Nama : Afina Putri Dayanti

NIM : 825200049

Jurusan : Sistem Informasi

Mata Kuliah : UTS Database Systems

1. Jelaskan pendekatan sistem berbasis data (database systems) dalam penanganan data dan berikan contoh keuntungan pendekatan ini, selain itu berikan contoh minimal 3 sistem berasis data yang digunakan perusahaan!

Pendekatan database memungkinkan setiap pengguna untuk memiliki tampilan database mereka sendiri yang hanya menjelaskan data yang menarik bagi pengguna tersebut. Ada berbagai manfaat memiliki banyak tampilan seperti:

* Kurangi kerumitan
* Berikan tingkat keamanan
* Menyediakan mekanisme untuk menyesuaikan tampilan database
* Menampilkan gambar yang konsisten dan tidak berubah dari struktur database, bahkan jika database yang mendasarinya berubah

Contoh 3 sistem basis data yang digunakan perusahaan

* Oracle
* MySQL
* Microsoft SQL Server
* PostgreSQL
* MongoDB

1. Sebutkan sifat-sifat (property) dari relasi, dan berdasarkan tabel client di bawah sebutkan yang mana property relasinya?

* Client.clientNo to Viewing.clientNo = One to Many
* Viewing.clientNo to Client.clientNo = One to One

1. Jelaskan tentang abstraksi data 3 level!

* **External level** terdiri dari tampilan database pengguna. Level ini biasa disebut view level beberapa pengguna dapat melihat data yang diinginkan dari level ini yang secara internal diambil dari database dengan bantuan conceptual and internal level mapping.)
* **Conceptual level** adalah tampilan komunitas dari database. Tingkat ini berisi struktur logic dari seluruh database seperti yang terlihat oleh Database Administrator.
* **Internal level** adalah tampilan komputer dari database. Tingkatan ini menjelaskan bagaimana data disimpan dalam database

1. Jelaskan tujuan dan aktifitas yang dilakukaan pada setiap tahapan database life cycle

* Database planning : merencanakan tahapan life cycle agar dapat direalisasikan paling efisien dan efektif
* System definition : menentukan ruang lingkup dan batasan database system, termasuk user views, users, dan application areas
* Requirements collection and analysis : Pengumpulan dan analisis persyaratan untuk database system baru
* Database design : conceptual, logical, dan physical design dari database
* DBMS selection (optional) : memilih DBMS yang cocok untuk database system
* Application design : merancang user interface dan aplikasi yang mana menggunakan dan memproses database
* Prototyping (optional) : membangun model database system, yang memungkinkan para desainer atau users untuk memvisualisasikan dan mengevaluasi bagaimana final system akan terlihat dan berfungsi
* Implementation : membuat definisi physical database dan program aplikasi
* Data conversion and loading : memuat data dari sistem lama ke sistem baru dan, jika memungkinkan, mengkonversi aplikasi yang ada untuk dijalankan di database baru
* Testing : Sistem database diuji untuk kesalahan dan divalidasi terhadap persyaratan yang ditentukan oleh users
* Operational maintenance : database system diimplementasikan sepenuhnya. Sistemnya terus dipantau dan dipelihara. Jika perlu, baru persyaratan dimasukkan ke dalam database system melalui tahap sebelumnya dari lifecycle

1. Jelaskan yang dihasilkan dari operasi aljabar relasi berikut ini:
2. ΠhotelNo(σprice > 50(Room))

* Ini akan menghasilkan realasi dengan atribut (hotelNo) yang memberikan jumlah hotel dengan harga Room lebih besar dari 50)

1. σHotel.hotelNo = Room.hotelNo(Hotel × Room)

* Ini akan menghasilkan JOIN dari Hotel dan relasi Room yang berisi semua atribut Hotel dan Room (akan ada dua salinan atribut hotelNo). Pada dasarnya ini akan menghasilkan hubungan yang berisi semua Room di semua hotel.

1. ΠhotelName(Hotel 1 Hotel.hotelNo = Room.hotelNo(σprice > 50(Room)))

* Ini akan menghasilkan JOIN Hotel dan tuple Room tersebut dengan harga lebih dari 50. Pada dasarnya ini akan menghasilkan relasi yang berisi semua nama Hotel dengan harga Room di atas 50.

1. Guest 5(σdateTo ≥‘1-Jan-2002’(Booking))

* Ini akan menghasilkan (left outer) JOIN Guest dan tupel Pemesanan dengan tanggal akhir (dateTo) lebih besar dari atau sama dengan 1-Jan-2002. Semua Guest yang tidak memiliki pemesanan dengan tanggal seperti itu akan tetap diikutsertakan. Pada dasarnya ini akan menghasilkan hubungan yang berisi semua Guest dan menunjukkan rincian setiap pemesanan yang mereka miliki setelah 1-Jan-2002

1. Hotel 2 Hotel.hotelNo = Room.hotelNo(σprice > 50(Room))

* Ini akan menghasilkan (semi) JOIN dengan Hotel dan tupel Room dengan harga lebih dari 50. Hanya atribut Hotel tersebut yang akan dicantumkan. Pada dasarnya ini akan menghasilkan JOIN yang berisi semua detail semua hotel dengan harga kamar di atas 50

(f ) ΠguestName, hotelNo(Booking 1 Booking.guestNo = Guest.guestNo Guest) ÷ ΠhotelNo(σcity = ‘London’(Hotel))

* Ini akan menghasilkan relasi yang berisi nama semua Guest yang telah memesan semua Hotel di London

1. Tabel -tabel berikut merupakan bagian dari database yang disimpan dalam DBMS relasionalHotel(hotelNo, hotelName, city)

• Room (roomNo, hotelNo, type, price)

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

• Guest (guestNo, guestName, guestAddress)

di mana Hotel berisi detail hotel dan hotelNo adalah kunci utama;

Room berisi detail kamar untuk setiap hotel dan (roomNo, hotelNo) adalah kunci utama;

Booking berisi detail pemesanan dan (hotelNo, guestNo, dateFrom) membentuk kunci utama;

Room berisi detail tamu dan guestNo adalah kunci utama.

Identifikasi kunci asing dalam skema ini. Jelaskan bagaimana entitas dan aturan integritas referensial berlaku untuk hubungan tersebut.

CREATE TABLE Room (

RoomNO INTEGER,

HotelNo VARCHAR,

Type VARCHAR,

Price DECIMAL,

PRIMARY KEY...

* Untuk setiap relasi, primary key tidak boleh mengandung nol.
* Kamar relate dengan Hotel melalui atribut hotelNo

Oleh karena itu, hotelNo di Room harus null atau berisi nomor hotel yang ada dalam relasi Hotel. Dalam studi kasus ini, mungkin tidak dapat diterima untuk memiliki hotelNo di Kamar dengan nilai null.

* Pemesanan terkait dengan Hotel melalui atribut hotelNo.

Oleh karena itu, hotelNo dalam Pemesanan harus null atau berisi nomor hotel yang ada dalam relasi Hotel. Namun, karena hotelNo juga merupakan bagian dari primary key, nilai null untuk atribut ini tidak dapat diterima. Demikian pula untuk guestNo. Pemesanan juga terkait dengan Room melalui atribut roomNo.

1. Jelaskan apakah yang dimaksud relasi, tuple, cardinality, degree dan atribut !

* Relasi : tabel dengan kolom dan baris
* Tuple : deretan relasi
* Cardinality : cardinality suatu relasi adalah jumlah tupel yang dikandungnya
* Degree : degree suatu relasi adalah jumlah atribut yang dikandungnya
* Atribut : nama kolom dari suatu relasi

1. Menampilkan nama kota dan type property yang harga sewanya kurang atau sama dengan 450 dan tuliskan hasilnya!

SELECT city, type FROM PropertyForRent WHERE rent <= 450

|  |  |
| --- | --- |
| city | type |
| London | Flat |
| Glasgow | Flat |
| Glasgow | Flat |
| Glasgow | Flat |

1. Menampilkan nama owner yang memiliki property dengan tipe house dan tuliskan hasillnya!

SELECT fName FROM PrivateOwner

JOIN PropertyForRent

ON PrivateOwner.ownerNo = PropertyForRent.ownerNo

WHERE type = “House”

|  |
| --- |
| fName |
| Joe |

1. Menampilkan nama nama client dan alamat property yang dilihat pada tanggal 26-may-04 dan tuliskan hasilnya!

SELECT Client.fName, PropertyForRent.street, PropertyForRent.city FROM Client

JOIN Viewing ON Client.clientNo = Viewing.clientNO

JOIN PropertyForRent ON Viewing. propertyNo = PropertyForRent.propertyNo

WHERE viewDate = 26-May-04

|  |  |  |
| --- | --- | --- |
| fName | street | city |
| Aline | 6 Lawrence St | Glasglow |

1. Menampilkan nama owner yang propertinya sudah dilihat client bernama Aline dan sebutkan hasilnya!

SELECT PrivateOwner.fName , PrivateOwner.lName FROM PropertyForRent

JOIN PrivateOwner ON PrivateOwner.ownerNo = PropertyForRent.ownerNo

JOIN Viewing ON Viewing.propertyNo = PropertyForRent.propertyNo

JOIN Client ON Client.clientNo = Viewing.clientNo

WHERE Client.fName = “Aline”

|  |  |
| --- | --- |
| fName | iName |
| Tina | Murphy |