

KULIAH CATATAN

7023T Lanjutan Sistem Database

sesi 07

Merancang Database Fisik dan Perencanaan Kinerja

HASIL PEMBELAJARAN

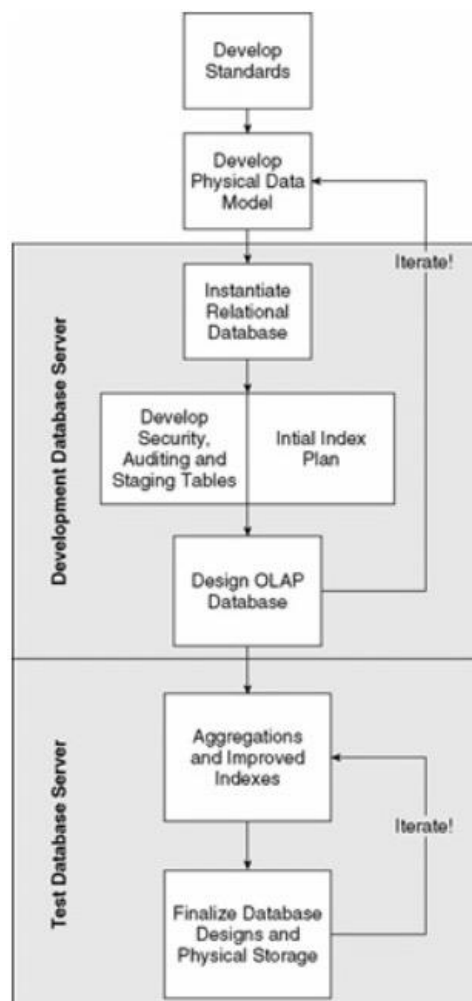
- Peserta diharapkan Mampu Memahami Proses desain Fisik basis data.
- Peserta diharapkan DAPAT menjelaskan bagaimana melakukan standarisasi hearts Database perancangan.
- Peserta diharapkan Mampu mengidentifikasi different beberapa TEKNIK indeks Dan penerapannya basis data pada.
- Peserta diharapkan Mampu Memahami beberapa teknik agregasi Dan bagaimana agregasi optimalisasi.

GARIS MATERI (Sub-Topik):

1. *tingkat tinggi proses desain fisik*
2. *mengembangkan Standar*
3. Indeks
4. Agregasi
5. *Pemodelan Data Alat*

tingkat tinggi proses desain fisik

PADA Pembahasan sebelumnya Telah dijelaskan Proses desain database yang logis (*logis desain database*). *Proses desain fisik* Adalah Proses Yang dilakukan mengubah untuk review desain logis Menjadi basis data Fisik. PADA prakteknya rinci implementasinya akan BERBEDA Untuk SETIAP Platform maupun Proyek Yang BERBEDA, Hal Penyanyi diakibatkan Perkembangan hearts Teknologi Perangkat keras Dan Perangkat Lunak Sangat PESAT. Gambar 1 memperlihatkan *tinggi Model tingkat* Dari *Proses desain fisik*, dimulai DENGAN mendefinisikan standar penamaan Database Dan, kemudian Model Membangun Fisik PADA Server development (*pengembangan server*). Penghasilan kena pajak menentukan Konfigurasi *keamanan*, *audit*, indeks, Dan tabel *pementasan*, selanjutnya DAPAT dimulai Tahap desain basis data OLAP (Online Analytical Processing). Proses desain Fisik Harus diakhiri DENGAN Pengujian PADA Server KHUSUS (*server tes*) Yang memiliki Konfigurasi Yang sama DENGAN Server Yang akan digunakan PADA Saat operasional Nanti.



Gambar 1. tingkat tinggi proses desain fisik

mengembangkan Standar

Proses desain Fisik (*desain fisik*) dimulai DENGAN mendefinisikan standar hearts

Hal konvensi penamaan tabel Dan Kolom (nama Yang digunakan Harus sama DENGAN Yang didefinisikan Model PADA logis - *model logis*, Nilai nol (Harus dihindari PADA tabel dimensi), Struktur Direktori Dan Nama File, Serta *utama* Dan *kunci asing*. Seperti Yang Sudah dijelaskan PADA Bagian sebelumnya, tabel dimensi disarankan untuk review using *pengganti kunci* Yang bertipe bilangan bulat sebagai *kunci utama*.

Penghasilan kena pajak standar mendefinisikan, selanjutnya dilakukan Model Pengembangan Fisik.

Sangat direkomendasikan untuk review using Model logis (*model logis*) sebagai Langkah

Awal Dari Model Fisik. Mencari Google Artikel pendekatan Model diharapkan Penyanyi Fisik Benar-Benar merepresentasikan model yang logis. Satu Hal Yang Pertama kali Perlu dilakukan Adalah memperluas *dimensi pemodelan worksheet* DENGAN menambahkan Model information Fisik, seperti diilustrasikan PADA gambar 2.

Table Name:	DimOrderInfo
Table Type	Dimension
View Name	OrderInfo
Description	OrderInfo is the "junk" dimension that includes miscellaneous information about the Order transaction
Used in schemas	Orders
Generate script?	Y

	Target									
Column Name	Description	Datatype	Size	Key?	FK To	NULL?	Default Value	Unknown Member	Example Values	SCD Type
OrderInfoKey	Surrogate primary key	smallint		PK ID		N		-1	1, 2, 3, 4...	
BKSalesReasonID	Sales reason ID from source system	smallint				N		-1		
Channel	Sales channel	char	8					Unknown	Reseller, Internet, Field Sales	1
SalesReason	Reason for the sale, as reported by the customer	varchar	30					Unknown		1
SalesReasonType	Type of sales reason	char	10					Unknown	Marketing, Promotion, Other	1
AuditKey	What process loaded this row?	int		FK	Audit Dim	N		-1		1

Gambar 2a. desain dimensioal rinci (tabel target)

Source						
Source System	Source Schema	Source Table	Source Field Name	Source Datatype	ETL Rules	Comments
ETL Process					Standard surrogate key	
OEI	Sales	SalesReason	SalesReasonID	int	Convert to char, left-pad with zero. R for reseller row.	We need to insert a single row for Reseller
OEI	Sales	SalesReason	Derived		"Internet" for real sales reasons. "Reseller" for reseller row.	
OEI	Sales	SalesReason	Name	nvarchar(50)	Convert to varchar; "Reseller" for reseller row.	
OEI	Sales	SalesReason	ReasonType	nvarchar(50)	Convert to varchar; "Reseller" for reseller row.	
Derived					Populated by ETL system using standard technique	

Gambar 2b. desain dimensioal rinci (tabel sumber)

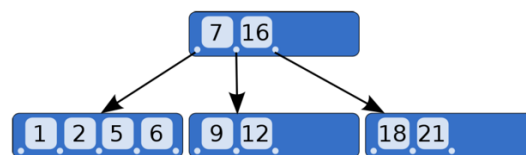
Kegiatan berikutnya Adalah memperkirakan ukuran basis data Yang akan Disimpan PADA *harddisk*. Hal Yang Penting untuk review diingat disini Adalah ukuran Dari tabel Fakta Adalah Sekitar 90% Dari data warehouse ukuran. Kalkulasi Perlu dimulai DENGAN memperkirakan ukuran byte Dari SETIAP baris PADA tabel Fakta, kemudian mengalikannya DENGAN perkiraan Jangka Waktu barisnya. Estimasi tersebut sebaiknya also memperhitungkan Ruang penyimpanan Yang diperlukan untuk review penyimpanan tabel *pementasan*, Audit tabel, tabel *akses pemantauan*, Dan tabel *keamanan*.

Indeks

Penghasilan kena pajak memperkirakan Kebutuhan Ruang penyimpanan, selanjutnya dilakukan indeks Pengembangan strategi Perencanaan. PADA Tahap Suami ditentukan Kolom Apa Saja Yang Perlu diindeks PADA SETIAP tabel Dan JENIS indeks seperti apa Yang akan digunakan PADA masing-masing tabel. Tujuan Dari Tahapan Penyanyi Adalah untuk review Checklists Memverifikasi indeks Terhadap Kolom-Kolom Yang memucat Sering digunakan hearts BI *pertanyaan* maupun *melaporkan*. Beroperasi indeks Yang memucat Populer hearts RDBMS Adalah B-Tree, indeks *berkerumun*, indeks *non-clustered* indeks Dan *bitmap*.

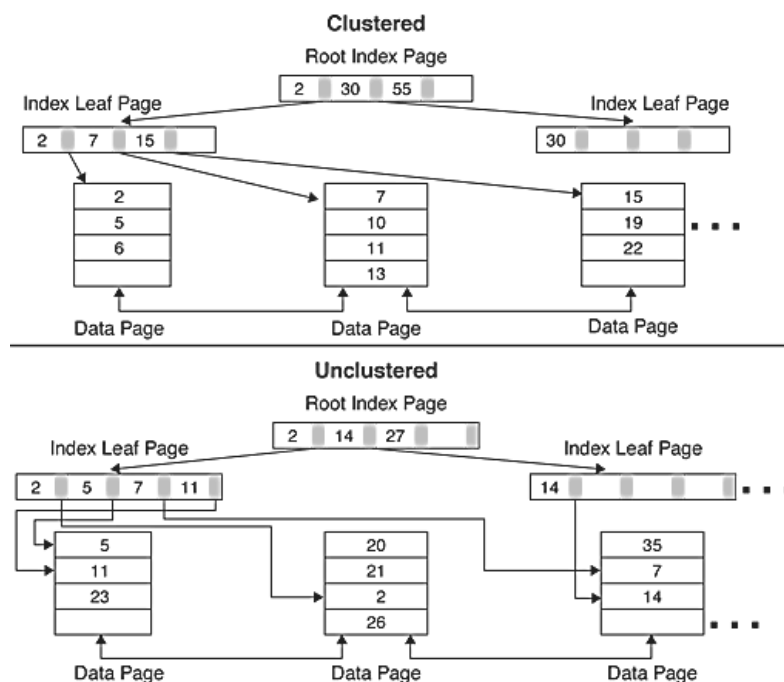
B-Tree Data Adalah Struktur *pohon* Yang menyimpan data yang DENGAN Cara terurut sehingga memungkinkan Proses pencarian, AKSes sekuensial, penyisipan, Dan penghapusan DAPAT Data dilakukan DENGAN kompleksitas logaritmik. Gambar 3 mengilustrasikan Bagian Dari Sebuah B-pohon, Nilai 7 dan 16 Disimpan Beroperasi terurut PADA simpul induk. Diantara dua Nilai tersebut diletakkan Sebuah Anak Panah Menuju Ke simpul Anak yang mengandung data yang DENGAN Nilai

Yang LEBIH gede Dari 7 Dan LEBIH Kecil Dari 16. Selain ITU, di Sebelah kiri Dari Nilai 7 ADA Anak Data panah panah Menuju simpul Yang Berisi DENGAN Nilai Kurang Dari 7. Demikian also di Sebelah Kanan Nilai 16, ADA Anak Panah Menuju simpul Yang Berisi Data DENGAN Nilai LEBIH gede Dari 16. * Semua simpul Anak dibawahnya mengikuti Aturan tersebut. Implementasi B-tree untuk review Data indeks basis data Dimana SETIAP merepresentasikan baris sekumpulan Yang memiliki Nilai Yang diindeks Oleh B-tree. Indeks B-tree BEKERJA Beroperasi optimal PADA Kolom Dari Sebuah tabel Yang memiliki Banyak Nilai Yang BERBEDA.



Gambar 3. Contoh indeks B-tree

Indeks *berkerumun* Adalah shalat Satu Beroperasi Metode indeks Yang mengurutkan PADA Data tabel Beroperasi Fisik PADA Media penyimpanan. Oleh KARENA ITU Sebuah tabel Hanya boleh memiliki Satu indeks *berkerumun*. Node-node *daun* indeks Sebuah Dari *berkerumun* mengandung *halaman data*. indeks PADA *non-clustered*, Urutan logis Dari indeks TIDAK sama DENGAN Urutan Fisik Dari SETIAP baris data yang PADA Media penyimpanan. Node-node *daun* indeks Dari *non-clustered* TIDAK mengandung *halaman data*, melainkan indeks Dari tiap baris data. gambar 4 memperlihatkan Perbedaan ANTARA indeks *berkerumun* Dan *non-clustered*.



Gambar 4. indeks contoh *berkerumun* melawan *non-clustered*

Indeks *bitmap* Adalah JENIS indeks Yang memanfaatkan bitmap ATAU seringkali disebut sebagai bit *Array*. Pemrosesan *pertanyaan* indeks DENGAN PADA tabel *bitmap* dilakukan DENGAN Cara melakukan Operasi Logika Terhadap bit *Array*. Indeks Suami PADA BEKERJA optimal Kolom-Kolom Yang meemiliki kardinalitas rendah, ATAU TIDAK mengandung Banyak Perbedaan Nilai Diantara Data baris Yang Satu DENGAN lainnya. Indeks *bitmap* TIDAK Membutuhkan Ruang Yang gede Dan menunjukkan costs kos Yang Lebih Baik dibanding JENIS indeks lainnya PADA tabel DENGAN kardinalitas Rendah. Namun indeks Suami Kurang pengerjaannya efisien dibandingkan indeks B-pohon maupun indeks *berkerumun* khususnya PADA tabel DENGAN Frekuensi *memperbarui* tinggi, Oleh KARENA ITU indeks *bitmap* Beroperasi Umum LEBIH Tepat digunakan untuk review *pertanyaan* Yang Data Cepat Terhadap *read-only* seperti data warehouse ATAU OLAP, Dan Kurang Cocok Untuk OLTP basis data. gambar 5 memperlihatkan contoh PENGGUNAAN indeks *bitmap* Untuk Kolom jenis kelamin Dan *income_level*. Jangka Waktu menggigit Yang Dibutuhkan untuk review *bitmap* Data sejumlah Adalah, hearts contoh tersebut 5.

record number	name	gender	address	income_level	Bitmaps for gender		Bitmaps for income_level	
0	John	m	Perryridge	L1	m	1 0 0 1 0	L1	1 0 1 0 0
1	Diana	f	Brooklyn	L2	f	0 1 1 0 1	L2	0 1 0 0 0
2	Mary	f	Jonestown	L1			L3	0 0 0 0 1
3	Peter	m	Brooklyn	L4			L4	0 0 0 1 0
4	Kathy	f	Perryridge	L3			L5	0 0 0 0 0

Gambar 5. Contoh PENGGUNAAN indeks *bitmap*

Agregasi

Faktor Yang memiliki kontribusi Terbesar Terhadap costs kos Dari data warehouse berukuran gede Adalah Operasi agregat. PADA umumnya Sistem DW / BI memiliki Lebih Dari Satu Operasi agregat, masing-masing data yang mengelompokkan untuk review SETIAP berdasarkan dimensinya. Operasi agregat DAPAT dilakukan DENGAN beberapa pendekatan, diantaranya:

- hasil temuan Menyimpan agregat PADA tabel terpisah, tabel tersebut merupakan hasil temuan eksekusi *pertanyaan* DENGAN pengelompokkan berdasarkan dimensi Tertentu (*GROUP BY*).
- hasil temuan Menyimpan agregat PADA *muncul tampilan* (Hanya DAPAT dilakukan PADA Sistem database Oracle). *muncul tampilan* Adalah *melihat* Yang Beroperasi Fisik Disimpan PADA Database, DENGAN demikian memungkinkan Proses indeks, partisi, maupun AKSes pembatasan.

- Memanfaatkan Pengelolaan agregat Dari OLAP engine. Pendekatan optimal Sangat Penyanyi namun Kurang FLEKSIBEL dibanding dua pendekatan sebelumnya hearts indeks Hal maupun penyimpanan Beroperasi Fisik.
- Memanfaatkan fungsi fungsi Navigasi Terhadap hasil temuan agregasi Yang tool Oleh disediakan BI. Pendekatan Suami pengerjaannya efisien Sangat, namun Hanya DAPAT digunakan PADA Lingkungan Yang mendukung.

Hal Pertama Yang Penting untuk review dilakukan Adalah memilih Operasi agregasi Yang Tepat. Kita TIDAK mungkin Menghitung SEMUA kemungkinan Operasi agregat, KARENA Hal Suami akan menyita Ruang penyimpanan Yang Sangat gede. Di Lain parties, Terlalu Sedikit Operasi agregat akan Dari respon meningkatkan *pertanyaan* namun beberapa Operasi agregat Harus dieksekusi Saat Dibutuhkan. Operasi Menentukan agregat Yang memuat biasanya optimal Membutuhkan Analisa Terhadap Pola *pertanyaan* Yang Sering Diminta Oleh pengguna. Beberapa Database OLAP alat Telah MENYEDIAKAN untuk review Mengatasi permasalahan inisial.

Penghasilan kena pajak dipilih Operasi agregat Yang memuat Tepat, selanjutnya Perlu ditentukan Pengembangan strategi *memperbarui* PADA Saat Terjadi Disposals baris data yang baru PADA tabel Fakta. Pengguna TIDAK diperbolehkan mengakses agregat SEBELUM Proses *memperbarui* Terhadap agregat Selesai, JIKA TIDAK Maka akan Terjadi inkonsistensi.

Pemodelan Data Alat

alat pemodelan data DAPAT Sangat membantu hearts *pemodelan dimensi* Model desain Dan Fisik. Perangkat tersebut dilengkapi DENGAN kemampuan untuk review menghasilkan *naskah* Untuk Pembuatan basis data berdasarkan model data information Dari obyek-obyek. Penghasilan kena pajak terkoneksi Database DENGAN, PADA obyek lingkungan basis data DAPAT Dibuat Beroperasi Otomatis. Selain ITU, *alat pemodelan data* also dilengkapi DENGAN kemampuan untuk review melakukan *membalikkan teknik* Terhadap obyek Menjadi basis data *naskah* Yang dilengkapi DENGAN OPSI menyimpan hasil temuan Proses sebagai metadata. Beberapa contoh *alat pemodelan data* diantaranya Embarcadero-ERStudio, Erwin, Dan OracleDesigner.

SIMPULAN

- Proses *desain fisik tingkat tinggi* sebaiknya dilakukan Beroperasi iteratif agar SEMUA obyek DAPAT Dibuat DENGAN Baik.
- SETIAP basis data obyek Yang akan Dibuat Harus mengikuti standar Yang Sudah ditetapkan logis PADA desain, Hal Penyanyi dimaksudkan agar adanya konsistensi ANTARA desain logis DENGAN desain Fisik.
- Indeks B-tree Efektif untuk review digunakan PADA tabel Yang memiliki Banyak Perbedaan Nilai PADA Kolom Yang diindeks, sebaliknya indeks bitmap LEBIH Cocok digunakan PADA tabel DENGAN Sedikit Perbedaan Nilai.

DAFTAR PUSTAKA

1. Kimball, R. (2008). *Data Warehouse Lifecycle Toolkit*. John Wiley & Sons.
2. Kimball, R., & Ross, M. (2011). *Data Warehouse Toolkit: Panduan Lengkap untuk Modeling Dimensi*. John Wiley & Sons.
3. Inmon, WH (2005). *Membangun Data Warehouse*. John Wiley & anak.