Chapter 1

INTRODUCTION TO DATABASE SYSTEMS

1.1 INTRODUCTION

An organization must have accurate and reliable data (information) for effective decision making. Data (information) is the backbone and most critical resource of an organization that enables managers and organizations to gain a competitive edge. In this age of information explosion, where people are bombarded with data, getting the right information, in the right amount, at the right time is not an easy task. So, only those organizations will survive that successfully manage information.

A database system simplifies the tasks of managing the data and extracting useful information in a timely fashion. A database system is an integrated collection of related files, along with the details of the interpretation of the data. A Data Base Management System is a software system or program that allows access to data contained in a database. The objective of the DBMS is to provide a convenient and effective method of defining, storing, and retrieving the information stored in the database.

The database and database management systems have become essential for managing business, governments, schools, universities, banks etc.

1.2 BASIC DEFINITIONS AND CONCEPTS

In an organization, the data is the most basic resource. To run the organization efficiently, the proper organization and management of data is essential. The formal definition of the major terms used in databases and database systems is defined in this section.

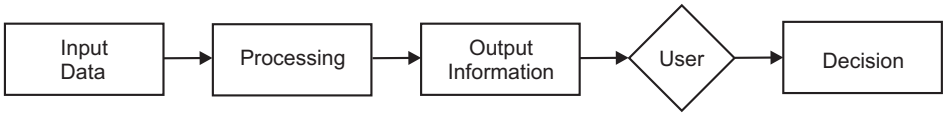
1.2.1 Data

The term data may be defined as known facts that could be recorded and stored on Computer Media. It is also defined as raw facts from which the required information is produced.

1.2.2 Information

Data and information are closely related and are often used interchangeably. Information is nothing but refined data. In other way, we can say, information is processed, organized or summarized data. According to Burch *et. al.,* "Information is data that have been put into a meaningful and useful content and communicated to a recipient who uses it to made decisions". Information consists of data, images, text, documents and voice, but always in a meaningful content. So we can say, that information is something more than mere data.

Data are processed to create information. The recipient receives the information and then makes a decision and takes an action, which may triggers other actions



In these days, there is no lack of data, but there is lack of quality information. The quality information means information that is accurate, timely and relevant, which are the three major key attributes of information.

1. **Accuracy :** It means that the information is free from errors, and it clearly and accurately reflects the meaning of data on which it is based. It also means it is free from bias and conveys an accurate picture to the recipient.
2. **Timeliness :** It means that the recipients receive the information when they need it and within the required time frame.
3. **Relevancy :** It means the usefulness of the piece of information for the corresponding persons. It is a very subjective matter. Some information that is relevant for one person might not be relevant for another and vice versa *e.g.,* the price of printer is irrelevant for a person who wants to purchase computer.

So, organization that have good information system, which produce information that is accurate, timely and relevant will survive and those that do not realize the importance of information will soon be out of business.

1.2.3 Meta Data

A meta data is the data about the data. The meta data describe objects in the database and makes easier for those objects to be accessed or manipulated. The meta data describes the database structure, sizes of data types, constraints, applications, autorisation etc., that are used as an integral tool for information resource management. There are three main types of meta data :

1. **Descriptive meta data :** It describes a resource for purpose such as discovery and identification. In a traditional library cataloging that is form of meta data, title, abstract, author and keywords are examples of meta data.
2. **Structural meta data :** It describes how compound objects are put together. The example is how pages are ordered to form chapters.
3. **Administrative meta data :** It provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it. There are several subsets of data.

1.2.4 Data Dictionary

The data dictionary contains information of the data stored in the database and is consulted by the DBMS before any manipulation operation on the database. It is an integral part of the database management systems and store meta data *i.e.,* information about the database, attribute names and definitions for each table in the database. It helps the DBA in the management of the database, user view definitions as well as their use.

Data dictionary is generated for each database and generally stores and manages the following types of information :

1. The complete information about physical database design *e.g.* storage structures, access paths and file sizes etc.
2. The information about the database users, their responsibilities and access rights of each user.
3. The complete information about the schema of the database.
4. The high level descriptions of the database transactions, applications and the information about the relationships of users to the transactions.
5. The information about the relationship between the data items referenced by the database transactions. This information is helpful in determining which transactions are affected when some data definitions are modified.

The data dictionaries are of two types : Active data dictionary and passive data dictionary.

1. **Active Data Dictionary :** It is managed automatically by the database management system (DBMS) and are always consistent with the current structure and definition of the database. Most of the RDBMS's maintain active data dictionaries.
2. **Passive Data Dictionary :** It is used only for documentation purposes and the data about fields, files and people are maintained into the dictionary for cross references. It is generally managed by the users of the system and is modified whenever the structure of the database is changed. The passive dictionary may not be consistent with the structure of the database, since modifications are performed manually by the user. It is possible that passive dictionaries may contain information about organisational data that is not computerized as these are maintained by the users.

1.2.5 Database

A database is a collection of interrelated data stored together with controlled redundancy to serve one or more applications in an optimal way. The data are stored in such a way that they are independent of the programs used by the people for accessing the data. The approach used in adding the new data, modifying and retrieving the existing data from the database is common and controlled one.

It is also defined as a collection of logically related data stored together that is designed to meet information requirements of an organization. We can also define it as an electronic filling system.

The example of a database is a telephone directory that contains names, addresses and telephone numbers of the people stored in the computer storage.

Databases are organized by fields, records and files. These are described briefly as follows :

1.2.5.1 Fields

It is the smallest unit of the data that has meaning to its users and is also called data item or data element. Name, Address and Telephone number are examples of fields. These are represented in the database by a value.

1.2.5.2 Records

A record is a collection of logically related fields and each field is possessing a fixed number of bytes and is of fixed data type. Alternatively, we can say a record is one complete set of fields and each field have some value. The complete information about a particular phone number in the database represents a record. Records are of two types **fixed length records** and **variable length records**.

1.2.5.3 Files

A file is a collection of related records. Generally, all the records in a file are of same size and record type but it is not always true. The records in a file may be of fixed length or variable length depending upon the size of the records contained in a file. The telephone directory containing records about the different telephone holders is an example of file. More detail is available in chapter 3.

1.2.6 Components of a Database

A Database consists of four components as shown in Figure 1.1.

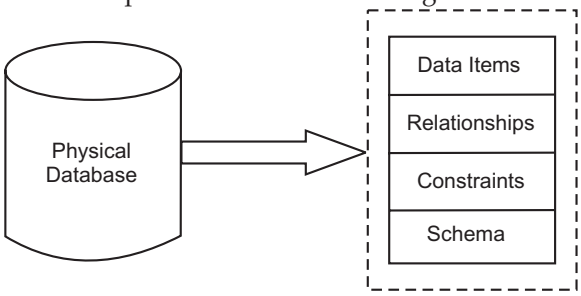


FIGURE 1.1. Components of Database.

1. **Data item :** It is defined as a distinct piece of information and is explained in the previous section.
2. **Relationships :** It represents a correspondence between various data elements.
3. **Constraints :** These are the predicates that define correct database states.
4. **Schema :** It describes the organization of data and relationships within the database. The schema consists of definitions of the various types of record in the database, the data-items they contain and the sets into which they are grouped. The storage structure of the database is described by the *storage schema*. The *conceptual schema* defines the stored data structure. The *external schema* defines a view of the database for particular users.

1.2.7 Database Management System (DBMS)

DBMS is a program or group of programs that work in conjunction with the operating system to create, process, store, retrieve, control and manage the data. It acts as an interface between the application program and the data stored in the database.

Alternatively, it can be defined as a computerized record-keeping system that stores information and allows the users to add, delete, modify, retrieve and update that information.

The DBMS performs the following five primary functions :

1. **Define, create and organise a database :** The DBMS establishes the logical relationships among different data elements in a database and also defines schemas and subschemas using the DDL.
2. **Input data :** It performs the function of entering the data into the database through an input device (like data screen, or voice activated system) with the help of the user.
3. **Process data :** It performs the function of manipulation and processing of the data stored in the database using the DML.
4. **Maintain data integrity and security :** It allows limited access of the database to authorised users to maintain data integrity and security.
5. **Query database :** It provides information to the decision makers that they need to make important decisions. This information is provided by querying the database using SQL.

1.2.8 Components of DBMS

A DBMS has three main components. These are Data Definition Language (DDL), Data Manipulation Language and Query Facilities (DML/SQL) and software for controlled access of Database as shown in Figure 1.2 and are defined as follows :

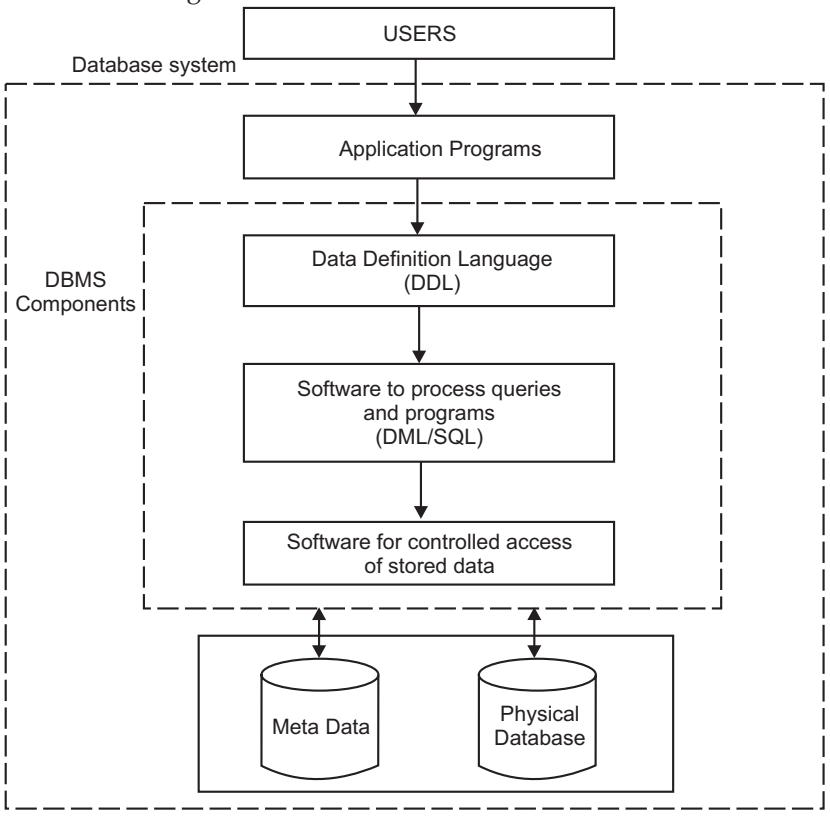


FIGURE 1.2. Components of DBMS.

1.2.8.1 Data Definition Language (DDL)

It allows the users to define the database, specify the data types, data structures and the constraints on the data to be stored in the database. More about DDL in section 1.5.

1.2.8.2 Data Manipulation Language (DML) and Query Language

DML allows users to insert, update, delete and retrieve data from the database. SQL provides general query facility. More about DML and SQL in section 1.5.

1.2.8.3 Software for Controlled Access of Database

This software provides the facility of controlled access of the database by the users, concurrency control to allow shared access of the database and a recovery control system to restore the database in case of ardware or software failure.

NOTE *The DBMS software together with the database is called a Database System.*

1.3 TRADITIONAL FILE SYSTEM VERSUS DATABASE SYSTEMS

Conventionally, the data were stored and processed using traditional file processing systems. In these traditional file systems, each file is independent of other file, and data in different files can be integrated only by writing individual program for each application. The data and the application programs that uses the data are so arranged that any change to the data requires modifying all the programs that uses the data. This is because each file is hard-coded with specific information like data type, data size etc. Some time it is even not possible to identify all the programs using that data and is identified on a trial-and-error basis.

A file processing system of an organization is shown in Figure 1.3. All functional areas in the organization creates, processes and disseminates its own files. The files such as inventory and payroll generate separate files and do not communicate with each other.

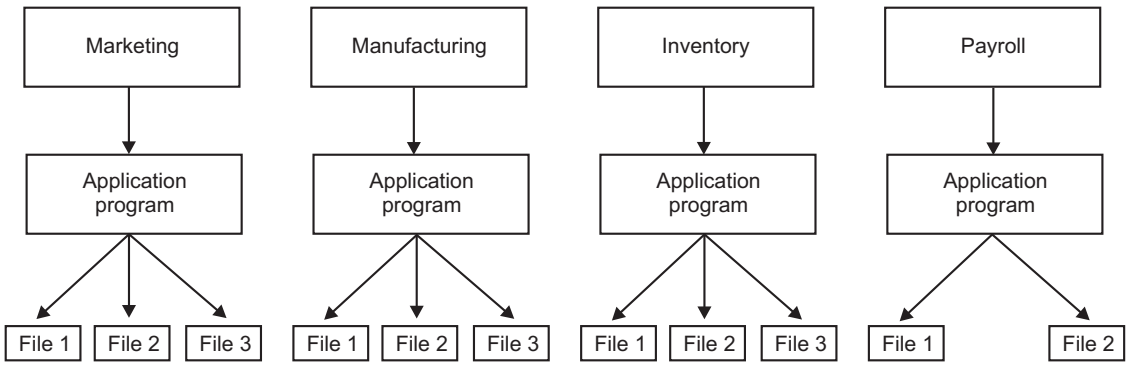


FIGURE 1.3. Traditional file system.

No doubt such an organization was simple to operate and had better local control but the data of the organization is dispersed throughout the functional sub-systems. These days, databases are preferred because of many disadvantages of traditional file systems.

1.3.1 Disadvantages of Traditional File System

A traditional file system has the following disadvantages:

1. **Data Redundancy :** Since each application has its own data file, the same data may have to be recorded and stored in many files. For example, personal file and payroll file, both contain data on employee name, designation etc. The result is unnecessary duplicate or redundant data items. This redundancy requires additional or higher storage space, costs extra time and money, and requires additional efforts to keep all files upto-date.
2. **Data Inconsistency :** Data redundancy leads to data inconsistency especially when data is to be updated. Data inconsistency occurs due to the same data items that appear in more than one file do not get updated simultaneously in each and every file. For example, an employee is promoted from Clerk to Superintendent and the same is immediately updated in the payroll file may not necessarily be updated in provident fund file. This results in two different designations of an employee at the same time. Over the period of time, such discrepencis degrade the quality of information contain in the data file that affects the accuracy of reports.
3. **Lack of Data Integration :** Since independent data file exists, users face difficulty in getting information on any ad hoc query that requires accessing the data stored in many files. In such a case complicated programs have to be developed to retrieve data from every file or the users have to manually collect the required information.
4. **Program Dependence :** The reports produced by the file processing system are program dependent, which means if any change in the format or structure of data and records in the file is to be made, the programs have to modified correspondingly. Also, a new program will have to be developed to produce a new report.
5. **Data Dependence :** The Applications/programs in file processing system are data dependent *i.e.,* the file organization, its physical location and retrieval from the storage media are dictated by the requirements of the particular application. For example, in payroll application, the file may be organised on employee records sorted on their last name, which implies that accessing of any employee's record has to be through the last name only.
6. **Limited Data Sharing :** There is limited data sharing possibilities with the traditional file system. Each application has its own private files and users have little choice to share the data outside their own applications. Complex programs required to be written to obtain data from several incompatible files.
7. **Poor Data Control :** There was no centralised control at the data element level, hence a traditional file system is decentralised in nature. It could be possible that the data field may have multiple names defined by the different departments of an organization and depending on the file it was in. This situation leads to different meaning of a data field in different context or same meaning for different fields. This causes poor data control.
8. **Problem of Security :** It is very difficult to enforce security checks and access rights in a traditional file system, since application programs are added in an adhoc manner.

9. **Data Manipulation Capability is Inadequate :** The data manipulation capability is very limited in traditional file systems since they do not provide strong relationships between data in different files.
10. **Needs Excessive Programming :** An excessive programming effort was needed to develop a new application program due to very high interdependence between program and data in a file system. Each new application requires that the developers start from the scratch by designing new file formats and descriptions and then write the file access logic for each new file.

1.3.2 Database Systems or Database System Environment

The DBMS software together with the Database is called a database system. In other words, it can be defined as an organization of components that define and regulate the collection, storage, management and use of data in a database. Furthermore, it is a system whose overall purpose is to record and maintain information. A database system consists of four major components as shown in Figure 1.4.

1. Data 2. Hardware 3. Software 4. Users

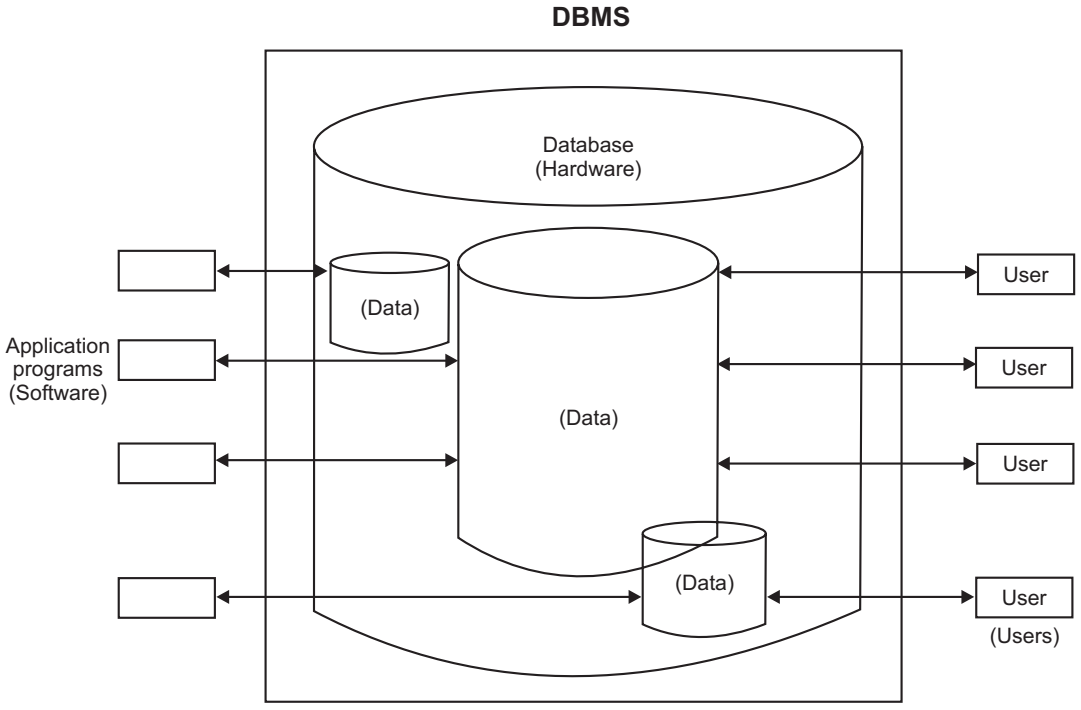


FIGURE 1.4. Database system.

1. Data : The whole data in the system is stored in a single database. This data in the database are both shared and integrated. Sharing of data means individual pieces of data in the database is shared among different users and every user can access the same piece of data but may be for different purposes. Integration of data means the database can be function of several distinct files with redundancy controlled among the files.

2. Hardware : The hardware consists of the secondary storage devices like disks, drums and so on, where the database resides together with other devices. There is two types of hardware. The first one, *i.e.,* processor and main memory that supports in running the DBMS. The second one is the secondary storage devices, *i.e.,* hard disk, magnetic disk etc., that are used to hold the stored data.

3. Software : A layer or interface of software exists between the physical database and the users. This layer is called the DBMS. All requests from the users to access the database are handled by the DBMS. Thus, the DBMS shields the database users from hardware details. Furthermore, the DBMS provides the other facilities like accessing and updating the data in the files and adding and deleting files itself.

4. Users : The users are the people interacting with the database system in any way. There are four types of users interacting with the database systems. These are Application Programmers, online users, end users or naive users and finally the Database Administrator (DBA). More about users in section 1.4.

1.3.3 Advantages of Database Systems (DBMS's)

The database systems provide the following advantages over the traditional file system:

1. **Controlled redundancy :** In a traditional file system, each application program has its own data, which causes duplication of common data items in more than one file. This duplication/redundancy requires multiple updations for a single transaction and wastes a lot of storage space. We cannot eliminate all redundancy due to technical reasons. But in a database, this duplication can be carefully controlled, that means the database system is aware of the redundancy and it assumes the responsibility for propagating updates.
2. **Data consistency :** The problem of updating multiple files in traditional file system leads to inaccurate data as different files may contain different information of the same data item at a given point of time. This causes incorrect or contradictory information to its users. In database systems, this problem of inconsistent data is automatically solved by controlling the redundancy.
3. **Program data independence :** The traditional file systems are generally data dependent, which implies that the data organization and access strategies are dictated by the needs of the specific application and the application programs are developed accordingly. However, the database systems provide an independence between the file system and application program, that allows for changes at one level of the data without affecting others. This property of database systems allow to change data without changing the application programs that process the data.
4. **Sharing of data :** In database systems, the data is centrally controlled and can be shared by all authorized users. The sharing of data means not only the existing applications programs can also share the data in the database but new application programs can be developed to operate on the existing data. Furthermore, the requirements of the new application programs may be satisfied without creating any new file.
5. **Enforcement of standards :** In database systems, data being stored at one central place, standards can easily be enforced by the DBA. This ensures standardised data formats

to facilitate data transfers between systems. Applicable standards might include any or all of the following—departmental, installation, organizational, industry, corporate, national or international.

6. **Improved data integrity :** Data integrity means that the data contained in the database is both accurate and consistent. The centralized control property allow adequate checks can be incorporated to provide data integrity. One integrity check that should be incorporated in the database is to ensure that if there is a reference to certain object, that object must exist.
7. **Improved security :** Database security means protecting the data contained in the database from unauthorised users. The DBA ensures that proper access procedures are followed, including proper authentical schemes for access to the DBMS and additional checks before permitting access to sensitive data. The level of security could be different for various types of data and operations.
8. **Data access is efficient :** The database system utilizes different sophisticated techniques to access the stored data very efficiently.
9. **Conflicting requirements can be balanced :** The DBA resolves the conflicting requirements of various users and applications by knowing the overall requirements of the organization. The DBA can structure the system to provide an overall service that is best for the organization.
10. **Improved backup and recovery facility :** Through its backup and recovery subsystem, the database system provides the facilities for recovering from hardware or software failures. The recovery subsystem of the database system ensures that the database is restored to the state it was in before the program started executing, in case of system crash.
11. **Minimal program maintenance :** In a traditional file system, the application programs with the description of data and the logic for accessing the data are built individually. Thus, changes to the data formats or access methods results in the need to modify the application programs. Therefore, high maintenance effort are required. These are reduced to minimal in database systems due to independence of data and application programs.
12. **Data quality is high :** The quality of data in database systems are very high as compared to traditional file systems. This is possible due to the presence of tools and processes in the database system.
13. **Good data accessibility and responsiveness :** The database systems provide query languages or report writers that allow the users to ask ad hoc queries to obtain the needed information immediately, without the requirement to write application programs (as in case of file system), that access the information from the database. This is possible due to integration in database systems.
14. **Concurrency control :** The database systems are designed to manage simultaneous (concurrent) access of the database by many users. They also prevents any loss of information or loss of integrity due to these concurrent accesses.
15. **Economical to scale :** In database systems, the operational data of an organization is stored in a central database. The application programs that work on this data can be

built with very less cost as compared to traditional file system. This reduces overall costs of operation and management of the database that leads to an economical scaling.

16. **Increased programmer productivity :** The database system provides many standard functions that the programmer would generally have to write in file system. The availability of these functions allow the programmers to concentrate on the specific functionality required by the users without worrying about the implementation details. This increases the overall productivity of the programmer and also reduces the development time and cost.

1.3.4 Disadvantages of Database Systems

In contrast to many advantages of the database systems, there are some disadvantages as well. The disadvantages of a database system are as follows :

1. **Complexity increases :** The data structure may become more complex because of the centralised database supporting many applications in an organization. This may lead to difficulties in its management and may require professionals for management.
2. **Requirement of more disk space :** The wide functionality and more complexity increase the size of DBMS. Thus, it requires much more space to store and run than the traditional file system.
3. **Additional cost of hardware :** The cost of database system's installation is much more. It depends on environment and functionality, size of the hardware and maintenance costs of hardware.
4. **Cost of conversion :** The cost of conversion from old file-system to new database system is very high. In some cases the cost of conversion is so high that the cost of DBMS and extra hardware becomes insignificant. It also includes the cost of training manpower and hiring the specialized manpower to convert and run the system.
5. **Need of additional and specialized manpower :** Any organization having database systems, need to be hire and train its manpower on regular basis to design and implement databases and to provide database administration services.
6. **Need for backup and recovery :** For a database system to be accurate and available all times, a procedure is required to be developed and used for providing backup copies *to all its users when damage occurs*.
7. **Organizational conflict :** A centralised and shared database system requires a consensus on data definitions and ownership as well as responsibilities for accurate data maintenance.
8. **More installational and management cost :** The big and complete database systems are more costly. They require trained manpower to operate the system and has additional annual maintenance and support costs.

1.4 DBMS USERS

The users of a database system can be classified into various categories depending upon their interaction and degree of expertise of the DBMS.

1.4.1 End Users or Naive Users

The end users or naive users use the database system through a menu-oriented application program, where the type and range of response is always displayed on the screen. The user need not be aware of the presence of the database system and is instructed through each step. A user of an ATM falls in this category.

1.4.2 Online Users

These type of users communicate with the database directly through an online terminal or indirectly through an application program and user interface. They know about the existence of the database system and may have some knowledge about the limited interaction they are permitted.

1.4.3 Application Programmers

These are the professional programmers or software developers who develop the application programs or user interfaces for the naive and online users. These programmers must have the knowledge of programming languages such as Assembly, C, C++, Java, or SQL, etc., since the application programs are written in these languages.

1.4.4 Database Administrator

Database Administrator (DBA) is a person who have complete control over database of any enterprise. DBA is responsible for overall performance of database. He is free to take decisions for database and provides technical support. He is concerned with the Back-End of any project. Some of the main responsibilities of DBA are as follows :

1. **Deciding the conceptual schema or contents of database :** DBA decides the data fields, tables, queries, data types, attributes, relations, entities or you can say that he is responsible for overall logical design of database.
2. **Deciding the internal schema of structure of physical storage :** DBA decides how the data is actually stored at physical storage, how data is represented at physical storage.
3. **Deciding users :** DBA gives permission to users to use database. Without having proper permission, no one can access data from database.
4. **Deciding user view :** DBA decides different views for different users.
5. **Granting of authorities :** DBA decides which user can use which portion of database. DBA gives authorities or rights to data access. User can use only that data on which access right is granted to him.
6. **Deciding constraints :** DBA decides various constraints over database for maintaining consistency and validity in database.
7. **Security :** Security is the major concern in database. DBA takes various steps to make data more secure against various disasters and unauthorized access of data.
8. **Monitoring the performance :** DBA is responsible for overall performance of database. DBA regularly monitors the database to maintain its performance and try to improveit.

9. **Backup :** DBA takes regular backup of database, so that it can be used during system failure. Backup is also used for checking data for consistency.
10. **Removal of dump and maintain free space :** DBA is responsible for removing unnecessary data from storage and maintain enough free space for daily operations. He can also increase storage capacity when necessary.
11. **Checks :** DBA also decides various security and validation checks over database to ensure consistency.
12. **Liaisioning with users :** Another task of the DBA is to liaisioning with users and ensure the availability of the data they require and write the necessary external schemas.

1.5 DATABASE OR DBMS LANGUAGES

The DBMS provides different languages and interfaces for each category of users to express database queries and updations. When the design of the database is complete and the DBMS is chosen to implement it, the first thing to be done is to specify the conceptual and internal schemas for the database and the corresponding mappings. The following five languages are available to specify different schemas.

1. Data Definition Language (DDL)
2. Storage Definition Language (SDL)
3. View Definition Language (VDL)
4. Data Manipulation Language (DML)
5. Fourth-Generation Language (4-GL)

1.5.1 Data Definition Language (DDL)

It is used to specify a database conceptual schema using set of definitions. It supports the definition or declaration of database objects. Many techniques are available for writing DDL. One widely used technique is writing DDL into a text file. More about DDL in chapter 7.

1.5.2 Storage Definition Language (SDL)

It is used to specify the internal schema in the database. The storage structure and access methods used by the database system is specified by the specified set of SDL statements. The implementation details of the database schemas are implemented by the specified SDL statements and are usually hidden from the users.

1.5.3 View Definition Language (VDL)

It is used to specify user's views and their mappings to the conceptual schema. But generally, DDL is used to specify both conceptual and external schemas in many DBMS's. There are two views of data the **logical view**—that is perceived by the programmer and **physical view**—data stored on storage devices.

1.5.4 Data Manipulation Language (DML)

It provides a set of operations to support the basic data manipulation operations on the data held in the database. It is used to query, update or retrieve data stored in a database. The part of DML that provide data retrieval is called query language.

The DML is of the two types :

(*i*) *Procedural DML :* It allows the user to tell the system what data is needed and how to retrieve it.

(*ii*) *Non-procedural DML :* It allows the user to state what data are needed, rather than how it is to be retrieved. More about DML in chapter 7.

1.5.5 Fourth-Generation Language (4-GL)

It is a compact, efficient and non-procedural programming language used to improve the efficiency and productivity of the DBMS. In this, the user defines what is to be done and not how it is to be done. The 4-GL has the following components in it. These are :

(*a*) Query languages
(*b*) Report
(*c*) Spread sheets
(*d*) Database languages
(*e*) Application generators
(*f*) High level languages to generate application program.

System Query Language (SQL) is an example of 4-GL. More about SQL in Chapter 7.

1.6 SCHEMAS, SUBSCHEMA AND INSTANCES

The plans of the database and data stored in the database are most important for an organization, since database is designed to provide information to the organization. The data stored in the database changes regularly but the plans remain static for longer periods of time.

1.6.1 Schema

A schema is plan of the database that give the names of the entities and attributes and the relationship among them. A schema includes the definition of the database name, the record type and the components that make up the records. Alternatively, it is defined as a frame-work into which the values of the data items are fitted. The values fitted into the frame-work changes regularly but the format of schema remains the same *e.g.,* consider the database consisting of three files ITEM, CUSTOMER and SALES. The data structure diagram for this schema is shown in Figure 1.5. The schema is shown in database language.

Generally, a schema can be partitioned into two categories, *i.e.,* (*i*) *Logical schema* and (*ii*) *Physical schema*.

(*i*) The *logical schema* is concerned with exploiting the data structures offered by the DBMS so that the schema becomes understandable to the computer. It is important as programs use it to construct applications.

(*ii*) The *physical schema* is concerned with the manner in which the conceptual database get represented in the computer as a stored database. It is hidden behind the logical schema and can usually be modified without affecting the application programs.

The DBMS's provide DDL and DSDL to specify both the logical and physical schema.

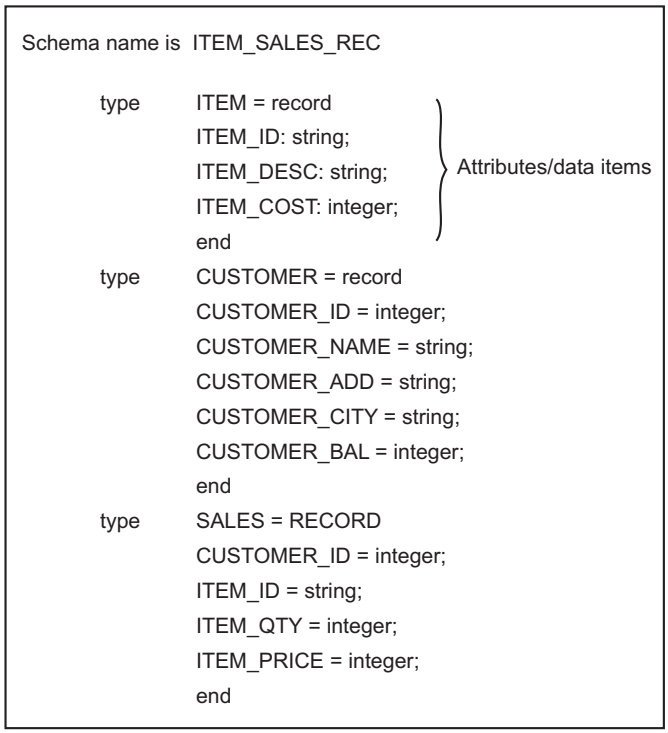


FIGURE 1.5. Data structure diagram for the item sales record.

1.6.2 Subschema

A subschema is a subset of the schema having the same properties that a schema has. It identifies a subset of areas, sets, records, and data names defined in the database schema available to user sessions. The subschema allows the user to view only that part of the database that is of interest to him. The subschema defines the portion of the database as seen by the application programs and the application programs can have different view of data stored in the database.

The different application programs can change their respective subschema without affecting other's subschema or view.

The Subschema Definition Language (SDL) is used to specify a subschema in the DBMS.

1.6.3 Instances

The data in the database at a particular moment of time is called an *instance* or a *database state*. In a given instance, each schema construct has its own current set of instances. Many instances or database states can be constructed to correspond to a particular database schema. Everytime we update (*i.e.,* insert, delete or modify) the value of a data item in a record, one state of the database changes into another state. The Figure 1.6 shows an instance of the ITEM relation in a database schema.

ITEM

ITEM-ID	ITEM_DESC	ITEM_COST
1111A	Nutt	3
1112A	Bolt	5
1113A	Belt	100
1144B	Screw	2

FIGURE 1.6. An instance/database state of the ITEM relation.

1.7 THREE LEVEL ARCHITECTURE OF DATABASE SYSTEMS (DBMS)

The architecture is a framework for describing database concepts and specifying the structure of database system. The three level architecture was suggested by ANSI/SPARC. Here database is divided into three levels **external level**, **conceptual level** and **internal level** as shown in Figure 1.7.

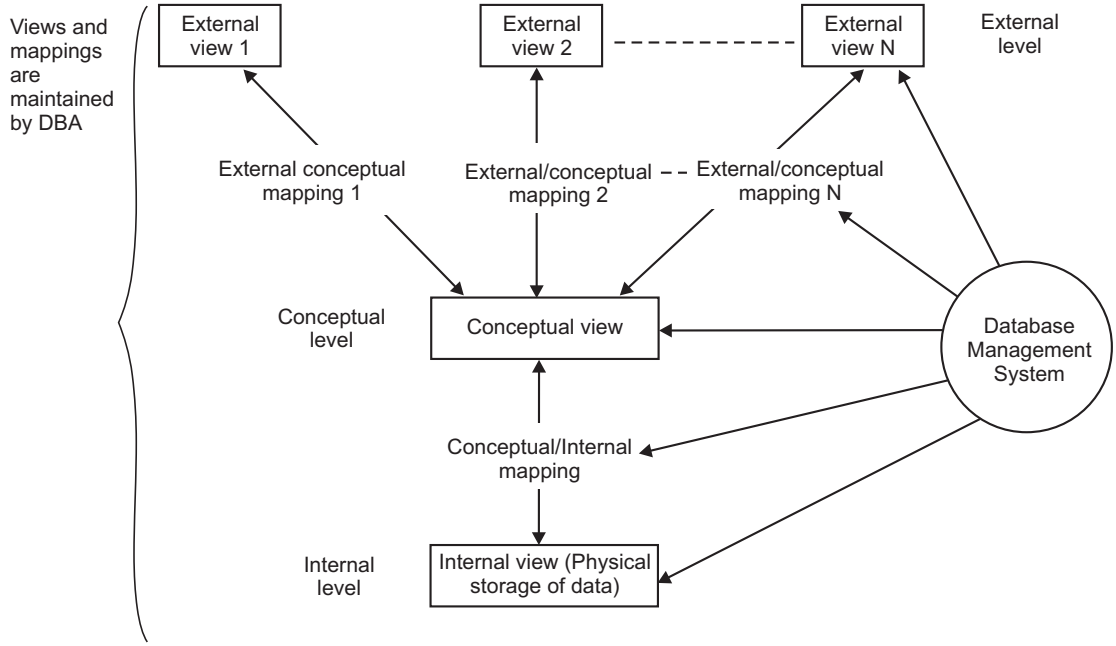


FIGURE 1.7. Three level architecture of DBMS.

1.7.1 Levels or Views

The three levels or views are discussed below:

(*i*) Internal Level : Internal level describes the actual physical storage of data or the way in which the data is actually stored in memory. This level is not relational because data is stored according to various coding schemes instead of tabular form (in tables). This is the low level representation of entire database. The internal view is described by means of an internal schema.

The internal level is concerned with the following aspects:

- Storage space allocation
- Access paths
- Data compression and encryption techniques
- Record placement etc.

The internal level provides coverage to the data structures and file organizations used to store data on storage devices.

(*ii*) Conceptual Level : The conceptual level is also known as logical level which describes the overall logical structure of whole database for a community of users. This level is relational because data visible at this level will be relational tables and operators will be relational operators. This level represents entire contents of the database in an abstract form in comparison with physical level. Here conceptual schema is defined which hides the actual physical storage and concentrate on relational model of database.

(*iii*) External Level : The external level is concerned with individual users. This level describes the actual view of data seen by individual users. The external schema is defined by the DBA for every user. The remaining part of database is hidden from that user. This means user can only access data of its own interest. In other words, user can access only that part of database for which he is authorized by DBA. This level is also relational or very close to it.

1.7.2 Different Mappings in Three Level Architecture of DBMS

The process of transforming requests and results between the three levels are called mappings. The database management system is responsible for this mapping between internal, external and conceptual schemas.

There are two types of mappings:

1. Conceptual/Internal mapping.
2. The External/Conceptual mapping.

1. The Conceptual/Internal Mapping : This mapping defines the correspondence or operations between the conceptual view and the physical view. It specifies how the data is retrieved from physical storage and shown at conceptual level and vice-versa. It specifies how conceptual records and fields are represented at the internal level. It also allows any differences in entity names, attribute names and their orders, data types etc., to be resolved.

2. The External/Conceptual Mapping : This mapping defines the correspondence between the conceptual view and the physical view. It specifies how the data is retrieved from conceptual level and shown at external level because at external level some part of database is hidden from a particular user and even names of data fields are changed etc.

There could be one mapping between conceptual and internal level and several mappings between external and conceptual level. The **physical data independence** is achieved through conceptual/internal mapping while the **logical data independence** is achieved through external/conceptual mapping. The information about the mapping requests among various schema levels are included in the system catalog of DBMS. When schema is changed at some level, the schema at the next higher level remains unchanged, only the mapping between the two levels is changed.

1.7.3 Advantages of Three-level Architecture

The motive behind the three-level architecture is to isolate each user's view of the database from the way the database is physically stored or represented. The advantages of the three-level architecture are as follows :

1. Each user is able to access the same data but have a different customized view of the data as per the requirement.
2. The changes to physical storage organization does not affect the internal structure of the database. *e.g.,* moving the database to a new storage device.
3. To use the database, the user is no need to concern about the physical data storage details.
4. The conceptual structure of the database can be changed by the DBA without affecting any user.
5. The database storage structure can be changed by the DBA without affecting the user's view.

1.7.4 Data Independence

It is defined as the characteristics of a database system to change the schema at one level without having to change the schema at the next higher level. It can also be defined as the immunity of the application programs to change in the physical representation and access techniques of the database. The above definition says that the application programs do not depend on any particular physical representation or access technique of the database. The DBMS achieved the data independence by the use of three-level architecture. The data independence is of TWO types:

1. Physical Data Independence : It indicates that the physical storage structures or devices used for storing the data could be changed without changing the conceptual view or any of the external views. Only the mapping between the conceptual and internal level is changed. Thus, in physical data independence, the conceptual schema insulates the users from changes in the physical storage of the data.

2. Logical Data Independence : It indicates that the conceptual schema can be changed without changing the existing external schemas. Only the mapping between the external and conceptual level is changed and absorbed all the changes of the conceptual schema. DBMS that supports logical data independence, changes to the conceptual schema is possible without making any change in the existing external schemas or rewriting the application programs. Logical data independence also insulates application programs from operations like combining of two records into one or splitting an existing record into more than one records.

1.8 DATA MODELS

A data model is a collection of concepts that can be used to describe the structure of the database including data types, relationships and the constraints that apply on the data.

A data model helps in understanding the meaning of the data and ensures that, we understand.

– The data requirements of each user.
– The use of data across various applications.
– The nature of data independent of its physical representations.

A data model supports communication between the users and database designers. The major use of data model is to understand the meaning of the data and to facilitate communication about the user requirements.

Characteristics of Data Models

A data model must posses the following characteristics so that the best possible data representation can be obtained.

(*i*) Diagrammatic representation of the data model.
(*ii*) Simplicity in designing *i.e.,* Data and their relationships can be expressed and distinguished easily.
(*iii*) Application independent, so that different applications can share it.
(*iv*) Data representation must be without duplication.
(*v*) Bottom-up approach must be followed.
(*vi*) Consistency and structure validation must be maintained.

1.8.1 Types of Data Models

The various data models can be divided into three categories, such as

(*i*) Record Based Data Models.
(*ii*) Object Based Data Models.
(*iii*) Physical Data Models.

(*i*) Record Based Data Models : These models represent data by using the record structures. These models lie between the object based data models and the physical data models. These data models can be further categorised into three types:

(*a*) Hierarchical Data Model
(*b*) Network Data Model
(*c*) Relational Data Model.

(*ii*) Object Based Data Models : These models are used in describing the data at the logical and user view levels. These models allow the users to implicity specify the constraints in the data. These data models can be further categorised into four types:

(*a*) Entity Relationship Model (ER-Model)
(*b*) Object Oriented Model
(*c*) Semantic Data Model
(*d*) Functional Data Model.

The models are discussed in the coming sections.

(*iii*) Physical Data Models : These models provide the concepts that describes the details of how the data is stored in the computer along with their record structures, access paths and ordering. Only specialized or professional users can use these models. These data models can be divided into two types:

(*a*) Unifying Model.

(*b*) Frame Memory Model.

1.8.1.1 Record based Data Models

Record based data models represent data by using the record structures. These are used to describe data at the conceptual view level. These are named because the database is structured in a fixed format records of several types. The use of fixed length records simplify the physical level implementation of the database. These models lie between the object based data models and the physical data models. These models provide the concepts that may be understood by the end users. These data models do not implement the full detail of the data storage on a computer system. Thus, these models are used to specify overall logical structure of the database and to provide high level description of implementation. These are generally used in traditional DBMS's and are also known as 'Representational Data Models'. The various categories of record based data models are as follows:

(*i*) Hierarchical Data Model

(*ii*) Network Data Model

(*iii*) Relational Data Model.

(*i*) Hierarchical Data Model : Hierarchical Data Model is one of the oldest database models. The hierarchical model became popular with the introduction of IBM's Information Management System (IMS).

The hierarchical data model organizes records in a tree structure *i.e.,* hierarchy of parent and child records relationships. This model employs two main concepts : Record and Parent Child Relationship. A record is a collection of field values that provide information of an entity.

A Parent Child Relationship type is a 1 : N relationship between two record types. The record type of one side is called the parent record type and the one on the N side is called the child record type. In terms of tree data structure, a record type corresponds to node of a tree and relationship type corresponds to edge of the tree.

The model requires that each child record can be linked to only one parent and child can only be reached through its parent.

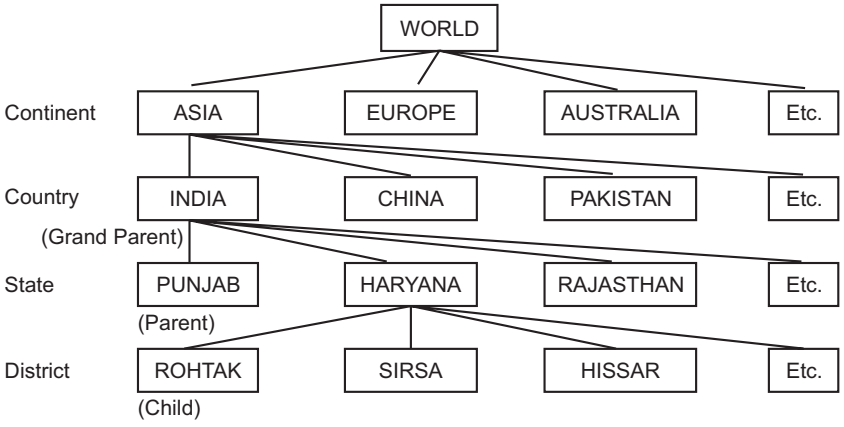


FIGURE 1.8. Hierarchical Model.

In the Figure 1.8, the 'WORLD' acts as a root of the tree structure which has many children's like Asia, Europe, Australia etc. These children can act as a parent for different countries such as ASIA continents acts as a parent for countries like India, China, Pakistan etc. Similarly these children can act as a parent for different states such as INDIA country acts as a parent for states Punjab, Haryana, Rajasthan etc. Further the same follows.

Consider child 'ROHTAK' which has a parent 'HARYANA' which further has a parent 'INDIA' and so on. Now 'India' will acts a grandparent for the child 'ROHTAK'.

The major advantages of Hierarchical Model are that it is simple, efficient, maintains data integrity and is the first model that provides the concept of data security. The major disadvantages of Hierarchical model are that it is complex to implement, Lacking of structural independence, operational anomalies and data management problem.

(*ii*) Network Data Model : As a result of limitations in the hierarchical model, designers developed the Network Model. The ability of this model to handle many to many (N : N) relations between its records is the main distinguishing feature from the hierarchical model. Thus, this model permits a child record to have more than one parent. In this model, directed graphs are used instead of tree structure in which a node can have more than one parent. This model was basically designed to handle non-hierarchical relationships.

The relationships between specific records of 1 : 1 (one to one), 1 : N (one to many) or N : N (many to many) are explicitly defined in database definition of this model.

The Network Model was standardized as the CODASYL DBTG (Conference of Data System Languages, Database Task Group) model.

There are two basic data structures in this model—Records and Sets. The record contains the detailed information regarding the data which are classified into record types. A set type represents relationship between record types and this model use linked lists to represent these relationships. Each set type definition consists of three basic elements : a name for set type an owner record type (like parent) and a member record type (like child).

To represent many to many relationship in this model, the relationship is decomposed into two one to many (1 : N) relationships by introducing an additional record type called an Intersection Record or *Connection Record.*

The major advantages of Network Model are that it is conceptually simple, Handles more relationship types, promotes database integrity, data access flexibility and conformance to the standards.

The major disadvantages of Network Model are that it is complex and lack of structural independence.

(*iii*) Relational data Model : The Relational Model was first introduced by Dr. Edgar Frank, an Oxford-trained Mathematician, while working in IBM Research Centre in 1970's.

The Relational Model is considered one of the most popular developments in the database technology because it can be used for representing most of the real world objects and the relationships between them.

The main significance of the model is the absolute separation of the logical view and the physical view of the data. The physical view in relational model is implementation dependent and not further defined.

The logical view of data in relational model is set oriented. A relational set is an unordered group of items. The field in the items are the columns. The column in a table have names.

The rows are unordered and unnamed. A database consists of one or more tables plus a catalogue describing the database.

The relational model consists of three components:

1. A structural component—A set of tables (also called relations) and set of domains that defines the way data can be represented.
2. A set of rules for maintaining the integrity of the database.
3. A manipulative component consisting of a set of high-level operations which act upon and produce whole tables.

In the relational model the data is represented in the form of tables which is used interchangeably with the word **Relation.** Each table consists of rows also knowns as **tuples** (A tuple represents a collection of information about an item, *e.g.,* student record) and column also known as **attributes.** (An attribute represents the characteristics of an item, *e.g.,* Student's Name and Phone No.). There are relationships existing between different tables. This model doesn't require any information that specifies how the data should be stored physically.

The major advantages of Relational Model are that it is structurally independent, improved conceptual simplicity adhoc query capability and powerful DBMS. The major disadvantages of relational model are substantial hardware and software overhead and facilitates poor design and implementation.

1.8.1.2 Object Based Data Models

Object Based Data Models are also known as conceptual models used for defining concepts including entries, attributes and relationships between them. These models are used in describing data at the logical and user view levels. These models allow the constraints to be specified on the data explicitly by the users.

An entity is a distinct object which has existence in real world. It will be implemented as a table in a database.

An attribute is the property of an entity, in other words, attribute is a single atomic unit of information that describes something about its entity. It will be implemented as a column or field in the database.

The associations or links between the various entities is known as relationships.

There are 4 types of object based data models. These are:

(*a*) Entity-relationship (E-R) Model

(*b*) Object-Oriented Model

(*c*) Semantic Data Model

(*d*) Functional Data Model

These are discussed as follows:

(*a*) Entity-Relationship (E-R) Model : The E-R model is a high level conceptual data model developed by Chen in 1976 to facilitate database design. The E-R model is the generalization of earlier available commercial model like the hierarchical and network model. It also allows the representation of the various constraints as well as their relationships.

The relationship between entity sets is represented by a name. E-R relationship is of 1 : 1, 1 : N or N : N type which tells the mapping from one entity set to another.

E-R model is shown diagrammatically using entity-relationship (E-R) diagrams which represents the elements of the conceptual model that show the meanings and relationships

between those elements independent of any particular DBMS. The various features of E-R model are:

(*i*) E-R Model can be easily converted into relations (tables).

(*ii*) E-R Model is used for purpose of good database design by database developer.

(*iii*) It is helpful as a problem decomposition tool as it shows entities and the relationship between those entities.

(*iv*) It is an iterative process.

(*v*) It is very simple and easy to understand by various types of users.

The major advantages of E-R model are that it is conceptually simple, have vishal representation, an effective communication tool and can be integrated with the relational data model.

The major disadvantages of E-R model are that there are limited constraint representation, limited relationship representation, no data manipulation language and loss of information content.

(*b*) Object-Oriented Data Model : Object-oriented data model is a logical data model that captures the semantics of objects supported in an object-oriented programming. It is based on collection of objects, attributes and relationships which together form the static properties. It also consists of the integrity rules over objects and dynamic properties such as operations or rules defining new database states.

An **object** is a collection of data and methods. When different objects of same type are grouped together they form a **class.** This model is used basically for multimedia applications as well as data with complex relationships. The object model is represented graphically with object diagrams containing object classes. Classes are arranged into hierarchies sharing common structure and behaviour and are associated with other classes.

Advantages of Object-Oriented Data Models

The various advantages of object-oriented data model are as follows:

(*i*) Capability to handle various data types : The object-oriented databases has the capability to store various types of data such as text, video pictures, voices etc.

(*ii*) Improved data access : Object oriented data models represent relationships explicitly. This improves the data access.

(*iii*) Improved productivity : Object-oriented data models provide various features such as inheritance, polymorphism and dynamic binding that allow the users to compose objects. These features increase the productivity of the database developer.

(*iv*) Integrated application development system : Object-oriented data model is capable of combining object-oriented programming with database technology which provides an integrated application development system.

Disadvantages of Object-Oriented Data Models

The various disadvantages of object-oriented data models are as follows:

(*i*) Not suitable for all applications : Object-oriented data models are used where there is a need to manage complex relationships among data objects. They are generally

suited for applications such as e-commerce, engineering and science etc. and not for all applications.

(*ii*) No precise definition : It is difficult to define what constitutes an object-oriented DBMS since the name has been applicable to wide variety of products.

(*iii*) Difficult to maintain : The definition of object is required to be changed periodically and migration of existing databases to confirm to the new object definition. It creates problems when changing object definitions and migrating databases.

(*c*) Semantic Data Models : These models are used to express greater interdependencies among entities of interest. These interdependencies enable the models to represent the semantic of the data in the database. This class of data models are influenced by the work done by artificial intelligence researchers. Semantic data models are developed to organize and represent knowledge but not data. This type of data models are able to express greater interdependencies among entities of interest. Mainframe database are increasingly adopting semantic data models. Also, its growth usage is seen in PC's. In coming times database management systems will be partially or fully intelligent.

(*d*) Functional Data Model : The functional data model describes those aspects of a system concerned with transformation of values-functions, mappings, constraints and functional dependencies. The functional data model describes the computations within a system. It shows how output value in computation are derived from input values without regard for the order in which the values are computed. It also includes constraints among values. It consists of multiple data flow diagrams. Data flow diagrams show the dependencies between values and computation of output values from input values and functions, without regard for when the functions are executed. Traditional computing concepts such as expression trees are examples of functional models.

1.8.2 Comparison of Various Data Models

The most commonly used data models are compared on the basis of various properties. The comparison table is given below.

Property	Hierarchical	Network	Relational	E-R Diagram	Object-oriented
1. Data element organization	Files, records	Files, records	Tables/tuples	Objects, entity sets	Objects
2. Identity	Record based	Record based	Value based	Value based	Record based
3. Data Independence	Yes	Yes	Yes	Yes	Yes
4. Relationship Organization	Logical proximity in a linearised tree.	Intersecting Networks	Identifiers of rows in one table are embedded as attribute values in another table.	Relational extenders that support specialized applications.	Logical containment
5. Access Language	Procedural	Procedural	Non-procedural	Non-procedural	Procedural
6. Structural Independence	No	No	Yes	Yes	Yes

1.8.3 Which Data Models to Use?

So far we have discussed a large number of data models. Data models are essential as they provide access techniques and data structure for defining a DBMS. In other words, a data model describe the logical Organization of data along with operations that manipulate the data.

We have large number of data models, the one which is best for the Organization depends upon the following factors:

- Is the database too small or too big.
- What are the costs involved.
- The volume of daily transactions that will be done.
- The estimated number of queries that will be made from the database by the organization to enquire about the data.
- The data requirements of the organization using it.

From the available record based data models, the relational data model is most commonly used model by most of the organizations because of the following reasons:

1. It increases the productivity of application programmers in designing the database. Whenever changes are made to the database there is no need of changing the application programs because of separation of logical level from conceptual level.
2. It is useful for representing most of the real world objects and relationships between them.
3. It provides very powerful search, selection and maintenance of data.
4. It hides the physical level details from the end users so end users are not bothered by physical storage.
5. It provides data integrity and security so that data is not accessed by unauthorized users and data is always accurate.
6. It provides adhoc query capability.

Some of the common DBMS using Relational model are MS-Access, Informix, Ingres, Oracle etc.

The hierarchical data model is used in those organizations which use databases consisting of large number of one to many relationships. Because of the restriction to one to many relationships, complexity of tree structure diagrams, lack of declarative querying facilities the hierarchical model lost its importance.

The network data model is used in those organizations which use databases consisting of large number of many to many relationships, but due to its complex nature it is also not preferred.

Most of the DBMS use object oriented data modelling techniques which are used by large number of organizations. For example—Latest versions of oracle are object relational hybrids because they support both relational and Object Oriented features.

1.9 TYPES OF DATABASE SYSTEMS

The database systems can be classified into three categories *i.e.,*

(*i*) According to the number of users

(*ii*) According to the type of use

(*iii*) According to database site locations

The various types of database systems are as follows:

1.9.1 According to the Number of Users

According to the number of users, the database systems can be further subdivided into two categories, namely:

(*a*) Single-user database systems

(*b*) Multiuser database systems.

(*a*) Single-user database systems : In a single user database system, the database reside on a PC–on the hard disk. All the applications run on the same PC and directly access the database. In single user database systems, the application is the DBMS. A single user accesses the applications and the business rules are enforced in the applications running on PC. A single user database system is shown in Figure 1.9. The example is DBASE files on a PC.

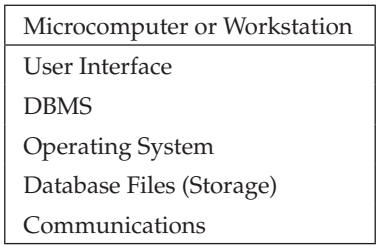


FIGURE 1.9. Single user database system.

(*b*) Multiuser database systems : In a multiuser database system, many PC's are connected through a Local Area Network (LAN) and a file server stores a copy of the database files. Each PC on the LAN is given a volume name on the file server. Applications run on each PC that is connected to the LAN and access the same set of files on the file server. The application is the DBMS and each user runs a copy of the same application and accesses the same files. The applications must handle the concurrency control and the business rules are enforced in the application. The example is MS-Access or Oracle files on a file server. A multiuser database system is shown in Figure 1.10.

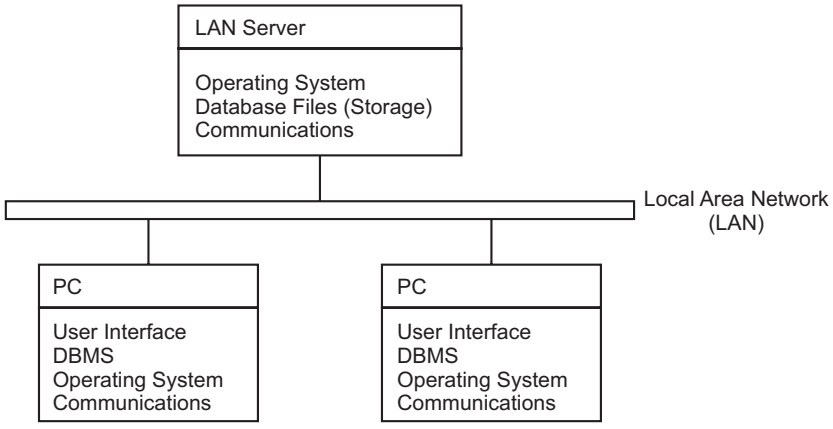


FIGURE 1.10. Multiuser database system.

Advantages of Multiuser Database System

There are many advantages of multiuser database system. Some of them are as follows:

(*i*) Ability to share data among various users.

(*ii*) Cost of storage is now divided among various users.

(*iii*) Low cost since most components are now commodity items.

Disadvantages of Multiuser Database System

The major disadvantage of the multiuser database system is that it has a limited data sharing ability *i.e.,* only a few users can share the data at most.

1.9.2 According to the Type of Use

According to the type of use, the database systems can be further subdivided into three categories, namely:

(*a*) Production or Transactional Database Systems

(*b*) Decision Support Database Systems

(*c*) Data Warehouses.

(*a*) Production or Transactional Database Systems : The production database systems are used for management of supply chain and for tracking production of items in factories, inventories of items in warehouses/stores and orders for items. The transactional database systems are used for purchases on credit cards and generation of monthly statements. They are also used in Banks for customer information, accounts, loans and banking transactions.

(*b*) Decision Support Database Systems : Decision support database systems are interactive, computer-based systems that aid users in judgement and choice activities. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model based reasoning. They support framing, modelling and problem solving. Typical application areas of DSS's are management and planning in business, health care, military and any area in which management will encounter complex decision situations. DSS's are generally used for strategic and tactical decisions faced by upper level management *i.e.,* decisions with a reasonably low frequency and high potential consequences.

A database system serves as a databank for the DSS. It stores large quantities of data that are relevant to the class of problems for which the DSS has been designed and provides logical data structures with which the users interact. The database system is capable of informing the user the types of data that are available and how to gain access to them.

(*c*) Data Warehouses : A data warehouse is a relational database management system (RDMS) designed specifically to meet the transaction processing systems. It can be loosely defined as any centralised data repository which can be queried for business benefit.

1.9.3 According to Database Site Locations

According to database site locations, database systems can be further subdivided into four categories namely:

(*a*) Centralized database systems

(*b*) Parallel database systems

(*c*) Distributed database systems

(*d*) Client/Server database systems.

(*a*) Centralized database systems : The centralised database system consists of a single processor together with its associated data storage devices and other peripherals. Database files resides on a personal computer (small enterprise) or on a mainframe computer (large enterprise). The applications are run on the same PC or mainframe computer. Multiple users access the applications through simple terminals that have no processing power of their own. The user interface is text-mode screens and the business rules are enforced in the applications running on the mainframe or PC. The example of centralized database system is DB2 database and Cobol application programs running on IBM 390.

The centralized database system is shown in Figure 1.11.

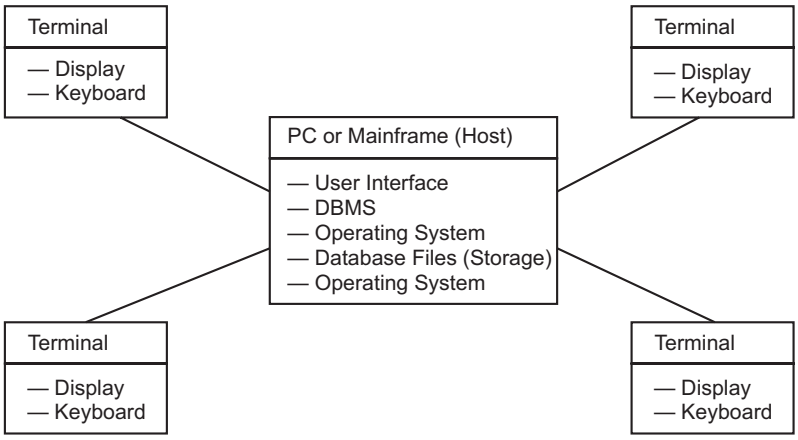


FIGURE 1.11. Centralized database system.

Advantages of Centralized Database System

There are many advantages of centralized database system some of them are as follows:

(*i*) The control over applications and security is excellent.

(*ii*) The incremental cost per user is very low.

(*iii*) The centralized systems are highly reliable due to proven mainframe technology.

(*iv*) Many functions such as query, backup, update etc., are easier to accomplish.

Disadvantages of Centralized Database System

The various disadvantages of centralized database system are as follows:

(*i*) The users are not able to effectively manipulate data outside of standard applications.

(*ii*) The system is not able to effectively serve advance user interfaces.

(*iii*) The failure of central computer blocks every user from using the system until the system comes back.

(*iv*) The communication costs from the terminal to the central computer is a matter of concern.

(*b*) Parallel database systems : A parallel database system can be defined as a database system implemented on a tightly coupled multiprocessor or on a loosely coupled multiprocessor. Parallel database systems link multiple smaller machines to achieve the same throughput as a single larger machine, often with greater scalability and reliability than single processor

database system. Parallel database systems are used in the applications that have to query extremely large databases or have to process an extremely large number of transactions per second. There are three main architectures for parallel database system. These are

(*i*) Shared memory architecture

(*ii*) Shared disk architecture

(*iii*) Shared nothing architecture.

More about these types is discussed in Chapter 12.

Advantages of Parallel Database Systems

There are many advantages of parallel database systems. Some of these are as follows:

(*i*) These are very useful in the applications where large databases have to be queried or where extremely large number of transactions per second has to be processed.

(*ii*) The response time is very high.

(*iii*) The throughput is also very high.

(*iv*) The input/output speeds and processing is very high.

(*v*) They have greater scalability and reliability than single processor system.

Disadvantages of Parallel Database Systems

The various disadvantages of parallel database systems are as follows:

(*i*) Due to start-up cost and start-up time, the overall speed up is adversely affected.

(*ii*) Due to processes executed in parallel, sharing the resources, a slow down may result offer each new process as it competes with existing processes for the resources.

(*c*) Distributed database systems : A distributed database system is a database system, in which, the data is spread across a variety of different databases. These are managed by a variety of DBMS's that are running on various types of machines having different operating systems. These machines are widely spread and are connected through the communication networks. Each machine can have is own data and applications, and can access data stored on other machines. Thus, each machine acts as a server as well as client.

Thus, distributed database system is a combination of logically interrelated databases distributed over a computer network and the distributed database management system (DDBMS). A distributed database system can be homogeneous or heterogeneous. A distributed database system is shown in Figure 1.12.

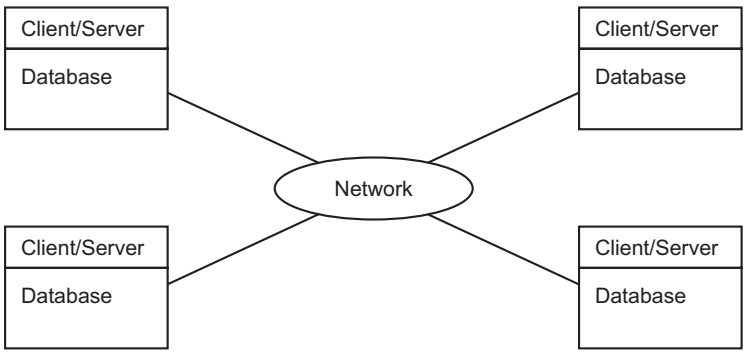


FIGURE 1.12. Distributed database system.

Advantages of Distributed Database Systems

The various advantages of distributed database systems are as follows:

1. Improved sharing ability
2. Local autonomy
3. Availability
4. Reliability
5. Improved performance
6. Easier expansion
7. Reduced communications overhead and better response time
8. More economical
9. Direct user interaction
10. No a single point failure
11. Processor independence.

Disadvantages of Distributed Database Systems

The various disadvantages of distributed database systems are as follows:

1. Architectural complexity
2. Lack of standards
3. Lack of professional support
4. Data integrity problems
5. Problem of security
6. High cost
7. Complex database design.

(*d*) Client/Server Database System : With the development of technology, hardware cost become cheaper and cheaper and more personal computers are used. There was a change and enterprises started use of client-server technology instead of centralized system. In client-server technology, there is a server which acts as a whole data base management system and some clients or personal computers which are connected with server through a network interface. The complete architecture is shown in Figure 1.13.

Components of Client-Server Architecture

There are three major components of client server architecture:

1. Server
2. Client
3. Network interface

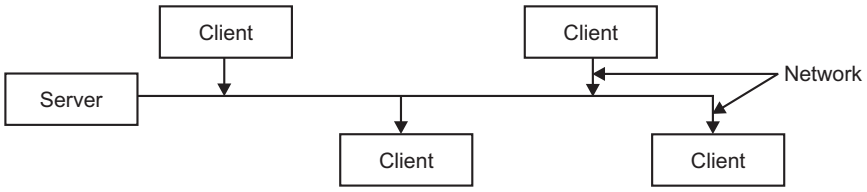


FIGURE 1.13. Client server database system.

1. Server : Server is DBMS itself. It consists of DBMS and supports all basic DBMS functions. Server components of DBMS are installed at server. It acts as monitor of all of its clients. It distributes work-load to other computers. Clients must obey their servers.

Functions of Server : The server performs various functions, which are as follows.

1. It supports all basic DBMS functions.
2. Monitor all his clients.
3. Distribute work-load over clients.
4. Solve problems which are not solved by clients.
5. Maintain security and privacy.
6. Avoiding unauthorized access of data.

2. Clients : Client machine is a personal computer or workstation which provide services to both server and users. It must obey his server. Client components of DBMS are installed at client site. Clients are taking instructions from server and help them by taking their load. When any user want to execute a query on client, the client first take data from server then execute the query on his own hardware and returns the result to the server. As a result, server is free to do more complex applications.

3. Network Interface : Clients are connected to server by network interface. It is useful in connecting the server interface with user interface so that server can run his applications over his clients.

In the client server architecture, there are more than one server. Sometimes, a server is used as Database Server, other as Application Server, other as Backup Server etc.

Advantages of Client-Server Database System

1. It increase the overall performance of DBMS.
2. Load can be distributed among clients.
3. It provides better user interface.
4. It is used to develop highly complex applications.
5. Clients with different operating systems can be connected with each other.
6. Single copy of DBMS is shared.
7. It reduces cost.

Disadvantages of Client-Server Database System

1. Network is error prone.
2. It is a kind of centralized system. If server is crashed or failed, there is loss of data.

3. Recovery is typical and additional burden on DBMS server to handle concurrency control.
4. Programming cost is high.
5. The implementation is more complex since one needs to deal with the middle ware and the network.

1.10 COMPARISON BETWEEN CLIENT/SERVER AND DISTRIBUTED DATABASE SYSTEM

Client/Server Database System	Distributed Database System
1. In this, different platforms are often difficult to manage.	1. In this, different platforms can be managed easily.
2. Here, application is usually distributed across clients.	2. Here, application is distributed across sites.
3. In this database system, whole system comes to a halt if server crashes.	3. Here, failure of one site doesn't bring the entire system down as system may be able to reroute the one site's request to another site.
4. Maintenance cost is low.	4. Maintenance cost is much higher.
5. In this system, access to data can be easily controlled.	5. In DDS not only does the access to replicate the data has to be controlled at multiple locations but also the network has to be made secure.
6. In this, new sites can not be added easily.	6. In this, new sites can be added with little or no problem.
7. Speed of database access is good.	7. Speed of database access is much better.

TEST YOUR KNOWLEDGE

True/False

1. A database actually consists of three parts: information, the logical structure of that information, and tables.
2. A data dictionary, or relation, is a two-dimensional table used to store data within a relational database.
3. A database management system (DBMS) allows you to specify the logical organization for a database and access and use the information within a database.
4. A physical view represents how the users view the data.
5. A database may have numerous physical views.
6. Fixed length record sometimes wastes space while variable length record does not waste space.
7. A database is any collection of related data.
8. A DBMS is a software system to facilitate the creation and maintenance of a computerized database.
9. End-users can be categorized into casual, designer, or parametric users.

10. Data redundancy exists when the same data is stored at multiple places.
11. A database always maintains a collection of unrelated data.
12. A database system is a software system to enable users to create and maintain a computerized database.
13. End-users can be categorized into casual, naïve, sophisticated, or stand-alone users.
14. Typical DBMS functionality is to define and create a particular database in terms of its data types, structures, and constraints.
15. Data redundancy exists when the same data is stored at one place.
16. A database is a very large software system used for processing related data.
17. The DBMS stores definitions of the data elements and their relationships (metadata) in a data dictionary.
18. Data about data is metadata.
19. One of the main functions of a database system is to provide timely answers to end users.
20. To work with data, the DBMS must retrieve the data from permanent storage and place it in RAM.

Fill in the Blanks

1. A(n) _____ contains the logical structure for the information.
2. A(n)_____ represents how data is physically stored on a storage device.
3. A(n)_____ represents how knowledge users see information.
4. A data model is a collection of concepts that can be used to describe the _____ of a database.
5. _____ schema describes physical storage structures and access paths.
6. In three-schema architecture, user views are defined at _____ schema.
7. _____ data model provides concepts that are close to the way many users perceive data.
8. External schema describes the various user _____.
9. A _____ is a collection of concepts that can be used to describe the structure of a database.
10. _____ data models use concepts such entities, attributes, and relationships.
11. Data stored in database at a particular moment in time is a database _____.
12. A _____ is a unit of work that includes one or more reads or updates of database records.
13. The description of schema constructs and constraints is called _____ .
14. The description of a database is called database _____.
15. The database state is called _____ of the schema.

Multiple Choice Questions

1. Manager's salary details are to be hidden from Employees Table. This technique is called as (*UGC-NET*)

(*a*) Conceptual level datahiding
(*b*) Physical level datahiding
(*c*) External level datahiding
(*d*) Logical level datahiding.

2. Which of the following is not a type of database management system? (*UGC-NET*)
   (*a*) Hierarchical (*b*) Network
   (*c*) Relational (*d*) Sequential.
3. A schema describes (*UGC-NET*)
   (*a*) Data elements (*b*) Records and files
   (*c*) Record relationship (*d*) All of the above.
4. Which data management language component enabled the DBA to define schema components? (*UGC-NET*)
   (*a*) DML (*b*) Subschema DLL
   (*c*) Schema DLL (*d*) All of these.
5. Which statement is false regarding data independence? (*UGC-NET*)
   (*a*) Hierarchical data model suffers from data independence.
   (*b*) Network model suffers from data independence.
   (*c*) Relational model suffers from logical data independence.
   (*d*) Relational model suffers from physical data independence.
6. Databases may be more expensive to maintain than files because of
   (*a*) backup and recovery needs
   (*b*) the complexity of the database environment
   (*c*) the need for specialized personnel
   (*d*) all of the above.
7. Typically, a database consists _____ but can support mul _____.
   (*a*) table, queries (*b*) information, data
   (*c*) physical view, logical view (*d*) information view, data view.
8. Which view of information deals with how the information is physically arranged, stored, and accessed?
   (*a*) Physical view (*b*) Logical view
   (*c*) Information view (*d*) None of the above
9. A mail order database has the following conceptual schemas:

   Employees (Eno,ename,zip,hDate)

   Parts(Pno,Pname,Price)

   Customers(Cno,Cname,Street,Zip,Phone)

   Orders(Ono,Cno,Eno,Received,Shipped)

   Odetails(Ono,Pno,Qty)

   ZipCodes(Zip,City)

   and has the following External View Schemas:

   Order_Report(Ono,Cno,Eno,Total Price)

   Consider the following statements

   (*a*) Adding a column zip to the orders schema to keep track of the zip code of each order does not affect the schema of Order_Report.

   (*b*) To improve query processing, a new index was created on the column Cno of Orders table.

(*i*) Which of these statements represent logical data independence?

(*a*) Both (*a*) and (*b*) (*b*) (*a*)

(*c*) (*b*) (*d*) None of (*a*) and (*b*)

(*ii*) Which of these statements represent physical data independence?

(*a*) Both (*a*) and (*b*) (*b*) (*a*)

(*c*) (*b*) (*d*) None of (*a*) and (*b*)

(*iii*) Which of the following is an example of controlled redundancy?

(*a*) Adding a column Ono to the Employees table.

(*b*) Adding a column Ono to the Employees table and making sure that the Ono value in an Employee record matches a value in the Ono column in the Orders table.

(*c*) Removing the column HDate from the Employees table.

(*d*) Both (*a*) and (*b*).

10. By redundancy in a file based system we mean that

(*a*) unnecessary data is stored

(*b*) same data is duplicated in many files

(*c*) data is unavailable

(*d*) files have redundant data.

11. Data integrity in a file based system may be lost because

(*a*) the same variable may have different values in different files

(*b*) files are duplicated

(*c*) unnecessary data is stored in files

(*d*) redundant data is stored in files.

12. Data availability is often difficult in file based system

(*a*) as files are duplicated

(*b*) as unnecessary data are stored in files

(*c*) as one has to search different files and these files may be in different update states

(*d*) redundant data are stored in files.

13. Some of the objectives of a database management system are to

(*i*) minimize duplication of data

(*ii*) ensure centralized management control of data

(*iii*) ease retrieval of data

(*iv*) maintain a data dictionary

(*a*) (*i*) and (*ii*) (*b*) (*i*), (*ii*) and (*iv*)

(*c*) (*i*) and (*iii*) (*d*) (*i*), (*ii*) and (*iii*)

14. A database is a

(*a*) collection of files

(*b*) collection of inputs and outputs of application

(*c*) collection of related data necessary to manage an organization

(*d*) data resource of an organization.

15. One of the main objectives of a DBMS is to

(*a*) create a database for an organization

(*b*) facilitate sharing of a database by current and future applications
(*c*) allow sharing application programs
(*d*) replace file based systems.

16. By data independence we mean application programs
(*a*) do not need data
(*b*) may be developed independent of data
(*c*) may be developed without knowing the organization of data
(*d*) may be developed with independent data.

17. Data independence allows
(*i*) no changes in application programs
(*ii*) change in database without affecting application programs
(*iii*) hardware to be changed without affecting application programs
(*iv*) system software to be changed without affecting application programs
(*a*) (*i*), (*ii*) (*b*) (*ii*), (*iii*)
(*c*) (*ii*), (*iii*), (*iv*) (*d*) (*i*), (*ii*), (*iv*)

18. Data independence allows
(*a*) sharing the same database by several applications
(*b*) extensive modification of applications
(*c*) no data sharing between applications
(*d*) elimination of several application programs.

19. By data integrity we mean
(*a*) maintaining consistent data values (*b*) integrated data values
(*c*) banning improper access to data (*d*) not leaking data values.

20. By data security in DBMS we mean
(*a*) preventing access to data
(*b*) allowing access to data only to authorized users
(*c*) preventing changing data
(*d*) introducing integrity constraints.

21. A subset of logical data model accessed by programmers is called a
(*a*) conceptual data model (*b*) external data model
(*c*) internal data model (*d*) an entity-relation data model.

22. When a logical model is mapped into a physical storage such as a disk store the resultant data model is known as
(*a*) conceptual data model (*b*) external data model
(*c*) internal data model (*d*) disk data model.

23. A DBMS has the following components
(*i*) a data definition language
(*ii*) a query language
(*iii*) a security system
(*iv*) audit trail.
(*a*) (*i*), (*i*) (*b*) (*i*), (*ii*), (*iii*)
(*c*) (*i*), (*ii*), (*iii*), (*iv*) (*d*) (*i*), (*ii*), (*iv*)

24. A database administrator
    (*a*) administers data in an organization
    (*b*) controls all inputs and all outputs of programs
    (*c*) is controller of data resources of an organization
    (*d*) controls all data entry operators.
25. The responsibilities of a database administrator includes
    (*i*) maintenance of data dictionary
    (*ii*) ensuring security of database
    (*iii*) ensuring privacy and integrity of data
    (*iv*) obtain an E-R model
    (*a*) (*i*), (*ii*) (*b*) (*i*), (*ii*), (*iii*)
    (*c*) (*i*), (*ii*), (*iii*), (*iv*) (*d*) (*ii*), (*iii*), (*iv*)
26. The sequence followed in designing a DBMS are
    (*a*) physical model conceptual model logical model
    (*b*) logical model physical model conceptual model
    (*c*) conceptual model logical model physical model
    (*d*) conceptual model physical model logical model.
27. What is data integrity?
    (*a*) It is the data contained in database that is non redundant.
    (*b*) It is the data contained in database that is accurate and consistent.
    (*c*) It is the data contained in database that is secured.
    (*d*) It is the data contained in database that is shared.
28. The metadata is created by the
    (*a*) DML compiler
    (*b*) DML pre-processor
    (*c*) DDL interpreter
    (*d*) Query interpreter
29. Which of the following statement is correct?
    Logical data independence provides following without changing application programs:
    (*i*) Changes in access methods.
    (*ii*) Adding new entities in database
    (*iii*) Splitting an existing record into two or more records
    (*iv*) Changing storage medium
    (*a*) (*i*) and (*ii*) (*b*) (*iv*) only,
    (*c*) (*i*) and (*iv*) (*d*) (*ii*) and (*iii*)
30. Manager salary details are hidden from the employee. This is
    (*a*) Conceptual level data hiding. (*b*) External level data hiding.
    (*c*) Physical level data hiding. (*d*) None of these.
31. A logical schema
    (*a*) is the entire database.
    (*b*) is a standard way of organizing information into accessible parts.

(*c*) describes how data is actually stored on disk.

(*d*) both (*a*) and (*c*)

32. The database environment has all of the following components except:

(*a*) users. (*b*) separate files.

(*c*) database. (*d*) database administrator.

33. A subschema expresses

(*a*) the logical view. (*b*) the physical view.

(*c*) the external view. (*d*) all of the above.

34. Which one of the following statements is false?

(*a*) The data dictionary is normally maintained by the database administrator.

(*b*) Data elements in the database can be modified by changing the data dictionary.

(*c*) The data dictionary contains the name and description of each data element.

(*d*) The data dictionary is a tool used exclusively by the database administrator.

35. Data independence means

(*a*) data is defined separately and not included in programs.

(*b*) programs are not dependent on the physical attributes of data.

(*c*) programs are not dependent on the logical attributes of data.

(*d*) both (*b*) and (*c*).

36. DBMS helps achieve

(*a*) Data independence (*b*) Centralized control of data

(*c*) Neither (*a*) nor (*b*) (*d*) Both (*a*) and (*b*)

37. It is better to use files than a DBMS when there are

(*a*) Stringent real-time requirements. (*b*) Multiple users wish to access the data.

(*c*) Complex relationships among data. (*d*) All of the above.

38. A data dictionary is a special file that contains:

(*a*) The name of all fields in all files. (*b*) The width of all fields in all files.

(*c*) The data type of all fields in all files. (*d*) All of the above.

39. Which of the following is incorrect?

(*a*) Database state is the actual data stored in a database at a particular moment in time.

(*b*) Database state is called database intension.

(*c*) Database state is called database instance.

(*d*) Database schema is a description of the structure of the data in a database.

40. Which of the following is correct?

(*a*) Database schema changes frequently, but database state does not change.

(*b*) Database schema is specified during database design.

(*c*) Database schema changes frequently.

(*d*) The database schema changes more often than the database state.

41. A DBMS that supports a database located at multiple sites is called _____ DBMS.

(*a*) centralized (*b*) multi-user

(*c*) distributed (*d*) single-user

42. Select the incorrect statement about database.
    (*a*) It represents some aspect of the real world.
    (*b*) It is a random assortment of data.
    (*c*) It is designed, built, and populated with data for a specific purpose.
    (*d*) It is a collection of related data.
43. Using DBMS has many advantages , except _____.
    (*a*) Increases redundancy
    (*b*) Restricts unauthorized access
    (*c*) Provides backup and recovery
    (*d*) Enforces integrity constraints
44. Consider the following schema for an investment portfolio database
    (*a*) Member (MemberId, Password,FName,LName)
    Security (SId,SName, CurrentPrice, AskPrice,BidPrice)
    Transaction (MemberId, SId, Tdate, Ttype,Qty,Price)
    (*b*) Member_Transaction(MemberId,Fname,LName,Tdate,Type,Qty,Price) seen by Members and Administrators
    Member_Password(MemberId,Password) seen only by Members
    (*c*) Data_Layout(Table_Name,Data_Item_Name, Starting_Position,Length_In_Bytes)
    (*i*) Which of the above schemas is an Internal Schema?
        (*a*) Only (*a*)
        (*b*) Only (*b*)
        (*c*) Only (*c*)
        (*d*) All of the above
    (*ii*) Which of the above schemas is a Conceptual Schema?
        (*a*) Only (*a*)
        (*b*) Only (*b*)
        (*c*) Only (*c*)
        (*d*) All of the above
    (*iii*) Which of the above schemas is an External Schema?
        (*a*) Only (*a*)
        (*b*) Only (*b*)
        (*c*) Only (*c*)
        (*d*) All of the above
45. Match the following:
    1. DDL (*a*) Manipulates the data base
    2. SDL (*b*) Specifies user views and their mappings
    3. VDL (*c*) Specifies internal schema
    4. DML (*d*) Specifies conceptual and internal schema both or conceptual schema only
        (*a*) 1-*a*, 2-*c*, 3-*d*, 4-*b*
        (*b*) 1-*d*, 2-*c*, 3-*b*, 4-*a*
        (*c*) 1-*b*, 2-*d*, 3-*a*, 4-*c*
        (*d*) 1-*c*, 2-*d*, 3-*a*, 4-*b*
46. Match the following:
    1. Relational data model
    2. Network model
    3. Hierarchical model
    (*a*) Represents database as collection of tables
    (*b*) Represents data as tree structures
    (*c*) Represents data as record types
        (*a*) 1-*c*, 2-*a*, 3-*b*
        (*b*) 1-*b*, 2-*c*, 3-*a*
        (*c*) 1-*b*, 2-*a*, 3-*c*
        (*d*) 1-*a*, 2-*c*, 3-*b*
47. Match the following:
    1. View level (*a*) Describes the part of database for a particular user group
    2. Conceptual level (*b*) Describes the structure of whole database for community of users

3. Internal level (*c*) Describes the part of database for a particular user group

(*a*) 1-*c*, 2-*b*, 3-*a* (*b*) 1-*b*, 2-*c*, 3-*a*

(*c*) 1-*b*, 2-*a*, 3-*c* (*d*) 1-*a*, 2-*c*, 3-*b*

48. Program-data dependence is caused by

(*a*) data descriptions being stored on a server.

(*b*) data cohabiting with programs.

(*c*) file descriptors being stored in each application.

(*d*) data descriptions being written into programming code.

ANSWERS

True/False

1. False
2. False
3. True
4. False
5. False
6. False
7. True
8. True
9. False
10. True
11. False
12. True
13. True
14. True
15. False
16. False
17. True
18. True
19. True
20. True

Fill in the Blanks

1. Data dictionary
2. Physical view
3. Logical view
4. structure
5. Conceptual
6. external
7. External
8. views
9. data model
10. conceptual
11. state
12. Transaction
13. Meta data
14. Schema
15. Extension

Multiple Choice Questions

1. (*c*)
2. (*d*)
3. (*d*)
4. (*d*)
5. (*d*)
6. (*d*)
7. (*c*)
8. (*a*)
9. (*i*) (*b*) (*ii*) (*c*) (*iii*) (*b*)
10. (*b*)
11. (*a*)
12. (*c*)
13. (*d*)
14. (*c*)
15. (*b*)
16. (*c*)
17. (*c*)
18. (*a*)
19. (*a*)
20. (*b*)
21. (*b*)
22. (*c*)
23. (*c*)
24. (*c*)
25. (*b*)
26. (*c*)
27. (*b*)
28. (*c*)
29. (*d*)
30. (*a*)
31. (*a*)
32. (*a*)
33. (*c*)
34. (*b*)
35. (*d*)
36. (*d*)
37. (*b*)
38. (*d*)
39. (*b*)

40. (*b*) 41. (*c*) 42. (*b*)
43. (*a*) 44. (*i*) (*c*) (*ii*) (*a*) (*iii*) (*b*) 45. (*b*)
46. (*d*) 47. (*a*) 48. (*c*)

EXERCISES

Short Answer Questions

1. What is data?
2. What is Information?
3. What is the difference between data and information?
4. What is Metadata?
5. Explain various types of Metadata?
6. What is data dictionary?
7. What is active data dictionary?
8. What is passive data dictionary?
9. What is the difference between active and passive data dictionary?
10. What is data base?
11. What are the main characteristics of a database?
12. What are the capabilities of a database?
13. Define database management system?
14. What are the various functions of DBMS?
15. What are the criteria of classifying DBMS?
16. What is a field?
17. What is record?
18. What is a file?
19. Differentiate between field, record and file?
20. Give names of components of database?
21. What are the main components of DBMS?
22. What is traditional file system?
23. How traditional file system is different from database system?
24. What are the disadvantages of traditional file system?
25. What is database system?
26. What are the components of database system?
27. What are the advantages of database system?
28. What are the disadvantages of database system?
29. What are various users of DBMS?
30. List the people associated with the database.
31. What is DBA?
32. What are the responsibilities of DBA?

33. List 5 DBA Activities in the order that they are most performed.

Ans. Backup/Restore, Startup/Shutdown, Capacity Planning (Disk Space), Performance, Connectivity, Transactional Problems (Concurrency, etc.)

34. What are the various languages of DBMS?

35. What is DDL?

36. What is SDL?

37. What is VDL?

38. What is DML?

39. What is 4GL?

40. What is the difference between DDL and DML?

41. What is an instance?

42. What is schema?

43. What is subschema?

44. What is the difference between schema and subschema?

45. What is the difference between schema and instance?

46. What is physical schema?

47. What is logical schema?

48. What is conceptual schema?

49. What are the three levels of three-tier architecture?

50. What is conceptual/Internal mapping?

51. What is external/conceptual mapping?

52. What are the advantages of three level architecture?

53. What is data independence?

54. What is physical data independence?

55. What is logical data independence?

56. What is the difference between physical and logical data independence?

57. Give some applications of DBMS.

58. What are the advantages of using a DBMS?

59. What are the criteria of classifying DBMS?

60. Give the levels of data abstraction?

61. What is storage manager?

62. What is an entity relationship model?

63. Define data model?

64. What are the categories of data models?

Long Answer Questions

1. What do you mean by data? How is it different from information, explain by example?

2. What are the four major components of database system? Explain.

3. What are the advantages of database systems? Explain in detail.

4. What is DBMS? What are the advantages and disadvantages offered by such systems as compared to file processing system? Explain.

5. What is schema and subschema? Why should a subschema be independent of schema? Also explain what is logical schema and physical schema and their differences?

6. Explain the terms:

Database and DBMS with examples. Explain the three schema architecture of a DBMS with the help of diagram. Why do we need mappings between the different schema levels? Also explain the main advantages of DBMS.

7. What are the main responsibilities of DBA? Explain.

8. Explain the term DBMS. Discuss the responsibilities of DBA. Also explain the three level architecture of DBMS.

9. Explain data independence and its types.

10. What is DBMS? What are the main facilities that every DBMS should provide? Explain.

11. Explain what is schema? What is subschema and why subschema be independent of schema? Also explain the various types of data independence and their advantages.

12. What is meant by Data Independence? State its importance in database technology.

13. Describe the architecture of a database system. Why a database is desired to be an integrated one?

14. Define data, database, DBMS, record, file and field by giving example of each.

15. Differentiate between the following:

(*i*) Physical data independence and logical data independence.

(*ii*) A normal file system and database management system.

16. Comment upon the following:

(*i*) Various components of DBMS. (*ii*) DDL and DML.

17. Explain the client server architecture. Also write advantages and disadvantages of it.

18. Explain different types of DBMS languages with example of each.

19. Explain different types of DBMS user's with their jobs and responsibilities.

20. Explain data dictionary by giving suitable example.

Chapter 2

E-R AND EER MODELS

2.1 INTRODUCTION

The entity-relationship (E-R) model was introduced by Chen in 1976. He described the main constructs of the E-R model *i.e.,* entities, relationships and their associated attributes. The E-R model continues to evolve but there is not yet a standard notation for E-R modeling. E-R model is mainly used for conceptual data modeling. It is popular due to the factors such as relative ease of use, CASE tool support, and the belief that entities and relationships are natural modeling concepts in the real world.

The E-R model is mainly used for communication between database designers and end users during the analysis phase of database development. This E-R model is a representation of the structure and constraints of a database that is independent of the DBMS and its associated data model that will be used to implement the database.

In 1980's many new database applications like Computer Aided Manufacturing (CAM), Computer Aided Design (CAD), Computer Aided Software Engineering (CASE), Digital publishing, World Wide Web (WWW), Telecommunication applications etc., were introduced. The basic E-R modeling concepts were no longer sufficient to represent the requirement of these newer and complex applications. Basic E-R model was not capable to represent additional semantic modeling concepts. Database designers and practitioners introduced a new model named as *Enhanced Entity-Relationship Model (EER)* which includes the basic E-R model concepts with additional semantic concepts like:

- Specialization
- Generalization
- Categorization.

The chapter describes the basic E-R model and in later sections it describes EER model.

2.2 BASIC CONCEPTS

1 Enterprise

Enterprise refers to any kind of organization.

Ex. Colleges, schools, banks, any company etc.

2 Entity

Entity refers to an "object" or "thing" in real world. Object may be any person, place, event etc.

Ex. Students of colleges and schools, loans in banks, employees in any company etc.

3 Attributes

These are the characteristics of any entity.

Ex., (*i*) A student can be described by his name, age, address, height, class etc.

(*ii*) Loans can be described by their types such as house loan, car loan etc.

(*iii*) Employees in any company can be described by their Employee ID, name, department, designation etc.

(*iv*) A car can be described by his color, model, company etc.

4 Value

Value is the information or data which is stored in attributes of any entity.

5 Entity Sets

All the entities having same attributes make an entity set.

6 Domain

Domain or value set is the set of all values or information about any attribute.

Ex. Consider the *student* table shown in Figure 2.1. It describes the basic concepts.

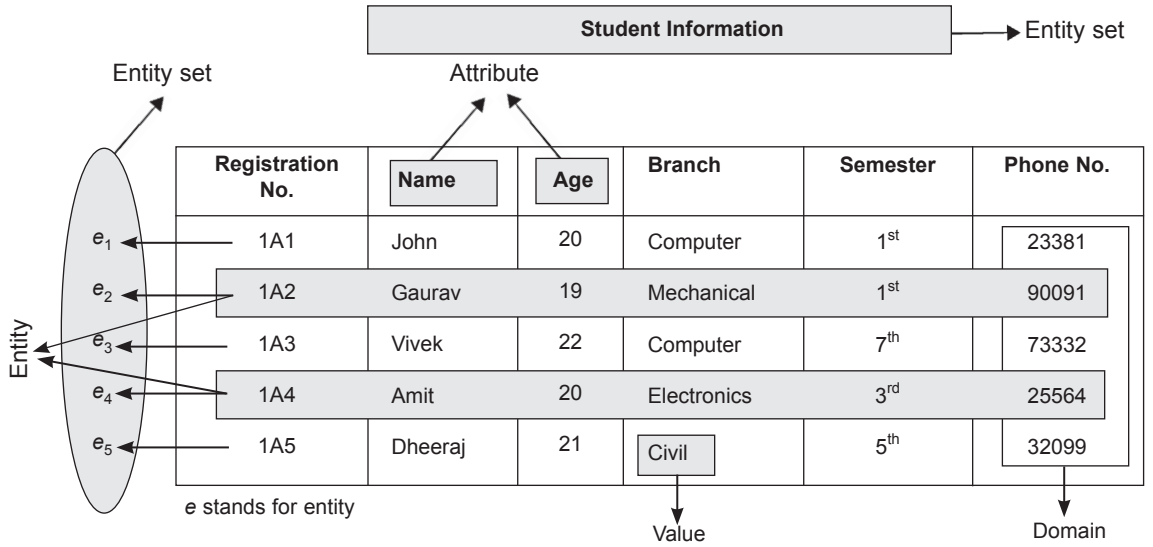


FIGURE 2.1. Table showing basic concepts of E-R model.

(*a*) *Enterprise :* Here, enterprise is college where students are studied.

(*b*) *Entity :* Here, entity refers to any single student with all his values.

Ex.

1A1	John	20	Computer	1st	23381

(*c*) *Attributes :* The students are described by Registration No., Name, Age, Branch, Semester, Phone No. These are the attributes of students.

(*d*) *Value :* The values are 1A1, 21, Civil, Gaurav, 90091, 5th etc.

(*e*) *Entity Set :* All students are described by same set of attributes. So, all these students combine together to make an entity set "Student Information".

(*f*) *Domain :* (Value set) for, attribute, *Name* it is John, Gaurav, Vivek, Amit, Dheeraj and for *Age*, it is 20, 19, 21, 22.

2.3 TYPES OF ATTRIBUTES

Attributes can be characterized by the following **Three** major types :

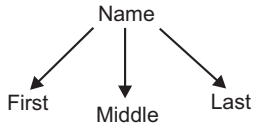
2.3.1 Simple and Composite Attributes

Simple Attributes are those which cannot be divided into subparts.

Ex. Age of student

Composite Attributes are those which can be divided into subparts.

Ex. Name of a student can be divided into First Name, Middle Name, and Last Name.



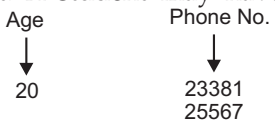
2.3.2 Single Valued and Multi-valued Attributes

Single Valued Attribute : An attribute having only single value for a particular entity is known as single value attribute.

Ex. Age of student.

Multi-Valued Attributes : An attribute having more than one possible value for a particular entity is known as multi-valued attribute.

Ex. Phone number of a student. A student may have more than one phone.



2.3.3 Derived Attributes and Stored Attributes

Derived Attributes : An attribute that can be derived from other known attributes is known as derived attribute.

Ex. Age of employees can be derived if you know date of birth and system date.

Age = System date – Date of birth

Stored Attributes : An attribute which cannot be derived by other known attributes is known as stored attribute.

Ex. Date of birth of any employee.

NULL Value : Null stands for nothing. An attribute have a null value if either the value of that attribute is not known or the value is not applicable.

Caution : NULL is not equal to Zero (0). But you can say that NULL is blank as shown in Figure 2.2.

Ex.

Name	Subject	Marks
abc	Maths	92
def	Science	— → This is null
ghi	Maths	0

FIGURE 2.2. Showing null value.

2.4 RELATIONSHIP SETS

1. **Relationship :** A relationship is the association among several entities. It connects different entities through a meaningful relation.
2. **Relationship Set :** A relationship set is a set of relationships of the same type.

Consider an example, employees work in different departments. Then relationship exists between employees and departments because each employee must belongs to some department. Relation of all employees with department when combined makes the relationship set because each employee has same kind of relation with departments.

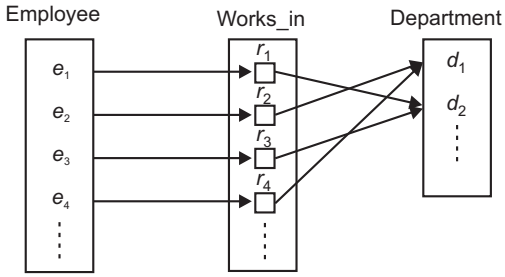


FIGURE 2.3. Binary relationship set.

Here, Employee and Department are two entity sets. *r* stands for relationship between Employee and Department. Works_in is the *relationship set* as shown in Figure 2.3.

- **Descriptive Attributes :** Attributes of any relationship set are known as descriptive attributes.

2.4.1 Degree of Relationship Sets

Total number of entity sets participate in a relationship set is known as degree of that relationship set.

1. Binary Relationship Set

A relationship set in which only two entity sets are involved is known as binary relationship set.

Ex. The Figure 2.3 shows the Binary relationship set.

2. Ternary Relationship Set

A relationship set in which three entity sets are involved is known as ternary relationship set or a relationship set having degree three.

Ex. The Figure 2.4 shows the relationship set works_in, which is a ternary relationship set.

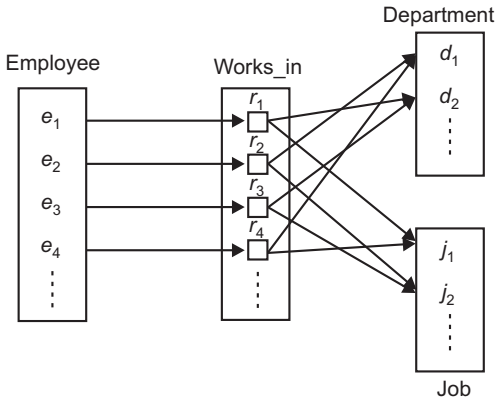


FIGURE 2.4. Ternary relationship set.

2.4.2 Role and Recursive Relationship Set

- **Role :** The function of any entity which it plays in relationship set is called that entity's role. *e.g.,* employee plays the role of worker in his department in Figure 2.4.
- **Recursive Relationship Set :** When the same entity sets participate in same relationship set more than once with different roles each time, then this type of recursive relationship set is known as Recursive Relationship set. *e.g.,* consider an example of relationship set works_in and two entity set student and college. A student who attends weekend classes in college as student may also be lecturer in that college. Then this person plays two roles (student, faculty) in same relationship set work_in.

2.5 MAPPING CONSTRAINTS

There are certain constraints in E-R model. Data in the database must follow the constraints. Constraints act as rules to which the contents of database must conform. There are *two types* of mapping constraints : (*a*) *Mapping cardinalities*, (*b*) *Participation constraints.*

2.5.1 Mapping Cardinalities (Cardinality Ratios)

It specifies the number of entities of an entity set that are associated with entities of another entity set through a relationship set.

Mapping Cardinalities are helpful in describing *binary relationship sets.*

Two entity sets X and Y having binary relationship set R must have one of the following mapping cardinality :

1. One to One (1 : 1) : An entity in X is associated with at most one entity in Y and an entity in Y is associated with at most one entity in X.

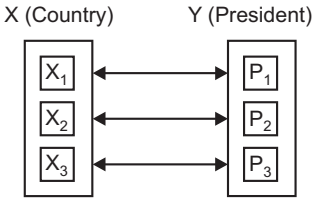


FIGURE 2.5. One to one cardinality ratio.

A country has only one president. Any person may be the president of at most one country.

2. One to Many (1 : N) : An entity in X is associated with any number of entities in Y. An entity in Y is associated with at most one entity in X.

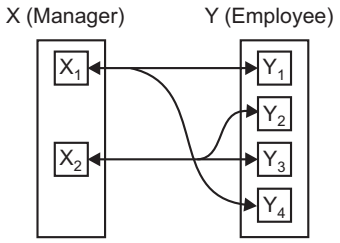


FIGURE 2.6. One to many cardinality ratio.

A manager has many employees under it but an employee works under only one manager.

3. Many to One (N : 1) : An entity in X is associated with at most one entity in Y. An entity in Y is associated with any number of entities in X.

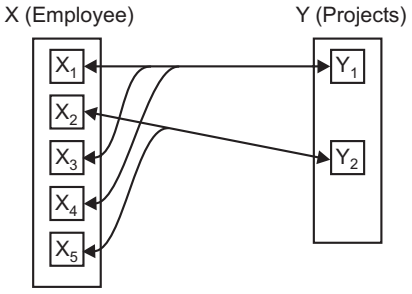


FIGURE 2.7. Many to one cardinality ratio.

A employee can work on single project while any project can be assigned to more than one employee.

4. Many to Many (M : N) : An entity in X is associated with any number (zero or more) of entities in Y and vice versa.

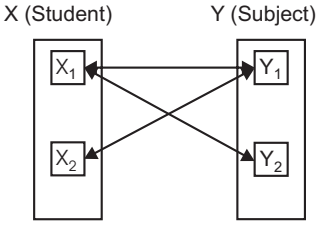


FIGURE 2.8. Many to many cardinality ratio.

A student can have more than one subject and one subject can be given to more than one student.

2.5.2 Participation Constraints

The participation constraints are discussed in section 2.9.6.

2.6 KEYS

A key is an attribute or set of attributes that is used to identify data in entity sets. The attributes which are used as key are known as **key attributes**. Rest of all are known as **Non-key attributes**.

2.6.1 Types of Keys

There are many keys that are used in the different tables. These are as follows:

1. Super Key : A super key is a set of collection of one or more than one attributes that can identify data uniquely.

Any entity set has more than one super key.

Employee

Reg. No	ID	Name	Salary	Dept-ID
S1D	1	Mohan	1500	10
A25	2	Sohan	2000	30
33Z	3	Vikas	3000	20
Z4X	4	Madhu	1000	10
A5C	5	Sonal	5000	20

(a)

Department

Dept-ID	Dept-Name
10	Sales
20	Marketing
30	Development

(b)

FIGURE 2.9. Entity sets employee and department.

Ex. In entity set Employee, shown in Figure 2.9(*a*), Super Keys are

(*a*) (ID, Name, Salary, Reg. No.)

(*b*) (ID, Name, Reg. No.)

(*c*) (ID) etc.

All combinations can identify data uniquely.

2. Candidate Key : The minimal super key is known as candidate key. Consider a super key and then take all of its proper subsets. If no one of the proper subsets are super key. Then this super key is taken as candidate key.

Ex. ID and Reg. No. are candidate key

Example: Find all possible candidate keys for the following relation based on its current tuples:

A	*B*	*C*	*D*
*a*1	*b*1	*c*1	*d*1
*a*1	*b*2	*c*2	*d*1
*a*2	*b*2	*c*1	*d*1
*a*2	*b*1	*c*2	*d*1

Ans. There are three candidate keys of this relation
{*A*, *B*}, {*A*, *C*}, {*B*, *C*}

Example: Given a relation STUDENTS as follows:
STUDENTS (SSN, Name, Home_Address, Birthdate, GPA).
(*a*) Determine some candidate keys of this relation?
(*b*) Determine a super key that is not a candidate key?

Ans. (*a*) {SSN}, {Name, Home_Address, Birthdate} are candidate keys.
(*b*) {SSN, Name} is a super key but not a candidate key.

3. Primary Key : An attribute which identifies data uniquely is known as Primary Key.

OR

The term Primary Key is used to denote Candidate key.

Any entity set can have more than one **Candidate key** but only one **Primary Key**.

Ex. In entity set Employee, either Reg. No. is primary key or ID is primary key.

4. Alternate Keys : All the candidate keys other than Primary Key are known as Alternate Keys.

Ex. If you take ID as Primary Key. Then, Reg. No. is an alternate key.

5. Secondary Key : An attribute or set of attributes which doesn't identify data uniquely but identifies a group of data is known as secondary key.

Ex. Name, Salary and Department No. are all secondary keys.

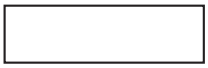
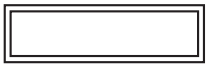
6. Foreign Key : A foreign key is an attribute in any entity set which is also a Primary Key in any other entity set.

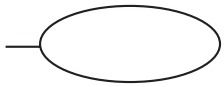
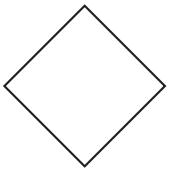
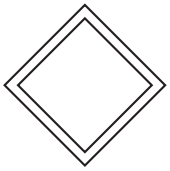
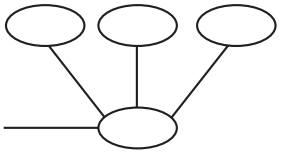
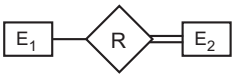
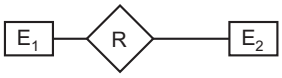
Ex. Dept_ID: This is an attribute in entity set Employee and also a primary key in entity set Department. Thus, it is a foreign key in Employee.

2.7 ENTITY—RELATIONSHIP DIAGRAM

E-R diagrams represents the logical structure of a database. Symbols used in E-R diagrams are shown in Table 2.1.

Table 2.1. Symbols in E-R diagram

S.No.	Name of Symbol	Symbol	Meaning
1.	Rectangle		Entity Set (Strong)
2.	Double Rectangle		Entity Set (Weak)

3.	Ellipse		Attribute
4.	Diamond		Relationship Set
5.	Double Diamond		Identifying Relationship Type
6.	Double Ellipses		Multi-valued attributes
7.	Dashed Ellipses		Derived attributes
8.	Ellipse with line inside it		Key attribute
9.	Ellipse joined with other ellipses		Composite attributes
10.	Double lines		Total Participation
11.	Single line		Partial Participation
12.	Triangle		Specialization or Generalization

Examples :

1. Make an E-R diagram having two entity sets, Customer and Item.

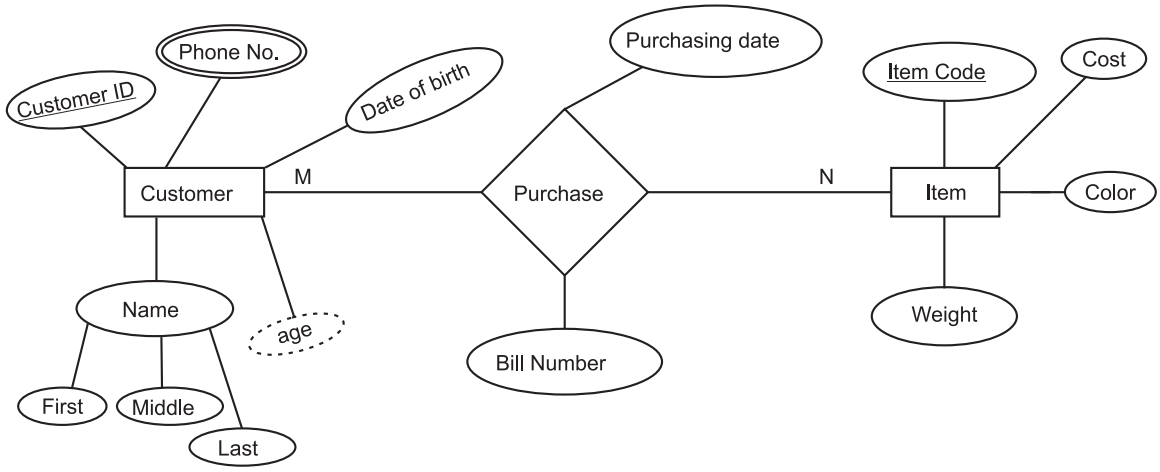


FIGURE 2.10. E-R diagram with customer and item entity sets.

Cardinality Ratio is many to many because a customer can buy any number of items and same item can be purchased by more that one customer.

2. Make an E-R diagram with entities Customer, Loan and Payment

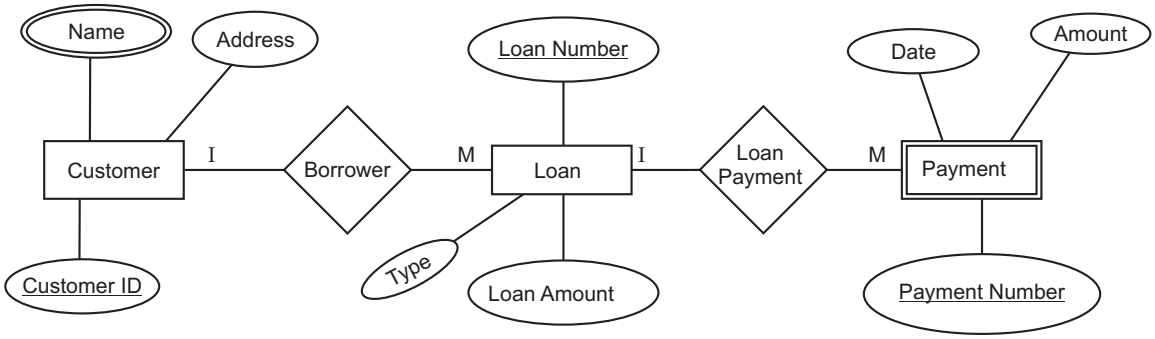


FIGURE 2.11. E-R diagram with customer, loan and payment sets.

2.7.1 Advantages of E-R model

The major advantages of E-R model are as follows:

1. **Straightforward relation representation:** The relation representation of the database model using E-R diagram are relatively more straightforward than other models.
2. **Mapping with relational model:** It can be easily mapped onto the relational model. The E-R diagrams used in the E-R model can easily be transformed into relational tables. The entities and attributes of E-R model can easily be transformed into relations (tables) and columns (fields) in a relational model.
3. **Communication tool:** It is very simple and easy to understand with a minimum of training efforts required. Therefore, the model can be used by the database designer to communicate the design to the end user.

4. **Design tool:** E-R model can also be used as a design plan by the database developer to implement a data model in specific database management software.
5. **Easy conversion to other models:** E-R diagrams can be easily converted to a network or hierarchical data model.
6. **Graphical representation:** E-R model provides graphical and diagrammatical representation of various entities, their attributes and relationships between entities.
7. **Easy to modify:** Modifications to E-R diagram at later stage is relatively easier than in other models.

2.7.2 Limitation of E-R Model

Limitation of E-R Model : E-R model cannot express relationships between relationships. In other words E-R model is not capable to express relationship set between relationship sets.

This limitation can be overcome by using *EER model*.

2.8 TYPES OF ENTITY SETS

There are two types of entity sets as given below:

2.8.1 Strong Entity Sets

Entity set having any key attributes are known as Strong Entity sets.

Ex. The Figure 2.12 shows strong entity set *Student* having key attribute Reg_No.

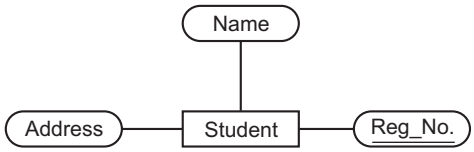


FIGURE 2.12. Strong entity set.

2.8.2 Weak Entity Sets

Entity sets having no key attributes are known as Weak Entity sets.

Ex. The Figure 2.13 shows weak entity set *Table* having no key attributes to identify the tuples uniquely.

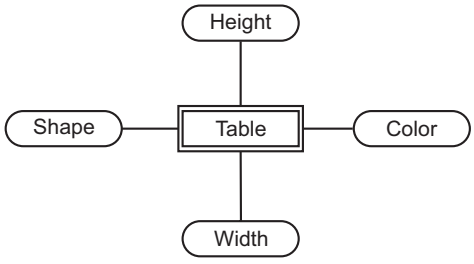


FIGURE 2.13. Weak entity set.

A weak entity set is always associated with another entity set to make it meaningful. This another entity set is known as **Identifying** entity set.

2.9 ENHANCED ENTITY-RELATIONSHIP (EER) MODEL

EER model is basically the enhanced version of E-R model which includes all the basic concepts of E-R model with capability to support additional semantic concepts of complex applications. These additional concepts are :

- Specialization
- Generalization
- Categorization.

Before discussing the concepts of specialization, generalization, and categorization two another entity types **superclass** (supertype) and **subclass** (subtype) are described.

2.9.1 Superclass and Subclass Entity Types

The most important new modeling construct introduced by EER was **superclass** and **subclass** entity types. These are also known as supertype and subtype entities respectively. By using these two entity types E-R model can be divided into more specialized sub-models or can join some sub-models to make a generalized E-R model.

- **Superclass Entity Type (Supertype) :** A superclass entity type is a generic entity type that includes one or more distinct subclasses that require to be represented in a data model. It means members belong to subclass are same as the entity in the superclass. The relationship between a superclass and subclass is a one-to-one (1 : 1) relationship. In some cases, a superclass can have overlapping subclasses.
- **Subclass Entity Type (Subtype) :** A subclass entity type is a more specialized entity type that has a distinct role in the organisation. A subclass is a member of superclass. It is one of the data-modeling abstractions used in EER. A subclass may be further divided and in that case it acts as superclass for its subclasses.

The superclass/subclass relationship is shown in Figure 2.14.

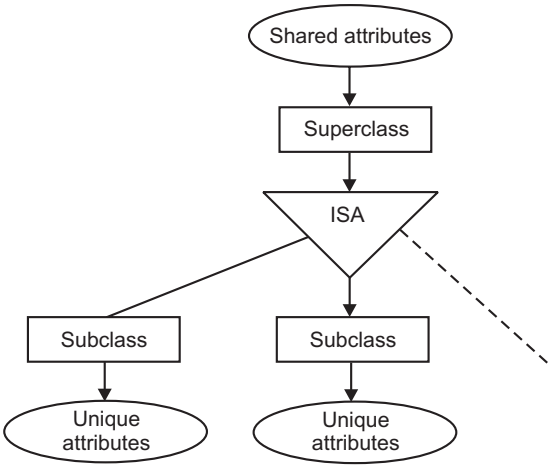


FIGURE 2.14. Superclass/subclass relationship.

Consider the example of a Bank as shown in Figure 2.15 in which PERSON entity is superclass entity type which is further divided into EMPLOYEE and CUSTOMER entities. Here EMPLOYEE and CUSTOMER entities are subclass entity type.

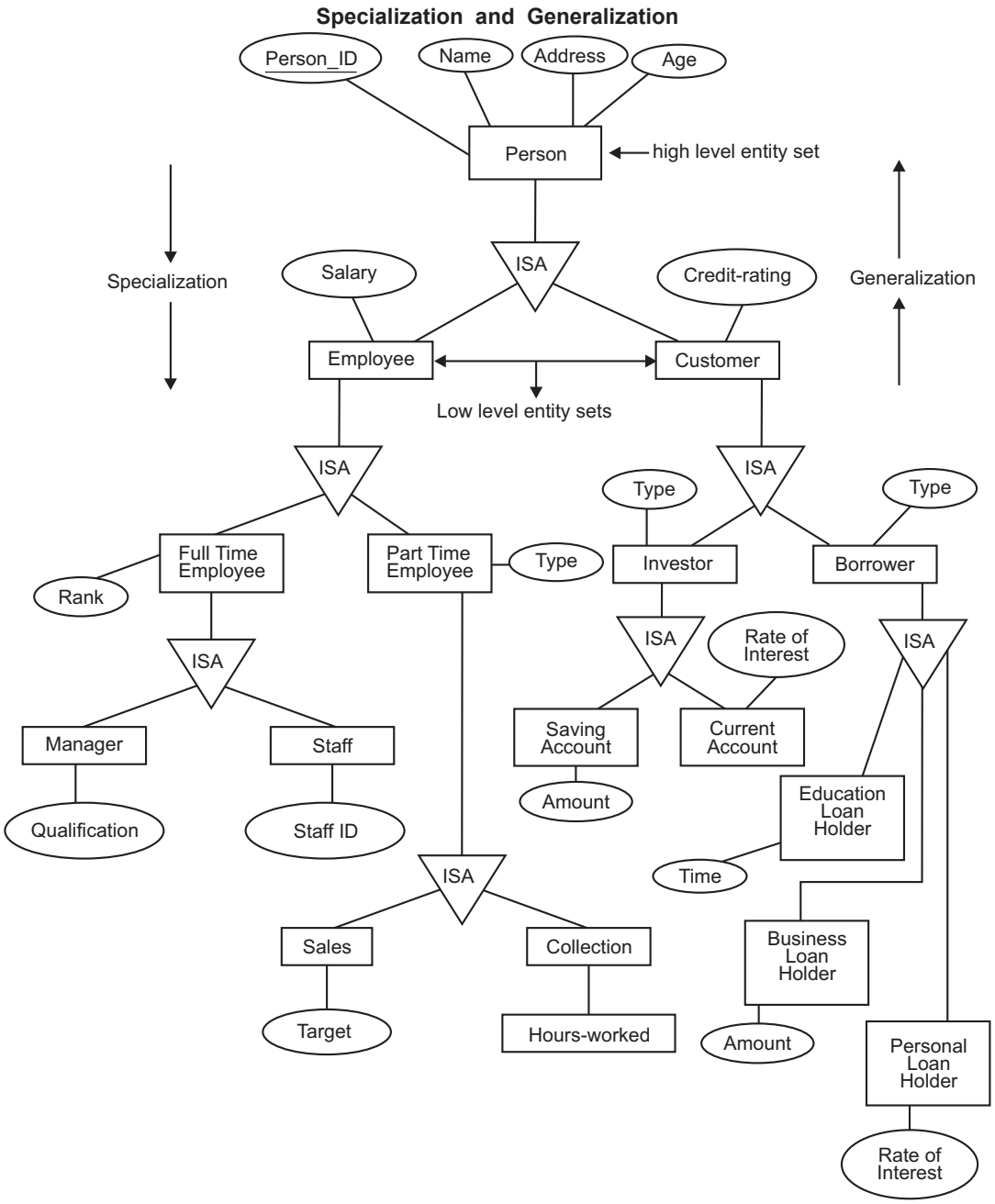


FIGURE 2.15. Specialization and generalization.

Basic concept is that by E-R model, a person belongs to a Bank is known, and by EER, how it belongs to Bank is known means, what is the exact relationship between Bank and

that person? A person may be an employee or customer which can be further categorized into Manager, Staff or Investor, Borrower and so on.

2.9.2 Specialization

Specialization includes subgrouping of entities within an entity set having some distinct nature then other entities. If deep information is needed then go towards specialization. In other words Specialization is a process by which any existing entity set is divided into smaller entity sets according to the distinct or different nature of entities.

Consider the example of Bank in Figure 2.15. Person is an entity set of all people who belongs to bank. Further Person is classified into Employees and Customers of bank. So, Person entity set is divided into Employee entity set and Customer entity set. Employees are further classified into two categories **full time** employees and **part time** employees and so on. Customers are also classified into **Investors** and **Borrowers** and so on.

2.9.3 Generalization

Generalization is a process by which two or more entity sets can be combined into a single entity set by determining similarities between the entities. Its an abstract view of any Enterprise. Generalization proceeds from the recognization that a number of entity sets share some common features. If an abstract view of information is needed then go towards generalization.

Consider the example in Figure 2.15. Here Investor and Borrower are two entity sets. They have common feature that both are Customer of the Bank. Similarly, Employee entity set and Customer entity set can be combined into Person entity set.

2.9.4 Attribute Inheritance

Specialization and generalization leads to attribute inheritance between higher level entity set and lower level entity set. Inheritance is a process by which lower level entity set inherits (or taken) some properties of its higher level entity set.

Consider the Figure 2.15. Here entity sets Employee and Customer inherits attributes Person_ID, Name, Address, Age from Person entity set.

2.9.5 Aggregation

Aggregation is an abstraction process in which a relationship set is considered as higher level entity set.

Consider an example of ternary relationship having three entity sets Employee, Job and Branch with relationship set works-on as shown in Figure 2.16. The information about Managers on employes, managers of particular jobs and of different branches can be taken easily.

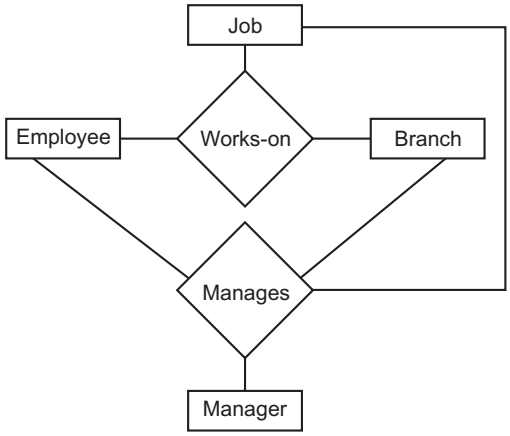


FIGURE 2.16. E-R model.

2.9.6 Specialization and Generalization Constraints

The following constraints are applied on specialization and generalization to capture important business rules of the relationships in an enterprise. There are **Two** types of constraints :

1. Participation Constraints : It tells the participation of entity set in relationship sets. There are two types of participations.

- *Partial participation :* If only some entities from entity set E is participated in relationships in set R then it is known as **Partial participation**. Partial participation is shown in Figure 2.17(*a*).
- *Total participation :* If every entity from entity set E is participated with at least one relation in relationship set R then it is known as **Total participation**. Consider the Figure 2.17(*b*).

Here Customer and Loan are two entity sets and Relationship set is Borrower.

— Every customer may or may not take the Loan so Customer entity set is partially participated.

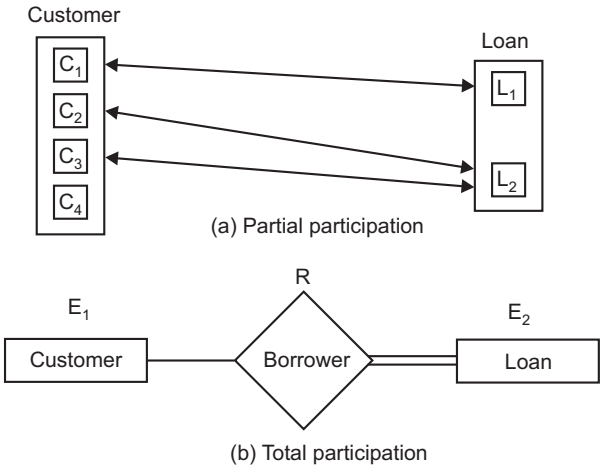


FIGURE 2.17. E-R model.

— But every loan is concerned with at least one customer of bank. So Loan entity set is totally participated.

2. Disjoint Constraints : Disjoint constraints describe the relationship between members of different subclasses. According to Disjoint constraint if the subclasses of a specialization/ generalization are disjoint then an entity can be a member of only one subclass of that specialization/generalization. Consider Figure 2.15, subclasses Full Time Employee and Part Time Employee of superclass Employee (discussed earlier that a subclass may be further categorized) are disjoint. Suppose any employee 'Martin' works as part time employee for Bank then it can only belongs to subclass 'Part Time Employee'.

2.9.7 Categorization

Categorization is a modeling process of a single subclass having relationship with more than distinct superclasses. The subclass having more than one superclass is known as category and the process of defining a category is known as categorization. The symbol shown in Figure 2.18(*a*) represents categorization. Consider Figure 2.18(*b*).

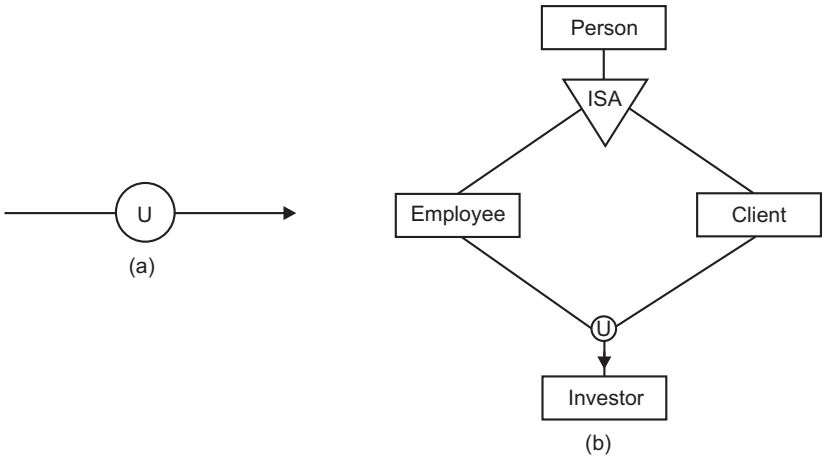


FIGURE 2.18. Categorization.

In a bank, a person can be either a employee or a client and both of them may be investors. So, here subclasses employee and client act as Disjoint Superclasses and Subclass Investor acts as Category.

You cannot combine works-on and managers relationship sets because some workers are not managers. Using aggregation, works-on relationship set acts as higher entity set and solve this drawback of E-R Model. E-R Model with Aggregation is shown in Figure 2.19.

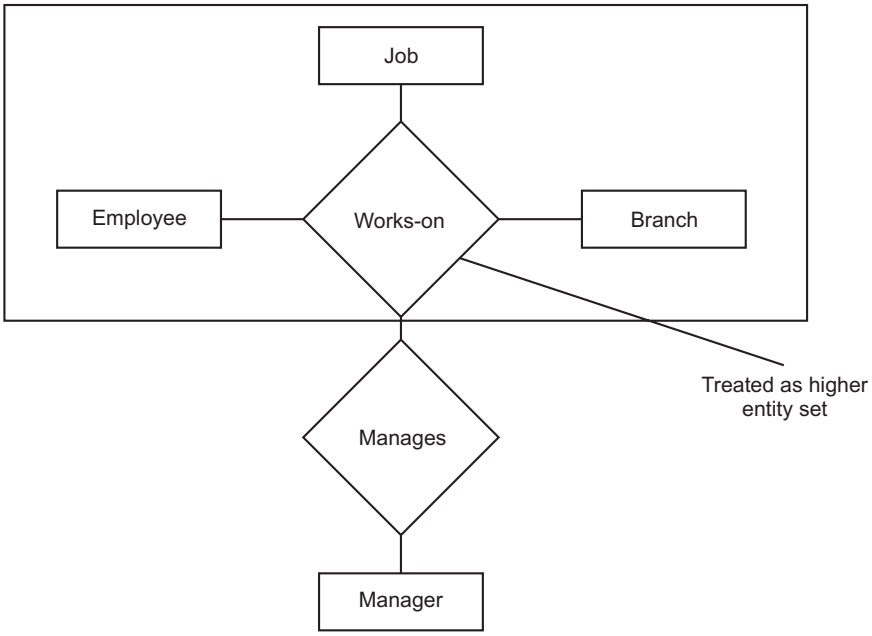


FIGURE 2.19. E-R model with aggregation.

2.10 REDUCTION OF AN E-R AND EER DIAGRAM INTO TABLES

To represent the database in tabular form, E-R diagrams have to be reduced in tables.

For each entity set, make different table and for each relationship set make a different table.

1. Reduction of Strong Entity Sets into Tables

For a strong entity set E with attributes a_1, a_2, ..., a_n, make a table having same name as of entity set E and having n number of columns or table name is equal to entity set name and number of columns is equal to number of attributes. Consider the Figure 2.20 having strong entity set Department with two attributes Dept-ID and Dept-name.

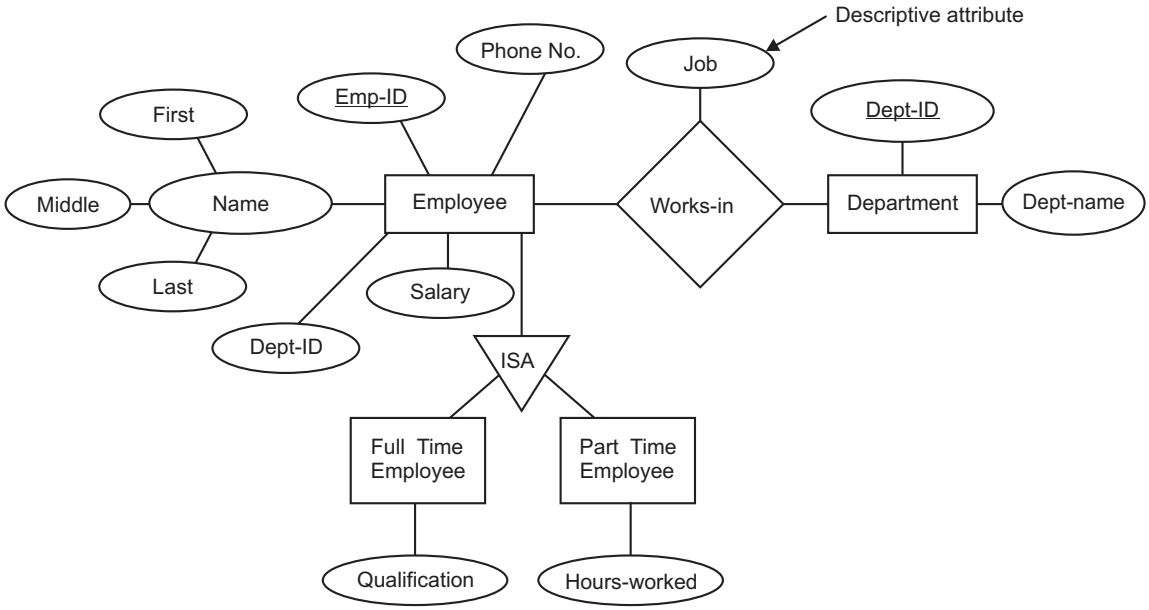


FIGURE 2.20. E-R model of employee and department entity sets.

The corresponding table is shown in Figure 2.21 with table name Department and two columns, Dept_ID and Dept_name.

Department

Dept-ID	Dept-name
10	Sales
20	Development
30	Testing
40	Accounts

FIGURE 2.21. The department table (Reduction of strong entity set).

2. Reduction of Composite Attributes

For a composite attribute, create a separate column for each component attribute or parts of composite attributes. Consider the example shown in Figure 2.20. The Name is a composite attribute with three component attributes First, Middle and Last. So, make three columns First-name, Middle-name and Last-name. The corresponding table is shown in Figure 2.22.

EID-ID	First-name	Middle-name	Last-name	Salary	Dept-ID
A12	Deepak	Kumar	Goyal	15,000	10
S50	Shivi	—	Goyal	75,000	20
51C	Anu	—	Parmar	8,000	10
67B	Ravi	—	—	5,000	40

FIGURE 2.22. The employee table (Reduction of composite attributes).

3. Reduction of Multi-valued Attributes

For multi-valued attributes, make a separate table with columns C1 which represent the primary key of entity set or relationship set and with columns C2 which represent the multi-valued attributes. Rows are equal to total number of values of that attribute. Consider Figure 2.20 in which Phone-No. is multi-valued attribute. So, make a table with two columns, one is Emp-ID (primary key of Employee) and second is Phone-No. (multi-valued attribute). Give any name to that table. The table is shown in Figure 2.23. If any employee has two phone numbers then it is possible to make two different entries in table and so on.

Emp-ID	Phone-No.
A-12	23896
A-12	23897
51-C	38976
51-C	23551
51-C	98941
67-B	23999

FIGURE 2.23. The phone-number table (Reduction of multi-valued attributes).

4. Reduction of Weak Entity Sets

Let A be the weak entity set and B be the strong entity set on which A depends. Then, it is possible to make a table with table name as of Weak Entity Set having columns equal to the

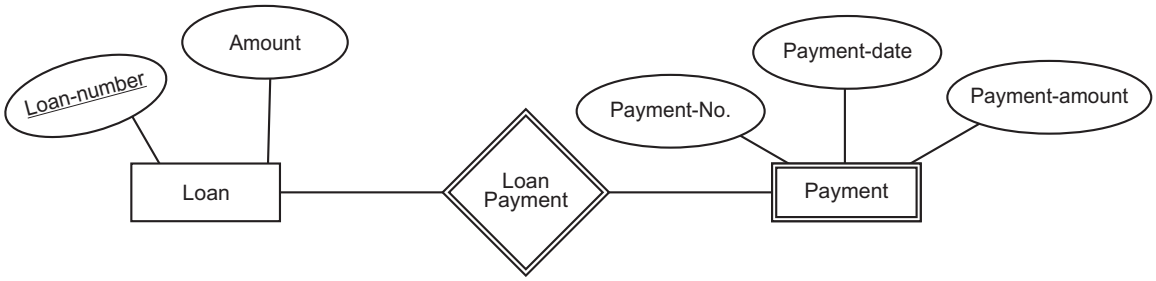


FIGURE 2.24. E-R diagram of weak entity set payment.

attributes of Weak Entity set plus Primary Key of the Strong Entity set on which Weak Entity Set depends. Consider the E-R diagram shown in Figure 2.24, in which Payment is a Weak entity set that depends upon Loan entity set. So, make a table with table name Payment having four columns as shown in Figure 2.25.

Payment

Loan-number	Payment-No.	Payment-date	Payment-amount
E–12	2	19-2-2004	6912
C–55	5	31-1-2005	5000
H–96	11	2-2-2005	2000
P-77	2	6-9-2005	2500

FIGURE 2.25. The payment table (Reduction of weak entity set).

5. Reduction of Relationship Sets

Let R be the relationship set and E1, E2, ..., EN be the entity sets participating in R. Make a table with table name as of Relationship Set having columns equal to number of attributes in relationship set (descriptive attributes) and primary keys of all participating entity sets.

Consider the ER diagram shown in Figure 2.20, having relationship set works-in having two participating entity sets, Employee and Department. The corresponding table is shown in Figure 2.26.

Works-in

Emp-ID	Dept-ID	Job
S-50	20	Engineer
A-12	10	Salesman
51-C	10	Salesman
67-B	40	Accountant

FIGURE 2.26. The works-in table (Reduction of relationship sets).

(*i*) Redundant Tables

The relationship set between weak and strong entity sets are treated specially. Consider the E-R diagram shown in Figure 2.24, where weak entity set, Payment depends on strong entity set Loan having relationship set loan-payment. Primary key of Entity set Loan is [loan-number] and of Weak entity set is [loan-number, payment-number]. Table of entity set Payment has four attributes [loan-number, payment-number, Payment-date, payment-amount]. If you make table of relationship set loan-payment then it contains attributes [loan-number, payment-number]. This combination is already present in table of Payment. Even, there are no descriptive attributes. So, this table is redundant and discard it.

(*ii*) Combination of Tables

Consider two entity sets X and Y connecting with relationship set XY. The *n*, three tables named X, Y and XY have to be made. If cardinality ratio between X and Y is many-to-many and X is totally participated then, combine tables X and XY. Consider the E-R diagram shown in Figure 2.27, having two entity sets, Customer and Loan. The relationship is many-to-many because a customer can take many loans and a single loan can be taken by more than one customer or joint loan. Loan entity set is totally participated because every loan refers to some customer. So, combine tables Loan and Borrower. But loan cannot exist with any customer so two tables are needed *i.e.*,

- Loan [loan-number, amount, customer-ID, Income]
- Customer [Customer-ID, Name]

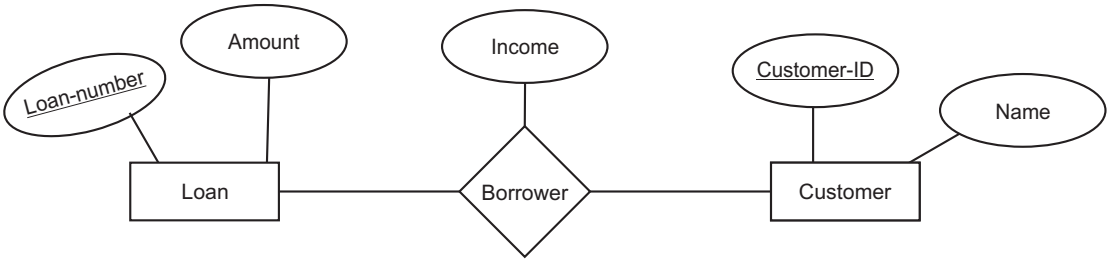


FIGURE 2.27. Combination of tables.

6. Reduction of Generalization

In generalizations, higher level entity sets and lower level entity sets are considered. Make a table for higher level entity set with all its attributes. For lower level entity set, make a table with all its attributes with primary key attributes of its higher level entity set. Consider E-R diagram shown in Figure 2.20, in which Employee is high level entity set and Full Time Employee and Part Time Employee are two lower level entity sets. So, make three tables as given below:

- Employee [Emp-ID, Dept-ID, First-Name, Middle-Name, Last-Name, Salary]
- Full Time Employee [Emp-ID, Qualification]
- Part Time Employee [Emp-ID, Hours-Worked]

7. Reduction of Aggregation

Reduction of aggregation into tables is simple. Consider the E-R diagram shown in Figure 2.19. For all entity sets, make tables as discussed earlier. For making tables for relationship sets, consider the same approach as discussed earlier. Take an example of relationship set Manages. Make a table manages with all descriptive attributes, primary key of entity set Manager and the relationship set works-on.

SOLVED PROBLEMS

Problem 1. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient, a log of various tests and examinations conducted. Construct the appropriate tables for this E-R diagram and list the tables with their attributes, primary key and foreign keys.

Solution. The E-R diagram is shown in Figure 2.28.

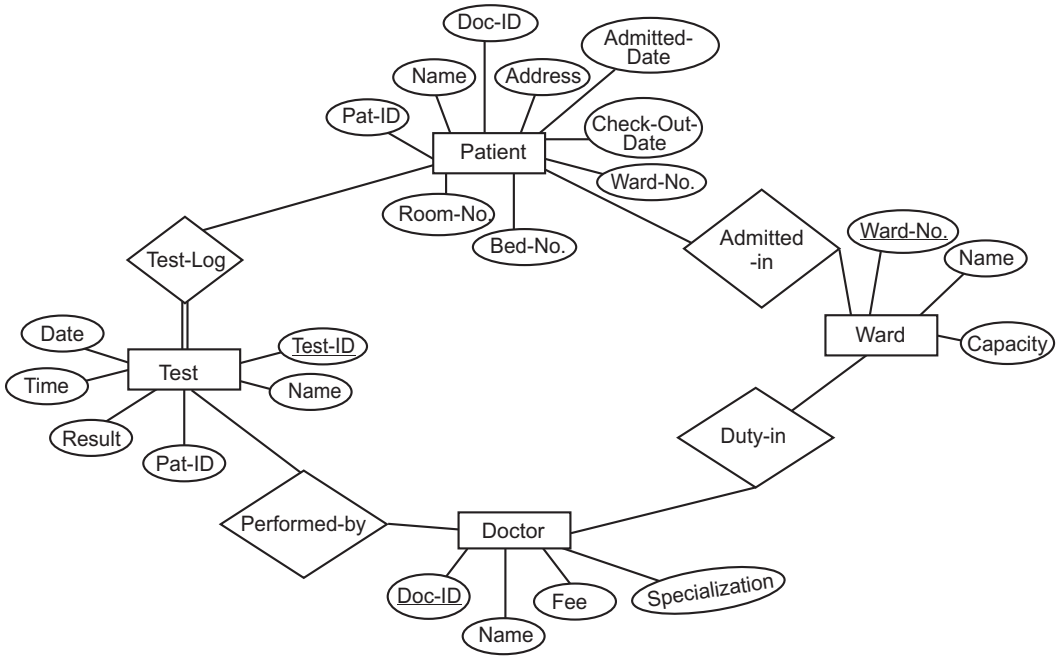


FIGURE 2.28. E-R diagram of hospital.

The Tables are as follows :

Patient (Pat-ID, name, address, admitted-date, check-out-date, room-no., bed-no., ward-no, doc-ID)

Ward (Ward-no., name, capacity)

Doctor (Doc-ID, name, fee, specialization)

Test (Test-ID, name, date, time, result, Pat-ID)

Primary key is shown by _____.

Foreign key is shown by _____.

Problem 2. The people's Bank offers five type of accounts : Loan, checking, premium savings, daily interest saving, and money market. It operates a number of branches and a client of bank can have any number of account. Accounts can be joint *i.e.,* more than one client may be able to operate a given accounts. Identify the entries of interest and show their attribute. What relationship exists among these entities? Draw the corresponding E-R diagram.

Solution. The E-R diagram is shown in Figure 2.29.

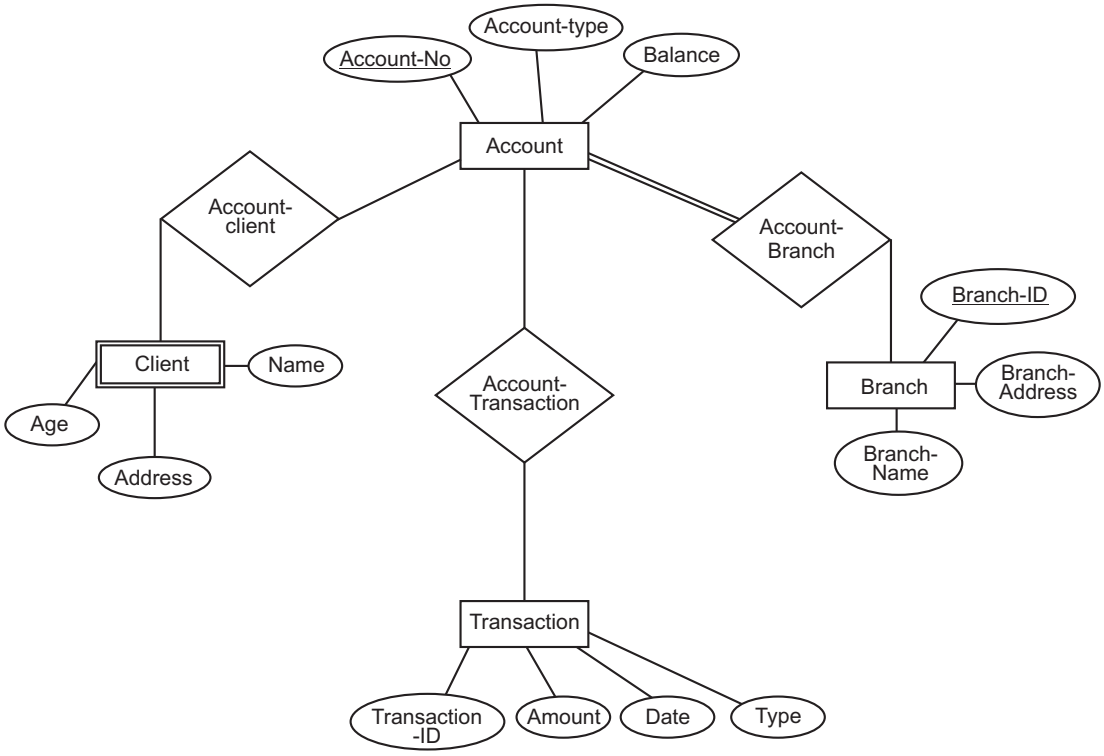


FIGURE 2.29. E-R diagram of bank.

The Entities are as follows :

Account (Account-no, Account-type, Balance)

Branch (Branch-ID, Branch-address, Branch-name)

Transaction (Transaction-ID, Amount, Date, Type)

Client (Account-No., Name, Age, Address)

Relationships are Account-Branch, Account-Transaction, Account-Client.

Problem 3. Draw an entity-Relationship diagram of a manufacturing company which records information about the projects it has on hand, the parts used in projects, the suppliers who supply the parts, the warehouses in which those parts are stored, the employees who work on these projects.

Solution. The E-R diagram is shown in Figure 2.30.

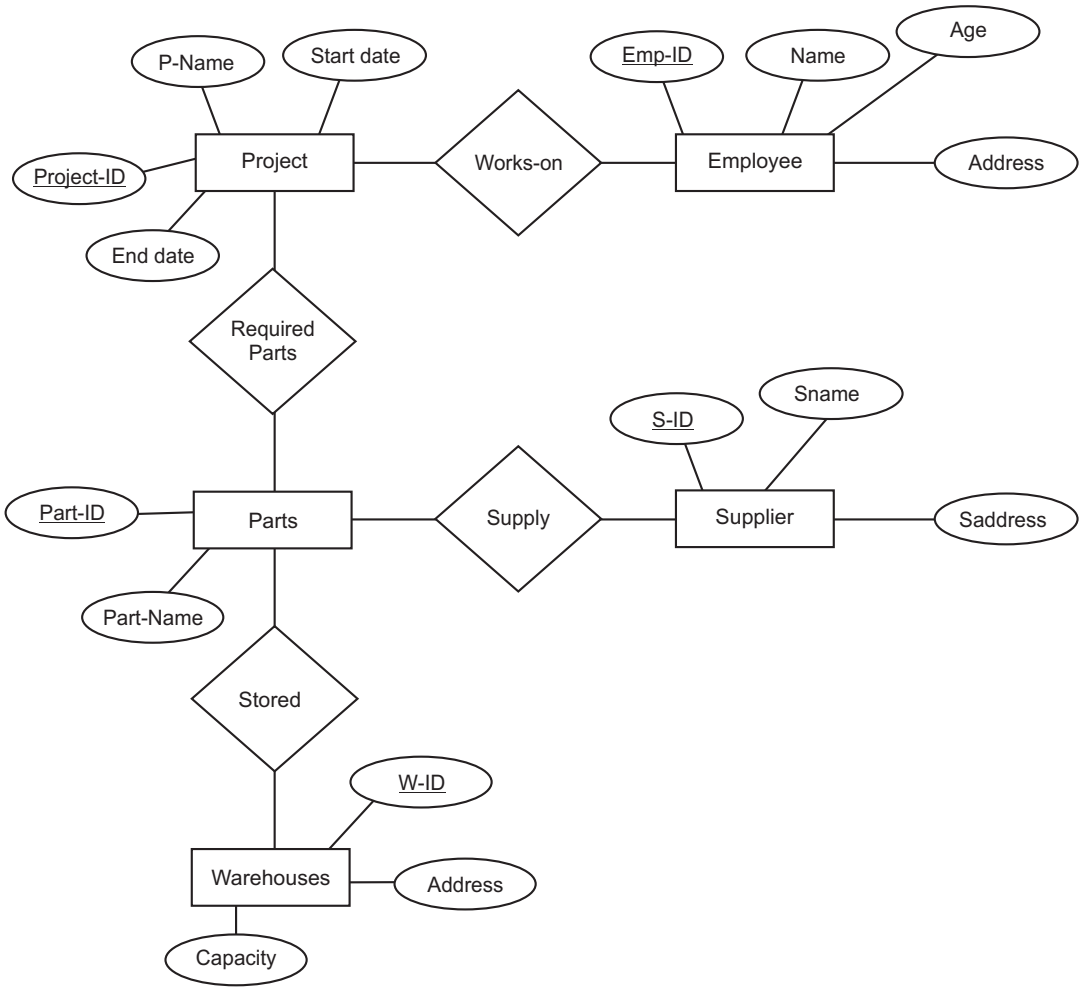


FIGURE 2.30. E-R diagram of manufacturing company.

Problem 4. A chemical or set of chemicals gives rise to another chemical or set of chemicals, when reacts under no condition or a set of conditions.

For example, Methane and Chlorine gives rise to Chloromethane when exposed to light. Here Methane and Chlorine are the reactants, which when under condition "Exposure to sunlight" giving Chloromethane as product.

Similarly, reaction of Water and Sodium gives Sodium Hydroxide and Hydrogen and no condition is required.

There are numerous reactions possible and each reactions has to be given a reaction number. Each chemical and condition has to be given a code.

Answer the following :

(*i*) Identify the entities in the above system.

(*ii*) Identify the attributes of the entities identified in (*i*)

(*iii*) Identify relations and their cardinalities

(*iv*) Draw E-R diagram for the above system.

Solution.

(*i*) Entities are Chemical, Condition and Reaction

(*ii*) Attributes of these entities are

Chemical (Chem-code, name, color, state)

Condition (Cond-Code, details)

Reaction (Reac-number, Reaction-Type)

(*iii*) Relations between these entities are

ON (between Chemical and Condition)

RESULTS (between Condition and Reaction)

PRODUCE (between Reaction and Chemical)

Many-to-Many cardinalities exist between all the above entities.

(*iv*) The E-R diagram is shown in Figure 2.31.

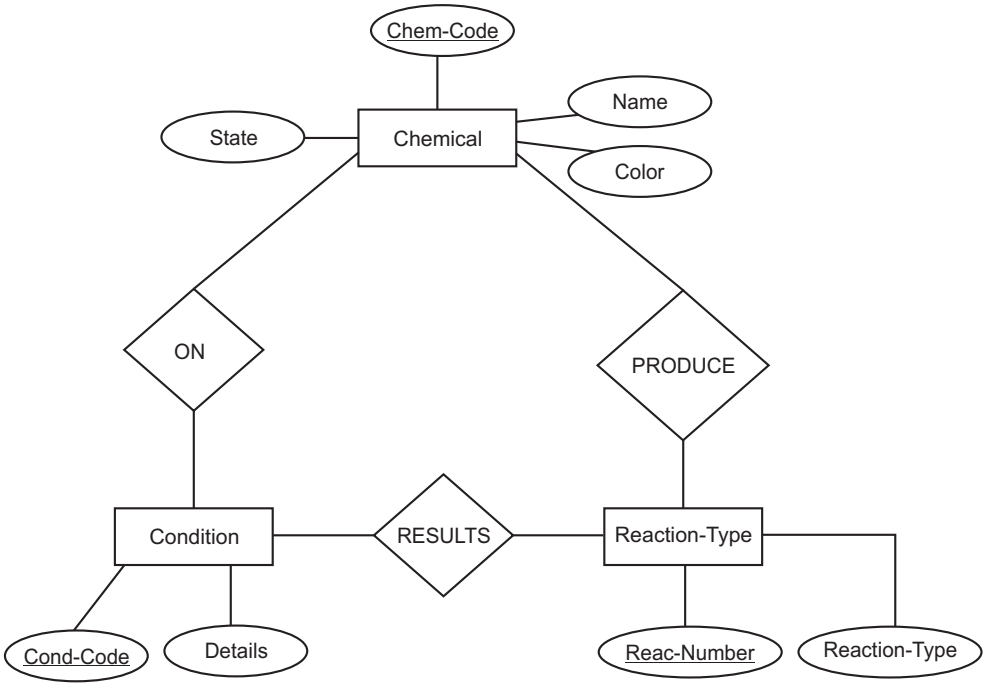


FIGURE 2.31. E-R diagram of chemical reaction.

Problem 5. In a manufacturing industry labourers are given different jobs on different days and each job has its own monthly basic and monthly DA rates as wages to be paid to labours. A labour is not given more than one type of job on a day. A database designer is given the job to design database for above situation and the designer designs one of the tables as :

Field	Type	Remarks
From date	Date	From this date to
To date	Date	This date
Labour Number	Number(6)	Labour Number
Job Done Code	Char(6)	Does the job Done Code
At Basic Rate	Number(10,2)	At the this Basic rate
At DA Rate	Number(10,2)	And this DA rate

Draw E-R diagram for above situation.

Solution. The E-R diagram is shown in Figure 2.32.

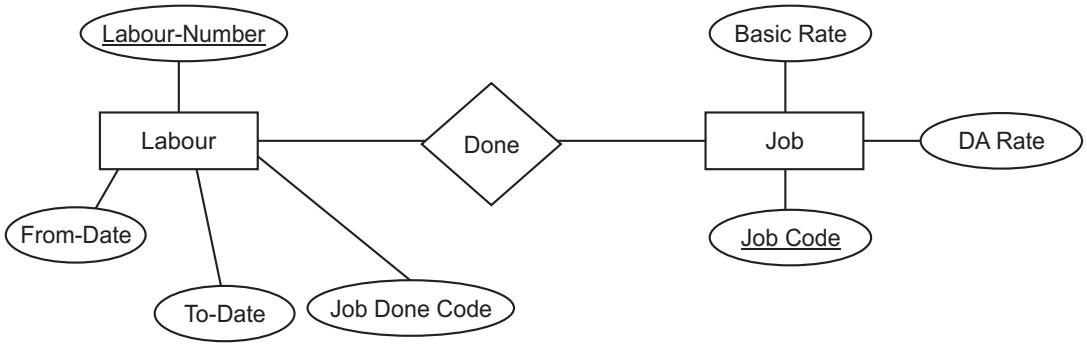


FIGURE 2.32. E-R diagram of manufacturing industry.

Problem 6. Translate the given E-R diagram to relational schema.

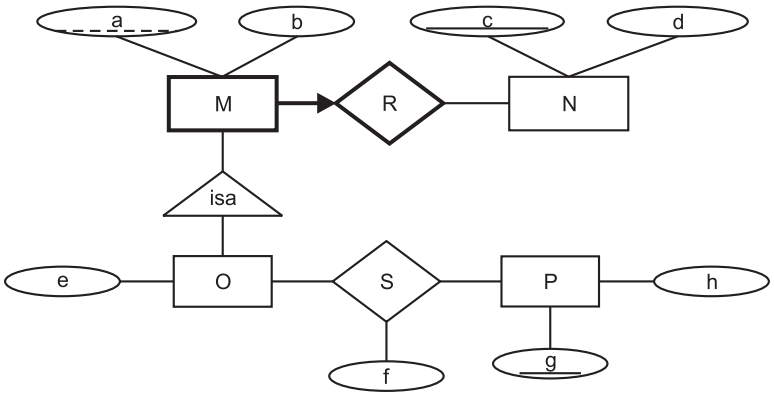


FIGURE 2.33

Solution.
- N is an entity, so we will create a table for it: $N(\underline{c}, d)$.
- P is an entity, so we will create a table for it: $P(h, \underline{g})$.

- Since M is a weak entity, we will create *one* table for it *and* R, which contains the key of N as a key: $M\_R(\underline{a}, b, \underline{c})$, where c is a foreign key of N. Because R is a weak entity, we must delete a M_R tuple if the corresponding N tuples disappears.
- Now we create a relation for O, which must include the key of M. The key of M includes the key of N since it is a weak entity, resulting in: $O(e, \underline{a, c})$, where a and c are a foreign key of M_R. Note that technically speaking c is really a foreign key of N, but since the requirements are that you must refer to the *entire* key of a table, we must have it refer to M_R's key, rather than N's.
- S is a many to many relationship, so we will create a table for it which includes the attributes of S and the keys of O and P, which together form the primary key of S: $S(f, \underline{a, c, g})$, where a and c are foreign key references to O, and g is a foreign key reference to P.

Problem 7. Consider the following E-R diagram:

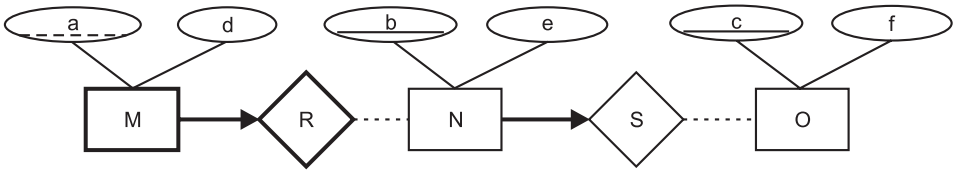


FIGURE 2.34

(*a*) This diagram presents two conflicting rules that apply to whether N should be represented by its own table in the relational schema. What are they?

(*b*) Which one would you use and why?

Solution. (*a*) The two rules are:

(*i*) M is a weak entity dependant on N

(*ii*) N is the many side of a many to one relationship (S – denoted by the arrow) so N and S should be represented by the same relation in the relational schema.

(*b*) Because M is a weak entity, we have no choice on how to model it; it must include the information about N's key. The choice is what do we do about NS. If we follow both rules, we have the relations:

(*i*) $NS(\underline{b}, e, c)$ – note that c is *not* needed as part of a key because we know which S relationship we are referring to based only on the many side of the relationship (N)

(*ii*) $MR(\underline{a, b}, d)$ – with a foreign key to NS

(*iii*) $O(\underline{c}, f)$

which would mean that the concept of MR depends now on NS, not just on N. On the one hand, one could argue that this isn't a problem. For one thing, it'd be worse if c were part of the key of NS, but it isn't. Besides, this makes for smaller numbers of tables, and less duplication. Since we have the fact that there is total participation for N in S (denoted by the thick line from N to S), there aren't going to be any null values. So combining them. On the other hand, we now have the fact that M depends on the relationship with S.

Problem 8. Convert the following ER – diagram into a relational database (the primary keys are underlined):

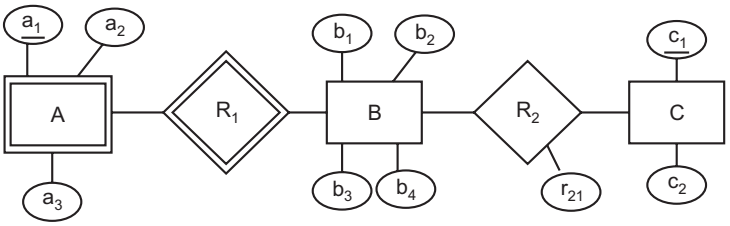


FIGURE 2.35

Solution. The relational database schema for the given *ER* diagram is as follows:

$A(\underline{a_1},\ \underline{b_1},\ a_2,\ a_3)$

$B(\underline{b_1},\ b_2,\ b_3,\ b_4)$

$C(\underline{b_1},\ c_2)$

$D(\underline{b_1},\ \underline{c_1},\ r_{21})$

Problem 9. Map the following ER diagram to a relational database. Give the relation names and attributes in them. Also mention the primary key and foreign keys if any for each table.

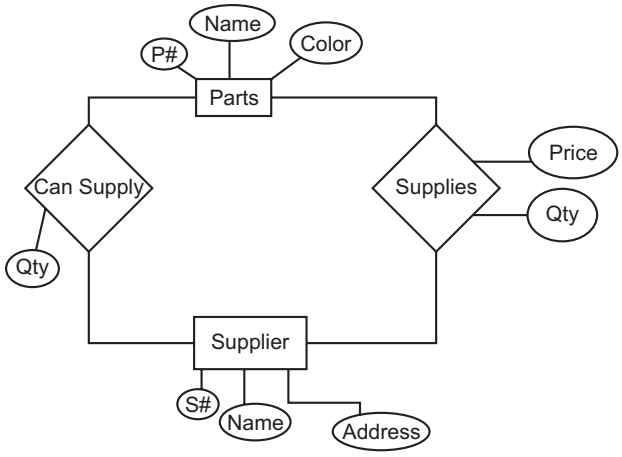


FIGURE 2.36

Solution. The following relations with attribute names are obtained from the given *ER* Diagram. The primary keys are underlined and Foreign keys are defined with each relation.

Parts(P#, Name, Color). There is no Foreign Key.

Supplier(S#, Name, Address). There is no Foreign Key.

Can_Supply(P#, S#, QTY). P# references Parts.P# and S# references Supplier.S#.

Supplies(P#, S#, Qty, Price). P# references Parts.P# and S# references Supplier.S#.

Problem 10. Consider the following ER diagram:

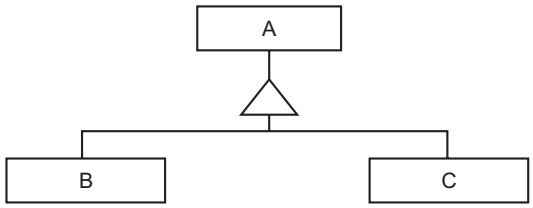


FIGURE 2.37

where *A*, *B* and *C* are entity sets.

1. Specify the condition(s) that is(are) necessary in order to represent all three sets with a single table.
2. Specify the condition(s) that is(are) necessary in order to represent all three sets with two tables, one for *B* and one for *C*.

Solution. 1. The ISA relationship must be disjoint.

B and *C* must have the same attributes.

2. The ISA relationship must be total.

Problem 11. Suppose we define a database about the customers of a bank and the loans they have received from the bank. For each customer we need to record information about their name, address, phone number and the company they work for. For each loan we need to record the amount, the interest rate, date the loan was issued, and the date the loan should be paid off.

(*i*) Is it a good idea to represent the company for which a customer works as an attribute of the customer or as a relationship? Briefly justify your answer.

(*ii*) Which is the best way to represent the relationship between the customer and their loans:

(*a*) by defining the loan as an attribute of the customer, or

(*b*) by making the loan a separate entity set and defining a relationship set between it and the customer?

Briefly justify your answer.

Solution. (*i*) The company should be an attribute of the customer, assuming each customer works for a single company. We don't need to keep any information for each company.

(*ii*) The loan should be a separate entity set associated with a customer through a relationship.

Reasons:

- A customer may have more than one loans.
- A loan has additional information on its own.

Problem 12. (*a*) Construct an E-R diagram for the following description.

Design a database for the reservation office of a bus company.

– Each bus has a unique number. We also store its class and capacity.

– Each place has a unique name and location information of the latitude and longitude.
– Routes have a starting place and an ending place; also, some of them have several intermediate places.
– A number of buses are scheduled to a route. A bus is assigned to one schedule; some buses can have multiple schedule. We store the date and starting time of each schedule.
– A member of our company can book a bus by specifying a schedule. We store a unique id, first name, and last name of each member.
– For each reservation, credit card number, the number of passengers, and the reservation datetime are stored.

(*b*) Convert your E-R diagram into a relational schema.

Solution. (*a*) The ER diagram is given below.

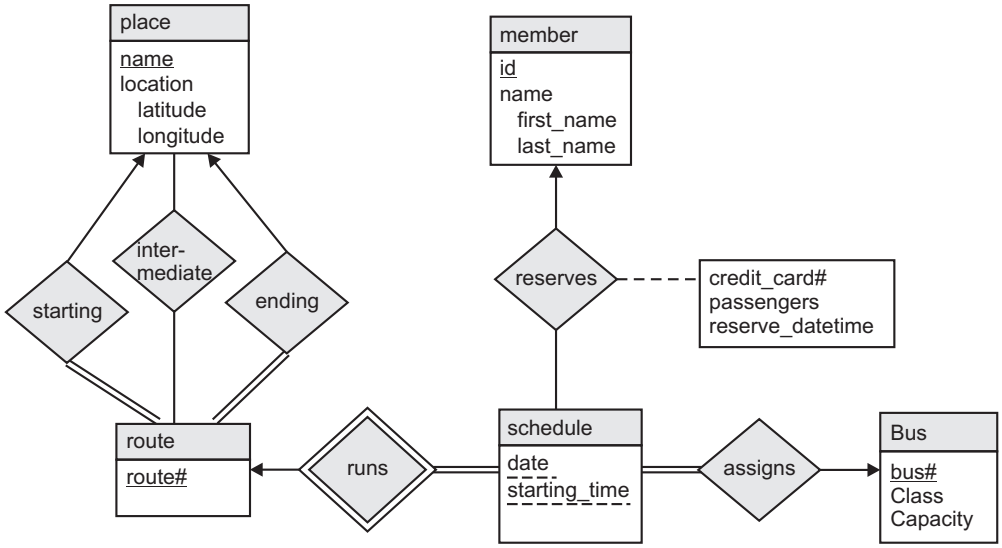


FIGURE 2.38

(*b*) The corresponding relational schema is as follows:

place(name, latitude, longitude)

route(route#)

starting_place(route#, name)

ending_place(route#, name)

intermediate_place(route#, name)

schedule(route#, day, starting_time)

bus(bus#, class,capacity)

assignment(route#, day, starting_time, bus#)

member(id, first_name, last_name)

reservation(id, route#, day, starting_time, credit_card#, passengers, reserve_datetime)

Problem 13. Consider the following Entity/Relationship diagram:

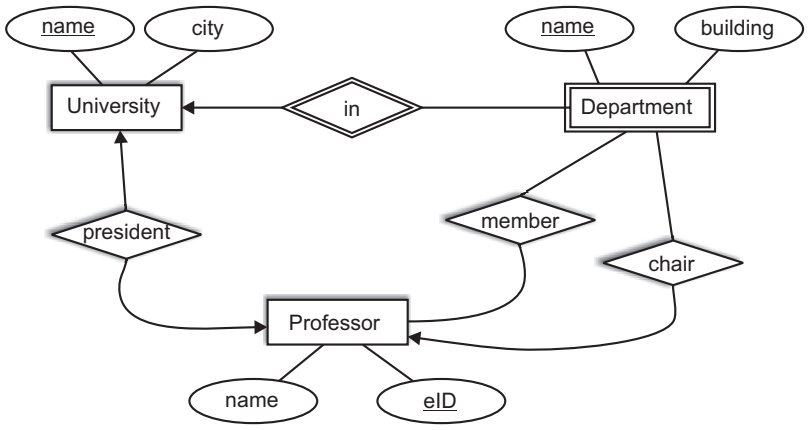


FIGURE 2.39

Which of the following statements are true according to this Entity/Relationship diagram?

1. Each department must be in exactly one university.
2. A university may have no departments.
3. No two departments can have the same name.
4. No two universities can have the same name.
5. A professor can be president of more than 1 university.
6. A university can have no president.
7. A department can have no chair.
8. A professor can be chair of more than one department.
9. There cannot be two universities in the same city.
10. Two departments with the same name must not be in two different universities.

Solution. 1. True 2. True 3. False 4. True 5. False 6. False 7. True 8. True 9. False 10. False

Problem 14. Consider the following ER diagram:

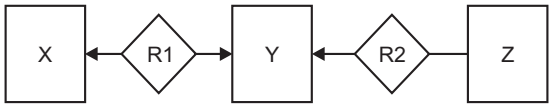


FIGURE 2.40

Which of the following cardinalities is valid for the entity sets? Do not guess. There is one point for each correct answer, –1 for each incorrect answer, and 0 points if you leave the answer blank.

1. $|X| = 0, |Y| = 0, |Z| = 0.$
2. $|X| = 0, |Y| = 0, |Z| = 8.$
3. $|X| = 5, |Y| = 5, |Z| = 0.$
4. $|X| = 3, |Y| = 3, |Z| = 6.$
5. $|X| = 2, |Y| = 0, |Z| = 0.$
6. $|X| = 0, |Y| = 5, |Z| = 5.$

Solution. 1. Valid 2. Invalid 3. Valid 4. Valid 5. Valid 6. Invalid

Problem 15. You have been tasked with designing a database for the Indian Census department to store all of their data, past and future. The database must conform to these constraints:

1. There has been a census every 10 years. The year of each census is unique to that census. There are also some notes as well as the total population of the India.
2. Each state has a unique name, and a value for its square area. Every state participates individually in every census, providing its population.
3. Every person has a unique SSN, as well as a name and birthday. Each person participates in every census by providing their age.
4. An address has a unique identifier, as well as a street name, city, state, and zipcode.
5. A person lives at only one address.

(*a*) Draw an ER diagram for this database. Be sure to mark the multiplicity of each relationship (1-1, 1-many, many-many, etc) of the diagram. Decide the key attributes and identify them on the diagram by underlining them. State all assumptions you make.

(*b*) Translate your ER diagram into a relational schema. Select approaches that yield the fewest number of relations; merge relations where appropriate. Specify the key of each relation in your schema. If the names of your foreign keys do not match the primary key name, please state the link in your assumptions.

Solution. (*a*)

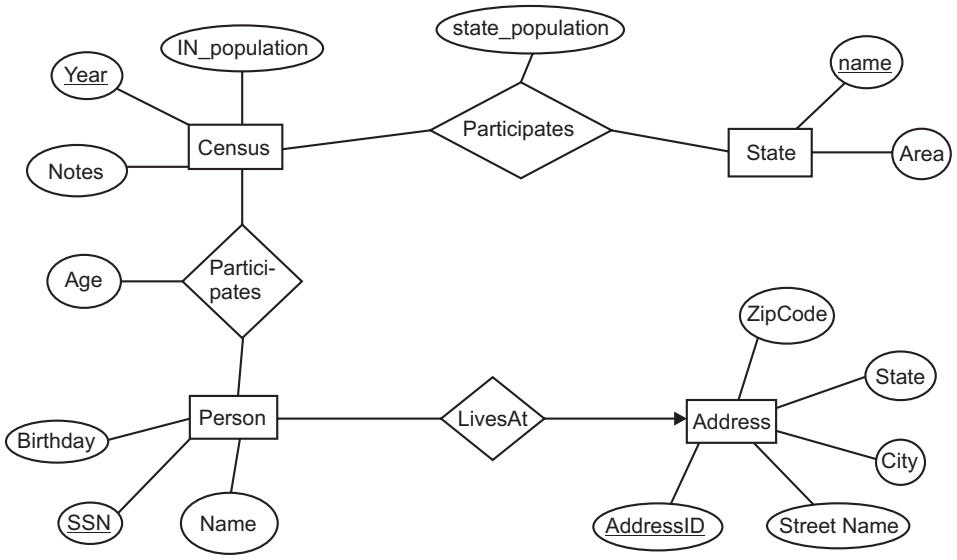


FIGURE 2.41

(*b*) Census(CensusYear, Notes, INPopulation);

State(StateName, Area);

StateParticipate(StateName, CensusYear, statepopulation);

Person(SSN, Name, Birthday, AddressID);

PersonParticipate(SSN, CensusYear, PersonAge);

Address(AddressID, Street, City, State, ZipCode).

Problem 16. Translate your Entity-Relationship Model (ER Diagram) into a logical model (DB Schema). For each relation in your schema, provide its name, attributes and keys (underlined attributes).Translate your Entity-Relationship Model (ER Diagram) from the question above into a logical model (DB Schema). For each relation in your schema, provide its name, attributes and keys (underlined attributes).

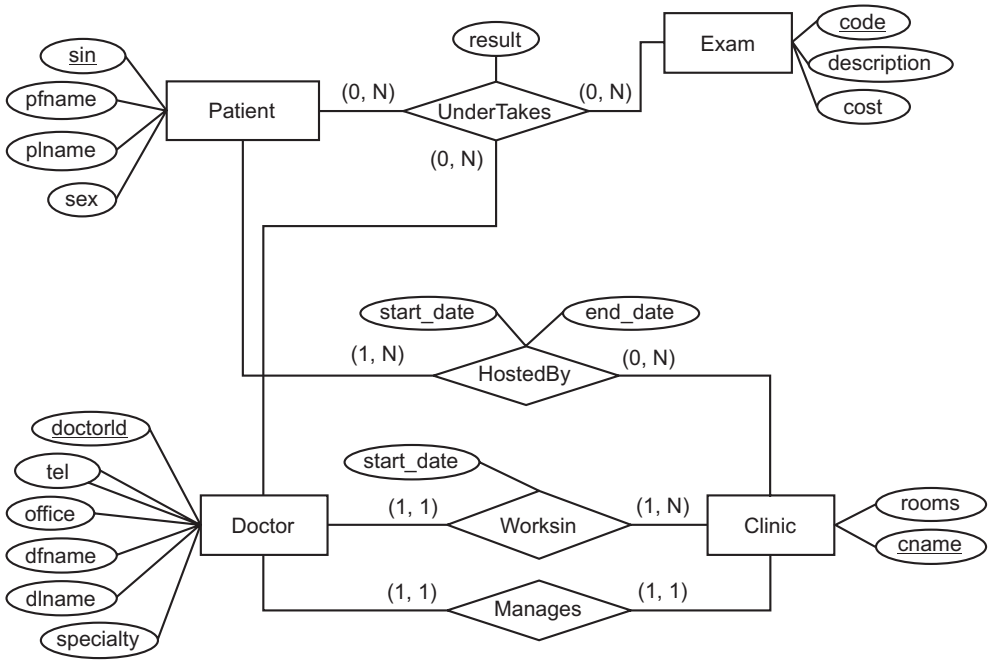


FIGURE 2.42

Solution.

Doctor

doctorId	dfname	dlname	office	specialty	cname	start_date

Tel

doctorId	tel

Patient

sin	pfname	plname	sex

Exam

code	description	cost

Clinic

cname	rooms	doctorId

Undertakes

sin	code	doctorId	result

Hosted By

sin	cname	start_date	end_date

Problem 17. Consider an application that needs to manage data for a travel agency. It needs to store the following entities and relationships:

- Hotels: have attributes name, address, price
- Resorts: are Hotels, that also have an attribute minimum-stay
- Activities: have attributes name, season
- Has: is a relationship between Resorts and Activities

Solution. Assumption: activities are uniquely identified their names (you could make other assumptions; it's O.K. as long as you stated them clearly).

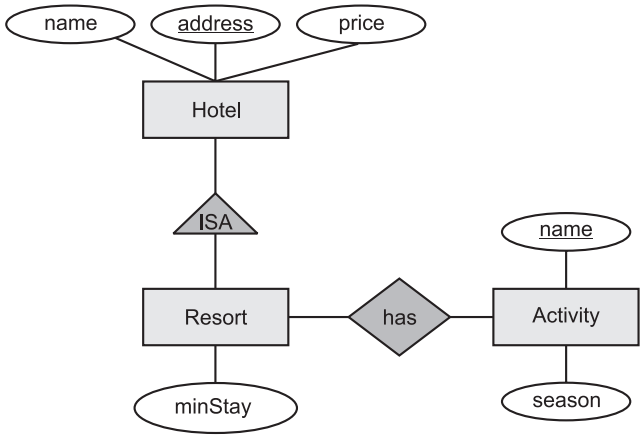


FIGURE 2.43

Problem 18. Design an E/R diagram for an application domain consisting of the following entity sets:

- Projects. Attributes: name, budget
- Teams. Attributes: team_name
- Empolyees. Attributes: name, phone_number
- Consultants. Attributes: name, phone_number, hourly_rate

And the following relationships:

- Each team works on one or more projects.
- Each project has an auditor, who is an employee
- Consultants are employees

Your answer should consist of an E/R diagram with entity sets, attributes (make sure you create appropriate keys: you may incorporate new attributes if needed), relationships, and inheritance.

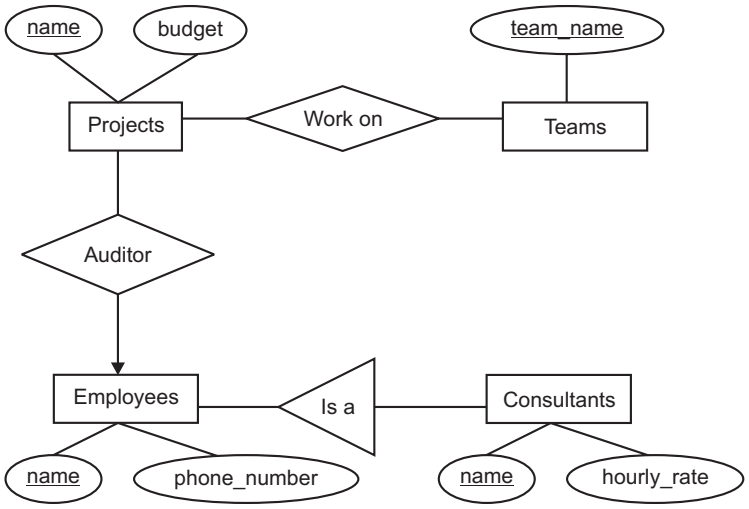


FIGURE 2.44

The E/R diagram shown below is for the following scenario: A publishing company produces academic books on various subjects. Authors who specialise in one or more particular subject write books. The company employs a number of editors who do not have particular specializations but who take sole responsibility for editing one or more publications. A publication covers a single subject area but may be written by one or more author – the contribution of each author is recorded as a percentage for the purposes of calculating royalties. Give a reason about which relation has the incorrect cardinality in the E/R diagram.

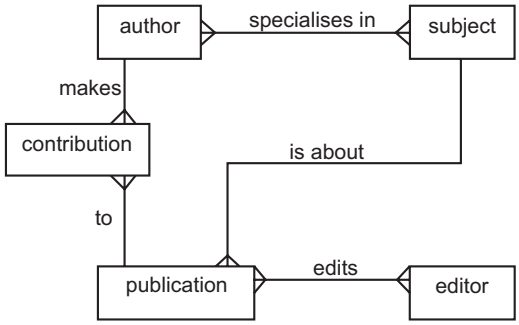


FIGURE 2.45

Solution. From the specification, "...[Editors] take sole responsibility for editing one or more publications...". Thus an editor can edit more than one publication (one to many), but each publication has only a single editor. Thus the relationship for "edits" should be one to many, not many to many.

TEST YOUR KNOWLEDGE

True/False

1. When transforming an E-R model into a relational database design, each entity is represented as a table.
2. The E-R model refers to a specific table row as an entity occurrence.
3. Attributes are types of entities.
4. A composite key is a primary key composed of more than one attribute.
5. All attributes are either simple or composite.
6. All simple attributes are also single-valued.
7. Composite attributes cannot be further subdivided.
8. A multivalued attribute can have lower and upper bounds.
9. An attribute value can be derived from another attribute.
10. The names on entity types and entity sets are different.
11. An entity cannot have more than one key attribute.
12. A relationship type of degree two is called as ternary relationship.
13. Relationship types can also have attributes.
14. The attribute of a relationship type can be added to participating entity types.
15. A weak entity type can have more than one identifying entity type.
16. The number of levels of weak entity types cannot be more than one.
17. The E-R model is high-level conceptual model.
18. Derived attributes are stored in a special database table.
19. Cardinality expresses the specific number of entity occurrences associated with every occurrence of a related entity.
20. All entity relationships can be characterized as weak or strong.
21. A weak entity has a primary key that is partially or totally derived from the parent entity in the relationship.
22. The underlined attribute in E-R diagram represents a primary key.
23. A domain is a set of composite values.
24. A domain need not be given a format.
25. It is possible for several attributes to have same domain.
26. An E-R diagram with m entities and n relationships will translate to $m+n$ tables.
27. If E is a weak entity set, then its key can only be the key attributes of E's supporting entity sets.

Fill in the Blanks

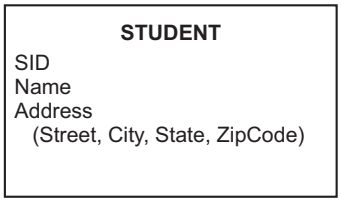
1. Attributes that are not divisible are called _____.
2. When the value of an attribute A is obtained from the value of an attribute B, then the attribute A is called _____.
3. _____ are characteristics of entities.
4. _____ specifies the set of values that can be assigned to the attribute.

5. The partial key attribute is underlined with a _____ line.
6. A person's social security number would be an example of a(n) _____ attribute.
7. _____ attributes can be subdivided.
8. A(n) _____ attribute cannot be subdivided.
9. An attribute representing one or more bank accounts belonging to a person would be a(n) _____ attribute.
10. It is better to store the date of birth and use the difference between that value and the system date as a(n) _____ attribute, rather than storing a person's age.
11. An entity type without a key attribute is called _____ entity type.
12. A(n) _____ attribute need not be physically stored within the database.
13. A(n) _____ relationship is also known as an identifying relationship.
14. A weak entity must be _____-dependent.
15. The relationship in which an entity type participates more than once is a _____ relationship.
16. If one entity occurrence does not require a corresponding entity occurrence in a particular relationship then participation is_____.
17. If you are unable to understand the distinction between mandatory and optional _____ in relationships, then it might yield designs containing unnecessary temporary rows to accommodate the creation of required entities.
18. A(n) _____ relationship exists when three entities are associated.
19. By identifying the attributes of the entities, you can better understand the _____ among entities.
20. The most serious drawback in the ER model is that it cannot depict _____.
21. A weak entity type always has a _____ participation constraint with respect to its identifying relationships.
22. The entity types are represented in ER-diagrams by _____.
23. The relationships are displayed as _____ in ER-diagrams.
24. The multivalued attributes are represented in ER-diagrams by _____.
25. _____ specifies the maximum number of relationship instances that an entity can participate.

Multiple Choice Questions

1. A person's name, birthday, and social security number are all examples of
   (*a*) Entities (*b*) Attributes (*c*) Relationships (*d*) Descriptions
2. An attribute that can be broken down into smaller parts is called a(n) ____ attribute.
   (*a*) simple (*b*) associative (*c*) complex (*d*) composite
3. Which of the following criteria should be considered when selecting an identifier?
   (*a*) Choose an identifier that will not be null.
   (*b*) Choose an identifier that doesn't have large composite attributes.
   (*c*) Choose an identifier that is stable.
   (*d*) All of the above.
4. A relationship where the minimum and maximum cardinality are both one is a(n) _____ relationship.
   (*a*) optional (*b*) mandatory link (*c*) unidirectional (*d*) mandatory one

5. Which statement is false?
   (*a*) Each attribute of a relation has a name.
   (*b*) Attribute values are (normally) required to be atomic.
   (*c*) The special value null is a member of every domain.
   (*d*) None of above
6. Customers, cars, and parts are examples of
   (*a*) entities (*b*) attributes (*c*) relationships (*d*) cardinals
7. The following figure shows an example of



   (*a*) a composite attribute (*b*) a multivalued attribute
   (*c*) a relational attribute (*d*) a derived attribute.
8. Which is false?
   (*a*) Relationship type is grouped by the same type of relationship instances.
   (*b*) The current state of a relationship type is the relationship set.
   (*c*) Relationship type identifies the relationship name and the participating entity types.
   (*d*) Relationship type identifies certain relationship constraints.
9. Which is false?
   (*a*) A relationship can have one or more attributes.
   (*b*) A weak entity must participate in an identifying relationship type with an owner or identifying entity type.
   (*c*) Cardinality ratio specifies maximum participation.
   (*d*) An attribute relate two entities.
10. A student can attend 5 courses. Different professors can offer the same courses. The relationship of students to professors is a _____ relationship.
    (*a*) many-to-many (*b*) one-to-many (*c*) one-to-one (*d*) many-to-one
11. In an E-R, *Y* is the dominant entity and *X* is a subordinate entity. Then which of the following is incorrect:
    (*a*) Operationally, if *Y* is deleted, so is *X*
    (*b*) Existence is dependent on *Y*
    (*c*) Operationally, if *X* is deleted, so is *Y*
    (*d*) Operationally, if *X* is deleted, and remains the same.
12. When an E-R diagram is mapped to tables, the representation is redundant for
    (*a*) weak entity sets (*b*) weak relationship sets
    (*c*) strong entity sets (*d*) strong relationship sets.
13. When we map a multivalued attribute for entity E from the ER model to the relational model, we will create

(*a*) many relations, one for each of the distinct values of the attribute
(*b*) one relation that contains a foreign key and a column for the attribute
(*c*) a column in the relation that represents the entity E
(*d*) none of the above .

14. The ERD is used to graphically represent the ____ database model
(*a*) Condensed (*b*) Physical (*c*) Logical (*d*) Conceptual.

15. An entity type having the or more parent entity type that themselves subclass entity-types within the same classification hierarchy is called
(*a*) Shared subclass (*b*) Associative subclass
(*c*) Subclass assistant (*d*) Entity type.

16. The set of possible values for an attribute is called a
(*a*) domain (*b*) range (*c*) set (*d*) key.

17. Weak entities can alternatively be modelled as a repeating group of
(*a*) Employees (*b*) Nodes (*c*) Attributes (*d*) Parent entity

18. The ideal number of attributes used to make up a primary key is ____.
(*a*) zero (*b*) one (*c*) two (*d*) six

19. Which of the following key consists of more than one attribute?
(*a*) Primary (*b*) Foreign (*c*) Composite (*d*) Domain.

20. Which of the following attribute can be further subdivided to yield additional attributes?
(*a*) Composite (*b*) Simple (*c*) Single–valued (*d*) Multivalued

21. Basic entity-relationship modelling and normalization techniques capture only the
(*a*) Modelling of complex and unstructured data types
(*b*) Structure and static relationships of structured data
(*c*) Dynamic and complex relationship of structured data
(*d*) Concept of modelled data

22. In an E-R diagram relationship is represented by
(*a*) circles (*b*) rectangles
(*c*) diamond shaped box (*d*) ellipse.

23. One entity may be
(*a*) related to only one other entity (*b*) related to itself
(*c*) related to only two other entities (*d*) related to many other entities.

24. Some attributes are classified as
(*a*) Simple (*b*) Complex (*c*) Defined (*d*) Grouped.

25. Which of the following might be represented with a multivalued attribute?
(*a*) Person's name (*b*) Household phone numbers
(*c*) Bank account balance (*d*) Book title.

26. Which of the following might be represented with a single-valued attribute?
(*a*) Person's phone number(s)
(*b*) Car's color
(*c*) Employee's educational background
(*d*) Person's social security number.

27. Which of the following type of attribute cannot be created in a DBMS?
    (*a*) Derived (*b*) Multivalued (*c*) Simple (*d*) Composite.
28. Which of the following should be a derived attribute?
    (*a*) Person's name (*b*) Person's age
    (*c*) Person's social security number (*d*) Person's phone number.
29. A derived attribute
    (*a*) must be stored physically within the database
    (*b*) need not be physically stored within the database
    (*c*) has many values
    (*d*) must be based on the value of three or more attributes
30. A relationship is an association between
    (*a*) objects (*b*) entities (*c*) databases (*d*) fields.
31. Knowing the ____ number of entity occurrences is very helpful at the application software level.
    (*a*) Maximum (*b*) Minimum (*c*) Exact (*d*) Both (*a*) and (*b*)
32. In the ERD, cardinality is indicated using the ____ notation.
    (*a*) (max, min) (*b*) (min, max) (*c*) [min ... max] (*d*) {min | max}
33. Making sure all ____ are identified is a very important part of a database designer's job.
    (*a*) business rules (*b*) cardinalities
    (*c*) derived attributes (*d*) relationships
34. By relation cardinality we mean
    (*a*) number of items in a relationship
    (*b*) number of relationships in which an entity can appear
    (*c*) number of items in an entity
    (*d*) number of entity sets which may be related to a given entity
35. When the *PK* of one entity does not contain the *PK* of a related entity, the relationship is
    (*a*) missing (*b*) weak (*c*) strong (*d*) neutral
36. If an entity's existence depends on the existence of one or more other entities, it is said to be ____-dependent.
    (*a*) existence (*b*) relationship (*c*) business (*d*) weak
37. A student can take not more than 5 subjects in a semester. The number of students allowed in a subject in a semester is not more than 40. The student – subject relationship is
    (*a*) 5 : 40 (*b*) 40 : 5 (*c*) N : 5 (*d*) 40 : M
38. Which of the following entity has a primary key that is partially derived from the parent entity in the relationship?
    (*a*) strong (*b*) weak (*c*) business (*d*) relationship
39. The term "____" is used to label any condition in which one or more optional relationships exist.
    (*a*) Participation (*b*) Optionality (*c*) Cardinality (*d*) Connectivity
40. The existence of a(n) ____ relationship indicates that the minimum cardinality is 1 for the mandatory entity.
    (*a*) mandatory (*b*) optional (*c*) multivalued (*d*) single-valued

41. If an employee within an EMPLOYEE entity has a relationship with itself, that relationship is known as a ____ relationship.
    (*a*) self (*b*) self-referring (*c*) looping (*d*) recursive
42. If an entity appears in N relationships then it is
    (*a*) a 1 : 1 relationship (*b*) a 1 : N relationship
    (*c*) a N : 1 relationship (*d*) a N : M relationship
43. Assume we want to map a 1 : N relationship type, which involves entity types R and S, where R is on the N side of the relationship type and S is on the 1 side, to the relational model. The attributes of the relationship type should be mapped to attributes of
    (*a*) the corresponding relation for the entity type R
    (*b*) the corresponding relation for the entity type S
    (*c*) the corresponding relations for both entity types R and S
    (*d*) none of the above
44. A weak entity type
    (*a*) must have total participation in an identifying relationship
    (*b*) does not have a key attribute(s)
    (*c*) both (*a*) and (*b*)
    (*d*) none of the above
45. Which of the following relationships are most common?
    (*a*) Unary (*b*) Binary (*c*) Ternary (*d*) Higher degree
46. Which of the following statements best decribes the function of an entity relation model?
    (*a*) An ER model is concerned primarily with a logical view of the data and secondly with the physical implementation
    (*b*) An ER model is concerned primarily with a physical implementation of the data and secondly with the logical view
    (*c*) An ER model provides a view of the logic of the data and not the physical implementation
    (*d*) An ER model is entirely concerned with modelling the physical implemetation
47. A many-to-many relationship from an ER diagram can be represented in the relational model by a _____
    (*a*) domain (*b*) primary key
    (*c*) relation with two foreign keys (*d*) a single attribute
48. Which of the following step occurs first in the process of building an ERD?
    (*a*) Develop the initial ERD.
    (*b*) Create a detailed narrative of the organization's description of operations.
    (*c*) Identify the attributes and primary keys that adequately describe the entities.
    (*d*) Identify the business rules based on the description of operations.
49. Which of the following complex requirements may dictate data transformations, and they may expand the number of entities and attributes within the design?
    (*a*) Information (*b*) Entity (*c*) Design (*d*) Processing
50. Which of the following do you create first when creating a database?
    (*a*) Primary keys (*b*) Foreign keys (*c*) Data dictionary (*d*) All of the above

51. Consider the following figure representing instances of a relationship between employees and the department that the employees work in.

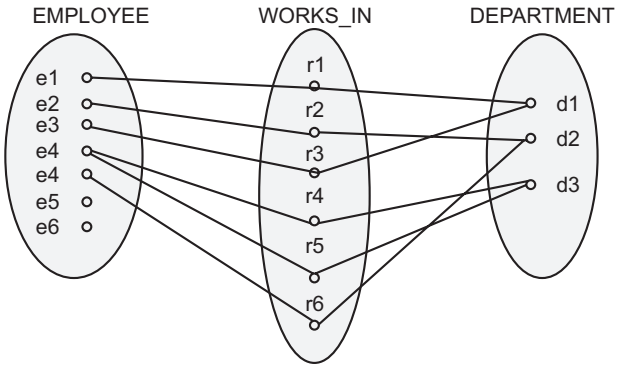


FIGURE 2.46

Which of the following relationships are represented by the above figure?

(*a*) 1 : 1 Cardinality Ratio between EMPLOYEE and DEPARTMENT and total participation from EMPLOYEE and total participation from DEPARTMENT

(*b*) 1 : N Cardinality Ratio between EMPLOYEE and DEPARTMENT and partial participation from EMPLOYEE and partial participation from DEPARTMENT

(*c*) M : N Cardinality Ratio between EMPLOYEE and DEPARTMENT and partial participation from EMPLOYEE and total participation from DEPARTMENT

(*d*) N : 1 Cardinality Ratio between EMPLOYEE and DEPARTMENT and total participation from EMPLOYEE and total participation from DEPARTMENT

52. The entity type on which the _____type depends is called the identifying owner.

(*a*) Strong entity (*b*) Relationship (*c*) Weak entity (*d*) E-R (*UGC-NET*)

53. Suppose we map the following ER diagram to the relations E1(A, B) and E2(B, C). The create table statement for E2 is defined as the following: Create table E2(B integer primary key, C integer). Which one of the following create table statement would be correct for E1?

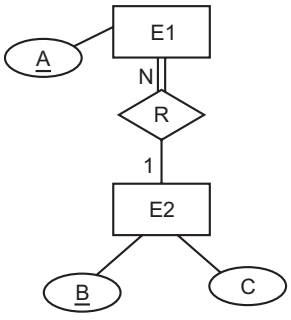


FIGURE 2.47

(*a*) Create table E1(A integer primary key, B integer not null)

(*b*) Create table E1(A integer unique, B integer references E2)

(*c*) Create table E1(A integer primary key, B integer not null references E2)

(*d*) Create table E1(A integer, B integer)

54. For the following ER diagram, we map the relationship, R, to a relation with attributes A, B and C. How many times can any specific combination of values for attributes A and B occur in that relation?

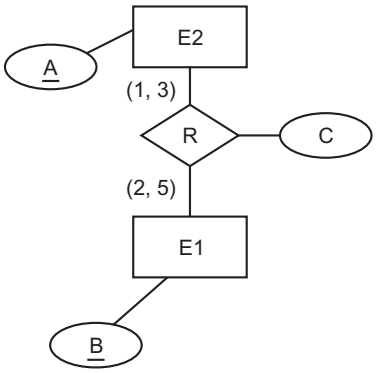


FIGURE 2.48

(*a*) 1 (*b*) 3 (*c*) 5 (*d*) 15

55. Which of the following relational database schemes is a correct representation (via the mapping steps from the text) for the following ER diagram?

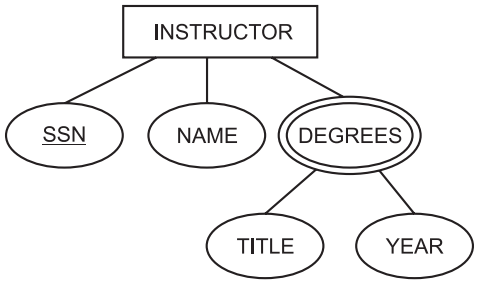


FIGURE 2.49

(*a*) INSTRUCTOR(SSN,NAME) DEGREES(SSN,TITLE,YEAR)

(*b*) INSTRUCTOR(SSN,NAME,TITLE,YEAR)

(*c*) INSTRUCTOR(SSN,NAME) DEGREESA(SSN,TITLE) DEGREESB(SSN,YEAR)

(*d*) None of the above

56. The E-R model is expressed in terms of (*UGC-NET*)

(*i*) Entities

(*ii*) The relationship among entities

(*iii*) The attributes of the entities.

Then

(*a*) (*i*) and (*iii*) (*b*) (*i*) and (*ii*) (*c*) (*ii*) and (*iii*) (*d*) none of these

57. An entity-relationship diagram is a tool to represent (*UGC-NET*)

(*a*) Data model (*b*) Process model (*c*) Event model (*d*) Customer model

58. A primary key for an entity is (*UGC-NET*)

(*a*) A candidate key (*b*) Any attribute

(*c*) A unique attribute (*d*) A superkey

59. An entity instance is a single occurrence of an (*UGC-NET*)

(*a*) Entity type (*b*) Relationship type

(*c*) Entity and relationship type (*d*) None of these

60. An entity has (*UGC-NET*)

(*i*) a set of properties

(*ii*) a set of properties and values for all the properties

(*iii*) a set of properties and the values for some set of properties may non-uniquely identify an entity

(*iv*) a set of properties and the values for some set of properties may uniquely identify an entity.

Which of the above are valid?

(*a*) (*i*) only (*b*) (*ii*) only (*c*) (*iii*) only (*d*) (*iv*) only

ANSWERS

True/False

1. T
2. T
3. F
4. T
5. T
6. F
7. F
8. T
9. T
10. F
11. F
12. F
13. T
14. T
15. T
16. F
17. T
18. F
19. F
20. T
21. T
22. T
23. F
24. F
25. T
26. F
27. F

Fill in the Blanks

1. atomic
2. derived
3. Attributes
4. Domain
5. dotted
6. single-valued simple
7. Composite
8. simple
9. multivalued
10. derived
11. weak
12. derived
13. strong
14. existence
15. recursive
16. optional
17. participation
18. ternary
19. relationships
20. relationships
21. total
22. rectangles
23. diamonds
24. double ovals
25. Cardinality

Multiple Choice Questions

1. (*b*)
2. (*d*)
3. (*d*)
4. (*d*)
5. (*d*)
6. (*a*)
7. (*a*)
8. (*b*)
9. (*d*)
10. (*a*)
11. (*c*)
12. (*b*)
13. (*b*)
14. (*d*)
15. (*a*)

16. (*a*)
17. (*d*)
18. (*b*)
19. (*c*)
20. (*a*)
21. (*b*)
22. (*c*)
23. (*d*)
24. (*a*)
25. (*b*)
26. (*d*)
27. (*b*)
28. (*b*)
29. (*b*)
30. (*b*)
31. (*d*)
32. (*b*)
33. (*a*)
34. (*b*)
35. (*b*)
36. (*a*)
37. (*b*)
38. (*b*)
39. (*b*)
40. (*a*)
41. (*d*)
42. (*b*)
43. (*a*)
44. (*c*)
45. (*b*)
46. (*c*)
47. (*c*)
48. (*b*)
49. (*a*)
50. (*c*)
51. (*c*)
52. (*c*)
53. (*c*)
54. (*a*)
55. (*a*)
56. (*b*)
57. (*a*)
58. (*c*)
59. (*a*)
60. (*d*)

EXERCISES

Short Answer Questions

1. What is entity? Give some examples.
2. What is attributes? Give some examples.
3. What is domain? Give an example.
4. What is entity set? Give an example.
5. Give names of various types of attributes.
6. What is simple attribute? Give an example.
7. What is composite attribute? Give an example.
8. Explain the difference between simple and composite attributes. Provide at least one example of each.
9. What is single valued attribute? Give an example.
10. What is multivalued attribute? Give an example.
11. What is the difference between single valued and multivalued attribute?
12. What is stored attribute? Give an example.
13. What is derived attribute? Give an example.
14. What is the difference between derived attribute and stored attribute?
15. What is an entity-relationship diagram and how is it read?
16. What is relationship?
17. What is relationship set? Give an example.
18. What is the degree of relationship set?
19. What are three types of data relationships?
20. What is required of two tables in order for the tables to be related?

21. What is binary relationship set? Give an example.

22. What is ternary relationship set? Give an example.

23. What is recursive relationship set? Give an example.

24. What is role? Give an example.

25. What is mapping constraints?

26. What is mapping cardinalities?

27. Explain various types of mapping cardinalities.

28. What are participation constraints?

29. What is a key?

30. Give names of various keys.

31. What is candidate key? Give example.

32. What is primary key? Give example.

33. What is foreign key? Give example.

34. What are the various symbols of ER diagram?

35. What are the limitations of ER diagram?

36. What is strong entity set? Give an example.

37. What is weak entity set? Give an example.

38. What is a logical data model? Identify some methods used to translate an entity-relationship diagram into a data model.

39. What is an integrity constraint? What are the five types of integrity constraints?

40. Is an attribute that is single-valued always simple? Why or why not?

41. What is a weak relationship? Provide an example.

42. What is a ternary relationship? Give some examples of business rules that specify the need for a ternary or higher-order relationship.

43. Is an E-R diagram with m entities and n relationships will translate to $m+n$ tables.

Ans. No, many-to-one relationships can often be merged and therefore reduce the overall number of tables needed.

44. If E is a weak entity set, then its key can only be the key attributes of E's supporting entity sets.

Ans. No, the key of E can contain its own attributes as well.

45. What is EER model?

46. What is supertype? Give an example.

47. What is subtype? Give an example.

48. What is specialization? Give an example.

49. What is generalization? Give an example.

50. What is attribute inheritance? Give an example.

51. What is aggregation? Give an example.

52. What are participation constraints?

53. What is partial participation? Give an example.

54. What is total participation? Give an example.

55. What are disjoint constraints? Give an example.

56. What is categorization? Give an example.

57. What are the steps to convert an ER diagram into tables?

58. A timetable database is required for a University Department. Each taught event is part of a module, each event will have exactly one member of staff associated and several individual students. Each event takes place in a single weekly time slot. Each time slot has a day of the week and a time of day associated. Each of the weekly time slots is exactly one hour long, however we wish to represent the fact that some events take more than one hour. Which of the following does not represent a possible solution AND why?

(*a*) A many-to-many relation between events and time-slots is established

(*b*) A one-to-many relation between events and time-slots is established

(*c*) Each event has an attribute "start" which refers to time-slots and "duration" which gives the length of the event in minutes

(*d*) Each event has an attribute "start" which refers to time-slots and "duration" which gives the number of slots spanned

(*e*) Each event has two attributes "first" and "last" each of which refer to time-slots

Ans. B, Currently the relation only consists of a start time, and it is believed to last 1 hour. The addition of a 1 : N relationship means that a single event can have multiple time-slots, and thus we can now imply how long the event takes and thus satisfy the new criteria. It however stops more than one event happening at the same time, which would severely weaken the timetable.

59. Is it possible for a relationship to be declared many-one and be symmetric? A symmetric relation is one that is its own inverse. Under what conditions would this be true?

Ans. A relationship that is declared M-O can be symmetric only if

– it runs from an entity set to itself and

– it is actually one-one

60. Consider the ER diagram shown below where A, B and C are entity sets.

(*a*) Specify the condition(s) that is(are) necessary in order to represent all three sets with a single table.

(*b*) Now specify the condition(s) that is(are) necessary in order to represent all three sets with two tables, one for B and one for C.

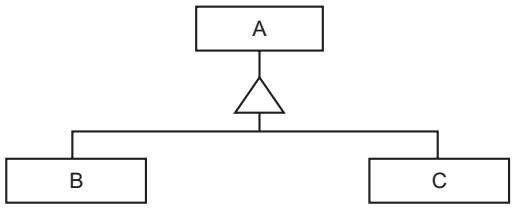


FIGURE 2.50

Ans. (*a*) The ISA relationship must be disjoint. B and C must have the same attributes.

(*b*) The ISA relationship must be total.

61. Suppose we define a database about the customers of a bank and the loans they have received from the bank. For each customer we need to record information about their name, address, phone number and the company they work for. For each loan we need to record the amount, the interest rate, date the loan was issued, and the date the loan should be paid off.

(*a*) Is it a good idea to represent the company for which a customer works as an attribute of the customer or as a relationship? Briefly justify your answer.

(*b*) Which is the best way to represent the relationship between the customer and their loans: by defining the loan as an attribute of the customer, or - by making the loan a separate entity set and defining a relationship set between it and the customer?

Ans. (*a*) The Company should be an attribute of the customer, assuming each customer works for a single company. We don't need to keep any information for each company.

(*b*) The loan should be a separate entity set associated with a customer through a relationship for the following reasons:

(*i*) A customer may have more than one loans.

(*ii*) A loan has additional information on its own.

Long Answer Questions

1. Explain the difference between a weak and a strong entity set. Why do we have the concept of weak entity set?

2. Explain the meaning of following with suitable examples:

(*i*) Primary key (*ii*) Super key

(*iii*) Condidate key (*iv*) Foreign key

(*v*) Alternate key.

3. Explain the following terms with examples:

(*i*) Cardinality (*ii*) Entity

(*iii*) Relationship (*iv*) Participation constraint.

4. Can a weak entity set have more than one primary key for the same foreign key? Justify with example.

5. "All candidate keys can be primary keys". Do you agree with the statement? Justify your answer.

6. Explain the following terms:

(*i*) Aggregation (*ii*) Generalization

(*iii*) Super key (*iv*) Candidate keys.

7. What is a weak-entity set? Explain with example.

8. Explain the distinction among the terms—primary key, candidate key and the super key.

9. Discuss in detail the various constructs used in ER-diagram, giving suitable example.

10. What do you understand by the term ER diagram? Sketch the ER diagram of Railway reservation system and then reduce this diagram into tables.

11. Explain the following terms briefly:

(*i*) Attribute (*ii*) Domain

(*iii*) Entity (*iv*) Relationship

(*v*) Entity set (*vi*) One to many relationship

(*vii*) Participation constraint (*viii*) Weak entity set

(*ix*) Aggregation (*x*) Composite key.

12. What is entity, entity type, entity sets and attribute? What is mapping cardinalities? Explain different cardinalities.

13. Discuss the role of ER diagrams in the design of relational database. Illustrate your answer with the help of an example.

14. Define the term ER diagram. Explain the various types of relationships in a ER diagram. Draw an ER diagram for taking your own example.

15. What is an ER diagram? Give an ER diagram for a database showing fatherhood, motherhood and spouse relationship among men and women.

16. Design the ER diagram for the Educational Institute System. Make you own assumptions about the system.

17. Differentiate between the following with examples:

(*a*) Entity and attributes (*b*) Primary and foreign key

(*c*) Candidate key and primary key.

18. In an organization several projects are undertaken. Each project can employ one or more employees. Each employee can work on one or more projects. Each project is undertaken on the request of a client. A client can request for several projects. Each project has only one client. A project can use a number of items and an item may be used by several projects. Draw an ER diagram and convert it into a relational schema.

19. A bank has many branches, with many customers. A customer can open many different kinds of accounts with the bank. Any customer of the bank can take loan from the bank. All branches can give loans. Banks have also installed automatic teller machines, from which a customer can withdraw from his/her bank. Draw the ER-diagram for the bank specifying aggregation, generalization or specialization hierarchy, if any. Create 3NF tables of your design. Make suitable assumptions if any.

20. Make an ER diagram for a diagnostic lab. It should keep track of customers, raw material, professional and support staff, and the reports being generated for tests. Make and state assumptions, if any.

21. A University has many academic units named schools. Each school is headed by a director of school. The school has teaching and non-teaching staff. A school offers many courses. A course consists of many subjects. A subject is taught to the students who have registered for that subject in a class by a teacher. Draw the ER diagram for the University specifying aggregation, generalization or specialization hierarchy, if any. Create 3NF tables of your design. Make suitable assumptions, if any.

22. A national bank and an international bank decide to merge. Assume, that both the organizations use almost similar ER diagrams. Prepare the ER diagram for the merged bank. Make, and state, suitable assumptions if any.

23. A construction company has many branches spread all over the country. The company has two types of constructions to offer : Housing and Commercial. The housing company provides low income housing, medium style housing and high end housing schemes, while on the commercial side it offers multiplexes and shopping zones. The customers of the company may be individuals or corporate clients. Draw the ER diagram for the company showing generalization or specialization hierarchies and aggregations, if any. Also create 3NF tables of your design.