

# 1. Fundamentals of DBMS

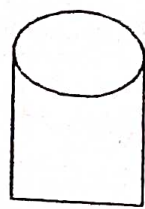
## Introduction :

Before defining a database management system (DBMS), we must have a clear understanding of what database is all about. A database consists of four elements as shown in figure 1.1.

- Data
- Relationships
- Constraints
- Schema

By data we mean known facts that can be recorded and that have implicit meaning.

➔ Data are binary computer representations of stored logical entities. They are distinct pieces of information, usually formatted in a special way. Software is divided into two general categories - data and programs. A program is a collection of instructions for manipulating data. Data can exist in a variety of forms as numbers or text on pieces of paper, as bits and bytes stored in electronic memory, or as facts stored in a person's mind.



Database

## प्रस्तावना :

डाटाबेस मैनेजमेंट सिस्टम (DBMS) को परिभाषित करने के पूर्व हमें स्पष्ट रूप से यह मालूम होना चाहिए कि डाटाबेस क्या होता है। DBMS डाटाबेस को मैनेज करता है। जैसा कि चित्र 1.1 में दर्शाया गया है, कोई भी डाटाबेस चार तत्वों से बनता है-

- डाटा
- रिलेशनशिप्स
- कन्स्ट्रैन्स
- स्कीमा

डाटा से हमारा तात्पर्य ऐसे ज्ञात तथ्यों से होता है, जो कोई किए जा सकते हैं और जिनके कुछ निहितार्थ होते हैं।

डाटा, संग्रहीत लॉजिकल तत्वों के बायनरी कंप्यूटर ही रीप्रेजेंटेशन होते हैं। वे इंफॉर्मेशन के विशिष्ट हिस्से होते हैं, जो खास तरह से फॉर्मेट किए हुए होते हैं। सॉफ्टवेयर दो आम श्रेणियों में बांटे जाते हैं - डाटा और प्रोग्राम। प्रोग्राम डाटा को मैनिप्यूलेट करने के लिए निर्देशों का संग्रह होता है। डाटा कई रूपों में मिल सकता है, जैसे कागज पर अंकों या टेक्स्ट के रूप में, इलेक्ट्रॉनिक मेमोरी में संग्रहीत बिट्स व बाइट्स के रूप में या किसी व्यक्ति के मस्तिष्क में संग्रहीत तथ्यों के रूप में।

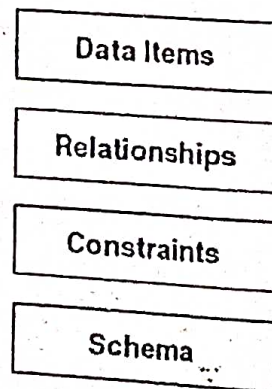


Fig. 1.1 Components of a Database

In database management systems, data files are the files that store the database information, whereas other files, such as index files and data dictionaries, store administrative information known as metadata.

डाटाबेस मैनेजमेंट सिस्टम में डाटा फाईल्स वे फाईल्स होती है, जो डाटाबेस इंफॉर्मेशन को संग्रहीत करती है, जबकि इंडेक्स फाईल्स व डाटा डिक्शनरी जैसी अन्य फाईलें प्रशासकीय जानकारी को संग्रहीत करती हैं। इस जानकारी को मेटाडाटा कहा जाता है।

→ Relationships represent a correspondence between the various data elements, constrain are predicates that define correct database states.

→ Schema describes the organization data and relationships within the database.

### Data Base :

→ A collection of data designed to be used by different people is called a database. It is a collection of interrelated data stored together with controlled redundancy to serve one or more applications in an optimal fashion. The data are stored in such a fashion that they are independent of the programs of people using the data. A common and controlled approach is used in adding new data and modifying and retrieving existing data within the database.

A database system is basically a computer based record keeping system. The collection of data, usually referred to as the database, contains information about one particular enterprise. It maintains any information that may be necessary to the decision-making processes involved in the management of that organization.

The intention of a database is that the same collection of data should serve as many applications as possible. Hence, a database is often conceived of as the repository of information needed for running certain functions in a corporation or organization. Such a database would permit not only the retrieval of data but also the continuous modification of data needed for control of operations.

### Purpose of Databases :

A database system should be repository of the data needed for an organization's data processing. That data should be accurate, private and protected from damage. It should be organized so that diverse applications with different data

रिलेशनशिप्स विभिन्न डाटा तत्वों के बीच आदान-प्रदान व अनुरूपता को दर्शाता है। कंस्ट्रेंट्स वे प्रडिकेट्स (कथन) हैं, जो सही डाटाबेस अवस्थाएँ दर्शाते हैं। स्कीमा, ऑर्गेनाइजेशन डाटा और डाटाबेस के भीतर रिलेशनशिप को वर्णित करता है।

### डाटाबेस :

डाटा का ऐसा संग्रह जो विभिन्न लोगों द्वारा उपयोग में लाने के लिए डिजाइन किया गया है, डाटाबेस कहलाता है। यह अंतरसंबंधित डाटा का संग्रह होता है, जिसे नियंत्रित प्रचुरता के साथ, एक साथ संग्रहित रखा जाता है, ताकि एक या अधिक एप्लीकेशन के लिए उसका सर्वाधिक संभव उपयोग किया जा सके। डाटा को कुछ इस प्रकार रखा जाता है कि वह उसका उपयोग करने वाले लोगों के प्रोग्राम्स से स्वतंत्र रहता है। डाटाबेस में नया डाटा शामिल करने और वर्तमान डाटा में फेरबदल करने और उसे हासिल करने के लिए एक समान और नियंत्रित तरीका अपनाया जाता है।

डाटाबेस सिस्टम मूलतः कम्प्यूटर आधारित रेकॉर्ड रखने का सिस्टम है। आमतौर पर जिसका उल्लेख डाटाबेस के रूप में किया जाता है, उस डाटा के संग्रह में एक विशिष्ट उपक्रम के बारे में इंफॉर्मेशन होती है। यह ऐसी कोई भी इंफॉर्मेशन संग्रहीत रखता है, जो किसी संस्थान के प्रबंधन की निर्णय लेने की प्रक्रिया के लिए आवश्यक हो।

डाटाबेस का इरादा यह होता है कि डाटा के समान संग्रह का उपयोग जितने एप्लीकेशन्स के लिए संभव हो सके, किया जाए। इसलिए डाटाबेस को प्रायः ऐसी इंफॉर्मेशन के कोष के रूप में लिया जाता है, जो किसी निगम या संस्थान के कुछ कार्यों को संपन्न करने के लिए जरूरी हो। ऐसा कोई भी डाटा न सिर्फ डाटा को फिर से हासिल करने की अनुमति देगा, बल्कि विभिन्न ऑपरेशन्स के लिए जरूरी डाटा के लगातार फेरबदल व रूपांतरण की सुविधा भी देगा।

### डाटाबेस का उद्देश्य :

डाटाबेस सिस्टम संस्थान की डाटा प्रोसेसिंग के लिए आवश्यक डाटा का कोष होना चाहिए। यह डाटा त्रिभुज, शुद्धिहीन, सही, निजी और क्षति के खिलाफ संरक्षित होना चाहिए। इसे इस प्रकार ऑर्गेनाइज किया जाना चाहिए कि भिन्न डाटा जरूरतों वाले, विस्तृत रूप से अलग एप्लीकेशन डाटा को

requirements can employ the data. Different application programmers and different end users will have different views of the data which must be derived from a common overall data structure. Their methods of accessing or searching the data will differ.

The ways in which end users want to utilize existing data will constantly change and in some cases demands for new uses of the data will arise rapidly and urgently. The extent to which these demands can be satisfied determines the overall value of the database system.

### Advantages of DBMS :

1. Database reduces the data redundancy (Duplication of data) to a large extent. In non-database systems, each application or department has its own private files resulting in considerable amount of redundancy of the stored data. Thus storage space is wasted. By having a centralized database most of this can be avoided. We do not say or suggest that all redundancy should be eliminated. Sometimes there are sound business and technical reasons for maintaining multiple copies of the same data. In a database systems, however, this redundancy can be controlled.

2. Database can control data inconsistency to a large extent. This is really a corollary to the above point. When the same data is duplicated and changed are made at one site, which is not propagated to the other site, it gives rise to inconsistency. Then the two entries regarding the same data will not agree. At such times the data is said to be inconsistent. So if the redundancy is removed chances of having inconsistent data is also removed.

3. Database Facilitate Sharing of Data : Sharing of data means that individual pieces of data in the database may be shared among several different users, in the sense that each of those

इस्तेमाल कर सकें। विभिन्न एप्लीकेशन प्रोग्रामर और एंड यूजर का, डाटा के बारे में भिन्न दृष्टिकोण होगा, लेकिन डाटा ऐसा होना चाहिए कि उसे एक कॉमन डाटा स्ट्रक्चर से हासिल किया जा सकें। डाटा को एक्सेस और सर्च करने की उनकी विधियाँ भी भिन्न होगी।

मौजूदा डाटा को एंड यूजर जिस तरह उपयोग करना चाहेंगे, उसमें लगातार बदलाव आता रहेगा और कुछ मामलों में तो डाटा के नए उपयोग की मांग तेजी से और अति आवश्यक रूप से बढ़ेगी। जिस हद तक ये मांगें पूरी की जा सकेंगी, उसी से डाटाबेस सिस्टम की कुल क्षमता का मूल्यांकन होगा।

### डाटाबेस मैनेजमेंट सिस्टम के फायदे :

1. डाटाबेस ने काफी हद तक डाटा रीडिंडेंसी (डाटा डुप्लीकेशन) घटा दी है। जिस सिस्टम में डाटाबेस नहीं होता है, उनमें प्रत्येक एप्लीकेशन और डिपार्टमेंट की अपनी निजी फाइल्स होती हैं, जिसके परिणामस्वरूप संग्रहित डाटा में काफी हद तक रीडिंडेंसी (फालतू सामग्री इकट्ठा होना) हो जाती है, परिणामस्वरूप स्टोरेज स्पेस बर्बाद होती है। केंद्रीकृत डाटाबेस बनाकर इसमें से ज्यादातर रीडिंडेंसी को टाला जा सकता है। यहाँ यह नहीं कहा या सुझाया जा रहा है कि सारी रीडिंडेंसी को समाप्त कर दिया जाना चाहिए। कभी-कभी व्यावसायिक और तकनीकी कारणों से एक ही डाटा की एक से अधिक कॉपी रखना बेहतर होता है। डाटाबेस सिस्टम में रीडिंडेंसी को नियंत्रण में रखा जा सकता है।

2. डाटाबेस एक बड़ी हद तक डाटा संबंधी अनियमितता को नियंत्रित कर सकता है। वास्तव में, यह उक्त बिंदु का ही परिणाम है। जब एक ही डाटा डुप्लीकेट होता है और एक साइट पर किए परिवर्तन अन्य साइट पर नहीं भेजे जाते हैं, तब वो इससे अनियमितता पैदा होती है। फिर एक ही डाटा की दो एंट्री नहीं मिलेगी। ऐसी स्थिति में डाटा अनियमित कहलाता है। इसलिए रीडिंडेंसी हटाने पर अनियमित डाटा होने के अवसर भी खत्म हो जाते हैं।

3. डाटाबेस से डाटा शेयरिंग संभव होती है : डाटा शेयरिंग का मतलब यह होता है कि डाटाबेस में डाटा के एकल हिस्सों को कई भिन्न यूजर उपयोग में ला सकते हैं। इस का अर्थ यह है कि प्रत्येक यूजर एक ही प्रकार के डाटा को एक्सेस कर सकता है और

users may have access to the same piece of data and each of them may use it for different purpose.

The database system make sure that not only the existing applications can share the data in the database, but also that new applications can be developed to operate against that same stored data.

✓ **4. Database Enforces Standards :** With the central control of the database, the database administrator can enforce standards.

✓ **5. Database can ensure data security and privacy :** Data Security refers to protection of data against accidental or intentional disclosure to unauthorized persons or unauthorized modification or destruction. And Privacy of Data refers to the rights of individuals and organizations to determine for themselves when, how and to what extent information about them is to be transmitted to others.

A database management system ensures data security and privacy by ensuring that the only means of access to the database is through the proper channel and also by carrying out authorization checks whenever access to sensitive data is attempted.

✓ **6. Integrity can be maintained through database :** Integrity means that the data in the database is accurate. Centralized control of the data helps in permitting the administrator to define constraints to the data in the database.

**7. Conflicting requirements can be balanced through database :** Knowing the overall requirements helps the database designers in creating a database design that is best for the organization.

✓ **Characteristics of Data in a Database :**

The data in a database should have the following features -

उन्में से प्रत्येक उस डेटा का अपनी तरह से भिन्न उद्देश्य के लिए उसी का उपयोग कर सकता है।

डाटाबेस सिस्टम न सिर्फ यह सुनिश्चित करता है कि न सिर्फ मौजूदा एप्लीकेशन डेटा को डाटाबेस में शेयर कर सकता है, बल्कि यह भी तय करता है कि उसी संग्रहित डेटा पर ऑपरेशन करने के लिए नए एप्लीकेशन भी विकसित हो सके।

4. डाटाबेस मानक स्थित करता है : डाटाबेस के सेन्ट्रल कंट्रोल के साथ डाटाबेस एडमिनिस्ट्रेटर स्टैंडर्ड्स लागू कर सकता है।

5. डाटाबेस सुरक्षा और निजता सुनिश्चित करता है : डाटा सुरक्षा का संबंध अनाधिकृत व्यक्ति को दुर्घटनावश या इरादतन डाटा जाहिर करने या अनाधिकृत रूप से डाटा का रूपांतरण करने अथवा उसे नष्ट करने से होता है। इसी प्रकार डाटा की निजता का संबंध न्यासी या संगठन को यह निर्धारित करने का अधिकार है कि कैसे और किस हद तक उनके बारे में इंफॉर्मेशन दूसरों को प्रसारित की जाती है।

एक डाटाबेस मैनेजमेंट सिस्टम सही चैनल के द्वारा डाटाबेस एक्सेस सुनिश्चित कर डाटा सुरक्षा और निजता तय करता है। इसके अलावा संबन्धशील डाटा के एक्सेस का जब संबंध आता है, तो यह ऑथोरिजेशन चेक्स के द्वारा भी सुरक्षा सुनिश्चित करता है।

6. डाटाबेस से इंटीग्रिटी भी बनाई रखी जाती है : इंटीग्रिटी का यहाँ यह अर्थ है कि डाटाबेस में डाटा बिल्कुल सही है। डाटा का केंद्रिक नियंत्रण एडमिनिस्ट्रेटर को डाटाबेस में डाटा के कंस्ट्रेंट्स को परिभाषित करने में मददगार होता है।

7. विरोधाभासी आवश्यकताओं में भी डाटाबेस के द्वारा संतुलन स्थापित किया जा सकता है : संपूर्ण जरूरतों की जानकारी होने से डाटाबेस डिजाइनर को ऐसा डाटाबेस डिजाइन करने में मदद मिलती है, जो संस्थान के लिए सर्वश्रेष्ठ है।

डाटाबेस में डाटा की विशिष्टताएँ :

डाटाबेस में डाटा में निम्न फीचर्स होना चाहिए -

1. **Shared** : Data in a database are shared among different users and applications.
2. **Persistence** : Data in a database exist permanently in the sense the data can live beyond the scope of the process that created it.
3. **Validity/Integrity/Correctness** : Data should be correct with respect to the real world entity that they represent.
4. **Security** : Data should be protected from unauthorized access.
5. **Consistency** : Whenever more than one data element in a database represents related real-world values, the values should be consistent with respect to the relationship.
6. **Non-redundancy** : No two data items in a database should represent the same real-world entity.
7. **Independence** : The three levels in the schema (internal, conceptual and external) should be independent of each other so that the changes in the schema at one level should not affect the other levels.

### Limitations of Database :

- Encoded data (data hard-coded in the application) inflexibility.
- Interdependence between programs and data files.
- Data repetition or redundancy.
- Data inconsistency.
- Lack of data integrity.
- Ad hoc representation of relationships.
- Ad hoc data management techniques.
- Lack of coordination across applications using common data.
- Lack of foolproof data security mechanisms.
- Inability to manage concurrent access of data.
- Non-uniform back-up and recovery methods.

1. **शेअर्ड** : डाटाबेस का डाटा विभिन्न यूजर्स और एप्लीकेशन के बीच बाँटा जाता है अर्थात् वे मिलकर उसका उपयोग करते हैं।
2. **पर्सिस्टेंस** : इसका अर्थ है निरंतरता। किसी डाटाबेस में डाटा स्थायी रूप से स्थिर होता है। यह इस अर्थ में कि डाटा उस प्रोसेस की परिधि के बाहर भी बना रह सकता है, जिसने इसे निर्मित किया था।
3. **वैलिडिटी/इंटिग्रिटी/कोरैक्टनेस** : अर्थात् वैधता, सच्चाई और अचूकता। डाटा जिस वास्तविक दुनिया का प्रतिनिधित्व करते है, उसके संबंध में उन्हें अचूक होना चाहिए।
4. **सिक्योरिटी** : अर्थात् सुरक्षा, डाटा को अनाधिकृत एक्सेस से सुरक्षित रहना चाहिए।
5. **कंसिस्टेंसी** : इसका अर्थ है अनुरूपता। जब भी डाटाबेस में एक से अधिक डाटा एलिमेंट संबंधित वास्तविक विश्व की वेल्यू को दर्शाते है, तब ये वेल्यू संबंधों के संदर्भ में संगत या अनुरूप होनी चाहिए।
6. **नॉन-रीडंडेंसी** : डाटाबेस के कोई भी दो डाटा आयटम ऐसे न हो, जो वास्तविक विश्व के एक ही तथ्य को दर्शाते हों।
7. **स्वतंत्रता** : स्वतंत्रता का तात्पर्य यह है कि स्कीमा के तीन स्तर (आंतरिक, अवधारणात्मक और बाह्य) एक दूसरे से स्वतंत्र होना चाहिए, ताकि एक स्तर की स्कीमा में फेरबदल का दूसरे स्तर पर कोई प्रभाव नहीं पड़े।

### डाटाबेस की सीमाएँ :

- एनकोड किए डाटा (एप्लीकेशन में हार्ड-कोड किया हुआ डाटा) में लचीलेपन का अभाव।
- प्रोग्राम और डाटा फाइल्स के बीच अंतर निर्भरता।
- डाटा का दोहराव या रीडंडेंसी।
- डाटा में अनुरूपता का अभाव।
- डाटा में अचूकता का अभाव।
- रिलेशनशीप का अस्थायी प्रस्तुतिकरण।
- अस्थायी डाटा मैनेजमेंट तकनीक।
- समान डाटा का उपयोग करने वाली एप्लीकेशन्स के बीच समन्वय का अभाव।
- चाक-चौबंद डाटा सुरक्षा प्रणाली का अभाव।
- डाटा के कॉन्कुरेंट एक्सेस के प्रबंधन की अयोग्यता।
- असमान बैकअप और रिकवरी विधियाँ।

## Rules of Database :

The database has a very specific meaning and rules and they are -

Some views of Real world are represented by database. Real world is called miniworld or the universe of discourse.

- Database should show the simultaneous change with the change in real (mini world).
- Database is logically connected data collection with some inherent meaning.
- A random collection of data is never a database.
- To perform a specific task, database is created and manipulated by some intended group of users.

## Database Management System :

A database management system (DBMS) is a collection of programs that enables users to store, create, modify and extract information from a database. The DBMS is hence a general purpose software system that facilitates the process of defining, constructing and manipulating database for various applications.

A database management system is a piece of software that provides for accessing a database, while maintaining all the required features of the data. The major activities, operations and services provided by the DBMS are shown in fig. 1.2.

**Transaction Management :** A transaction is a sequence of database operations that represents a logical unit of work and that accesses a database and transforms it from one state to another. A transaction can update a record, delete one, modify a set of records, etc. When the DBMS does a 'commit' the changes made by the transaction are made permanent. If you don't want to make the changes permanent you can rollback the transaction and the database will remain in its original state.

## डाटाबेस के नियम :

डाटाबेस का विशिष्ट अर्थ होता है और इसके कुछ नियम होते हैं, ये इस प्रकार हैं -

- रीयल वर्ल्ड के कुछ दृष्टिकोण डाटाबेस द्वारा प्रदर्शित किए जाते हैं। रीयल वर्ल्ड को यहाँ मिनी वर्ल्ड कहते हैं अर्थात् सूचना के आदान-प्रदान का संसार।
- रीयल (मिनी वर्ल्ड) में परिवर्तन के साथ डाटाबेस में बदलाव नजर आना चाहिए।
- डाटाबेस लॉजिकली कनेक्ट किया हुआ डाटा होता है, जिसका एक आधारभूत व अभिन्न अर्थ होता है।
- डाटा का अव्यवस्थित संग्रह कभी डाटाबेस नहीं होता है।
- कोई विशेष काम करने के लिए इच्छित उपयोगकर्ताओं का कोई समूह प्रायः डाटाबेस को निर्मित और मेनिप्युलेट करना है।

## डाटाबेस मैनेजमेंट सिस्टम :

डाटाबेस मैनेजमेंट सिस्टम प्रोग्राम्स का ऐसा संग्रह होता है, जो यूजर्स को डाटाबेस में इन्फॉर्मेशन संग्रहीत, निर्मित, रूपांतरित और इससे इन्फॉर्मेशन निकालने की सुविधा देता है। इसलिए DBMS एक आम उपयोगी सॉफ्टवेयर सिस्टम है, जो कि भिन्न एप्लीकेशन्स के लिए डाटाबेस परिभाषित, निर्मित और मेनिप्युलेट (संचालित) करने की प्रक्रिया को आसान बनाता है।

कोई भी डाटाबेस सिस्टम ऐसा सॉफ्टवेयर है, जो डाटा के सभी आवश्यक फीचर्स बनाए रखते हुए डाटाबेस एक्सेस की सुविधा उपलब्ध कराता है। DBMS द्वारा उपलब्ध की जाने वाली प्रमुख गतिविधियाँ, ऑपरेशन्स और सेवाएं चित्र 1.2 में दर्शाई गई हैं।

**1. ट्रांजेक्शन मैनेजमेंट :** ट्रांजेक्शन डाटाबेस ऑपरेशन्स का क्रम होता है, जो काम की तार्किक इकाई को दर्शाता है। यह डाटाबेस को एक्सेस करके इस इकाई को एक से दूसरी अवस्था में रूपांतरित करता है। ट्रांजेक्शन सेट किए हुए रिकॉर्ड को डिलिट या मॉडिफाई आदि कर सकता है। जब DBMS कमेंट करता है, तो ट्रांजेक्शन से किए गए परिवर्तन स्थायी हो जाते हैं। यदि आप परिवर्तनों को स्थायी नहीं बनाना चाहते हैं, तब आप ट्रांजेक्शन को रोलबैक कर सकते हैं और डाटाबेस मूल अवस्था में बना रहेगा।

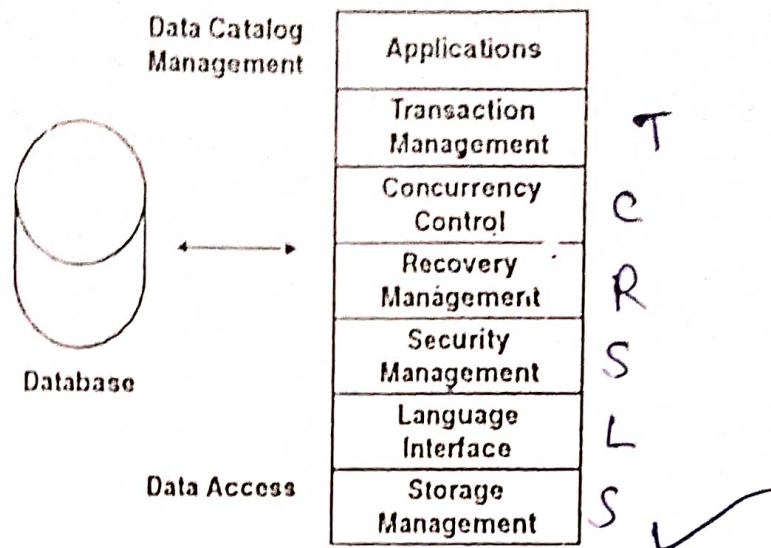


Fig. 1.2 Services Provided by a DBMS

✓ **2. Concurrency Control :** It is the database management activity of coordinating the actions of database manipulation processes that operate concurrently that access shared data and can potentially interfere with one another. The goal of an ideal concurrency management mechanism is to allow concurrency while maintaining the consistency of the shared data.

2. कान्कॉरेंसी कंट्रोल : कान्कॉरेंसी कंट्रोल, डाटाबेस मैनेजमेंट कि वह गतिविधि है, जिसके तहत वह एक साथ होने वाली ऐसी प्रोसेस में समन्वयन करता है, जो शेयरड डाटा एक्सेस करते हैं और जिनके एक-दूसरे में हस्तक्षेप करने की प्रबल संभावना होती है। आदर्श कान्कॉरेंसी मैनेजमेंट प्रणाली का लक्ष्य शेयरड डाटा में एकरूपता बरकरार रखते हुए कान्कॉरेंसी की सुविधा देना है।

✓ **3. Recovery Management :** The recovery management system in a database ensures that the aborted or failed transactions create no adverse effects on the database or the other transactions. Recovery mechanisms in a DBMS make sure that the database is returned to a consistent state after a transaction fails or aborts. Recovery is very related to concurrency in the sense that, the more the concurrency the more is the chance that an aborted transaction can affect many other transactions.

3. रिकवरी मैनेजमेंट : डाटाबेस की यह सुविधा यह सुनिश्चित करती है कि एबोर्ट किए या विफल ट्रांजेक्शन्स, डाटाबेस या अन्य ट्रांजेक्शन्स में कोई विपरीत प्रभाव न उत्पन्न करें। किसी DBMS में रिकवरी प्रणाली यह सुनिश्चित करती है कि किसी विफल ट्रांजेक्शन या फेल्यूर की स्थिति के बाद डाटाबेस वापस अपनी एकरूप स्थिति में आ जाए। रिकवरी का कान्कॉरेंसी से संबंध इस अर्थ में है कि कान्कॉरेंसी जितनी ज्यादा होगी उतने ही इस बात के ज्यादा अवसर होंगे कि एबोर्ट किया ट्रांजेक्शन कई अन्य ट्रांजेक्शन को प्रभावित करे।

✓ **4. Security Management :** Security refers to the protection of data against unauthorised access. Security mechanisms of a DBMS make sure that only authorized users are given access to the data in the database. The level of access for each user and the operations that each user can perform on the data will be monitored and controlled by the DBMS depending on the access privileges of the users.

4. सिक््योरिटी मैनेजमेंट : इसका संबंध अनाधिकृत एक्सेस के खिलाफ डाटा की सुरक्षा से है। सुरक्षा प्रणाली यह सुनिश्चित करती है कि केवल अधिकृत यूजर को ही डाटाबेस में डाटा तक एक्सेस मिले, यूजर्स के एक्सेस विशेषाधिकार के आधार पर DBMS प्रत्येक यूजर और उसके द्वारा सम्पन्न किए जाने वाले ऑपरेशन्स में एक्सेस लेवल की निगरानी और नियंत्रण करना है।

**5. Language Interface :** The DBMS provides support languages used for the definition and manipulation of the data in the database. The data structures are created using the data definition language commands. The data manipulation is done using the data manipulation commands. By providing the language support for data definition and manipulation the DBMS creates an environment where the users can do their jobs without worrying about the physical implementation.

**6. Storage Management :** The DBMS provides a mechanism for management of permanent storage of the data. The internal schema defines how the data should be stored by the storage management mechanism and the storage interfaces with the operating system to access the physical storage.

**7. Data Catalog Management :** Data Catalog or Data Dictionary is a system database that contains descriptions of the data in the database (metadata). It contains information about data, relationships, constraints and the entire schema that organize these features into a unified database. The data catalog can be queried to get information about the structure of the database.

### Needs of DBMS :

A DBMS provides a secure and survivable medium for the storage and retrieval of data. If the users and applications are not sharing data among themselves and if the longevity of the data is only till the end of the process or application that created it, then there is not much need for a DBMS. But in the real world, the data is shared and is persistent (that is the data has a life beyond the boundaries of the programs and applications that created it). Also the real world data have a structure, is related to one another and have constraints. These features are well represented and can be efficiently managed using a DBMS. Also the different users of the data need

5. लैंग्वेज इंटरफेस : डाटाबेस में डाटा को परिभाषित करने और उसके मैनिप्यूलेशन (संचालन) के लिए DBMS सपोर्ट लैंग्वेज उपलब्ध कराता है। डाटा मैनिप्यूलेशन, डाटा मैनिप्यूलेशन कमांड का उपयोग करके किया जा सकता है। डाटा डेफिनिशन और मैनिप्यूलेशन के लिए लैंग्वेज सपोर्ट उपलब्ध कराकर DBMS एक ऐसा वातावरण तैयार करता है, जहाँ यूजर्स अपना काम बिना फिजिकल इम्प्लीमेंटेशन (भौतिक रूप से अपल) की चिंता किए बिना कर सकता है।

6. स्टोरेज मैनेजमेंट : DBMS डाटा के स्थायी संग्रहण के लिए एक प्रणाली उपलब्ध कराता है। आंतरिक स्कीमा यह परिभाषित करता है कि स्टोरेज मैनेजमेंट प्रणाली कैसे डाटा स्टोर करेगी और भौतिक संग्रहण पर एक्सेस के लिए स्टोरेज, ऑपरेटिंग सिस्टम से कैसे इंटरफेस करेगी।

7. डाटा केटलॉग मैनेजमेंट : डाटा केटलॉग या डाटा डिक्शनरी एक ऐसा सिस्टम डाटाबेस है, जिसमें डाटाबेस (मेटाडाटा) में मौजूद डाटा का ब्यौरा होता है। इसमें डाटा, रिलेशनशीप, कंस्ट्रेंट्स और उस पूरी स्कीमा की जानकारी होती है, जो एकीकृत डाटाबेस में इन फीचर्स को ऑर्गेनाइज करते हैं। डाटाबेस के स्ट्रक्चर का पता लगाने के लिए डाटा केटलॉग में क्वेरी की जा सकती है।

### DBMS की आवश्यकता :

DBMS, डाटा के स्टोरेज और उसके रीट्राइबल के लिए सुरक्षित तथा बना रहने वाला माध्यम उपलब्ध कराता है। यदि यूजर्स और एप्लीकेशन्स आपस में डाटा शेयर नहीं करते हैं और डाटा की उम्र सिर्फ उस प्रोसेस या एप्लीकेशन खत्म होने तक के लिए है, जिसने इसे निर्मित किया है, तो ऐसी दशा में DBMS की ज्यादा आवश्यकता नहीं होती है। हालांकि रियल वर्ल्ड में डाटा शेयर भी किया जाता है और यह परसिस्टेंट भी होता है, अर्थात् डाटा का जीवन इसे निर्मित करने वाले प्रोग्राम और एप्लीकेशन की सीमाओं के परे भी होता है। फिर रियल वर्ल्ड डाटा का स्ट्रक्चर होता है, इनका एक-दूसरे से संबंध होता है और कंस्ट्रेंट्स भी होते हैं। DBMS का उपयोग करके इन फीचर्स को अच्छी तरह रीप्रेजेंट और मैनेज किया जा सकता है। फिर विभिन्न यूजर्स को डाटा निर्मित



to create, access and manipulate the data. The DBMS provides mechanism to achieve these objectives without compromising security and integrity of the data. So, if the data is shared, if it is persistent, if the users want it to be secure and easy to access and manipulate, then use of a DBMS is the best available alternative.

### Applications :

In today's competitive world, application of DBMS is not just limited to a Bank, but it plays a vital role in Stock Exchange, Railway and Airline Ticket Reservation, Schools & the list goes on and on. Not only at this level, database and database management system is required at all levels and areas of an organisations. Also DBMS is required, if you want to -

- Build computerised business applications.
- Create business applications that operate on multiple locations
- Conduct business on the internet.
- Enable customers to place orders using the web.

### Application of DBMS in a Organisation :

- Marketing professional to analyze sales data.
- Human Resource Managers to evaluate employees.
- Operations Manager to track and improve quality.
- Accountant to integrate data across the enterprise.
- Financial analyst to analyze firm's performance.

Some specific applications of DBMS are as follows :

- **Banking** : For customer information, accounts and loans and banking transactions.
- **Airlines** : For reservation and schedule

करने, उसे एक्सेस व मैनियूलेट करने की आवश्यकता होती है। DBMS ऐसी प्रणाली देता है, जो इन लक्ष्यों को सुरक्षित व अचूकता से समझौता किए बिना हासिल कर सके। इसलिए यदि डाटा शेयर होता है, परसिस्टेंट है और यूजर इसे सुरक्षित व एक्सेस करने में आसानी भी चाहता है, तो फिर डाटाबेस मैनेजमेंट प्रणाली सर्वश्रेष्ठ उपलब्ध विकल्प है।

### एप्लीकेशन्स :

आज की प्रतिरचनात्मक दुनिया में DBMS का उपयोग सिर्फ बैंक तक सीमित नहीं है, बल्कि यह शेयर बाजार, रेलवे व हवाई यात्रा, टिकट आरक्षण, स्कूल आदि में भी महत्वपूर्ण भूमिका निभाता है। एप्लीकेशन्स की यह सूची लंबी है। न सिर्फ इस स्तर पर बल्कि डाटाबेस और डाटाबेस मैनेजमेंट सिस्टम की आवश्यकता संस्थानों के सभी स्तर व क्षेत्रों में पड़ती है। DBMS की आवश्यकता आपको तब भी पड़ सकती है, यदि आप -

- ✓ कम्प्यूटरीकृत बिजनेस एप्लीकेशन्स निर्मित करें।
- ✓ ऐसे बिजनेस एप्लीकेशन्स बनाएँ जो एक से अधिक लोकेशन पर ऑपरेट किए जाते हों।
- ✓ इंटरनेट पर व्यवसाय संचालित करते हों।
- ✓ ग्राहकों को वेब साइट्स पर ऑर्डर देने की सुविधा देते हों।

### किसी संस्थान में DBMS के एप्लीकेशन्स :

- विक्रय संबंधी डाटा की मार्केटिंग का लोगों द्वारा विश्लेषण करने के लिए।
- कर्मचारियों का मूल्यांकन करने हेतु मानव संसाधन मैनेजर्स के लिए।
- गुणवत्ता पर नजर रखने और उसे सुधारने हेतु ऑपरेशन मैनेजर्स के लिए।
- पूरे उपक्रम में डाटा इंटीग्रेशन (एकीकरण) हेतु अकाउंटेंट के लिए।
- कंपनी के प्रदर्शन के विश्लेषण हेतु वित्तीय विश्लेषकों के लिए।

### DBMS के कुछ विशिष्ट एप्लीकेशन्स इस प्रकार हैं :

- **बैंकिंग** : ग्राहकों को सूचना देने के लिए, अकाउंट्स (खातों), ऋण आदि बैंक के अन्य व्यवहारों के लिए।
- **एयरलाइंस** : आरक्षण और उड़ान कार्यक्रम की सूचना देने के

information. Airlines were among the first to use databases in geographically distribute.

- **Universities** : For student information, course registration and grades.
- **Credit Card Transactions** : For purchases on credit cards & generation of monthly statements.
- **Telecommunication** : For keeping records of calls made, generating monthly bills, maintaining balances or prepaid calling cards and storing information about the communication network.
- **Finance** : For storing information about holdings, sales and purchases of financial instruments such as stocks and bonds.
- **Sales** : For customer, product and purchase information.
- **Manufacturing** : For management of supply chain & for tracking production of items in factories, inventories of items in stores and orders for items.
- **Human Resources** : For information about employees, salaries payroll taxes & benefits and for generation of pay checks.

The list doesn't stop here besides these uses, DBMS has opened doors for usage of information. DBMS has drastically changed and is continuously changing the way of doing business. There are several new applications, which have become increasingly important in recent years.

Some of these are - Decision Support System, Data Mining, Data Analysis, Data Ware Housing, Information Retrieval System, Distributed Information Systems, etc.

### Data Models :

Data Models are different models that can be used to design a database. Design of database includes describing data, data relationships, data semantics and consistency constraints. Various

लिए। एयरलाइन्स पहले ऐसी संस्थान है, जिन्होंने डाटाबेस का उपयोग भौगोलिक रूप से विभिन्न क्षेत्रों के लिए किया।

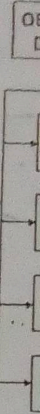
- **विश्वविद्यालय** : विद्यार्थियों को जानकारी देने, पाठ्यक्रम में पंजीयन तथा ग्रेड देने के लिए।
- **क्रेडिट कार्ड व्यवहार** : क्रेडिट कार्ड पर खरीदी और मासिक स्टेटमेंट्स तैयार करने के लिए।
- **दूरसंचार** : किए गए फोन कॉल्स का रेकॉर्ड रखने, मासिक बिल तैयार करने, बैलेंस या प्रीपेड कॉल्स का हिसाब रखने और कम्युनिकेशन नेटवर्क संबंधी जानकारी संग्रहीत करने के लिए।
- **फायनेंस (वित्त)** : शेयर होल्डिंग व इनसे संबंधित जानकारी रखने के लिए। शेयर, ब्रांड आदि जैसे वित्तीय उपकरणों की खरीदी-विक्री की जानकारी रखने के लिए।
- **विक्रय** : ग्राहक, उत्पाद और खरीदी संबंधी सूचना दर्ज करने के लिए।
- **उत्पादन** : आपूर्ति श्रृंखला के प्रबंधन हेतु। कारखानों में उत्पादन आयटमों पर निगरानी हेतु। स्टोर्स में आयटम की इनवेंटरी तथा आयटम के ऑर्डर संबंधी जानकारी रखने हेतु।
- **मानव संसाधन** : कर्मचारियों, वेतन, वेतन-पत्रक, कर और अन्य लाभों की जानकारी तथा वेतन भुगतान संबंधी चेक बनाने के लिए।

ये सूची यही नहीं थमती, DBMS ने सूचना के विविध उपयोग के दरवाजे खोल दिए हैं। DBMS ने व्यवसाय करने के तरीके को आमूलचूल बदल दिया है और यह लगातार इसे और भी परिवर्तित करता जा रहा है। ऐसे कई नई एप्लीकेशन्स हैं, जो हाल ही के वर्षों में उत्तरोत्तर महत्वपूर्ण होते गए हैं।

इनमें से कुछ ये हैं - डिसीजन सपोर्ट सिस्टम, डाटा माईनिंग, डाटा एनालिसिस, डाटा वेयर हाउसिंग, इंफॉर्मेशन रीट्राइवल सिस्टम, डिस्ट्रिब्यूटेड इंफॉर्मेशन सिस्टम्स आदि।

### डाटा मॉडल्स :

डाटा मॉडल ऐसे भिन्न मॉडल होते हैं, जिनका उपयोग डाटाबेस को डिजाइन करने में होता है। डाटा डिजाइन में डाटा का वर्णन, डाटा रिलेशनशीप, डाटा की भाषा विषय जानकारी और कांसिस्टेंसी कन्स्ट्रेंट्स जैसी बातें शामिल होती हैं। विभिन्न



### (A) Object-Bas

Object-base  
describing data  
They are chara  
provide fairly fle  
allow data cons  
There are many  
likely to come. S  
ones are -

- ✓ The entity-relat
- ✓ The object-orie
- ✓ The semantic o
- ✓ The functional

In this boo  
relationship mod  
as representativ  
based logical m  
model, has gai  
design and is wid  
oriented model i  
of the entity-rela  
executable code  
gaining acceptar

information. Airlines were among the first to use databases in geographically distribute.

- **Universities** : For student information, course registration and grades.
- **Credit Card Transactions** : For purchases on credit cards & generation of monthly statements.
- **Telecommunication** : For keeping records of calls made, generating monthly bills, maintaining balances or prepaid calling cards and storing information about the communication network.
- **Finance** : For storing information about holdings, sales and purchases of financial instruments such as stocks and bonds.
- **Sales** : For customer, product and purchase information.
- **Manufacturing** : For management of supply chain & for tracking production of items in factories, inventories of items in stores and orders for items.
- **Human Resources** : For information about employees, salaries payroll taxes & benefits and for generation of pay checks.

The list doesn't stops here besides these uses, DBMS has opened doors for usage of information. DBMS has drastically changed and in continuously changing the way of doing business. There are several new applications, which have become increasingly important in recent years.

Some of these are - Decision Support System, Data Mining, Data Analysis, Data Ware Housing, Information Retrieval System, Distributed Information Systems, etc.

### Data Models :

Data Models are different models that can be used to design a database. Design of database includes describing data, data relationships, data semantics and consistency constraints. Various

लिए। एयरलाइन्स पहले ऐसी संस्थान है, जिन्होंने डाटाबेस का उपयोग भौगोलिक रूप से विभिन्न क्षेत्रों के लिए किया।

- **विश्वविद्यालय** : विद्यार्थियों को जानकारी देने, पाठ्यक्रम में पंजीयन तथा ग्रेड देने के लिए।
- **क्रेडिट कार्ड व्यवहार** : क्रेडिट कार्ड पर खरीदी और मासिक स्टेटमेंट्स तैयार करने के लिए।
- **दूरसंचार** : किए गए फोन कॉल्स का रेकॉर्ड रखने, मासिक बिल तैयार करने, बैलेंस या प्रीपेड कॉल्स का हिसाब रखने और कम्यूनिकेशन नेटवर्क संबंधी जानकारी संग्रहीत करने के लिए।
- **फाइनेंस (वित्त)** : शेयर होल्डिंग व इनसे संबंधित जानकारी रखने के लिए। शेयर, ब्रांड आदि जैसे वित्तीय उपकरणों की खरीदी-बिक्री की जानकारी रखने के लिए।
- **विक्रय** : ग्राहक, उत्पाद और खरीदी संबंधी सूचना दर्ज करने के लिए।
- **उत्पादन** : आपूर्ति श्रृंखला के प्रबंधन हेतु। कारखानों में उत्पादन आयटमों पर निगरानी हेतु। स्टोर्स में आयटम की इनवेंटरी तथा आयटम के ऑर्डर संबंधी जानकारी रखने हेतु।
- **मानव संसाधन** : कर्मचारियों, वेतन, वेतन-पत्रक, कर और अन्य लाभों की जानकारी तथा वेतन भुगतान संबंधी चेक बनाने के लिए।

ये सूची यही नहीं थमती, DBMS ने सूचना के विविध उपयोग के दरवाजे खोल दिए हैं। DBMS ने व्यवसाय करने के तरीके को आमूलचूल बदल दिया है और यह लगातार इसे और भी परिवर्तित करता जा रहा है। ऐसे कई नई एप्लीकेशन्स है, जो हाल ही के वर्षों में उत्तरोत्तर महत्वपूर्ण होते गए हैं।

इनमें से कुछ ये हैं - डिजीजन सपोर्ट सिस्टम, डाटा माईनिंग, डाटा एनालिसिस, डाटा वेयर हाउसिंग, इंफॉर्मेशन रीट्राईवल सिस्टम, डिस्ट्रिब्यूटेड इंफॉर्मेशन सिस्टम्स आदि।

### डाटा मॉडल्स :

डाटा मॉडल ऐसे भिन्न मॉडल होते हैं, जिनका उपयोग डाटाबेस को डिजाइन करने में होता है। डाटा डिजाइन में डाटा का वर्णन, डाटा रिलेशनशीप, डाटा की भाषा विषय जानकारी और कांसिस्टेंसी कन्स्ट्रेंट्स जैसी बातें शामिल होती हैं। विभिन्न

data models have been proposed with different structure & abstractions.

स्ट्रक्चर और अमूर्तताओं (एब्स्ट्रैक्शन्स) के साथ विभिन्न डाटा मॉडल प्रस्तावित किए गए हैं।

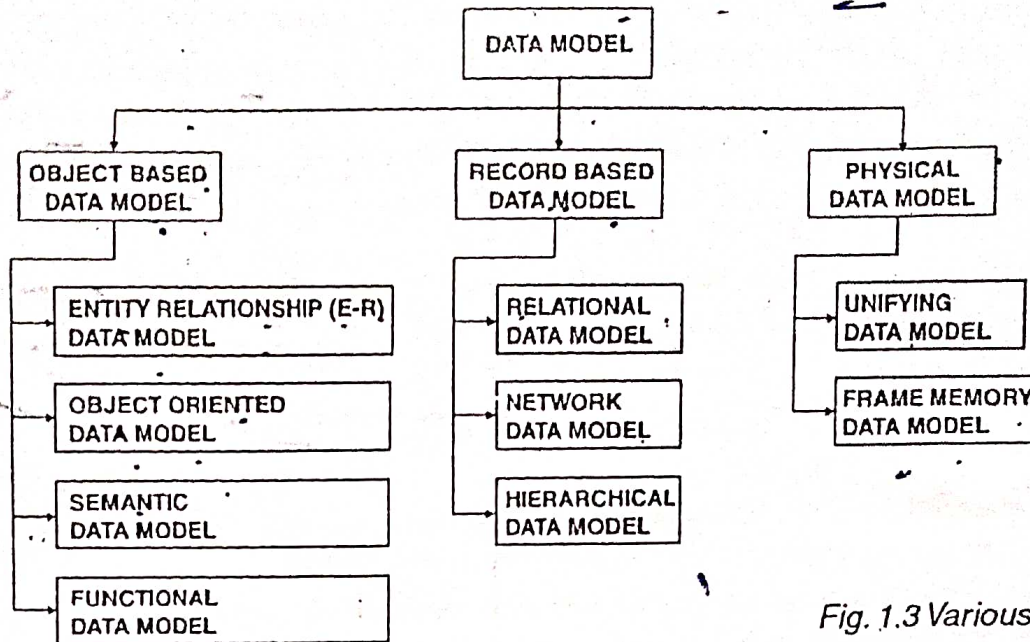


Fig. 1.3 Various Data Model

### (A) Object-Based Data Models :

Object-based logical models are used in describing data at the logical and view levels. They are characterized by the fact that they provide fairly flexible structuring capabilities and allow data constraints to be specified explicitly. There are many different models and more are likely to come. Several of the more widely known ones are -

- ✓ The entity-relationship model
- ✓ The object-oriented model
- ✓ The semantic data model
- ✓ The functional data model

In this book, we examine the entity-relationship model and the object-oriented model as representatives of the class of the object-based logical models. The entity-relationship model, has gained acceptance in database design and is widely used in practice. The object-oriented model includes many of the concepts of the entity-relationship model, but represents executable code as well as data. It is rapidly gaining acceptance in practice.

### (A) ऑब्जेक्ट-आधारित डाटा मॉडल्स :

लॉजिकल व व्यू लेवल पर ऑब्जेक्ट आधारित लॉजिकल मॉडल्स का उपयोग डाटा को डिस्ट्राइब करने के लिए प्रयुक्त होता है। उनकी विशिष्टता इस तथ्य में है कि वे काफी लचीली ढांचागत क्षमता उपलब्ध कराते हैं और डाटा कन्स्ट्रैन्ट्स को स्पष्ट रूप से स्पेसिफॉय करने की सुविधा देते हैं। यहाँ कई भिन्न मॉडल हैं तथा आगे और भी आ रहे हैं। कई व्यापक रूप से जाने पहचाने मॉडल इस प्रकार हैं -

- एंटीटी रिलेशनशीप मॉडल
- ऑब्जेक्ट ओरियेंटेड मॉडल
- सिमेंटिक डाटा मॉडल
- फंक्शनल डाटा मॉडल

इस पुस्तक में हम ऑब्जेक्ट आधारित लॉजिकल मॉडल्स के प्रतिनिधि के रूप में एंटीटी रिलेशनशीप और ऑब्जेक्ट ओरियेंटेड मॉडल का परीक्षण करेंगे। एंटीटी रिलेशनशीप मॉडल ने डाटाबेस डिजाईन में काफी हद तक जगह बना ली है और वह व्यापक रूप से प्रयोग किया जा रहा है। ऑब्जेक्ट ओरियेंटेड मॉडल में एंटीटी रिलेशनशीप मॉडल की कई अवधारणाएँ शामिल हैं, लेकिन यह एक्जीक्यूट किए जा सकने वाले कोड के साथ डाटा को भी प्रदर्शित करता है। व्यवहार में इसकी लोकप्रियता तेजी से बढ़ रही है।

### (1) The Entity-Relationship Data Model :

The entity-relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called entities and of relationships among these objects. An entity is a 'things' or 'object' in the real world that is distinguishable from other objects. Entities are described in a database by a set of attributes. A relationship is an association among several entities. The set of all entities of the same type, and the set of all relationships of the same type termed as entity set and relationship set, respectively.

In addition to entities and relationships, the E-R Model represents certain constraints to which the contents of a database must conform. One important constraint is mapping cardinalities, which express the number of entities to which another entity can be associated via a relationship set.

The overall logical structure of a database can be expressed graphically by an E-R diagram, which is built up from the following components :

- Rectangles, which represent entity sets
- Ellipses, which represent attributes
- Diamonds, which represent relationships among entity sets
- Lines, which link attributes to entity sets and entity sets of relationships

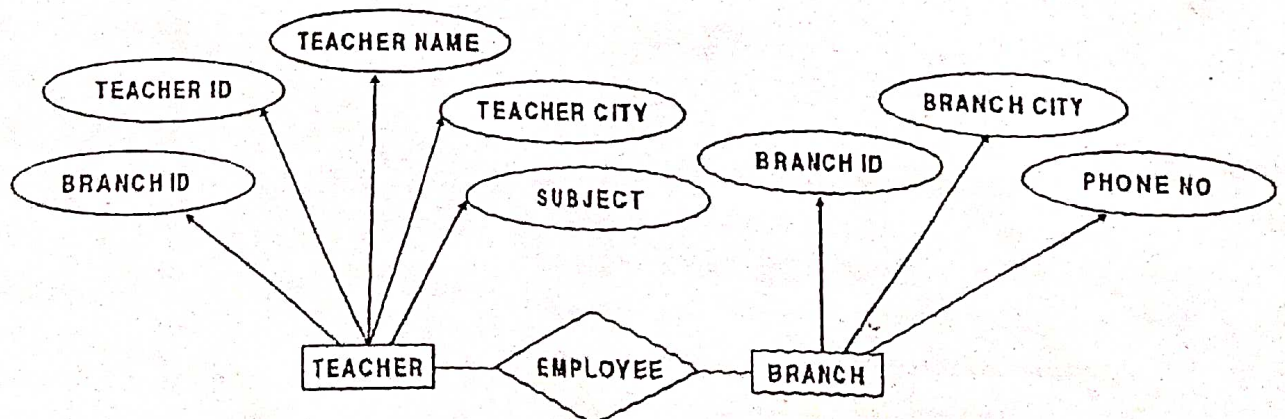


Fig. 1.4 E-R Models

### (1) एंटीटी रिलेशनशीप डेटा मॉडल :

एंटीटी रिलेशनशीप (E-R) डेटा मॉडल उस रीयल वर्ल्ड की सपझ पर आधारित है, जो मूलभूत ऑब्जेक्ट जिन्हें एंटीटीज कहा गया है, के संग्रह से बना हुआ है। इस सपझ में इन ऑब्जेक्ट्स के आपसी संबंध भी आते हैं। एक एंटीटी रीयल वर्ल्ड की ऐसी वस्तु या लक्ष्य है, जो अन्य वस्तुओं से भिन्न है। किसी ऑब्जेक्ट में एंटीटीज को एंटीब्यूट्स के सेट से वर्णित किया जाता है। कई एंटीटीज का आपसी संघ ही रिलेशनशीप कहलाता है। एक ही प्रकार की सभी एंटीटीज के सेट और एक ही प्रकार की रिलेशनशीप के सेट को क्रमशः एंटीटी सेट और रिलेशनशीप सेट कहते हैं।

एंटीटीज और रिलेशनशीप के अलावा, E-R मॉडल कुछ कंस्ट्रेन्ट्स को भी प्रदर्शित करता है, डाटाबेस को बिनके अनुरूप होना चाहिए। एक महत्वपूर्ण कंस्ट्रेन्ट है, मैपिंग कार्डिनलिटीज। एंटीटीज की वह संख्या बताती है, बिनसे रिलेशनशीप सेट के जरिये अन्य एंटीटी जोड़ी जा सकती है।

डाटाबेस का ओवरऑल लॉजिकल ढांचा ग्राफिक तरीके से E-R चित्र के साथ अभिव्यक्त किया जा सकता है, जो निम्न कंपोनेंट्स से बना होता है -

- आयत, जो एंटीटी सेट को दर्शाता है।
- अंडाकार आकृति, जो एंटीब्यूट्स को दर्शाता है।
- समचतुर्भुज, यह एंटीटी सेट्स के बीच संबंधों को प्रदर्शित करता है।
- लाईंस, यह एंटीब्यूट्स को एंटीटी सेट्स और एंटीटी सेट्स की रिलेशनशीप्स से जोड़ता है।

## (2) The Object-Oriented Data Model :

Like the E-R model, the object-oriented model is based on a collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods.

Objects that contain the same types of values and the same methods are grouped together into classes. A class may be a type definition for objects. This combination of data and methods comprising a type definition is similar to a programming language abstract data type.

The only way in which one object can access the data of another object is by invoking a method of that other object. This action is called sending a message to the object. Thus, the call interface of the methods of an object defines that object's externally visible part. The internal part of the object, the instance variables and method code are not visible externally. The result is two levels of data abstraction.

Unlike entities in the E-R model, each object has its own unique identity, independent of the values that it contains. Thus, two objects containing the same values are nevertheless distinct. The distinction among individual objects is maintained in the physical level through the assignment of distinct object identifiers.

## (B) Record-Based Data Model :

Record-based logical models are used in describing data at the logical and view levels. In contrast to object-based data models, they are used both to specify the overall logical structure of the database and to provide a higher-level description of the implementation.

Record-based models are so named because the database is structured in fixed-format records of several types. Each record type

## (2) ऑब्जेक्ट-ओरियंटेड डाटा मॉडल :

E-R मॉडल की तरह, ऑब्जेक्ट-ओरियंटेड मॉडल ऑब्जेक्ट के संग्रह पर आधारित है। ऑब्जेक्ट में इसके इंस्टेंस वेरियेबल्स में संग्रहित वेल्यूज होती हैं। ऑब्जेक्ट में कोड संबंधी वॉडीज होती हैं, जो ऑब्जेक्ट पर ऑपरेट होती है। कोड की इन वॉडीज को मेथड कहते हैं।

ऑब्जेक्ट जिसमें समान प्रकार की वेल्यूज और समान प्रकार की मेथड होती है, को आपस में क्लासेस (श्रेणियों) में ग्रुप किया जाता है। क्लास, ऑब्जेक्ट के लिए एक प्रकार की डिफिनेशन है। टाईप डिफिनेशन से बना डाटा व मेथड का यह काम्पैनिशन प्रोग्रामिंग लैंग्वेज एब्स्ट्रेक्ट डाटा टाईप के समान ही है।

एकमात्र तरीका जिससे एक ऑब्जेक्ट दूसरे ऑब्जेक्ट के डाटा को एक्सेस कर सकता है, वह उस अन्य ऑब्जेक्ट की मेथड को बुलाना (invoke) करना ही है। यह क्रिया ऑब्जेक्ट को मैसेज भेजना कहलाती है। इस प्रकार किसी ऑब्जेक्ट की मेथड का कॉल इंटरफेस उस ऑब्जेक्ट के बाह्य रूप से दर्शनीय हिस्से को डिफाईन करता है। ऑब्जेक्ट का आंतरिक हिस्सा, इंस्टेंस वेरियेबल्स और मेथड कोड बाहरी तौर पर नजर नहीं आते हैं। इसका परिणाम द्विस्तरीय डाटा एब्स्ट्रैक्शन में निकलता है।

E-R मॉडल की एंटीटीज के विपरीत प्रत्येक ऑब्जेक्ट की अपनी सर्वथा भिन्न पहचान होती है, जो इसमें निहित वेल्यूज से स्वतंत्र होती है। इस प्रकार एक ही वेल्यू वाले दो ऑब्जेक्ट बिल्कुल भिन्न होते हैं। डिस्टिंक्ट ऑब्जेक्ट आयडेंटिफायर के असाईनमेंट के द्वारा भौतिक स्तर पर ऑब्जेक्ट्स के बीच भिन्नता बनाए रखी जाती है।

## (B) रेकॉर्ड आधारित डाटा मॉडल :

रेकॉर्ड आधारित लॉजिकल मॉडल का उपयोग लॉजिकल व व्यू लेवल पर डाटा डिस्क्रीब करने के लिए किया जाता है। ऑब्जेक्ट आधारित डाटा मॉडल के विपरीत उनका उपयोग डाटाबेस के ओवरऑल लॉजिकल स्ट्रक्चर को स्पेसिफाई करने और इम्प्लीमेंटेशन का उच्चस्तरीय डिस्क्रीप्शन उपलब्ध कराने, दोनों के लिए होता है।

रेकॉर्ड आधारित मॉडलों को ऐसा नाम इसलिए दिया गया है क्योंकि डाटाबेस को कई प्रकार के फिक्स फॉर्मेट रेकॉर्ड्स में ढाला जाता है। प्रत्येक रेकॉर्ड टाईप तय संस्था में फील्ड या

defines a fixed number of fields, or attributes and each field is usually of a fixed length. The use of fixed-length records simplifies the physical-level implementation of the database. This simplicity is in contrast to many of the object-based models, whose richer structure often leads to variable-length records at the physical level.

The three most widely accepted record-based data models are the relational, network and hierarchical models.

**(1) Relational Data Model :**

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns and each column has a unique name. Figure 1.4 presents a sample relational database comprising of two tables.

Relational database are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in different ways. Another feature of relational systems is that a single database can be spread across several tables. This differs from flat-file database, in which each database is self-contained in a single table.

प्रोसेस को डिजाइन करना है और प्रत्येक फ़ील्ड आमतौर पर निम्न लंबाई का होता है। तब लंबाई वाले रेकॉर्ड का उपयोग डेटाबेस के भौतिक स्तर पर क्रियान्वयन को सरल बना देता है। रेकॉर्ड आधारित मॉडल की यह सरलता कई ऑब्जेक्ट आधारित मॉडल की तुलना में विशेषतामयी लगती है। ऐसा इसलिए है, क्योंकि ऑब्जेक्ट आधारित मॉडल का समूह लोगों, प्रायः भौतिक स्तर पर विभिन्न लंबाई वाले रेकॉर्ड का क्रम करता है।

रिश्तात, नेटवर्क और हायरआर्ची मॉडल वे तीन व्यापक रूप से स्वीकार्य रेकॉर्ड आधारित मॉडल हैं।

**(1) रिलेशनल डेटा मॉडल :**

यह मॉडल डेटा और उन डेटा के बीच संबंधों को व्यक्त करने के लिए टेबल के संग्रह का उपयोग करता है। प्रत्येक टेबल में कई कॉलम होते हैं और प्रत्येक कॉलम का विशिष्ट नाम होता है। चित्र 1.4 में दो टेबल से बने रिलेशनल डेटाबेस का नमूना दिया है।

रिलेशनल डेटाबेस बहुत शक्तिशाली होता है, क्योंकि उन्हें डेटा कैसे संबंधित है या इसे डेटाबेस से कैसे निकाला जाएगा। इस बारे में कुछ मान्यताओं की आवश्यकता होती है। इसका परिणाम यह है कि एक ही डेटाबेस भिन्न तरीके से देखा जा सकता है। रिलेशनल सिस्टम का दूसरा फीचर्स यह है कि एकल डेटाबेस कई टेबलों में फैलाया जा सकता है। यह फ्लैट फ़ाइल डेटाबेस से भिन्न है, जिसमें प्रत्येक डेटाबेस एक ही टेबल में निहित होता है।

Teacher ID	Teacher Name	Teacher City	Phone No.	Subject	Branch ID
101	Ashish	Ahemdabad	216338	ASP	B001
102	Rajesh	Baroda	602973	VB	B002
103	Mittal	Ahemdabad	674555	Oracle	B001
104	Nitin	Udaipur	-	Java	B003
105	Cumar	Indore	240780	VB	B004
106	Abijeet	Aurangabad	-	Networking	B006
107	Devendra	Khandwa	327178	C++	B005
108	Mittal	Ahemdabad	647555	HTMC	B001

Branch ID	Branch City	Phone No.
B001	Ahemdabad	6701717
B002	Baroda	331717
B003	Udaipur	491717
B004	Indore	761717
B005	Khandwa	281717
B006	Aurangabad	451717

Fig. 1.5 Relational Model

## Features of Relational Model :

This model has the following features -

1. All the data is represented in the form of table.
2. Relationship among the data is represented by the column values.
3. It eliminates the need to change application programs when a change is made to the database.
4. User need not know the exact physical structures to use the database.
5. Users are protected from any changes made to these structures.
6. Relational is the only data structure used in the relational data model to represent both entities and the relationships between them.
7. Rows of the relation are referred to as a tuples of the relation and the columns are its attributes.
8. Each attribute of a relation has distinct name.
9. The values for an attribute or a column are drawn from a set of values known as a domain.

## Advantages of Relational Model :

Major advantages of the relational model are-

**1. Structural Independence :** The relational model does not depend on the navigational data access system thus freeing the database designers, programmers and end users from learning the details of data storage. Changes in the database structure do not affect the data access. When it is possible to make change to the database structure without affecting the DBMS's capability to access data, we can say that structural independence have been achieved. So relational database model has structural independence.

**2. Conceptual Simplicity :** The relational data model frees the designer from the physical data

## रिलेशनल मॉडल के गुण :

रिलेशनल मॉडल के निम्न गुण होते हैं -

1. संपूर्ण डाटा, टेबल के रूप में प्रदर्शित किया जाता है।
2. डाटा के बीच रिलेशनशीप को कॉलम वैल्यू से दर्शाया जाता है।
3. यह डाटाबेस में फेरबदल की स्थिति में एप्लीकेशन प्रोग्राम में बदलाव लाने की आवश्यकता को खत्म कर देता है।
4. यूजर को डाटाबेस का उपयोग करने के लिए भौतिक रचना स्वरूप की ठीक-ठीक जानकारी होने की आवश्यकता नहीं होती है।
5. भौतिक रचना स्वरूप में किए गए किसी भी फेरबदल के खिलाफ यूजर्स को संरक्षण मिलता है।
6. रिलेशनल डाटा मॉडल में एंटीटी और उनके बीच रिलेशनशीप दोनों को दर्शाने के लिए केवल रिलेशनल डाटा स्ट्रक्चर का उपयोग किया जाता है।
7. रिलेशन्स की पंक्तियों को रिलेशन के tuples के रूप में और कॉलम को इसके गुणों के रूप में उल्लेखित किया जाता है।
8. रिलेशन का प्रत्येक गुण का अपना विशिष्ट नाम होता है।
9. एंटीब्यूट या कॉलम के लिए वैल्यू, वैल्यूज के सेट में से निकाली जाती है। इस सेट को डोमेन कहते हैं।

## रिलेशनल मॉडल के लाभ :

रिलेशनल मॉडल के प्रमुख लाभ इस प्रकार है -

**1. ढांचागत या रचनात्मक स्वतंत्रता :** रिलेशनल मॉडल नेविगेशनल डाटा एक्सेस सिस्टम पर निर्भर नहीं होता है और इस प्रकार यह डाटाबेस डिजाईनर्स, प्रोग्रामर्स और एंड यूजर्स को डाटा स्टोरेज के ब्यौरे सीखने से मुक्त रखता है। डाटाबेस स्ट्रक्चर में परिवर्तन डाटा एक्सेस को प्रभावित नहीं करता है। जब DBMS's की डाटा को एक्सेस करने की क्षमता को प्रभावित किए बिना डाटाबेस स्ट्रक्चर में परिवर्तन करना संभव हो तो यह कहा जाता है कि ढांचागत (structural) स्वतंत्रता हासिल कर ली गई है। इस प्रकार स्पष्ट है कि रिलेशनल डाटाबेस मॉडल में ढांचागत स्वतंत्रता है।

**2. अवधारणात्मक सरलता :** रिलेशनल डाटा मॉडल, डिजाईनर को भौतिक डाटा के ब्यौरों से मुक्त रखता है। डिजाईनर्स



storage details, the designers can concentrate on the logical view of the database.

**3. Design, Implementation, Maintenance and Usage Ease :** The relational database model achieves both data independence and structural independence making the database design, maintenance, administration and usage much easier than the other models.

**4. Ad Hoc Query Capability :** The presence of very powerful, flexible and easy-to-use query capability is one of the main reasons for the immense popularity of the relational database model. The query language of the relational database models structured query language or SQL makes ad hoc queries a reality. SQL is a fourth generation language (4GL). A 4GL allows the user to specify what must be done without specifying how it must be done. So using SQL, the users can specify what information they want and leave the details of how to get the information to the database. The relational database will perform the task of translating the user queries (specified as SQL statements) into the technical code required to retrieve the requested information.

#### Disadvantages of Relational Model :

The disadvantages are -

**1. Hardware Overheads :** Relational database systems hides the implementation complexities and the physical data storage details from the users. For doing this, i.e., for making things easier for the users, the relational database systems need more powerful hardware computers and data storage devices. So the RDBMS needs powerful machines to run smoothly. But as the processing power of modern computers is increasing at an exponential rate and in today's scenario, the need for more processing power is no longer a very big issue.

डेटाबेस के लॉजिकल व्यू (ए.ए.एल) का उपयोग करने में है।

3. डिजाईन, क्रियान्वयन, रखरखाव और उपयोग में आसानी : रिलेशनल डेटाबेस मॉडल, डेटा व संरचनात्मक स्वतंत्रता, दोनों हासिल करता है, जिसके कारण डेटाबेस डिजाईन, रखरखाव, प्रशासन और उपयोग, अन्य मॉडलों की तुलना में कहीं अधिक आसान हो जाता है।

4. एडहॉक क्वेरी क्षमता : रिलेशनल डेटाबेस मॉडल की बत्रदस्त लोकप्रियता के प्रमुख कारणों में से एक इसकी बहुत शक्तिशाली, लचीली और आसानी से उपयोग में लाई जा सकने वाली क्वेरी कैपेबिलिटी (जानकारी पूछे जा सकने की क्षमता) है। रिलेशनल डेटाबेस मॉडल की क्वेरी लैंग्वेज, स्ट्रक्चर्ड क्वेरी लैंग्वेज अथवा SQL एडहॉक क्वेरी को वास्तविक बना देती है। SQL चौथी पीढ़ी की भाषा (4GL) है। 4GL यूजर को बिना यह स्पष्टिभाव किए कि कैसे करता है, यूजर को क्या करना है, इसे स्पष्टिभाव करने की अनुमति देती है। इस प्रकार SQL के उपयोग से यूजर जो इंफॉर्मेशन वे चाहते हैं, उसे तो स्पष्टिभाव कर सकते हैं और इंफॉर्मेशन कैसे प्राप्त करें, इसके ब्यौरे डेटाबेस पर छोड़ सकते हैं। रिलेशनल डेटाबेस वाली गई इंफॉर्मेशन को रिट्राईव करने के लिए यूजर स्टेमेंट (SQL स्टेटमेंट के रूप में स्पष्टिभाव की गई) को टेक्नीकल कोड में ट्रांसलेट करने का काम करेगा।

#### रिलेशनल मॉडल की खामियाँ :

ये खामियाँ हैं -

1. हार्डवेयर में अतिरिक्त खर्च : रिलेशनल डेटाबेस सिस्टम, क्रियान्वयन संबंधी जटिलताओं और फिजिकल डेटा स्टोरेज ब्यौरे को यूजर्स से छिपाकर रखता है। इसे करने के लिए अर्थात् यूजर्स के लिए चीजें आसान बनाने के लिए रिलेशनल डेटाबेस सिस्टम को अधिक शक्तिशाली हार्डवेयर कम्प्यूटर्स और डेटा स्टोरेज सेवाओं की आवश्यकता होती है। इसलिए RDBMS को सुगमता से चलाने के लिए शक्तिशाली मशीन लगती है। हालांकि आधुनिक कम्प्यूटर्स की प्रोसेसिंग पॉवर घातों की दर से बढ़ रही है और आज के परिदृश्य में अधिक प्रोसेसिंग पॉवर की आवश्यकता अब कोई बड़ा मुद्दा नहीं है।

storage details, the designers can concentrate on the logical view of the database.

**3. Design, Implementation, Maintenance and Usage Ease :** The relational database model achieves both data independence and structural independence making the database design, maintenance, administration and usage much easier than the other models.

**4. Ad Hoc Query Capability :** The presence of very powerful, flexible and easy-to-use query capability is one of the main reasons for the immense popularity of the relational database model. The query language of the relational database models structured query language or SQL makes ad hoc queries a reality. SQL is a fourth generation language (4GL). A 4GL allows the user to specify what must be done without specifying how it must be done. So using SQL, the users can specify what information they want and leave the details of how to get the information to the database. The relational database will perform the task of translating the user queries (specified as SQL statements) into the technical code required to retrieve the requested information.

#### Disadvantages of Relational Model :

The disadvantages are -

**1. Hardware Overheads :** Relational database systems hides the implementation complexities and the physical data storage details from the users. For doing this, i.e., for making things easier for the users, the relational database systems need more powerful hardware computers and data storage devices. So the RDBMS needs powerful machines to run smoothly. But as the processing power of modern computers is increasing at an exponential rate and in today's scenario, the need for more processing power is no longer a very big issue.

डाटाबेस के लॉजिकल व्यू पर अपना ध्यान केंद्रित रख सकते हैं।

3. डिजाईन, क्रियान्वयन, रखरखाव और उपयोग में आसानी : रिलेशनल डाटाबेस मॉडल, डाटा व हांचागत स्वतंत्रता, दोनों हांसिल करता है, जिसके कारण डाटाबेस डिजाईन, रखरखाव, प्रशासन और उपयोग, अन्य मॉडलों की तुलना में कहीं अधिक आसान हो जाता है।

4. एडहॉक क्वेरी क्षमता : रिलेशनल डाटाबेस मॉडल की जबरदस्त लोकप्रियता के प्रमुख कारणों में से एक इसकी बहुत शक्तिशाली, लचीली और आसानी से उपयोग में लाई जा सकने वाली क्वेरी कैपेबिलिटी (जानकारी पूछे जा सकने की क्षमता) है। रिलेशनल डाटाबेस मॉडल की क्वेरी लैंग्वेज, स्ट्रक्चर्ड क्वेरी लैंग्वेज अथवा SQL एडहॉक क्वेरी को वास्तविक बना देती है। SQL चौथी पीढ़ी की भाषा (4GL) है। 4GL यूजर को बिना यह स्पेसिफाय किए कि कैसे करता है, यूजर को क्या करना है, इसे स्पेसिफाय करने की अनुमति देती है। इस प्रकार SQL के उपयोग से यूजर जो इंफॉर्मेशन वे चाहते हैं, उसे तो स्पेसिफाय कर सकते हैं और इंफॉर्मेशन कैसे प्राप्त करें, इसके ब्यौरे डाटाबेस पर छोड़ सकते हैं। रिलेशनल डाटाबेस चाही गई इंफॉर्मेशन को रिट्राईव करने के लिए यूजर क्वेरीज (SQL स्टेटमेंट के रूप में स्पेसिफाय की गई) को टेक्नीकल कोड में ट्रांसलेट करने का काम करेगा।

#### रिलेशनल मॉडल की खामियाँ :

ये खामियाँ हैं -

1. हार्डवेयर में अतिरिक्त खर्च : रिलेशनल डाटाबेस सिस्टम, क्रियान्वयन संबंधी जटिलताओं और फिजिकल डाटा स्टोरेज ब्यौरे को यूजर्स से छिपाकर रखता है। इसे करने के लिए अर्थात् यूजर्स के लिए चीजें आसान बनाने के लिए रिलेशनल डाटाबेस सिस्टम को अधिक शक्तिशाली हार्डवेयर कम्प्यूटर्स और डाटा स्टोरेज सेवाओं की आवश्यकता होती है। इसलिए RDBMS को सुगमता से चलने के लिए शक्तिशाली मशीन लगती है। हालांकि आधुनिक कम्प्यूटर्स की प्रोसेसिंग पॉवर घातों की दर से बढ़ रही है और आज के परिदृश्य में अधिक प्रोसेसिंग पॉवर की आवश्यकता अब कोई बड़ा मुद्दा नहीं है।

**2. Ease of Design can lead to bad design :** The relational database is an easy-to-design and use system. The users need not know the complex details of physical data storage. They need not know how the data is actually stored to access it. This case of design and use can lead to the development and implementation of very poorly system designed database management systems. Since the database is efficient, these designs to inefficiencies will not come to light when the database is designed and when there is only a small amount of data. As the database grows, the poorly designed databases will slow the system down and will result in performance degradation and data corruption.

**(2) Network Data Model :**

Data in the network model are represented by collections of records and relationships among data are represented by links, which can be viewed as pointers. The records in the database are organized as collections of arbitrary graphs. Figure 1.5 presents a sample network database.

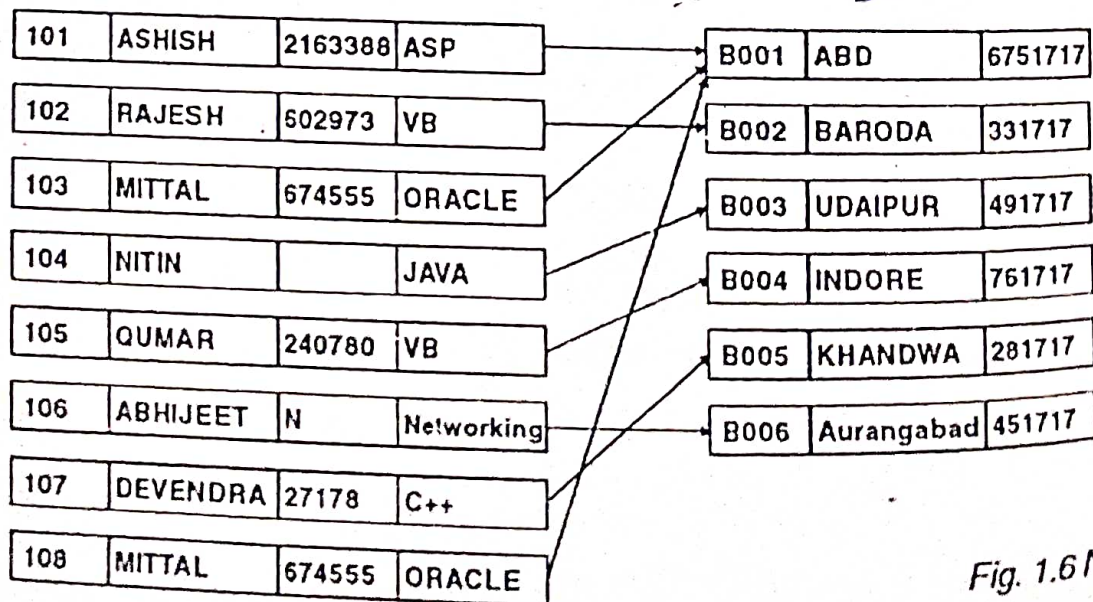


Fig. 1.6 Network Model

**Features of the Network Model :**

Network model is has the following features -

1. Data is represented by records and links.
2. Relationship among the data is represented by links or pointers

2. डिजाइन की आसानी खराब डिजाइन को जन्म दे सकती है : रिलेशनल डाटाबेस आसानी से डिजाइन और उपयोग किया जा सकने वाला सिस्टम है। यूजर्स को फिजिकल डाटा स्टोरेज के जटिल ब्यौरे जानने की आवश्यकता नहीं है। उन्हें यह भी जानने की आवश्यकता नहीं है कि डाटा, एक्सेस करने के लिए वास्तव में कैसे स्टोर किया गया है। डाटा डिजाइन का वह केस व. उसका प्रयोग इसे बहुत ही घटिया प्रणाली के विकास एवं प्रयोग की ओर ले जाता है, जिसे डिजाईड डाटाबेस मैनेजमेंट प्रणाली कहा जाता है। चूंकि डाटाबेस कार्यक्षम होता है, जब डाटाबेस डिजाईन किया गया था तथा जब डाटा का अत्यंत सूक्ष्म भाग ही होता है, तब ये डिजाईनिंग की खामियाँ प्रकट नहीं होती हैं। जैसे ही डाटाबेस बढ़ता है, घटिया किस्म के बने हुए डाटा बेसेस, प्रणाली की गति को धीमा कर देंगे तथा उसका परिणाम होगा कार्य की निम्नतर श्रेणी तथा डाटा कर्षण।

**(2) नेटवर्क डाटा मॉडल :**

नेटवर्क मॉडल में डाटा में रेकॉर्ड और रिलेशनशीप के संग्रह को लिंक से दर्शाया जाता है और इन्हें पाईटर्स के रूप में देखा जा सकता है। डाटाबेस में रेकॉर्ड को मनमाने ग्राफ के संग्रह के रूप में ऑर्गेनाईज किया जाता है। चित्र 1.5 में नमूने के तौर पर नेटवर्क डाटाबेस दर्शाया गया है।

**नेटवर्क मॉडल के गुण :**

नेटवर्क मॉडल के निम्न गुण होते हैं -

1. डाटा को रेकॉर्ड और लिंक द्वारा दर्शाया जाता है।
2. डाटा के बीच रिलेशनशीप को लिंक और पाईटर्स से दर्शाया किया जाता है।

3. All the records are organized in arbitrary form.
4. A set is used to represent a directed relationship between two records and these record are known as owner records and member records.
5. The set has one to many (1:M) relationship between the owner and member record.
6. The set can have more than one member record but only one owner record.

#### Advantages of Network Data Model :

- (1) **Conceptual Simplicity** : Just like the hierarchical model, the network model is also conceptually simple and easy to design.
- (2) **Capability to handle more relationships types** : The network model can handle the one-to-many (1:n) and many-to-many (n:n) relationships, which is a real help in modeling the real life situations.
- (3) **Ease to data access** : The data access is easier than and flexible than in the hierarchical model. An application can access an owner record and all the member records within a set and if one member in the set has two owners (like the employee working for two departments), then one can move from one owner to another.
- (4) **Data Integrity** : The network model does not allow a member to exist without an owner. Thus a user must first define the owner record and then the member record. This ensures the data integrity.
- (5) **Data Independence** : The network model is better in isolating (at least partially if not fully) the programs from the complex physical storage details. This, to a certain extent, ensures that the changes in data characteristics do not require changes to the application programs.
- (6) **Database Standards** : The network model is based on the standards formulated by the DBTG

3. सभी रेकॉर्ड को बिना किसी नियम वाले arbitrary फॉर्म में ऑर्गेनाइज किया जाता है।
4. दो रेकॉर्ड के बीच निर्देशित संबंधों को दर्शाने के लिए सेट का उपयोग किया जाता है और ये ऑनर रेकॉर्ड्स तथा मेम्बर रेकॉर्ड्स के रूप में जाने जाते हैं।
5. सेट में ऑनर और मेम्बर रेकॉर्ड के बीच एक से कई (1:M) रिलेशनशीप है।
6. सेट में मेम्बर रेकॉर्ड तो एक से ज्यादा हो सकते हैं, लेकिन ऑनर रेकॉर्ड एक ही होगा।

#### नेटवर्क डाटा मॉडल के लाभ :

- (1) **अवधारणात्मक सरलता** : हायर्कीकल मॉडल की तरह नेटवर्क मॉडल भी अवधारणा के स्तर पर बहुत सरल और डिजाइन करने में बहुत आसान है।
- (2) **अधिक रिलेशनशीप टाईप को हैंडल करने में सक्षम** : नेटवर्क मॉडल एक से कई (1:n) और कई से कई (n:n) रिलेशनशीप को संभाल सकता है। वास्तविक जीवन की स्थितियों को मॉडल करने में यह वास्तविक मदद है।
- (3) **डाटा एक्सेस में आसानी** : हायर्कीकल मॉडल की तुलना में डाटा एक्सेस आसान और लचीला है। कोई भी एप्लीकेशन ऑनर रेकॉर्ड एक्सेस कर सकती है। इसी प्रकार एक सेट के भीतर सभी मेम्बर रेकॉर्ड भी एक्सेस कर सकता है। यदि सेट में किसी मेम्बर के दो ऑनर है। (जैसे दो विभागों के लिए कोई एक कर्मचारी काम करे) तो एक्सेस चाहने वाला एक से दूसरे ऑनर पर जा सकता है।
- (4) **डाटा शुद्धता** : नेटवर्क मॉडल किसी भी मेम्बर को बिना ऑनर के नहीं रहने देता है। इस प्रकार यूजर को पहले ऑनर रेकॉर्ड परिभाषित करना चाहिए और उसके बाद मेम्बर रेकॉर्ड। इससे डाटा की शुद्धता बनी रहती है।
- (5) **डाटा की स्वतंत्रता** : नेटवर्क मॉडल जटिल फिजिकल स्टोरेज व्यूरी से प्रोग्राम को अलग करने (पूरी तरह से नहीं तो आंशिक रूप से ही सही) के लिए अच्छा है। यह कुछ हद तक यह सुनिश्चित करता है कि डाटा कंटेक्टरस्टिक्स में परिवर्तन के कारण एप्लीकेशन प्रोग्राम में फेरबदल की आवश्यकता न हो।
- (6) **डाटाबेस स्टैंडर्ड्स** : नेटवर्क मॉडल DBTG (Database Task Group of CODASYL

(Database Task Group of CODASYL Committee) and augmented by ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee) in the 1970s. All the network database management systems conformed to these standards. These standards included a data definition language (DDL) and a data manipulation language (DML), thus greatly enhancing database administration and portability.

### Disadvantages of Network Model :

The disadvantages of the network model are-

(1) **System Complexity** : The network model provides a navigational access to the data in which the data are accessed one record at a time. This navigational data access mechanism makes the system implementation very complex and consequently the database administrators, database designers, programmers and even the end users should be familiar with the internal data structures in order to access the data.

(2) **Absence of Structural Independence** : Since the data access method in the network database model is a navigational system, making structural changes to the database very difficult in most cases and impossible in some cases. If changes are made to the database structure then all the application programs need to be modified before they can access data. Thus, even though the network database model succeeds in achieving data independence, it still fails to achieve structural independence.

(3) **Hierarchical Data Model** : The hierarchical model is similar to the network model in the sense that data and relationships among data are represented by records and links, respectively. It differs from the network model in that the records are organized as collections of trees rather than arbitrary graphs. Figure 1.6 represents a sample hierarchical database.

Committee) द्वारा फार्मूलेट और ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee) ने 1970 में ऑगमेंट किया था। सभी नेटवर्क डाटाबेस मैनेजमेंट सिस्टम इन स्टैंडर्ड के अनुरूप होते हैं। इन स्टैंडर्ड में एक डाटा डेफिनेशन लैंग्वेज (DDL) और एक डाटा मैनिप्यूलेशन लैंग्वेज (DML) होती है। इस प्रकार यह डाटाबेस एडमिनिस्ट्रेशन और पोर्टेबिलिटी को बहुत बढ़ा देता है।

### नेटवर्क मॉडल की खामियाँ :

नेटवर्क मॉडल के मुख्य दोष निम्न हैं -

(1) **सिस्टम कॉम्प्लेक्सिटी** : नेटवर्क मॉडल डाटा को नेवीगेशनल एक्सेस की सुविधा देता है, जिसमें डाटा एक समय में एक रिकॉर्ड को एक्सेस करता है। यह नेवीगेशनल डाटा एक्सेस मेकेनिज्म, सिस्टम इम्प्लीमेंटेशन को बहुत जटिल बनाता है तथा डाटाबेस एडमिनिस्ट्रेटर, डाटाबेस डिजाईनर, प्रोग्रामर्स तथा एंड यूजर्स को डाटा को एक्सेस करने के लिए इंटरनल डाटा स्ट्रक्चर से फेमेलियर होना ही चाहिए।

(2) **स्ट्रक्चरल स्वतंत्रता की अनुपस्थिति** : चूंकि डाटाबेस नेटवर्क मॉडल में डाटा एक्सेस करने की विधि नेवीगेशनल सिस्टम है, तो अधिकतर स्थितियों में डाटाबेस पर कोई भी संरचनात्मक परिवर्तन कठिन होता है तथा कुछ स्थितियों में यह असंभव होता है। यदि डाटाबेस संरचना में परिवर्तन किए जाते हैं, तो सभी एप्लीकेशन प्रोग्रामों को डाटा एक्सेस करने के पहले संशोधित करने की आवश्यकता होती है। इस तरह, यदि नेटवर्क डाटाबेस मॉडल डाटा स्वतंत्रता प्राप्त करने में सफल हो जाता है, तो भी यह संरचनात्मक स्वतंत्रता प्राप्त करने में असफल रहता है।

(3) **हायरार्कीकल डाटा मॉडल** : हायरार्कीकल मॉडल, नेटवर्क मॉडल की तरह ही इन अर्थों में समान है कि डाटा तथा रिलेशनशीप को क्रमशः रेकॉर्ड्स लिंक द्वारा प्रदर्शित किया जाता है। नेटवर्क मॉडल से यह इस अर्थ में भिन्न है कि रेकॉर्ड को आर्बिट्ररी ग्राफ के बजाय ट्रेस में ऑर्गेनाइज किया जाता है। चित्र 1.6 में हायरार्कीकल डाटाबेस का नमूना दर्शाया गया है।

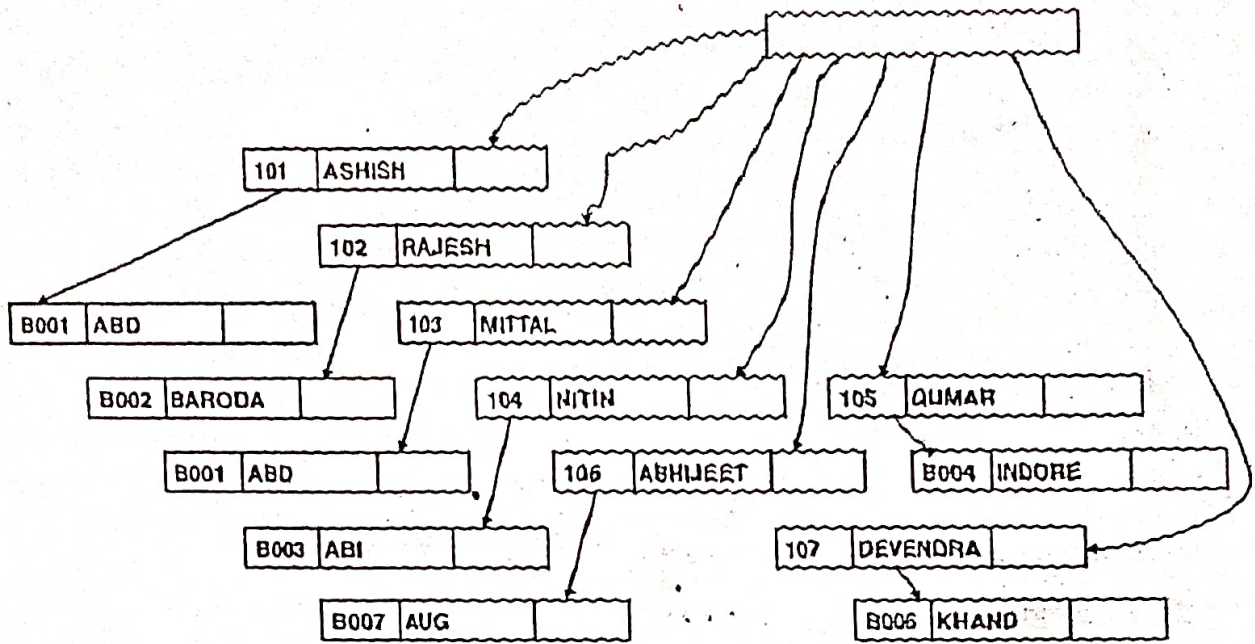


Fig. 1.7 Hierarchical Model

Hierarchical Database model is one of the oldest database models, dating from late 1950s. One of the first hierarchical databases Information Management System (IMS) was developed jointly by North American Rockwell Company and IBM. IMS became the world's leading mainframe hierarchical database system in the 1970s and early 1980s. The hierarchical model assumes that a tree structure is the most frequently occurring relationships. This assumption is recognized today as misleading. In fact many of the limitations and shortcomings of the hierarchical model result from this overly restrictive view of relationships.

### Features of Hierarchical Model :

Hierarchical model has the following features-

1. It is very similar to network model except the records are organized in tree form.
2. Hierarchy is an ordered tree and is easy to understand.
3. Again data is represented in the form of records.
4. Similarly relationship among the data is represented by records or links.

हायरकीकल डाटाबेस मॉडल सबसे पुराने डाटाबेस मॉडल में से है। यह 1950 के दशक के उत्तरार्ध से अस्तित्व में है। पहले हायरकीकल डाटाबेस इंफॉर्मेशन मैनेजमेंट सिस्टम (IMS) में से एक नॉर्थ अमेरिकन रॉकवेल कंपनी और IBM ने संयुक्त रूप से विकसित किया था। 1970 व 1980 के दशक के प्रारंभ में IMS दुनिया का अग्रणी मेनफ्रेम हायरकीकल डाटाबेस सिस्टम बन गया। हायरकीकल मॉडल में यह मानकर चला जाता है कि रिलेशनशीप सबसे अधिक ट्री स्ट्रक्चर में ही होती है। आज यह अवधारणा गलत मानी जाती है। तथ्य तो यह है कि हायरकीकल मॉडल की कई सीमाएँ व खामियाँ रिलेशनशीप को लेकर इस सीमित दृष्टिकोण का ही परिणाम है।

### हायरकीकल मॉडल के गुण :

हायरकीकल मॉडल में निम्न गुण हैं -

1. यह बहुत कुछ नेटवर्क मॉडल से मिलता-जुलता है। अंतर इतना ही है कि इसमें रेकॉर्ड्स ट्री फॉर्म में ऑर्गेनाइज किए गए हैं।
2. हायरकी एक सुव्यवस्थित ट्री है, जो समझने में आसान है।
3. फिर डाटा भी रेकॉर्ड्स के रूप में दर्शाया जाता है।
4. इसी प्रकार डाटा के बीच रिलेशनशीप रेकॉर्ड अथवा लिंक से दर्शाया जाता है।

5. A tree may be defined as a set of nodes. Such that there is one specially designated node called the root.
6. At the root of the tree is the single parent, the parent can have none, one or more children.
7. In hierarchical model no dependent record can occur without its parent record occurrence.
8. No dependent record occurrence may be connected to more than one parent record occurrence.

### Advantages of Hierarchical Data Model:

The advantages of this database model are -

1. **Simplicity** : Since the database is based on the hierarchical structure, the relationship between the various layers is logically (conceptually) simple. Thus the design of a hierarchical database is simple.
2. **Data Security** : Hierarchical model was the first database model that offered the data security that is provided and enforced by the DBMS.
3. **Data Integrity** : Since the hierarchical model is based on the parent/child relationships, there is always a link between the parent segment and the child segments under it. The child segments are always automatically referenced to its parent, this model promotes data integrity.
4. **Efficiency** : The hierarchical database model is a very efficient one when the database contains a large number of 1:n relationships (one-to-many relationships) and when the users require large number of transactions, using data whose relationships are fixed.

### Disadvantages of Hierarchical Data Model :

The limitations of the hierarchical model are -

1. **Implementation Complexity** : Although the hierarchical database model is conceptually simple and easy to design it is quite complex to implement. The database designers should have

5. ट्री को नोड्स के सेट के रूप में डिफाइन किया जा सकता है, जैसे इसमें एक विशेषतौर पर निर्दिष्ट कोड होता है, जिसे root कहते हैं।
6. ट्री के सेट में एकमात्र पैरेंट होता है। पैरेंट के कोड में एक या अधिक बच्चे हो सकते हैं।
7. हायरकीकल मॉडल में पैरेंट रेकॉर्ड आकरंस के बिना कोई भी डिपेंडेंट रेकॉर्ड नहीं हो सकता।
8. कोई भी डिपेंडेंट रेकॉर्ड आकरंस एक से अधिक पैरेंट रेकॉर्ड आकरंस से कनेक्ट नहीं किया जा सकता।

### हायरकीकल डाटा मॉडल के लाभ :

इस डाटाबेस मॉडल के मुख्य लाभ यह हैं -

1. **सरलता** : चूंकि डाटाबेस हायरकीकल स्ट्रक्चर पर आधारित है, विभिन्न लेयर्स के बीच संबंध लॉजिकली (अवधारणात्मक रूप से) सरल है। इस तरह हायरकीकल डाटाबेस का डिजाईन सरल है।
2. **डाटा सुरक्षा** : हायरकीकल मॉडल पहला ऐसा डाटाबेस मॉडल था जिसमें डाटा सुरक्षा दी गई थी और इसे DBMS में लागू किया था।
3. **डाटा शुद्धता** : चूंकि हायरकीकल मॉडल पैरेंट/चाइल्ड (पालक/बालक) संबंध पर आधारित है, इसमें हमेशा पैरेंट और चाइल्ड सेगमेंट में लिंक बनी रहती है। चाइल्ड सेगमेंट हमेशा स्वचालित तरीके से अपने पैरेंट सेगमेंट को रेफरेंस करते हैं। इस तरह यह मॉडल डाटा इंटीग्रिटी को बढ़ाता है।
4. **कार्यक्षमता** : हायरकीकल डाटाबेस मॉडल बहुत ही कार्यक्षम साबित होता है, जब डाटाबेस में बड़ी संख्या में 1:n रिलेशनशिप (एक-से-कई) हो और यूजर को फिक्स रिलेशनशिप वाले डाटा का उपयोग करके बहुत सारे ट्रांजेक्शन की आवश्यकता हो।

### हायरकीकल डाटा मॉडल की खामियाँ :

हायरकीकल मॉडल की मुख्य खामियाँ इस प्रकार हैं -

1. **क्रियान्वयन की जटिलता** : हालांकि हायरकीकल डाटाबेस मॉडल अवधारणात्मक रूप से सरल है और डिजाईन में भी आसान है, लेकिन अमल में लाने में काफी जटिल है।

very good knowledge of the physical data storage characteristics.

**2. Database Management Problems :** If you make any changes in the database structure of a hierarchical database, then you need to make the necessary changes in all the applications programs that access the database. Thus maintaining the database and the applications can become very difficult.

**3. Lack of Structural Independence :** Structural independence exists when the changes to the database structure does not affect the DBMS's ability to access data. Hierarchical database systems use physical storage paths to navigate to the different data segments. So the applications programmer should have a good knowledge of the relevant access paths to access the data. So if the physical structure is changed the applications will also have to be modified. Thus in a hierarchical database the benefits of data independence is limited by structural dependence.

**4. Programming Complexity :** Due to the structural dependence and the navigational structure, the application programmers and the end users must know precisely how the data is distributed physically in the database in order to access data. This requires knowledge of complex pointer systems, which is often beyond the grasp or ordinary users (users who have little or no programming knowledge).

**5. Implementation Limitation :** Many of the common relationships do not conform to the 1:n format required by the hierarchical model. The many-to-many (n:n) relationships, which are more common in real life are very difficult to implement in a hierarchical model.

### (C) Physical Data Models :

Physical data models are used to describe data at the lowest level. In contrast to logical

डाटाबेस डिजाईनर्स को फिजिकल डाटा स्टोरेज का बहुत अच्छा ज्ञान होना चाहिए।

**2. डाटाबेस मैनेजमेंट की समस्याएँ :** यदि आप किसी हायरकीकल डाटाबेस के डाटाबेस स्ट्रक्चर में परिवर्तन करें, तो फिर आप को डाटाबेस को एक्सेस करने वाले सारे एप्लीकेशन्स प्रोग्राम में आवश्यक बदलाव करने होंगे। इस प्रकार डाटाबेस और एप्लीकेशन्स का रखरखाव बहुत कठिन हो जाता है।

**3. स्ट्रक्चरल स्वतंत्रता का अभाव :** स्ट्रक्चर या आधारभूत ढांचे संबंधी स्वतंत्रता तब होती है, जब डाटाबेस स्ट्रक्चर में परिवर्तन से DBM की डाटा एक्सेस करने की क्षमता प्रभावित नहीं होती है। हायरकीकल डाटाबेस सिस्टम विभिन्न डाटाबेस सिस्टम में नेवीगेट करने के लिए फिजिकल स्टोरेज पॉथ का उपयोग करता है। इस वजह से डाटा एक्सेस करने के लिए एप्लीकेशन्स प्रोग्रामर को प्रासांगिक एक्सेस पॉथ का अच्छा ज्ञान होना चाहिए। ऐसी स्थिति में यदि भौतिक ढांचा बदला जाता है, तो एप्लीकेशन्स भी रूपांतरिक करना होंगे। इस प्रकार हायरकीकल डाटाबेस में मूलभूत रचनागत निर्भरता के कारण डाटा स्वतंत्रता के फायदे सीमित ही हैं।

**4. प्रोग्रामिंग संबंधी जटिलताएँ :** रचनागत निर्भरता और नेविगेशनल स्ट्रक्चर के कारण डाटा एक्सेस करने के लिए एप्लीकेशन प्रोग्रामर तथा एंड यूजर को ठीक-ठीक इस बात की जानकारी होना चाहिए कि डाटाबेस में डाटा भौतिक रूप से किस प्रकार विपरीत हैं। इसके लिए जटिल पाईटर सिस्टम का ज्ञान जरूरी है, जो प्रायः सामान्य उपयोगकर्ताओं (जिन्हें प्रोग्रामिंग का थोड़ा या बिल्कुल ही ज्ञान नहीं है) के बस के बाहर की बात है।

**5. क्रियान्वयन की सीमाएँ :** आम रिलेशनशीप में से कई 1:n फॉर्मेट के अनुरूप नहीं होती जो हायरकीकल मॉडल की आवश्यकता है। वास्तविक जीवन में अधिक आम कई से कई (n:n) रिलेशनशीप को हायरकीकल मॉडल में लागू करना बहुत कठिन है।

### (C) फिजिकल डाटा मॉडल :

फिजिकल डाटा मॉडल का उपयोग निम्नतम स्तर पर डाटा वर्णित करने के लिए किया जाता है। लॉजिकल डाटा मॉडल के



data models, there are few physical data models in use. Two of the widely known ones are the unifying model and the frame memory model.

विपरीत उपयोग में लाए जा रहे फिजिकल डाटा मॉडल बहुत थोड़े हैं। व्यापक रूप से फिजिकल मॉडल में दो मॉडल यूनैफाइंग मॉडल और फ्रेम मेमोरी मॉडल हैं।

**Comparison between the Hierarchical, Network & Relational Database Models:**

हायरकीकल, नेटवर्क और रिलेशनल डाटाबेस मॉडल में तुलना :

Model	Data Element Organization	Relationship Organization	Identity	Access Language
Hierarchical	Files, Records	Logical Proximity in a Linearized tree	Record based	Procedural
Network	Files, Records	Intersecting Networks	Record based	Procedural
Relational	Tables	Identifiers of rows in one table are embedded as attribute values in another table	Value based	Non-procedural

**Database Languages :**

डाटाबेस लैंग्वेजेस :

**(1) Data Definition Language (DDL) :**

(1) डाटा डेफिनेशन लैंग्वेज (DDL) :

The language used to define database is known as Data Definition Language (DDL). The commands used for defining database are known as DDL command. DDL includes all commands used for altering tables, dropping tables and defining constraints.

डाटाबेस को डिफाईन करने के लिए उपयोग में लाई जा रही लैंग्वेज डाटा डेफिनेशन लैंग्वेज (DDL) के रूप में जानी जाती है। डाटाबेस को डिफाईन करने के लिए उपयोग में लाई जाने वाली कमांड्स DDL कमांड्स कहलाती हैं। DDL में टेबल बदलने, टेबल ड्रॉप करने या कन्स्ट्रैन्स डिफाईन करने वाली सभी कमांड्स शामिल हैं।

**(2) Data Manipulation Language (DML) :**

(2) डाटा मेनिप्यूलेशन लैंग्वेज (DML) :

The language that enable users to access or manipulate data is known as Data Manipulation Language. The commands used for accessing and manipulating data are known as DML commands. DML includes commands for insertion, deletion and updation besides retrieval.

वह लैंग्वेज जो यूजर को डाटा एक्सेस या मेनिप्यूलेट करने देती है, डाटा मेनिप्यूलेशन लैंग्वेज कहलाती है। डाटा एक्सेस या मेनिप्यूलेट करने के लिए उपयोग में लाई जा रही कमांड्स DML कमांड्स कहलाती हैं। DML में रिट्राईवल के अलावा इंसर्शन, डिलिशन और अपडेशन कमांड्स शामिल होती हैं। DML मूलतः दो प्रकार की होती है -

DMLs are basically of two types :

(i) **Procedural DMLs :** It requires user to specify what data are needed and how to get those data. The codes generated by procedural DMLs is efficient.

(i) **प्रोसीजरल DMLs :** इसमें यूजर को यह स्पेसिफाई करना आवश्यक है कि किस डाटा की आवश्यकता है, कैसे उसे प्राप्त किया जा सकता है। प्रोसीजरल DMLs द्वारा जनित कोड दक्ष होते हैं।

(ii) **Non-procedural DMLs :** It requires user only to specify what data are needed. In non-

(ii) **नॉन-प्रोसीजरल DMLs :** इसमें यूजर को केवल यह बताना होता है कि किस डाटा की आवश्यकता है। नॉन-प्रोसीजरल

procedural DMLs the user does not specify how to get those data. The code generated by non-procedural DMLs is less efficient as compared to procedural DMLs.

### (3) Data Control Language (DCL) :

The language used to control database is known as Data Control Language (DCL). The commands used to control a database are known as DCL Commands. DCL controls the database by administrating previlages & committing (saving) of data.

### (4) Transaction Control Language (TCL):

The process of performing operation over the database is known as transaction. When ever certain transaction is performed over database then it must be performed under the control. The language used to perform such operation is known as TCL.

### Database Abstraction :

As we know that a collection of interrelated files and a set of programs that allow users to access and modify these files is known as a database management system. A major purpose of a database system is to provide the users only that much information that is required by them. This means that the system does not disclose all the details of data, rather it hides certain details of how the data is stored and maintained. A good database system ensures easy, smooth and efficient data structures in such a way so that every type of database user; end user, application system analyst and physical storage system analyst, is able to access its desired information efficiently. (An end user is a person who is not a computer trained person but uses the database to retrieve some information. For example, in a bank database, a customer, who wants to know how much balance remains in his account, is an end-user. An application system analyst is the one who is concerned

DMLs में, यूजर यह स्पेसिफाई नहीं करता है कि डाटा कैसे प्राप्त होगा। नॉन प्रोसीजरल DMLs द्वारा जनित कोड, प्रोसीजरल DMLs की तुलना में कम दक्ष होते हैं।

### (3) डाटा कंट्रोल लैंग्वेज (DCL) :

डाटाबेस कंट्रोल करने के लिए उपयोग की जाने वाली लैंग्वेज को डाटा कंट्रोल लैंग्वेज (DCL) कहते हैं। डाटाबेस को कंट्रोल करने वाली कमांड्स DCL कमांड्स कहलाती हैं। DCL, विशेषाधिकार के उपयोग और डाटा कमिटिंग (सेविंग) डाटाबेस को कंट्रोल करता है।

### (4) ट्रांजेक्शन कंट्रोल लैंग्वेज (TCL) :

डाटाबेस पर ऑपरेशन निष्पादन की प्रक्रिया को ट्रांजेक्शन कहते हैं। जब भी डाटाबेस पर एक निश्चित ट्रांजेक्शन निष्पादित होता है, तो यह निष्पादन कंट्रोल के अंतर्गत होना चाहिए। इस ऑपरेशन को निष्पादित करने के लिए जो लैंग्वेज प्रयोग में लाई जाती है, उसे TCL के नाम से जाना जाता है।

### डाटाबेस एब्स्ट्रैक्शन :

जैसा कि हम जानते हैं, आपस में संबंधित फाइलों के संग्रह और प्रोग्रामर्स का वह सेट, जो यूजर्स को ये फाइलें एक्सेस व मॉडिफाई करने देती है। वे डाटाबेस मैनेजमेंट सिस्टम कहलाते हैं। डाटाबेस सिस्टम का प्रमुख उद्देश्य यूजर को सिर्फ उतनी इंफॉर्मेशन उपलब्ध कराना है, जिनकी उन्हें आवश्यकता है। इसका मतलब है कि सिस्टम डाटा के सारे ब्यौरे उजागर नहीं करता है। इसके बजाय डाटा कैसे स्टोर किया जाता है और इसका रख-रखाव कैसे होता है, इसके ब्यौरे छिपाता है। एक अच्छा डाटाबेस सिस्टम आसन, सुगम और सक्षम डाटा स्ट्रक्चर इस तरह से सुनिश्चित करता है कि प्रत्येक प्रकार का डाटाबेस यूजर, एंड यूजर, एप्लीकेशन सिस्टम एनालिस्ट और फिजिकल स्टोरेज सिस्टम एनालिस्ट वांछित इंफॉर्मेशन को सक्षमता से एक्सेस कर सके। एंड यूजर वह व्यक्ति होता है, जो कम्प्यूटर में प्रशिक्षित व्यक्ति नहीं है, लेकिन कोई इंफॉर्मेशन हासिल करने के लिए डाटाबेस का उपयोग करता है। उदाहरण के लिए एक बैंक डाटाबेस में एक ग्राहक जो यह जानना चाहता है कि उसके खाते में बैलेंस कितना है, एंड यूजर कहलाएगा। एप्लीकेशन एनालिस्ट वह होता है, जिसका

about all of the database of logical level i.e., what all data constitute the database, what are the relationships between the data entities etc. without considering the physical implementation details. The third type of user, the physical storage system analyst is concerned with the physical implementation details of the database i.e., how would the database be stored on which storage device? What will be the starting address of the database? What will be the storage technique? etc.

Since the requirements of different users differ from one another, the complexity of the database is hidden from them, if needed, through several levels of abstraction in order to simplify their interaction with the system. The various levels of database implementation are being discussed in the following section.

### Various Levels of Database Abstraction:

A database is implemented through three general levels, internal, conceptual and external so as cater to the needs of its users.

**1. Internal Level (Physical Level) :** The lowest level of abstraction, the internal level is the one closest to physical storage. This level is also sometimes termed as physical level. It describes how the data are actually stored on the storage medium. At this level, complex low-level data structures are described in details.

**2. Conceptual Level :** This level of abstraction describes what data are actually stored in the database. It also describes the relationships existing among data. At this level, the database is described logically in terms of simple data structures. The users of this level are not concerned with how logical data structures will be implemented at the physical level. Rather, they just are concerned about what information is to be kept in the database.

संबंध लॉजिकल लेवल पर सारे डाटाबेस से होता है अर्थात् डाटाबेस में डाटा किस प्रकार का है, डाटा एंटीटीज में रिलेशनशीप कैसी है, आदि। वह इस बात पर निर्भर नहीं करता कि फिजिकल इम्प्लीमेंटेशन डिटेल क्या है। तीसरे प्रकार का यूजर, फिजिकल स्टोरेज सिस्टम एनालिस्ट का संबंध डाटाबेस के फिजिकल इम्प्लीमेंटेशन ब्योरों से होता है। अर्थात् डाटाबेस कैसे और किस स्टोरेज डिवाइस पर स्टोर किया जाएगा? डाटाबेस का शुरूआती एड्रेस क्या होगा? स्टोरेज तकनीक क्या होगी? आदि।

चूंकि विभिन्न यूजर्स की आवश्यकताएँ एक-दूसरे से भिन्न होती हैं, डाटाबेस की जटिलता उनसे छिपी रहती है। ऐसा एब्स्ट्रैक्शन के कई लेवल से किया जाता है कि सिस्टम के साथ उनका इंटरैक्शन आसान हो जाए। विभिन्न स्तर के डाटाबेस इम्प्लीमेंटेशन नीचे डिस्कस किए गए हैं।

### विभिन्न प्रकार के डाटाबेस एब्स्ट्रैक्शन :

डाटाबेस तीन सामान्य स्तरों से इम्प्लीमेंटेड किया जाता है। ये स्तर हैं - आंतरिक, अवधारणात्मक और बाहरी, ताकि यूजर्स की आवश्यकताएँ पूरी की जा सकें।

**1. इंटरनल लेवल (भौतिक स्तर) :** एब्स्ट्रैक्शन का निम्नतम स्तर, इंटरनल लेवल (आंतरिक स्तर) फिजिकल स्टोरेज के सबसे निकट रहता है। यह डिस्क्राइब करता है कि डाटा वास्तव में स्टोरेज मीडियम पर कैसे स्टोर किया जाता है। इस स्तर का जटिल लौ-लेवल डाटा स्ट्रक्चर विचार से डिस्क्राइब किया जाता है।

**2. अवधारणात्मक स्तर :** एब्स्ट्रैक्शन का यह स्तर डिस्क्राइब करता है कि किस प्रकार का डाटा वास्तव में डाटाबेस में स्टोर किया गया है। यह डाटा के बीच में संबंधों को भी डिस्क्राइब करता है। इस स्तर पर डाटा को लॉजिकली सिंपल डाटा स्ट्रक्चर के संदर्भ में डिस्क्राइब किया जाता है। इस स्तर के यूजर्स इस बात की चिंता नहीं करते कि फिजिकल लेवल पर लॉजिकल डाटा स्ट्रक्चर किस प्रकार इम्प्लीमेंट किया जाएगा। इसके बजाय उनकी चिंता सिर्फ यह रहती है कि कैसी इंफॉर्मेशन डाटाबेस में रखी जाना है।

**3. External Level (View Level) :** This is the level closest to the users and is concerned with the way in which the data are viewed by individual users. Most of the users of the database are not concerned with all the information contained in the database. Instead, they need only a part of the database relevant to them. For example, even though the bank database stores a lot many information, an account holder (a user) is interested only in his account details and not with the rest of the information stored in the database. To simply such users, interaction with the system, this level of abstraction is defined. The system, thus provides many views for the same database. The following fig. illustrates the interrelationship among these three levels of abstraction.

3. बाहरी स्तर (व्यू स्तर) : यह लेवल यूजर के निकटतम रहता है और इसका संबंध उस तरीके से रहता है, जिससे अलग-अलग यूजर डाटा व्यू करेगा। डाटाबेस के ज्यादातर यूजर्स को इस बात की चिंता नहीं होती कि डाटाबेस में पूरी इंफॉर्मेशन क्या है। इसके बजाय उन्हें उनसे संबंधित डाटाबेस के अंश की आवश्यकता होती है। उदाहरण के लिए बैंक डाटाबेस बहुत सारी इंफॉर्मेशन स्टोर करता है, एक खाताधारी (यूजर) को सिर्फ अपने खाते के व्यौरों में दिलचस्पी होती है। उसे डाटाबेस में स्टोर की गई शेष जानकारी से कोई मतलब नहीं होता है। सिस्टम के साथ ऐसे यूजर्स के इंटरैक्शन भर के लिए इस लेवल के एब्स्ट्रैक्शन को डिफाइन किया जाता है। नीचे चित्र 1.8 में एब्स्ट्रैक्शन के तीन स्तरों का अंतर्संबंध दर्शाया गया है।

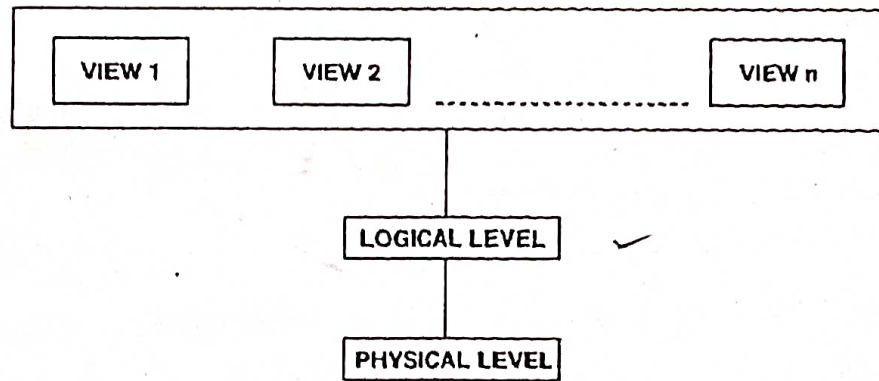


Fig. 1.8 Levels of Data Abstraction

### Schema, Subschema & Instances :

The overall logical design of the database is called **schema**. A schema is a logical database description and is drawn as a chart of the types of data that are used. It is a framework on which values of data item can be fitted.

The term schema is used to mean an overall chart of all the data item types and record types stored in a database. The term **subschema** refers to the same view but for the data item types and record types which are used in particular application or by particular user. Many different subschemas can be derived from one schema.

### स्कीमा, सबस्कीमा और इंस्टेंसेस :

डाटाबेस का ओवर ऑल लॉजिकल डिजाइन स्कीमा कहलाता है। यह एक लॉजिकल डाटाबेस डिस्क्रिप्शन है और इसे उपयोग में लाने वाले डाटा टाईप के चार्ट के रूप में दर्शाया जाता है। यह एक फ्रेमवर्क है, जिन पर डाटा आयटम की वेल्यूज फिट की जाती है।

स्कीमा टर्म का उपयोग डाटाबेस में स्टोर किए सभी डाटा आयटम टाईप्स और रेकॉर्ड टाईप्स के ओवर ऑल चार्ट के लिए किया जाता है। सब-स्कीमा टर्म का उल्लेख उसी प्रकार के व्यू में किया जाता है, लेकिन इसका उपयोग किसी खास एप्लीकेशन या खास यूजर द्वारा उपयोग में लाए जा रहे, डाटा आयटम टाईप्स और रेकॉर्ड टाईप्स के लिए किया जाता है। एक स्कीमा से कई भिन्न सब-स्कीमा निकाले जा सकते हैं।

Database system have several schemas, partitioned according to the levels of abstraction. At the lowest levels is the **physical schema**, at the intermediate level is the **logical schema** and at the highest level is a **subschema**. In general, database system support one physical schema, one logical schema & several subschemas.

The collection of information stored in the database at a particular moment is called an instance of the database. A variable of a given type has a particular value at a given instant. The value of a variable in programming languages corresponds to an instance of a database schema.

### Data Independence :

Any software has mainly two parts, first is the database and second is the application or program which accesses and uses the database. First the database is designed and then the application program are written for it, which heavily depends on database design. Sometimes, after the application program has been written there is a need to change the storage structure or database design. Data Independence is all about, the ability to modify a schema definition (i.e. storage structure) in one level without affecting a schema definition in the next higher level. There are two levels of data independence.

**(1) Physical Data Independence :** Physical data independence is the ability to modify the physical schema without causing application programs to be rewritten.

**(2) Logical Data Independence :** Logical data independence is the ability to modify logical schema without causing application programs to be rewritten.

Modification at the logical level are necessary whenever the logical structure of the database is altered.

Logical data independence is more difficult to achieve than is physical data independence,

डाटाबेस सिस्टम में स्कीमा होते हैं, जो एब्स्ट्रैक्शन लेवल के हिसाब से बंटे होते हैं। निम्नतम स्तर पर फिजिकल स्कीमा बीच के स्तर पर लॉजिकल स्कीमा और उच्चतम स्तर पर एक सब-स्कीमा होता है। साधारणतः डाटाबेस सिस्टम एक फिजिकल और एक लॉजिकल स्कीमा के अलावा कई सब-स्कीमा सपोर्ट करता है।

किसी डाटाबेस में किसी खास क्षण इंफॉर्मेशन का संग्रह डाटाबेस का इंस्टंस कहलाता है। किसी दिए हुए टाईप के वैरियेबल को दिए हुए इंस्टंस पर कोई विशिष्ट वैल्यू होती है। प्रोग्रामिंग लैंग्वेज में वैरियेबल की वैल्यू डाटाबेस स्कीमा के इंस्टंस के अनुरूप होती है।

### डाटा स्वतंत्रता :

किसी भी सॉफ्टवेयर में मुख्यतः दो भाग होते हैं, पहला तो डाटाबेस और दूसरा एप्लीकेशन या प्रोग्राम जो डाटाबेस को एक्सेस व उपयोग करता है। पहले डाटाबेस डिजाइन किए जाते हैं और फिर इसके लिए एप्लीकेशन प्रोग्राम लिखे जाते हैं, जो डाटाबेस डिजाइन पर जबरदस्त तरीके से निर्भर होते हैं। कभी-कभी एप्लीकेशन प्रोग्राम लिखे जाने के बाद स्टोरेज स्ट्रक्चर या डाटाबेस डिजाइन को बदलने की आवश्यकता होती है। डाटा स्वतंत्रता का कुल जमा संबंध स्कीमा डेफिनेशन (अर्थात् स्टोरेज स्ट्रक्चर) को अगले उच्च स्तर को प्रभावित किए बिना, एक स्तर पर मॉडिफाय करने की क्षमता से होता है। डाटा स्वतंत्रता के दो स्तर होते हैं -

**(1) फिजिकल डाटा इंडिपेंडेंस :** यह एप्लीकेशन प्रोग्राम को फिर से लिखे बगैर फिजिकल स्कीमा को मॉडिफाय करने की क्षमता है।

**(2) लॉजिकल डाटा इंडिपेंडेंस :** यह एप्लीकेशन प्रोग्राम को फिर से लिखे बगैर लॉजिकल स्कीमा को मॉडिफाय करने की क्षमता है।

जब भी डाटाबेस का लॉजिकल स्ट्रक्चर बदला जाता है, तो लॉजिकल लेवल पर मॉडिफिकेशन जरूरी हो जाता है।

चूंकि एप्लीकेशन प्रोग्राम उस डाटा के लॉजिकल स्ट्रक्चर पर अत्यधिक निर्भर करता है, जिसे वह एक्सेस करता है, इसलिए

since application programs are heavily dependent on the logical structure of the data that they access.

The concept of data independence is to hide implementation details from users & helps the users to concentrate on the general structure, rather than on low-level implementation details.

### Database Administrator :

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. The person who has such central control over the system is called the database administrator (DBA). The functions of the DBA include the following :

1. **Schema definition** : The DBA creates the original database schema by writing a set of definitions that is translated by the DDL compiler to a set of tables that is stored permanently in the data dictionary.
2. **Storage structure and access-method definition** : The DBA creates appropriate storage structures and access methods by writing a set of definitions, which is translated by the data-storage and data-definition-language compiler.
3. **Schema and physical-organization modification** : Programmers accomplish the relatively rare modifications either to the database schema or to the description of the physical storage organization by writing a set of definitions that is used by either the DDL compiler or the data-storage and data-definition language compiler to generate modifications to the appropriate internal system tables (for example, the data dictionary).
4. **Granting of authorization for data access** : The granting of different types of authorization allows the database administrator to regulate which parts of the database various users can

फिजिकल डाटा इंडिपेंडेंस के बजाय लॉजिकल डाटा इंडिपेंडेंस को हासिल करना अधिक मुश्किल है।

डाटा इंडिपेंडेंस की अवधारणा यूजर्स से क्रियान्वयन संबंधी विस्तृत ब्यौरे छिपाना है। इससे यूजर्स को निचले स्तर के क्रियान्वयन संबंधी ब्यौरों के बजाय जनरल स्ट्रक्चर पर ध्यान केंद्रित करने में मदद मिलती है।

### डाटाबेस एडमिनिस्ट्रेटर :

DBMSs को उपयोग करने के प्रमुख कारणों में से एक यह है कि इससे डाटा और डाटा को एक्सेस करने वाले प्रोग्राम्स दोनों पर केंद्रिय नियंत्रण रहता है। वह व्यक्ति जिसका सिस्टम पर यह केन्द्रिय नियंत्रण रहता है, उसे डाटाबेस एडमिनिस्ट्रेटर (DBA) कहते हैं। DBA के कार्यों में निम्न बातें शामिल रहती हैं-

1. **स्कीमा डेफिनेशन** : डेफिनेशन्स का एक सेट लिखकर DBA, ओरिजनल डाटाबेस स्कीमा निर्मित करता है। इसे DDL कम्पाईलर टेबल्स के एक सेट में ट्रांसलेट करता है। डाटा डिक्शनरी में इसे स्थायी रूप से स्टोर कर लिया जाता है।
2. **स्टोरेज स्ट्रक्चर एंड एक्सेस मेथड डेफिनेशन** : DBA डेफिनेशन्स का सेट लिखकर उचित स्टोरेज स्ट्रक्चर और एक्सेस मेथड निर्मित करता है। डेफिनेशन्स के इस सेट को डाटा स्टोरेज और डाटा डेफिनेशन लेंग्वेज कम्पाईलर द्वारा ट्रांसलेट किया जाता है।
3. **स्कीमा और फिजिकल ऑर्गनाइजेशन मॉडिफिकेशन** : DDL कम्पाईलर, डाटा स्टोरेज और डाटा डेफिनेशन लेंग्वेज कम्पाईलर द्वारा उचित इंटरनल सिस्टम टेबल्स (जैसे, डाटा डिक्शनरी) में मॉडिफिकेशन जनरेट करने के लिए जिस डेफिनेशन सेट का उपयोग करते हैं, उन्हें लिखकर प्रोग्रामर, डाटाबेस स्कीमा या फिजिकल स्टोरेज ऑर्गनाइजेशन में तुलनात्मक रूप से दुर्लभ मॉडिफिकेशन हासिल करते हैं।
4. **डाटा एक्सेस के लिए ऑथोराइजेशन देना** : विभिन्न प्रकार के ऑथोराइजेशन देने से डाटाबेस एडमिनिस्ट्रेटर को इस बात का नियमन करने की सुविधा मिलती है कि विभिन्न यूजर्स डाटाबेस के कौन से हिस्से उपयोग में ला सकते हैं। ऑथोराइजेशन इंपॉर्मेशन

access. The authorization information is kept in a special system structure that is consulted by the database system whenever access to the data is attempted in the system.

**5. Integrity-constraint specification :** The data values stored in the database must satisfy certain consistency constraints. For example, the minimum balance in an account can be Rs. 500. Such a constraint must be specified explicitly by the database administrator. The integrity constraints are kept in a special system structure that is consulted by the database system whenever an update takes place in the system.

**6. Routine Maintenance :** DBA Routine maintenance activities include :

- Periodically backing up database.
- Ensuring that enough free disk space is available for normal operations & upgrading disk space as required.
- Monitoring jobs running on the database.

### Transaction Management :

Transaction is a collection of operations. For example a fund transfer, in which one account (say Q) is debited and another account (say P) is credited. Clearly, it is essential that either both the credit and debit occur, or that neither occur i.e., the funds transfer must happen in its entirety or not at all. This all or nothing requirement is called **atomicity**. It is also essential that the execution of the funds transfer preserve the consistency of the database. That is, the value of the sum  $P + Q$  must be preserved. This correctness requirement is called **consistency**. And after the successful execution of a funds transfer, the new values of accounts P and Q must persist, despite the possibility of system failure. This persistence requirement is called **durability**.

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

को स्पेशल सिस्टम स्ट्रक्चर में रखा जाता है। जब भी सिस्टम में डाटा को एक्सेस करने का प्रयास होता है, तो डाटाबेस सिस्टम इसी स्ट्रक्चर से कंसल्ट करता है।

5. **इंटिग्रिटी-कन्स्ट्रेंट स्पेसीफिकेशन :** डाटाबेस में स्टोरी की गई वेल्यूज को कुछ कंसिस्टेंसी कन्स्ट्रेंट्स पर खरा उतारना होता है। उदाहरण के लिए किसी खाते में न्यूनतम बैलेंस की राशी 500 रूपए हो सकती है। डाटाबेस एडमिनिस्ट्रेटर को ऐसे कन्स्ट्रेंट्स स्पष्ट रूप से स्पेसीफॉय करने चाहिए। इंटिग्रिटी कन्स्ट्रेंट्स को एक खास स्ट्रक्चर में रखा जाता है, जिसे डाटाबेस सिस्टम द्वारा सिस्टम में अपडेट होने पर कंसल्ट किया जाता है।

6. **रूटीन रखरखाव :** DBA रोजमर्रा के रखरखाव में निम्न बातें शामिल रहती हैं -

- डाटाबेस को समय-समय पर बैकअप करना।
- यह सुनिश्चित करना है कि सामान्य ऑपरेशन्स के लिए पर्याप्त फ्री स्पेस उपलब्ध हो। डिस्क को आवश्यकतानुसार अपग्रेड करना।
- डाटाबेस में चल रहे जॉब की मॉनिटरिंग करना।

### ट्रांजेक्शन मैनेजमेंट :

ट्रांजेक्शन ऑपरेशन्स का संग्रह होता है। उदाहरण के लिए फंड ट्रांसफर जिसमें एक खाते (मान लें Q) से राशी डेबिट होती है और दूसरे (मान लें P) से राशी क्रेडिट होती है। साफ है कि या तो क्रेडिट व डेबिट दोनों होना चाहिए या कुछ भी नहीं होना चाहिए। अर्थात् फंड ट्रांसफर इसकी संपूर्णता में होना चाहिए या होना ही नहीं चाहिए। यह सब या कुछ भी नहीं आवश्यकता को **atomicity** कहते हैं। यह भी आवश्यक है कि फंड ट्रांसफर का एक्जिक्यूशन डाटाबेस की कंसिस्टेंसी संरक्षित करें। अर्थात्  $P + Q$  का सम की वेल्यू संरक्षित है। अचूकता की इस आवश्यकता को कंसिस्टेंसी कहते हैं। सफल फंड ट्रांसफर के बाद P और Q खाते की वह वेल्यू सिस्टम फेल्यूर की संभावना के बावजूद बनी रहना चाहिए। परसिस्टेंस की यह आवश्यकता ड्यूरैबिलिटी कहलाती है।

ट्रांजेक्शन, ऑपरेशन्स का ऐसा संग्रह है, जो किसी डाटाबेस एप्लीकेशन में सिंगल लॉजिकल फंक्शन को परफॉर्म

database application. Each transaction is a unit of both atomicity and consistency. Thus, it is required that transactions do not violate any database consistency constraints. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates.

It is the programmer's responsibility to define properly the various transactions, so that each preserves the consistency of the database. Transactions Management Component has the responsibilities of ensuring.

The database system must perform failure recovery, that is, detect system failures and restore the database to the old consistent before the failure to ensure atomicity.

Also, when several transactions update the database concurrently, the consistency of data may no longer be preserved, even though each individual transaction is correct. It is the responsibility of the concurrency control manager to control the interaction among the concurrent transactions, to ensure the consistency of the database.

### Database Users :

A primary goal of a database system is to provide an environment for retrieving information from and storing new information into the database. There are four different types of database-system users, differentiated by the way that they expect to interact with the system.

(1) **Application programmers** are computer professionals who interact with the system through DML calls which are embedded in a program written in a host language (for example Cobol, Pascal, C). These programs are commonly referred to as application programs.

Since the DML syntax is usually markedly different from the host language syntax, DML calls

करता है। प्रत्येक ट्रांजेक्शन एटॉमिसिटी और कंसिस्टेंसी की एक इकाई है। इस प्रकार यह आवश्यक है कि ट्रांजेक्शन किसी डाटाबेस कंसिस्टेंसी कंस्ट्रेंट्स का उल्लंघन न करें। अर्थात् यदि ट्रांजेक्शन शुरू होने के समय डाटाबेस कंसिस्टेंट है तो ट्रांजेक्शन सफलतापूर्वक पूरा होने पर डाटाबेस कंसिस्टेंट होना चाहिए।

विभिन्न ट्रांजेक्शन्स को ठीक तरह से डिफाईन करना प्रोग्रामर की जिम्मेदारी है, ताकि प्रत्येक डाटाबेस की कंसिस्टेंसी को संरक्षित करें। ट्रांजेक्शन मैनेजमेंट कम्पोनेंट के पास यह सुनिश्चित करने की जिम्मेदारी होती है।

डाटाबेस सिस्टम को फेल्यूर रिकवरी परफॉर्म करना चाहिए। इसका मतलब यह है कि इसे सिस्टम के फेल होने का पता लगाकर डाटाबेस को फेल्यूर के पहले की स्थिति में री-स्टोर करना चाहिए ताकि एटॉमिसिटी सुनिश्चित हो सके।

फिर जब डाटाबेस को कई ट्रांजेक्शन एक साथ अपडेट कर रहे हों तो डाटा की कंसिस्टेंसी हो सकता है, संरक्षित न रहे, फिर चाहे व्यक्तिगत ट्रांजेक्शन बिल्कुल सही क्यों न हों। यह कान्कोर्सी कंट्रोल मैनेजर की जिम्मेदारी है कि वह डाटाबेस की कंसिस्टेंसी सुनिश्चित करने के लिए कान्कुरंट ट्रांजेक्शन के बीच इंटरैक्शन कंट्रोल करे।

### डाटाबेस यूजर्स :

डाटाबेस सिस्टम का प्राथमिक लक्ष्य डाटाबेस से इंफॉर्मेशन री-ट्राईव करने और नई इंफॉर्मेशन स्टोर करने के लिए वातावरण उपलब्ध कराना है। चार भिन्न प्रकार के डाटाबेस सिस्टम यूजर होते हैं। सिस्टम से वे किस तरह से इंटरैक्ट करने की अपेक्षा रखते हैं, उसके आधार पर उन्हें एक-दूसरे से अलग किया जाता है।

(1) **एप्लीकेशन प्रोग्रामर**, वे कम्प्यूटर प्रोफेशनल होते हैं, जो सिस्टम से DML कॉल्स के माध्यम से इंटरैक्ट करते हैं। ये DML कॉल्स होस्ट लैंग्वेज (उदाहरण, Cobol, Pascal, C) में लिखे प्रोग्राम में मिले होते हैं। इन प्रोग्राम्स को आमतौर पर एप्लीकेशन प्रोग्राम के रूप में रेफर किया जाता है।

चूंकि DML सिन्टेक्स उल्लेखनीय रूप से होस्ट लैंग्वेज सिन्टेक्स से भिन्न होते हैं, DML कॉल्स आमतौर पर एप



are usually prefaced by a special character so that the appropriate code can be generated. A special preprocessor, called the DML precompiler, converts the DML statement to normal procedure calls in the host language. The resulting program is then run through the host-language compiler, which generates appropriate object code.

(2) **Sophisticated users** interact with the system without writing programs. Instead, they form their requests in a database query language. Each such query is submitted to a query processor whose function is to break down DML statement into instructions that the storage manager understands. Analysts who submit queries to explore data in the database fall in this category.

(3) **Specialized users** are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework. Among these applications are computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types (for example, graphics data and audio data) and environment-modeling systems.

(4) **Naive users** are unsophisticated users who interact with the system by invoking one of the permanent application programs that have been written previously. For example, a bank customer who needs to transfer Rs. 500 from account A to account B invokes a program called transfer. This program asks the customer for the amount of money to be transferred, the account from which the money is to be transferred and the account to which the money is to be transferred.

### Database System Structure :

A database system is divided into modules that deal with different responsibilities of the overall system. The functional components of a database system can be broadly divided into the storage manager and the query processor components.

स्पेशल कैरेक्टर से प्रीफिक्स किए होते हैं, ताकि उचित कोड जनरेट किया जा सके। एक विशेष प्री-प्रोसेसर जिसे DML प्रीकम्पाइलर कहते हैं। DML स्टेटमेंट को नार्मल प्रोसेसर कॉलस को होस्ट लैंग्वेज में बदलता है। परिणामस्वरूप प्रायः प्रोग्राम फिर होस्ट लैंग्वेज कम्पाइलर से चलाया जाता है, जो उचित ऑब्जेक्ट कोड को जन्म देता है।

(2) **सांफिस्टिकेटेड यूजर्स**, ये सिस्टम से प्रोग्राम लिखे वगैर इंटेरेक्ट करते हैं। इसके बजाय वे उनकी रिक्वेस्ट डाटाबेस क्वेरी लैंग्वेज में बनाते हैं। ऐसी प्रत्येक क्वेरी एक क्वेरी प्रोसेसर में सबमिट की जाती है, जिसका काम DML स्टेटमेंट को ऐसी इंस्ट्रक्शन्स में तोड़ना होता है, जिन्हें स्टोरेज मैनेजर समझता है। ऐसे एनालिस्ट, जो डाटाबेस में डाटा एक्सप्लोर करने के लिए क्वेरी सबमिट करते हैं, इस श्रेणी में आते हैं।

(3) **स्पेशलाइज्ड यूजर्स** : ये वे यूजर्स होते हैं, जो ऐसे स्पेशलाइज्ड डाटाबेस एप्लीकेशन लिखते हैं, जो परंपरागत डाटा प्रोसेसिंग फ्रेमवर्क में फिट नहीं होते हैं। इन एप्लीकेशन्स में कम्प्यूटर एडेड डिजाइन सिस्टम, नॉलेज बेस एंड एक्सपर्ट सिस्टम, कॉम्प्लेक्स डाटा टाईप्स में डाटा स्टोर करने वाले सिस्टम (उदाहरण, ग्राफिक व आडियो डाटा) और एनवायरमेंट मॉडलिंग सिस्टम आते हैं।

(4) **नाइव यूजर्स** : ये वे अपरिष्कृत यूजर्स हैं, जो सिस्टम से पूर्व में लिखे स्थायी एप्लीकेशन प्रोग्राम को इनवोक करके इंटेरेक्ट करते हैं। उदाहरण के लिए बैंक का एक ग्राहक जिसे एक खाते A से खाता B में 500 रूपए ट्रांसफर करना है, वह ट्रांसफर नाम के प्रोग्राम को इनवोक करेगा। यह प्रोग्राम कस्टमर से ट्रांसफर करने वाली राशी पूछता है तथा जिस खाते से पैसा ट्रांसफर किया जाना है और जिसमें किया जाता है, उस खाते के बारे में पूछता है।

### डाटाबेस सिस्टम स्ट्रक्चर :

एक डाटाबेस सिस्टम कई मॉड्यूल्स में विभाजित होता है, जो ओवरऑल सिस्टम की विभिन्न जिम्मेदारियों से निपटते हैं। डाटाबेस सिस्टम के फंक्शनल कम्पोनेंट को मोटे तौर पर स्टोरेज मैनेजर और क्वेरी प्रोसेसर कम्पोनेंट्स में बांटा जा सकता है।

(1) **Storage Manager** : A large amount of storage space is required for storing corporate databases (which may range from hundreds to gigabytes to terabytes of data) and to manage this storage manager is required. Data are to moved between disk storage and main memory as per requirement because main memory of computer cannot store this much information.

A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system. The storage manager translates the various DML statements into low-level file-system commands. Thus, the storage manager is responsible for storing, retrieving and updating data in the database.

The storage manager components include -

- **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and check the authority of users to access data.
- **Transaction manager**, which ensures that the database remains in a consistent (correct) state despite system failures and that concurrent transaction executions proceed without conflicting.
- **File manager**, which manages the allocation of space on disk storage and the data structure used to represent information stored on disk.
- **Buffer manager**, which is responsible for fetching data from disk storage into main memory and deciding what data to cache in main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much large than the size of main memory.

(1) **स्टोरेज मैनेजर** : कॉर्पोरेट डेटाबेस को स्टोर करने के लिए बहुत बड़े स्टोरेज स्पेस की आवश्यकता होती है (जो मैकडों गिगाबाइट से टर्बाइट डेटा तक की हो सकती है) और इसे मैनेज करने के लिए स्टोरेज मैनेजर की आवश्यकता होती है। आवश्यकता के अनुसार डेटा को डिस्क स्टोरेज और मेमोरी के बीच मूव किया जाता है, क्योंकि कंप्यूटर की मेमोरी इतनी इन्फॉर्मेशन स्टोर नहीं कर सकती है।

स्टोरेज मैनेजर एक प्रोग्राम मॉड्यूल है, जो डेटाबेस में स्टोर किए लो-लेवल डेटा और सिस्टम को सबमिट किए एप्लीकेशन व डेटा के बीच इंटरफेज उपलब्ध कराता है। स्टोरेज मैनेजर, फाइल मैनेजर के साथ इंटरैक्शन के लिए जिम्मेदार है। एक फाइल सिस्टम का उपयोग करके कच्चा डेटा डिस्क पर स्टोर किया जाता है। यह फाइल सिस्टम आमतौर पर कन्वेंशनल ऑपरेटिंग सिस्टम द्वारा उपलब्ध कराया जाता है। स्टोरेज मैनेजर विभिन्न DML स्टेटमेंट्स को लो-लेवल फाइल सिस्टम कमांड्स में ट्रांसलेट करता है। इस प्रकार स्टोरेज मैनेजर डेटाबेस में डेटा स्टोरींग, री-ट्राईविंग और अपडेटिंग के लिए जिम्मेदार होता है।

स्टोरेज मैनेजर कम्पोनेंट में ये हिस्से शामिल रहते हैं -

- **ऑथोरिजेशन और इंटीग्रिटी मैनेजर**, इंटीग्रिटी कंस्ट्रेंट्स को टेस्ट करता है और डेटा एक्सेस करने की यूजर्स की ऑथोरिटी चेक करता है।
- **ट्रांजेक्शन मैनेजर**, सुनिश्चित करता है कि डेटाबेस कंसिस्टेंट (कोरेक्ट) अवस्था में रहे, फिर चाहे सिस्टम फेल्यूर ही क्यों न हो जाए। वह यह भी देखता है कि एक साथ होने वाले ट्रांजेक्शन एक्जिक्यूशन बिना किसी संघर्ष के संपन्न हो जाए।
- **फाइल मैनेजर**, डिस्क स्टोरेज पर स्पेस एलोकेशन और डिस्क पर स्टोर की गई इन्फॉर्मेशन को दर्शाने के लिए उपयोग में लाए गए, डेटा स्ट्रक्चर को मैनेज करता है।
- **बफर मैनेजर**, डिस्क स्टोरेज से मेमोरी में डेटा फेज करने और मेमोरी में कौन-सा डेटा (cache) करना है, इसका निर्णय लेने के लिए जिम्मेदार होता है। बफर मैनेजर डेटाबेस सिस्टम का महत्वपूर्ण हिस्सा होता है, क्योंकि यह डेटाबेस को मेमोरी के साईज से बड़े डेटा साईज को हैंडल करने योग्य बनाया है।

The storage manager implements several data structures as part of the physical system implementation :

- **Data files**, which store the database itself.
- **Data dictionary**, which stores metadata about the structure of the database, in particular the schema of the database.
- **Indices**, which provide fast access to data items that hold particular values.
- **Statistical data**, which store statistical information about the data in the database.

(2) **Query Processor** : A query processor is important as it helps the database system to simplify and facilitate data access.

High-level views help to achieve this goal, with them, users of the system are not be burdened unnecessarily with the physical details of the implementation of the system. However, quick processing of updates and queries written in a nonprocedural language, at the logical level, results into an efficient sequence of operations at the physical level.

The query processor components include :

- **DDL interpreter**, which interprets DDL statements and records the definitions in the data dictionary.
- **DML compiler**, which translate DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs **query optimization**, that is, it picks the lowest cost evaluation plan from among the alternatives.

- **Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

स्टोरेज मैनेजर, फिजिकल सिस्टम इम्प्लीमेंटेशन के हिस्से के रूप में कई डाटा स्ट्रक्चर अमल में लाता है -

- डाटा फाईल्स, जो डाटाबेस को ही स्टोर करती है।
- डाटा डिक्शनरी, जो डाटाबेस स्ट्रक्चर के बारे में मेटा डाटा स्टोर करती है, विशेषतौर से डाटाबेस की स्कीमा।
- **इन्डायसेस**, ये ऐसे डाटा आयटम्स को फॉस्ट एक्सेस देते हैं, जिसमें कोई खास वेल्थ होती है।
- **स्टैटिस्टिकल डाटा**, जो डाटाबेस के डाटा के बारे में सांख्यिकीय जानकारी स्टोर करती है।

(2) **क्वेरी प्रोसेसर** : क्वेरी प्रोसेसर इस अर्थ में महत्वपूर्ण है, क्योंकि यह डाटाबेस सिस्टम को डाटा एक्सेस को सरलीकृत करता है और सहायक होता है।

हायलेवल व्यू व इस लक्ष्य को हासिल करने में मदद करता है। इनके होते सिस्टम के यूजर्स पर सिस्टम इम्प्लीमेंटेशन के अनावश्यक भौतिक व्यौरों का बोझ नहीं पड़ता है। हालांकि, नॉन प्रोसिजरल लैंग्वेज में लिखी क्वेरीज और अपडेट्स की तीव्र प्रोसेसिंग लॉजिकल लेवल पर ऑपरेशन्स के कुशल सिक्वेस के रूप में सामने आती है।

क्वेरी प्रोसेसर कम्पोनेंट्स में ये शामिल हैं -

- **DDL इंटरप्रेटर** जो DDL स्टेटमेंट्स की व्याख्या करता है और डाटा डिक्शनरी में डेफिनेशन रेकॉर्ड करता है।
- **DML कम्पाईलर**, यह DML स्टेटमेंट को क्वेरी लैंग्वेज में एक इवेल्यूएशन प्लान में ट्रांसलेट करता है, जो ऐसी लो-लेवल इंस्ट्रक्शन्स का बना होता है, जिसे क्वेरी इवेल्यूएशन इंजन समझता है।

किसी क्वेरी को आमतौर पर कितने भी वैकल्पिक इवेल्यूएशन प्लान में ट्रांसलेट किया जा सकता है, जो समान परिणाम देता है। DML कंपाईलर, क्वेरी ओप्टिमाइजेशन भी परफॉर्म करता है। अर्थात् यह विकल्पों में से सबसे कम कॉस्ट इवेल्यूएशन प्लान को उठाता है।

- **क्वेरी इवेल्यूएशन इंजन**, DML कम्पाईलर द्वारा पैदा की गई लो-लेवल इंस्ट्रक्शन्स को एक्जिक्यूट करता है।

Figure 1.9 shows these components and the connections among them.

चित्र 1.9 इन कम्पोनेंट और उनके बीच संबंध को दर्शाता है।

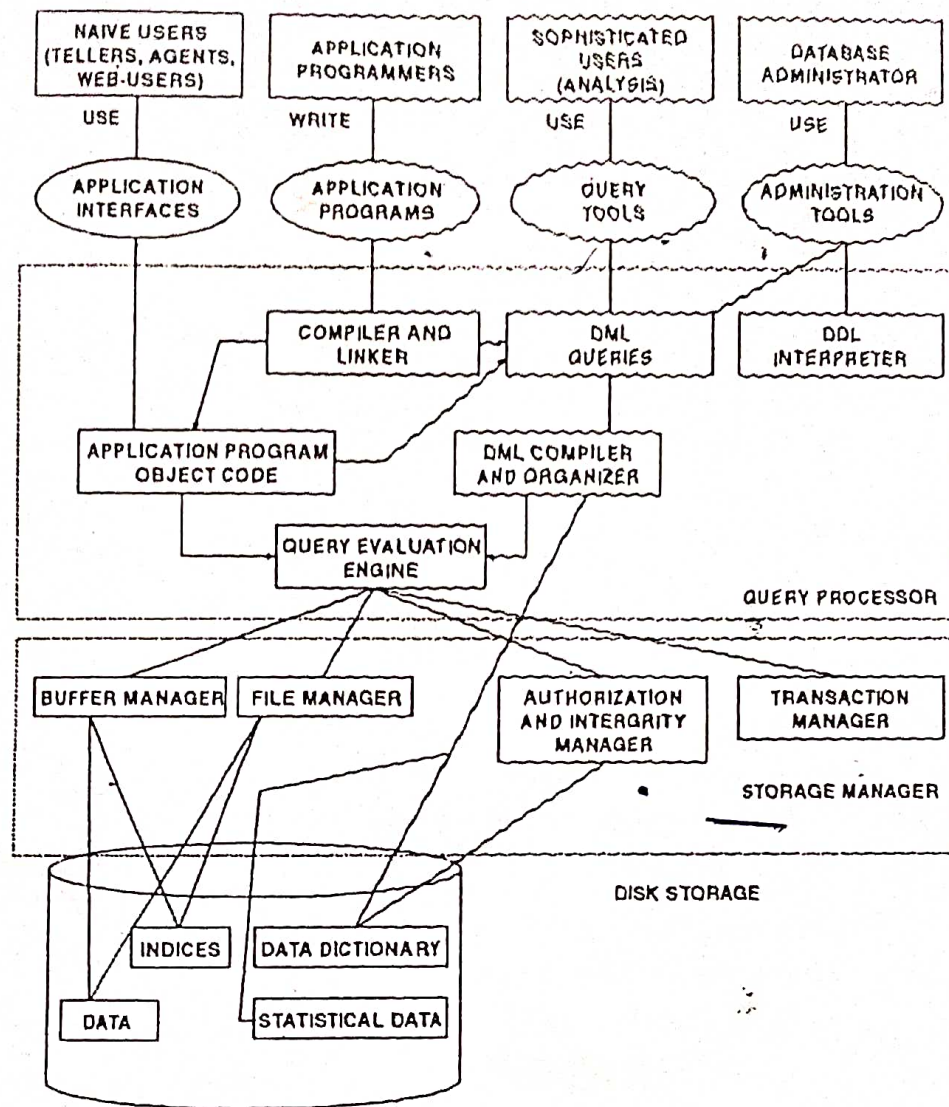


Fig. 1.9 Database System Structure

### Limitations of DBMS :

In spite of the advantages of using a DBMS, there are a few situations in which such a system may involve unnecessary overhead costs as that would not be incurred in traditional file processing. The overhead costs of using a DBMS are due to the following -

1. High initial investment in hardware, software and training.
2. Generality that a DBMS provides for defining and processing data.

### DBMS की सीमाएँ :

DBMS उपयोग करने के फायदों के बावजूद ऐसी स्थितियाँ भी हैं, जिसमें ऐसे सिस्टम में अनावश्यक अतिरिक्त लागत आती है। यह लागत परंपरागत फाइल प्रोसेसिंग में वहन नहीं करना पड़ती है। DBMS उपयोग करने की अतिरिक्त लागत इन कारणों से आती हैं -

1. हार्डवेयर, सॉफ्टवेयर और ट्रेनिंग में उच्च प्रारंभिक निवेश।
2. डाटा डिफाईनिंग और प्रोसेसिंग के लिये DBMS द्वारा उपलब्ध कराई गई साधारणता।

3. Overhead for providing security, concurrency control, recovery and integrity functions.

### Database Development Process :

Database development can be a classically thought of as a systematic process which can be divided in following steps :

- Interpreting the needs of an organization.
- Identifying the business rules.
- Analyzing the business situation & data.
- Designing the database & defining the functional dependencies.

### Database & Information System Development :

Database is a part of information system. A particular database provides the data for one or more information systems.

Generally, database development begins with enterprise data modeling in which the scope and general contents of organizational databases are specified. An enterprise data model may encompass many databases, describes the scope of data maintained by the organization. In enterprise data modeling current systems are reviewed, nature of the business areas to be supported are analyzed, data needed at very high level of abstraction is described and one or more database projects are planned.

### Information System Architecture :

Information system architecture is a conceptual blueprint or plan that expresses the desired future structure for the information systems in an organization.

We would develop an enterprise data model as part of developing the total information systems architecture during information systems planning. According to Zachman (1987) and Sowa and Zachman (1992), an information systems architecture consists of six key components :

3. सिक्योरिटी, कांकरेंसी कंट्रोल, रिकवरी और इंटीग्रिटी फंक्शन उपलब्ध कराने का ओवरहेड (अतिरिक्त खर्च)।

### डाटाबेस डेवलपमेंट प्रोसेस :

डाटाबेस डेवलपमेंट शास्त्रीय शैली में एक ऐसी प्रणालीबद्ध प्रक्रिया के रूप में लिया जाता है, जिसे निम्न चरणों में विभाजित किया जा सकता है -

- किसी संस्थान की आवश्यकताओं की व्याख्या करना।
- बिजनेस रूल्स की पहचान।
- बिजनेस परिस्थितियों एवं डाटा का विश्लेषण।
- डाटाबेस को डिजाइन करना और फंक्शनल डिपेंडेंसी को डिफाइन करना।

### डाटाबेस और इंफॉर्मेशन सिस्टम डेवलपमेंट :

डाटाबेस इंफॉर्मेशन सिस्टम का हिस्सा होता है। कोई विशिष्ट डाटाबेस एक या अधिक इंफॉर्मेशन सिस्टम के लिए डाटा उपलब्ध कराता है।

आमतौर पर डाटाबेस डेवलपमेंट की शुरुआत एंटरप्राइज डाटा मॉडलिंग से होती है। इसमें ऑर्गेनाइजेशनल डाटाबेस का स्कोप और जनरल कंटेंट स्पेसीफाय किए जाते हैं। एक इंटरप्राइज डाटा मॉडल में कई डाटाबेस हो सकते हैं और यह संस्थान द्वारा मेनटेन किए डाटाबेस के स्कोप को डिस्क्राइब करता है। इंटरप्राइज डाटा मॉडलिंग में कंटेंट सिस्टम की समीक्षा की जाती है, सपोर्ट किए जाने वाले बिजनेस एरिया का विश्लेषण किया जाता है। एब्स्ट्रैक्शन के बहुत उच्च स्तर पर आवश्यक डाटा को डिस्क्राइब किया जाता है और एक या अधिक डाटाबेस प्रोजेक्ट प्लान किए जाते हैं।

### इंफॉर्मेशन सिस्टम आर्किटेक्चर :

- इंफॉर्मेशन सिस्टम आर्किटेक्चर, एक अवधारणात्मक खाका या योजना है, जो किसी संस्थान में भविष्य में वांछित इंफॉर्मेशन सिस्टम के स्ट्रक्चर को अभिव्यक्त करता है।

इंफॉर्मेशन सिस्टम प्लानिंग के दौरान संपूर्ण इंफॉर्मेशन सिस्टम आर्किटेक्चर विकसित करने के हिस्से के रूप में हम एक इंटरप्राइज डाटा मॉडल विकसित करेंगे। जैचमैन (1987) तथा सोवा व जैचमैन (1992) के मुताबिक एक इंफॉर्मेशन सिस्टम आर्किटेक्चर छह प्रमुख हिस्सों से बना होता है -