DATABASE RELASIONAL

DIANA RAHMAWATI

FILE & DATABASE

File :

gabungan dari beberapa catatan yang saling berhubungan

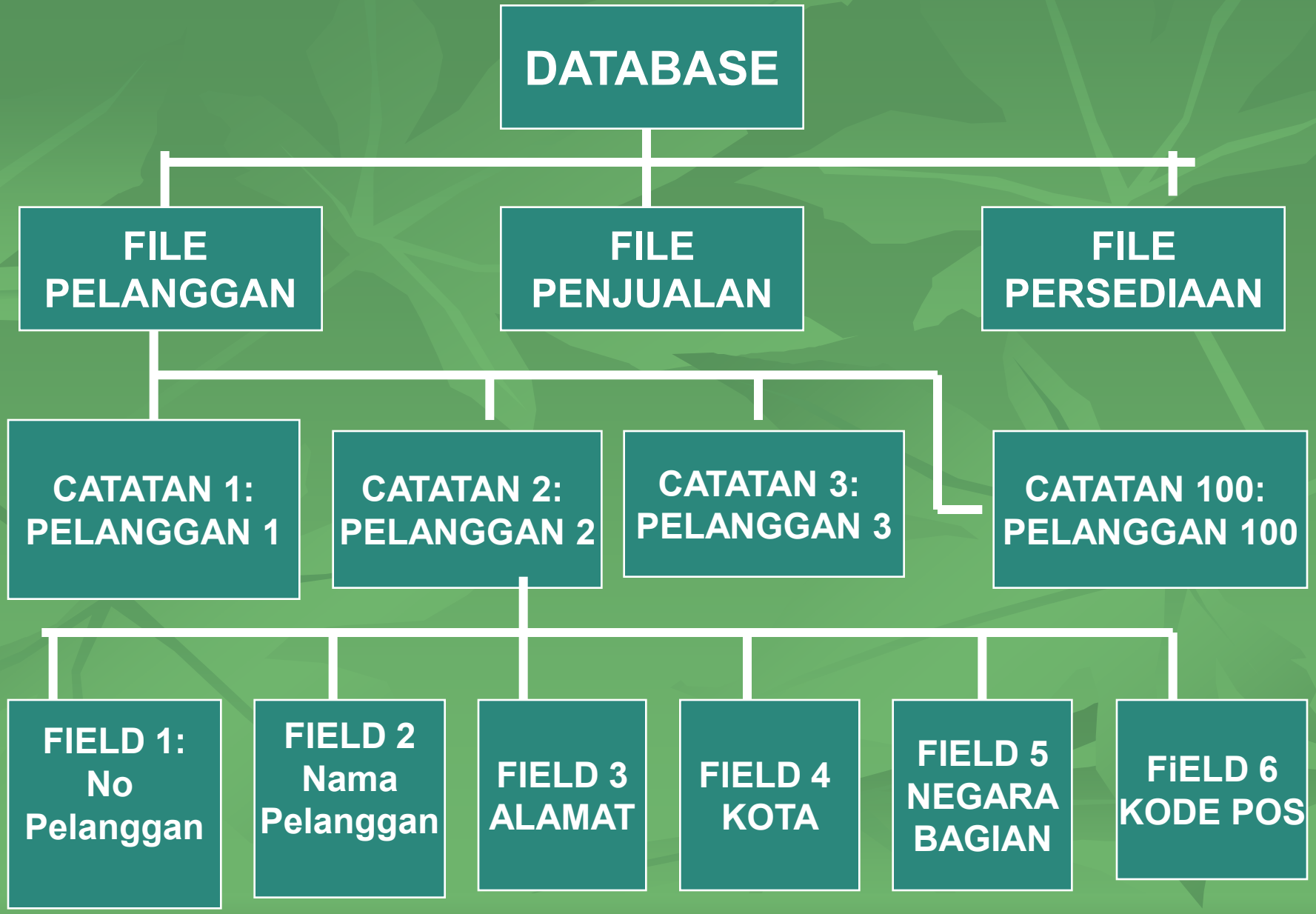
Jenis File :

1. File utama : konsepnya sama dengan buku besar (dalam SIA manual); bersifat permanen
2. File transaksi : konsepnya sama dengan jurnal

Database :

gabungan dari file yang saling berhubungan dan dikoordinasi secara terpusat /terintegrasi

Contoh File & Database Piutang



Pendekatan File vs Pendekatan Database

SISTEM DATABASE

Dalam Database, data dapat dilihat dalam dua pandangan :

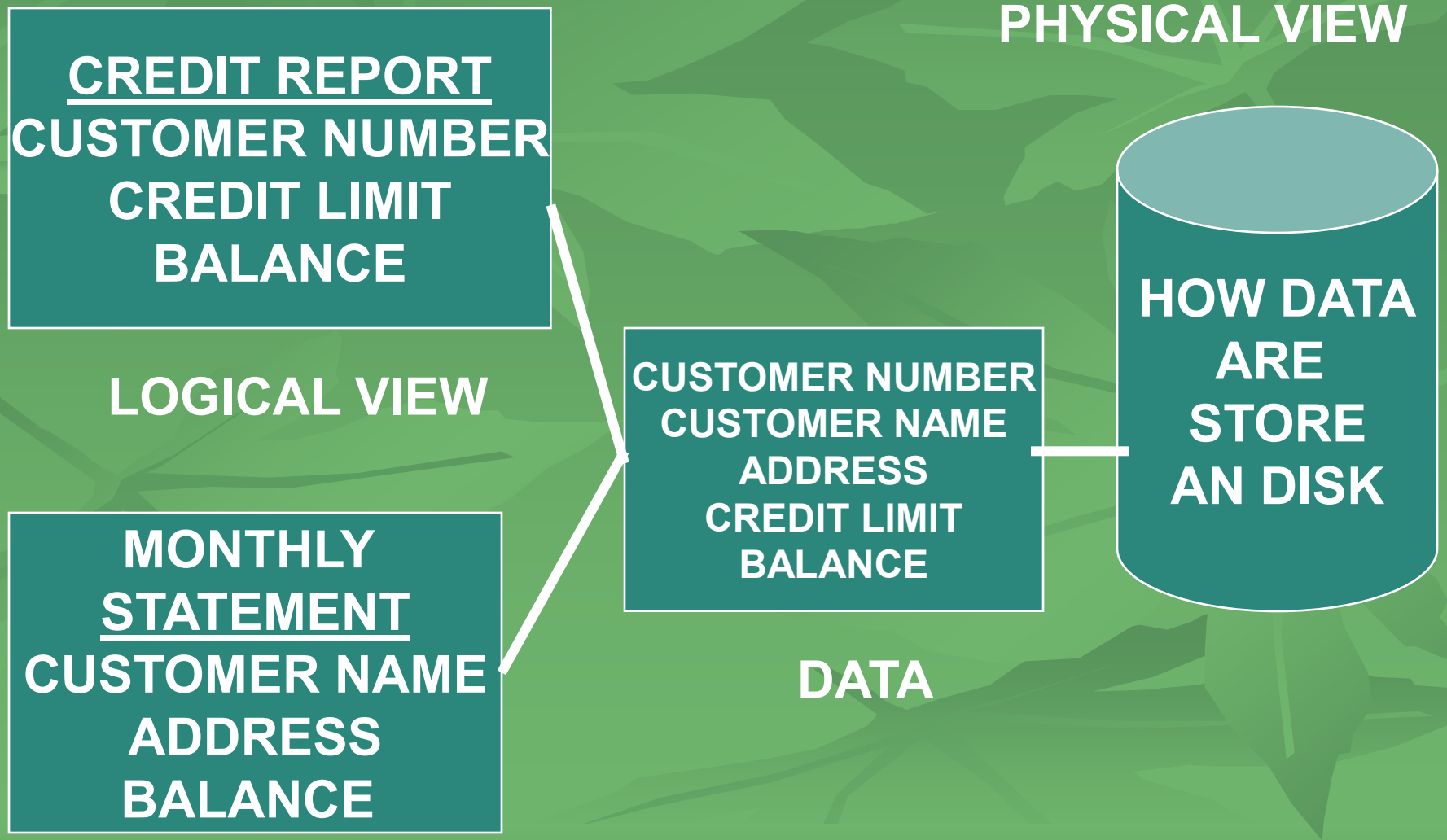
1. Logical View (pandangan logis) :

   berhubungan dengan bagaimana users secara konseptual mengorganisasi, melihat dan memahami hubungan antar data

2. Physical View (pandangan phisik)

   berkaitan dengan bagaimana dan dimana secara phisik data akan ditat dan disimpan dalam disket, flash disk, CD atau media lain.

LOGICAL AND PHYSICAL VIEWS OF DATA IN CUSTOMER DATABASE



DATA BASE MANAGEMENT SYSTEMS

In Chapter 4 we introduced the distinction between the logical and physical views of data. The way the user or programmer conceptually organizes and understands the relationships among data items represents the logical view of the data; the way that those data items are actually stored represents the physical view of the data. For example, a sales manager may conceptualize all information about customers as being stored in the form of a table (the logical view), although that data may actually be stored in an indexed sequential file (the physical view). As shown in Fig. 5.1, the data base management system (DBMS) translates the user's logical view of the data into the underlying physical view so the desired data can be retrieved and presented to the user.

Although users' logical views of the data base may differ, the system stores data in only one way. In some cases, however, the physical organization of data can be quite different from the user's perception of how those data are stored. For example, data items such as customer account balance, name, address, and credit history may be stored in separate locations, or even on separate disks, even though users perceive a close logical relationship between those items. The DBMS hides these details about physical storage from users, so that they can concentrate on the logical relationships among data items. Ideally, the

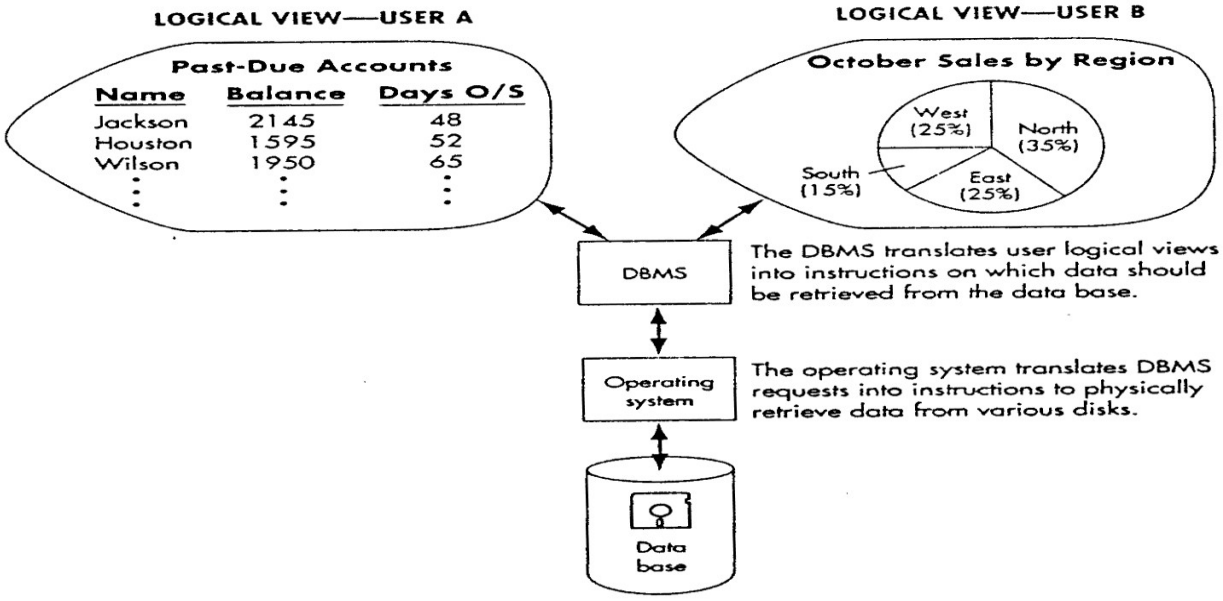
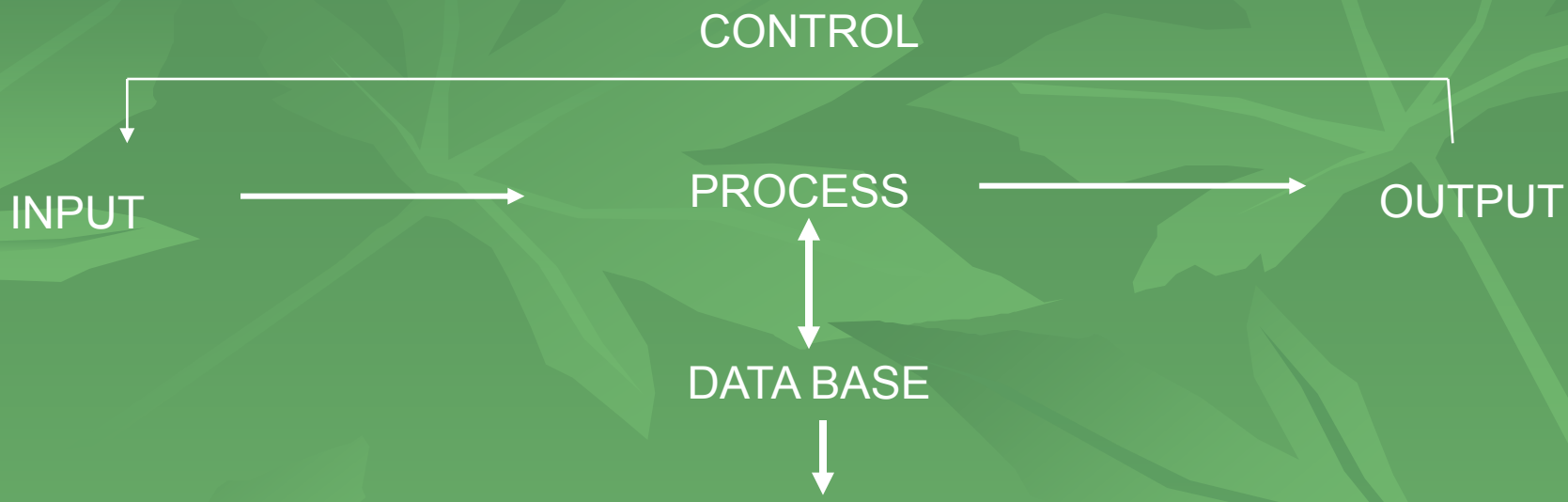


Figure 5.1
Function of the DBMS

DATA BASE SYSTEM



Schema

External level

Conceptual level

Internal level

Dictionary

All information about data base

Language

DDL, DML, DQL, Dbase, Acess, ETC

Administrator

Develop & govern all org. data. (7 functions : 137)

Keterangan

1. Skema :

berfungsi mendeskripsikan struktur logis database

Terdapat 3 macam skema :

a. Skema tingkat eksternal:sekumpulan pandangan logis pemakai individual mengenai bagian dari database.

b. Skema tingkat konseptual : pandangan seluruh databasepada tingkat organisasi. Didalam skema ini mendaftar elemen-elemen data dan hubungan antar elemen tsb.

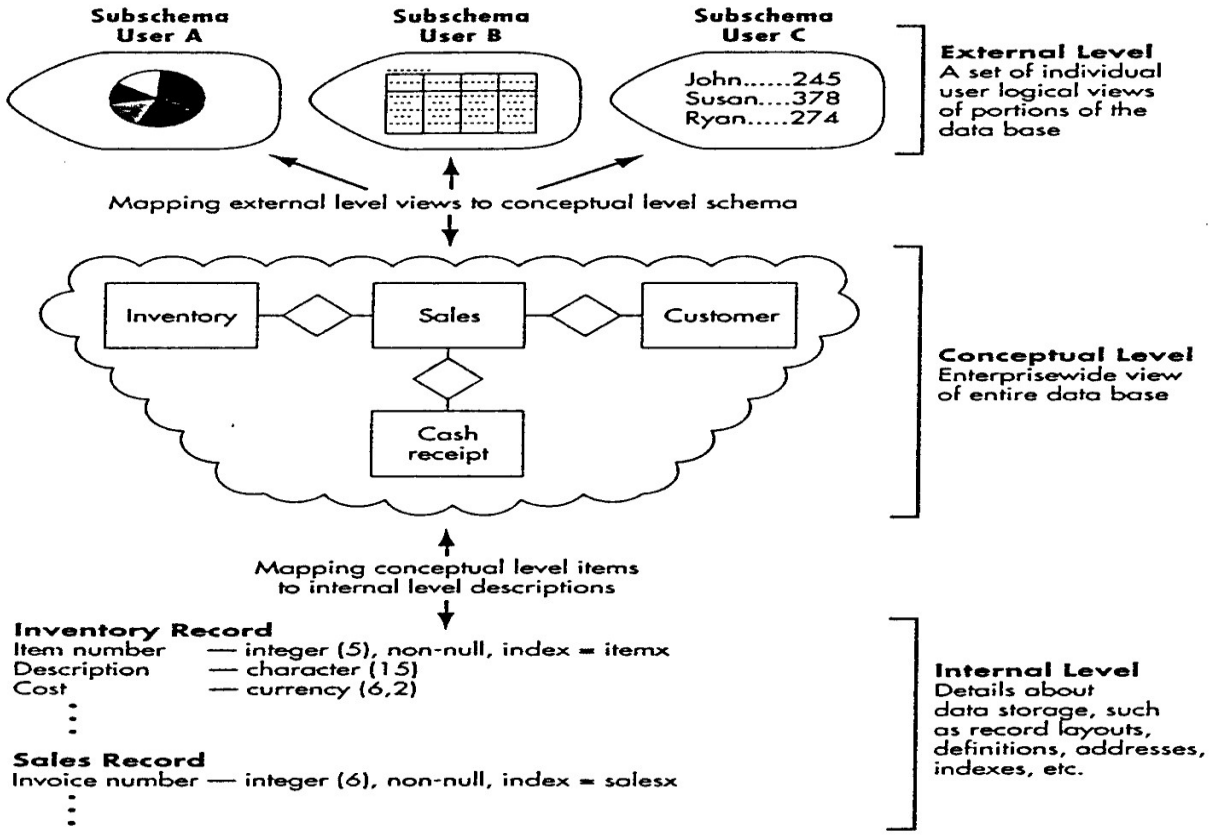
c. Skema tingkat internal : tampilan tingkat rendah dari database. Skema internal mendeskripsikan rincian mengenai penyimpanan data.

Schemas

A **schema** describes the logical structure of a data base. There are three levels of schemas: the conceptual, the external, and the internal. Figure 5.2 shows the relationships between these three levels.

The **conceptual level schema** is an organization wide view of the entire data base. It lists all data elements and the relationships between them. The **external level schema** consists of a set of individual user views of portions of the data base, each of which views is also referred to as a **subschema.** The **internal level schema** provides a low-level view of the data base. It describes how the data are actually stored and accessed, including information about pointers, indexes, record lengths, and so forth.

Figure 5.2
Three Levels of Schemas



2. Kamus Data:

merupakan salah satu komponen kunci dari DBMS yang mencakup seluruh informasi mengenai struktur database

3. Bahasa DBMS :

Merupakan sekelompok perintah yang digunakan untuk menjalankan fungsi menciptakan, mengubah dan mempertanyakan database

Terdapat 3 bahasa : bahasa definsi data (DDL), bahasa manipulasi data (DML) dan bahasa permintaan (DQL)

Table 5.1 **Example of a Data Dictionary**

Data Element Name	Description	Records in Which Contained	Source	Field Length	Field Type	Programs in Which Used	Outputs in Which Contained	Authorized Users	Other Data Names
Customer number	Unique identifier of each customer	A/R record, customer record, sales analysis record	Customer number listing	10	Alphanumeric	A/R update, customer file update, sales analysis update, credit analysis	A/R aging report, customer status report, sales analysis report, credit report	No restrictions	None
Customer name	Complete name of customer	Customer record	Initial customer order	20	Alphanumeric	Customer file update, statement processing	Customer status report, monthly statement	No restrictions	None
Address	Street, city, state, and zip code	Customer record	Credit application	30	Alphanumeric	Customer file update, statement processing	Customer status report, monthly statement	No restrictions	None
Credit limit	Maximum credit that can be extended to customer	Customer record, A/R record	Credit application	8	Numeric	Customer file update, A/R update, credit analysis	Customer status report, A/R aging report, credit report	R. Drummond W. Francom H. Heaton	CR_limit
Balance	Balance due from customer on credit purchases	A/R record, sales analysis record	Various sales and payment transactions	8	Numeric	A/R update, sales analysis update, statement processing, credit analysis	A/R aging report, sales analysis report, monthly statement, credit report	O. Cherrington J. Hansen K. Stocks	Cust_bal

DBMS Languages

Every DBMS must provide a means of performing the three basic functions of creating, changing, and querying the data base. The set of commands used to perform these functions are referred to as the data definition, data manipulation, and data query languages, respectively.

The **data definition language (DDL)** is used to (1) build the data dictionary, (2) initialize or create the data base, (3) describe the logical views for each individual user or programmer, and (4) specify any limitations or constraints on security imposed on data base records or fields. Table 5.2 shows an example of a DDL command used to create a table to store information about vendors.

The **data manipulation language (DML)** is used for data maintenance, which includes such operations as updating, inserting, and deleting portions of the data base. The DML simplifies the writing of programs to accomplish these tasks by requiring references only to the names of data items, rather than to their physical storage locations. Table 5.3 shows examples of the three basic types of DML commands.

The **data query language (DQL)** is used to interrogate the data base. Whereas the DML is used to change the contents of the data base, the DQL merely retrieves, sorts, orders, and presents subsets of the data base in response

Table 5.2 **Example of Data Definition Language (DDL) Command**

The following command creates a table to store information about vendors:

```
CREATE TABLE        Vendor
    (Vendor#            INTEGER (5) NOT NULL,
    Name                CHARACTER(15),
    Street_Address      CHARACTER(20),
    City                CHARACTER(12),
    State               CHARACTER(2),
    Zipcode             CHARACTER(10),
    Balance             FLOATING(10)   )
```

This command creates a vendor table with seven columns. The vendor number and balance columns must contain only numeric values: vendor# will take integer values, whereas balance can be any numeric value, including decimal format. The remaining columns may contain either numbers or letters. The number in parentheses indicates the maximum number of characters that can be stored in that column. Finally, the constraint NOT NULL indicates that vendor# cannot be left blank.

Result of the command:

Vendor

Vendor#	Name	Street_ Address	City	State	Zipcode	Balance

Note: This command creates an empty table. Filling the table requires the use of data manipulation language commands.

Table 5.3 **Example of Data Manipulation Language (DML) Operations**

This command inserts a new row, containing information about St. Louis Appliances, into the vendor table:

```
INSERT
INTO      Vendor (Vendor#, name, street_address, city, state, zipcode, balance)
VALUES    (10004, 'St. Louis Appliances', '2455 Chippewa', 'St. Louis', 'MO',
          '63109-2643', 0)
```

This command updates the address of St. Louis Appliances:

```
UPDATE    Vendor
SET       Street_address = '3542 Chippewa',
          Zipcode = '63110-2214'
WHERE     Vendor# = 10004
```

This command deletes from the vendor table the row containing information about St. Louis Appliances:

```
DELETE
FROM      Vendor
WHERE     Vendor# = 10004
```

to user queries. Most DQLs contain a fairly powerful, but easy to use, set of commands that enable users to satisfy many of their own information needs, without the assistance of a programmer. Table 5.4 presents a sample query command.

Many DBMSs also include a **report writer,** which is a language that simplifies report creation. Typically, users need only specify which data elements they want printed and how the report should be formatted. The report writer then searches the data base, extracts the specified data items, and prints them out according to the user-specified format.

DBMS Functions and Users

All users generally have access to both the DQL and the report writer. Access to the DDL and DML, however, should be restricted to those employees with administrative and programming responsibilities. This helps to limit the number of people who have the capability to make changes to the data base. Let us now briefly discuss the principal administrative and programming functions in a data base system.

Data Administrator. The **data administrator (DA)** is responsible for developing general policies and procedures governing all organizational data, not just what is stored in the data base. The DA is ultimately responsible for understanding the information needs of the organization in order to decide what should be included in the data base.

Data Base Administrator. The **data base administrator (DBA)** is responsible for coordinating, controlling, and managing the data base. In a way, the

Table 5.4 **Example of Data Query Language (DQL) Command**

Query:

```
SELECT      Name, Balance
FROM        Vendor
ORDER BY    Balance, Descending
```

Result:

Name	*Balance*
South Side Electronics	3,987.00
St. Louis Electronics Supply	3,250.67
St. Louis Computer Warehouse	2,311.85
Oakville Electronics	954.95

This query specifies that the name and balance columns (the SELECT command) in the vendor table (the FROM command) be displayed. The result is arranged in descending order (the ORDER BY and the DESCENDING commands) by the amount owed.

DBA can be thought of as the human equivalent of the DBMS. That is, the DBA must be aware not only of users and their data requirements but also of how the DBMS operates and how data are stored and processed. Specifically, the DBA has the following major responsibilities:

- Create logical models of the data base.
- Establish data standards and specifications.
- Approve changes to the data base structure.
- Develop retrieval methods as required by data base users.
- Specify and maintain the physical structure of the data base.
- Maintain the data dictionary.
- Design and implement controls to ensure the accuracy and security of the data base.

The DBA uses the DDL to specify the structure of the data base, implement and maintain the data dictionary, and specify controls.

Application Programmers. The programs that interact with the DBMS to process data are written by **application programmers.** Billing and payroll are two common examples of application programs. These application programs use the DML to access and change the contents of the data base. The DML improves the productivity of application programmers by allowing them to concentrate solely on correctly specifying the logical description of the task to be performed, without concern about how to access the data or about the structure of that data. To see how this simplifies programming, refer back to Table 5.3 and examine the DML command to update the vendor table. Notice the absence of information about field lengths, locations, and so forth, all of which would have to be included in traditional programming languages.

DATABASE RELASIONAL

DBMS (*database Management System*) dikarakterisasikan/digambarkan dalam bentuk suatu model logis data.

Model data: perwakilan abstrak dari isi suatu database.

Perkembangan saat ini DBMS disebut juga sebagai database relasional.

Database relasional: database yang menggunakan model relasional data.

Model relasional data: isi dari suatu database/semua yang disimpan didatabase dalam bentuk tabel-tabel yang saling berhubungan

Setiap baris dalam sebuah tabel yang di sebut tuple berisi data mengenai keberadaan spesifik jenis entitas tertentu

Jenis-Jenis Atribut

Tabel-tabel dalam database relasional memiliki tiga jenis atibut yaitu:

1. Kunci utama (*primary key*) :atribut atau kombinasi dari beberapa atribut yang secara unik mengidentifikasi baris tertentu dalam sebuah tabel.
2. Kunci luar (*foreign key*) : atribut yang muncul dalam suatu tabel, yang juga merupakan kunci utama dalam tabel lainnya.
3. Bukan merupakan kunci baik kunci utama maupun kunci luar didalam tabel yang menyimpan informasi penting mengenai entitasnya.

Persyaratan Dasar Model Data Relasional

1. Setiap kolom dalam sebuah baris harus berlainan nilainya
2. Kunci utama (*primary key*) tidak boleh bernilai nol
3. Kunci luar (*foreign key*) jika tidak bernilai nol, harus memiliki yang nilai sesuai dengan nilai kunci utama dihubungan yang lain.
4. Seluruh atribut yang bukan merupakan kunci utama ataupun lunci luar dalam sebuah tabel harus mendeskripssikan obyek yang diidentifikasi oleh kunci utama.

Masalah Jika Seluruh Data Dalam Satu Tabel

Salah satu masalah yang timbul adalah terjadinya banyak pengulangan.

Tiga masalah yang lain yang timbul apabila seluruh data disimpan dalam satu tabel adalah :

a. *Anomali pembaharuan*
b. Anomali penyisipan
c. Anomali penghapusan.

Solusi masalah: Penggunaan serangkaian tabel

Dua Pendekatan dalam Desain Database

1. Pendekatan normalisasi

   semua data pada awalnya disimpan dalam satu tabel besar. Kemudian diikuti sejumlah peraturan untuk memisah-misahkan tabel awal menjadi serangkaian tabel yang dinormalisasi agar terbebas dari anomali pembaharuan, penyisipan dan penghapusan.

2. Pembuatan model data Semantik

   dalam pendekatan ini desainer database menggunakan pengetahuaanya mengenai proses bisnis yang biasanya berlangsung dan kebutuhan informasi yang berhubungan dengan proses transaksi, membuat gambar grafis yang seharusnya dimasukkan dalam database.

Sistem Database Dan Masa Depan Akuntansi

Pengaruh Sistem Database :

1. Mempengaruhi sifat dasar akuntansi (mulai ditinggalkannya model pembukuan berpasangan /*double entry*)
2. Sistem database dapat mengubah sifat pelaporan eksternal.
3. Pengaruh yang paling signifikan adalah dalam hal cara informasi akuntansi akan digunakan dalam pengambilan keputusan.
4. Sistem database relasional menyediakan kemampuan untuk mengintegrasikan data keuangan dan operasional.
5. Sistem database meningkatkan penggunaan dan nilai informasi akuntansi dalam pembuatan keputusan yang taktis dan strategis.