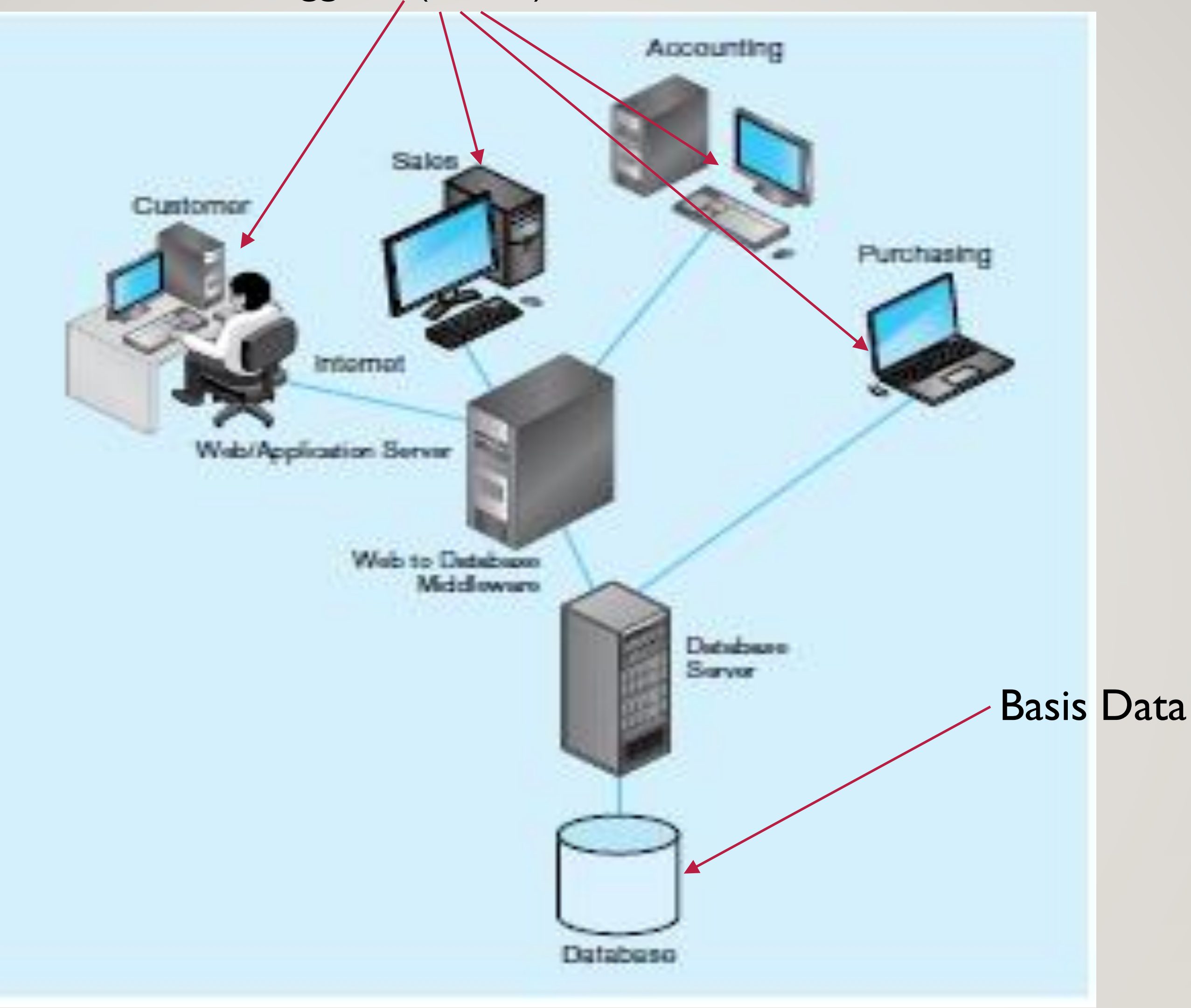
STA261
MANAJEMEN DATA RELASIONAL

Konsep Dasar Basis Data

DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR

SEMESTER GANJIL 2021/2022

Pengguna (Users)



Basis Data

Pendekatan Basis Data

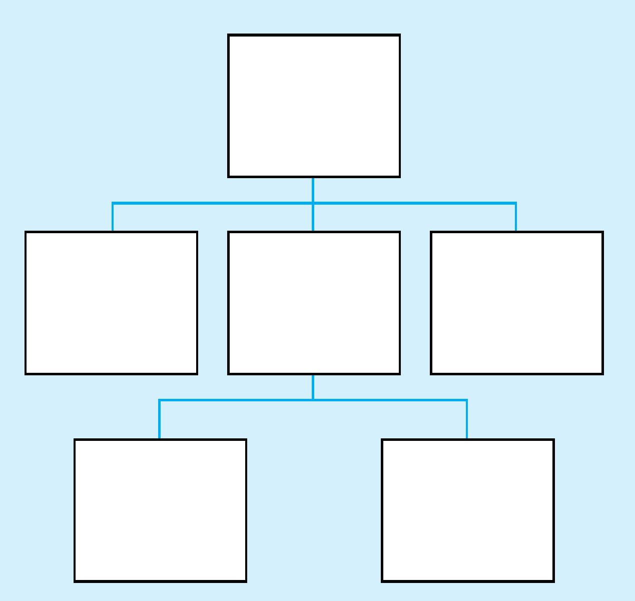
- Model Data (*Data Model*)
- Entiti (*Entity*)
- Hubungan antar Entiti (*Relationship*)
- Basis Data Relasional (*Relational Database*)

Model Data

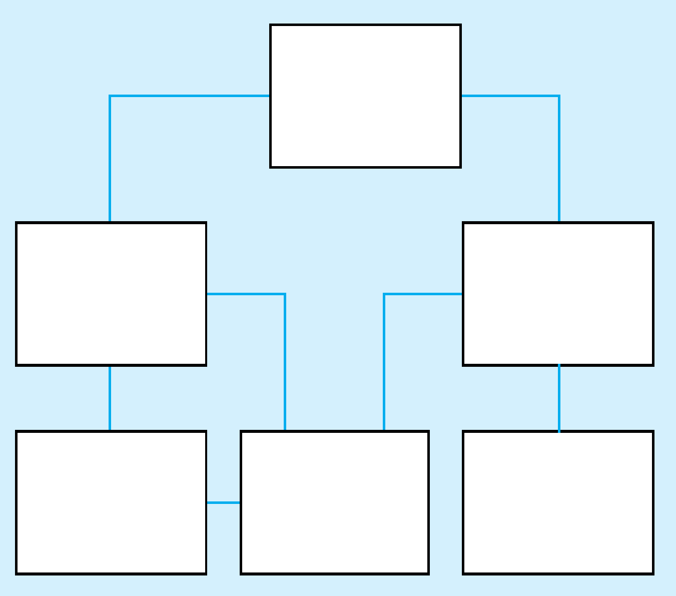
- Gambaran umum tentang hubungan (*relationship*) antar data secara grafis.
- Model data digunakan pada tingkat abstraksi suatu basis data berupa konsep rancangan.
- Model Data Projek yang menggambarkan lebih rinci mengenai tampilan (*view*), kesesuaian struktur data dalam basis data atau kumpulan basis data (*data warehouse*).

Kategori Model Data → struktur basis data

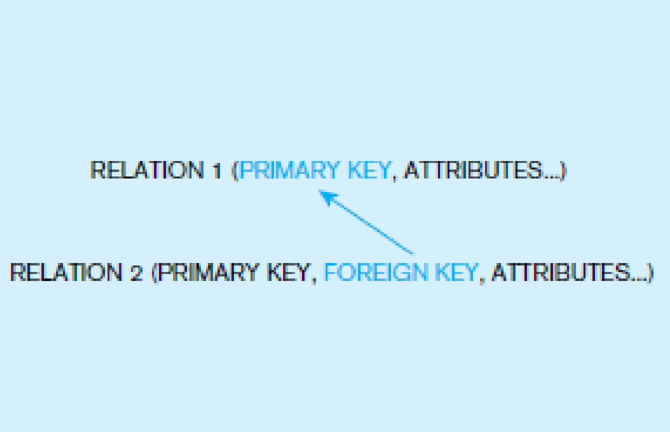
- ✓ **High-level** atau **conceptua**l data models (entities, attributes, and relationships) → **entity–relationship model** ➔ **relational** **data model**, **network** and **hierarchical** **models**
- ✓ **Low-level** atau **physica**l data models to describe the details of how data is stored on the computer storage media, typically magnetic disks → computer specialists



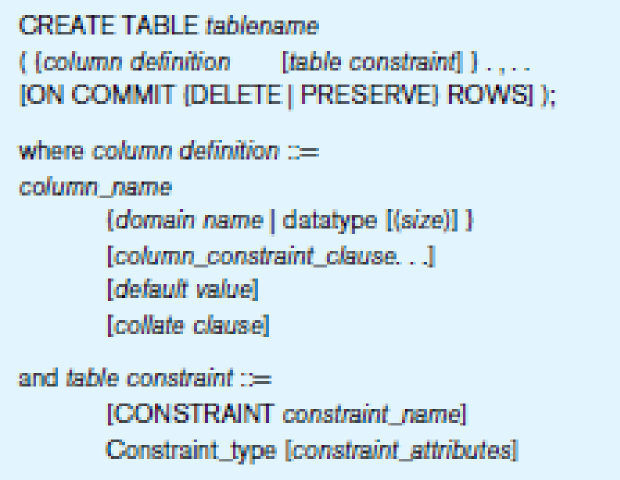
Hierarchical database model



Network database model


RELATION 1 (PRIMARY KEY, ATTRIBUTES...)
RELATION 2 (PRIMARY KEY, FOREIGN KEY, ATTRIBUTES...)

Relational database model



```
CREATE TABLE tablename
( {column definition      [table constraint] } . , . .
[ON COMMIT {DELETE | PRESERVE} ROWS] );

where column definition ::=
column_name
        {domain name | datatype [(size)] }
        [column_constraint_clause. . .]
        [default value]
        [collate clause]

and table constraint ::=
        [CONSTRAINT constraint_name]
        Constraint_type [constraint_attributes]
```

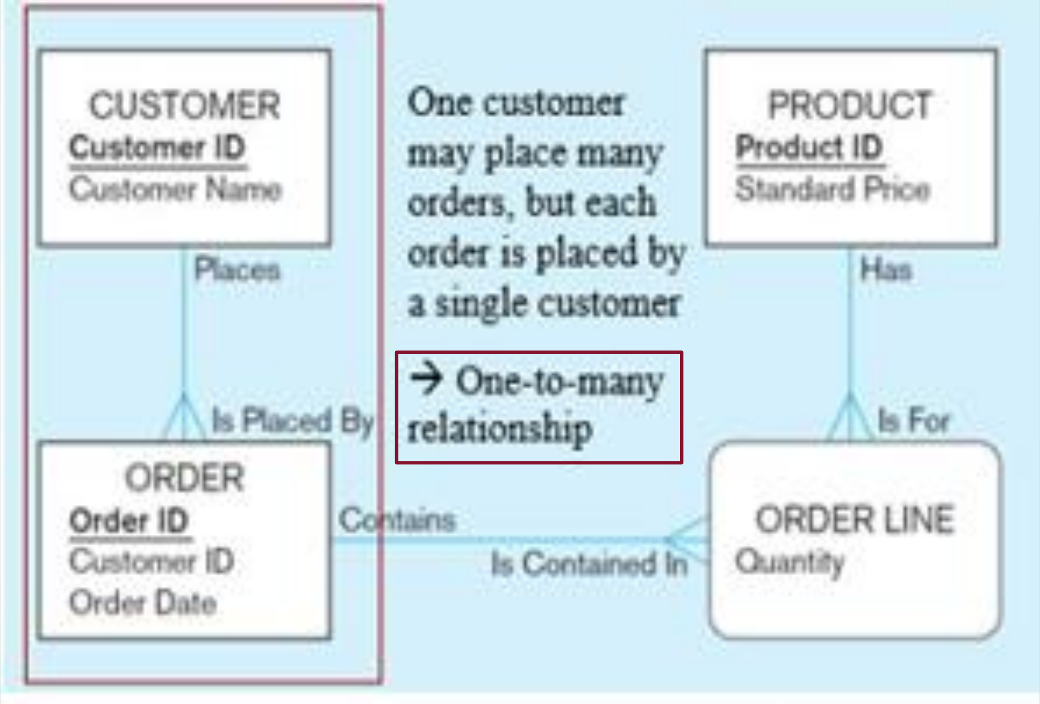
SQL syntax

Entiti

- Bentuk objek seperti orang, tempat, kejadian, atau konsep lingkungan pengguna yang diharapkan memelihara data.
- Terdiri dari atribut-atribut

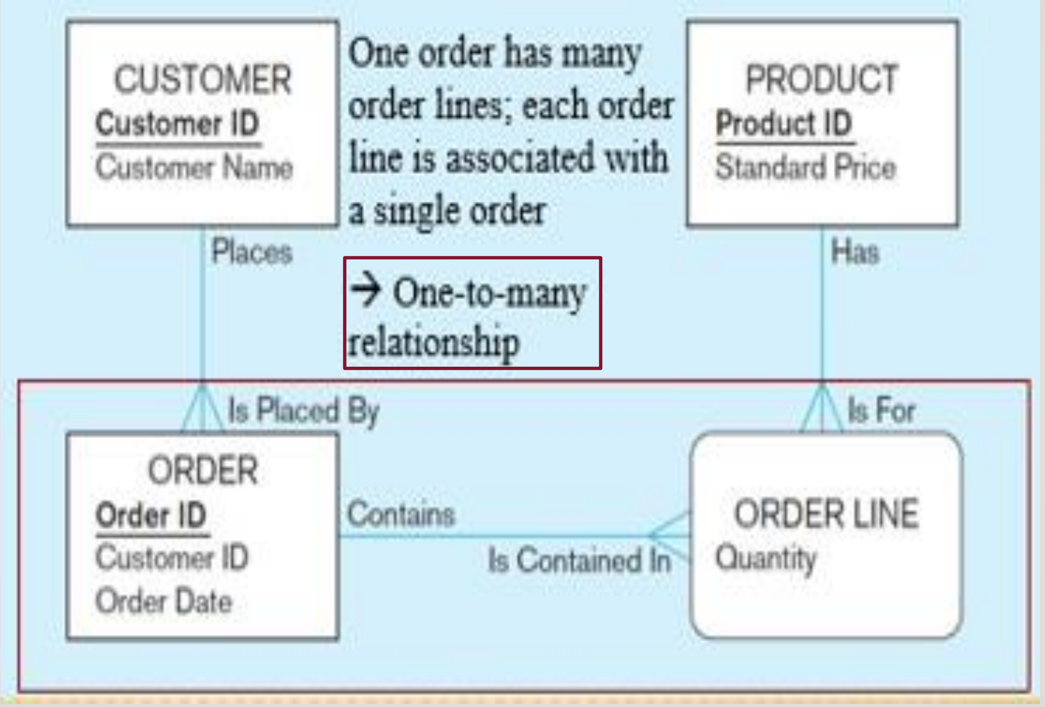
Hubungan antar Entiti (*Relationship*)

- Hubungan *one-to-many* (1:M)
- Hubungan *many-to-many* (M:N)



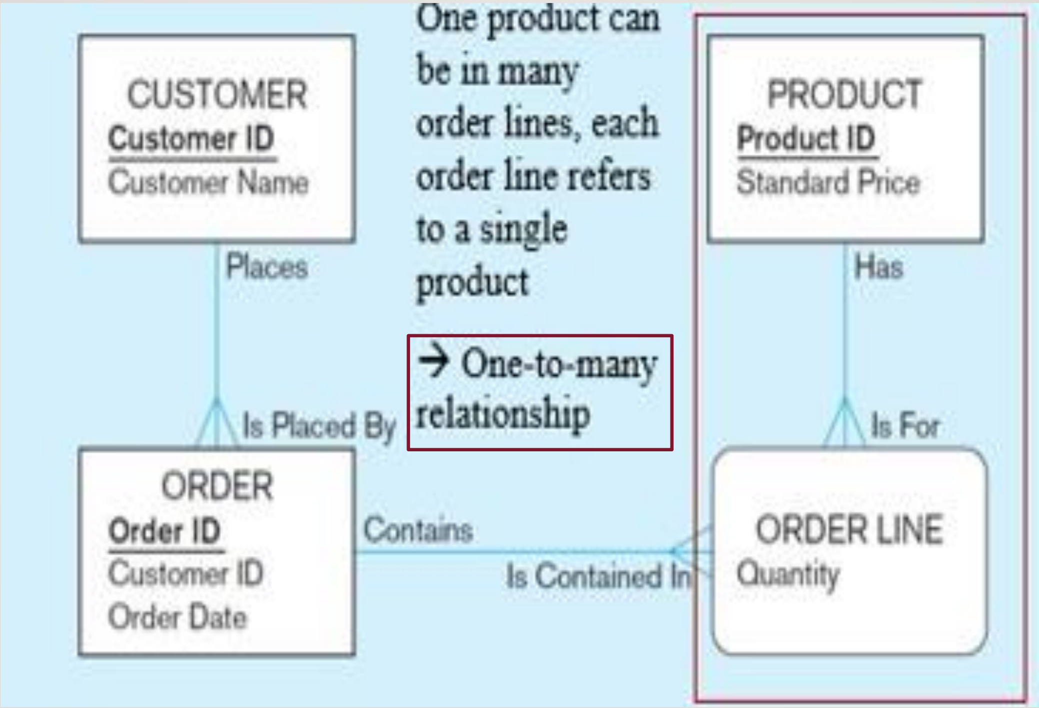
One customer may place many orders, but each order is placed by a single customer

→ One-to-many relationship



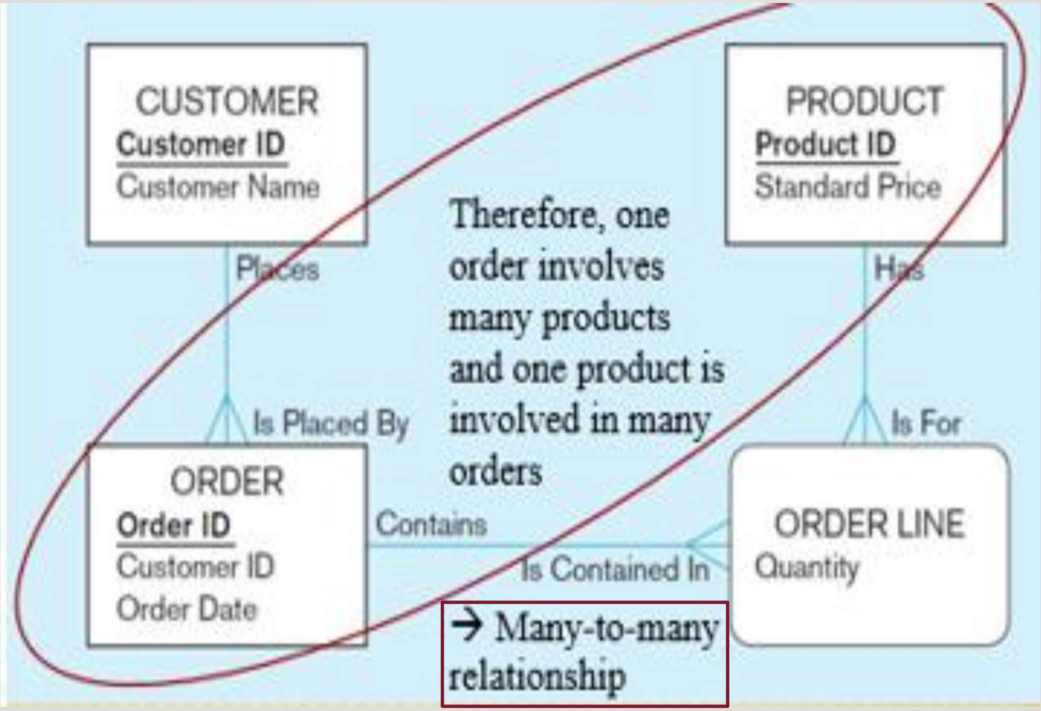
One order has many order lines; each order line is associated with a single order

→ One-to-many relationship



One product can be in many order lines, each order line refers to a single product

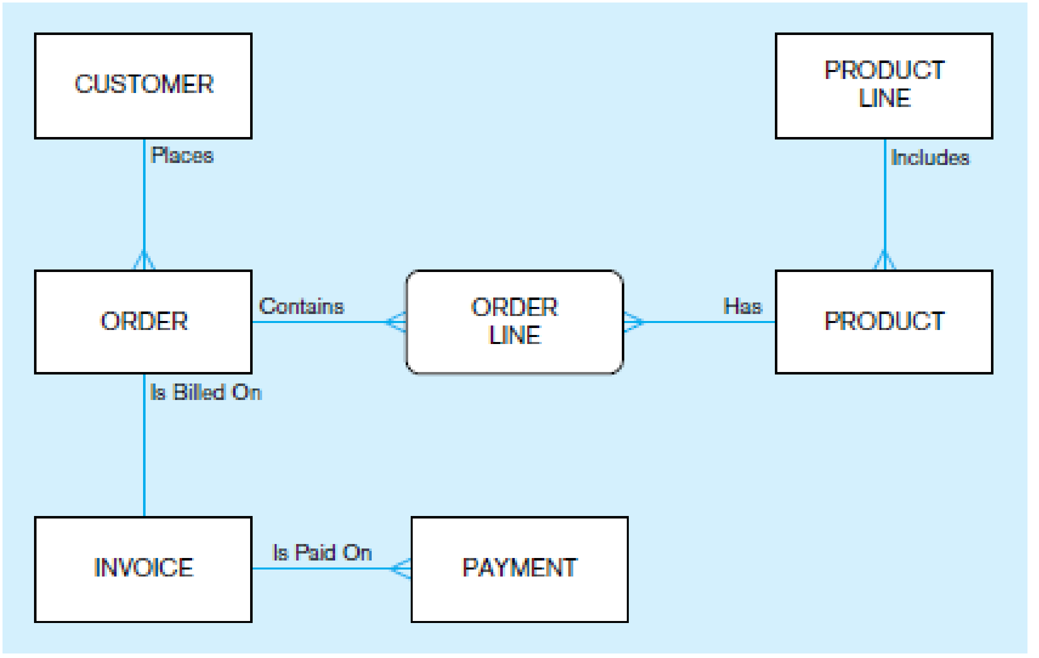
→ One-to-many relationship



Therefore, one order involves many products and one product is involved in many orders

→ Many-to-many relationship

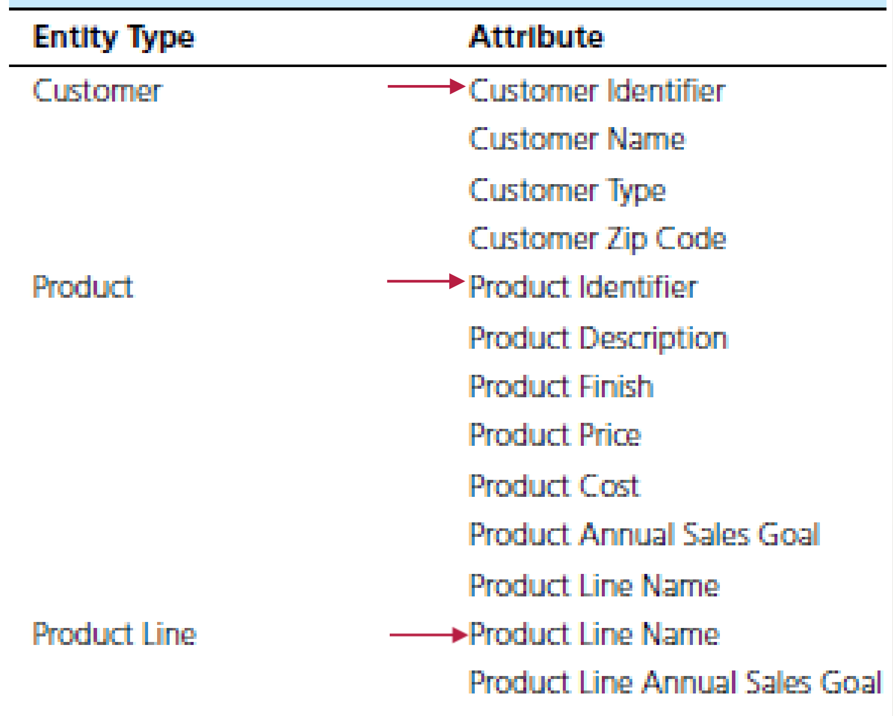
Contoh Model Data

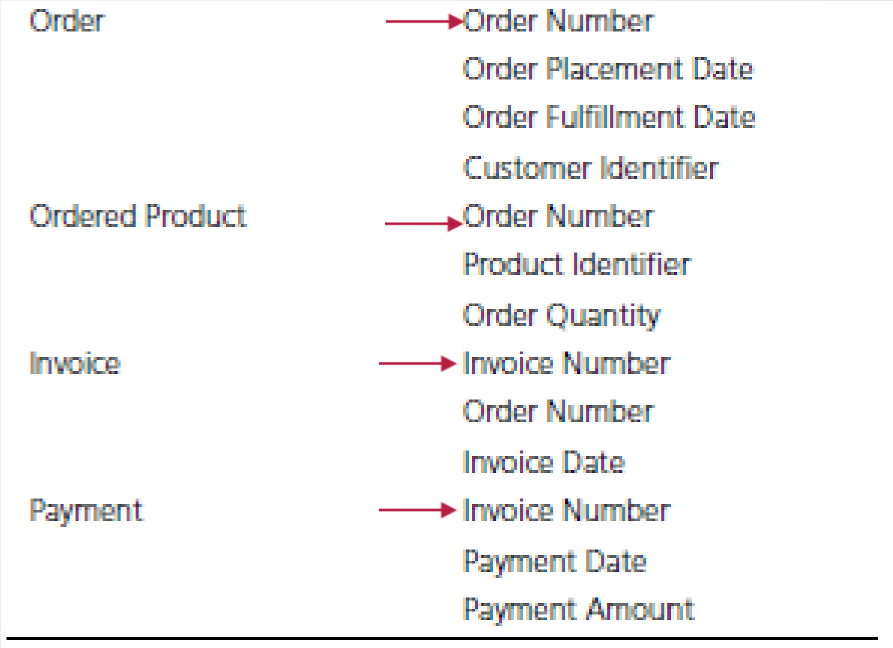


Analisis Kebutuhan Basis Data

1. Setiap CUSTOMER memesan banyak ORDER; sebaliknya, setiap ORDER dipesan oleh seorang CUSTOMER.
2. Setiap ORDER terdiri dari sejumlah ORDER LINE; sebaliknya, setiap ORDER LINE berisi hanya satu ORDER.
3. Setiap PRODUCT mempunyai banyak ORDER LINEs; sebaliknya, setiap ORDER LINE hanya untuk satu PRODUCT.
4. Setiap ORDER dengan satu INVOICE dan setiap INVOICE untuk satu ORDER.

Atribut-Atribut Data untuk Entiti-Entiti dalam Model Data

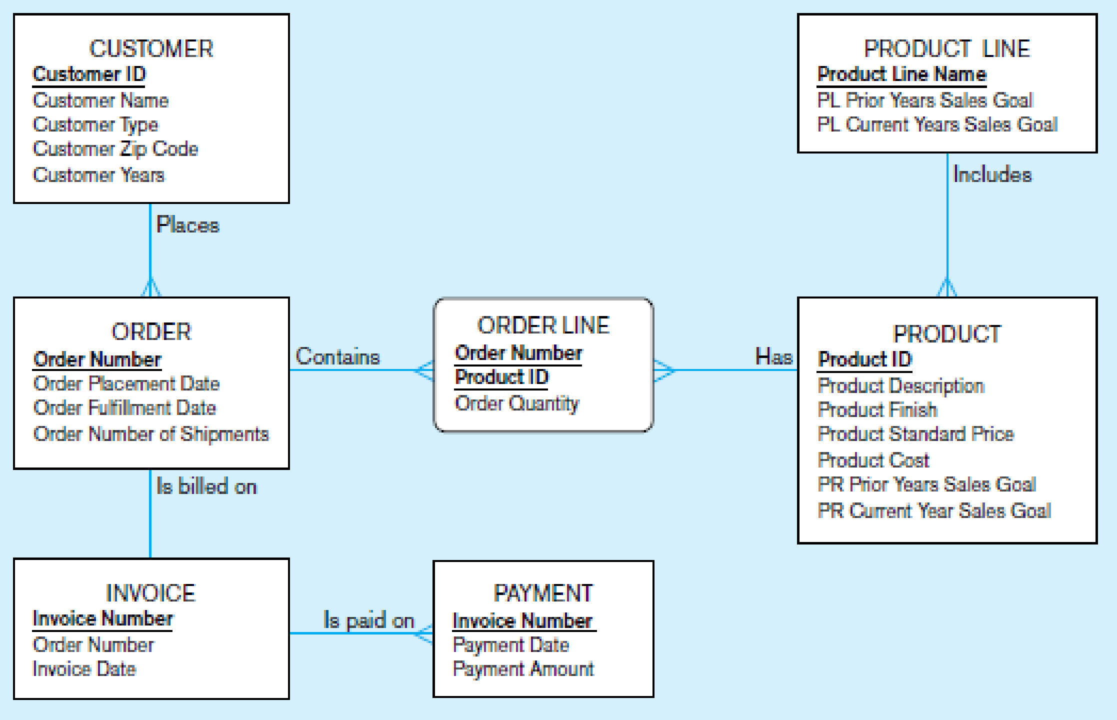




Entity Type	Attribute
Customer	→ Customer Identifier
	Customer Name
	Customer Type
	Customer Zip Code
Product	→ Product Identifier
	Product Description
	Product Finish
	Product Price
	Product Cost
	Product Annual Sales Goal
	Product Line Name
Product Line	→ Product Line Name
	Product Line Annual Sales Goal
Order	→ Order Number
	Order Placement Date
	Order Fulfillment Date
	Customer Identifier
Ordered Product	→ Order Number
	Product Identifier
	Order Quantity
Invoice	→ Invoice Number
	Order Number
	Invoice Date
Payment	→ Invoice Number
	Payment Date
	Payment Amount

→ KEY setiap entiti

Perancangan Basis Data



CUSTOMER
Customer ID
Customer Name
Customer Type
Customer Zip Code
Customer Years

PRODUCT LINE
Product Line Name
PL Prior Years Sales Goal
PL Current Years Sales Goal

ORDER
Order Number
Order Placement Date
Order Fulfillment Date
Order Number of Shipments

ORDER LINE
Order Number
Product ID
Order Quantity

PRODUCT
Product ID
Product Description
Product Finish
Product Standard Price
Product Cost
PR Prior Years Sales Goal
PR Current Year Sales Goal

INVOICE
Invoice Number
Order Number
Invoice Date

PAYMENT
Invoice Number
Payment Date
Payment Amount

Sistem Basis Data

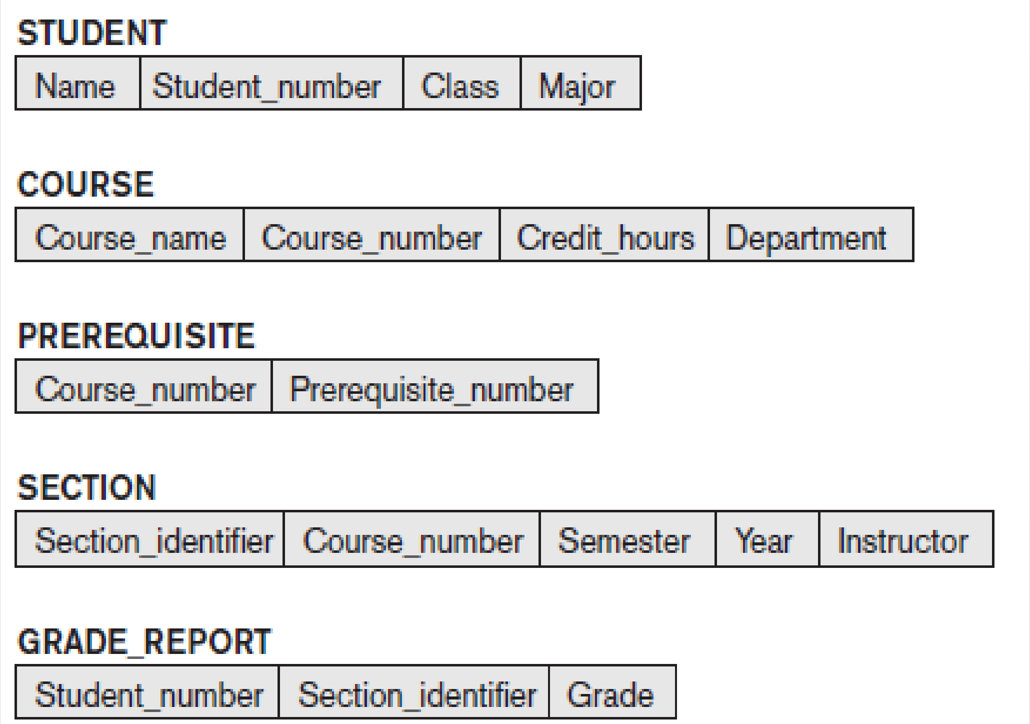
- **client/server** → komputer secara terpusat (*large **centralized** mainframe computers*)
- **distributed architectures** → banyak server (*many large servers **(big data)***)

Dua tipe modul dalam *client/server DBMS*:

- **Client module**
  - ✓ Dirancang untuk alat alat *mobile* , *user workstation*, atau computer pribadi (PC)
  - ✓ Program aplikasi dan *user interfaces* untuk akse basis data
- **Server module**
  - ✓ Menangani penyimpanan data storage, akses, pencarian data, dll

Skema Basis Data (*Database Schema*)

- ✓ Deskripsi suatu basis data yang ditetapkan selama perancangan basis data dan diharapkan tidak sering ada perubahan
  → **diagram skema (*schema diagram*)**



STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

Three-Schema Architecture

- ✓ external or view level → **external schema**
- ✓ conceptual level → **conceptual schema**
- ✓ internal level → **internal schema** (logical and physical schemas)

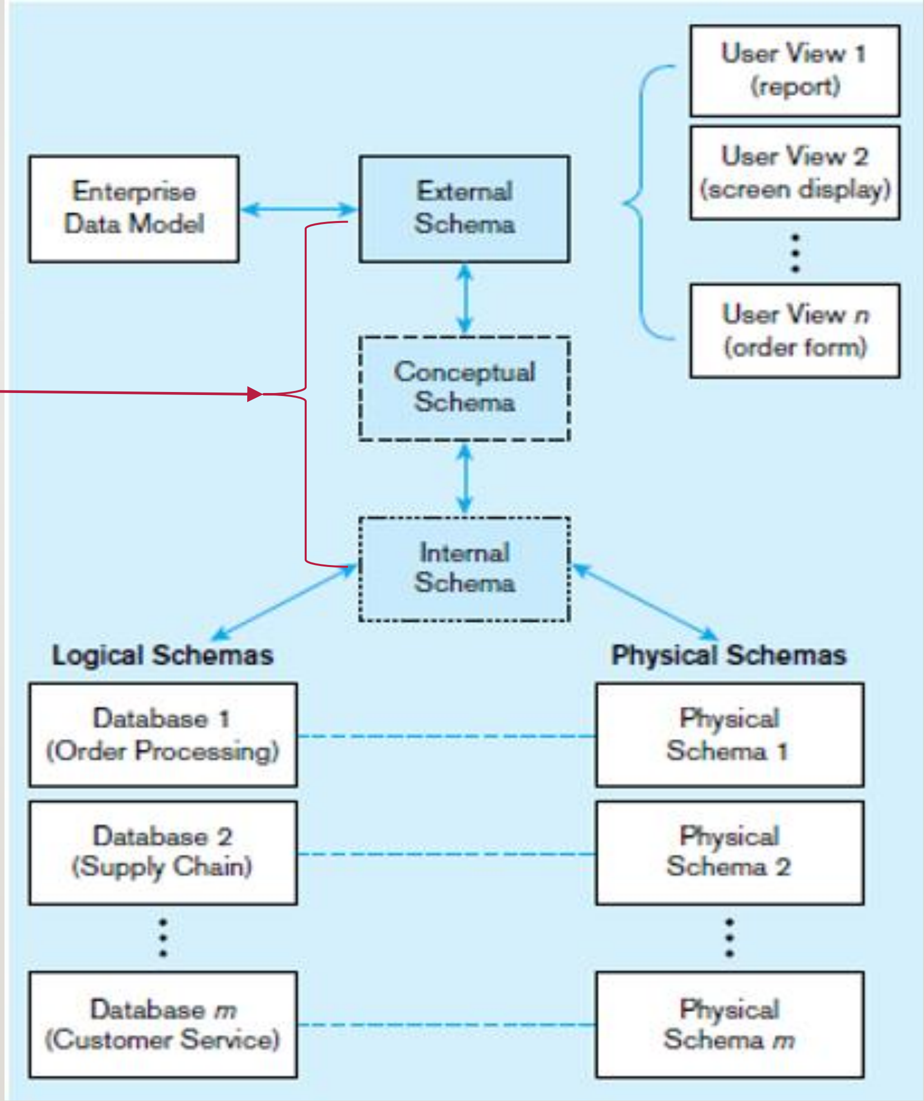
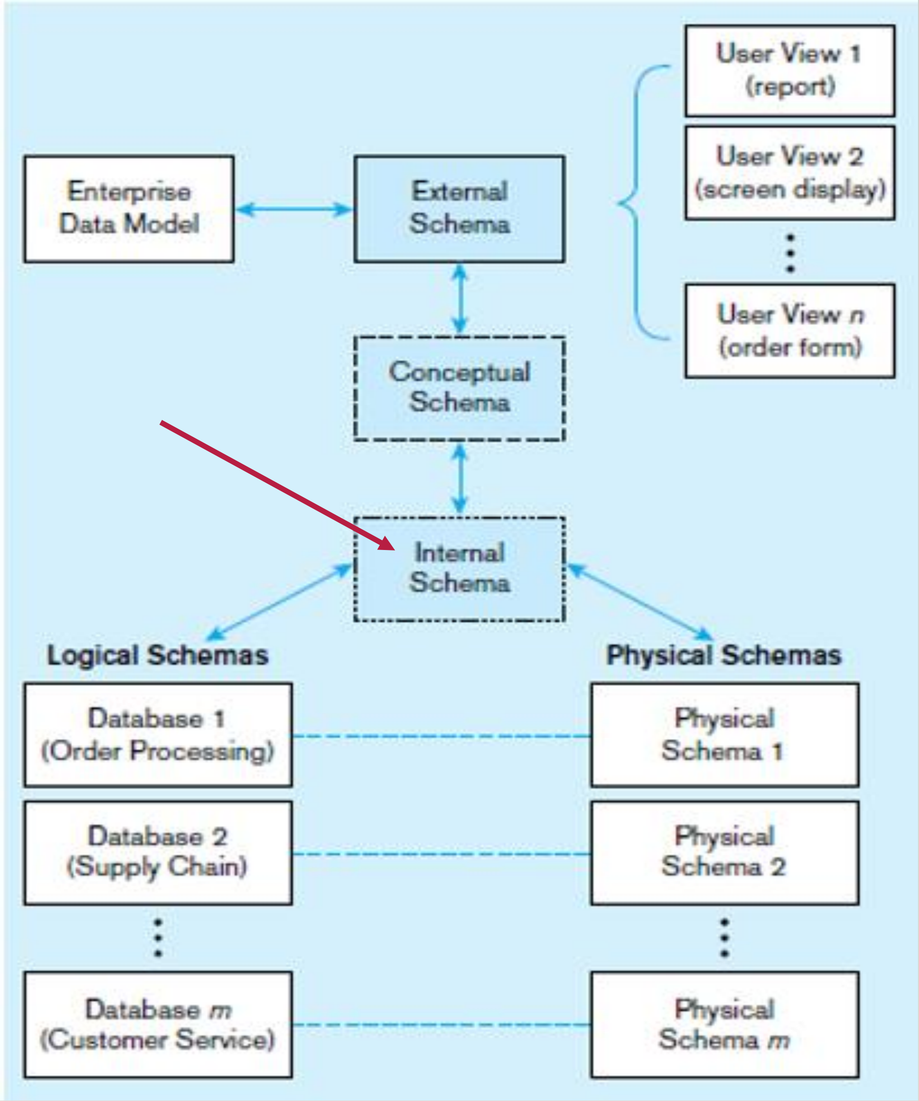


Figure 1-9 Three-schema architecture

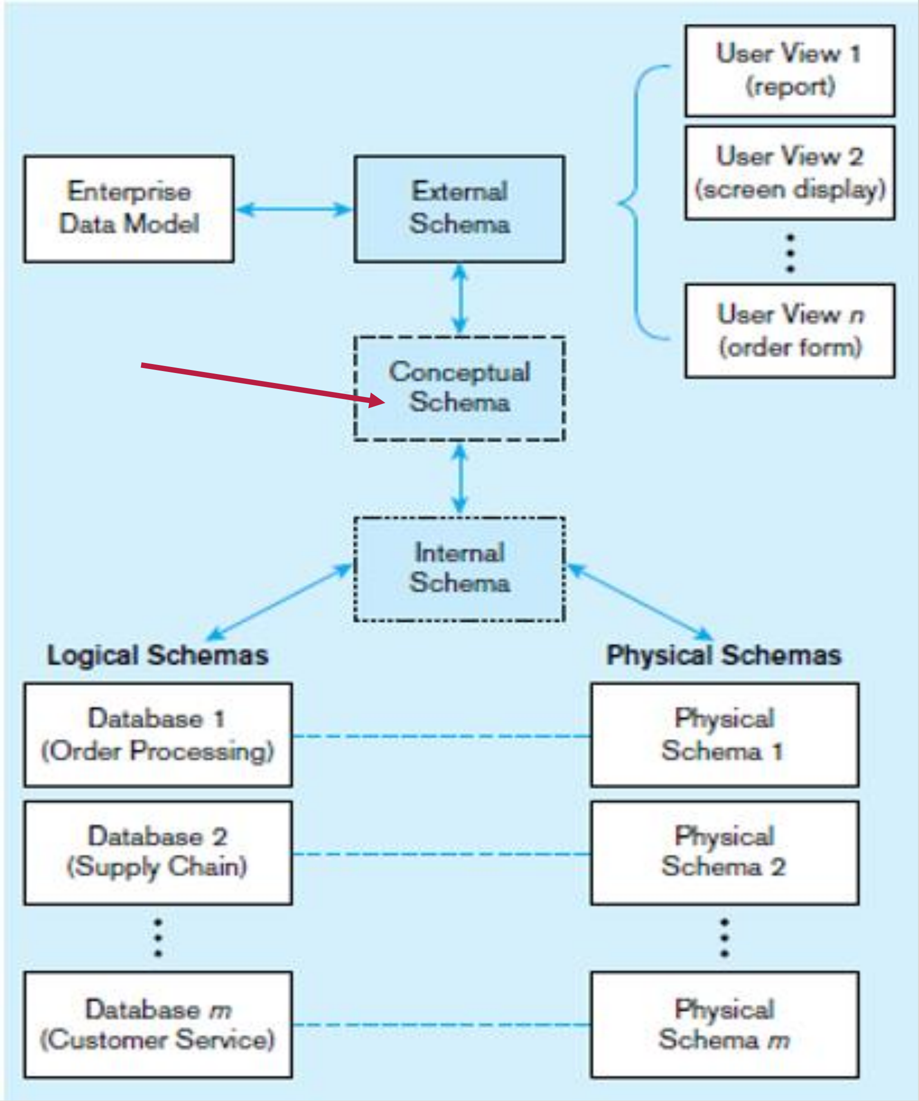
Internal Schema (logical and physical schemas)

- describes the physical storage structure of the database
- uses a physical data model
- describes the data storage and access paths for the database



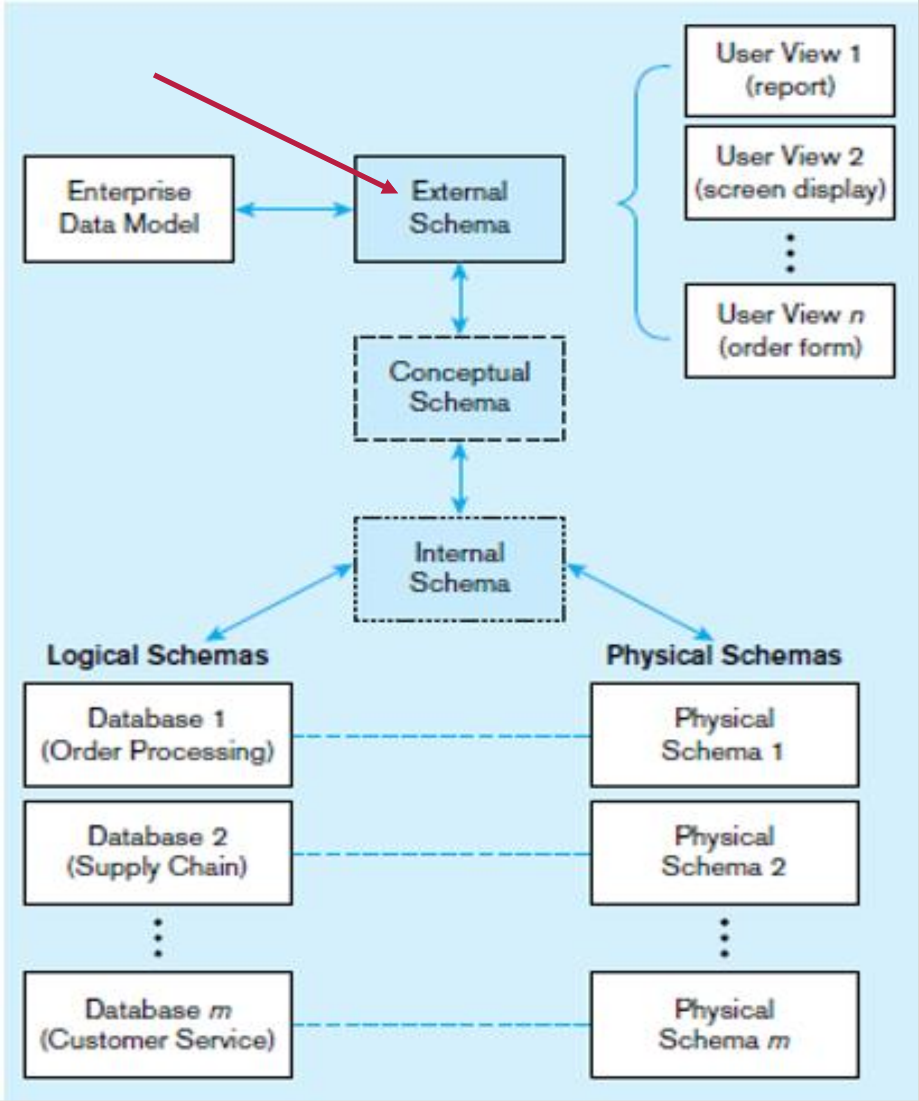
Conceptual Schema

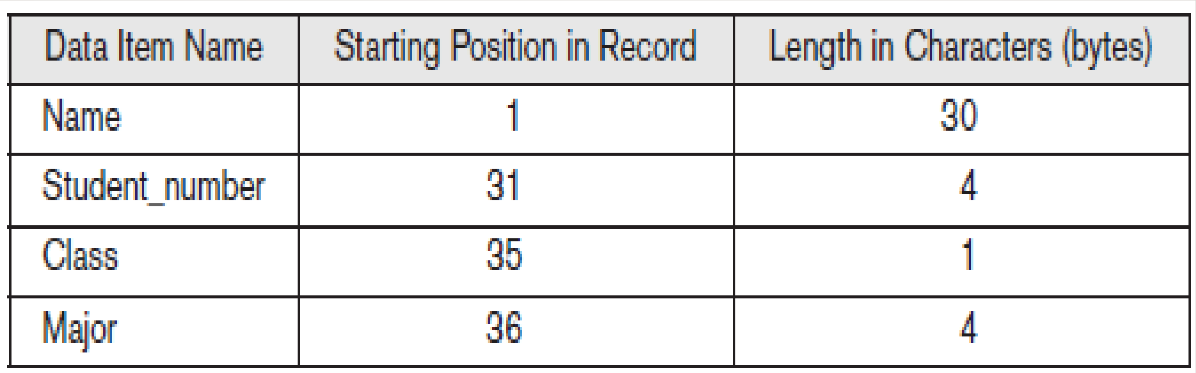
- the structure of the whole database for users.
- hides the details of physical storage structures.
- describes entities, data types, relationships, user operations, and constraints.



External Schema

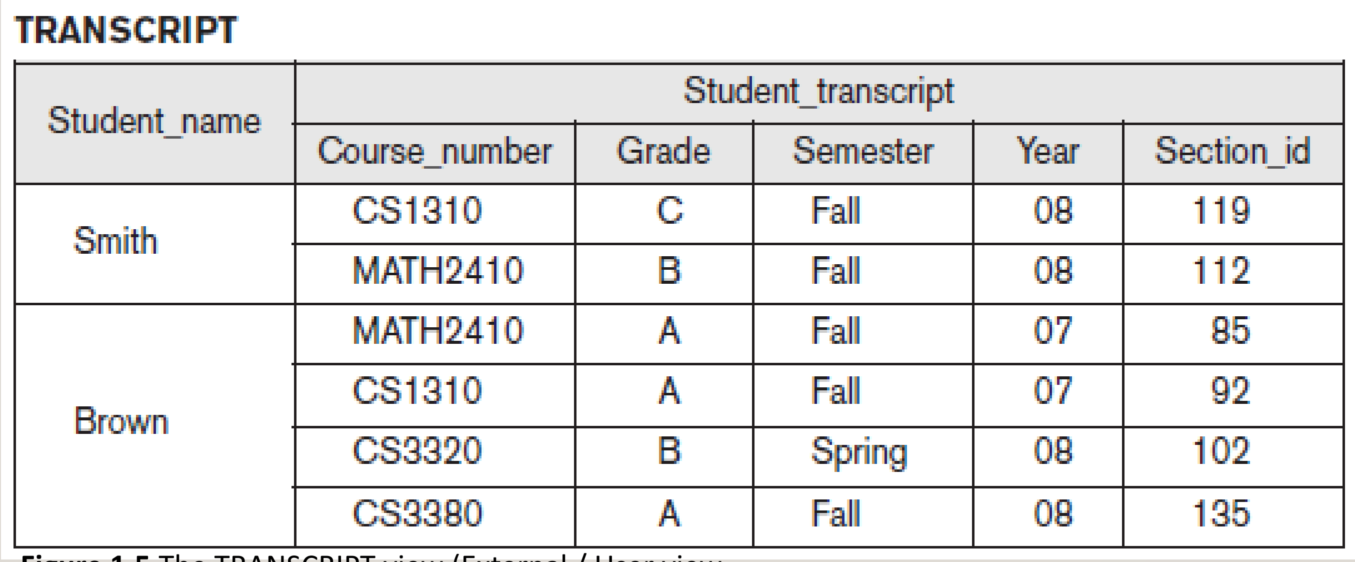
- a number of external schemas or user views.
- each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
- each external schema is implemented using a representational data model, based on an external schema design in a high-level conceptual data model.





Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

Figure 1.4 **Internal storage** format for a STUDENT record



TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

Figure 1.5 The TRANSCRIPT view (External / User view

Data Independence

- ✓ **Logical data independence**: perubahan pada *conceptual schema* **tanpa** harus merubah *external schemas* atau program aplikasi.
- ✓ **Physical data independence**: perubahan pada *internal schema* (reorganisasi *physical files*) **tanpa** harus merubah *conceptual schema* dan *external schemas*.

Bahasa DBMS

1. Data Definition Language (DDL)
   → used by the DBA and by database designers to define schemas
   → used to define both conceptual and external schemas

2. Storage Definition Language (SDL)
   → used to specify the internal schema

3. View Definition Language (VDL)
   → specify user views and their mappings to the conceptual schema

4. Data Manipulation Language (DML)
   → retrieval, insertion, deletion, and modification

Classification of Database Management Systems

The main **data model** used in many current commercial DBMSs is **relational data model → SQL systems**

1. **Centralized** DBMS if the data is stored at a single computer site.
2. **Distributed** DBMS (DDBMS) can have the actual database and DBMS software distributed over many sites connected by a computer network
   - **Homogeneous** DDBMSs use the same DBMS software at all the sites, whereas
   - **Heterogeneous** DDBMSs can use different DBMS software at each site.

Latihan:

Istilah-istilah dalam Basis Data:

1. Entity adalah setiap objek yang akan disimpan.
2. Field / Attribute adalah unsur-unsur pada entity yang berisi data.
3. Record adalah kumpulan unsur-unsur data yang berhubungan pada entity.
4. Data Value adalah data yang disimpan pada setiap unsur.
5. Key adalah pengenal yang unik untuk mengidentifikasi suatu entity.
6. Table adalah kumpulan record untuk setiap entity.

Perancangan Basis Data untuk data berikut:

1. Data Mahasiswa
2. Data Mata Kuliah
3. Data Fakultas
4. Data Program Studi

Tetapkan:

- Entity dan Field/Attribute setiap Entity masing-masing data
- Key masing-masing Entity
- Relation antar Entity

Konsep Dasar Basis Data