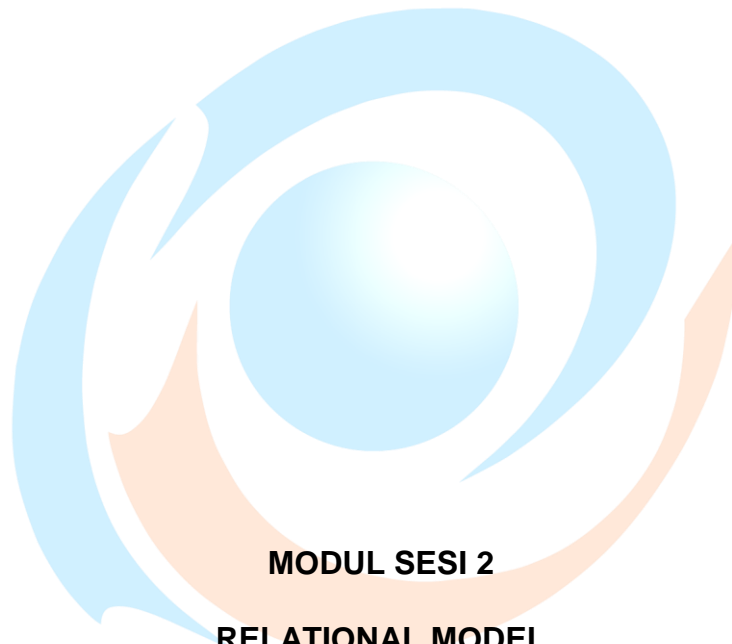




MODUL
DATABASE
(CCS120)



MODUL SESI 2
RELATIONAL MODEL

Universitas
Esa Unggul

DISUSUN OLEH
NOVIANDI, S.Kom, M.Kom

UNIVERSITAS ESA UNGGUL

2020

BAB II

MODEL DATA RELATIONAL

Tujuan

1. Menjelaskan tujuan tiga level arsitektur database
2. Menjelaskan content dari level eksternal, konseptual dan internal
3. Menjelaskan mapping dari eksternal/konseptual dan konseptual/internal
4. Menjelaskan konsep model data relational
5. Menjelaskan data model
6. Menjelaskan model entity-relationship
7. Menjelaskan model relational

Teori

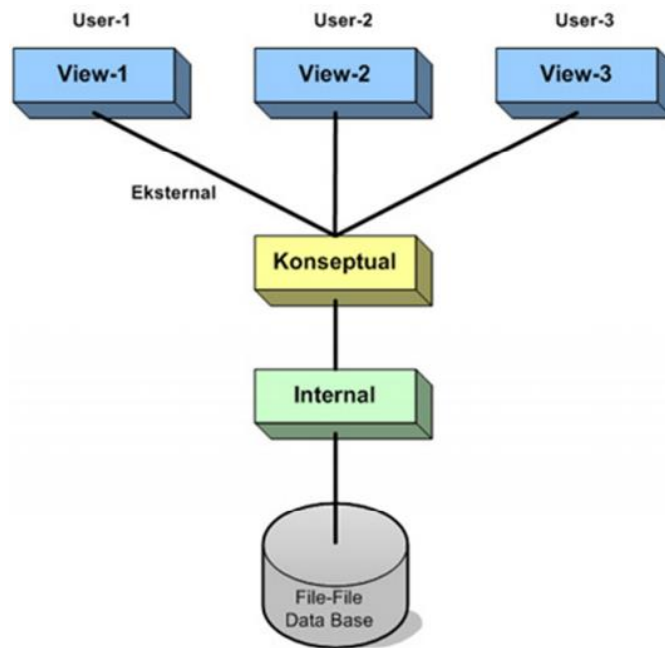
Tujuan utama database system adalah menyediakan user melalui suatu pandangan abstrak mengenai data, dengan menyembunyikan detail dari bagaimana data disimpan dan dimanipulasi.

Tiga level arsitektur database menurut ANSI/SPARC memiliki tujuan yaitu:

1. Memungkinkan *user* mengakses data yang sama dengan pandangan yang berbeda
2. Menyembunyikan rincian penyimpanan fisik dari *user*

Tiga level arsitektur database menurut ANSI/SPARC, yaitu:

1. Level internal
 - a. Level terdekat dengan penyimpanan fisik
 - b. Memperhatikan cara data disimpan secara fisik
2. Level konseptual
Perwujudan dari seluruh informasi database
3. Level eksternal
Memperhatikan cara bagaimana data dilihat oleh *user individual*



Gambar 2.1 Tiga level arsitektur database

Level internal

- Merupakan perwujudan level rendah dari seluruh basis data.
- Berkenaan dengan penyimpanan secara fisik.
- Digambarkan dengan skema internal yang mendefinisikan beragam jenis record tersimpan dan spesifikasi index apa yang ada.

Level konseptual

- Disebut juga level logis atau level logika komunitas
- Merupakan perwujudan dari seluruh isi informasi basis data
- Data ditampilkan dalam bentuk skema konseptual
- Skema konseptual menampilkan pandangan data yang menjelaskan simpanan data dalam database dan relasi data.

Level eksternal

- Level pemakai individual
- Pemakai bisa berupa pembuat program aplikasi atau pemakai akhir
- Pada level ini data ditampilkan sesuai dengan kebutuhan individu

Mapping

Dalam arsitektur tiga level melibatkan pemetaan yang berfungsi mendefinisikan hubungan antara skema eksternal dan skema konseptual, skema konseptual dan skema internal. Pemetaan merupakan kunci dari ketentuan kemandirian data logis dan fisik.

Terdapat dua pemetaan yakni:

1. Pemetaan konseptual/internal
2. Pemetaan eksternal/konseptua

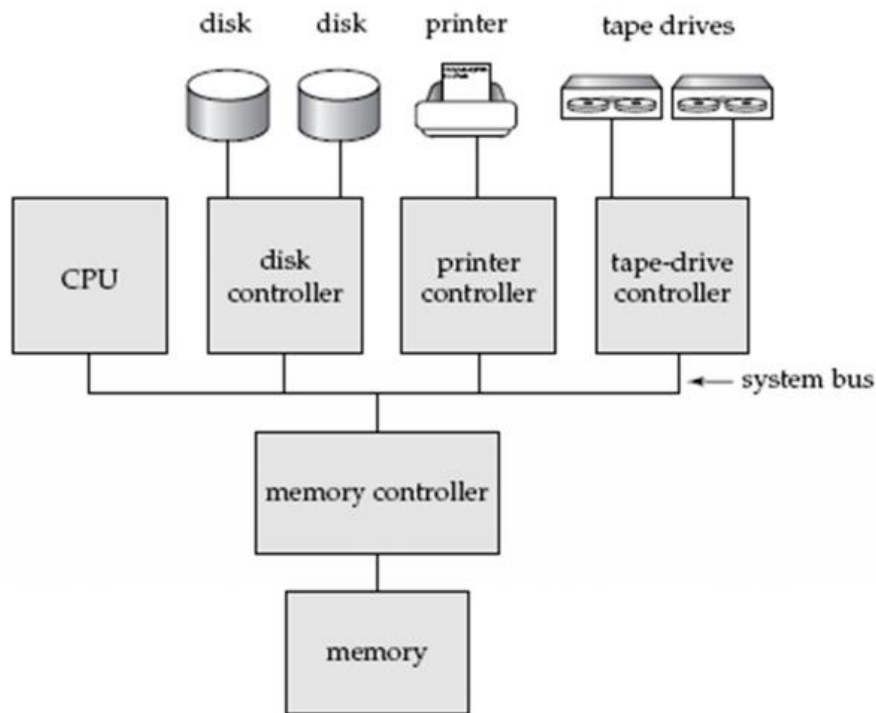
Pemetaan konseptual/internal

- Mendefinisikan kesesuaian antara pandangan konseptual dan basisdata tersimpan.
- Serta mendefinisikan bagaimana record dan field diwujudkan pada level internal
- Pemetaan ini harus tetap invarian antara struktur basis data fisik dengan skema konseptual.
- Sehingga ketika struktur basis data fisik berubah maka skema konseptual juga berubah
- Merupakan kunci kemandirian data fisik

Arsitektur Database System

1. Sistem terpusat
 - Berjalan pada sistem komputer tunggal dan tidak berinteraksi dengan sistem komputer yang lain.
 - Sistem komputer multi-guna: satu ke banyak CPU dan beberapa alat pengendali yang terhubung melalui sebuah bus yang memungkinkan akses pembagian memori.
 - Sistem single-user (mis, PC atau workstation): unit desk-top, single user, biasanya hanya terdiri satu CPU dengan satu atau dua hard disk; SO hanya mendukung untuk satu user.
 - Sistem Multi-user: lebih banyak penyimpanan, memori lebih besar, multiple CPU, dan menggunakan SO multi-user. Melayani banyak pengguna

yang terhubung dengan sistem melalui terminal (dumb-terminal). Biasanya disebut sistem server.



Gambar 2.2 Sistem Terpusat

Keuntungan system terpusat adalah:

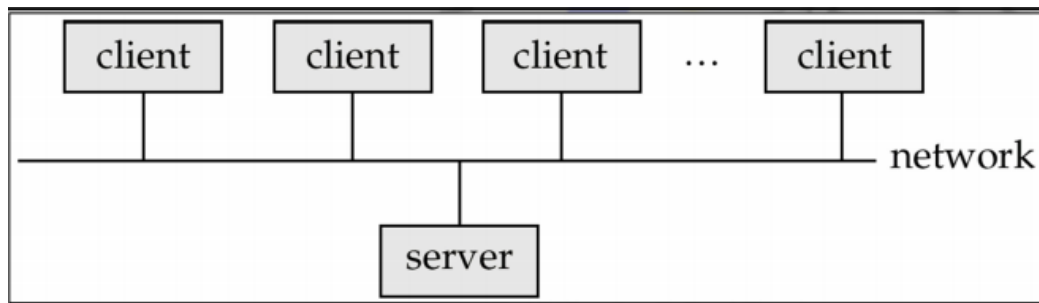
- Standarisasi tercapai
- Kemampuan mengakses semua informasi dalam satu lokasi
- Keamanan lebih terjamin
- Kemudahan dalam maintenance

Kelemahan system terpusat adalah:

- Beban kinerja server meningkat
- Kerusakan terhadap sistem akan mengakibatkan berhentinya seluruh sistem.

2. Sistem Client-Server

Sistem *server* melayani permintaan m sistem *client*, yang mempunyai struktur umum seperti berikut:



Gambar 2.3 *Client-Server*

Server

- Merupakan DBMS itu sendiri
- Mendukung semua fungsi dasar DBMS
- Menyediakan dukungan eksternal, konseptual, dan internal
- “server” dalam konteks ini hanyalah nama lain dari Database Management System.

Arsitektur Sistem Server

Dibagi menjadi dua, yaitu:

1. Transaction Server

- Disebut juga Query-Server
- Menyediakan interface tiap client untuk mengirim permintaan query

2. Data Server

Memungkinkan user melakukan pembacaan, pengubahan data

Transaction Server

Berisi proses-proses sebagai berikut:

- Server process
- Lock manager process
- Database writer process
- Log writer process
- Check point process
- Process monitor

Client

- Merupakan beragam aplikasi yang bekerja diatas DBMS
- Baik aplikasi built-in atau yang ditulis pemakai
- Aplikasi ini yang akan terhubung ke server dan digunakan user untuk mengambil atau mengelola data dalam basis data.

Keuntungan Sistem Client-Server

- Beban pemrosesan tersebar antara client dan server
- DBMS mencapai kinerja tinggi karena hanya didedikasikan untuk transaksi pengolahan
- Aplikasi client dapat mengambil keuntungan penuh dengan interface yang canggih seperti graphical user interface.

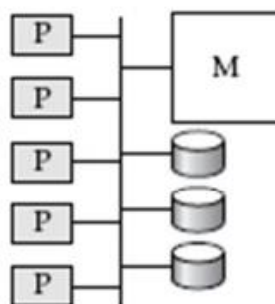
Keuntungan Sistem Client-Server

- Implementasi lebih kompleks karena harus menangani middleware dan jaringan
- Beban DBMS bertambah untuk menangani masalah concurency.

3. Sistem parallel

a. *Shared memory*

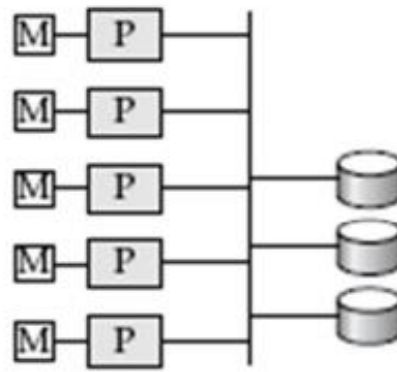
Prosesor membagi memory



(a) shared memory

b. *Shared disk*

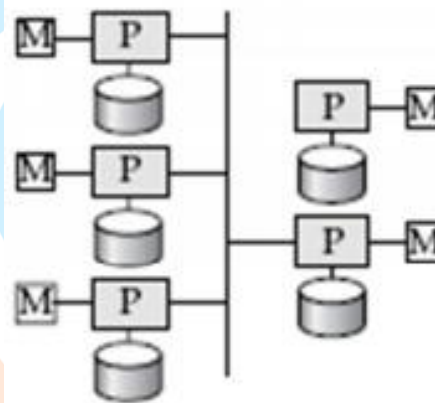
- Prosesor membagi sebagian kapasitas disk
- Sistem dengan shared disk tersebut sering disebut cluster



(b) shared disk

c. *Shared nothing*

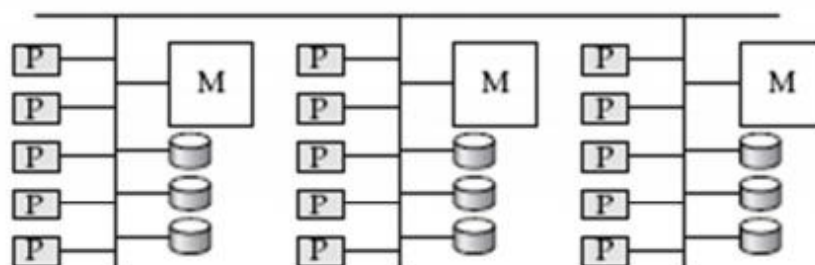
Prosesor membagi bukan memori dan bukan disk



(c) shared nothing

d. *Hierarchical*

Gabungan berbagai arsitektur



(d) hierarchical

Keuntungan Sistem Paralel

- Pemrosesan yang sangat cepat
- Meningkatkan reliabilitas: Tidak tergantung dengan kerusakan sistem yang lain
- Kinerja meningkat dan kapasitas yang besar

Keuntungan Sistem Paralel

- Biaya yang meningkat
- Membutuhkan sumber daya yang besar untuk mendukung pemrosesan yang paralel
- Pengelolaan menjadi lebih rumit

4. Sistem terdistribusi

- Data tersebar di beberapa mesin
- Jaringan terinterkoneksi dengan mesin
- Data digunakan secara bersama-sama oleh beberapa pengguna pada beberapa mesin
- Dalam sistem tersebar server dan client dijalankan pada mesin yang berbeda
- Pada sistem ini client bisa memiliki datanya sendiri dan server bisa memiliki aplikasinya sendiri.
- Setiap mesin dapat bertindak sebagai server untuk kebanyakan pemakai.
- Dalam sistem tersebar setiap mesin dapat memiliki DBMS, sistem operasi, dan jaringan komunikasi yang berbeda

Kelebihan Sistem Terdistribusi

- Menggambarkan struktur organisasi
- Meningkatkan reliabilitas
- Meningkatkan performance
- Lebih ekonomis karena dapat menggunakan banyak komputer berukuran kecil.

Kelemahan Sistem Terdistribusi

- Lebih rumit dalam pemeliharaan dibanding sistem terpusat
- Terdapat banyak titik untuk masuk ke sistem database sehingga lebih rawan terhadap masalah keamanan
- Kesulitan dalam menjaga integritas data
- Desain database menjadi lebih kompleks

Model Relasional

Model relasional diusulkan oleh *Codd* pada tahun 1970. Model relasional saat ini adalah model hierarki dan model jaringan. Saat ini model relasional menjadi model data yang dominan dan mendasar pada produk andalan system manajemen basis data (SMBD). Model relasional digunakan luas oleh berbagai vendor seperti, Informix, Oracle, Sybase, Access, SQL Server, FoxBase, Paradox dan lain-lain.

Database merupakan kumpulan dari satu atau lebih relasi. Masing – masing relasi merupakan tabel yang terdiri dari baris dan kolom. Keuntungannya jika dibandingkan dengan model data sebelumnya adalah representasi datanya sederhana dan mudah diekspresikan dalam bentuk query.

Apa itu Data Model?

Model data adalah notasi untuk menggambarkan data atau informasi.

Deskripsi umum model data dilihat dari tiga bagian:

1. Struktur data
 - Cara penyimpanan, penyusunan dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien.
 - Dalam teknik pemrograman, struktur data berarti tata letak data yang berisi kolom-kolom data, baik itu kolom yang tampak oleh pengguna (*user*) ataupun kolom yang hanya digunakan untuk keperluan pemrograman yang tidak tampak oleh pengguna
2. Operasi pada data
 - Dalam bahasa pemrograman, operasi pada data umumnya adalah segala sesuatu yang dapat diprogram.

- Dalam model data basis data, biasanya ada serangkaian operasi terbatas yang dapat dilakukan.

Misalnya:

- Operasi yang dapat memanggil informasi kembali
- Operasi yang dapat merubah database

3. *Constraint* pada data

- *Constraint* digunakan untuk membatasi jenis data yang bisa masuk ke tabel.
- Memastikan akurasi dan keandalan data dalam tabel.

Jenis model data

1. Model data File datar(Flat-file data model)

- Terdiri dari satu array, dua dimensi elemen data, dimana semua anggota kolom yang diberikan diasumsikan nilai yang sama, dan semua anggota baris diasumsikan terkait satu sama lain.

Misalnya:

- Kolom untuk nama dan kata sandi yang dapat digunakan sebagai bagian dari basis data keamanan sistem.
- Setiap baris akan memiliki kata sandi spesifik yang terkait dengan pengguna individu.
- Kolom tabel sering memiliki tipe yang terkait dengannya, mendefinisikannya sebagai data karakter, informasi tanggal atau waktu, bilangan bulat, atau angka floating point.
- Format tabel ini adalah pendahulu untuk model relasional.

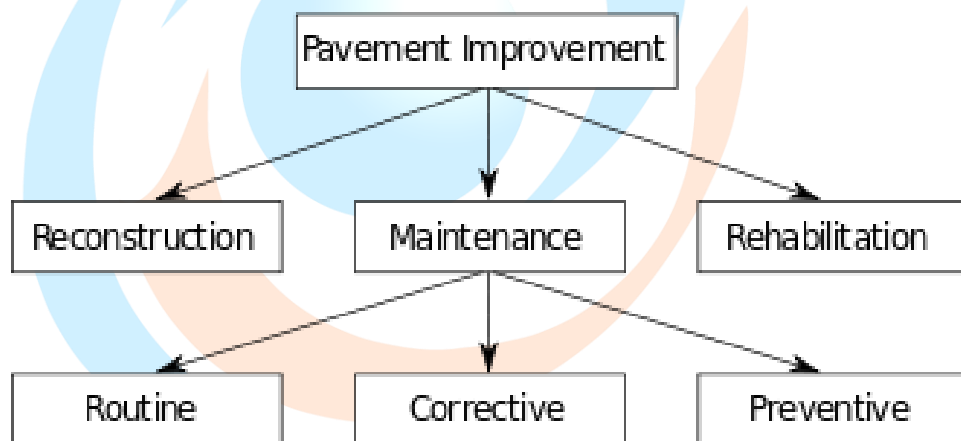
Flat File Model

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	Patching
Record 3	SR-301	33	Crack seal

Gambar 2.4 Flat File Model

2. Model data Hirarki(Hierarchichaldata model)

- Dalam model hierarkis, data diorganisasikan ke dalam struktur mirip pohon, yang menyiratkan satu induk tunggal untuk setiap record.
- Field semacam menyimpan record dalam urutan tertentu.
- Struktur hierarkis banyak digunakan dalam sistem manajemen basis data mainframe awal, Seperti Sistem Manajemen Informasi (IMS) oleh IBM, dan sekarang menggambarkan struktur dokumen XML.
- Struktur Hirarki memungkinkan hubungan satu-ke-banyak antara dua jenis data.
- Struktur Hirarki sangat efisien untuk menggambarkan banyak hubungan di dunia nyata; *recipes, table of contents, ordering of paragraphs/verses*, dan segala informasi bersarang yang disortir

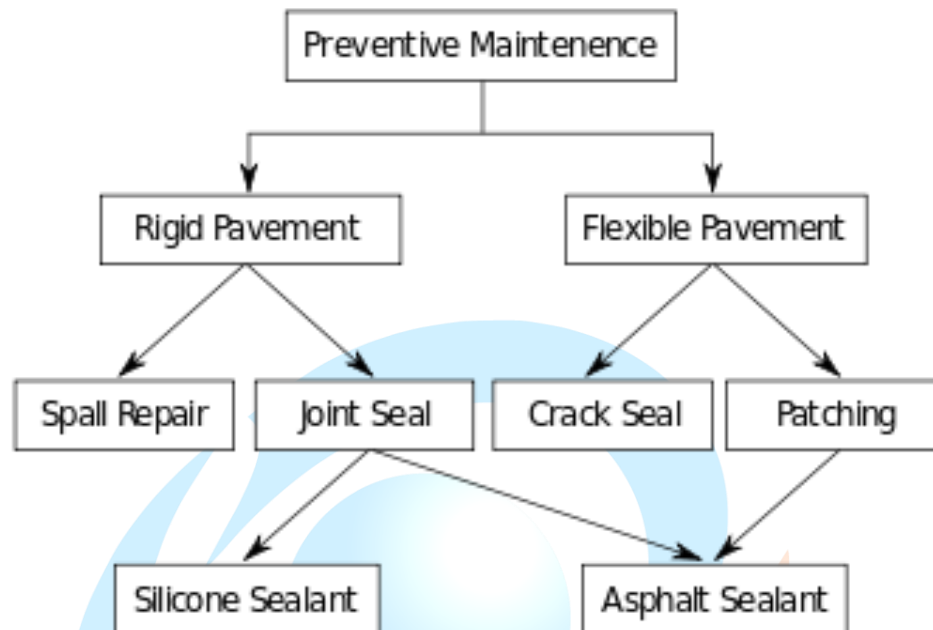


Gambar 2.5 Gambar Hierarki Model

3. Model data Jaringan(Network data model)

- Model jaringan memperluas struktur hierarki, memungkinkan hubungan banyak ke banyak dalam struktur seperti pohon yang memungkinkan *multi parent*
- Model database yang diyakini sebagai cara fleksibel mewakili objek dan hubungan mereka.
- Model data jaringan memiliki fitur istimewa yang pada skema, diperlihatkan sebagai grafik dengan tipe objek ialah node, tipe hubungannya ialah kurva, yang tidak terbatas dengan menjadi hierarki atau berkisi.

- Model jaringan mengatur data menggunakan dua konsep dasar, yang disebut record (records berisikan fields) dan set (mendefinisikan one-to-many relationships diantara records).

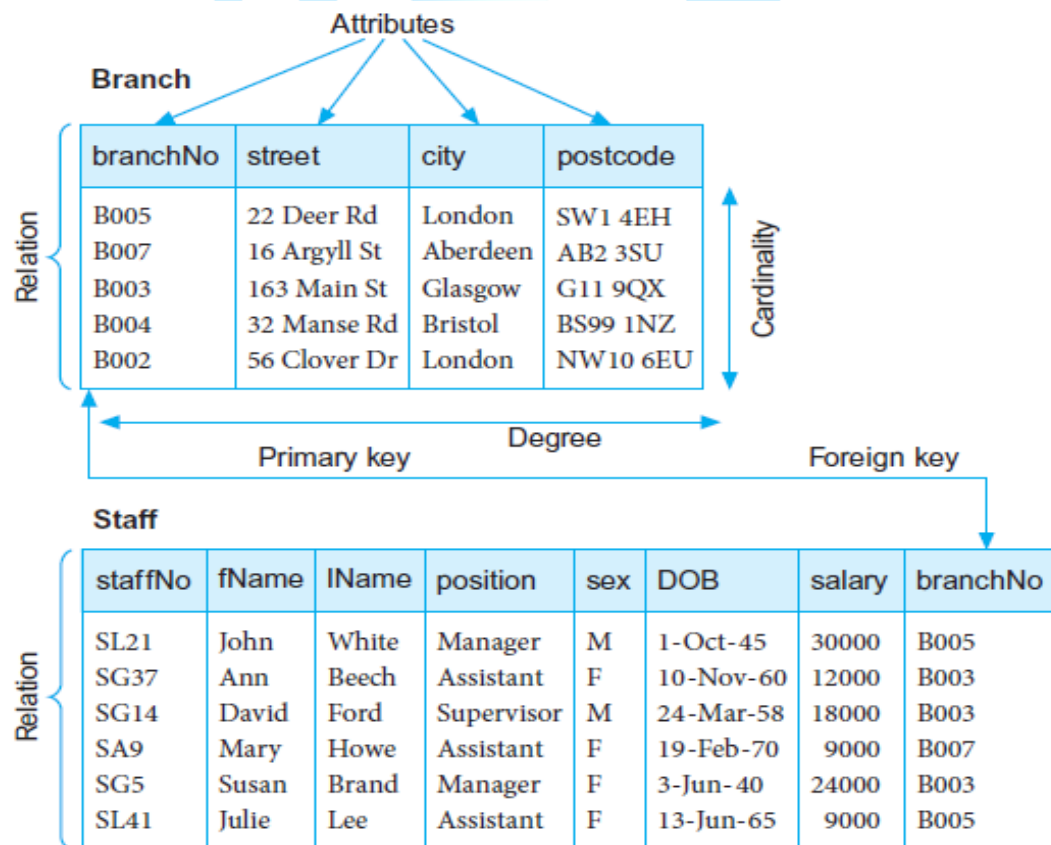


Gambar 2.6 Model Jaringan

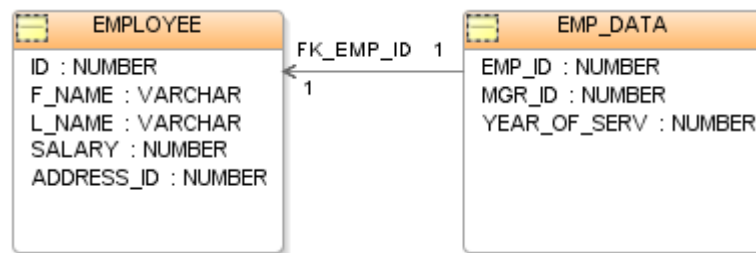
4. Model data Relasional(Relational data model)

- Model yang dikenalkan oleh E.F. Codd pada tahun 1970, sebagai cara untuk membuat system manajemen database yang lebih independent dari aplikasi tertentu.
- Secara umum disebut sebagai database relasional
- Model relasional memberi cara untuk merepresentasikan data: sebagai tabel dua dimensi yang disebut dengan relasi.
- Dalam model database relasional terdiri atas beberapa struktur, yaitu:
 - Relasi
Relasi adalah table dengan kolom dan rows
 - Atribut
Kolom dari relasi
 - Tuples
 - Baris relasi, selain baris header yang berisi nama atribut, disebut tupel.

- Sebuah tupel memiliki satu komponen untuk setiap atribut relasinya.
- Domain
 - Himpunan nilai yang diijinkan untuk satu atau lebih atribut.
 - Model relasional mensyaratkan bahwa setiap komponen dari setiap tupel menjadi atom; Artinya, harus dari beberapa jenis dasar seperti integer atau string. Nilai tidak boleh berupa struktur record, set, list, array, atau jenis lain apa pun yang nilainya dapat dipecah menjadi komponen yang lebih kecil.



Gambar 2.7 Konsep Database Relasional Model



Gambar 2.8 Model Relational

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

Gambar 2.9 Domain untuk beberapa atribut dari Branch dan Staff relations

- Degree
Derajat suatu relasi adalah jumlah atribut yang dikandungnya
- Cardinality
Kardinalitas suatu relasi adalah jumlah tupel yang dikandungnya.
- Relational database
Kumpulan relasi yang dinormalisasi dengan nama relasi yang berbeda.

Kelebihan database relasional

- Data sangat cepat diakses
- Struktur database mudah dilakukan perubahan
- Data direpresentasikan secara logik, user tidak membutuhkan bagaimana data disimpan.
- Mudah untuk membentuk query yang kompleks dalam melakukan retrieve data

- Mudah untuk mengimplementasikan integritas data
- Data lebih akurat
- Mudah untuk membangun dan memodifikasi program aplikasi
- Telah dikembangkan Structure Query Language (SQL).

Kelemahan database relasional

- Kelompok informasi/tables yang berbeda harus dilakukan joined untuk melakukan retrieve data
- User harus familiar dengan relasi antar tabel
- User harus belajar SQL.

Database Relasi

Dalam relasi database, kita mengenal istilah

- a. Skema relasi
Nama relasi yang ditentukan oleh sekumpulan atribut dan pasangan nama domain.
- b. Skema relasi database
Sekumpulan skema relasi, masing-masing dengan nama berbeda.

Properti dari Relasi

Relasi memiliki properti berikut:

- Relasi memiliki nama yang berbeda dari semua nama relasi lain dalam skema relasional;
- Setiap sel dari relasi berisi tepat satu nilai atom (tunggal);
- Setiap atribut memiliki nama yang berbeda;
- Nilai atribut semuanya berasal dari domain yang sama;
- Setiap tupel berbeda; tidak ada duplikat tupel;
- Urutan atribut tidak memiliki signifikansi;
- Urutan tupel tidak memiliki signifikansi, secara teoritis. (namun, dalam praktiknya, urutan dapat memengaruhi efisiensi mengakses tupel.)

Relational Keys

a. Superkey

Atribut, atau sekumpulan atribut, yang secara unik mengidentifikasi tupel dalam suatu relasi.

b. Candidate key

- Superkey sehingga tidak ada subset yang tepat adalah superkey dalam relasi.
- Candidate key mempunyai dua properties, yaitu:
 - a. *Uniqueness*
 - b. *Irreducibility*

c. Primary key

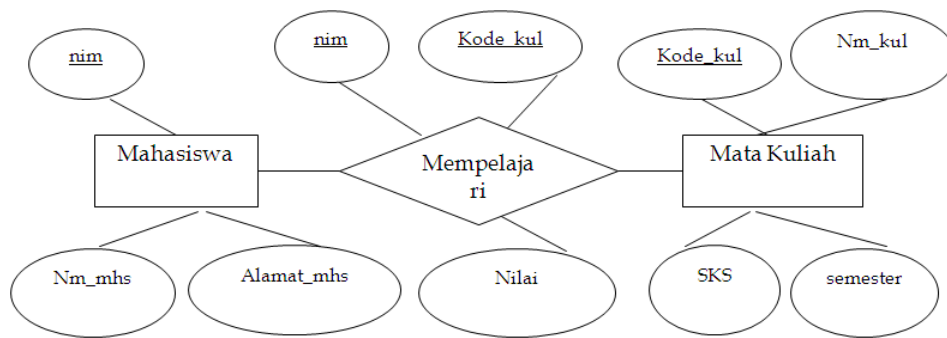
Kunci kandidat yang dipilih untuk mengidentifikasi tupel secara unik dalam relasi.

d. Foreign key

Atribut, atau serangkaian atribut, dalam satu relasi yang cocok dengan *candidate key* dari beberapa relasi (mungkin sama).

5. Model data *Entity Relationship*

- Entity-relationship model (ERM), kadang-kadang disebut sebagai entity-relationship diagram (ERD), dapat digunakan untuk mewakili model data konseptual abstrak (atau model data semantik atau model data fisik) yang digunakan dalam rekayasa perangkat lunak untuk mewakili data terstruktur.
- Menjelaskan hubungan antar data dalam system database berdasarkan suatu persepsi bahwa realworld terdiri dari obyek-obyek dasar yang mempunyai hubungan relasi antara obyek-obyek tersebut
- Relasi antara obyek dilukiskan dengan menggunakan simbol-simbol grafis tertentu



Gambar 2.10 Entity Relationship Diagram

6. Model data Berorientasi Objek(Object Oriented data model)

- Model data berorientasi objek adalah metode lain untuk merepresentasikan objek dunia nyata.
- Model data berorientasi objek menganggap setiap objek di dunia sebagai objek dan mengisolasinya satu sama lain.
- Model data berorientasi objek mengelompokkan fungsionalitas terkaitnya bersama-sama dan memungkinkan mewarisi fungsinya ke sub-grup terkait lainnya.

Kelebihan database berorientasi objek

- Programmer hanya dibutuhkan memahami konsep berorientasi objek untuk mengkombinasikan konsep berorientasi objek dengan storage database relasional
- Objek dapat dilakukan sifat pewarisan dari objek yang lain
- Secara teoritis mudah untuk mengatur objek
- Model data berorientasi objek lebih kompatibel dengan tools pemrograman berorientasi objek.

Kelemahan database berorientasi objek

- User harus memahami konsep berorientasi objek, karena database berorientasi objek tidak dapat bekerja dengan metoda pemrograman tradisional

Pentingnya data model

Ada dua model data yang sangat penting untuk sistem database, yaitu:

1. Relasional model, termasuk object-relational extensions
2. Model data semistructured, termasuk XML dan standar relasi

Pengantar model relasional

- Model database yang menggunakan table dua dimensi, yang terdiri dari baris dan kolom untuk menggambarkan sebuah berkas data.
- Konstruksi utama merepresentasikan data dalam model relasional adalah relasi
- Relasi terdiri dari skema relasi dalam contoh relasi
- Skema relasi
 - Mendeskripsikan kepala kolom/field dari table
 - Menentukan nama relasi, nama masing-masing field (kolom/atribut) dan domain dari masing-masing field.
 - Domain yang diacu memiliki nama domain dan kumpulan nilai yang sesuai.

Contoh relasi:

Students(sid : string, name: string, login : string, age: integer, gpa : real)

Sid	Name	Login	Age	gpa
20182001	Luki	luki@gmail.com	20	3.3
20182006	Smith	smit@gmail.com	19	3.4
20182010	Smith	smith@yahoo.com	21	3.8
20182009	Reza	reza@gmail.com	22	2.8
20182012	Lia	lia@gmail.com	18	3.0

Latihan

Tabel Relation Account

AccNo	Type	Balance
12345	Savings	12000
23456	Checking	1000
34567	Savings	25

Tabel Relation Customer

firstName	lastName	idNo	Account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

1. Tabel diatas merupakan database perbankan. Sebutkan:
 - c. Atribut dari setiap relasi
 - d. Tuple dari setiap relasi
 - e. Komponen satu tupel dari setiap relasi
 - f. Relasi skema dari setiap relasi
 - g. Skema database
 - h. Domain yang cocok untuk setiap atribut
 - i. Cara lain yang sama untuk menampilkan setiap relasi

Kasus untuk soal 2-5:

Tabel berikut merupakan bagian dari database yang disimpan dalam DBMS relasional:

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

dimana Hotel berisi detail hotel dan hotelNo sebagai primary key;

Room berisi room details untuk setiap hotel dan (roomNo, hotelNo) membentuk primary key;

Booking berisi details dari bookings dan (hotelNo, guestNo, dateFrom) membentuk primary key;

Guest berisikan guest details dan guestNo adalah the primary key.

Soal:

2. Identifikasi foreign key dalam skema ini. Jelaskan bagaimana entitas dan aturan integritas referensial berlaku untuk hubungan ini.
3. Buat beberapa tabel contoh untuk relasi ini yang mengamati aturan integritas relasional. Sarankan beberapa batasan umum yang sesuai untuk skema ini.
4. Analisis RDBMS yang sedang Anda gunakan. Tentukan dukungan yang diberikan sistem untuk primary key, alternative key, foreign key, integritas relasional, dan views.
5. Terapkan skema di atas di salah satu RDBMS yang saat ini Anda gunakan. Buat dua tampilan pengguna yang dapat diakses dan diedit serta dua tampilan pengguna lain yang tidak dapat diperbarui.



Universitas
Esa Unggul