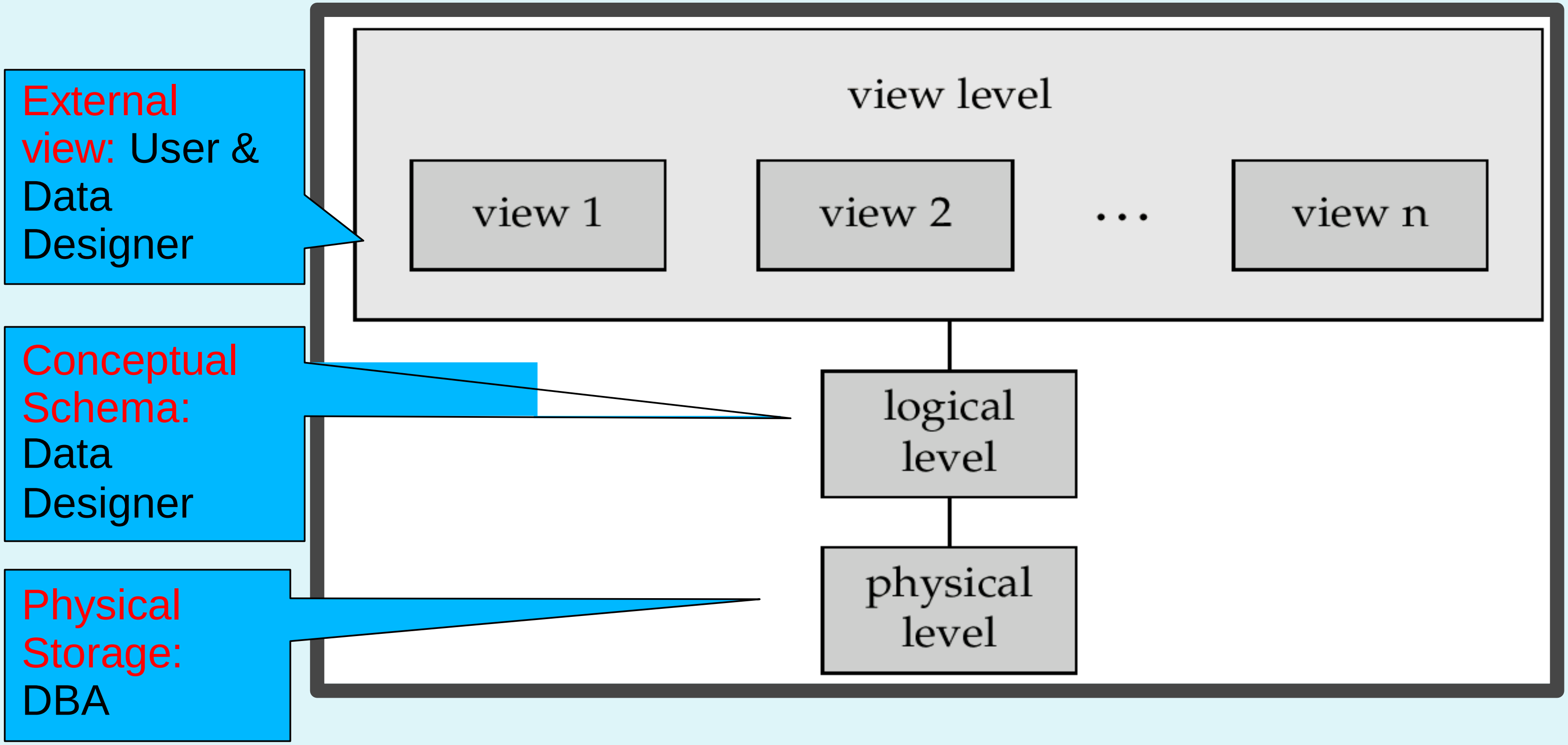
Views of Data



External view: User & Data Designer

Conceptual Schema: Data Designer

Physical Storage: DBA

An architecture for a database system

Views of Data

- **Physical level/Internal level :** The physical representation of the database on the computer. This level describes *how* the data is stored in the database.
    - It includes :
        - Where the data is located
        - File structures
        - Access methods
        - Indexes.

The physical schema is managed by the DBA.

Views of Data

- **Logical level/Conceptual level:** The community view of the database. This level describes *what* data is stored in the database and the relationships among the data.
  - What are the entities and Relationships in organization.
  - What information these entities and relationships should store in database.
  - What integrity constraints/business rules it should have?
  - It consists of the schemas we have described with CREATE TABLE statements.

Views of Data

- **View level/ External Level:** The users view of the database. This level describes that part of the database that is relevant to each user.
    - Each external schema is a combination of base tables and views, tailored to the needs of a single user.
    - It is managed by the data designer and the user.

Components of the DBMS

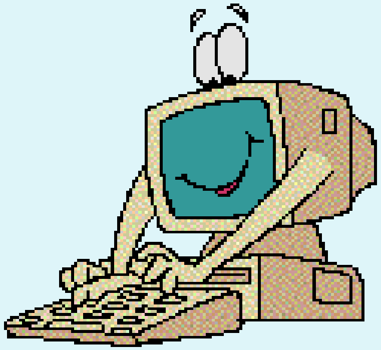
1. Hardware – DBMS and the applications require hardware to run.
   - It can range from PC to mainframe or network of computers.
   - It depends on the organization’s requirements and the DBMS used.
2. Software – It comprises of following :
   - DBMS software itself
   - Application program
   - Operating System including network software
3. Data - Most important component from end-user’s point of view.
   - It acts as a bridge between the machine components and the human component.
   - The database contains both operational data and the metadata.

Components of the DBMS

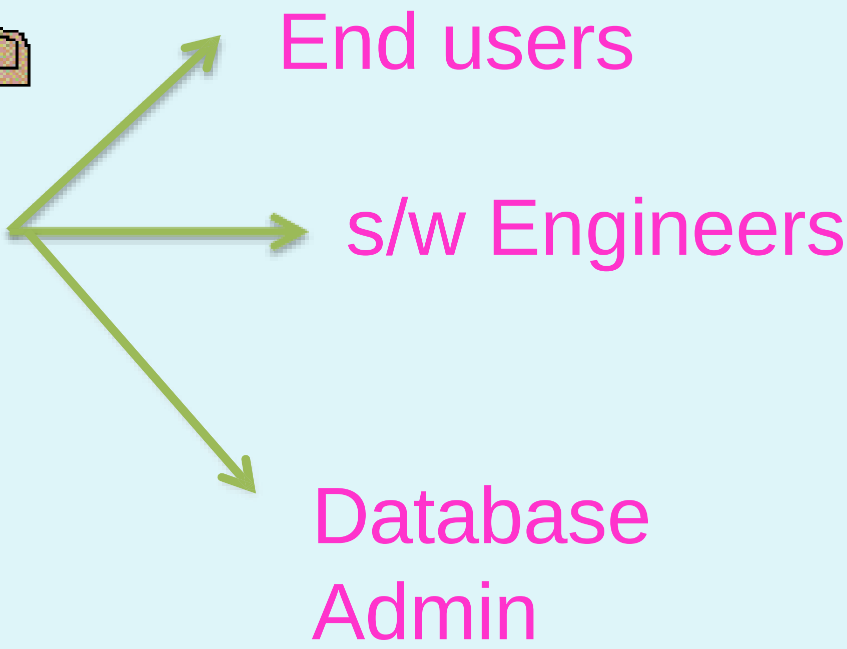
4. Procedures – It refers to the instructions and rules that govern the design and use of the database.
   - Users of the system require documented procedures on how to use/run the system.
   - It may consists of instructions like
     - Log on to the DBMS
     - Use a particular DBMS facility or application program.
     - Start and stop DBMS
     - Make backup copies of the database
     - Handle H/W and S/W failures.
     - Change structure of table to improve performance

5. People – i.e USERS

DBMS Users



Classified into mainly three category



End users

s/w Engineers

Database Admin

A)End User:

- ✓ Deal only with the highest level of abstraction.
- ✓ May not be concerned with the details of the DBMS.
- ✓Involved in updates to the database or queries on the database

❖Categories of end users

I. Casual End Users
II. Naïve or Parametric users
III. Sophisticated users
IV. Specialized users

I. Casual End Users

✓Access data base occasionally when needed
✓They need different information each time
✓They use sophisticated query language to specify their request
✓example: High level Managers who access the data weekly or biweekly.

II. Naive or parametric users

- ✓They make up large section of the database
- ✓They communicate with the database on regular period
- ✓communicate with the system by invoking one of the applications programs that have been written previously
- ✓Their job is to constantly querying and updating database using standard queries
- ✓ This is called canned transaction
- ✓Ex: Bank teller, reservation clerks etc

III. Sophisticated users

✓They include business analyst,scientist, engineers,other thoroughly familiar with the system capability

✓They interact with the system without writing programs,instead they form their requests in a database query language

✓ They submit each query to a query processor

Query processor :

->The query processor in a database management system receives as input a query request in the form of SQL text, parses it, generates an execution plan, and completes the processing by executing the plan and returning the results to the client.

->**Its function is to break down DML statements into instructions that the storage manager can understands**

Storage manager : It is a program module that provides the interface between the low level data stored in the database and application programs and queries submitted to the system

It translates various DML statements into low level file system commands

IV.Specialized users

- Also Sophisticated users who write specialized database application that do not fit into the traditional data processing framework
- Write specialized applications like CAD (Computer Aided design),Multimedia database programs

B) S/w Engineers

I.Application programmer

→Writing database programs in some programming languages (Such as COBOL,C++ or some forth generation languages)

→Application programs access database by issuing the appropriate request to the DBMS.

II. System analysts

Architects, as well as the project leaders, of an information system.

- **It is their job**
  - ▪ Develop solutions to users problems
  - ▪ Determine the technical and operational feasibility of their solutions
  - ▪ Estimate the costs to develop and implement them.

C)Database Administrator

The DBA can be a single person or a team comprising a group of persons

The functions of the DBA include the following:

- ✓ Definition of the Conceptual Schema
- ✓ Definition of the Internal Schema
- ✓ Liaising with users
- ✓ Storage Structure and access method definition
- ✓ Granting of authorization for data access
- ✓ Defining Integrity constraints
- ✓ Monitoring performance and responding to changes in requirements

1. Definition of the Conceptual Schema

➢To decide exactly what information is to be held in the database.

➢Identifies the entities and the information to be recorded about those entities.This process is usually referred to as logical database design.

➢Once the DBA has decided the content of the database at an abstract level, he creates the corresponding conceptual schema

2. Definition of the Internal Schema

- ➢Decide how the data is to be represented in the database. This process is usually referred to as physical database design.
- ➢Having done the physical design, the DBA must then create the corresponding storage structure definition.
- ➢In addition, the DBA must also define the associated conceptual/internal mapping

3. Liaising with users

- ➢The DBA liaises with users to ensure that the data they need is available and to write the necessary external schema.
- ➢In addition, the DBA must also define the associated external/conceptual mapping

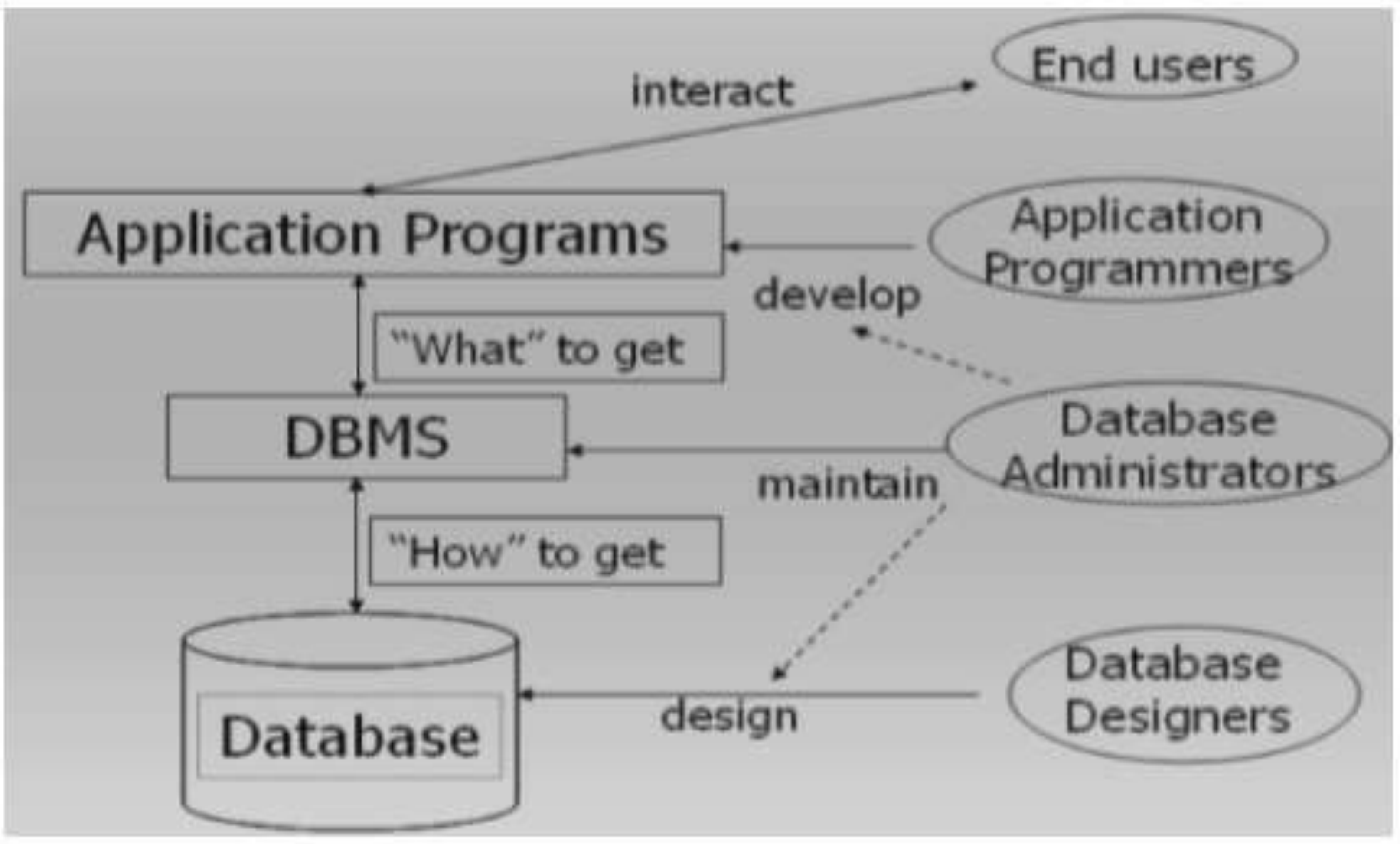
4. Granting of authorization for data access

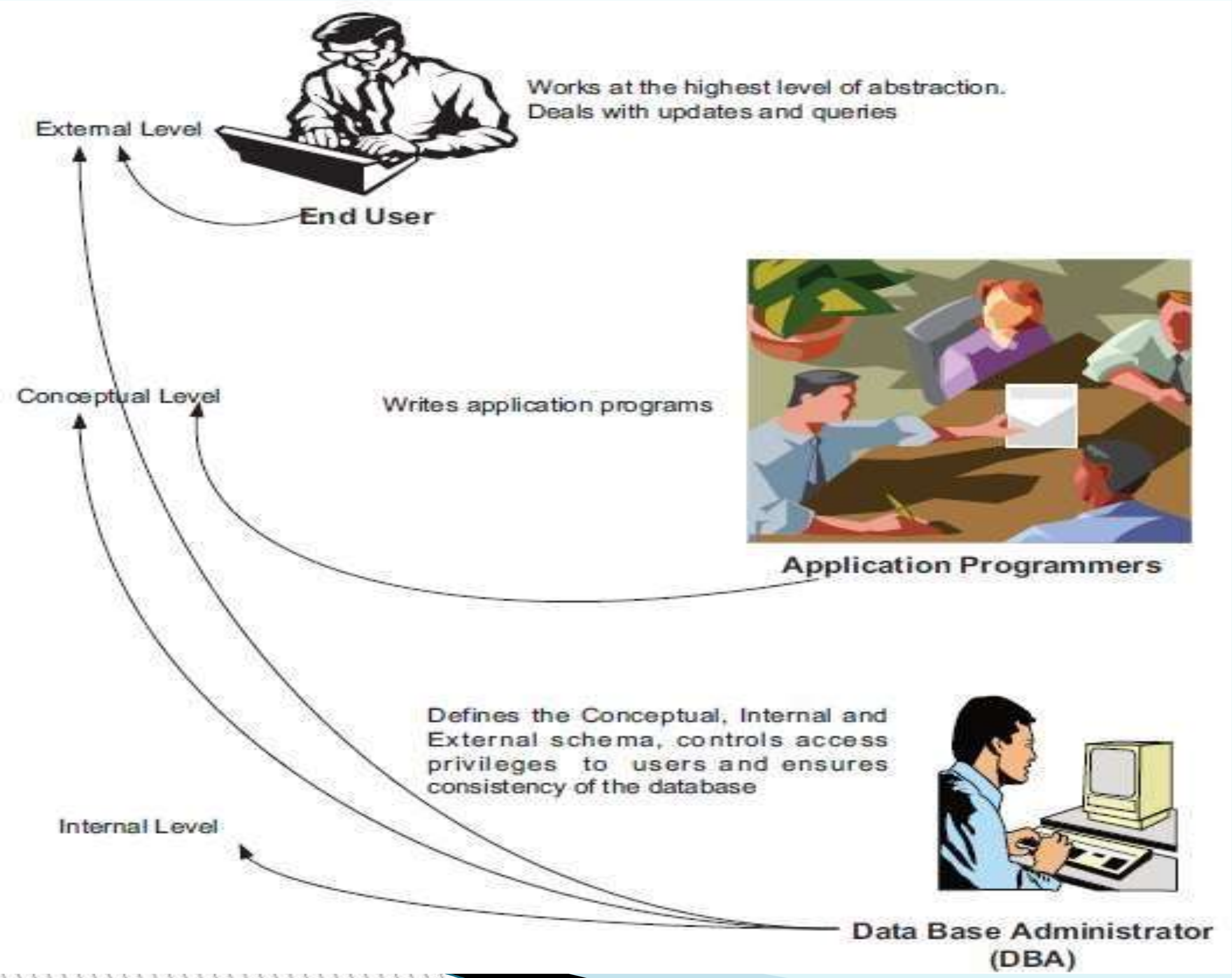
- ➢The granting of different types of authorizations (read, write, etc.) allows the DBA to regulate which parts of the database various users can access

5. Defining Integrity constraints

- ➢The data values stored in the database must satisfy certain consistency constraints

Database Administration's interaction with other users





External Level

Works at the highest level of abstraction.
Deals with updates and queries

End User

Conceptual Level

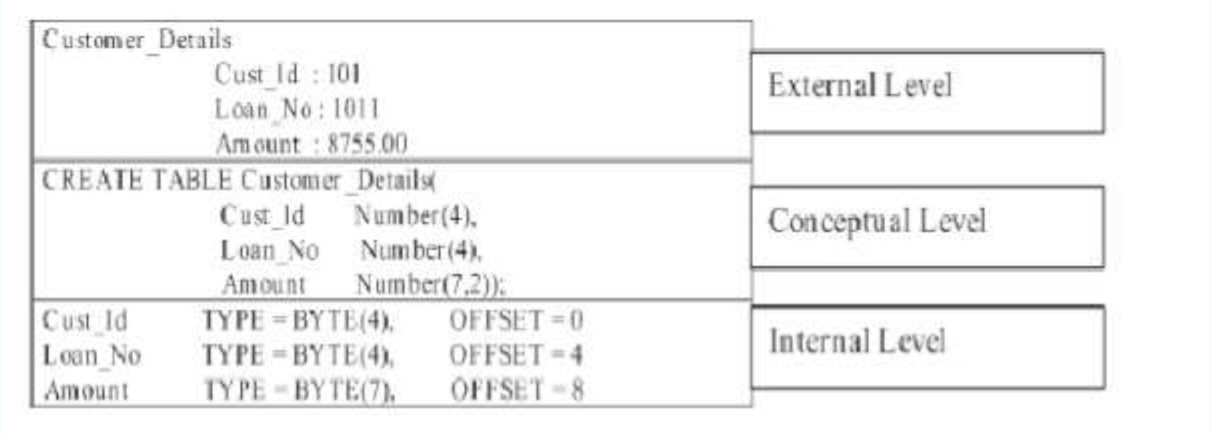
Writes application programs

Application Programmers

Defines the Conceptual, Internal and External schema, controls access privileges to users and ensures consistency of the database

Internal Level

Data Base Administrator (DBA)



```
Customer_Details
                    Cust_Id  : 101
                    Loan_No : 1011
                    Amount  : 8755.00
```

External Level

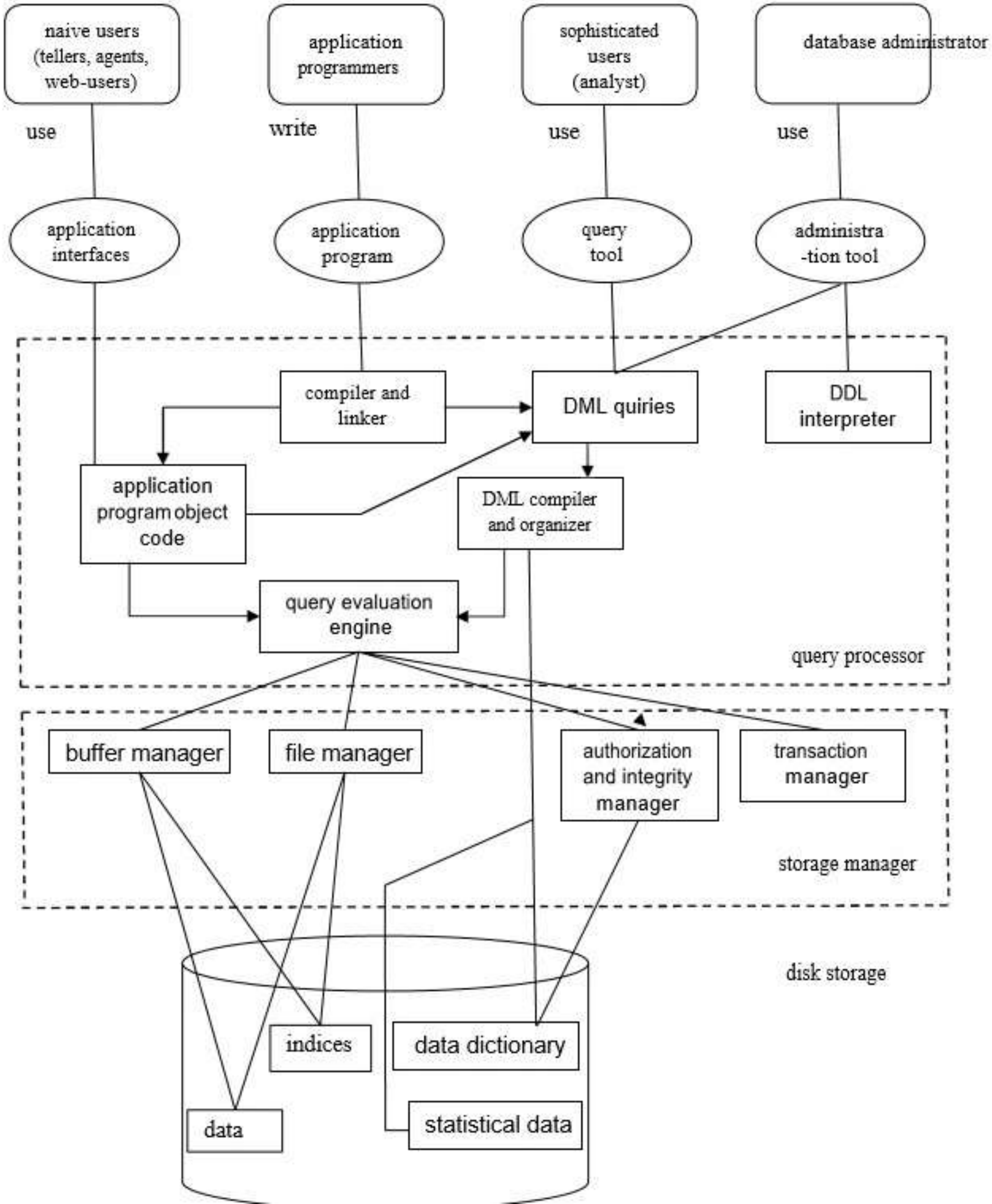
```
CREATE TABLE Customer_Details(
                    Cust_Id       Number(4),
                    Loan_No       Number(4),
                    Amount        Number(7,2));
```

Conceptual Level

```
Cust_Id        TYPE = BYTE(4),       OFFSET = 0
Loan_No        TYPE = BYTE(4),       OFFSET = 4
Amount         TYPE = BYTE(7),       OFFSET = 8
```

Internal Level

Structure of DBMS



DDL

- ✓ create, alter, drop, truncate, rename ,commit

DCL

- ✓ grant , revoke

DML

- ✓ insert, update , delete

A ***transaction*** is a collection of operations that performs a single logical function in a database application

The ***transaction manager*** performs two primary functions:

- ✓backup and recovery
- ✓concurrency control

Backup and recovery ensures that the database remains in a consistent (correct) state despite failures:

system, power, network failures
operating system crashes
transaction failures.

Concurrency-control involves managing the interactions among concurrent transactions.

The ***buffer manager*** loads data into main memory from disk as it is needed by the DBMS, and writes it back out when necessary.

The buffer manager is responsible for:

loading pages of data from disk into a segment of main memory called "the buffer"; sometimes also called the "cache" determining which pages in the buffer get replaced writing pages back out to disk managing overall configuration of the buffer, decomposition into memory pools, page time-stamps, etc.

The ***file manager*** is responsible for managing the files that store data.

- ✓formatting the data files
- ✓managing free and used space in the data files
- ✓defragmenting the data files
- ✓inserting and deleting specific data from the files

The ***authorization & integrity manager*** performs two primary functions:

- ✓data security
- ✓data integrity

Data security:

ensure that unauthorized users can’t access the database
ensure that authorized users can only access appropriate data

Data integrity:

in general, maintains & enforces integrity constraints
maintains data relationships in the presence of data modifications
prevents modifications that would corrupt established data relationships

A given query can be implemented by a DBMS in many different ways.

The ***query optimizer*** attempts to determine the most efficient strategy for executing a given query.

The strategy for implementing a given query is referred to as a *query plan*

DDL, DML , DCL

DDL - Data Definition Language

- o. Used by DB designers to define schema
- o. DDL compile converts DDL statements and generate a set of tables which are stored in.
- o. Data dictionary contains metadata
- o. e.g - SQL, ORCALE

DML - Data Manipulation Language

- o. For accessing and manipulation thedata
- o. DML also known as query language
- o. Procedural :- user specifies what data is required and how to get those data.

DBMS

Basic Concepts – Data , information, database and DBMS

DDL

- ✓ create, alter, drop, truncate, rename ,commit

DCL

- ✓ grant , revoke

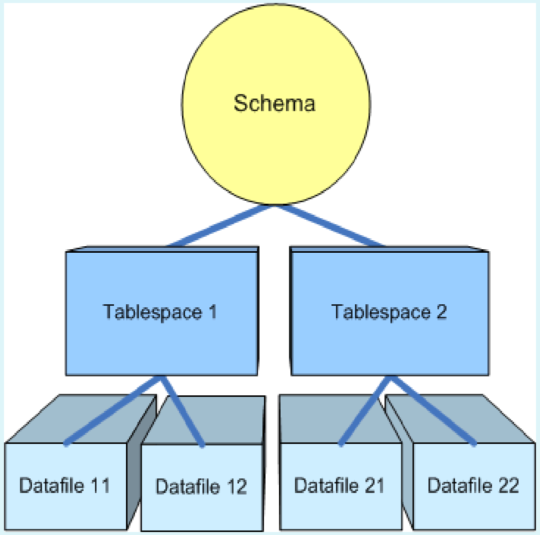
DML

- ✓ insert, update , delete

Data Models and Types of Data Models

Definition - What does Database Model mean?

Data Model



A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed

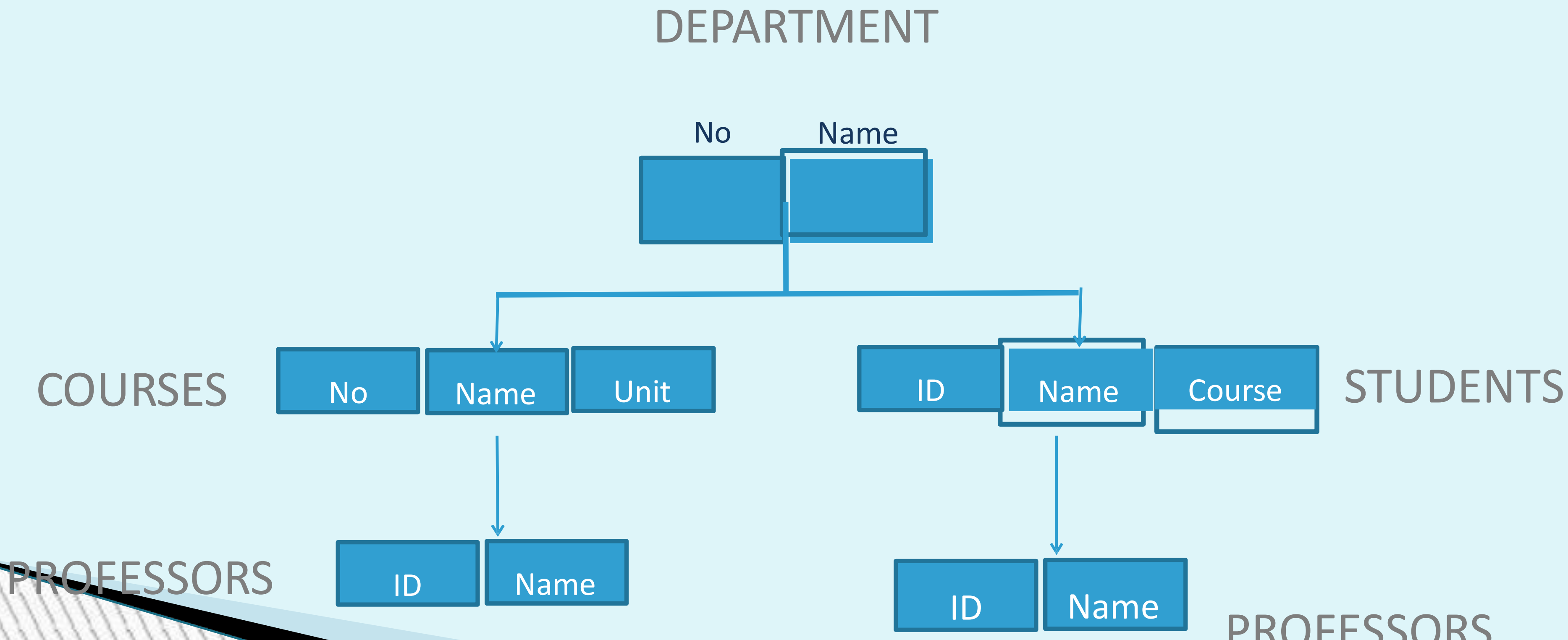
A Database model defines the logical design of data. The model also describe the relationships between different parts of the data.

A database model refers to the logical structure, representation or layout of a database and how the data will be stored, managed and processed within it. It helps in designing a database and serves as blueprint for application developers and database administrators in creating a database

Hierarchical Model

- In the hierarchical model, data is organized as an inverted tree, Each entity has only one parent but can have several children. At the top of the hierarchy , there is one entity, which is called the root

- The hierarchical data model organizes data in a tree structure



- tree like structure
- one parent may have multiple child but
- the child will have only parent

Hierarchical Model

- ✓ The hierarchical model was the first database model developed to overcome the limitations of the traditional file system

- ✓ Deletion of parent results in deletion of child records

- ✓ The hierarchical models was developed in the 1960 to manage large amount of data for complex projects such as the Apollo rocket that landed on the moon in 1969.

- ✓ Its basic logical structure is represented by an upside-down tree

- ✓ The hierarchical structure contains levels or segments.

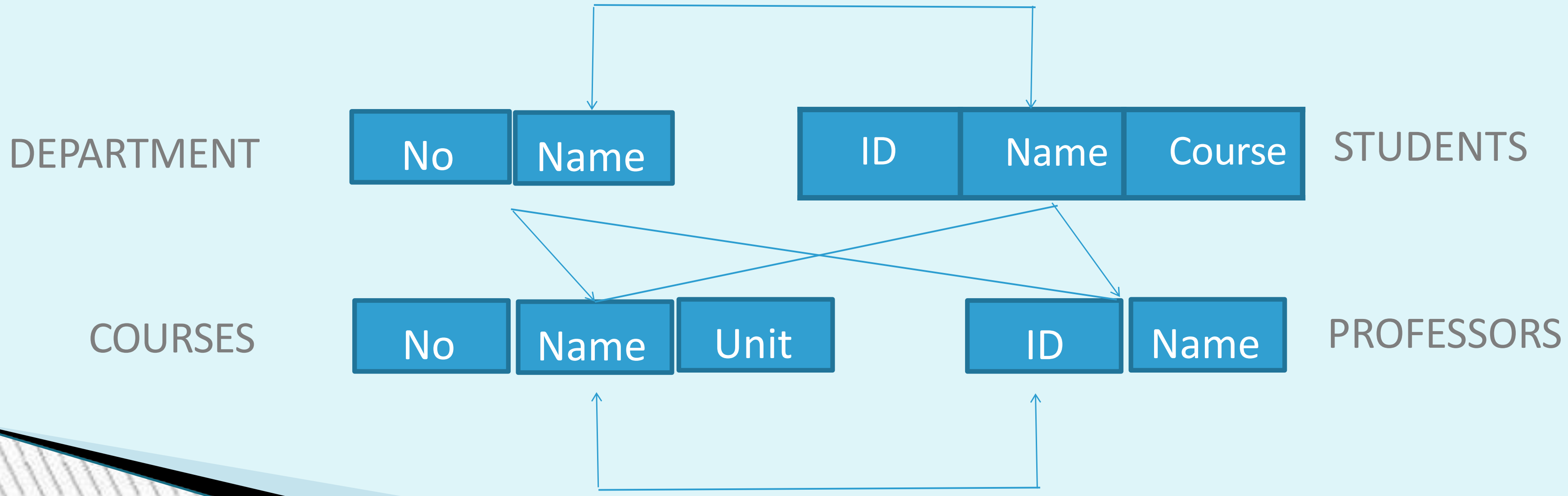
- ✓ The hierarchical model depicts a set of one-to many relationalship between a parents and its children segments

Disadvantages of hierarchical model

1. Complexity of Implementation
2. Difficulty in Management
3. Complexity of Programming
4. Poor Portability
5. Database Management Problems
6. Lack of structural independence
7. Programs Complexity
8. Operational Anomalies
9. Implementation Limitation

Network model

- In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths
- This model was developed to overcome the problems of hierarchical model

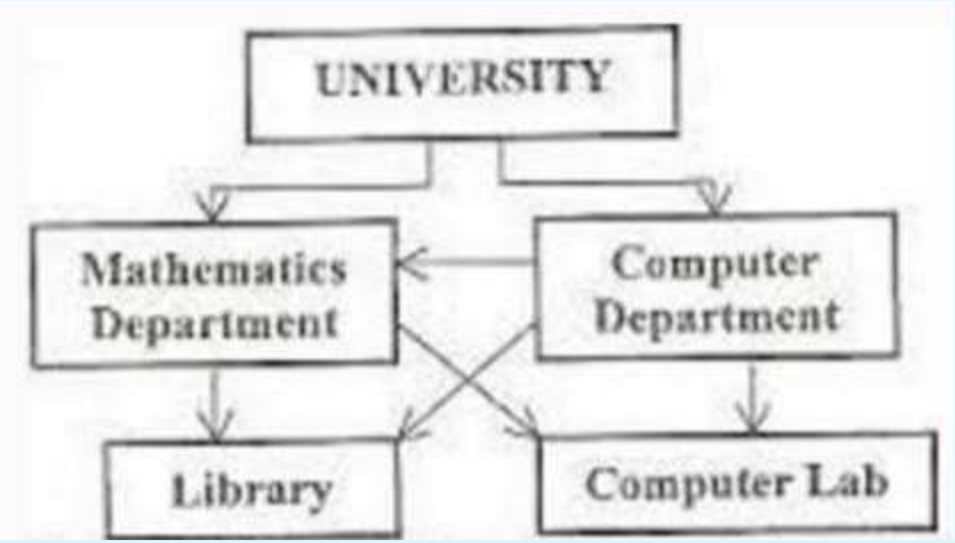


Network model

- Graph structure
- Many-to-many relationships between linked records,
- Some data might have multiple parents
- Allow more connation between nodes
- ex. A employee work for two department is not possible in hierarchical mode, but here it is possible
- The network model is a database model conveyed as a flexible way of representing objects and their relationships

✓Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy

✓The network model replaces the hierarchical model with a graph thus allowing more general connections among the nodes. The main difference of the network model from the hierarchical model is its ability to handle many to many relationships.



Advantages of network model

- ✓ Conceptual simplicity
- ✓ Ability to handle many relationships
- ✓ Easy to access
- ✓ Data integrity (current data, specify , fixed)
- ✓ Data independence
  -Application programmers work independently of the data
  -Any change made to the data do not affect application program

DISADVANTAGES

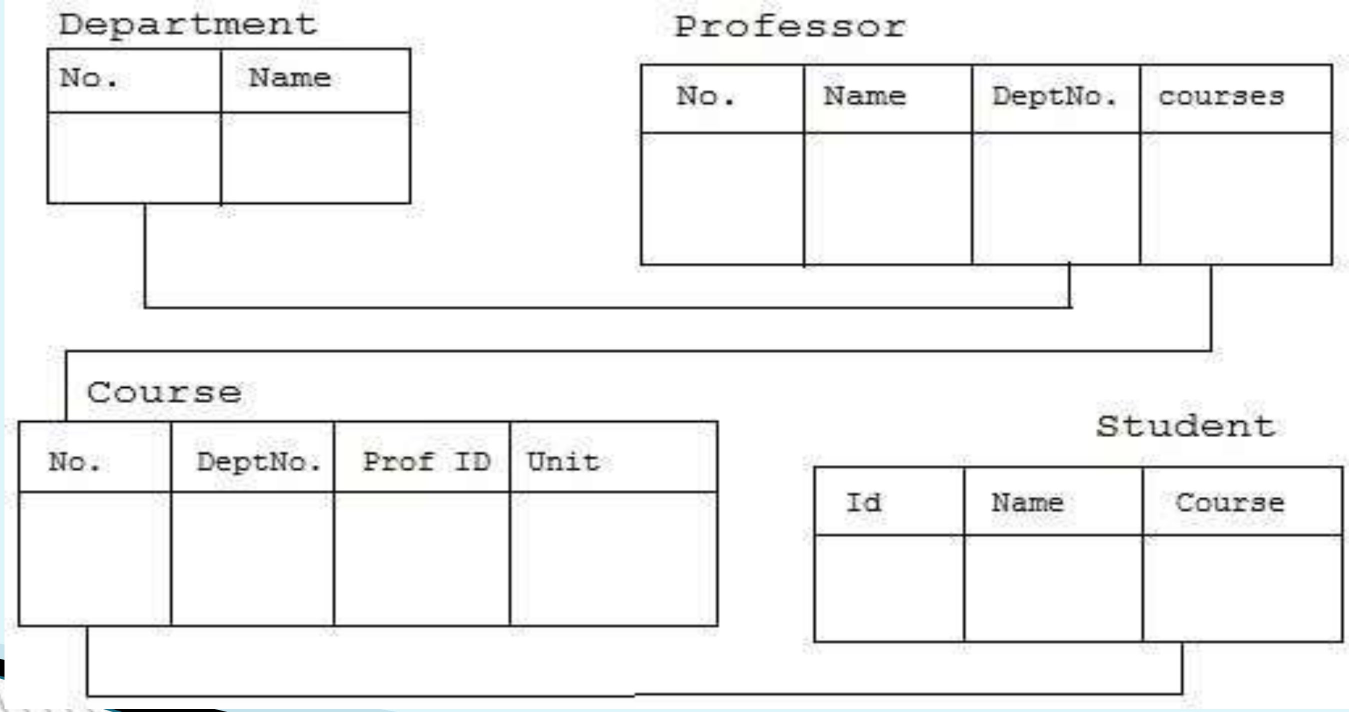
1.) System complexity- All the records are maintained using pointers and hence the whole database structure becomes very complex.

2.) Operational Anomalies- The insertion, deletion and updating operations of any record require large number of pointers adjustments.

3.) Absence of structural independence-structural changes to the database is very difficult

Relational model in DBMS

- In relational model, the data and relationships are represented by collection of inter-related tables. Each table is a group of column and rows, where column represents attribute of an entity and rows represents records.

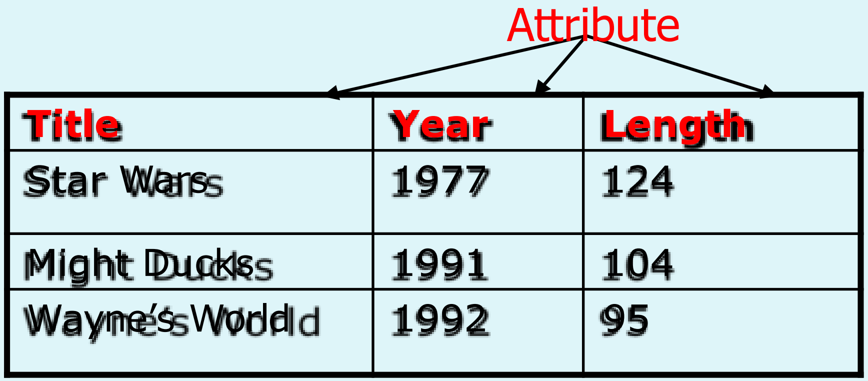


Relational model in DBMS

- A relational database is a collection of one or more relations where each relation is a table with rows and columns

Table = *relation*.

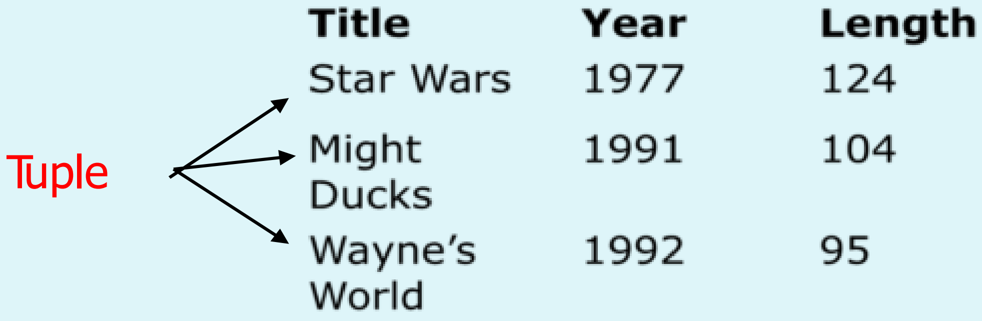
Column headers = *attributes*



Title	Year	Length
Star Wars	1977	124
Might Ducks	1991	104
Wayne's World	1992	95

Table = *relation*.

Rows = *tuple*



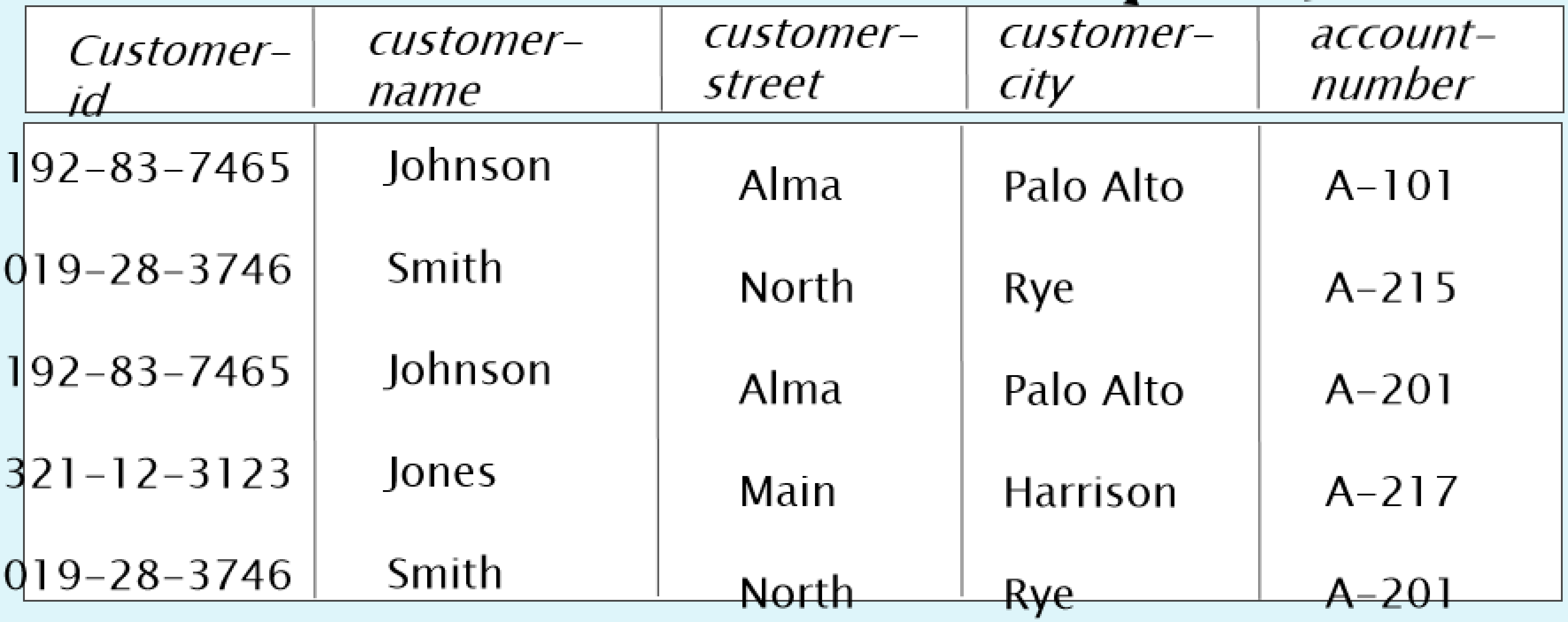
Title	Year	Length
Star Wars	1977	124
Might Ducks	1991	104
Wayne's World	1992	95

Relational model in DBMS

- ✓ First proposed by Dr. E.F Codd of IBM Research in 1970
- ✓ Implemented through Relational DBMS (RDBMS)
- ✓ Represent database as a collection of relations
- ✓ Informally, each relation look like a table of values
- ✓ Relation database matches data by using common characteristics found within the data set
- ✓ The resulting groups of data are organized and are much easier for people to understand.

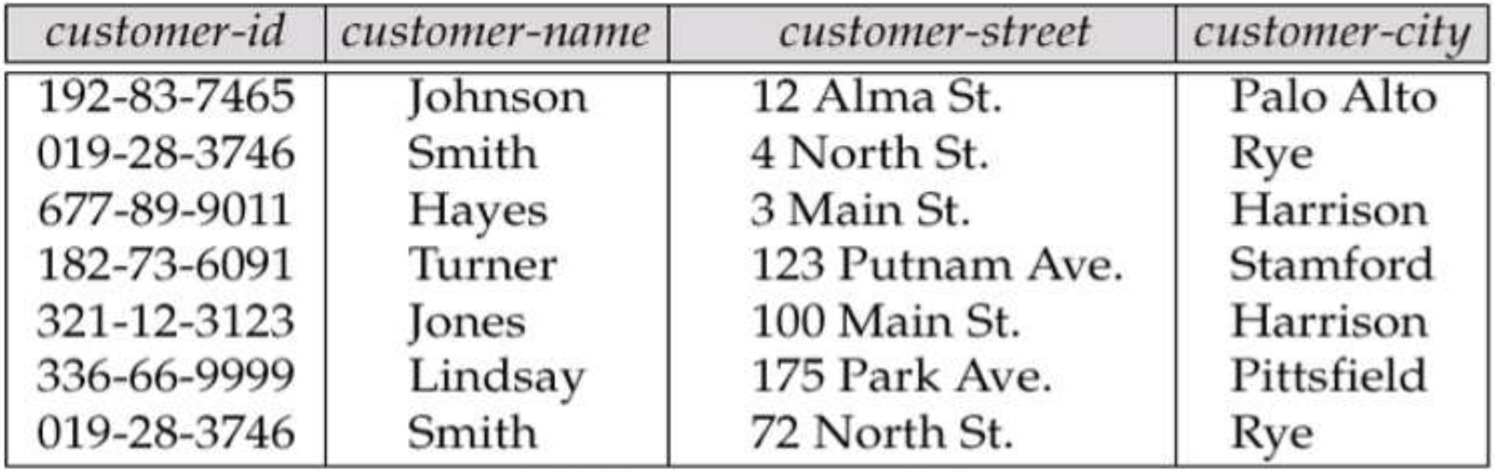
- Example of tabular data in the relational model

Attributes



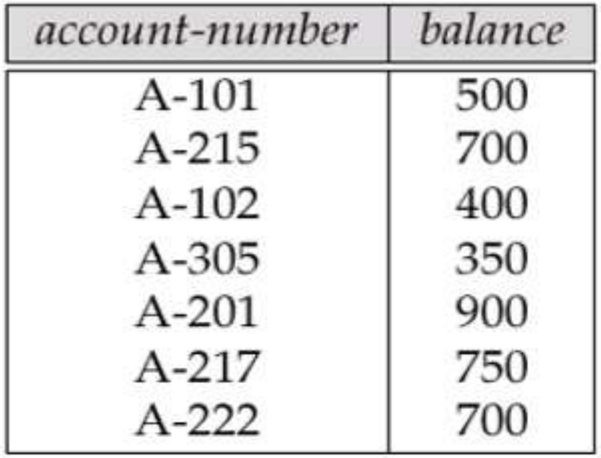
Customer–id	*customer–name*	*customer–street*	*customer–city*	*account–number*
192–83–7465	Johnson	Alma	Palo Alto	A–101
019–28–3746	Smith	North	Rye	A–215
192–83–7465	Johnson	Alma	Palo Alto	A–201
321–12–3123	Jones	Main	Harrison	A–217
019–28–3746	Smith	North	Rye	A–201

A Sample Relational Database



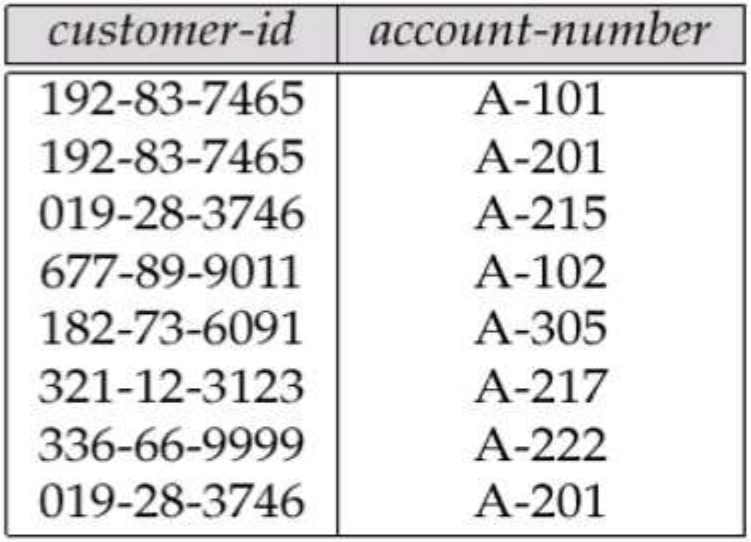
customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table



account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table



customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Advantages

- – Structural independence: changes in the relational data structure do not affect the DBMS's data access in any way
- – Improved conceptual simplicity by concentrating on the logical view
- – Easier database design, implementation, management, and use
- – Ad hoc query capability –SQL
- – Powerful database management system

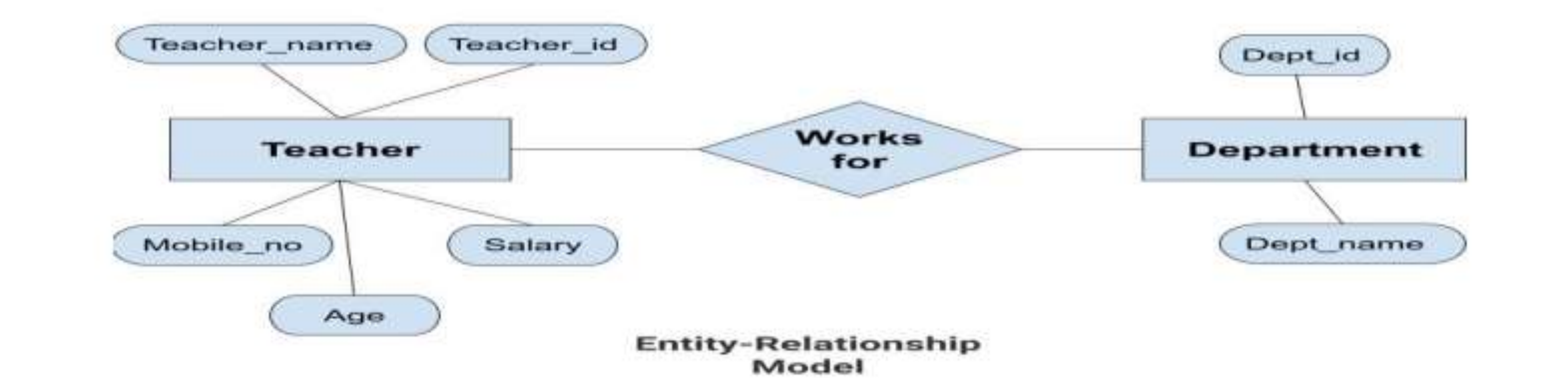
Disadvantages

- – Substantial hardware and system software overhead
- – Can facilitate poor design and implementation
- – May promote "islands of information" problems

Entity Relationship Model

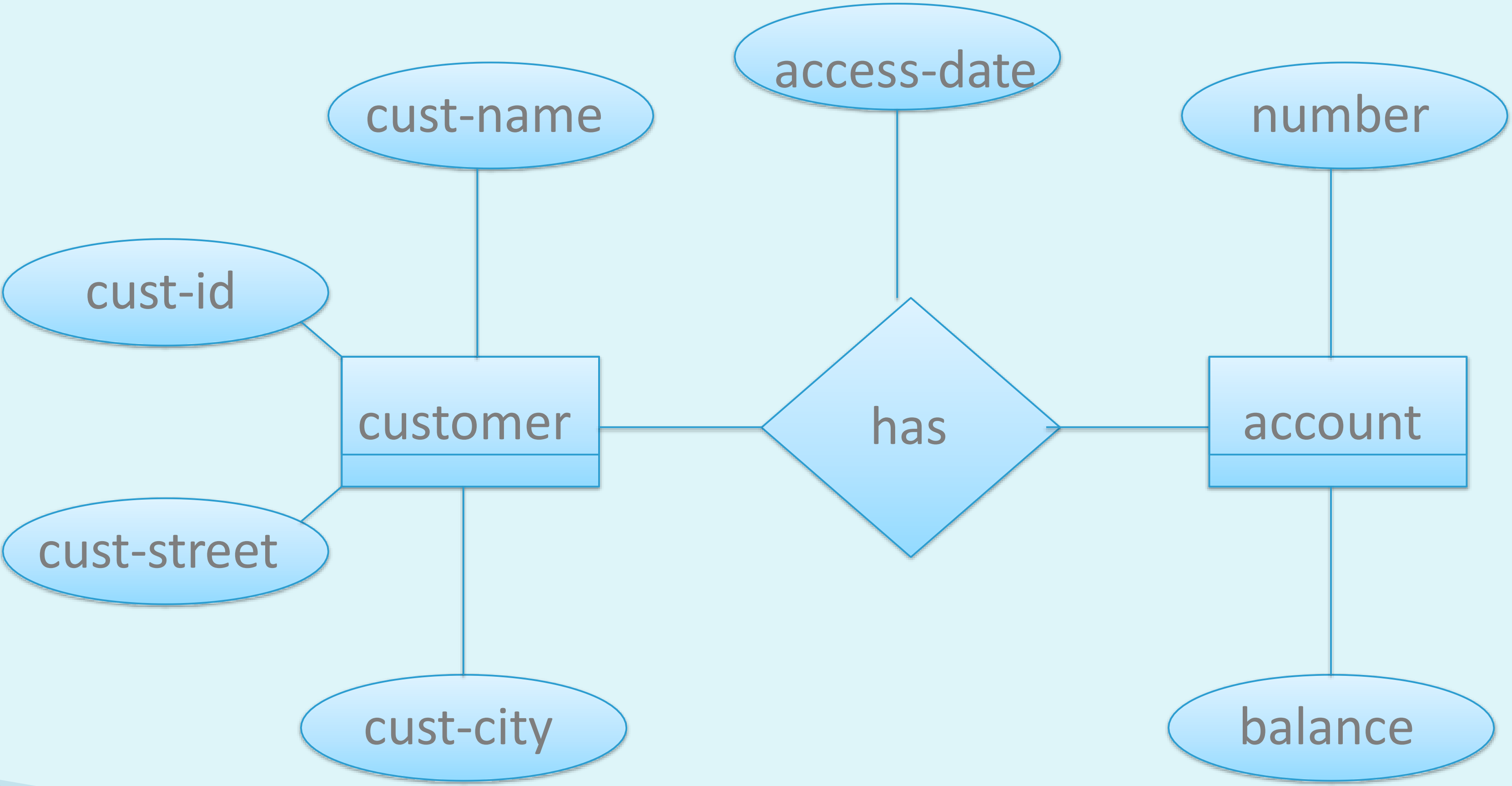
- An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data- an object or concept about which data is stored.
- Widely accepted and adapted graphical tool for data modeling
- Introduced by Peter Chen in 1976
- Graphical representation of entities and their relationships in a database structure

Example:



Entity-Relationship Model

ER Model Diagram



Advantages of ER Model

- **Conceptually it is very simple:** ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.

- **Better visual representation:** ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.

- **Effective communication tool:** It is an effective communication tool for database designer.

- **Highly integrated with relational model:** ER model can be easily converted into relational model by simply converting ER model into tables.

- **Easy conversion to any data model:** ER model can be easily converted into another data model like hierarchical data model, network data model and so on

Disadvantages of ER Model

- **Limited constraints and specification**

- **Limited relationship representation:** ER model represents limited relationship as compared to another data models like relational model etc.

- **No industry standard for notation:** There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.

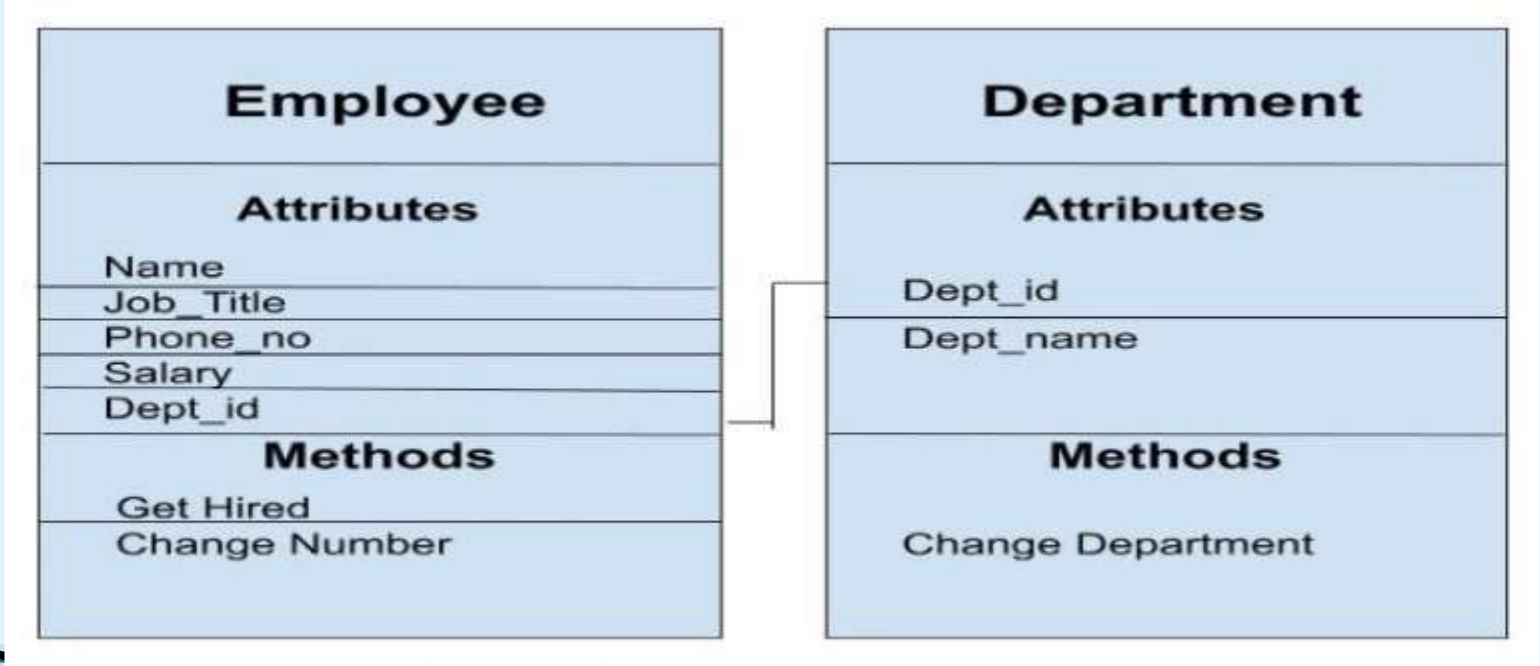
- **Hidden information:** Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

- **Limited manipulation:**There is no data manipulation language available for an E- R model as it is a largely abstract concept

Object Oriented Model

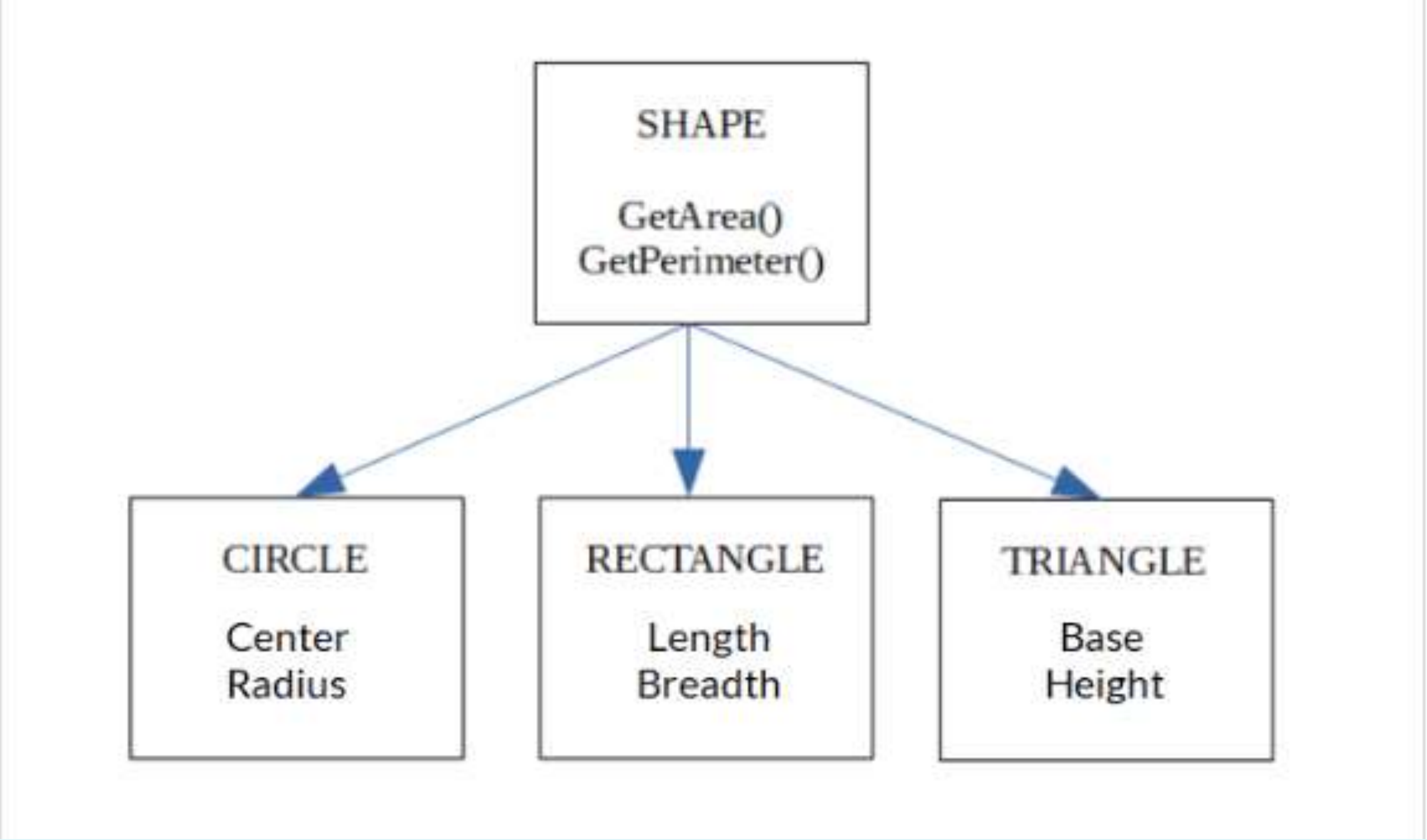
- Semantic data model (SDM) developed by Hammer and McLeod in 1981
- Modeled both data and their relationships in a single structure known as an object
- Data Stored in the form of objects, the structure which are called "Classes" that display data within it.
- Defines database as a collection of objects that contains both data members values and operations that are allowed on the data.
- The DBMS developed using this mode is called OODBMS
- Combine the capabilities of object oriented programming language
- Evolved to handle more complex applications such as database for scientific experiments, geographic information system, CAD (computer Aided Design)

The real-world problems are more closely represented through the object-oriented data model. In this model, both the data and relationship are present in a single structure known as an object. We can store audio, video, images, etc in the database which was not possible in the relational model(although you can store audio and video in relational database, it is adviced not to store in the relational database). In this model, two are more objects are connected through links. We use this link to relate one object to other objects. This can be understood by the example given below.



Object_Oriented_Model

Example:



SHAPE

GetArea()
GetPerimeter()

CIRCLE

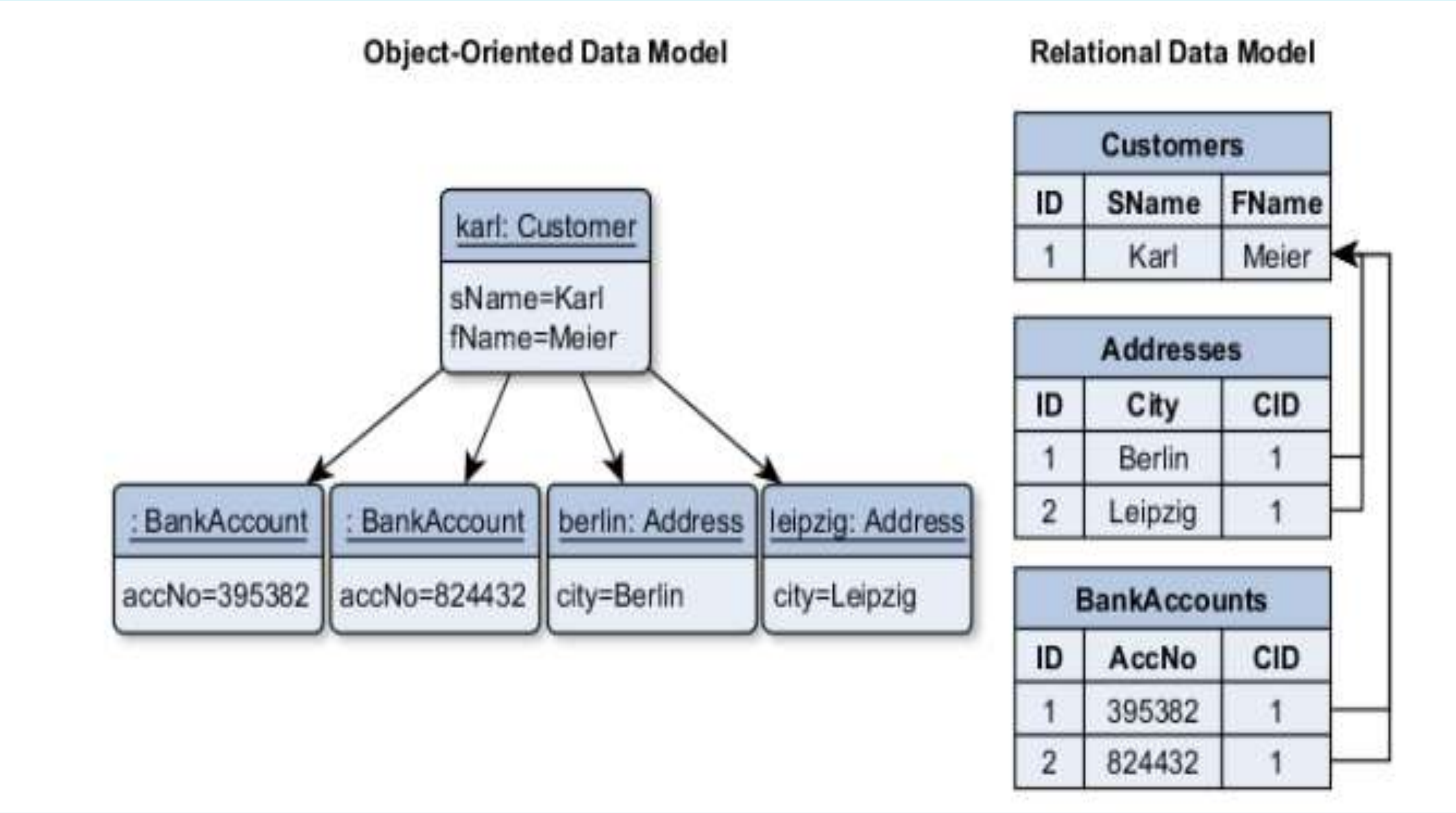
Center
Radius

RECTANGLE

Length
Breadth

TRIANGLE

Base
Height



Object-Oriented Data Model

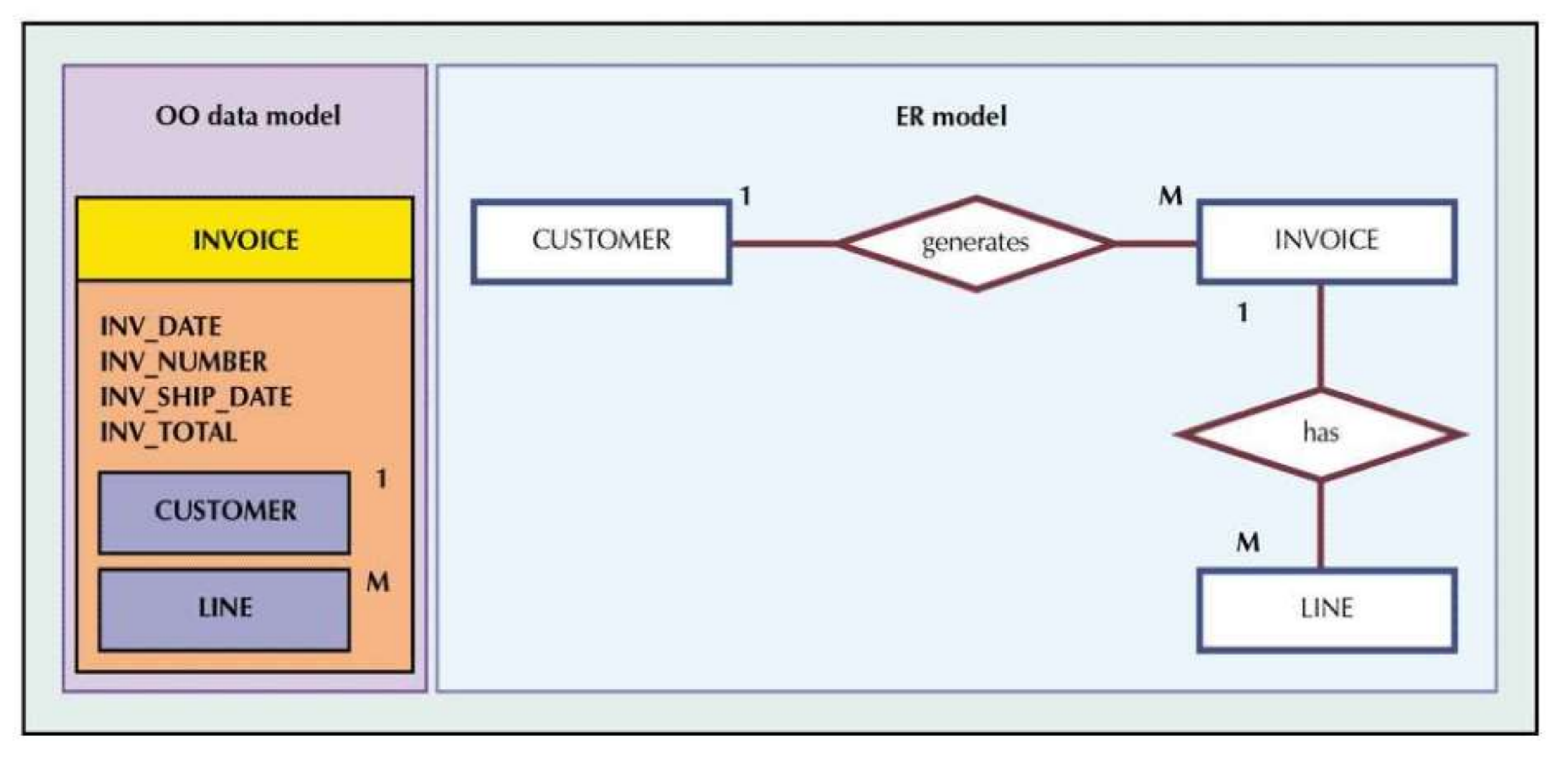
Relational Data Model

Customers		
ID	SName	FName
1	Karl	Meier

Addresses		
ID	City	CID
1	Berlin	1
2	Leipzig	1

BankAccounts		
ID	AccNo	CID
1	395382	1
2	824432	1

Comparision of OOModel and ER Model



Advantages

- Visual presentation includes semantic content

- Due to inheritance, the data types can be reused in different objects. This reduces the cost of maintaining the same data in multiple locations.

- The object oriented model is quite flexible in most cases.

- It is easier to extend the design in Object Oriented Model.

Disadvantages

- It is not practically implemented in database systems as it is mostly a theoretical approach.
- Object databases are not as popular as RDBMS. It is difficult to find object DB developers.
- Not many programming language support object databases.
- RDBMS have SQL as a standard query language. Object databases do not have a standard.
- Object databases are difficult to learn for non-programmers.

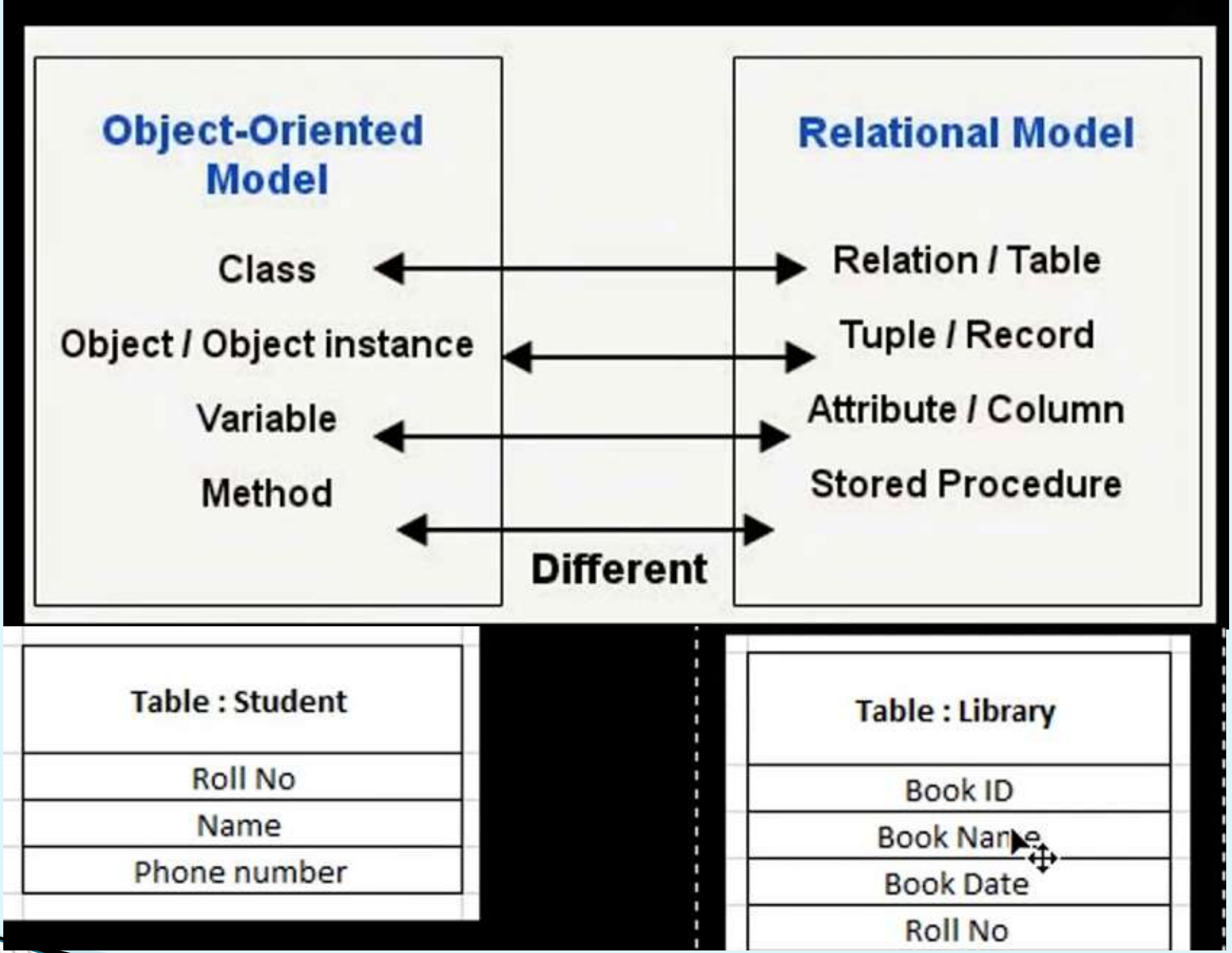
Object Relational Model

✓This hybrid database model combines the simplicity of the relational model with some of the advanced functionality of the object-oriented database model.

✓In essence, it allows designers to incorporate objects into the familiar table structure.

✓ An Object relational model is a combination of a Object oriented database model and a Relational database model. So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.

✓One of the major goals of Object relational data model is to close the gap between relational databases and the object oriented practises frequently used in many programming languages such as C++, C#, Java etc.



Object-Oriented Model

Relational Model

Class ↔ Relation / Table

Object / Object instance ↔ Tuple / Record

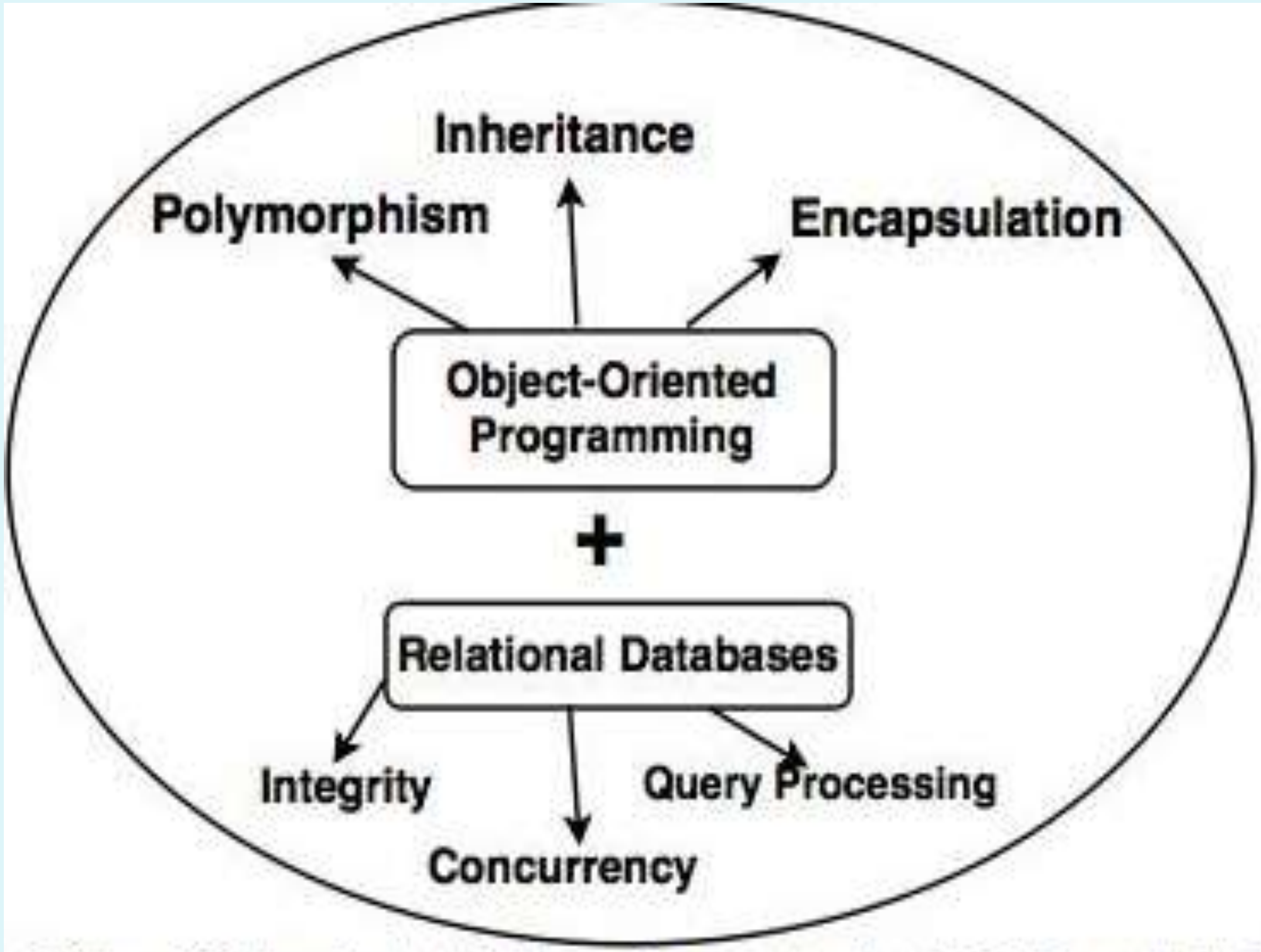
Variable ↔ Attribute / Column

Method ↔ Stored Procedure

Different

Table : Student
Roll No
Name
Phone number

Table : Library
Book ID
Book Name
Book Date
Roll No



(Object-Oriented database is product of OOP and RDB)

Object-Oriented database

Advantages

Reuse and Sharing: The main advantages of extending the Relational data model come from reuse and sharing. Reuse comes from the ability to extend the DBMS server to perform standard functionality centrally, rather than have it coded in each application.

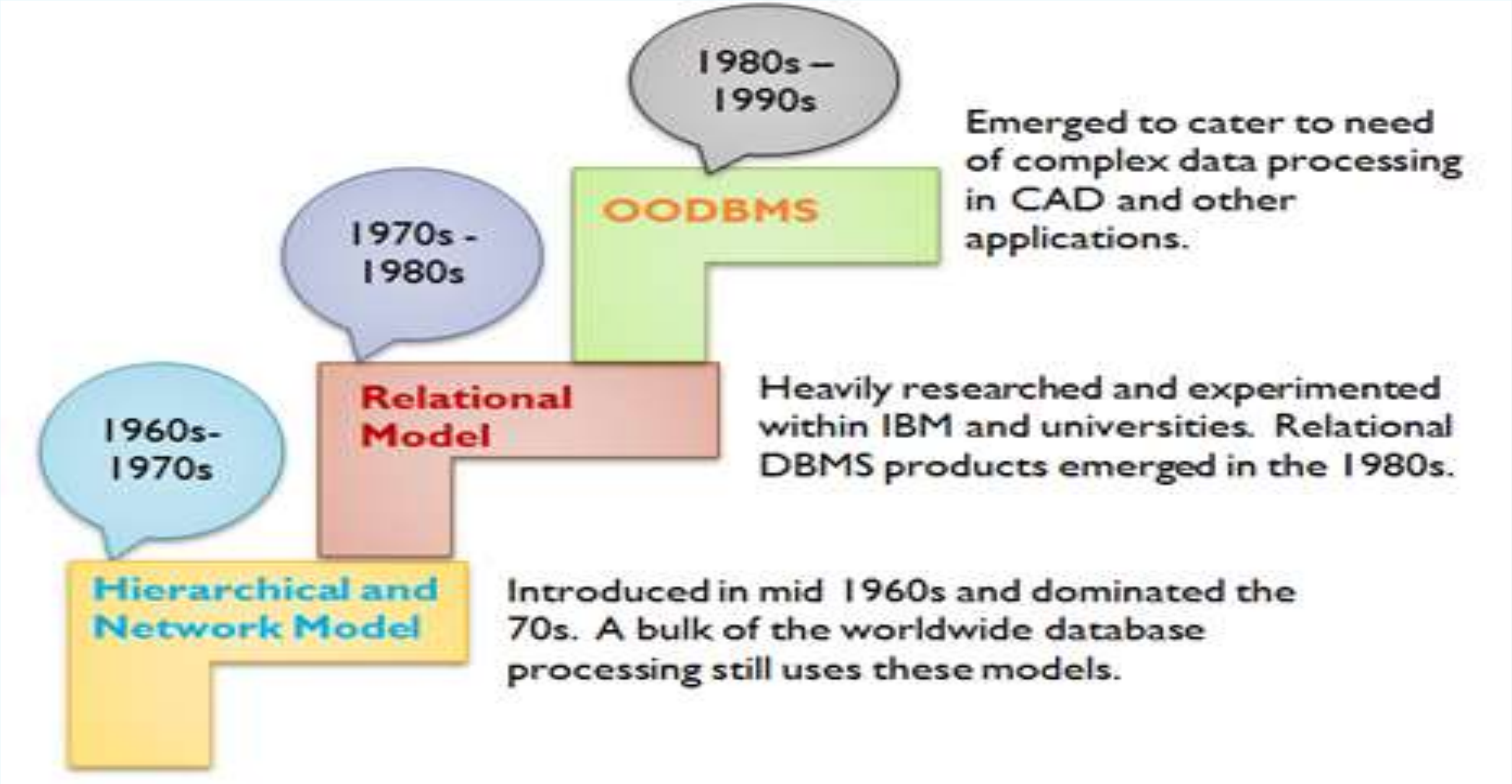
Increased Productivity: ORDBMS provides increased productivity both for the developer and for the end user

Use of experience in developing RDBMS: Another obvious advantage is that the extended relational approach preserves the significant body of knowledge and experience that has gone into developing relational applications. This is a significant advantage, as many organizations would find it prohibitively expensive to change.

Disadvantages

- complexity
- increased costs

Evolution of Data Models

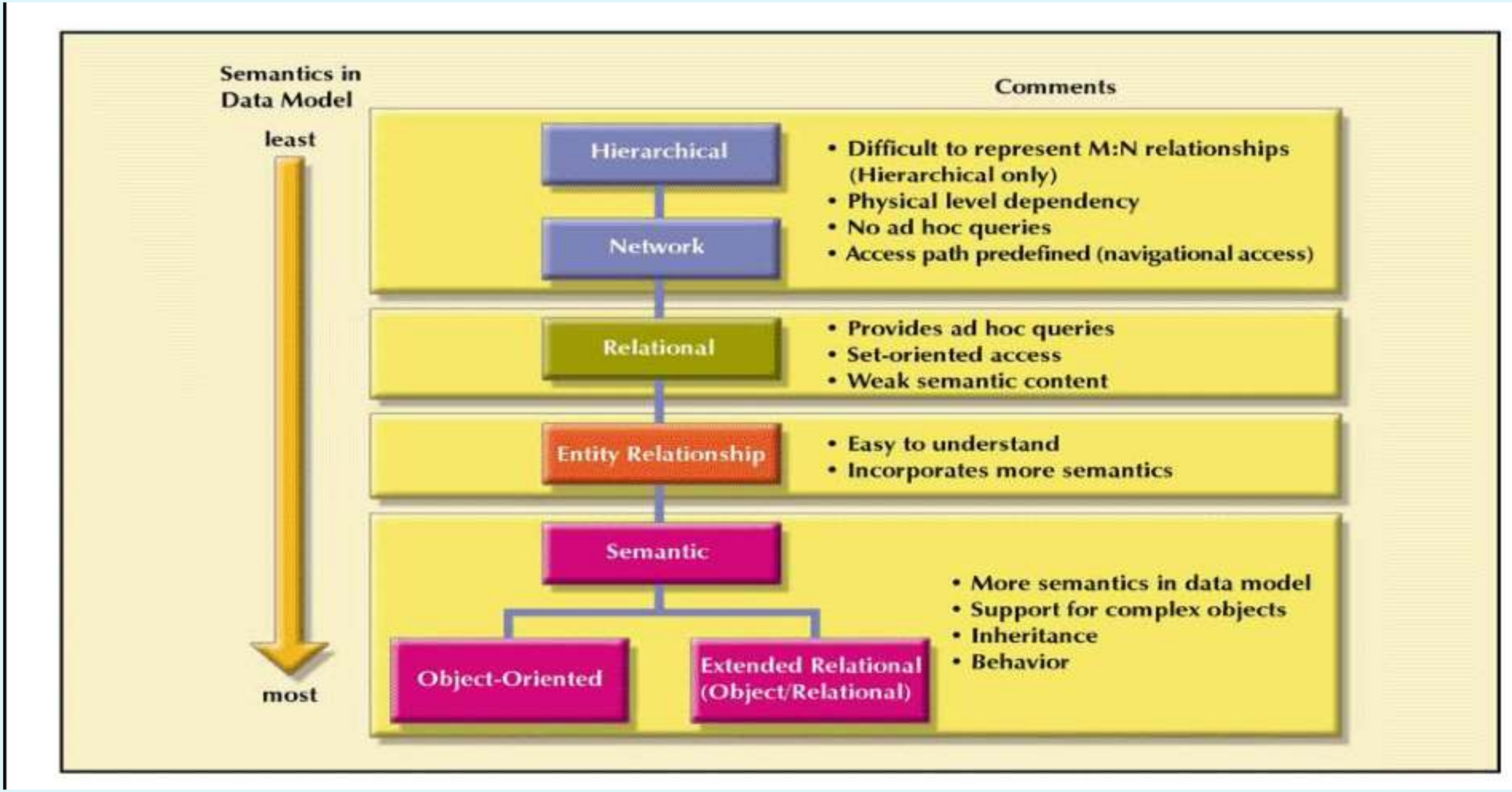


Emerged to cater to need of complex data processing in CAD and other applications.

Heavily researched and experimented within IBM and universities. Relational DBMS products emerged in the 1980s.

Introduced in mid 1960s and dominated the 70s. A bulk of the worldwide database processing still uses these models.

Development of Data Models



- Difficult to represent M:N relationships (Hierarchical only)
- Physical level dependency
- No ad hoc queries
- Access path predefined (navigational access)

- Provides ad hoc queries
- Set-oriented access
- Weak semantic content

- Easy to understand
- Incorporates more semantics

- More semantics in data model
- Support for complex objects
- Inheritance
- Behavior

Real Time Applications of Data Models

Hierarchial Model:It is used in **applications** where high performance is required such as telecommunications and banking.
IBM Information Management System (IMS) and RDM Mobile

Network Model: Some of the popular network databases are,
Integrated Data Store (IDS)
IDMS (Integrated Database Management System)
Raima Database Manager
TurboIMAGE
Univac DMS-1100

Relational Model: Microsoft SQL Server, Oracle **Database** and IBM DB2

Object Oriented Model: The idea of object databases was originated in 1985 and today has become common for various common OOP languages, such as C++, Java, C#, Smalltalk, and LISP. Common examples are Smalltalk is used in GemStone, LISP is used in Gbase