Rakamin Academy

id/x partners

Project Based Internship

Data Warehouse Partitioning

Vertical Partitioning

Daftar Isi

Exploration Source

[Article]

- Vertical Partitioning vs Horizontal Partitioning - GeeksforGeeks
- Data Partitioning Techniques in System Design
- Data partitioning guidance - Azure Architecture Center | Microsoft Learn

[Video]

- Horizontal vs Vertical Database Partitioning

A. *Introduction*

Pada artikel ini, kita akan lebih fokus ke *vertical partitioning*. Sederhananya, *vertical partitioning* merupakan teknik partisi yang menekankan pembagian tabel berdasarkan **kolom**. Ini berguna ketika Anda memiliki tabel dengan banyak kolom, dan tidak semua kolom sering diakses bersama-sama.

B. Implementasi Proses *Vertical Partitioning*

Sama halnya dengan *horizontal partitioning,* teknik *vertical partitioning* juga memiliki pendekatan ketika kapankah waktu yang tepat untuk menerapkan *vertical partitioning*. Berikut adalah beberapa situasi di mana *vertical partitioning* dapat menjadi pilihan yang tepat :

1. Kolom dengan Tingkat Akses yang Berbeda

   Jika beberapa kolom data sering diakses, sementara yang lain jarang diakses, *vertical partitioning* memungkinkan penempatan kolom yang sering digunakan dalam satu tabel kecil, meningkatkan efisiensi akses.

2. Normalisasi Data

   Jika prinsip normalisasi diinginkan untuk meminimalkan redundansi dan menjaga integritas data, *vertical partitioning* dapat digunakan untuk membuat tabel-tabel yang lebih kecil dengan fokus pada informasi yang spesifik.

3. Keamanan Data

   Jika perlu memisahkan data sensitif dari data yang tidak sensitif, *vertical partitioning* dapat membantu dalam meningkatkan tingkat keamanan data dengan menempatkan kolom-kolom sensitif dalam satu tabel terpisah.

4. Struktur Tabel yang Kompleks

Jika tabel memiliki struktur yang kompleks dan memiliki kolom-kolom yang jarang dibutuhkan dalam query yang sering dilakukan, *vertical partitioning* dapat memfasilitasi manajemen dan pemrosesan data yang lebih efisien.

5. Optimasi Kinerja Spesifik Query

   Jika beberapa query spesifik sering dijalankan dan hanya membutuhkan sebagian kecil dari kolom data, *vertical partitioning* dapat mengoptimalkan kinerja query dengan memisahkan kolom-kolom tersebut.

Vertical partitioning dapat memberikan manfaat khususnya dalam situasi di mana peningkatan efisiensi akses dan manajemen data lebih diutamakan daripada mempertahankan data dalam satu tabel besar.

Studi Kasus: *Vertical Partitioning* dalam Bank Digital

Client Anda adalah sebuah *bank* digital mengelola tabel transaksi yang besar dengan beberapa kolom yang jarang dibutuhkan dalam query umum. Untuk meningkatkan efisiensi akses dan manajemen data, *bank* ini memutuskan untuk menerapkan *vertical partitioning* pada tabel transaksi mereka.

SOLUSI

1. **Query Pembentukan Tabel Transaksi Awal**

```
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY,
    AccountNumber INT,
    TransactionDate DATE,
    Amount DECIMAL(15, 2),
```

```
    TransactionType VARCHAR(20),
    CustomerName VARCHAR(50),
    BranchCode VARCHAR(10),
    AdditionalInfo TEXT
);
```

Penjelasan :

- Tabel **Transactions** dibuat dengan semua kolom yang relevan untuk menyimpan data transaksi.

2. **Query Pembentukan Tabel Vertical Partitioning**

```
CREATE TABLE CompactTransactions (
    TransactionID INT PRIMARY KEY,
    AccountNumber INT,
    TransactionDate DATE,
    Amount DECIMAL(15, 2),
    TransactionType VARCHAR(20)
);

CREATE TABLE ExtendedTransactions (
    TransactionID INT PRIMARY KEY,
    CustomerName VARCHAR(50),
    BranchCode VARCHAR(10),
    AdditionalInfo TEXT
);
```

Penjelasan :

- Tabel **CompactTransactions** hanya berisi kolom-kolom yang sering digunakan dalam query umum.
- Tabel **ExtendedTransactions** berisi kolom-kolom yang jarang dibutuhkan dalam query umum.

3. **Query Contoh Pemilihan Data Gabungan dari Kedua Tabel**

```
SELECT C.TransactionID, C.AccountNumber, C.TransactionDate, C.Amount, C.TransactionType, E.CustomerName, E.BranchCode
FROM CompactTransactions C
JOIN ExtendedTransactions E ON C.TransactionID = E.TransactionID
WHERE C.TransactionType = 'Deposit';
```

Penjelasan :

- Query ini menggabungkan data yang diperlukan dari kedua tabel untuk memberikan informasi lengkap.
- *Vertical partitioning* memungkinkan seleksi data yang lebih efisien tergantung pada kebutuhan query.

4. **Query Pembaruan Informasi Pelanggan**

```
UPDATE ExtendedTransactions
SET CustomerName = 'John Doe'
WHERE TransactionID = 12345;
```

Penjelasan :

- Dengan *vertical partitioning*, pembaruan informasi pelanggan hanya memengaruhi tabel yang bersangkutan, tidak mempengaruhi tabel lain yang tidak terpengaruh oleh perubahan ini.

Melalui penerapan *vertical partitioning*, *bank* digital ini berhasil meningkatkan efisiensi akses data, meminimalkan redundansi, dan memfasilitasi manajemen data yang lebih fleksibel.

REFERENCE

How does vertical partitioning affect the performance and scalability of a DBMS?

Vertical Partitioning in System Design

What are horizontal and vertical partitions in database and what is the difference?