DESIDOC Bulletin of Information Technology, Vol. 22, No. 1, January 2002, pp. 3-20

A Sketch of a Code for Developing Computer-Based Library Information Systems Based on Classified Catalogue Code of Ranganathan (Part - 1)*

Madan Mohan Kashyap

Abstract

The purpose of this paper is to look into how best the certain techniques of database technology and traditional library tools and techniques can be converted as an effective tool for designing and developing the computer-based library information systems and services. This paper proposes a code for developing library information systems or their databases for converging the techniques belonging to the domain library and information science and the domain of information technology for designing, developing and implementing computer-based library information systems and services. The paper explains briefly, various concepts relating to specifying conceptual schema of a library database; identify and specify the various entities or entity-types that are associated with library systems.

1. INTRODUCTION

The use and application computer technology demands that we must take holistic or gestalt type approach, while designing or developing computer-based information system and services. According to this approach, the computer based activities or functions of the organisation must be examined, structured or designed keeping in view that though each sub-unit or subsystem of the organisation is functionally independent of each other, their computerised activities or functions get more cohesive or blended. Thus, while designing computer-based systems and services of an organisation, its activities and functions need to be looked and structured not in isolation of each other, but as a whole. For example, in a computerised system, the output of one system can form input of another; or two functional subsystems can jointly convert input into an output. While performing library functions manually, very often, we repeat certain tasks or activities performed in one unit in the other units. For example, bibliographic data about an item is collected and recorded or produced as output while selecting an item, while writing an order, and preparing a catalogue entry, afresh, every time. In computer-based system, not only repetition in reproduction of same data can be avoided, but also it can be restructured or moulded easily in many formats. The same data can be shared by many applications. Thus, computerisation of library systems demands that we think of data as a resource of whole organisation. The computer based library data must be created

* Part II of this paper will be published in next issue of DBIT which will identify and describe the attributes and data elements of the entities that are relevant or needed for creating the library database records. Select rules of the CCC in amended structure to form the part of the proposed code will be given. The amendment is proposed keeping in view the need of manual cataloguing practices and computerised online catalogues.

in such a manner that the same data can be used for different applications.

This situation also forces us to converge the various library techniques and processes in such a manner that their use and application can be more fruitfully and holistically applied while designing, developing and handling computer-based library information systems and services. The reappraisal and convergence (unification) of library techniques, in the context of using electronic technology and its applications, will enable us to achieve enhanced benefits or better results in our endeavour to develop better library systems and services on modern lines.

2. SCOPE OF CLASSIFIED CATALOGUE CODE

The latest edition Classified Catalogue Code of Ranganatha[25] broadly deals with the following:

- Evolution of the code
- General laws of library science, the normative principles and the canons of cataloguing in the light of which a catalogue code should be written and interpreted at the time of their application
- Mechanics of cataloguing - such as writing and arranging entries
- Standard terminology for use in discipline of cataloguing
- Determination of the author of a bibliographic item, structure of name-of-person in different culture groups
- Rules for rendering different kinds of name - a name-of-person, corporate body, government, institution, bibliographic item, series, etc.
- Technique of chain indexing, class index entries, cross reference index entries
- Supplementary rules necessary in making entries for a composite bibliographic Item and their parts (serial and composite monographs and the independent contributions in these types of bibliographic items)
- Additional rules necessary in building a union catalogue of serials or monographs, a national bibliography, an abstracting or indexing periodical publication, and a catalogue of non-conventional bibliographic Item.

We feel that all the areas described above need to be covered in the proposed code in updated form in addition to the directions required for developing a comprehensive computerised database for a library management system. In designing the code afresh, we have to take total or comprehensive view of the library system, including its objectives, functions, processes or activities (task systems) to be computerised.

The application of tools and techniques of database technology is an essential element in building of computerised databases of library systems. Further, the library databases are going to be the repository of information of various types of information entities associated with the library systems whose databases need to be created.

Thus, while bringing revised version of the code, the emphasis should not only be on to revise the rules for creating electronic records of bibliographic documents but also all other types of entities associated with library systems. In addition, the emphasis should be given on to show the emphasis should aiso be on to way and means to apply the tools and techniques of database technology for building and searching library databases efficiently and effectively.

In the subsequent part, the paper will:

(a) Explain briefly, various concepts relating to specifying conceptual schema of a library database,

(b) Identify and specify the various entities or entity-types that are associated with library systems.

3. APPROACH TO DESIGN A CONCEPTUAL SCHEMA FOR A LIBRARY DATABASE SYSTEM

A database represent some aspect of real world and contains a collection of related data or data records. A database is created and maintained by a database management system (DBMS), such as CDS/ISIS[29.] (*also* *refer*, Dates[8]; Kashyap[16]; Oxborrow[24]; Vossen[30]).

When we define a new database, we first specify its conceptual schema. The conceptual schema is a description of overall database structure, as perceived by a library database administrator, keeping in view the needs of library end users. It is basically an overall view of the library data, taken by the database administrator to describe, organise and present the data to the application programmer, who derives his view of data for internal or physical organisation of data on a storage medium. A conceptual schema or model (also called metadata) can be defined using Entity-Relationship approach to data modelling and analysis and displayed by means of the graphical notation known as Entity-Relationship Diagrams (E-R diagrams) (*cf*. Chen[2, 3]; Codd[6, 7]*; Kashyap*[17, 18,20]; Mair[21]; Teorey[28]; Vossen[30]).

We define the structure of database records by specifying the entity-types or the object sets about which we want to keep data or facts, and their attribute types whose values are to be stored in each entity record, without being concerned with storage details. Entity class specification or description means definition of its relevant attributes (data elements), whose values are actually stored in corresponding data fields comprising records of the entities defined. Every individual entity belonging to a class though possesses all the attributes of the class, but differ from one another in the values (data items) associated with their attributes or data elements.

The conceptual schema is later transformed into a database schema using a DBMS. The database schema is an implementation model of the DBMS. At this point, the corresponding database state is empty, with no data or records. We start building a database when data is first loaded using DBMS.

A loaded database record represent a set of attributes, data elements or facts about the particular real world entity represented by the record and the logical connection among the data is maintained by their physical representation with in the storage medium according to the defined data structure. As such, the data, which the users present to or receive from DBMS is also called logical data.

4. CONCEPT DEFINITION

4.1 Entity

In database terms, an *entity* or *object* is defined as anything (physical or abstract) in the real world about which we store information or data in a database record. An entity may be a tangible object with physical existence, such as, a particular person, an employee, a library member, a book, a serial, a chair, or a table. It may be a non-tangible object with conceptual existence or an abstract concept, such as, an event, a transaction, a job, a procedure, a subject of study, or a university course. In other words, everything that exits in reality or is perceived as being in existence is an entity.

All entities are distinct from one another in the sense that each possesses a particular set of properties, attributes or values that distinguish it from the others. Thus, a bibliographic item has properties such as title, size, price and cost, etc.; a library user possesses attributes such as name, age, educational qualification, place of residence etc.; and a library employee has attributes such as name, address, basic salary, etc. We contrast or differentiate between entities, for some purpose, based on some *characteristic(s),* even if in the real world they are from the same class of objects. For example:

- ❑ Library users and library employees are both *people*, but library user is a *person* who uses library collection while library employee is a *person* who organises and maintains the library collection for the use of the library members.
- ❑ Monographs and serials are both *bibliographic items*, but a monograph is a *non-serial bibliographic item* (i.e., a bibliographic item complete in one part or finite number of parts, or intended to be completed, in a finite number of parts), but a serial is a periodical publication or

bibliographic item published/issued in successive parts bearing numeric and/or chronological designations and intended to be continued indefinitely. Each part is generally made up of distinct and independent contributions, not forming a continuous exposition.

The objects we referred above as entities can be called *entity-types* (or entity classes) because each denotes a set of objects (individual entities), each of which exhibits the properties/attributes described for the class. Thus, the entity-type 'library user' is made up of individual entities, each of which has attributes, name, age, and educational qualification, place of residence, etc. Every individual entity belonging to this class possesses all the attributes of the class but differ from one another in the values of data elements or items associated with their attributes.

Entities for data modelling purpose are categorised into five broad groups or classes, namely: persons, things, places, concepts, and events. Although the entities for data modelling purpose are initially divided into five groups, even these groups are too general to work with.

For the purpose of developing a library information system design, and in particular the data models that are integral part of the system design, each of these five classes must be divided into two or more restrictive classes for purpose of optimal efficiency and usefulness of the database system.

Entities of each group may have to be further divided in large homogeneous groups where all the members are capable of being described in the same manner having identical attributes, or they may be fragmented or considered into many subtypes, each with a description which are either slightly different, or in some cases radically different, from the other members of same group.

Ranganathan's theory of classification, principles and canons are very helpful methods that can be used by data modellers in constructing these grouping.

4.2 Entity-Type

An *entity-type* is a set of all entities to which a specific definition and common attributes and relationships apply.

A database usually contains records of groups of entities that are similar. For example, a library possessing hundreds of different kinds of monographs store almost similar type of information or data in bibliographic records concerning each of the monograph. These monographs share almost the same attributes, but each monograph (entity) has its own value(s) for each attribute. As such, a set of entities that have same attribute and relationships with other entities can be defined as *entity-type*.

The concept entity-type (also called entity class, object-type, object set, or entity set) is very important for developing logical data models, and to specify real-world items in an application or all things (concrete or abstract) about which we store data in an information system.

Entity class description means definition of its relevant attributes (data elements), whose values are actually stored in corresponding data fields comprising records of the entities defined. Every individual entity belonging to a class though possesses all the attributes of the class, but differ from one another in the *values* (data items) associated with their attributes or data elements.

While creating a database, we may consider an entity-type disjointedly or divide it into its subsets. For example, we may say that a library collection constitutes of an entity-type-bibliographic documents, instead of monographs and serial. However, in certain situation it may be necessary to divide entity-types into entity-subtypes for creating a database. For example, we find both the entities subtypes monograph and serial have some distinct attributes and associations with other entity-types of a library system. Therefore, in such a situation, the entity-type bibliographic document may have to be divided into entity-subtypes monographs and serial. In case, we store essentially the same information about the entity-types monograph and serial, we need not consider them entities subtypes and regard these two categories or subsets merely *attribute values* of the entity-type biblographic document the description of an entity-type normally includes the following:

(a) The set of *attribute-types* that describe various properties of each entity-type

(b) A key attribute for each entity-type, whose values can be used to identify each entity uniquely, and

(c) Specification of *unique identifiers* - names, symbols, or tags of the chosen attributes and their characteristics.

4.3 Week Entity-Types

Some entity-types may not have key attributes of their own. Such entity-types are called *week entity-types.* Entities belonging to this category are identified by relating to specific entities belonging to other entity-types, in combination with the values of their attributes. We call the other entity-types the identifying owners. We call the relationship type that relates a week entity to its owner *the identifying relationship of the week entity-type.* Consider the relationship between the entity-types family member and employee. Two or more dependants of two or more distinct employees may have same name (or may have same values for their attribute-types—dependants name, relationship, and date-of-birth). They can be identified or recognised as some distinct persons, only after identifying the employee entity with which each is related and what is the relationship between them. Here each employee entity is said to own the dependent entities that are related to it. However, not every existence dependency results in a week entity. For example, a membership-identity card entity cannot exist unless it is related to a member(person) entity. However, each ID card has its own unique key or identifier, namely: ID card-no. and, hence, is not a week entity.

4.4 Attributes

Every entity or object has some attributes that characterise it. 'Name' of a person and, Title' of a book are attributes of the entity-type 'Person' and' Book', respectively. A Specific value of an attribute of a particular entity constitutes a fact or data about that Entity. For instance, the value of the attribute 'Name' of particular person', say, Swami Dayanad Sraswati is a particular fact or data about him, and serve as his identifier.

An attribute of an entity-type can be intrinsic or natural, such as, size, weight, colour etc. or extrinsic or attained, such as, name, qualification, and so forth.

Entities also attain special type of attributes by having some relationship or connection with other entities, or due to actions performed on them by other entities or agents and bring changes in their state or conditions. For example, the attribute name-of-father of a person or name-of-son of a person is acquired by an entity Person due to its relationships with another entity Person. The relationships being father-of or son-of between the person A and B. Similarly the relationships written-by and classified-by between the entity types Book and Person establishes the attribute name of author and name of classifier of a book entity. The attributes that show relationship or connection between entities and entity classes we name them relationship-related attributes (including one acquired due to actions performed on them by other entities or agents. Some entity attributes do not describe the entities, as such, but describe what it does how it is used, why it is useful, or what is its use etc. The things an entity does are called activities, The attributes that describe the activities of entities; we name them activities-related attributes. We will call relationship-related attributes; action-related attributes and activities-related attributes collectively as relationship-related attribute or attribute of a relationship.

Thus, we can say that an attribute is a property, a piece of information or data about an entity, or about its relationship(s) with another entity or entities. Title,for instance, is an attribute of an entity type (entity class) Monographs, and so is Price, size, name of the author, name of the publisher etc.

Please note, name of the author and name of the publisher are the relationship-related attributes of an entity type Monograph acquired by it due to its relationships written-by and published-by with entity types Author and Publisher, respectively. The relationship, linkage or connection between the entities (or entity classes) Monograph and Author, or Monograph and Publisher are represented by the terms written-by or publisher-by and the relationship-related attributes acquired by entity type Monograph are being represented by the terms name of the author and name of the publisher.

A set of attributes including relationship-related attributes (or link attributes), that is the attributes acquire by an entity due its association with other entities, enables us to describe, identify or locate an entity. A particular entity possesses specific value for each of its attributes. The values of the attributes (its own or acquired due to association or relationship with other entities) become major part of the data stored in the database.

An attribute has one or more data elements associated with it and is represented by one or more sub fields comprising a field, which contains data values about an attribute of an entity in an entity record stored in a storage medium.

Attributes names are represented in a text of a book by boldface and, lower letters and names of entity types in capital letters (e.g., name is an attribute of entity type Employee or title is an attribute of entity type Monograph).

To computerise the activities of a library system, the entity classes (also referred to as entity types) and their attributes (or data elements) must be identified, defined and organised.

4.4.1 Properties of Attributes

Attributes of entity-types have certain properties of their own and when an attribute is defined, the characteristics of each of its property must be specified. For example, the value of the property of an attribute (or a data element) may be represented in the form of an integer, a real number, a character string, and a boolean value or in any other way. This property of an attribute is known as type of data value' an attribute has. Thus, we say 'type-of-attribute' we mean the kind of data value an attribute may possess. These aspects need to be specified, while defining a database.

We must clearly distinguish between the terms '*type-of-attribute'* and *'attribute-type*'. When we say that the 'Name of person' is an *attribute-type*, we mean, it is an attribute possessed by many entities-types 'Human'. Whereas, the 'Name of person' is a *type-of attribute* that is represented by a character string. Similarly, age of person is a *type-of-attribute*, which is represented by an integer.

At times, a given attribute may have more than one value. For example, a particular person or place may be known by two or more names, or is conferred with two or more names, or his/its name has been changed two or more times. In this case, the attribute-type name of the entity-type person and place shall be multivalued. As such, logic requires that we must specify whether a particular attribute (or data element) is single-valued or multi-valued.

To summarise the preceding discussion, we list the various kinds of attributes having different characteristics. Generally, the following kinds of attributes occur in an E-R model—primary versus non-prime (or secondary); simple versus composite; single-valued versus multi-valued; and, stored versus derived. In the succeeding sections, we give detailed description of various kinds of attributes.

4.4.2 Kinds of Attributes

★ *Primary, Non-prime, Secondary and Concatenated Attributes*

As stated earlier, each entity-type with which we may be concerned in a given situation, has a set of attributes associated with it. The associated or chosen attributes are the characteristics of the entity-type that serve to describe the particular entities of that type. Though, all attributes describe an entity-type, one or more attributes uniquely identify or distinguish one entity occurrence from all others of same type comprising an entity set. Such an attribute or attribute group is known as primary attribute, key attribute or simply *a primary key*.

In other words, we can say that a *primary attribute* is that attribute whose value is distinct for each entity. Besides, its value can be used to identify each entity (or an entity record) uniquely. Others, which are not primary attributes are called the non-prime attributes (also called *descriptive attributes that add meaning to the entity-type). A non-prime attribute* does not uniquely identify entity-type. They are also called supportive attributes, particularly those that indicate or support relationships between entity-types. non-prime attributes have one-to-one links with other attributes. One value of a non-secondary attribute is associated with one value of other attributes (or data elements). All attributes (or data elements) are either *primary attributes* (or *keys*) or *non-prime attributes*.

For example the attribute-type membership-no is an attribute whose particular value specifies a particular member of a library and no other member has the same value or membership number.We can say membership-no is the primary attribute or primary key of the entity-type library member. The attribute-types of the entity-type library member such as member-name, member-address are non-prime attributes (or data element types).

A *secondary* attribute or key does not uniquely identify another attribute or data element. It is a non-prime attribute with one or more links with other attributes. That is, one value of a *secondary key* is associated with one or many values of another attribute or data element.

Under certain conditions an entity-type can only be uniquely identified with the help of combined values of more than one attribute, when this occurs, then these attributes are given the name the *concatenated attributes*or simply *concatenate key* or *entity identifier*. For example, we can uniquely identify a particular volume (a single item or document) belonging to a library by the concatenation of three attributes class-no., book-no., collection-no of an entity-type document. we can say the attributes class-no. + book-no. + collection-no. together constitute the entity identifier.

★ Simple and Composite Attributes

Attributes are differentiated or treated as *simple attributes* and *composite* or *aggregated attributes*, respectively. Attributes that are not divisible are called simple or atomic attributes. Composite attribute is an attribute formed from or constituted of simple attributes. *Composite attribute* can be divided into smaller parts with independent meaning of their own, called atomic attributes or data elements. For example, address attribute can be further subdivided into the subparts namely: building; street address, locality; village, town or city; taluk/tehsil, district, region or state; country; postal code. The value of a composite attribute is the concatenation of the values of its constituent simple or atomic attributes. The term composite attribute corresponds to a grouped data element, a grouped field or a combined field in a record.

We can treat an attribute either as a composite attribute or as a simple attribute in certain cases. For example, the name of a particular person can be considered as simple attribute treated as a whole. We can also, consider it as composite attribute by breaking it into two parts - ^aEntry element and ^bOther part of the name, or into three parts or components - ^aForename (i.e., first name), ^bMiddle name, and ^cSurname (i.e., last name). In other words, an attribute consisting of simple or atomic attributes called *Composite Attribute.* Whereas, a simple attribute constitutes of a single data element or constituted of more than one atomic attributes or atomic data elements, treated as whole.

We must also specify whether a given attribute is a simple attribute (or is to be treated as simple attribute), or as a composite attribute (sometimes called aggregated attribute).

★ **Derived and Stored Attributes**

Sometimes values of two attributes are related and processing or calculating from the value(s) of other attribute(s) derives the value of one. For example, adding the cost of individual item listed on the bill often derives total amount on a book bill. Next, the age and date of birth attributes of a person are related and the age of a person can be determined by subtracting from the value of that person's date of birth the value of present-date. In these cases the attributes -total amount and age are derived attributes and said to be derivable from cost attribute, and birth date and present (or current) date attributes respectively, which are called stored attributes.

In case of derived attributes we may or may not store their values in a database. They may be processed whenever data is retrieved based on physical or implementation level decision. The logical representation of data includes such attributes without saying whether or not the data are stored.

★ **Single-Valued and Multi-Valued Attributes**

An attribute is differentiated on the bases of values it possesses. For example, a book or document might deal with more than one subject or may be written-by more than one author. Therefore, we may say that the attributes '*subject*' and 'authorship' of the entity type bibliographic item are multi-valued or multi-occurring attributes. Whereas, the attribute 'height of a book shall always have solitary value. Such an attribute of an entity, which has solitary or one value, is called single-valued or single-occurring attribute.

Thus, m*ulti-occurring* or *multi-valued* attribute may be defined as an attribute having one or more values. Whereas, *single-valued or single-occurring attribute* may be defined as one that has only one or solitary value.

Multi-valued attributes (or data elements) and single valued attributes (or data elements) are called repeating group and non-repeating group, respectively. By definition, a record that contains repeating group must be variable in length. Although repeating groups are important elements of conceptual design, most DBMS cannot directly deal with them, as they have no provision or capability to deal with variable in length record or fields.

In such cases, if an entity-type table contains a repeating group then either it is split into two new entity-type tables or a number of independent fields are provided, which are equal to the maximum number of distinct values a multi-occurring attribute can possess. For example, a bibliographic item can be written by two or more authors, as such the relationship or attribute written-by of bibliographic item is a multi-valued attribute and is represented in the form written-by author 1; written-by author 2; written-by author 3 so on, as if they are two or more distinct sets of attributes.

★ ***Essential and Optional Attributes***

In certain situations, it is not required to show the value of an attribute possessed by an entity. We must, therefore, indicate whether a particular attribute-type is *essential* (*required* or *mandatory*) or *optional* i.e., whether or not value must be supplied for the attribute or data element in a record. For example, consider the following entity-type:

User (*entity-type*): (membership # (key attribute), name, address, phone-no (*secondary attributes*)).

We find that the value of the attribute membership # is always a unique value for each member so that each member record in a database can be correctly identified and described. The same is probably true for *name* . We say that membership # and name are essential attributes, whereas the remaining secondary attributes could be nonessential or optional (e.g., the attribute *phone-no* can be considered as *optional*).

4.4.3 Value Sets (Domain) of Attributes

In building a database, one examines real-world entities, their attributes and relationships, and represent their values in the form of a string of symbols (numerical, alphanumerical or alphabetical) in the database. A string of symbols to represent an attribute or data element of a particular entity is called *value* or *content* of the field. For instance 'Rs. 360' is a value of the attribute cost.

Further, while creating a database we need to describe each attribute or data element of the entity-type under consideration. For this purpose, it is necessary to consider and give answer to the following questions:

(a) What kind of value each attribute can take, or what its valid values are?

(b) How the values of an attribute should be recorded or rendered in database record?

Thus, the replies of these questions facilitate the systematic construction of database records and carry out the operation of a database, effectively and efficiently. The set of all possible, valid or legal values for an attribute or data element is called the *domain or value set of the attribute* or the data element. As an example, the domains of the attribute colour could be black, brown, white, yellow, red, purple, etc.; the domain of the attribute name could be the names of all members or employees of a library system. The domain of the attribute employee-name would be the names of all employees of a particular organisation. If in a given context two or more attributes represent same sets of facts (e.g., if the attributes employee-name and name both represent names of same group of people), then in this case such attributes are said to be domain compatible .

4.5 Relationship

A relationship is an association between two and more entity-types. Relationships connect entities and represent meaningful dependencies between them. Relationships specify specific properties or attributes acquired by entities due to some relation or association with other entities. For examples, author-of and *written-by* are the relation indicators between the entity-types author and document and are the associative or link attributes of the entity-types author and document, respectively. Similarly, the terms *member-of* (or enrolled-in) and *enrolled-by* show association between the entity-types library and library member (person). Thus, a relationship can be defined as a logical connection or dependency between occurrence of one entity-type and occurrence of another entity-type. A relation occurrence is an individual entity's connection with other entity or entities concerning to a specific relationship. Entity relationships are described by:

❑ Their dependencieson each other and

❑ The extent of the relationship.

Entity dependencies are of two types:

(a) *Existence Dependency*—i.e., one entity is unable to, exist in the database unless other is first present. For example, orders cannot exist without item(s) ordered and Supplier, to whom the order is placed for the supply of items.

(b) *Identification Dependency*—i.e., an entity cannot be uniquely identified by its own attributes

Identification is possible only through the relationship it has with some other associated entity or entities. For example, in a large group two persons may have same names. Thus, a person can only be identified uniquely within a large group either by his name and parent's name or by his name and institution, subgroup, or unit with which he may be associated.

Extent of dependency includes:

❑ The direction of the relationship

❑ The type of association between them.

4.5.1 Degree or Types of Relationships

All relationships or associations are described by words or symbols that indicate the number of occurrence of one entity that can exist for a single occurrence of the related entity, and vice a versa. In general, there are three types of relationships (associations) or degree of relationships that exit between entities:

❑ One-to-One (1:1) relationship

❑ One-to-Many (1:M) relationship

❑ Many-to-Many (M:M or N:M) relationships.

These three degrees of relationships associations are also called the relationship cardinality. The term cardinality refers to how many of one entity is associated with another. The cardinality ratio specifies the numbers of relationship instances that an entity can participate in.

4.6 Data element

The most elemental piece of data is a data element. It is sometime also called a data item. It is a smallest piece of data, into which data is divided logically, containing no substructure. It forms smallest unit of an entity record that is explicitly identified. An entity record constitutes of data fields and within a data field; a data element forms a sub field and is identified with a unique subfield identifier.

A *data element* represents smallest fact or data about an entity or its attribute, and the data element value is what is actually stored in a subfield of an entity record stored in computer storage. In other words, data elements are smallest units of data type comprising an attribute of an entity or field of an entity record. For example, forename (or first name), middle name, surname (or family name or last name are components of a full name of an entity-type person. The term forename, middle name, or surname (family name or last name) refers to a type of data element. A data element can be a component of simple or aggregated (composite) attribute.

A data element is explicitly or uniquely identified by a identifier name and a unique code called (or subfield identifier) generally constituting of two characters immediately preceding and identifying a subfield or data element. The first character is named the subfield identifier flag; the second character is named the subfield code.

It is the values of data elements (or attribute) of an entity or object that comprise an electronic record of the entity in a database.

Thus, the term data element can be defined as general class or category of data and constitute component of an attribute, or of a field or subfield in a database record.

The various entity-types and their associated attributes or data elements with which we may be concerned for creating a library database are identified and described in the following sections.

5. ENTITY-TYPES ASSOCIATED WITH LIBRARY SYSTEMS

A library is established by an individual, or by an institution or a corporate body to serve a defined groups of users, with a view to make them available information and sources of knowledge they require. Its functional activities are carried out and managed by a defined group of staff. It has connection with other library systems, business enterprises, or corporate bodies, such as booksellers, library vendors, contractors and so on.

Broadly speaking the entity-types associated with library system falls in following categories:

- ❑ Information or knowledge sources. This 'information universe' may be thought of as sum total of recorded information or knowledge in any form, whether bibliographic and non-bibliographic in nature
- ❑ Persons-Library users, employee, businessmen (i.e., individual booksellers, publishers, equipment dealers, etc.)
- ❑ Corporate bodies-Parent bodies, co-operating libraries, and business companies (i.e., booksellers, publishers, equipmentdealers, etc.)
- ❑ Outfits-land, building, equipment, stationary, furniture, software, hardware, etc., commonly purchased or kept by libraries
- ❑ Financial resources.
- ❑ Library units, jobs (posts), projects, databases, and so forth
- ❑ Library functional activities, transactions, events and services.

6. UNIVERSE OF RECORDED INFORMATION/KNOWLEDGE

The entities comprising this 'information universe' having association with the library system may be named as 'documents' and the entity-type 'document' can be subdivided into following sub sets: (a) Document: bibliographic, and (b) Document: non-bibliographic. The definition of entity-type 'document' and its sub-types given below:

6.1 Entity-type: Document

6.1.1 Document

A document is a systematically organised thought, information or knowledge recorded on any medium and in any language or form, such as a book, a serial, web pages, a legal or official paper, a bill of sale, stock register, etc. A document is either bibliographic or non-bibliographic in nature.

6.1.2 Bibliographic Document

A bibliographic document can be defined as a record of a systematically organised thought, information, knowledge, or work on a paper or on any other material, fit for physical handling, transport across space, and preservation through time.

A bibliographic document (also called packages of information) or its part forming the basis for a distinct bibliographical record (or description) is called *bibliographic item*. Bibliographic documents are acquired, processed and maintained by a library system for the use of its members or clients. Web Pages constitute new variety of sources of information that can be accessed through Internet and available information can be downloaded for the use of library users.

6.1.3 Bibliographic Item

Bibliographic item may defined as any bibliographic document or individual piece or part of a bibliographic document which contains recorded knowledge, information, or data. This information entity-type can belong to different hierarchical levels. For example, the largest bibliographic entities are the *libraries and web pages. monographs, serials, series* are example of lower level bibliographic entities than libraries. Further, single volume monographs, multivolume monographs, and composite monographs, as well as distinct portions of these bibliographic entities/items (e.g., articles published in a journal (serial), chapters of a book, parts of a composite book, sub-series of a series, individual monographs or serials published in a series, variety of linked sources of information with a parent web pages, belong to different hierarchical levels of bibliographic or information entity.

The knowledge or information contained in a bibliographic item may be in textual, graphic, numerical form, or multimedia form, which incorporate text and images, audio, video or animations (a combination). The bibliographic item may be in the following format:

- ❑ Hard copy form (printed on pa per, etc.)
- ❑ Audio-visual form (constituting of various type of me dia for pic tures and sound and combination of these)
- ❑ Electronic form (i.e., in machine-readable form)
- ❑ Multimediaform.

The bibliographic items are differentiated or categorised based on various criteria, such as, language; recording script, method or medium; subject, etc. However, all types of bibliographic items can be grouped into the following four principal categories:

6.1.4 Monograph

A non serial bibliographic item is a bibliographic item completes in one part or that intended to be completed, in a finite number of parts (called as multi-volume monograph). A monograph may be published in the form of composite document, that is, with two or more contributions, each with its own title, not forming a continuous exposition and often, though not necessary, by different authors. A composite monograph is either provided with a single generic title to denote all the contributions collectively or without a generic title to denote all the contributions collectively.

6.1.5 Periodical Publication/Serial

A publication in any medium issued in successive parts bearing numeric and/or chronological designations and intended to be continued indefinitely. Each part is generally made up of distinct and independent contributions, not forming a continuous exposition. The term periodical publication or serial includes periodicals; newspapers; annuals (yearbooks, reports, etc.); the journals; abstracting periodicals; proceedings; memoirs, transactions, etc. of societies; and numbered monograph series.

6.1.6 Contribution

An independent work forming a part of a monograph, a serial or web pages for which an autonomous and independent entry is to be made, in addition to its host document.

6.1.7 Series

A set of separate documents or bibliographic items related to one another by the fact that each item bears, in addition to its own title proper, a collective title applying to the set as whole, called 'name of the series' the items are issued, normally successively by one publisher or one sponsoring body or person, usually in a uniform style, and having some similarity of subject or standard or purpose each item is or can be assigned a distinct number, called the serial number.

6.1.8 Web Pages

Web pages are electronic documents available on internet, which contain information in the form of hypertext and hypermedia pages that combine, text, graphics, sound, etc., and have the added feature that the user can follow the links to other documents located virtually any where on the World Wide Web (WWW). Web pages are designed using hyper text mark-up language (HTML) and every individual document or page is assigned a unique address, called the uniform resource locator (URL), which can be used to locate and retrieve the resource on the Internet. Each URL includes the following: Protocol that is used to access the resource, such as 'http' for web, 'gopher' for Gopher, 'ftp' for FTP (File Transfer Protocol), and 'telnet' for Telnet, 'mailto' for electronic mail, and so on.

❑ Internet host or domain on which the resource is accessible

❑ Port number on the host through which the resource is being made available (usually this number is absent and a default is assumed), and

❑ Location, usually the directory path name, within the host at which the resource may be found.

The examples of URLs are:

http://www.mtnl.net.in/

http://www.indiatimes.com/

http://www.w3.org/

http://www.firstmonda.dk/issue/

http://www.unesco.org/webworld/isis/isis.html/

In the internet DNS (Domain Name Server), a particular pattern is followed.

For example in the URL: http://www.mtnl.net.in/, the last two-character code 'in' denotes the country name *India*, where site is located. The string character immediately before the country name 'net' or the last one in case of URL: http://www.indiatimes.com/ 'com' denotes the domain. Here, 'com' denotes that the website is in the field of *commerce*. There are other domain types, such as 'gov' for *government*, 'org' for *non profit organisation*, 'ed' for the web site in the field of *education*, and so on.

The string characters immediately before the domain name is the subdomain. In our examples: (a) 'mtnl', which stands for 'Mahanagar Telephone Nigam Limited', (b) 'W3C', which stands for 'World Wide Web Consortium', an organisation that develops interoperable technologies (specifications, guidelines, software, and tools) to lead the web to its full potential as a forum for information, commerce, communication, and collective understanding, (c) 'indiatimes' is name of a commercial organisation sponsored by Times of India, and (d) 'firstmonday', which is the title of electronic journal, published monthly on first monday of each month, *are the subdomains*. The string character '/issue/' or '/webworld/isis/isis.html/' denotes the *directory path name*, at which the particular resource may be found.

Note: Hypertext Mark-up Language HTML is the language for publishing hypermedia information on Web. It is a subset of Standard Generalised Mark-up Language (SGML) (*refer* ISO: 8879 - 1986, 14).

6.1.8 Non-Bibliographic Documents

A written or printed paper or a set of papers, or a digitised record furnishing information about some transaction(s), event(s), or operation(s) carried out in the library system ca be considered as non bibliographic document.

It can be any official, legal, or business document of the library, or received by it from any other person or organisation, and which is not strictly bibliographical in nature such as, purchase order, bill of supply or sale, payment voucher, user's query, legal contract (supply, maintenance and other agreements made by a library), accession or asset register, bill register, staff attendance register, payroll budget register, expenditure register payment receivable register, payment payable register, procedure manual and so forth.

6.2 Entity-Types: Person and Corporate Body

We find that the entity-types 'person' and 'corporate body' (permanent or ad hoc body) are not only associated with a library system but also with a document and can be defined as follows:

6.2.1 Person

A man, woman or child that is identified by a name and is associated with a library system, a document, or with any other organisation, or business enterprise associated with a library system. This category constitutes of library personnel, potential library employees, library users, personal authors of works, corporate authors, the people linked with or employed in any other organisation connected with a library system, and so forth.

6.2.2 Corporate Body

A group of persons taken collectively-usually organised, or coming together (formally or informally), for a common cause or common action and act or may act, as an entity identified by a particular name. Typical examples of corporate bodies are governments, associations, institutions, business enterprises, non-profit enterprises, government agencies, political organisations, religious bodies and so forth.

6.2.3 Corporate Body (Ad hoc)

Ad hoc corporate body is an assembly or a meeting of individuals, or representative of various bodies for the purpose of deliberation, or formulation and expression of collective opinion, sentiment or statement on topics of common interest, or an ad hoc meeting of representatives of a corporate body that constitutes its legislative or governing body.

An entity-type person, a library, an institution, a business enterprise or a corporate body, having relationship with a library system can be its:

- ❑ Parent body-a person, institution or a corporate body with whose courtesy, authority, and/or finance or under whose auspices library system is established.
- ❑ Organ or sub-unit of library system
- ❑ Co-operating library
- ❑ Employee
- ❑ Member/user.

Supplier: A person, company, business enterprise or an institution that sells or supplies various items: books, serials, furniture and other equipment, to the library on order, as gift or exchange to a library on order as gift or exchange to a library.

Contractor: A person or a company that repairs or maintains library equipment, building, etc.; bind library books; arranges to supply goods to the library or provides any other service on contractual basis.

Similarly an entity-type person, a corporate body, or a conference having relationship with a document-bibliographic or non-bibliographic in nature can be its:

Author/originator: The person or corporate body chiefly responsible for the creation of the intellectual or artistic contents of a work.

Corporate author: Corporate body as author, the responsibility for the thought and expression constituting the work resting solely on it or any organ of it, and not in the private capacity of any person or persons forming part of or holding office in or in any other way connected with the body.

Collaborator: A person or corporate body associated with a work and/or its authors or authors in a secondary capacity not amounting to authorship.

Sponsor: Person or corporate body with whose courtesy, authority, and/or finance or under whose auspices a work or a bibliographic item is published, though responsibility of thought and expression constituting the work does not at all rest with the said person or corporate body, but rests solely on the author of the work as a whole or respective authors of different contributions in the bibliographic item.

Publisher: A person, company or an organisation that makes available recorded information to people in the form of books, periodical publication, etc.; or publishes or produces various types of documents or packages of information and sell them in the market.

7. IDENTIFICATION AND DESCRIPTION OF ATTRIBUTE OF ENTITIES OF A LIBRARY SYSTEM

As discussed earlier, to represent facts or data about an entity-type in a database record, one has to recognise and define its relevant attributes that collectively describe an entity-type, the data element comprising each attribute, and the corresponding fields and sub fields that may comprise a database record structure. Select and designate among them those are considered essential/ (mandatory) or optional for including their values for creating records of various entities-types.

Assign field tags and sub field identifier to the defined fields and sub fields (data elements), respectively, which serve as means of identifying the values contained in the fields and sub fields in a machine-readable record. This is a difficult task, as the entities associated with a system are always in state of flux or change. The change in the state of an entity also brings change in its properties or attribute and relationships. In a new state, an entity may loose or gain certain attributes and relationships, and/or the values of certain attributes may change. In a library, for instance, the entity 'monograph' proceeds through a number of different states. It is first acquired-selected, ordered, received, accessioned, then technically processed -classified, catalogued, and stored in the stack. It is picked up by a library member for study in the library or to get it issued on loan and finally to be returned. It may ultimately be withdrawn from the library collection to be destroyed or sold, if it is worn out. One can outline or chart the change in the state of a particular entity in a system using Entity Life History (ELH) diagramming technique, originally designed by Rosenquist[27] and consequently, identify and describe the relevant attributes which collectively describe an entity-type in different states.

A state is a moment in the life of an entity or object. For instance, a bibliographic item may be characterised as being selected item, after its selection (an event) for purchase until its order (an event) is placed with a vendor. After an order (an event), the state of a given bibliographic item also changes. It turns into is an ordered item. A state is, thus, a moment or interval of time between two events (selection and order, in our example) affecting upon an entity or object (bibliographic item, in our example). An event is something that happens at a moment of time that changes the state of an entity. For conceptual purpose, we assume that an event has no duration.

Events may be related, in the sense that one logically precedes or follows another. Events may be unrelated, in which case we say they are concurrent. For example, a typical event may occur when two or more library members reserve a loaned bibliographic item. The event concurrent in this case will be reservation of the loaned item.

The necessary attributes (data elements) for creating records of bibliographic documents have been well defined in various library-cataloguing codes. We shall, therefore, make attempt to identify and define all the possible attributes and the associated data elements of each attribute of various entity-types other than bibliographic items, whose values shall normally comprise an entity record in a library database.

It may also be pointed out that the libraries serve a defined group of users, or appoint a defined group of staff, book vendors, suppliers of items, and so on. One of their first tasks is therefore to establish criteria by which membership eligibility, appointment of a staff member, book vendor, equipment dealer, etc., will be considered.

For example, a university library may define its user groups as those recognised as academic staff, administrated staff, students of the institution and/or of colleges affiliated with it. A public library may define its user groups based on different criteria. Each user group may of course be further sub-divided by reference to other criteria -as when a student group may be sub-divided into undergraduates, postgraduates, and research scholars, to be treated as distinct groups for loan of documents by a university library system. Thus, we find that there is bound to be difference in the approaches and structural design of databases for different kinds of library systems.

However, our endeavour here is to identify the entity classes mostly associated with different types of library systems and specify their most likely attributes which must be considered for developing databases of all types of library systems. The decision to select or add among those defined in the code is left to the database designer of an information system pertaining to a particular library system.

In part-II of this paper, the attributes and data elements of the entities that are relevant or needed for creating the library database records are identified and described in annexure 1-10 and also some of the rules of classified code are examined and restated. On the basis of the restated rules and the list of identified attributes and data elements of various entity-types associated with a library system, we will subsequently prepare a generalised list of fields and their respective subfields that may be commonly used to define database structure of an integrated database system of a library system.

We will also identify and specify whether a particular field or subfield is mandatory or optional, repeatable or non-repeatable, and provide them unique field tags and subfield identifiers.

These descriptions can be considered as a framework for designing and developing new code, which may constitute part of the revised code for developing library information system or databases.

REFERENCES

1. Gorman, Michael & Winkler, Paul W. (Ed). Anglo-American Cataloguing Rules. Rev Ed. 2. American Library Association, Chicago, 1988.
2. Chen, Peter Pin-Sen. The entity-relationship model-towards a unified view of data. *ACM Transaction on database Systems,* 1976, **1**(1), 9-36.
3. Chen, Peter Pin-sen, Ed. Entity-Relationship approach to information modelling and analysis. North-Holland, Amsterdam, 1983.
4. CCF/B: the common communication format for bibliographic information. Paris, UNESCO. 1992. (PGI-92/WS/9).
5. CCF/F: the Common Communication Format for factual information. Paris, UNESCO, 1992. (PGI-92/WS/9).
6. Codd, E. F. Extending the database relational model to capture more meaning. *ACM Transactions on Database,* 1979, **4**(4), 397-434.
7. Codd, E. F., Relational database A practical foundation for productivity. *Commun. ACM*., 1982, **25**(2), 109-17.
8. Dates, C. J. An introduction to database systems. Ed. 3. Addison-Wesley, Reading, MA. 1981.
9. Hopkinson, Alan, Comp. Implementation notes for users of the common communication format for factual

information (CCF/F). UNESCO, Paris. 1993. 47p.

10. International Federation of Library Associations and Institutions (IFLA). Form and structure of corporate headings. IFLA International Office for UBC, London, 1980.

11. International Federation of Library Associations and Institutions (IFLA). Names of persons: national usage for entry in catalogue. IFLA International Office for UBC, London, 1980.

12. IS: 2381-1978. Recommendation for bibliographical references essential and supplementary elements (Rev.1.). Indian Standard Institution, New Delhi, 1979.

13. ISO: 2709-1981. Documentation: format for bibliographic information interchange on magnetic tape. Ed. 2. International Organisation for Standardisation (ISO), Geneva, 1981.

14. ISO: 8879-1986. Information processing: text and office system: Standard genralised markup language (SGML). International Standards Organisation, Geneva, 1986.

15. Kashyap, M. M. Common communication format for online ordering, a proposal for book trade and libraries. Paper presented at 15th Annual Convention and Conference of the Society for Information, 18-20 January 1996, Bangalore. *In* Digital libraries: Dynamic storehouse of digitised information, edited by N. M. Malwad, *et al*. New Age International, New Delhi, 1996. pp. 75-89. 26-33.

16. Kashyap, M.M. Database system: design and development. Sterling Publishers, New Delhi, 1993. x+382p.

17. Kashyap, M.M. Database technology: basic concepts and terminology. *Library Herald*, 1991, **30**(1), 1-19.

18. Kashyap, M.M. Integrated database design for a Library system: employing library techniques developed by Ranganathan and CDS/ISIS DBMS. *Journal of Library and Information Science*, 1993, **18**, 82-141.

19. Kashyap, Madan Mohan. Online library catalogues and standards for common application in a network environment: Paper presented at XVI IASLIC Seminar, Bombay, 1994. *In* Networking of Libraries: Problem and Prospects. IASLIC Seminar, Bombay, 1995, **16**, 1-25.

20. Kashyap, Madan Mohan. Computer-based library information system designing techniques. Sterling Publishers, New Delhi. 1999. 259-60.

21. Mair, D. The theory of relational database. Computer Science Press, Rockville, Mary Lands, 1983.

22. Neelameghan, A. Application of Ranganathan's general theory of knowledge classification in designing specialised databases. *Libri,* 1992, **4**(3), 202-26.

23. Neelameghan, A. Concept categorisation and knowledge organisation in specialised databases: A case study. *International Classification,* 1991, **18**, 92-97.

24. Oxborrow, E., Databases and database management systems: concepts and issues. Ed. 2. Chartwell-Bratt, Chichester, 1981.

25. Ranganathan, S.R. Classified catalogue code: with additional rules for dictionary catalogue code. Ed. 5. 1988.

26. Ranganathan, S.R. Classified catalogue code: with additional rules for dictionary catalogue code. Ed. 5. 1988. Chap BF. 39-46.

27 Rosenquist, C.J. Entity life cycle models and their applications to information systems development. *Computer Journal,* 1982, **25**, 307-15.

28. Simmons, Peter & Hopkinson, Alan, Ed. CCF: common communication format. Ed. 2. (PGI-88/WS/2). UNESCO, Paris, 1988.

29. Teorey, J.T., *et al.* Logical design methodology for relational database. *Computing Survey*, 1986, **18**(2).

30. Unesco, Division of Software Development and Applications, Office of Information Programmes and Services. Mini-micro CDS/ISIS Reference Manual (version 2.3). Unesco, Paris, 1989.

31. Vossen, G. Data models, database language and database management systems. Addison-Wesley, Wokingham, 1991.

Contributor: Prof. Madan Mohan Kashyap has retired from Dept. of Library & Information Science, University of Delhi. His residential address is 201 Kadambari Niwas, Kadambari marg, Sector-9, Rohini, New Delhi-110 085. Tele (R): 786 2699; e-mail: mmkashyap@bol.net.in; kashyap1@indiatimes.com