 | 120000 | 20 | 2023-02-10 | 1 |
| 4 | Novel Misteri | 2 | 2 | 90000 | 15 | 2023-03-05 | Fiction |

**Tabel Pengarang**

|  |  |
| --- | --- |
| Id\_pengarang | Nama Pengarang |
| 1 | Jhon Doe |
| 2 | Jane Smith |
| 3 | Mark Brown |

**Tabel Penerbit**

|  |  |
| --- | --- |
| Id\_penerbit | Nama Penerbit |
| 1 | PT Maju Mundur |
| 2 | PT Cahaya Ilmu |

**Tabel Katagori**

|  |  |
| --- | --- |
| Id\_katagori | Katagori |
| 1 | Programming |
| 2 | Fiction |

Bentuk Normal Pertama (1NF):

Menjamin bahwa setiap sel dalam tabel hanya berisi satu nilai, tidak ada nilai yang dapat dibagi.

1. Pastikan setiap kolom hanya berisi nilai atomik (tidak dapat dibagi).
2. Identifikasi dan atasi duplikasi data.
3. Pastikan setiap baris dapat diidentifikasi secara unik dengan menggunakan primary key.

Bentuk Normal Kedua (2NF)

Menghilangkan ketergantungan parsial, yaitu setiap kolom yang bukan bagian dari primary key bergantung sepenuhnya pada seluruh primary key.

1. Pastikan tabel sudah dalam 1NF.
2. Identifikasi kolom yang hanya bergantung pada sebagian primary key.
3. Pindahkan kolom-kolom tersebut ke dalam tabel terpisah, yang memiliki primary key yang sesuai.

Bentuk Normal Ketiga (3NF):

Menghilangkan ketergantungan transitif, yaitu setiap kolom yang bukan primary key tidak boleh bergantung pada kolom non-primary key lainnya

1. Pastikan tabel sudah dalam 2NF.
2. Identifikasi ketergantungan transitif dan pindahkan kolom-kolom tersebut ke dalam tabel terpisah dengan primary key yang sesuai.

Ketergantungan parsial adalah :

Ketergantungan parsial terjadi ketika atribut non-kunci (non-primary key) dalam sebuah tabel bergantung hanya pada sebagian dari kunci primer (primary key).

Contoh: Misalkan kita memiliki tabel dengan kunci primer (A, B) dan atribut non-kunci C. Jika C bergantung pada A saja dan tidak pada keduanya (A, B), itu disebut ketergantungan parsial.

ketergantungan transitif adalah:

Ketergantungan transitif terjadi ketika sebuah atribut non-kunci bergantung pada atribut non-kunci lainnya, bukan pada kunci primer langsung.

Contoh: Misalkan kita memiliki tabel dengan kunci primer (A, B) dan atribut non-kunci C. Jika terdapat ketergantungan C → B di mana C bergantung pada B, yang pada gilirannya bergantung pada kunci primer (A, B), itu disebut ketergantungan transitif.

Contoh :

|  |  |  |
| --- | --- | --- |
| A | B | C |
| A1 | B1 | C1 |
| A2 | B2 | C2 |
| A3 | B3 | C3 |

Ketergantungan Parsial:

Jika C hanya bergantung pada A tanpa memperhatikan nilai B, itu adalah ketergantungan parsial, misalnya, C → A.

Ketergantungan trasitif:

Jika terdapat ketergantungan C → B, yang kemudian B → A, itu adalah ketergantungan transitif karena C sebenarnya bergantung pada A melalui B.