Structuring Database for Accounting

14

LEARNING OBJECTIVES

After studying this chapter, you will be able to :

- *identify the resources of MS ACCESS as DBMS;*
- *explain basic concepts of database system;*
- *express accounting reality in the context of Entity Relationship (ER) Model;*
- *transform ER presentation of accounting reality into database;*
- *develop database design for computerised system using Relational Data Model;*
- *formulate basic queries for retrieving accounting data and information.*

In the earlier chapters, you have already learnt that accounting of transactions are documented with vouchers. Let us consider a few accounting transaction to understand as to how these vouchers are used.

On April 01, 2014 M/s Kshipra Computers commences business with initial capital of Rs.5,00,000, which is deposited into bank. Recall the journal entry that is recorded using manual accounting system. This journal entry has data contents that are filled-up using a simple transaction voucher, which is prepared by Smith and authorised by Aditya.

M/s Kshipra Computers

TRANSACTION VOUCHER

Voucher No: 01 **Date:** Apr. 01, 2014

Debit Account: 642001 Bank Account

Credit Account: 110001 Capital Account

Amount in Rs. : 5,00,000

Narration: Commenced business by depositing initial capital into bank

Authorised By: Aditya **Prepared By** Smith

Fig. 14.1 : *A sample transaction voucher to document simple transactions involving one debit and one credit*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

The same transaction can also be documented using a credit voucher that is capable of recording multiple credits against one debit, as shown below:

CREDIT VOUCHER

Voucher No: 01 **Date:** April 01,2014

Debit Account: 642001 Bank Account **M/s Kshipra Computers**

Credit Accounts

S.No	Code	Name of Account	Amount	Narration
1	110001	Capital Account	5,00,000	Commenced Business
		Total Amount	5,00,000	

Authorised By: Aditya **Prepared By** Smith

Fig. 14.2 : *A sample voucher for multiple credits against one debit*

Now consider the following transaction :
On April 03-2014 M/s Kshipra Computers bought goods costing Rs.50,000 from M/s R.S. and Sons, paying Rs.2,000 as cartage to M/s Saini Transports. This transaction involves multiple debits of accounts with one account being credited. The debit voucher that is used to document this transaction appears as follows :

DEBIT VOUCHER

Voucher No: 05 **Date:** April 03, 2014

Credit Account: 642001 Bank Account **M/s Kshipra Computers**

Debit Accounts

S.No	Code	Name of Account	Amount	Narration
1	711001	Purchases	50,000	Purchases from R.S & Sons
2.	711003	Carriage Inwards	2,000	Paid to M/s Saini Transports
		Total Amount	52,000	

Authorised By: Aditya **Prepared By** Smith

Fig. 14.3 : *A sample vouchers for multiple debits against one credit*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

The process of computerised accounting involves identifying, storing and retrieving the data content of an accounting transaction. This requires a mechanism to store such data content of vouchers in a manner that allows its easy and convenient retrieval as and when required. This is achieved by designing suitable database for accounting. Such a database consists of inter-related data tables that are structured in a manner that ensures data consistency and integrity. In this chapter we shall discuss the basic concepts of database system of accounting.

14.1 Data Processing Cycle

In order to understand the dynamics of database design, let us understand the data processing cycle in the context of accounting. Data processing involves the technique of collecting, sorting, relating, interpreting and computing data items in such a manner as to provide meaningful and useful information for decision-making. The necessary steps involved in data processing cycle are data capturing, inputing, processing and generating information available to the user. Data processing cycle, when thought of in the context of accounting, requires a series of steps that have been described below briefly :

(i) *Source Documents* : The first step is to capture accounting data from transaction(s) so as to prepare a document, called voucher (as already stated earlier), that expresses and documents an accounting transaction. The relevant accounting data is set out in the voucher, the sample of which is shown in figures 14.1 to 14.3. These documents are so designed as to permit the recording of accounting data in a systematic manner.

(ii) *Input of Data* : The accounting data contained in vouchers is to be entered in a computer's storage device. This is achieved by using a pre-designed Data Entry Form. This data entry form is designed in a manner that it is similar to physical voucher document. The data entry form is designed using software and it is made to appear on the computer monitor so that the data is entered.

(iii) *Data Storage* : A suitable data storage structure is required to provide for a blank data record as shown below:

Code	*Name*	*Type*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

The above blank record that is used for storing the input of data pertaining code of account, name of account and the category type to which it belongs is shown below as :

Code	*Name*	*Type*
11001	Capital Account	4
711001	Purchases Account	1

Hypothetically, the category type 4 above refers to *Liabilities* and the category type 1 indicates *Expenses*. The data storage structures (also called data tables) are created as a part of structuring database for accounting.

(iv) *Manipulation of Data* : The stored data is manipulated for necessary transformation to generate final reports. Such transformed data may be stored separately and subsequently used for generating final reports. Alternatively, the transformed data can be directly presented in the form of a report.

(v) *Output of Data* : The accounting reports such as ledger, trial balance, etc. are obtained in a pre-designed format by accessing the transformed data.

Now that you have understood the way data content is stored in structured manner, we shall discuss how the data structures are designed in consonance with the data content that emerges from accounting transactions.

14.2 Designing Database for Accounting

Both computerised and computer-based AIS require a definite data structure for storing the accounting data. As already mentioned, the databases are used for storing accounting data. The process of designing database (for accounting) begins with a reality (or accounting reality) that is expressed using elements of a conceptual data model. The process of designing a database for accounting is best described through a flow chart (Figure : 14.4).

Reality : It refers to some aspect of real world situation, for which database is to be designed. In the context of accounting, it is accounting reality that is to be expressed with complete description.

ER Design : This is a formal blue print, with a pictorial presentation, in which Entity Relationship (ER) Model concepts are used to represent description of reality.

Relational Data Model : It is representational data model through which ER design is transformed into inter-related data tables along with the restriction in the form of rules that are specified to ensure the consistency and integrity of stored data.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

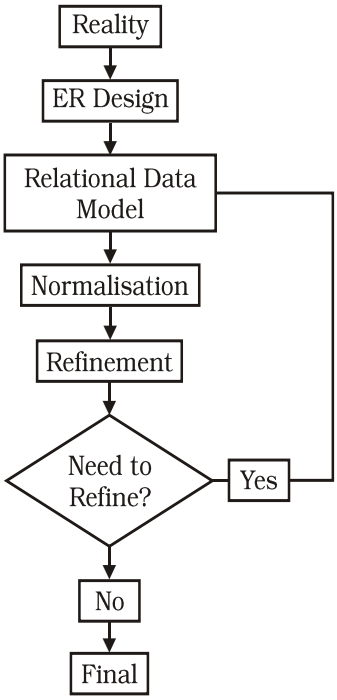


Fig. 14.4 : *Flow Chart depicting the process of designing a database for accounting*

Normalisation : This is process of refining a database design (that consists of inter-related data tables) through which the possibility of duplicate or redundant data items is reduced or eliminated.

Refinement : This is the outcome of the process of normalisation as mentioned above. The final database design is arrived at after the process of normalisation is completed.

14.3 Entity Relationship (ER) Model

It is a popular conceptual data model, which is mostly used in database-oriented applications. The major elements of ER Model are entities, attributes, identifiers and relationships that are used to express a reality for which a database is to be designed. The model is best depicted with the help of ER symbols, the list and description of which is shown in figure 14.5. While preparing an ER Diagram, the following symbols are used to represent different types of entities, attributes, identifiers and relationships :

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

The elements of ER model that are meant to describe and display the reality are discussed in the context of an accounting reality given below :

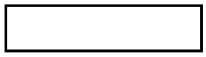
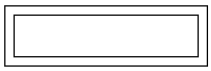
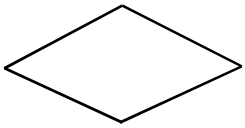
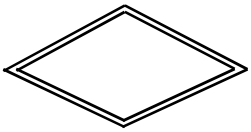
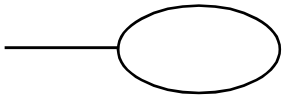
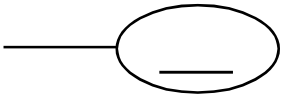
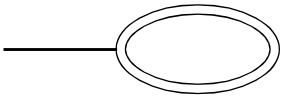
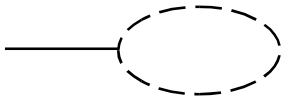
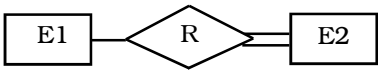
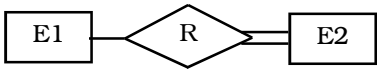
Meaning	**Symbols**
Entity Type as Rectangular Box	
Weak entity Type as double lined Rectangular Box	
Relationship Type as diamond shaped Box	
Identifying relationship Type as double lined diamond shaped Box	
Attribute names enclosed in ovals and attached to their entity type by straight lines.	
Key attribute names enclosed in ovals and attached to their entity type by straight lines.	
Multi-valued attributes by double ovals.	
Derived attributes by dashed line Ovals	
Total participation of **E2** in **R**	
Cardinality Ratio 1 : N for **E1 : E2** in **R**	

Fig. 14.5 : *Symbols used for constructing an ER diagram*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Accounting Reality Describing the System of Accounting

Using a hypothetical example of accounting system of an organisation, following statements of reality becomes the starting point of discussion in describing the ER Model concepts.

Example Reality :

- Accounting Transactions of an organisation are documented using a voucher.
- Each vouchers is assigned a serial number, which begins with "01" indicating first vouchers of the accounting period. There is only one simple transaction voucher used for documenting the transactions (See Figure : 14.1).
- Each **voucher** documents date of transaction, account name along with its account code for debit as well as credit entry.
- Each voucher indicates the amount and narration with respect to accounting transaction.
- **Support documents** such as bills, receipts, contracts, etc. also may be attached to an accounting voucher.
- Each Voucher is prepared by a particular **Employee** and authorised by another employee.
- There is an exhaustive list of **Accounts** with respect to which the transactions are documented. Each Account carries a unique numeric code with its width equal to six digits.
- Each Account is classified as belonging to one of the **Accounts Types**: Expenditure, Income, Assets and Liabilities.

Fig. 14.6 : *Example reality on accounting system*

14.3.1 Entities

Anything in the real world with independent existence is called **entity** such as an **object** with *physical existence* (e.g. car, person, house) or *conceptual existence* (e.g. a company, job, university course, account, voucher). In the context of above accounting reality, there exist five entities: Accounts, Vouchers, Employees, AccountsType and SupportDocuments. The accounting data is captured through these entities.

14.3.2 Attributes

Attributes are some properties of interest (or characteristics) that further describe the entity such as height, weight and date of birth in case of a *person* and code and name in case of *accounts*. An entity has a **value** for each of its attributes, which is the data stored in the database.

There are several types of attributes of an entity that have been described as follows :

(i) *Composite vs. Simple (or atomic) attributes* : The composite attributes can be divided into smaller sub-parts to represent some more basic

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

attributes with independent meanings. The simple attributes cannot be further sub-divided. For example, Name of a person that is normally sub-divided into First Name, Middle Name and Last Name is a composite attribute. Height of a person is a simple attribute as it is devoid of further sub-division.

(ii) *Single-valued vs. Multi-valued Attributes* : An attribute with a single value for an entity is single-valued as opposed to those which multiple values. For example, height of a person is single-valued attribute while qualifications of that person are a multi-valued attribute.

(iii) *Stored vs. Derived Attributes* : Two or more attributes may be related in such a way that one or more becomes basic while the other becomes dependent on that basic attribute. For example, date of birth of a person is a stored attribute while age of that person is derived attribute.

(iv) *Null Values* : Absence of a data item is represented by a special value called null value. There are three situation which may require the use of null values
- When a particular attribute does not apply to an entity;
- Value of an attribute is unknown, although it exists;
- Unknown because it does not exist.

(v) *Complex Attributes* : The composite and multi-valued attributes may be nested (or grouped) to constitute complex ones. The parenthesis () are used for showing grouping of components of composite attributes. The *braces* {} are used for showing the multi-valued attributes

In the context of the example on accounting reality, the following attributes specific to each entity types have been stated below as :

Entity Type	*List of Attributes*
AccountsType	CatId, Category
Accounts	Code, Name, Type
Employees	EmpId, Fname, Minit, Lname, SuperId
Vouchers	Vno, Date, Debit, Credit, Amount, Narration, AuthBy, PrepBy
SupportDocuments	Sno, dDate, Name

AccountsType is a conceptual entity that is meant to express the various categories of accounts in accounting system. The CatId is an attribute of AccountType entity, the value of which is used to identify the category of accounts.

Accounts is a conceptual entity that is meant to express various accounts, each one of which belongs to a particular category of accounts in Accounts Type Entity. Every account is assigned a unique code by which it is

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

identified. The Name attribute specifies the name of account and Type refers to the type of account (or category of account) as mentioned above.

Employees is a physical entity that is meant to express the various employees who are in some way connected with the accounting system. The EmpId (Employee ID) attribute is meant to identify an Employee; Fname, Minit and Lname are respectively the first, middle and Last names of an employee; and SuperId refers to EmpId of the immediate boss of an employee.

Vouchers is an entity that expresses various transactions vouchers. It is attributes together provide the structure of transaction data.

SupportDocuments is an entity, which expresses various support documents that may be attached with a particular voucher of a transaction. Sno attribute of this entity specifies the serial number of support document attached, dDate specifies the document date and Name specifies the name of document that is attached with the voucher.

(vi) *Entity Types and Entity Sets* : An **Entity Type** is defined as a collection of entities, which share a common definition in terms of their attributes. Each entity type is assigned a name for its subsequent identification. The attributes of entity type are used to describe it in the database. The values of attributes of an entity belonging to entity type are known as **Entity Instance**. For example, (110001 Capital Account 4) is an entity instance of an account whose code = 110001, Name = Capital Account and Type = 4. An **Entity Set** is a collection of all entity instances of a particular entity type. An Entity Type is described by a set of attributes called "schema". The set of entities pertaining to a particular entity type share the same set of attributes. The collection of entities of a particular entity type is grouped into entity set, called the *extension* of the entity type. For example,

Entity Type : Accounts
Intension (or structure) of entity type

Code	*Name*	*Type*

Entity Set: Collection of entity instances of an entity type "Accounts"
Extension (or instances) of entity type

110001	Capital Account	4
221019	Jain & Co.	4
221020	Jayram Bros.	4

Fig. 14.7 : *Examples of entity type and entity set*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(vii) *Value Sets of Attributes* : Each simple attribute is associated with a **value set,** which specifies the set of possible values that may be assigned to a particular attribute. For example, the value set of voucher date is all those dates that fall within the dates valid for a given accounting period. Similarly, if accounting reality states that each code of an account is numeric with its width equal to six digits, its possible value set shall be 000001 to 999999. The value set as described above is called domain of values.

14.3.3 Identifier (or Key Attributes of an Entity Type)

Almost every entity type has one of its attributes, which contains unique values for identifying the entity instance. For example, RollNo as attribute of Entity type *students* has unique values through which a student instance can be identified. Similarly, *Code* is a key attribute of entity type *Accounts* because its data values are required to be unique.

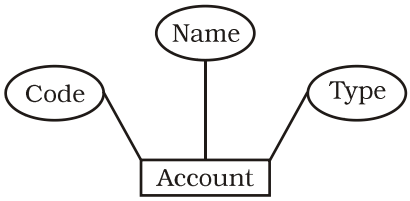


Fig. 14.8 : *Diagrammatic presentation of an entity type accounts with code as key attribute*

Some times two or more such attribute together (called composite key) may constitute such distinct values. For example, the student entity type that has entity instances across several sections of a class in a school shall require a composite key of attributes (Sections and RollNo). But in any case, it is a constraint that does not allow any two-entity instances from having the same value for the key attribute at a point of time. Some entities may have more than one Key attribute. The entity types, which do not have a key attribute at all are called weak entities.

14.3.4 Relationships

Relationship among two or more entity types represents an interaction among their respective entities. Whenever *an attribute* (say Debit) of one entity type (say vouchers) refers to another entity type (say Accounts), there exists a relationship between these entities (Vouchers and Account).

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

For example, vouchers and accounts are related in two ways: vouchers contain debit account(s) and vouchers contain credit account(s). In ER Model, these *references* are represented as explicit *relationships* rather than attributes.

(i) *Types of relationships* : Whenever entities from different entity types are related to one another in a particular manner, they constitute a relationship type. The relationship prepared by between the two entity types vouchers and employees associates each voucher with the employee who prepared it. Similarly, the relationship authorised by between the two entity types vouchers and employees associates each voucher with the employee who authorises it. Each relationship instance of prepared by (short named as **PrepBy**) associates one voucher entity with one employee entity. In ER diagrams, relationship types are displayed as diamond shaped boxes, connected by straight lines to the rectangular boxes, which represent the participating entity types.



Fig. 14.9 : *Diagram showing binary relationship between vouchers and employees*

(ii) *Degree* : The degree of a relationship type is the number of participating entity types. A relationship type of degree two is called binary and that of degree three is called ternary. A VOUCHER (entity), Authorised_by (relationship) and EMPLOYEES (entity) together signify a *binary* relationship. A SUPPLIER (entity) SUPPLY (relationship) PARTS (entity) to PROJECT (entity) signify a *ternary* relationship because three entities, namely supplier, parts and projects are participating in supply relationship in any transaction.

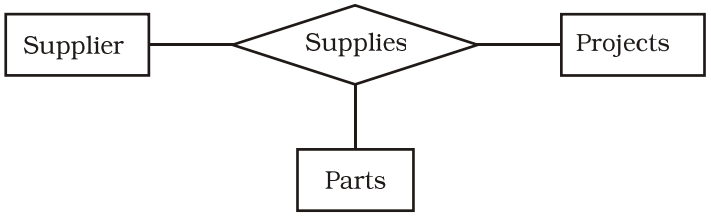


Fig. 14.10 : *Diagram showing ternary relationship between suppliers, parts and projects*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(iii) *Role Names* : Each entity type that participates in a relationship type plays a particular role in the relationship. The role name signifies the role that a participating entity of an entity type plays in each relationship instance. In PREPARED BY relationship type, EMPLOYEE plays the role of *document creator* and voucher plays the role of *document created*.

(iv) *Structural Constraints* : The reality may impose certain constraints (or restrictions) that may limit the possible combinations of entities, participating in a given relationship set. These are of two types : *Cardinality Ratio and participation.*

- *Cardinality Ratios* for binary relationship specifies the number of relationship instances that an entity can participate in. In PREP_BY binary relationship type, VOUCHER:EMPLOYEE is of cardinality ratio N:1 implying thereby that a set of vouchers can be created by a particular employee. The possible cardinality ratios are one to one (1:1),one to many(1:N),many to one(N:1), and many to many(N:M).
- *Participation* constraint specifies as to whether the existence of an entity type depends on its being related to another entity via a relationship type or not. The two types of such constraints are: total and partial. Whenever semantics of reality require that every entity of an entity type must relate to another entity type, such an entity can exist only if it participates in that specific relationship. Such a participation is called total participation. For example, the participation of ACCOUNTS in CLASSIFY relationship is total participation. This is because every account must refer to at least one of the accounts type or a category of accounts. This participation is also called *existence dependency*. Since every employee is not expected to prepare at least one of the vouchers, the participation of employee in PREPARED BY relationship is partial, implying that some of employee entities are related to the voucher entity via PREPARED BY relationship. In ER diagram, *total participation* is displayed as *double line* connecting the participating entity type to the relationship, whereas *partial participation* is represented by a *single line.*

14.3.5 Weak Entity Types

Entity Types, which do not have identifier (or key attributes) of their own are, called *weak entity* types. Such entity types are identified by being related to specific entities from another entity type in combination with some of their attribute values. These other entity types are called *identifying or owner entity* type. Accordingly, the relationship type that relates a weak entity type to its owner is called *identifying relationship* of the weak entity.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

A weak entity type always has a total participation constraint (existence dependency) with respect to its identifying relationship because it cannot be identified without its owner entity. For example, a voucher may be accompanied by a set of support documents such as bills, issued by other parties to the transaction, details of which need be stored. Such SUPPORT" DOCUMENT entity type which is used to keep track of support documents attached to each voucher via 1:N relationship, is a weak entity. This is because they are identified as distinct entities only after determining the particular voucher. A weak entity type normally has a partial key, which is a set of attribute that can uniquely identify weak entities that are related to the same owner entity. Assuming that two support documents of a voucher do not have the same *document Id*, the said Id can be a good partial key. Otherwise a composite attribute of all the weak entity's attributes will be the partial key.

Initial Conceptual Design for an Example Reality : Using a hypothetical example of an accounting system, as already stated above in Fig: 14.6, following initial design based on ER Model concepts becomes the starting point of illustration.

Conceptual Design : According to the requirements listed in example reality, there exist five entities: Vouchers, Accounts, Employees, SupportDocuments and AccountsType·

- An entity type *Vouchers* with attributes Voucher No, Serial No, Voucher Date, Debit Account, Credit Account, Amount, Narration, authorised by, prepared by are used for storing accounting data of a transactions. Debit and amount are multi-valued attributes for debit vouchers and credit and amount are multi-valued for credit vouchers. *Voucher No and Sno* together constitutes the only key attribute of entity type vouchers. Therefore, it is specified to be unique.
- A Conceptual entity type *Accounts* with attributes Code, Name and Type is used for keeping and maintaining a record of all accounts. Both *Code* and Name qualify to be the key attributes because of being specified as unique.
- An Entity Type *Employee* with attributes Employee ID (EmpId), Name, Address, Phone, ID of immediate boss (SuperId) is used to maintain records of employees in the organisation. Name is a composite attribute with its simple attributes as: First Name (Fname), Middle Initial (Minit) and Last Name (Lname). The *EmpId*, specified to be unique, is the key attribute. SuperId indicates the EmpId of the controlling officer, the immediate boss.
- An entity type, *Accounts Type* with attributes CatId and Category is used to maintain records of various categories of accounts so that each of the accounts as stored in accounts entity are able to find their suitable place in financial accounting reports: profit and loss account and also the balance sheet.
  An entity type called Support with attributes Sno. and Name is used to maintain records of all the support documents, which are annexed to the accounting voucher.

Fig. 14.11 : *Details of initial conceptual design based on example reality*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

14.3.6 ER Presentation of Accounting Reality

The example reality shown at Figure: 14.11 can be shown below diagrammatically by using the ER notations.:

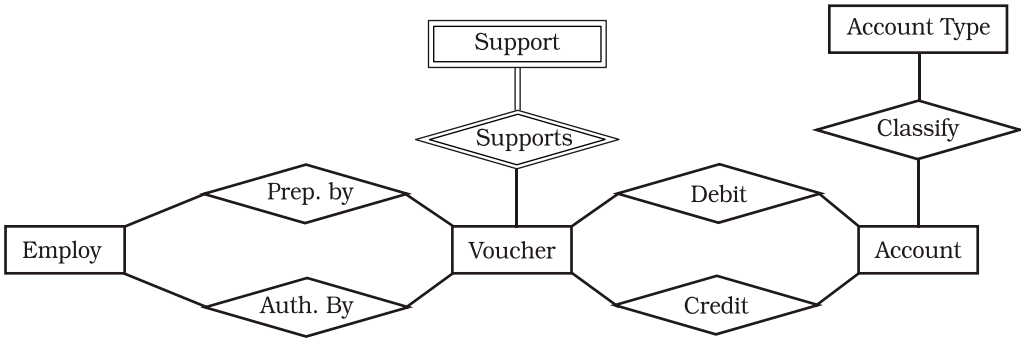


Fig. 14.12 : *ER Schema diagram for accounting database*

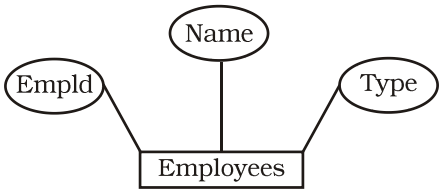


Fig. 14.13 : *Diagrammatic presentation of an entity type accounts with code as key attribute*

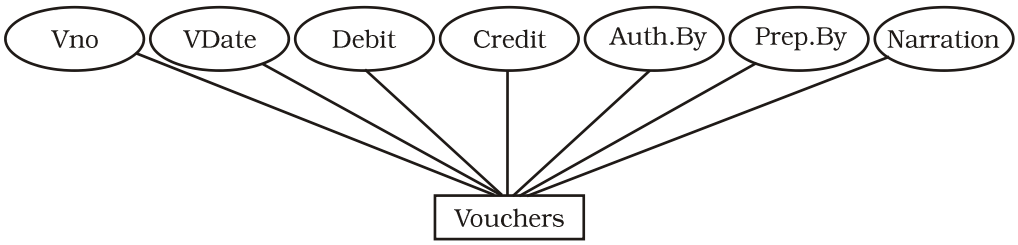


Fig. 14.14 : *Diagrammatic presentation of an entity type accounts with code as key attribute*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

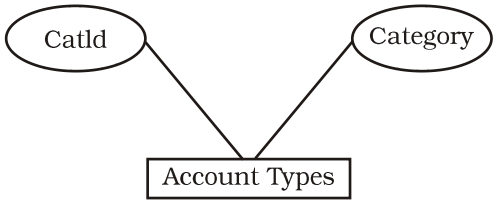


Fig. 14.15 : *Diagrammatic presentation of an entity type accounts with code as key attribute*

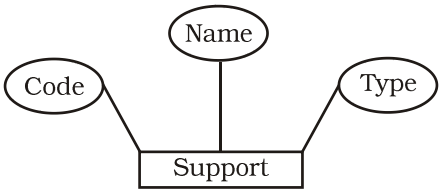


Fig.14.16 : *Diagrammatic presentation of an entity type accounts with code as key attribute*

14.4 Database Technology

It refers to a set of techniques that are used to design a database. These techniques use certain concepts, which are crucial to the creation of structure and development of the design. These concepts are: Reality, data, database, information, DBMS and database system. A brief description of these concepts is given below:

(a) *Reality* : It implies some aspect of the real world. It consists of an organisation, its different components and the environment in which the organisation exists and operates. Any organisation includes people, facilities and other resources that are organised to achieve certain goals. Each organisation operates within an environment. While operating, the organisation *interacts*, influences and gets influenced by the environment.

An organisation may be viewed as a system consisting of several components called its sub-systems. Each of these sub-systems follows certain procedures and continuously interacts with each other and their external environment to accomplish the goals of organisation. During the course of their interaction, events take place, which take the shape of data items. These sub-systems communicate continuously with AIS to provide data and seek information. A part of AIS is Financial Accounting System, which is designed for processing accounting transactions. For example,a firm uses a voucher to document an accounting transaction. The contents of voucher consist of accounting data, which need be stored in an organised manner.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

This continuous interaction results in real world transactions. These transactions are analysed with a view to identify the components called data items. A data item is the smallest named unit of data in an information system. In a transaction, the names of accounts (or their accounting codes), date of transaction, amount, etc. is all data items.

(b) *Data* : Data are known facts that can be recorded and which have implicit meaning. Data represent facts concerning people, places, objects, entities, events or even concepts. Data can be quantitative and qualitative or they can be financial and non-financial in character. Consider the following transaction :

April 01, 2014 Commenced business with Cash Rs. 5,00,000.

This transaction, before being recorded through a Transaction Voucher, as shown in figure 14.1, need be split up into its data contents as "01", 01-Apr-14, 642001, Bank Account, 110001,Capital Account, Rs.5,00,000. Data are not useful for decision-making unless they are processed to suit to the requirements of decision-making situation.

(c) *Database* : The data, after being collected, has to be stored so that different people can use them. This requires the creation of a database. A database is a shared collection of interrelated data tables, files or structures, which are designed to meet the varied informational needs of an organisation (See Example database in figure 14.19. It has two important properties (or characteristics): *one* it is integrated and *second* it is shared. *Integrated property* implies that distinct data tables have been logically organised. The purpose is to reduce or eliminate redundancy (or duplicity) and also to facilitate better data access. The *shared property* means that all those who are authorised to use data/information have access to relevant data. Thus, a database is a collection of related data that represents some aspect of the real world (called *mini-world* or Reality). Accordingly, accounting database is a collection of related accounting data to represent some aspect of an accounting information system. Database is designed, built and populated (or loaded) with data for a specific purpose.

(d) *Information* : refers to data that have been processed and organised in a form, which is suitable for decision-making. The raw data when processed in accordance with decision usefulness of a decision-maker becomes information. In other words, information is a data that have been processed and refined and then presented in a format that is convenient for decision-making or other organisational activities.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

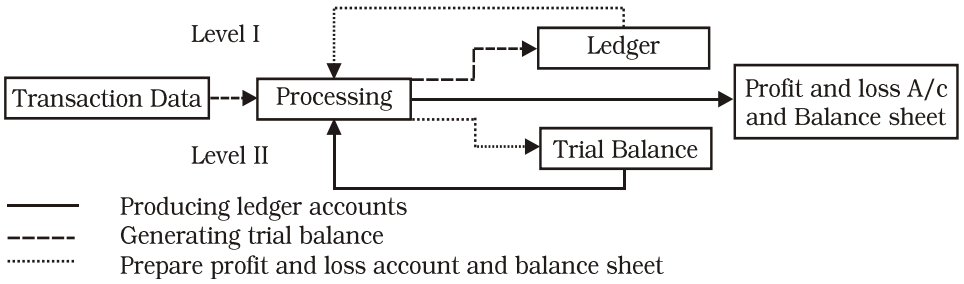


Fig. 14.17 : *The diagram showing the transaction data processing and information levels*

However, information may be viewed as data at one level. But when it is processed keeping in view the requirements of decision situation, it becomes information at another level. For example, accounting data at transaction level is processed to produce balances of each account. The balances are summarised to prepare the trial balance. The amounts given in trial balance constitute data to produce profit and loss account and balance sheet.

(e) *Database management System* (DBMS) is a collection of programs that enables users to create and maintain a database. Formally, it may be defined as a general-purpose software system that facilitates the processes of defining, constructing and manipulating (or processing) databases for various applications. General-purpose software is defined as a set of programs, which are designed and developed for a community of users and not for any particular application with respect to a particular user.

14.5 An Illustration of Accounting Database

Consider an example of **ACCOUNTING** database for maintaining data pertaining to accounting transactions, support documents, accounts and employees with which the students of accounting are familiar. Figure 14.18 shows below the database structure and some sample data for this database, depicting the following transactions :

Date	*Transactions*	*Amount Rs.*
2014		
Apr. 01	Commenced business with cash	5,00,000
Apr. 01	Cash deposited Into bank	4,00,000
Apr. 02	Goods purchased and payment made by Cheque No. 765421	1,50,000
Apr. 02	Rent for the month of April, 2014 paid by Cheque No. 765423	9,000
Apr. 03	Goods purchased for cash from R.S. & Sons	50,000

Fig. 14.18 : *Accounting transactions of an organisation*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

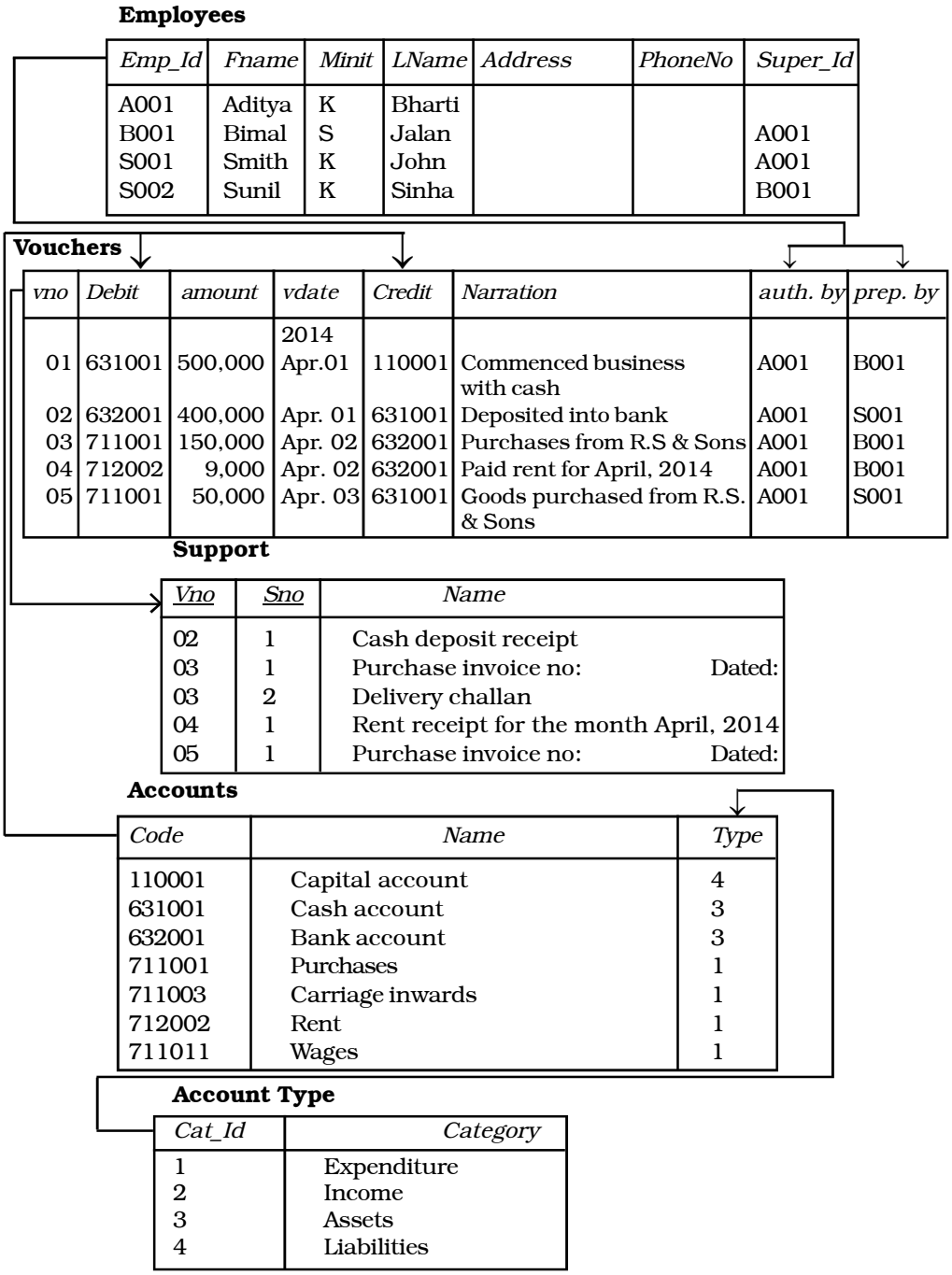
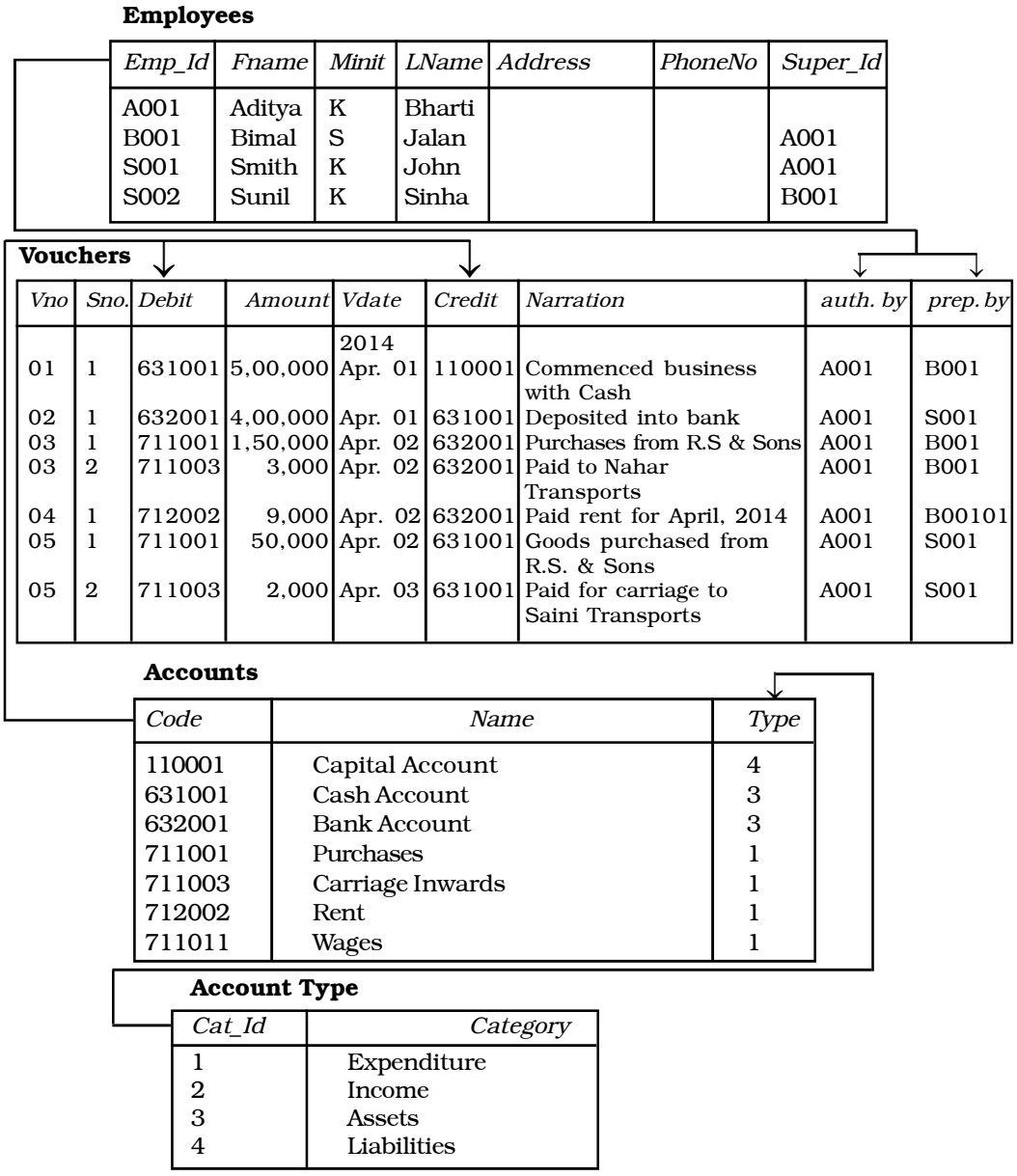


Fig. 14.19 : *An example of an accounting database that stores simple accounting transactions*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)



Employees

Emp_Id	*Fname*	*Minit*	*LName*	*Address*	*PhoneNo*	*Super_Id*
A001	Aditya	K	Bharti			
B001	Bimal	S	Jalan			A001
S001	Smith	K	John			A001
S002	Sunil	K	Sinha			B001

Vouchers

Vno	*Sno.*	*Debit*	*Amount*	*Vdate*	*Credit*	*Narration*	*auth. by*	*prep. by*
				2014				
01	1	631001	5,00,000	Apr. 01	110001	Commenced business with Cash	A001	B001
02	1	632001	4,00,000	Apr. 01	631001	Deposited into bank	A001	S001
03	1	711001	1,50,000	Apr. 02	632001	Purchases from R.S & Sons	A001	B001
03	2	711003	3,000	Apr. 02	632001	Paid to Nahar Transports	A001	B001
04	1	712002	9,000	Apr. 02	632001	Paid rent for April, 2014	A001	B00101
05	1	711001	50,000	Apr. 02	631001	Goods purchased from R.S. & Sons	A001	S001
05	2	711003	2,000	Apr. 03	631001	Paid for carriage to Saini Transports	A001	S001

Accounts

Code	*Name*	*Type*
110001	Capital Account	4
631001	Cash Account	3
632001	Bank Account	3
711001	Purchases	1
711003	Carriage Inwards	1
712002	Rent	1
711011	Wages	1

Account Type

Cat_Id	*Category*
1	Expenditure
2	Income
3	Assets
4	Liabilities

Fig. 14.20 : *An example of an accounting database to store accounting transactions according to debit and credit vouchers support table omitted*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Modified Version of Accounting Database : An attempt to accommodate Debit and Credit vouchers, as shown in Figure: 14.2 and 14.3, results in adding a new column Sno to Vouchers table of database, which is shown in modified database in figure 14.19. This results in data redundancy as shown in figure 14.20.

ER Model, as already discussed above, is a conceptual model, which need be transformed into a representational data model so that a database design is formed for being implemented and operated upon by using DBMS. From among several representational models, Relational Data Model (RDM) is the most popular and widely used in actual practice. Let us understand some important concepts of RDM.

14.6 Relational Data Model

The relational data model represents the database as collection of relations, which resembles a table of values (or data table). Each row of the table, therefore, represents a collection of related data values and hence typically corresponds to real world entity or relationship. The table name and column names are used to help in interpreting the meaning of values in each row. Each row of a table is called a data record. All values in a column, which belong to a particular domain, are of same data type

Consider the following table of data items, named as *Accounts*. The table has rows and columns. The column arrow points to a column called Name. The Row arrow points to a data record consisting of (110001, Capital Account and 4) each of which corresponds to Code, Name and Type, which are three different columns of the table.

Name of Table : Accounts

Code	*Name*	*Type*
110001	Capital Account	4
221019	Jain & Co.	4
221020	Jayram Bros.	4
411001	Furniture Account	3

Fig. 14.21 : *Example data table of accounts and their attribute values*

Formally, a *row* is called a **tuple**, a *column* header is called an **attribute** and the *table* as such is called a **relation**. The data type describing the types of *values* (*such as text value, numeric values, date values, currency value, etc.*) *that* can *appear in each column* is called a **domain.** A domain is a set of indivisible values. Associated with every domain is a data type such as Number,

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Text, Currency, Date/Time, etc. Each domain must also be named so as to help in interpreting its values. Besides this, a domain must be given a *format* and any *additional information* to enable correct interpretation of values. For example, a numeric domain such as distance should have units of measurement: Miles or Kilometers

(a) *Relations* : A relation schema is made up of a relation name and a list of its attributes. Each attribute is the name of role played by some domain in the relation schema**.** A relation is given an identity by its name and description by its schema. The degree of a relation is indicated by the number of attributes it contains. For example, the degree of a relation schema accounts is three as shown below :

ACCOUNTS (*Code*, Name, Type) ← Relation with attributes

ACCOUNTS *is name of the relation which has three attributes*;

Code = Identity of Account;
Name = Names of Account;
Type = Category of Account

A Relation represents an entity type. A relation (or relation state) is a set of tuples wherein each tuple is an ordered list of values corresponding to attributes of relation. Each of these values must belong to the domains of their respective attributes. Each tuple in this relation *represents* a particular entity. A relation schema may be interpreted as a declaration in the nature of an assertion. For example, the schema of accounts relation, as shown above, asserts that every account has a Code, Name and a Type. As a result, each tuple in accounts relation can be interpreted as a fact or an instance of assertion. Some relations represent facts about entities while others might represent facts about relationships.

(b) *Values in Tuples* : Each value in a tuple is an indivisible value to imply that it is not divisible into components within the framework of the basic relational model. This implies that composite and multi-valued attributes are not allowed. Composite attributes are represented by their simple components. The multi-valued attributes are represented by separate relations. A special value called *Null* is used to represent unknown or not applicable values of attributes in a tuple. It is also possible to devise different types of code values for different types of null value situation.

14.7 Relational Databases and Schemas

A relational database schema is a set of relation schemas and a set of integrity constraints. A relational database state is a set of relation states such that every relational database state satisfies the integrity constraints specified on relational database schema.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

In this context the following points merit a special consideration :

(a) A particular attribute, which stands for the same real word concept, might appear in more than one relation with same or different name. For example, in vouchers relation, the account Number is represented as *debit* and *credit* whereas in accounts relation, it is represented as *Code* (figure 14.19). EmpId appearing in Employees relation is represented in Vouchers as Auth.By and Prep.By.

(b) The particular real world concept appearing more than once in a relation must be represented by different names. For example, in employees relation, employee is represented as subordinate, by using *EmpId* and as superior by using *SuperId*.

(c) The Integrity constraints, specified on database schema, must hold in every database state of that schema.

14.8 Constraints and Database Schemas

There are *four* different *constraints*, which can be specified on relational databases. These are: *domain* constraint; *key* constraint; *entity* integrity constraint; *referential* integrity constraints.

(a) *Domain* : The value of each attribute of a relation must be an indivisible value and drawn out of possible values associated with its domain. The value of an attribute, therefore, must conform to the data type associated with the domain.

(b) *Key Constraints and NULL Values* : Each data record, which corresponds to a tuple of a relation, in a table must be distinct. That means no two tuples (or rows) in a relation (or table) can have the same combination of values for all their data items. This is because that a relation, as set of tuples, has to have all its tuples distinct by definition. Every relation has at least one key by default, which is the combination of all its attributes. This is called super-key by default. Any such super-key, therefore, specifies *uniqueness constraint.* Such a combination, representing super-key, may have redundant attributes, implying thereby that a more useful concept is that of a *key* which has not redundancy. This can be shown diagrammatically as shown in figure 14.22. Therefore, minimal super-key (also called *Key*) is defined as that part of super-key from which any attribute cannot be removed without sacrificing the uniqueness constraint. The value of key attribute can be used to identify each tuple in a relation. A key is determined from the meaning of the attributes. The uniqueness feature of key must continue to hold when new tuple in a relation is added. Sometimes a relation may have more than one key in which case each of such keys is called a *candidate key*. One such key is termed as primary key of relation. *The choice of which candidate key to be primary is generally subjective*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

and may depend on circumstances of mini-world. For Example: Both PAN(Permanent Account Number) and EMPID are candidate keys in EMPLOYEES relation because of being unique. But EMPID should be selected in an organisation being native to the organisational environment.

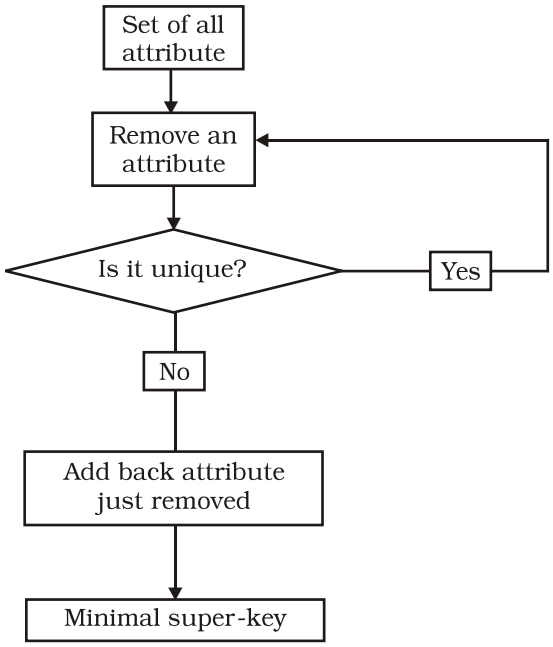


Fig. 14.22 : *Flow chart to reach a minimal super-key*

(c) *Entity integrity constraint* : States that no primary key value can be null because it is used to identify individual tuple in a relation. Null value implies that we cannot identify such tuples or identify these as alike. A failure to distinguish them means they are duplicates.

(d) *Referential integrity constraint* : While key and entity constraints are specified on individual relation, the referential integrity constraint is specified between two or more relations. This constraint is specified to maintain consistency among the tuples of such relations. Accordingly, a tuple in one relation that refers to another relation must refer to an *existing tuple* in that other relation. In referencing *Accounts Type, Accounts* relation uses its attribute *Type,* which acts as foreign key to reference the tuples of relation *Accounts Type* through its primary key *CatId*. The value of Type cannot be null because of total participation of Accounts in classify relationship. Similarly, consider another example in which the relation *Vouchers*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(*Vno, Sno*, Vdate, Debit, Amount, Credit, Amount, Prep_by, Auth_by, Narration) references two other relations as shown in figure 14.19.

First it references, *Accounts* (*Code, Name, Type*). In referencing Accounts, the *Vouchers* relation uses its attributes Debit and Credit, which act as Foreign Keys to reference the tuples of relation *Accounts* through its primary key, *Code*. The values of debit and credit cannot be null because of total participation of vouchers in debit and credit relationship.

Second, it references *Employees* (EmpId, Fname, Minit, Lname, Address, PhoneNo, SuperId). While referencing **Employees**, the **Vouchers** relation makes use of its other attributes Prep.By and Auth.By. These attributes act as foreign keys to reference the tuples of relation Employees through its key attribute EmpId. The values of PrepBy and AuthBy cannot be null because of total participation of vouchers in PrepBy and Authby relationships.

The referential integrity constraint stands violated in above example, if there is a debit or credit code in voucher relation, the tuple for which does not exist in Accounts relation. Similarly, referential integrity fails, if there exists a value corresponding to *Auth.By* or *Prep.By* attribute of vouchers, the tuple for which does not exist in employees relation.

14.9 Operations and Constraint Violations

There are two categories of operations on relational model : *updates* and *retrieval* The three basic types of updates are as given below :

(a) *Insert* : This operation is performed to add a new tuple in a relation. For example, an attempt to add another record of an account with data values corresponding to Code, Name and its Type to Accounts relation shall be made by performing Insert operation. The insert operation is capable of violating any of the four constraints discussed above.

(b) *Delete* : This operation is carried out to remove a tuple from a relation. A particular data record from a table can be removed by performing such a operation. The delete operation can violate only referential integrity, if tuple being removed is referenced by foreign key from other tuples in the database.

(c) *Modify* : The operation aims at causing a change in the values of some attributes in existing tuples. This is useful in modifying existing values of an accounting record in a data table. Usually, this operation does not cause problems provided the modification is directed on neither primary key nor foreign key.

Whenever applied, these operations must enforce integrity constraints specified on relational database schema.

Retrieval operation on Relational Data Model does not cause violation any integrity constraints.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

14.10 Designing Relational Database Schema

The rules or guidelines required to design the relational database schema attempt to provide a step-by-step procedure that transforms ER design into Relational Data model design to constitute the desired database. In the context of ER model as shown in design figure14.12, the following specific steps are required to cause its transformation into relational data model :

(i) *Create a relation for every strong entity* : For each strong entity type (which has primary key) in ER schema, a separate relation that includes all the simple attributes of that entity is created. Either choose one of the key attributes of such an entity as the *primary key for this relation*, or choose a set of simple attributes that uniquely identify this entity as the primary key of the relation so created. For example, employee entity is strong because it finds its primary key in *EmpId* which is one of its unique attribute. Therefore, a separate relation for Employee has been created as shown below :

Employee (EmpId, Fname, Minit,Lname,Address, PhoneNo, SuperId)

Similarly, separate relations need be created for the following strong entities whose Primary Key attribute have been underlined.

Accounts (Code, Name, Type)
Vouchers (VNo,vDate, amount, narration)
Accounts Type (CatId, Category)

(ii) *Create a separate relation for each weak entity type* : Every weak entity has an owner entity and an identifying relationship through which such weak entity type is identified. For every weak entity type, a separate relation is created by including its attributes. The primary key of this new relation is the combination of its unique attribute(s) for a particular tuple of the owner relation along with primary key attribute of such owner relation. Furthermore, the primary key of owner entity is included as foreign key in such a relation key of owner entity and the partial key of weak entity. For example, Support Entity, with Vouchers as its owner Entity, does not have a primary key of its own. It has partial key which is the Sno assigned to each document. Therefore, the Primary key of Vouchers, Vno along with Sno is designed as composite key for support entity and the relation so formed is shown below as :

Support (vNo,Sno, dName,sDate)

(iii) *Identify entity types participating in binary 1:N relationship type* : Identify the first relation on n-side of relationship and second on 1-side of such relationship. The primary key of second relation should be included in first relation as its foreign key. For Example, An employee can authorize a number of vouchers. It implies that *Vouchers* entity participates in *Auth.By*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

relationship on n-side while *Employees* entity participates in same relationship on 1-side. Therefore, the vouchers relation as already formed above in step 1, must also include as foreign key the primary key of Employees**,** which is *EmpId***.** Similarly, we can deal with *Prep.By* relationship in which Employees and Vouchers again participate in binary 1:N relationship. The end result of mapping both these relationships is to include twice the *EmpId*, but in different roles. Since a relation cannot have same name (here *EmpId* twice to mean *AuthBy* and *PrepBy*), we use their role names as attributes in *Vouchers* relation as foreign keys to reference *Employees* relation.

Accordingly, the modified *Vouchers* relation appears as given below:

Vouchers (VNo, vDate, Amount, Narration, Auth.By, Prep.By)

Similarly, there exist two relationships between the relations Vouchers and Accounts. The relation Vouchers as modified above shall further include as foreign key the primary key of Accounts relation, which is code. This code is to be included twice. One to represent debit and another to represent credit relationship. Since a relation cannot have same name (here *Code* is being included twice to mean *Debit* and *Credit*), we use their role names as attributes in *Vouchers* relation as foreign keys to reference *Accounts* relation. The modified vouchers relation shall appear as follows:

Vouchers (Vno,Vdate, Debit, Credit, Amount, Narration, AuthBy, Prep.By)

(iv) *Identify entity types participating in binary M:N relationship type* : For each binary M:N relationship type, create a new relation to represent such relationship. This new relation should include as foreign keys, the primary keys of the relations that represent the participating entity types. For example, consider the following entities and relationships in the context of credit voucher shown in figure 14.23, which has one debit with multiple credit accounts :

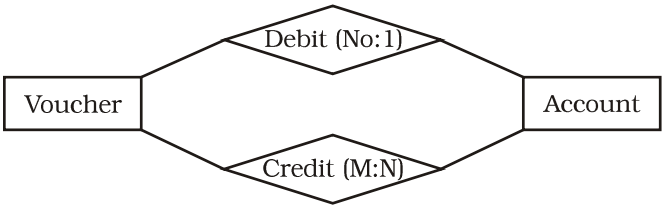


Fig. 14.23 : *ER Diagram showing relationships between vouchers and accounts in the context of credit vouchers, with one debit and several credit entries*

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

In this case, relationship Credit has cardinality ratio of M:N between Vouchers and Accounts(many vouchers are related to many accounts), While relationship Debit has cardinality ratio of N:1 (many vouchers refer to one account). Further Credit relationship has Sno, amount and narration has its attributes. Accordingly, we create a new relation as follows :

Credit (vNo, Sno, Code, Amount, Narration)

In above relation credit Code is included as foreign key to represent primary key of accounts relation, Vno is included as foreign key to represent primary key of relation vouchers. (Vno,Code) constitute the primary key of this new relation credit. By analogy, we can arrive at the following relation for Debit voucher:

Debit (vNo, Sno, Code, Amount, Narration)

Finally, the following relations have been formed to constitute the relational data model for our example reality.

Employee (EmpId, Fname, Minit,Lname,Address, PhoneNo, SuperId**)**
Accounts (Code, Name, Type**)**
Vouchers (VNo,Vdate, debit, credit, amount, narration, AuthBy, PrepBy**)**
AccountsType (CatType, Category)
Support (VNo,Sno,Dname,Sdate**)**

If we adopt the additional semantics the vouchers relation shall appear in two different schemas :

Situation A : The schema given below is compatible with Debit voucher as shown if figure 14.3.

Vouchers (vNo,vDate, Credit, Auth.By, Prep.By**)**
Debit (vNo, Sno, Code, Amount, Narration)

Situation B : The schema given below is compatible with Credit voucher as shown if figure 14.2.

Vouchers (vNo,vDate, debit, AuthBy, PrepBy**)**
Credit (vNo, Sno, Code, Amount, Narration)

A generalised Schema for the two schemas shall be

Vouchers (vNo,vDate,Vtype, AccCode, vType, AuthBy, PrepBy**)**
Details (vno, Sno, Code, Amount, Narration)

Where in another attribute vType has been introduced to indicate whether this generalised schema applies to Situation A(vType=0) or Situation B(vType=1). Debit and Credit attribute of vouchers relation have been renamed as AccCode to mean Debit and Credit, depending on the value of Vtype. Debit and Credit relations have been generalised into Details because both shared a set of common attributes.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

14.11 Illustrating the Database Structure for Example Realities

DBMS software is used to implement the data model by creating several tables, setting their interrelationships and imposing constraints as may be set out in database design. After, the design is implemented, it must also allow for retrieval of data and information. This is achieved by querying the database, for which purpose, SQL statements are put to use. These retrieval requests result in emergence of new virtual tables that may be formed out of one or more of existing tables. A clear understanding of these SQL statements is a first step towards the theoretical foundations for computerised reporting. This is because a report is an organised set of information, which is extracted on the basis of these retrieval requests. For a practical understanding of these operations, consider the following Models, herein referred to as Model-I and Model-II. Each of these models, which consist of a set of relations (or tables) and the integrity constraints, constitutes the database design for accounting.

Model-I : This is based on initial conceptual design of example reality shown in Figure: 14.11

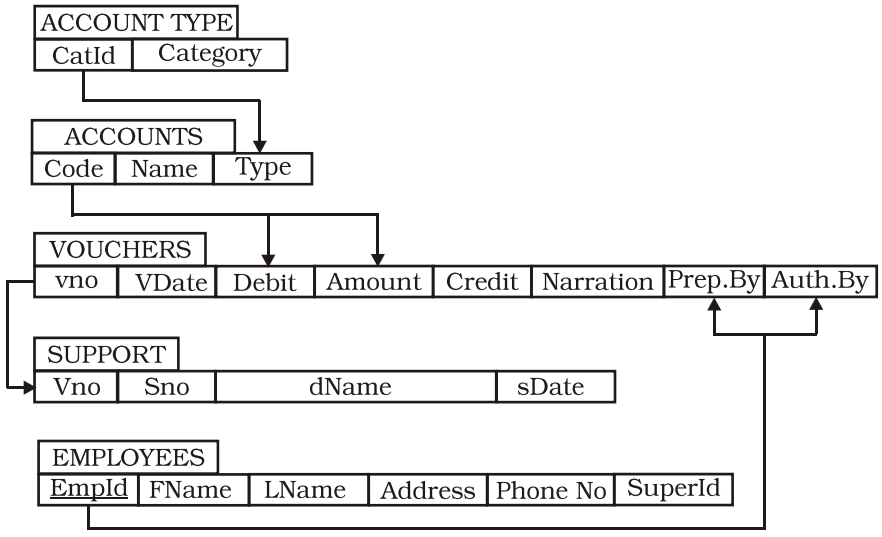


Fig. 14.24 : *Schema diagram for the accounting system relational database schema*

Model-II : The set relations given below are based on modified example reality that uses Credit and Debit vouchers shown in figures 14.2 and 14.3.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

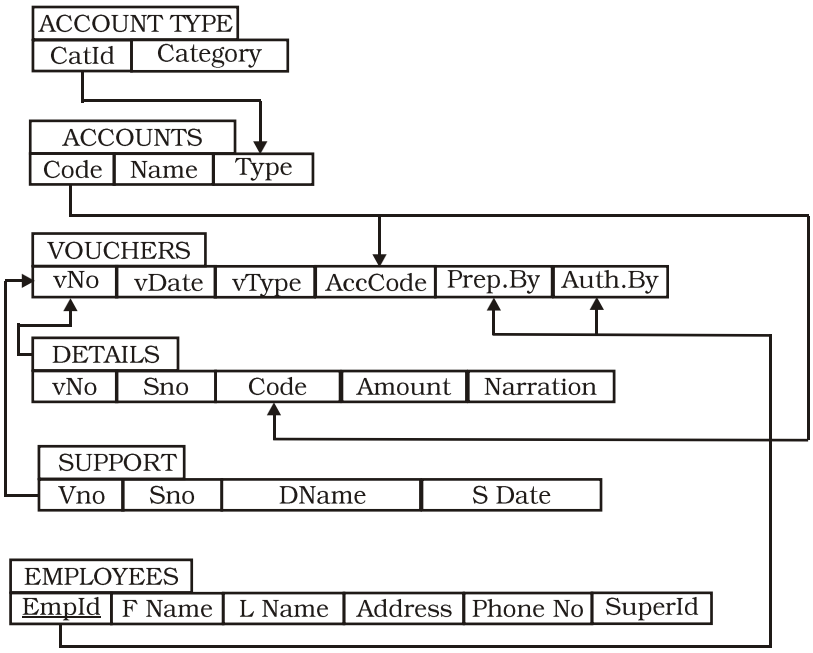


Fig. 14.25 : *Schema diagram for the accounting system relational database Schema*

Illustration No 1

Mr. Philips commenced business with cash and for that purpose opened a bank account on April 1, 2014. His transactions for the month are as given below :

Date	*Transactions*	*Amount Rs .*
2014		
Apr. 01	Commenced business with cash	5,00,000
Apr. 01	Cash deposited Into bank	4,00,000
Apr. 02	Goods purchased and payment made by Cheque No. 765421	1,50,000
	Cheque No. 765422 issued to M/s Nahar Transports for carriage	3,000
Apr. 02	Rent for the month April, 2014 paid by Cheque No. 765423	9,000
Apr. 03	Goods purchased for cash from M/s R.S. & Sons	50,000
	Paid for carriage to M/s Saini Transports	2,000
Apr. 04	Goods sold to Kemp & Co.	1,75,000
Apr. 05	Goods purchased from M/s Jayram Bros.	2,50,000
Apr. 06	Sold goods for cash to M/s Kumbley & Co.	45,000
Apr. 08	Paid for adverisement by Cheque No. 765424 to M/s ABN Cables	2,500

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Apr. 09	Received a bill of exchange from Kemp & Co.payable after 3 months	1,75,000
Apr. 10	Bill of exchange received from Kemp & Co. discounted for	1,71,500
Apr. 12	Goods returned to Jayram Bros. being defective	15,000
Apr. 15	Advance cash payment to salesman for marketing tour	10,000
Apr. 17	Paid for insurance of godown Cheque No. 765425	5,500
Apr. 18	Paid for fuel, power and electricity	1,000
Apr. 18	Salary paid in advance to bimal	10,000
Apr. 19	Accepted a bill of exchange payable after four months in favour of Jay Ram Bros.	2,35,000
Apr. 21	Returns from M/s Kumbley & Co., settled by Cheque No. 765427	5,000
Apr. 23	Cash withdrawn by proprietor for household expenses	20,000
Apr. 25	Advance to salesman adjusted for cash after recording expenses :	
	Entertainment	4,500
	Travelling	2,200
	Boarding and Lodging	3,500
Apr. 27	Goods taken from stock for personal use	5,000
Apr. 28	Furniture purchase from M/s S.N. Furnitures by Cheque No. 765428	45,000
Apr. 29	A part of existing stock set a side for usage as office furniture	35,000
Apr. 30	Salary for the month paid by Cheques	
	Cheque No. 765429 to Aditya	9,000
	Cheque No. 765430 to Bimal (one-fourth of advance adjusted)	5,500
	Cheque No. 765431 to Smith	6,000
	Cheque No. 765432 to Sunil	5,000
Apr. 30	Payment of telephone bill by Cheque No. 765433	1,500
Apr. 30	Paid for wages by cash	7,000

The database state pertaining to Accounts and Employees table is as given below :

Accounts

Code	*Name*	*Type*
110001	Capital Account	4
221019	Jain & Co.	4
221020	Jayram Bros.	4
222001	Bill Payables	4
411001	Furniture Account	3
411002	Office Fittings	3
412002	Plant and Machinery Account	3

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

621001	Kemp & Co. 3	
621002	Kumble & Sons	1
631001	Cash account	3
632001	Bank account	3
641001	Salary in advance account	3
641002	Advance to salesman	3
642001	Bills receivable	3
651001	Drawings	4
711001	Purchases	1
711002	Purchases returns	1
711003	Carriage inwards	1
711004	Fuel, power and electricity	1
711011	Wages	1
712001	General expenses	1
712002	Rent account	1
712003	Salaries account	1
712004	Discount account	1
712005	Adverisement	1
712006	Entertainment	1
712007	Travelling	1
712008	Boarding and Lodging	1
712009	Communication expenses	1
712010	Insurance	1
811001	Sales account	2
811002	Sales returns	2

Account Type

CatId	*Category*
1	Expenditure
2	Income
3	Assets
4	Liabilities

Employees

EmpId	*Fname*	*Minit*	*LName*	*Address*	*PhoneNo*	*SuperId*
A001	Aditya	K	Bharti			
B001	Bimal	S	Jalan			A001
S001	Smith	K	John			A001
S002	Sunil	K	Sinha			B001

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

The solution based on *Model-I* which lends support to Transaction Voucher with one Debit and one Credit as shown in figure 14.19, shall appear as follows :

Vouchers

vNo	*Debit*	*amount Rs.*	*vDate*	*Credit*	*narration*	*AuthBy*	*PrepBy*
			2014				
01	631001	5,00,000	Apr. 01	110001	Commenced business with cash	A001	B001
02	632001	4,00,000	Apr. 01	631001	Deposited into bank	A001	S001
03	711001	1,50,000	Apr. 02	632001	Purchases from R.S & Sons	A001	B001
04	711003	3,000	Apr. 02	632001	Paid to M/s Nahar Transports	A001	B001
05	712002	9,000	Apr. 02	632001	Paid rent for April, 2014	A001	B001
06	711001	50,000	Apr. 03	631001	Goods purchased from R.S. & Sons	A001	S001
07	711003	2,000	Apr. 03	631001	Paid for carriage to M/s Saini Transports	A001	S001
08	621001	1,75,000	Apr, 04	811001	Goods sold	A001	S002
09	711001	2,50,000	Apr. 05	221020	Invoice no. dated :	B001	S002
10	631001	45,000	Apr. 06	811001	Goods sold to M/s Kumbley & Co.	S001	S002
11	712005	2,500	Apr. 08	632001	Paid to M/s ABN Cables	A001	S002
12	642001	1,75,000	Apr. 09	621001	Maturity Date : July 12, 2014	A001	S002
13	711002	15,000	Apr. 10	221020	Goods returned Note No. dated :	A001	S002
14	712004	3,500	Apr. 12	642001	Discount on Bill of exchange from Kemp & Co.	A001	S002
15	641002	10,000	Apr. 12	631001	Advance payment to sales for marketing tour	B001	S001
16	712010	5,500	Apr. 17	632001	Insurance of godown	S001	B001
17	711004	1,000	Apr. 18	631001	Payment for fuel, power and electricity	S001	B001
18	641001	10,000	Apr. 18	631001	Salary paid in advance to Bimal	B001	B001
19	221020	2,35,000	Apr. 19	222001	Settlement by accepting a bill of exchange	B001	S001
20	811002	5,000	Apr. 21	632001	Goods returned by M/s Kumbley & Co.	A001	S001
21	651001	20,000	Apr. 23	631001	Withdrawal by proprietor for household expenses	A001	S001
22	712006	4,500	Apr. 25	641002	Expenses during tour : Support vouchers 1-4	A001	S001

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

23	712007	2,200	Apr. 25	641002	Expenses during tour : Support vouchers 5-7	A001	S001
24	712008	3,500	Apr. 25	641002	Expenses during tour : Support vouchers 8-11	A001	S001
25	641002	200	Apr. 25	631001	Final settlement of Refer to J.V No : 04/21	A001	S001
26	651001	5,000	Apr. 27	711001	Goods taken for private use	A001	S002
27	411001	45,000	Apr. 28	632001	Furniture purchased from S.N. Furniture	A001	S002
28	411001	35,000	Apr. 29	711001	Goods purchased for trading put to office use	A001	S002
29	712001	9,000	Apr. 30	632001	Salary to Aditya-Apr,2001	A001	S001
30	712001	5,500	Apr. 30	632001	Salary to Bimal-April, 2001 after adjustment	A001	S001
31	712001	6,000	Apr. 30	632001	Salary to Smith-April 2001	A001	S001
32	712001	5,000	Apr. 30	632001	Salary to Sunil-April, 2001	A001	S001
33	712009	1,500	Apr. 30	632001	Telephone bill	A001	B001
34	711011	7,000	Apr. 30	631001	Payment of Wages	A001	S001

Shortcomings

The above solution, being based on transaction voucher with one debit and one credit in a transaction requires multiple vouchers for one real transaction. For example, a transaction dated April 30, 2014 "Salary for the month paid by cheque" requires four vouchers 29 to 32. One transaction should be recorded possibly through one voucher only.

Solution

The solution based on *Model-II* which lends support to Debit Voucher (with Multiple Debits and one Credit) and Credit voucher (with one Debit and multiple Credits) as shown in Figure: 14.2 and figure 14.3 shall appear as follows :

Vouchers

Vno	*Vdate*	*Acc_code*	*Vtype*	*PrepBy*	*AuthBy*
	2005				
01	Apr. 01	631001	1	B001	A001
02	Apr. 01	632001	1	S001	A001
03	Apr. 02	632001	0	B001	A001
04	Apr. 02	632001	0	B001	A001
05	Apr. 03	631001	0	S001	A001
06	Apr. 04	811001	0	S002	A001
07	Apr. 05	221020	0	S002	B001

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

08	Apr. 06	631001	1	S002	S001
09	Apr. 08	632001	0	S002	A001
10	Apr. 09	621001	0	S002	A001
11	Apr. 10	632001	1	S002	A001
12	Apr. 10	221020	0	S002	A001
13	Apr. 12	642001	0	S002	A001
14	Apr. 12	631001	0	S001	B001
15	Apr. 17	632001	0	B001	S001
16	Apr. 18	631001	0	B001	S001
17	Apr. 18	631001	0	B001	B001
18	Apr. 19	222001	0	S001	B001
19	Apr. 21	632001	0	S001	A001
20	Apr. 23	631001	0	S001	A001
21	Apr. 25	641002	0	S001	A001
22	Apr. 25	631001	0	S001	A001
23	Apr. 27	711001	0	S002	A001
24	Apr. 28	632001	0	S002	A001
25	Apr. 29	711001	0	S002	A001
26	Apr. 30	632001	0	S001	A001
27	Apr. 30	632001	0	B001	A001
28	Apr. 30	631001	0	S001	A001

Details

Vno	*Sno*	*Code*	*Amount*	*Narration*
01	1	110001	5,00,000	Commenced business with cash
02	1	631001	4,00,000	Deposited into bank
03	1	711001	1,50,000	Purchases from R.S & Sons
03	2	711003	3,000	Paid to M/s Nahar Transports
04	1	712002	9,000	Paid rent for April, 2014
05	1	711001	50,000	Goods purchased from R.S. & Sons
05	2	711003	2,000	Paid for carriage to M/s Saini Transports
06	1	621001	1,75,000	Goods sold
07	1	711001	2,50,000	Invoice No. dated:
08	1	811001	45,000	Goods sold to M/s Kumbley & Co.
09	1	712005	2,500	Paid to M/s ABN cables
10	1	642001	1,75,000	Maturity date July 12, 2014
12	1	711002	15,000	Goods returned Note No. dated.
13	1	712004	3,500	Discount on bill of exchange from Kemp & Co.
14	1	641002	10,000	Advance payment to sales for marketing tour
15	1	712010	5,500	Insurance of godown
16	1	711004	1,000	Payment for fuel, power and electricity
17	1	641001	10,000	Salary paid in advance to Bimal
18	1	221020	2,35,000	Settlement by accepting a bill of exchange

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

19	1	811002	5,000	Goods Returned by M/s Kumbley & Co.
20	1	651001	20,000	Withdrawal by proprietor for household expenses
21	1	712006	4,500	Expenses during tour: Support Vouchers 1-4
21	2	712007	2,200	Expenses during tour: Support Vouchers 5-7
21	3	712008	3,500	Expenses during tour: Support Vouchers 8-11
22	1	641002	200	Final settlement of Refer to J.V no. 04/21
23	1	651001	5,000	Goods taken for private use
24	1	411001	45,000	Furniture purchased from S.N. Furniture
25	1	411001	35,000	Goods purchased for trading put to office use
26	1	712001	9,000	Salary to Aditya Apr. 2014
26	2	712001	5,500	Salary to Bimal Apr. 2014 after adjustment
26	3	712001	6,000	Salary to Smith Apr. 2014
26	4	712001	5,000	Salary to Sunil Apr. 2014
27	1	712009	1,500	Telephone bill
28	1	711011	7,000	Payment of Wages

Test Your Understanding

A. Indicate against each of the following statements, True or False :

(a) Every relation has at least one super key by default, which is the combination of all its attributes.
(b) Data transformation is called Information.
(c) Referential integrity constraint arises because of relationships between various entities.
(d) The complete absence of WHERE clause in SELECT statement implies that no tuples of a relation shall be selected.
(e) ER model is an example of representational data model.

B. Fill in the blanks, an appropriate word(s)

(a) A ..... does not have key attributes of its own.
(b) The ..... for binary relationship specifies the number of relationship instances that an entity can participate in.
(c) Each simple attribute of an entity type is associated with a value set called ..... of values.
(d) When structure of AIS is based on both human and computer resources, it is called ..... AIS.
(e) An ..... is a collection of all entities of a particular entity type.
(f) A weak entity type always has a ..... constraint with respect to its identifying relationship.
(g) When a relation has more than one attribute with unique values, each such attribute is called .....

After appreciating the way accounting data is presented in above database models, let us understand as to how the *queries* on such databases are expressed as relational operations.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

14.12 Interacting with Databases

One of the major reasons for the success of commercial databases is the SQL language support they enjoy. This is because SQL became standard for relational databases. As a result, users have become less concerned about **migrating** their database applications from one database to another database. Another advantage in using standard SQL is that users may write statements in a database application program that can access data stored in two or more relational DBMS without having to change the database sub-language (SQL) provided both the DBMS enjoy the support of a particular SQL standard.

The name SQL stands for Structured Query Language, which was originally called SEQUEL (**S**tructured **E**nglish **QUE**ry **L**anguage), designed and implemented at IBM Research as an interface for experimental relational database system called SYSTEM-R.

Being a comprehensive database language, it has statements for data definition, query and update. Besides this, it has the capability to define user-oriented views of database, specify security and authorisation, define integrity constraints and various other operations. Many computer-programming languages can act as good host languages to incorporate the statements of SQL. In this sense, it can be used as a sub-language in a database-programming context.

Basic Queries in SQL : Data Query Language (DQL), which is a sub-set of SQL is widely used to answer most of the basic queries. The basic set of queries consists of those, for which the SELECT-FROM-WHERE Structure is put to use as described below :

- *SELECT* : This clause is used to specify the data or information that is desired to answer the query.
- *FROM* : This clause is used to specify the source of data for answering the query. It can be a data table, an existing query or both.
- *WHERE* : This clause is meant to specify the conditions that are used to narrow down the choice of data to extract the information desired in **select** clause.

The following queries have been considered using the database design given in Model-I and Model-II. The solution to queries has been given using *MS ACCESS* implementation.

I. *Query to retrieve all columns of data records from a table, subject to a condition* : To project all the attribute values of selected tuples, an asterisk (*) need be specified. This asterisk stands for all the attributes.

(1) To retrieve all columns of voucher records whose voucher has been authorised by an employee whose EmpId is equal to “A001”.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

(*Model-I and Model-II*)

```
SELECT *
FROM      vouchers
WHERE     AuthBy="A001";
```

II. *Query to retrieve selected columns of data records from a table, subject to a condition.*

(2) To Retrieve vouchers with Vno, Vdate, AuthBy columns wherein the vouchers are dated "12/Apr/2014"

Solution

(*Model-I and Model-II*)

```
SELECT    Vno, Vdate, AuthBy
FROM      vouchers
WHERE     Vdate = #04/12/2014#;
```

(3) To retrieve vouchers with Vno, Vdate, Auth_by columns, which are dated "12/Apr/2014". The columns of records retrieved by the query are to be renamed as Voucher, Date and Employee

Solution

(*Model-I and Model-II*)

```
SELECT    Vno As Voucher, Vdate As Date, Prep_by As Employee
FROM      vouchers
WHERE     Vdate = # 04/12/2014#;
```

III. *Unspecified WHERE Clause* : Absence of WHERE clause in SELECT statement implies that the tuples from a relation are to be selected without applying any condition. This in turn means that all tuples of a relation specified in FROM clause qualify for being selected for the result of query. Consider the following query with reference to Model-I.

(4) Find out the list of accounts which have been debited

Solution

(*Model-I*)

```
SELECT  DISTINCT Debit As Code
FROM    vouchers;
```

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

(Model-II)

```
SELECT      AccCode As Code
FROM        vouchers
WHERE       vType = 0;
UNION
SELECT      Details.Code
FROM        vouchers, Details
WHERE       vType = 1 AND vouchers.vNo = Details.vNo;
```

Save above query as **DebitAccounts,** and thereafter execute another query as given below to get the final results.

```
SELECT DISINCT *
FROM        Debit Accounts ;
```

(5) Find out the list of accounts which have been credited

Solution

(*Model-I*)

```
SELECT  DISTINCT Credit As Code
FROM        vouchers ;
```

Solution

(*Model-II*)

```
SELECT      AccCode As Code
FROM        vouchers
WHERE       Vtype = 1;
UNION
SELECT      Details.Code
FROM        vouchers, Details
WHERE       vType = 0 AND vouchers.vNo = Details.vNo;
```

Save above query as **CreditAccounts,** and thereafter execute another query as given below to get the final results.

```
SELECT DISINCT *
FROM    CreditAccounts;
```

(6) Find out the list of accounts which have been debited as well as credited

Solution

(*Model-I*)

```
SELECT  DISTINCT Debit As Code
FROM        vouchers
WHERE       Debit IN (SELECT Credit As Code
FROM        vouchers);
```

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

(*Model-II*)

```
SELECT *
FROM      DebitAccounts
WHERE     Code IN (SELECT *
FROM      CreditAccounts);
```

Save above solution query as **DebitCredit**, both for Model-I and Model-II

(7) Find out the list of accounts which have been debited but not credited

Solution

(*Model-I*)

```
SELECT    DISTINCT Debit As Code
FROM      vouchers
WHERE     Debit NOT IN (SELECT Code
FROM      DebitCredit);
```

Solution

(*Model-II*)

```
SELECT *
FROM      DebitAccounts
WHERE     Code NOT IN (SELECT *
FROM      DebitCredit)
```

(8) Find out the list of accounts which have been credited but not debited

Solution

(Model-I)

```
SELECT DISTINCT Credit As Code
FROM      vouchers
WHERE     Credit NOT IN (SELECT Code
FROM      DebitCredit);
```

Solution

(Model-II)

```
SELECT *
FROM      CreditAccounts
WHERE     Code NOT IN ( SELECT *
FROM      DebitCredit)
```

IV. *Ambiguous Attribute Names and Renaming* (*Aliasing*) : SQL allows the use of homonyms (that is same name for two or more attributes) as long as such attributes are in different relations. If the use of a common attribute with a particular name across the relations prevails, it becomes necessary

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

to qualify the attribute name with relation name in which it exits. This is achieved by prefixing the relation name to the attribute name and separating the two by a period symbol dot. In Model-II, the attribute Vno, referring to *voucher* number in vouchers relation, also exists in *details* relation. Whenever vouchers and details relations are used in a query, the use of Vno attribute must precede the name of relation or its alias name. For example,

(9) Retrieve a list of accounts and the amounts debited because of cash payments. The Cash Account code begins with "631".

Solution

(*Model-I*)

```
SELECT    Narration, Debit As Code, Amount
FROM      Vouchers
WHERE     Credit LIKE "631*";
```

Solution

(*Model-II*)

```
SELECT    Narration,Acc_code AS Code, Amount
FROM      Vouchers AS V, Details AS D
WHERE     tType=1 AND V.vNo=D.vNo
AND       acc_code like "631*"
UNION
SELECT    Narration,Code, Amount
FROM      Vouchers AS V, Details AS D
WHERE     tType = 0 AND V.vNo = D.vNo
AND code LIKE "631*";
```

(10) To retrieve a detailed list of all accounts, giving their code, Name and category.

Solution

(*Model-I and Model-II*)

```
SELECT Code, Name, Category
FROM  Accounts, AccountType
WHERE CatId = Type
```

(11) To retrieve a detailed list of all account, giving their code, Name and category, which have been debited

Solution

(*Model-I*)

```
SELECT    DISTINCT Debit AS Code, Name, Category
FROM      Vouchers AS V,Accounts AS A, AccountType
WHERE     V.Debit = A.Code AND CatId = type
```

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

(*Model-II* by using query solution saved as DebitAccounts in Q.No: 4)

```
SELECT Code, Name, Category
FROM DebitAccounts AS D, Accounts AS A, Category
WHERE D.Code = A.Code AND Type = CatId
```

(12) To retrieve Code, Name and Category of Expense accounts which have been debited

Solution

(*Model-I*)

```
SELECT    Debit AS Code, Name, Category
FROM      Vouchers, Accounts, AccountType
WHERE     Debit = Code AND Type = CatId
          AND Category = "Expenses"
```

Solution

(*Model-II* by using query solution saved as Debit Accounts in Q.No: 4)

```
SELECT    D.Code, Name, Category
FROM      DebitAccounts AS D, Accounts AS A, AccountType
WHERE     D.Code = A.code AND Type = CatId
          AND Category = "Expenses"
```

(13) To retrieve Narration and Amount of transactions where Expense head "Carriage Inwards" has been debited.

Solution

(*Model-I*)

```
SELECT    Narration, Amount
FROM      Vouchers, Accounts
WHERE     Debit = Code
          AND Name LIKE "Carriage Inw*";
```

Solution

(*Model-II* by using query solution saved as DebitAccounts in Q.No: 4)

```
SELECT    Narration, Amount
FROM      Details AS T,DebitAccounts AS D, Accounts AS A
WHERE     T.Code = D.Code AND D.Code = A.Code
          AND Name LIKE "Carriage Inw*"
```

V. *Sub-string Comparisons and Arithmetic Operators and Ordering and use of functions* : SQL allows comparison on sub-strings (that are some parts of a character string). This can be achieved by use of **LIKE** Operator. This like operator instead of equal to (=) operator can be used when exact value

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

of comparison is not known. Partial strings or sub-strings are specified by using * and range specification within rectangular brackets. For Example:

(14) To make a list of accounts pertaining to the assets of the company, given that each of the assets account code begins with "4", following query need be executed:

Solution

(*Model-I and Model-II*)

```
SELECT    Code, Name
FROM      accounts
WHERE     Code like "4*"
```

(15) To make a list of employees whose names start from a to k, following query need be executed :

Solution

(*Model-I and Model-II*)

```
SELECT    Fname & " " & Minit & " " & Lname  As 'Name of Employee'
FROM      Employees
WHERE     Fname like "[a-e]*"
```

VI. Another comparison operator used in SQL is **BETWEEN ....AND ....**operator. This operator facilitates numeric range tests for selection of tuples. For Example:

(16) To retrieve vouchers with amount ranging between 5,000 and 10,000, following query need be formulated.

Solution

(*Model-I*)

```
SELECT    Vno, Amount
FROM      Vouchers
WHERE     Amount BETWEEN 5000 AND 10000 ;
```

Solution

(Model-II)

```
SELECT    Vno, Amount
FROM      Vouchers AS V, Details AS D
WHERE.    V.vno = D.vno AND Amount   BETWEEN 5,000 AND 10,000;
```

VII. Another feature of SQL permits the use of standard arithmetic operators, which can be directly applied to numeric values appearing in a query statement. Consider the following query:

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(17) To find various amounts of sales during the month of April, 2014 and the amounts of such sales if the prices of products are allowed to be raised by 16%.

Solution

(*Model-I*)

```
SELECT    Vdate, Credit, Amount, Amount*1.16 AS Expected
FROM      Vouchers, Accounts
WHERE     Credit = Code AND name LIKE "Sales Account*"
```

Solution

(*Model-II*)

```
SELECT    Vdate, D.code, Amount, Amount*1.16 AS Expected
FROM      Vouchers AS V, Details AS D, accounts AS  A
WHERE     V.vNo = D.vNo AND D.code = A.Code AND A.Name LIKE
          "Sales Account*" AND tType = 1
UNION
SELECT    Vdate, V.Acc_code, Amount, Amount*1.16 AS Expected
FROM      Vouchers AS V, Details AS D, accounts AS  A
WHERE     V.vno = D.vno AND V.acc_code = A.code AND A.name LIKE
          "Sales Account*" AND Ttype = 0;
```

VIII. SQL also allows ordering of resultant tuples according to some specified attribute, which may or may not form part of the resultant relation. Consider the following example:

(18) To retrieve list of Accounts in dictionary order of their Names :

Solution

(*Model-I and Model-II*)

```
SELECT  *
FROM Accounts
ORDER BY Name
```

IX. SQL queries allow the use of supported functions within the query itself. List of these functions varies from one implementation to another depending on the specific RDBMS. Consider the following example :

(19) To List details of vouchers released during April, 2014.

Solution

(*Model-I and Model-II*)

```
SELECT  *
FROM vouchers
WHERE Month(vDate) = 4
```

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

To execute above query, **month()** function is used which accepts within parenthesis the data a parameter and returns the numeric value of one month varying from 1 through 12. In this case the relevant value to be compared for the month of April is 4.

X. *Explicit Sets and NULL in SQL* : Query results can be retrieved even for rows in which value of an attribute is missing. This is achieved by using **NULL** in **Where** clause while specifying the condition. If more than one value is to be compared with an attribute, the value set can be given in Where clause by specifying **IN** operator.

(20) To retrieve Details of Accounts with following Codes: relating to "621001", "632021" and "642002".

Solution

(*Model-I and Model-II*)

```
SELECT *
FROM        Accounts
WHERE       Code IN("621001","632001","642002");
```

(21) To retrieve name of all employees who do not have supervisors.

Solution

(*Model-I and Model-II*)

```
SELECT *
FROM        Employees
WHERE       SuperId = NULL;
```

XI. *Aggregate Functions and Grouping* : The concept of aggregate functions as referred to in relational operations, is implemented by SQL. Five such functions commonly used for aggregate of data items are: **COUNT,SUM, MAX, MIN** and **AVG**. These functions when applied on a set of numeric values, return respectively number of rows, the sum , maximum, minimum and average of these values. The **GROUP BY** clause is used for providing the basis of creating collection of data items on which these functions are to be applied. Consider the following examples.

(22) To find the sum, minimum and maximum of cash payment during April, 2014. The cash account code begins with "631"

Solution

(*Model-I*)

```
SELECT      Debit AS Code, SUM(Amount) AS Total,
            MIN(Amount) As Minimum, MAX(Amount) As Maximum
FROM        Vouchers
WHERE       Debit like "631*"
GROUP BY    Debit
```

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Solution

(*Model-II*)

SELECT Code, **SUM(**Amount**) AS** Total,
MIN(Amount) As Minimum, **MAX(**Amount**)** As Maximum
FROM Vouchers **AS** V, Details **AS** D
WHERE V.Vno=D.Vno, Ttype=0 and Code Like "631*"
GROUP BY D.Code

Key Terms Introduced in the Chapter

- Database System
- Reality Database
- Accounting Intermedia
- Credit Voucher
- Attributes
- Designing Database for Accounting
- Entity Relationship (ER) Model
- Rational Data Model
- Transaction Voucher
- Debit Voucher
- Interacting with Database

Summary with Reference to Learning Objectives

(1) *Database Concepts*

Reality : It consists of different components of an organisation such as people, facilities and other resources.

Data : It represent data concerning people, places, objects entities, events, etc. and non-financial 14 nature.

Database : It was a shared collection of inter-related data tables, tiles or structures which are designed to most varied information needs of all organisation.

International : Processed data organisation in a form that is suitable for decision-making.

DBMS : A collection of programmes that enable users to create and maintain a database.

(2) *Database System Concepts and Architecture*

Data model : Collection of concepts used to describe the structure of a database.

Database Schemes : The description of a database is called its scheme.

Data Base State and Instances : Data in a database at a particular movement is called database state.

(3) *Entity Relationship (ER) Model*

An important concept of data model mostly used in data base oriented application. The major elements of ER model are entities, Attributes, identities and relationship that are used to express reality for which a data base is to be designed.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(4) *Relation Data Model (RDM)*

It represent the database at collection of tables comprising different volumes. It consists of rows and columns. The table name and column name are used to help in interpreting the meaning of volumes of each row. Each row of table is called a data record.

Questions for Practice

Short Answers

1. State main categories of data models.
2. How are computers useful in processing the accounting data?
3. What do you understand by accounting data? Discuss the stages through which it is finally transformed for being presented as information in financial statements.
4. What do you understand by database. How does it differ from DBMS?
5. What is meant by entity type? How it is different from entity set? Illustrate by giving suitable example from accounting reality.
6. What do you understand by relationship type? How is it different from relationship instance and relationship set?
7. What do you understand by multi-valued attribute? How is it different from complex and composite attribute? Illustrate by giving suitable example.
8. What do you understand by the concept of weak entity used in data modelling? Explain the relevance of owner entity type, partial key and identifying relationship in the context of such modelling.
9. What is a participation role? State the circumstances under which the use of role names becomes necessary in description of relationship types.
10. Define foreign key. How is this concept useful in relational data model? Illustrate with suitable example.
11. What is meant by NULL value? What are the reasons that lead to their occurrence in database relations?
12. Why are duplicate tuples not allowed in a relation?
13. What do you understand union compatibility of relations? For which operations such compatibility is required and why?
14. What is the need for database normalisation?

Long Answers

1. Discuss the basic concepts of Entity Relationship (ER) Model. Illustrate as to how an ER model is diagrammed.
2. What integrity constraints are specified on database schema? Why is each considered important?
3. Discuss the different types of update operations in relation to the integrity constraints which must be satisfied in a relational database model.
4. Discuss the steps you would take to transform an ER Model into various relations of Relational Data Model. Give suitable examples.

Project Work

(i) Consider the following reality in a business enterprise, which is engaged in trading activity.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

- It buys and sells a given number of items each of which is uniquely identifiable. Each unit of item is expressed in numbers or Kilograms.
- It procures its supplies from a given number of suppliers who can supply any number of items at a time. Each transaction is on credit for a particular period of time expressed in days.
- It sells various items to its customers on credit for a definite period of time expressed in days.
- Each purchase is made through a regular invoice, which has its distinct number for the supplier. It is duly dated, mentions the items being transacted, their quantities and prices and total amount of invoice.
- Design an ER schema for a database application for purchase and sales accounting and also show as to how it shall be transformed into various relations of a relational data model.

(ii) Following transactions of M/s Soumya Enterprises are given to you for the period ending March 31, 2014.

March 2014	Additional capital brought in cash by proprietor, Rs.5,00,000, out of which deposited into a bank account Rs.4,50,000
02	Received Cheque for Rs.56,000 from K & Co. on account
08	Issued Cheque for Rs.75,000 in favour of Jain & Sons
10	Payment of rent for the month Rs.15,000
12	Goods purchased Rs.34,000 by Cash
16	Goods sold to R & Co Rs.45,000
20	Purchased furniture for office use Rs.25,000
24	Paid fire insurance premium by Cheque Rs. 12,000
28	Paid cash to Jayram Bros. Rs.29,000 in full settlement of their account standing at Rs.29,500
30	Payment of salary to staff Rs.20,000

All these transactions have been stored in database tables as shown below under (Model-I of database design). Data in Accounts table appears as follows:

Accounts

Code	*Name*
110001	Capital Account
221019	Jain & Sons
411001	Furniture Account
411002	Fixtures & Fittings Account
621001	K & Co
631001	Cash Account
632001	Bank Account
641001	Salary in Advance Account
711001	Cartage Account
711002	Salaries Account
711003	Rent Account
711005	Insurance Premium
711006	Discount Account
811001	Sales Account

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

Show how will these transactions appear as accounting data in following vouchers table.

VOUCHERS					
VNo	VDate	Debit	Amount	Credit	Narration

Vno : Identity of a transaction stored through a voucher.
Vdate : to date of transaction
Debit : to code of account being debited
Amount : Amount of transaction
Credit : Code of account being credited
Narration : Narration of transaction.

(iii) M/s Soumya Exports set up a garments export business on March1, 2015. Their transactions for the month ending March 31, 2015 are given below :

March 01	Capital brought in cash by proprietor, Rs.5,00,000, out of which deposited into a bank account Rs.4,50,000
03	Received Cheque for Rs.86,000 from Kailash Nath & Co. as advance account
04	Issued Cheque for Rs.85,000 to Jackson Bros. as advance for supplies
11	Payment of rent for the month Rs.18,000
14	Purchased Computer system for office use Rs.53,000, payment for which made by Cheque
14	Goods purchased Rs.1,30,000 , payment made by Cheque.
16	Goods purchased from Jackson and Bros. for Rs.97,500
19	Goods sold to Rajeshwar & Sons Rs.45,000
22	Purchased Furniture for office use Rs.25,000
25	Paid fire insurance premium by Cheque Rs. 12,000
29	Paid Cash To Jackson Bros. Rs.12,000 in full settlement of their outstanding balance of Rs.12,500
30	Payment of salary to staff Rs.20,000

All these transactions have been stored in database tables as shown below under (Model-I of database design). Data in Accounts table appears as follows:

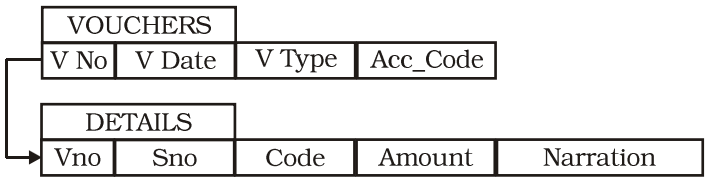
Accounts

Code	*Name*
110001	Capital Account
221019	Jackson Bros.
411001	Furniture Account

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

413001	Office Equipment
621001	Kailash Nath & Co
621002	Rajeshwar & Sons
631001	Cash Account
632001	Bank Account
641001	Salary in Advance Account
711001	Cartage Account
711002	Salaries Account
711003	Rent Account
711005	Insurance Premium
711006	Discount Account
811001	Sales Account

Show how will these transactions appear as accounting data in following accounting data tables.



Vno : Identity of a transaction stored through a voucher
Vdate : date of transaction
Acc_code : code of account being debited or credited
Code : Codes of accounts being credited or debited, depending on value of Vtype(= 0, means codes being debited, 1 means codes being credited)
Sno : Serial number of accounts being debited in debit voucher and those being credited in credit voucher
Vtype : 0 = means debit voucher, 1 = credit voucher
Amount : Amount of transaction
Narration : Narration of transaction

(iv) Write relational operation expressions and relevant SQL statements for following queries using Database Design Model-I and Model-II :

(a) Retrieve the voucher details and type of voucher authorised by a particular employee.
(b) Retrieve every bank payment voucher details, account name, amount. You are given that bank account code ="632001".
(c) Find details of cash vouchers pertaining to an expense account whose account code = "711003". You are given that cash account code="631001".
(d) Make a list of accounts and amount with respect to which a voucher has been either prepared or authorised by a particular employee.
(e) Retrieve details of vouchers without support documents.

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)

(f) List details of documents with at least one support document.
(g) Find all vouchers with total amounts raised during a particular month.
(h) Retrieve all vouchers prepared by an employee whose First name is "Smith".

(iv) Write relational operation expressions and relevant SQL statements for following queries using Database Design Model-I and Model-II.
(a) Retrieve all vouchers pertaining to a particular account with amounts ranging between Rs. 10,000 to Rs. 20,000.
(b) Retrieve details of each voucher whose support document has the same date as that of the voucher itself.
(c) Retrieve details of voucher authorised by employees who do not have supervisors.
(d) Find sum of cash payments, maximum payments, minimum payments and average.
(e) Find sum of cash payment, maximum and minimum amount with respect to a particular account Code.
(f) Retrieve every bank payment voucher details, account name, amount pertaining to a particular period ranging from Date1 to Date 2.
(g) Find details of cash vouchers pertaining to a particular expense account.
(h) Make a list of accounts and amount with respect to which a voucher has been either prepared or authorised by a particular employee.
(i) Find all vouchers with total amounts raised during a particular month.
(j) Retrieve all vouchers prepared by an employee whose last name is Dev.
(k) Retrieve details of each voucher whose support document has the same date as that of the voucher itself.

Checklist to Test Your Understanding

A. (a) T (b) T (c) T (d) F (e) F

B. (a) Weak entity
(b) Computer based
(c) Timeware
(d) Liveware
(e) Total participation
(f) Multi-valued
(g) Full functional

© copyright reserved to BSTBPC, Government of BIHAR (www.bstbpc.gov.in).
Publishing web/app/books will be a crime. Published by Abdiel Solutions (www.absol.in)