

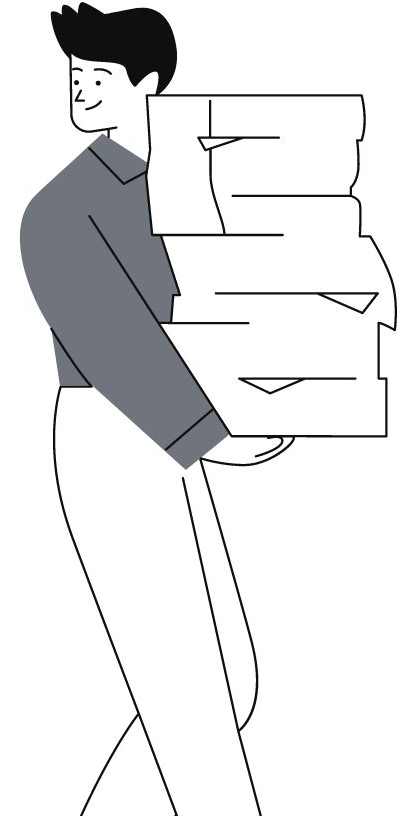


Universitas Bosowa
Teknologi Informasi

Abdillah.S.A.S.,S.Kom.,M.Pd.

Teknologi Database

Mengenal Database
Partitioning
(Sharding Method)



Perkenalan

Abdillah.S.A.S.,S.Kom.,M.Pd. bisa di panggil Pak **Abdhi** ,Bisa menjadi dosen dan teman sharing yang baik.

Lebih Lanjut Follow



Abdillah.S.A.S.,S.Kom.,M.Pd.



085242179993



Abdhi Griffindors



@abdhi



@abdhi_tux



abdhisas.com

Mengenal Database Partitioning

Universitas Bosowa
Teknologi Informasi



Database Partitioning atau Partisi Database merupakan sebuah Teknik untuk memecah belah table ke dalam beberapa segmen (partisi atau subpartisi)



Mengapa Partisi Database Dibutuhkan ?



- Digunakan pada pengelolaan VLDB (Very Large Database)
- VLDB yaitu database dengan ukuran 10 GB – 10 TB
- Agar proses modifikasi data tidak terhambat dengan banyaknya data
- Untuk mengelolah data menjadi kelompok-kelompok



Metode Partisi Database



1. Horizontal Partitioning (Sharding)

Horizontal Partitioning atau dikenal dengan istilah Sharding adalah suatu konsep partisi dengan membagi data dari satu table menjadi beberapa tabel dengan jumlah kolom yang sama tetapi jumlah record-nya lebih sedikit.

Partisi ini bertujuan untuk mempercepat pencarian data suatu tabel karena pada saat mencari data, jumlah baris yang di scan lebih sedikit sehingga waktu yang digunakan akan lebih singkat.

Catatan :

- Saat melakukan partisi perlu dinilai jumlah baris di table baru harus sama*
- Jika nilai yang rentangnya digunakan untuk sharding tidak dipilih dengan hati-hati, maka skema partisi akan menyebabkan server menjadi tidak seimbang.*



Metode Partisi Database



2. Vertical Partitioning

Vertical Partitioning adalah suatu konsep partisi dengan membagi data dari suatu tabel menjadi beberapa table dengan jumlah record yang sama dengan jumlah record pada table semula, namun jumlah kolomnya lebih sedikit.

Catatan :

- a. *Skenario ideal untuk jenis partisi ini adalah ketika anda tidak membutuhkan semua informasi tentang satu table dalam kueri anda*
- b. *Sistem partisi ini membagi data menjadi lebih spesifik*
- c. *Ketika aplikasi update, database mungkin memerlukan partisi lebih lanjut.*
- d. *Sistem partisi ini mudah diterapkan dan beresiko rendah pada aplikasi.*



Metode Partisi Database



3. Directory – Based Partitioning

Directory – Based Partitioning adalah suatu konsep partisi dengan membagi data berbasis direktori/berkas/folder.

Catatan :

- a. *Sistem partisi ini memungkinkan layanan pencarian untuk mengetahui skema partisi dan memisahkannya dari kode akses database.*
- b. *Mengizinkan penambahan server database atau perubahan skema partisi tanpa mempengaruhi aplikasi*
- c. *Partisi jenis ini mudah menjadi titik kegagalan.*



Ilustrasi Partisi Database



Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

1

Original Table (Tabel Asli)

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	BAGCAN
4	JIM	PEPPER

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	GREEN
3	PURPLE
4	AUBERGINE

Horizontal Partitions

HP1

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN

HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

2

Vertical Partition (Partisi Vertical)

3

Horizontal Partition (Partisi Horizontal/Sharding)

Metode Penerapan Partisi Pada Tabel

- 1 Range Partitioning
- 2 List Partitioning
- 3 Hash Partitioning
- 4 Composite Partitioning

Metode Penerapan Partisi Pada Tabel



1. Range Partition

Pada range partition, data dikelompokkan berdasarkan range (rentang) nilai yang kita tentukan. Range partition ini cocok digunakan pada kolom yang nilainya terdistribusi secara merata.

Contoh : Kolom Tanggal



Metode Penerapan Partisi Pada Tabel



2. List Partition

Pada list partition, data dikelompokkan berdasarkan nilainya. Cocok untuk kolom yang variasi nilainya tidak banyak.

Contoh : Kolom Kota/Daerah



Metode Penerapan Partisi Pada Tabel



3. Hash Partition

Jika ingin melakukan partisi namun tidak cocok dengan RANGE ataupun LIST maka kita bisa menggunakan HASH partition. Penentuan “nilai mana di taruh di partisi mana” itu diatur berdasarkan hash value.

Contoh : Kolom Unik (No Invoice, NIM, dll)



Metode Penerapan Partisi Pada Tabel



4. Composite Partition

Merupakan penerapan metode partisi dengan menggabungkan beberapa metode sebelumnya seperti range-list atau range-hash.



Masalah Umum Pada Sharding



- **Join dan Denormalization (Penggabungan dan Denormalisasi)**

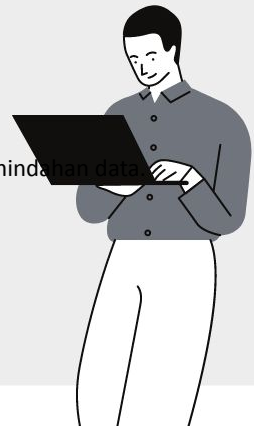
- Gabungan tidak akan menghasilkan kinerja yang efisien karena data harus dikompilasi dari beberapa server.
- Solusi: denormalisasi database sehingga kueri dapat dilakukan dari satu tabel. Namun hal ini dapat menyebabkan ketidakkonsistenan data.

- **Referential integrity (Integritas referensial)**

- Sulit untuk menerapkan batasan integritas data (mis. Foreign Keys).
- Solusi 1. Integritas referensial ditegakkan oleh kode aplikasi., 2. Aplikasi dapat menjalankan pekerjaan SQL untuk membersihkan referensi yang menggantung.

- **Rebalancing (Penyeimbangan Kembali)**

- Perlunya penyeimbangan kembali
- 1. Distribusi data tidak seragam.
- 2. Banyak beban pada satu pecahan.
- Membuat lebih banyak pecahan DB atau menyeimbangkan kembali pecahan yang ada akan mengubah skema partisi dan memerlukan pemindahan data.



Kesimpulan



Sharding bisa menjadi solusi yang bagus bagi mereka yang ingin menskalakan database mereka secara horizontal.

Namun, ini juga menambah banyak kerumitan dan menciptakan lebih banyak potensi titik kegagalan untuk aplikasi Anda.

Sharding mungkin diperlukan untuk beberapa orang, tetapi waktu dan sumber daya yang dibutuhkan untuk membuat dan memelihara arsitektur sharding dapat lebih besar daripada manfaatnya bagi orang lain.



Ada Pertanyaan ?

Ayo Diskusi, dan
Belajar Bersama.

Are you ready?

Terima Kasih

