

4.1 OVERVIEW

4.1.1 Data Integrity

Database integrity refers to the correctness and consistency Of data. It is another form of database protection. While it is related to security and precision, it has some broader implications as well. Security involves protecting the data from unauthorized operations, while integrity is concerned with the quality of data itself. Integrity is usually expressed in terms of certain constraints which are the consistency rules that the database is not permitted to violate. Following two are the most important constraints in relational databases:

* + - 1. Entity Integrity: is a constraint on primary values that states that no attribute of a primary key should contain nulls.
      2. Referential integrity: is a constraint on foreign key values that states that if a foreign key exists in a relation, then either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null.

4.1.2 Normalization

Normalization is the process of converting complex data structures into simple and stable data structures. It is based on the analysis offunctional dependence.

In other words, Normalization is a technique for reviewing the entity/attribute lists to ensure that attributes are stored "where they belong". It is the basis for a relational data base system. In practice, it is simply an applied common sense. More formally stated, it is the process of analyzing the dependencies of attnbutes within entities. Attributes for each entity are checked consecutively against three sets of rules, making adjustments when necessary to put the entity in First, Second and Third normal form. First, we discuss what is functional dependence.

"A functional dependency is a particular relationship between two 

For any relation R, attribute B is functionally dependent on attribute A if, for 

every valid instance of that value of A uniquely determines the value of B". The functional dependence ofB on A is represented by.enarrow, as

A B) An attribute may be fixnctionally dependent on two or more attributes rather than a single attribute. For example, consider the relation:



COURSE (STUDID , CRSNO , CRSDATE)

The functional dependency in the relation is represented as follows:

STUDID , CRSNO CRSDATE

The attribute on the left hand side Of arrow is called determinant.

Before NORMALIZATION process, the initial entity/attribute list(s) must be checked for errors or oversights. There may' be some hidden problems as:

(i) Synonyms± A synonym is created when two different names are Used for the same information (attrl%ute). If an attribute resides in more than one entity, make Sure that all entities use the same attribute name. For example, consider the R)llowing two entities:

SUPPLIER

Stock no Supplier\_ld (error)

 Item colour Supplier\_Name



4

Supplier\_Code

We should use Supplier\_Code instead of Supplier\_ld in SUPPLIER.

Homonyms: A homonym is created when same name is used for Two different attributes. Consider the following example:



CUSTOMER 

Company\_Name Company\_Name (error)

We should use Supplier\_Narne instead of Company\_Name in SUPPLIER.

1. Redundant Information: Storing the same information in two different ways or forms. Consider the example:



Employee\_Age (error)

D O Birth

Only one attribute can serve the purpose (The programmer can manipulate the Age by using D\_O\_Birth as the basis).

1. Mutually Exclusive Data: Mutually exclusive data exists when attributes occur whose values can be expressed as "yes/no" indicators,

true for any single entity. As an exanvle, consider the  proposed attributes of"MARRIED" and "SINGLE" in an Envioyee entity.



of

can

not

all

Employee 

Married (a flag set ifthe employee. is married) error Single (a flag set if the employee is single) error

Quite often, errors of this type represent values of a larger category. Whenever possible, resolve the error by creating the larger categorical attribute. In this case, these two elements should be combined into a single attribute of 'MARITAL STATUS" which would have a value of either M (married) or S (single). 

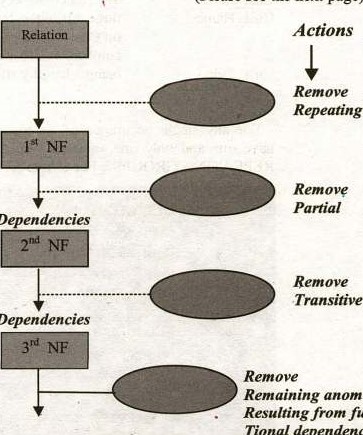
Employee

MARITAL STATUS (An indicator of the employees Marital status) Normalization Steps:

Normalization is onen accomplished in steps, each Of which corresponds to a normal form- It can be graphically expressed as follows:

(Please see the next page)

Stages



Repeating

Groups

Dependencies

Dependencies

anomalies

fu

nc—

Tional

dependencies

 Higher Normalization forms



A Normal Form is a state of a relation that can be determined by applying simple rules, regarding dependencies (or relationship between attributes), to that relation. Following is a brief discussion on different stages:

(i) First Normal Form ( I NF)

"A relation R is in First Normal Form if and only if all underlying domains contain atomic values only".

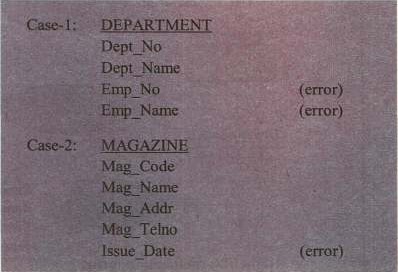
The pre-requisite is that, A relation has always a primary key associated with it. Thus, we can define it as follows also:

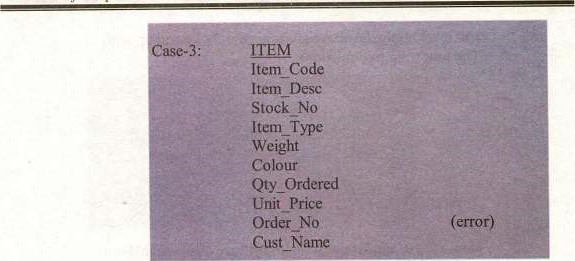
* All entities must have a key, composed of a combination of one or more attributes which uniquely identi$' one occurance of the entity. For example:

CUSTOMER

|  |  |
| --- | --- |
| cust Id | (We can create key on these |
| Cust Name | three attributes, but the key on Cust Name could be cumbersome (because Of |
| Cust Telno | being a lengthy attribute) |

* For any single occurance of an entity, each attribute must have one and only one value or "An attrlbute must have no REPEATING GROUPS". For example:





of

The error exists in the above examples because some Attribute(s) are being repeated for a single occurance of Each record. We should try to avoid this repitition. Following are the steps to achieve this:

|  |  |
| --- | --- |
| Step- l : | Whenever repeating groups occur, the repeating Attribute must be removed and placed "where it Belongs", under the entity that it describes. |
| Step-2: | Next, study the relationship of where the Repeating attn%ute came from, and where the Attribute went . Determine if the From-To Relationship is I or M:N, |

Let us apply these steps in case-I. We end up with the

Following two relations.



Now ask for the relationship in this case:

"For one department, are there one or more mployees?"

Then repeat the question in reverse. "For one employee,

Are there one or many departments?" In this case, one

Department has many employees, but one employee has

Only one department. Therefore, the relationship is I

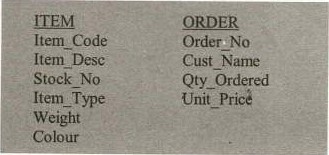
When the relationship is I this is acceptable. No

Further adjustment is necessary to make it I :M or M:l.

40

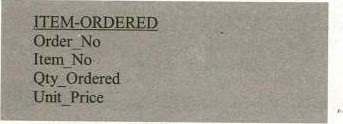


But if we apply the above steps in case-3, we end up with The following two relations:



Now ask the similar question in this case also, we find out that for each item, there are many orders, and for each Order, there are many items. Thus it is M:N relation. The

Main problem here is, "where to store the intersection Data i.e„ Qty-Ordered and Unit-price". To solve this, Create another entity "ITEM\_ORDERED" as:

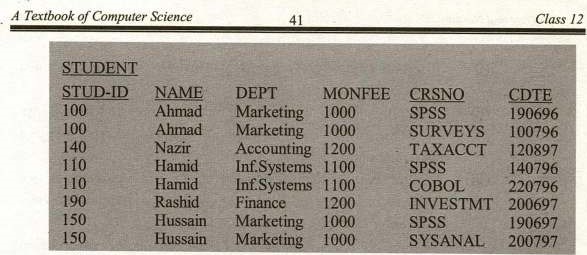


And define the key as 'the combination of Order No and Item\_No on this entity. This provides us two more tables Having I :M relationship, which is acceptable.

To have a better understanding about the whole concept, let us consider the following table of data (having repeating groups):

|  |
| --- |
| ID NAME  797 |
|

To bring it to first Normal Form, we eliminate the repeating groups from the Table and fill in the missing information (in the cells having no information). Name the table as "STUDENT'. 



(ii) Second Normal Form (2 NF): A relation is In second normal form (2 NF) if it is in 1 NF and every non-key attribute is fully functionally dependent on the primary key. More precisely.

'To be in 2 NF, every non-key attribute must depend on the key and all parts ofthe key". 

A table (relation) will be in 2NF if any of the following conditions apply.

1. The primary key consists of only one attnl)ute.
2. No non-key attributes exist in the relation.
3. Every non-key attribute is ionally dependant on the Set of primary key attributes.

Now consider the table STUDENT (which is in 1 NV).There are a lot of redundancies in this table, so it is not an acceptable stage. In shorthand notation, it is expressed as:



The functional dependencies in this relation are the as follows:

STUD-ID —+

 STUD-ID,CRSNO —+ CDTE

The primary key in ii above is the composite key:

STUD-ID + CRSNO.

Therefore, the non-key attributes NAME,DEPT and MONFEE are functionally dependent on part of the primary key (STUD-ID) but not on CRSNO.

A partial functional dependency exists when one or rnore non-key (such as NAME) are Rmctionally dependant on part (but not all) of the primary key. 

The partial functional dependency in the above table creates redundancy in that table, which results in certain anomalies when the table is updated. i.e.,



Normalization 42

Insertion Anomaly: To insert a row for the table, we must provide the values for both STUD-ID and CRSNO.

1. Deletion Anomaly: If we delete a row for one student, we lose the information that the student completed a course on a particular date.
2. Modification Anomaly. If a students monthly fee changes, we must record the change in multiple rows (for students, who have completed more than one course).

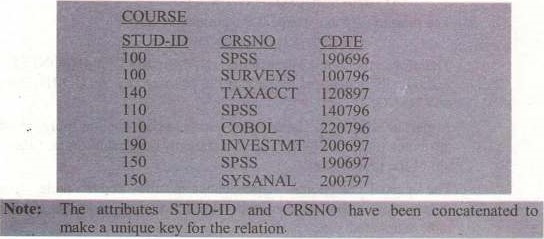
TO convert a relation to 2 NF, we decompose the relation (having redundant data) into two relations that satisty one of the conditions descrlbed above.

Now, splitting the relation (STUDENT) , we will get the following two relations, namely STUDENTI and COURSE. This step is done to get rid of the redundant data) The two tables are shown below:

TABLE l : STUDENT I (STUD-ID,NAME. DEPT. MONFEE)

|  |
| --- |
| STUD-ID NAME DEPT  100 Ahmad Marketing  140 Nazir Accounting  110 H arilid Inf.Systems  190 Rashid Finance  150 Hussain Marketing  This relation (table) satisfies condition I stated above). |

TABLE 2: COURSE(STLJD-ID.CRSNO.CDTE)





of

This relation (table 2) satisfies condition 3 above), thus it is in 2 NF.

These two relations are free of anomalies now.

(iii) Third Normal Form (3 NE)

A relation is in third normal form (3 NF) ifit is in 2 NF and no transitive dependencies exist:

What is a Transitive Dependency? It is a functional dependency in a relation between two (or more) non-key attributes.

A more precise definition for 3 NF non-key attribute must not depend on any other non-key attribute" or if a nonkey attribute's value can be obtained simply by knowing the value Of another non—key attribute, the relation is not in

Consider a relation as follows:



Where CUSTNO is the primary key.

The following functional dependencies exist in the relation,

1. CUSTNO NAME,SALESMAN
2. SALESMAN > Region (since each salesman is assigned a unique legion)

Notice that SALES is in 2 NF, because the primary kay consists of a single attribute (CUSTNO). However, there is a transitive dependency, because REGION is functionally dependent on SALESMAN which in turn is functionally dependent on CUSTNO. As a result, there are update anomalies in relation

SALES.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 | |  | |  | | |  | | | |  | South | | |
| 167 | |  | | BBBB U | | | Bashir | | | |  | West | | |
| 924 | |  | | cccc | | |  | | | |  | South | | |
| 837 | |  | | DDDD | | | Khalid | | | |  | East | | |
| 596 | |  | | EEEE | | | Bashir | | | |  |  | | |

Figure : A relation with Transitive dependency

The Anomalies:

1. Insertion Anomaly: A new salesman (Abid), assigned to the North region can not be entered until a customer has been assigned to that salesman (since a value of CUSTNO must be provided to insert a row in the table(relation).



1. Deletion Anomaly. If customer number 6837 is deleted from the relation, we lose the information that salesman Khalid is assigned to the east region.
2. Modification Anomaly: If salesman is reassigned to the east region, several rows must be changed to reflect the fict (two rows in this case).

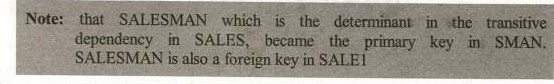
These anomalies arise as a result of the transitive dependency. This problém (the transitive dependency) can be removed by de-composing the relation SALES into two relations as shown below:

SALE I

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  |  |  |  |  |  |  |  |
| 8023 | | | | AA-XA | | | | Ahmad | | | |  |
| 9167 | | | | BBBB | | | | Bashir | | | |  |
| 7924 | | | | cccc • | | | | Ahmad | | | |  |
| 6837 | | | | DDDD | | | | Khalid | | | |  |
| 8596 | | | | EEEE | | | | Bashir | | | |  |
|  |  |  | |  |  |  | |  |  |  | |  |



Now, both the relations (SALEI & SMAN) are in 3 NF, since no transitive dependency exist. We can veriß' that the anornalies that exist is SALES are not present in SALEI and SMAN.



A o Co 12

|  |  |
| --- | --- |
|  | er Science |



 Fill in the blanks:

Entity integrity constraint states that the can not be null.

key must refer to the primary key in another table or it must be

1. Normalization is the process Of convertingstructures into simple and stable structures.



1. A(n)is a partial relationship between attributes of an entity.
2. During the first normal form groups are removed.

 To be in 2NF, a relation must be in

1. In 3NF, no dependency exists.
2. A(n) non-key attributes are functionally dependant on part ofthe primary key.



when

One

Or

1. When a new record is added in a relation, it may cause anomaly. (x) Referential integrity is a constraint On

2. Select the correct option:

(i) In 3NF, which form ofdependency is removed?  a) functional b) non-functional

c) associative d) transitive

(ii) In relational database, a table is also called a:

a) tuple b) relation

c) file d) schema

(iii) In 3NF, a non-key attribute must not depend on a(n): 

a) non-key attribute b) key attribute

c) cornposite key d) sort key

(W) Different attributes in two different tables having same narne are referred to 

a) synonym b) homonym

c) acronym d) mutually exclusive

 Every relation must have a:

a) primary key b) candidate key

c) secondary key d) composite key

Normalization



1. Mark as True or False (i) Normalization is the process of converting complex data structures into simple data structures.
   1. A relation is decomposed to convert it from I NF to 2NF.
   2. The primary key can not be a composite key.

Jn 2NF, every non-key attribute must depend on the key attribute.

* 1. A relationship involving three relations is known as a ternary relationship.
  2. A database anomaly leads the database to an inconsistent State.
  3. Partial dependencies are removed in 3NF.
  4. A relation may have multiple primary keys.
  5. In relational database. no relation can exist in isolation.
  6. The database is normalized to avoid certain database anomalies.

1. What is meant by data integrity'? What are the two types?
2. W'hat do we do to attain entity integrity?
3. Define referential integrity. How can it be achieved?
4. Explain the following terms: 
   1. Synonym
   2. Homonym
   3. Redundancy
   4. Mutual Exclusiveness of data
5. What is normalization? How it can be used to bring the database in a consistent state?
6. When is a relation in first normal form? Explain with example.
7. What are the conditions for a relation to be in second norrnal form? Give example.

Define transitive dependency. How it can be removed? Explain with the context of normalization.

1. What are the database anomalies? Briefly discuss insertion, deletion and modification anomalies.
2. What anomalies arises due to transitive dependency? Discuss briefly.
3. Define functional dependency? How partial dependencies effect a relation?
4. Convert the ER diagram you have designed in the previous exercise for the admission system of your college to relational database. Also normalize the relations up to thirdnormal form.



14.1

OVERVIEW

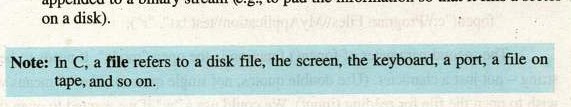
So far we have been writing programs to work with ternporary data. The user had to enter data each time the Program was executed. All programs, we have seen in previous chapters, were unable to store data and results permanently. The data is stored on permanent storage in the form of files. A file is a set Of related records. Here, we shall explore the basic file handling features of C.

14.2 THE STREAM

 Although C does not have any built-in method of performing file I/O, however the C standard library (stdio) contains a very rich set of I/O functions providing an efficient, powerful and flexible approach for file handling.

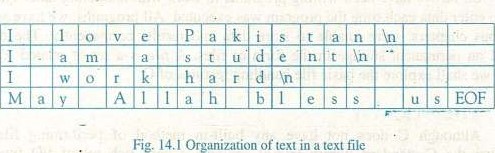
A very important concept in C is the stream. A stream is a logical interface to a file. A stream is associated to a file using an open operation. A stream is disassociated from a file using a close operation. There are two types of streams:

|  |  |  |  |
| --- | --- | --- | --- |
| A Textbook | Com | Science | 12 |

* + Text Stream: A text stream is a sequence of characters. In a text stream. certain character translations may occur (e.g. a newline may be convened to a carriage return/line-feed pair). This means that there may not be a one-to-one relationship between the characters written and those in the external device
  + Binary Stream: A binary stream is a sequence of bytes with a one-to-one correspondence to those on the external device (i.e. no translations occur). The number of bytes written or read is the same as the number on the external device. (However, an implementation-defined number of bytes may be appended to a binary Stream (e.g., to pad the information so that it fills a sector

14.3 NEWLINE AND EOF MARKER

text file is a named collection of characters saved in secondary storage e.g. on a disk. A text file has no fixed size. To mark the end of a text file, a special end-offile character is placed after the last character in the file (denoted by EOF in C). When we create a text file using a text editor such as notepad, pressing the ENTER key causes a newline character (denoted by in in C) to be placed at the end of each line, and an EOF marker is placed at the end of the file. For example, consider the organization of text in a text file in the following figure; There are four lines of text, each ends with a newline character except the last one which ends with an end of file marker i.e., EOF.



14.3 OPENING A FILE

Before reading from or writing to a file, it must be opened. All standard file handling functions of C are declared in stdio.h. Thus it is included in almost every program. TO open a file and associate it with a Stream, the fopen() function is used. Its prototype is shown here:

FILE\* fopen (const char\* filename, const char\* mode) ;

The fopen() function takes two parameters. The first is the name of the file. If the file is not in the current directory then its absolute path is be given. In this case, we 'need to escape the backslashes (i.e., use instead of in the absolute path. For example:

fopen("c:\\Program "r");

The second parameter of fopen() function is the open "mode". It needs to be a string — not just a character. (Use double quotes, not single quotes). The "r" means we wish to open the file for reading (input). We could use a if we wanted to open the file for writing (output).

The fopen() function returns the NULL pointer if it fails to open the file for some reason. The most common reason for fopen() to fail is that the file does not

A o Com uter Class 12



exist. There are, however, other reasons for failure so don't assume that is what went wrong for certain. For example:

FILE \*fp; if ( (fp z  NULL)

printf( "Error opening file\n") ; exit (1) ;

14.4.1 File Opening Modes

A file can be opened in any of the following modes:

|  |  |
| --- | --- |
| W  A  R+ | Open a tekt e or readlng. The file must already  Open a text file for writing. If the file already exists its contents are overwritten. If it does not exist, it will be created.  Open a text file for append. Data is added to the end of the existing file. If the file does not exist, it is created.  Open a text file for both reading and writing. The file muse already exist.  Open a text file for reading and writing and its contents are overwritten. If the file does not exist, it is created.  Open a text file for both reading and appending. If the file does not exist, it is created for both reading and writing. |

14.4.2 The File Pointer

A file pointer is a variable of type FILE that is defined in stdio.h. To obtain a file pointer variable, a statement like the following is used:

FILE\* fp;

We know the symbol as the arithmetic multiplication operator. But, it has entirely different meaning when used with a data type such as int, double, or FILE, It represents a pointer to the variable of type with which it is used e.g. int\* represents a pointer to an integer, float\* represents a pointer to a float variable, and FILE\* represents a pointer to a variable of type FILE. Conceptually, a pointer is a memory cell whose content is the address of another memory cell.

|  |
| --- |
| Consider the following line or code:  int\* var;  The Variable var is a pointer to an integer type Variable. It contains the address of a memory location (i.e. 0002) where an integer value can be stored. The contents of the memory location pointed to by the pointer var are referred to as \*var. A pointer is a memory location that contains the address of another memory location. The file pointer i.e. FILE\* is-a pointer to the file information which defines various properties of the file (including name, status and Current position). |
|
|

|  |  |
| --- | --- |
| 12 | var |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

0001

0002

Memory

fig. 14.1 Understanding the Pointer

Consider the following program that demonstrates the use of

# include <stdio . #include <conio .

void main (void)

int\* var; int num 25

clrscr ( ) ; var = &num; printf("Address Of variable num is ex", &num) ; printf (0 \nContentS (i.e. value) of num is , num) ; printf ( "\nAddress of memory location poin€ed to by var is \*x", var) ; printf ( "Nncontents of memory pointed to by var is td", \*var) ;



A

of

Computer

It is clear from the program that a pointer type variable stores the address of a memory location containing the value, not the value itself. The address of the variable num i.e. ff4 may be different when you would execute this program on your computer. This is because a different memory location may be assigned to the variable num each time the program is executed.

|  |
| --- |
| Output: -Here is the output ofthe program,v  Address Of variable num is fff4  Contents (i.e. value) of num is 25  Address of memory location pointed to by var is fff4  Contents of memor•,' pointed to bv Var is 25 |

14.5 CLOSING A FILE

When a program has no further use of a file, it should close it with fclose() library function. The syntax of fclsoe() is as follows:

int fclose (FILE\* fp)

Thefclose() function closes the file associated with fp, which must be a valid file pointer previously obtained using fopen(), and disassociates the stream from the file. It also destroys structure that was created to store information about file. The fclose() function returns O if successful and EOF (end of file) if an error occurs.

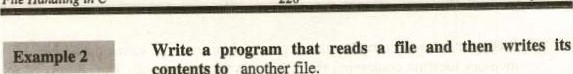
14.6 READING AND WRITING CHARACTERS TO A FILE

Once a file has been opened, depending upon its opening mode, a character can be read from or written to it by using the following two functions.

int getc (FILE\* fp) int putc (int ch, FILE\* fp)

The getc() function reads the next character from the file and returns it as an integer and if error occurs returns EOF. The getc() function also returns EOF when the end of file is encountered.

The putc() function writes the character stored in the variable ch to the file associated With fp as an unsigned char. Although ch is defined as an int yet we may use a char instead. The putc() function returns the character written if successful or EOF if an error occurs.



contents

to

#include <stdio .

void main (void)

FILE \*input; FILE \*output; int ch;

// Try to open the input file. If it fails, print a // message.

if ( (input fopen(i'afile.txt" ,  NULL)

printf ( "Can't open afile. txt for reading! \n") ;

// Now try to open the output file. If it fails,

// close the input.

else if ( (output = fopen( "bfile.txt"

NULL)

printf( "Can't open bfile. txt for writing! In") ; fclose (input) ;

// If the files opened successfully, loop over the // input one character at a time. el se

while (ch getc(input) ) EOF)

// Process ch and output it. putc (chi output) ;

// Close the files fclose (input) ; fclose(output) ;



Output: This program copies the contents of afile.txt to bfile.txt, both files are in current directory (i.e. the directory in which this .c file resides). The following figure shows the output of the program:

+F-e Edlt Net' Eat Fum\* Help

A one -dimensional way 'g When is A one-drnrnsional array is used when it ig necessary to keep a Izge number ofItems necessary to keep a large number ofitems memory and reference all the items tn memory and all the in manner. C, elements of an 'Otuform manner. C, a] elements ofan array have Exe same fixed. predeterrrined array have the same fixed. predetermined size, and all have san-Le data type. size, and al have same du type

-2.1

Fig. 14.2 Afile.txt is copied to bfile.txt by the program

14.7 STRING HANDLING

Until now, we have not discussed the topic of string handling. This book will remain incomplete without having a discussion on strings. In most of the programs we have to work with strings. For example, we may want to keep a list of names and telephone numbers of our friends, a shopkeeper may need to prepare records of items and their prices in his shop, and a law-enforcement agency might be interested in keeping records of criminals including their names, pictures. telephone numbers and addresses; in all of these cases we need to handle strings. So, in this section we shall see how strings are handled in a C program.

In different programs, we have been displaying strings on screen with printf() function. But still wc are not familiar with string variables — the way C stores a string in a variable. Unlike variables of different numeric data types, C follows a different approach to handle strings. It stores a string as an array of characters. An array is a group of contiguous memory locations, which can store data of the same data type. Let us see how we can declare an array in C? The general form is:

data\_type arr\_nameln];

The data\_type specify the type of data that is stored in every memory location of the array, arr\_name describe the array name, and 'n' is the subscript of array which shows the total number of memory locations in the array. For example, the



int balls[6]; double temperaturell01;



|  |  |
| --- | --- |
|  | temperature[O] = 37; |
| balls[l] = 0; | temperature[2] = 26; |
| balls[4] = 6; | temperature[3] = 19;  \_TOS |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 37 |  | 26 | 19 |  |  |  |  |  |  |

define two arrays named balls and temperature (two sets of six and ten contiguous memory locations as shown below). In balls we can store six integer values, whereas in temperature we can store ten floating point values. Each value of array can be accessed via its subscripts. For example, consider the following statements:

balls(6] temperature[ 101

Now we have prepared a base for understanding string manipulation in C. As strings are amay of characters in C, that's why it was necessary to have a concept of arrays. We shall not prolong our discussion on arrays as it is out of scope of this book. You will study more about this topic in next classes. However, here we shall briefly discus strings — the array of characters.

14.7.1 Declaring and Initializing String Variables

As we mentioned earlier, a string in C is implemented as an array. So declaring a string variable is the same as declaring an array of type char, such as:

char name [16] ;

the variable name can hold string from 0 to 15 characters long. The last character of every string in C is ' \ O ' , the null terminator which indicates the end of the string. In this way the C let us manipulate each character of the string individually. Like variables of other data types, the strings can also be initialized:

char namel 161 = "Lahore";



Notice the above figure showing the memory arrangement for the string variable name; the name161 contains the character . This is the null character that marks the end of the string. This end marker allows the strings to have variable lengths. The rest of the memory locations in the amay remains empty and are not allocated to any other variables. All of the C's string

A OfCom ce 231



handling functions simply ignore whatever is stored in the cells following the null character. The following figure shows another string, longer than the previous, that the variable name can store.

char name[161 = "I love Pakistan";



Notice that, in the initialization statement of the string we did not put a null character (NO) at the end. When we initialize a string, a null character is added at the end of it by default.

14.7.2 String Assignment

Assigning a value to a string variable is not as simple as assignment to other variables. For example, we can assign an integer value to a variable of type int and a floating point value to a variable of type float by using assignment operator (i.e„ z). But, it does not work with strings. So the following statement will cause an emor:

name = "1 love Pakistan";

As name does not consist of a single memory location — it is an array. So, different characters arc put in different memory locations Of the array. This is done by copying every character of the string to respective index (subscript) of the array. For this purpose C provide a library for handling string manipulation i.e„ library of string.h. Most of the string manipulation functions of C are part of this library. There is a function named strcpy which is used to copy a string to an array of characters (i.e„ string variable). The syntax of strepy is as follows:

char\* strcpy (char\* dest, const char\* source) ;

Hence, the following statement will successfully copy the string to the variable name.

strcpy (name, love Pakistan") ;

14.8 STRING HANDLING IN TEXT FILES

When working with text files, C provides four functions which make file operations easier. The first two are called fputs() and fgers(), which write or read a



string from a file, respectively. Their prototypes are:

int fputs (char \*sty, FILE \*fp) char \*fgets (char \*str, int num, FILE \*fp)

The fputs() function writes the string pointed to by str to the file associated withfp. It returns EOF if an error occurs and a non-negative value if successful. The null that terminates str is not written and it does not automatically append a carriage return/linefeed sequence.

The fgets() function reads string of characters from the file associated with fp into a string pointed to by str until num-I characters have been read, a new line character (in) is encountered, or the end of file (EOF) is encountered. The function returns str if successful and a null pointer if an error occurs.

Example 3 Write a program that accepts name and telephone numbers of your friends and write them in a file.

#include

#include <string . void main (void)

FILE \*ptrFi1e; char name [30) ; char tel [ 11) ; if ( (ptrFiIe txt" ,

NULL) printf( "Can't open bfile.txt for writing! ;

// If the file opened successfully, Get the name // and telephone number and store them in the file else do

printf(nEnter the name (or press ENTER to quit) :

gets (name) ; if (strlen (name) > 0 )

printf ("Enter telephone number (max 10 characters) : " gets (tell ;

// write name and telephone number to file fputs (name, pt:rFi1e) ; fputs (" ! ptrFi1e) ; fputs (tel. ptrFi1e) ; fputs( "\n", ptrFi1e) ;

(name) > O) ; // Close the files. fclose (ptrFi1e) ;

This program demonstrates the typical use of strings in text files. A sentinel loop reads name and telephone numbers unless the user enters an empty string for the name. In addition to fgets() and fputs(), this program makes use of a new string handling function i.e., gets(). The gets function accepts a string from keyboard and assigned it to the variable tel (an array of characters). The contents of the file contacts. txt are as follows:

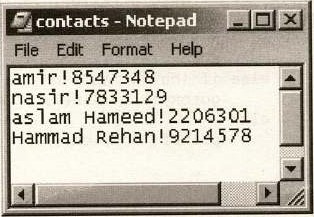
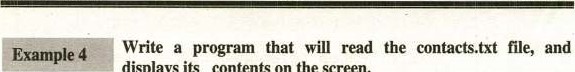


Fig. 14.3 Contents of Contacts.txt

Here an exclamation sign (!) separates the name and the telephone number fields in each record. We may use another symbol such as a colon (:), as a separator. In text files, a separator is used to mark the end of the data for one filed, Whereas the data for the next field follow this separator.



displays

its

contents

on

the

screen.

#include <stdio.h> \*include <conio void main (void)

FILE\* ptrFi1e; char ch; int line = 3; clrscr() ;

txt", "r") )

NULL) not open file") ; else

printf ( "Name") ; gotoxy (35, I) ; printf ( printf("————-— ;



while

(

(ch

=

getc

(ptrFi1e)

)

EOF)

if (ch = gotoxy (35, line) ; else if (ch



gotoxy

(1,

++line)

;

fclose (ptrFiIe) ; getch() ;

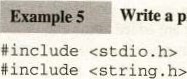
The function gotoxy() moves the cursor to a specified location on the screen. To use this function, the conio.h file must be included in the program. Its syntax is:

gotoxy(int Col, int row)

The arguments of the gotoxy() function specify the coordinates of the screen where the cursor should move to.

|  |
| --- |
| tput• This program reads the file contacts-txt and displays its contents the screen. The following is the output of the program:  Phone#  8547348  7833129 aslam Hameed 2206301 Hammad Rehan 9214578 |

The process of appending a file is same as that of writing a file, just open the file in append mode. Consider the following example:

Write a program that will append records in contacts.txt file.



o

void main (void)

FILE \*ptrFi1e; char name [30] ; char tel [11 ) ; if ( (ptrFi1e = fopen ( "d: \ NContacts . txt" ,

NULL)

open bfile.txt for writing! \nn) ;

// If the file opened successfully, get the name

// and telephone number and append them in the file

do

printf ( "Enter the name(or press ENTER tö quit) : 

gets (name) ;

if (strlen (name)

printf ( "Enter telephone number (max 10 characters) : , gets (tel) ;

/ // write name and telephone number to file fputs (name, ptrFi1e) ; fputs( " ! " ,ptrFi1e) ; fputs(tel, ptrFi1e) ; fputs ptrFi1e) ;

(name) > O) ; // Close the files. fclose (ptrFi1e) ;

This program seems very much similar to the program in example3 except that it opens contacts.txt in append mode, so new records are added at the end Of the contacts.txt file. The following figure shows the contents Of contacts.txt after appending three records:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| Nnrppad | | | | | |  |
| 8347348  aslam Hameed!2206301 Har-,unadRehan'9214578  Shoukat'9208754 Abdullah19203647  Zahid18745617 | | | | | |  |

Fig. 14.4 Contents of Contacts.txt after adding three more records

14.9 FORMATTED 1/0

The other two file handling functions to be covered are fprintfl) and fscanfl). These functions operate exactly like printft) and scanj() except that they work with files. Their prototypes are:

int fprintf (FILE \*fp, char \*control-string, ) int fscanf (FILE \*fp, char \*control—string . )



A

of

Com

ter

237

Instead of directing their I/O operations to the console, these functions operate on the file specified by fp. Other-vise their operations are the same as their consolebased relatives. The advantages to fprintfl) and fscanfl) is that they make it very easy to write a wide variety Ofdata to a file using a text format.

|  |
| --- |
| 6 |

'"nple Example3 can be re-written using formatted I/O as follows:

#include

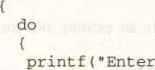
#include <string . void main (void)

FILE \*PtrFi1e; char name (30] ; char tel [111 ;

if ( (ptrFiIe = fopen 

NULL) open bfile. txt for writing! \nn) ;

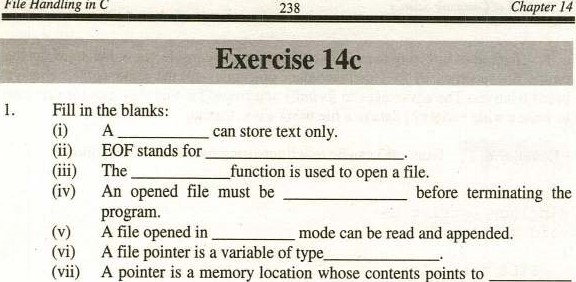
// If the file opened successfully, Get the name // and telephone number and store them in the file else

 the name(or press ENTER to quit) : " ) • gets (name) ; if (strlen (name) 0 )

printf( "Enter telephone number (max 10 characters) : " ) ; gets (tel);

// write name and telephone number to file fprintf (ptrFi1e,  name, eel) ;

Iwhile(strlen (name) > O) ; // Close the Eile. fclose (ptrFi1e) ;

A file pointer is a variable Of type

A pointer is a memory location whose contents points to memory location.

1. In C, every valid string ends with a
2. A string is an of characters.
3. The fopen() returns a if it fails to open a file for some

|  |  |
| --- | --- |
| 2. Choose the correct option:  (i) A file is stored in: |  |
| a) RAM | b) hard disk |
| c) ROM | d) cache |

1. Which Of the following mode open only an existing file for both reading and writing:
2. Which of the following functions is used to write a string to a file?
   1. puts() b) putc()

c) fputs() d) fgets()

1. On successfully closing a file, thefclose() returns:
   1. NULL b) O (zero)

c) I (one) d) FILE pointer

1. An array subscript should be:
   1. int b) float

c) double d) an array

A o co ter 12

|  |  |
| --- | --- |
|  | 239 |

1. Write T for true and F for false statement.
   1. A picture can not be stored in a text file.
   2. EOF marks the end Of a string.
   3. A null character marks the end of a text file.
   4. Text files are stored in a FILE\* (file pointer).
   5. The name of thearray points to its first element.

Array subscript is used to access array elements.

* 1. An array of characters can Store data of any data type.
  2. A binary file is a group of contiguous memory locations.
  3. C can handle text files only.
  4. When an existing file is opened in "w" mode, its contents are over, written.

1. Can a file be used for both input and output by the same program?
2. What is a stream? Illustrate the difference between text and binary streams.
3. How many modes are there for opening a file in C? Discuss characteristics of different file opening modes.

 What is a file pointer? Briefly explain the concept.

1. Write a program to merge the contents of two text files.
2. Write a program that counts the total number of characters in a text file.

[Note: consider the blank space a character]

1. Write a program that counts the number of words in a text files and display the count on the screen.

ANSWERS

Exercise lc

1. (i) Database Management System (ii) record (iii) records

 (iv) transaction files (v) databasedata dictionary

* 1. Structured Query Language(viii) Organizational Chart (ix) facts, figures, statistics (x)Information

1. (i) b d (iii)(iv)b
2. (i)(iii)T(v)

(vii) T (viii) F Ox)

Exercise 2c

Table (ii) Entity (iii) Insignificant

 Primary key (v) AttributeForeign key

* 1. DBA (Viii) Composite keyView

(x) data

1. (i)(iii)b
2. (i) T (ii) T (iii)

 (vi) T (vii)(viii) T fix)

Exercise 3c

 (i) Requirement analysisData Flow Diagram(iii) Data Modeling

 (iv) Cardinality(v) Mandatory

(vi) Entity Relationship Diagram (vii) Entity

 (viii) Centralized (is) Replicated (x)Distributed

1. (i)(iii)(iv)
2. (i)(iii) F

(vi) T (vii) T (viii)

Exercise 4c

1. (i) primary key (ii) Foreign key (iii) Complex

(iv) Functional dependency (v) Repeating (vi) I-NF

(vii) Transitive (Viii) Partial functional dependency

 (ix) Insertion anomaly (x) Foreign

1. (i)b(iii)b (v)
2. T (ii) T (iii)(iv)

 T Wii) T (viii) F fix) T (x)

Exercise sc

Integrated development environmentDBMS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | (v) | Query | (vi) | Auto number | (Vii) | Field |
|  | (viii) | Tuple or Record | Ox) | Form | (X) | Four |
| 2. | (i) | d a | (iii) | a | (v) |  |

 (iii) Relational database management systemTable

b

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | F | (iii) |  |  |  |
| (vii)  Exercise 6c |  | (viii) | T (ix) T | (x) |  |
| Primary key |  | (ii) | Datasheet | (iii) | Table |
| (iv) ERD |  | (v) | Right head arrow |  | composite |
| (vii) Wildcard |  | (viii) | Table | fix) | cross Table |
| (X) Asterisk |  | (xi) | Memo | (xii) | Cardinality |
| (Xiii) Degree |  | (Riv) | Select | (xv) |  |

 b (iii) a or d

 (vii) b (Viii)(ix)

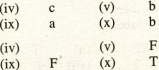
(i) T (iii) T (iv)

|  |  |  |  |
| --- | --- | --- | --- |
| (vi)  Exercise 7c |  |  |  |
| Enter, view, modify or delete |  |  | Wizard |
| (iii) Tabular form (iv)  Report (vii)  (viii) Columnar, Tabular, Justified | sub-form | (v) | Four |

 (vii)F fix) F

report

 fix) front end (x) Report



(ix)

1. (i) b b (iii) b (vii)(viii)
2. T (iii)

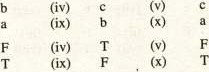
(vii) T (viii) T

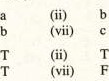
Exercise 8c

1. (i) Computer program American National Standard Institute

(iii) Source program Loader (v) Unstructured

(vi) Preprcxessor directives (vii) Constant Macro (viii) Include

 Function (X) Semicolon(;) (xi) Assembler

 (Xii) Syntax

2.(iii)

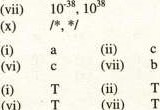
(viii)

3. (i)(iii)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exercise 9c |  |  |  |  |
| 1. (i) Letter | (ii) | Unary operators | (iii) | Variable |

(viii)

(iv) One -32768 , 32767 6. 65635

* 1. (viii)Comments

2.(iii)b

* 1. b

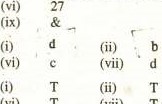
3.(iii)

(viii)

Exercise 10c

|  |  |
| --- | --- |
| 1. (i) getch | (ii) |
| (iii) Hexadecimal integers | (iv) |

(vii)



(vi)

T

T

 (x)

2.(iii)

(viii)

3.(iii) (vii) (Viii)

|  |  |
| --- | --- |
| Exercise llc |  |
| control structure | (ii) |
| flowchart | (v) |
| (vii) integer. character  (x) true, false | (viii) |

T (iii)



(vii)

 (Viii)

Exercise 12c  three (ii) nested loop (v)

(vii) three (Viii)

(x) label

 T (iii)

1. (vii)(viii)

Exercise 13c  function (ii)

(iv) lifetime 

1. one (viii)

(x) modular

1. (i) b (ii) a (iii)
   1. (vii) b (viii)
2. (i)(iii)
   1.  (vii) T (viii)

Exercise 14c

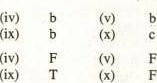
1. O) text file (ii)  closed (V)

(vii) another (Viii) (x) NULL  b (iii)

3. (i)(iii)

T

|  |  |  |
| --- | --- | --- |
| printf |  |  |
| Backslash O) | (v) | stdio.h |
| Octal | (viii) | 80 |

b

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | compound statement | | (iii) | condition | | nested if |  |  | switch-case | | break |  |  | default | | F | F | (v) |  | | repetition |  | (iii) | do-while | | goto |  | (vi) | loop | | true  T (iv) |  | (x) | complexity | | libraries |  | (iii) | prototype | | scope |  | (vi) | global | | formal |  | (ix) | actual | | b (iv) |  | (v) | c | |  | b | (x) |  | |  |



|  |  |  |
| --- | --- | --- |
| End Of File | (iii) | fopen |
|  | (vi) | FILE' |
| null character | (ix) | array |



 (iv) 

F fix) 

GLOSSARY



Attribute: The characteristics of an entity are also called attributes of the entity.

Alternate keys: Candidate keys are also called alternate keys. (see candidate keys]

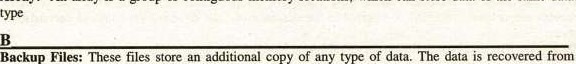
Action query: An action query makes changes in specified records of an existing table. or creates a new table.

Append query: An append query adds a group of records from one or more tables to the end of one or more tables.

Atomic value: A value which can not be further sub-divided or decomposed. For example, telephone number is not an atomic value because it can be decomposed in to country\_code, city\_code, and phone\_number. However, the and city\_code are atomic values because these can not vided.

ANSI: American National Standard Institute

Array: An array is a group of contiguous memory locations, which can store data of the same data



these files in case of loss of original file.

Binary operators: The operators which have two operands are called binary ovxrators.

Built-in functions: Built-in functions are predefined functions that provide us convenient ways to perform variety of tasks.

Binary stream: A binary stream is a sequence of bytes with a one-to-one correspondence to those on the external device.



Candidate key: There can be more than one keys or key combinations that qualify to be selected as primary key. However, in a relation there can be only one primary key. Rest of the keys or key combinations which uniquely' identify each record in the relation are called candidate keys.

Comm.site keys: These keys consists of more than on attributes and can be subdivided into simple attributes e.g. address which can be subdivided into house number, street. ciw and country etc. which are simple attributes.

Concatenate keys: Composite keys are also called concatenate keys.

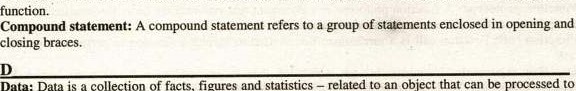
Control keys: Sort keys are also called control keys.

Cardinality of the relation: The number of record in a relation is called the cardinality of the relation Centralized Distribution: All data is at a single site.

Compiler: Compiler translates the source program into an object program with d'j extension.

Constant: Like variable, a constant is also a memory lcxation. However its contents can not be changed during program execution.

Control structures: Ihese are statements used to control the flow of execution in a program or function.



an

object

Data:

Data

is

a

collection

of

facts,

figures

and

prcxluce meaningful information

Data Files: •mese are the files that contain data and are created by the software tring used.

Database: A database is a collection of logically related data sets or files.

Data Set: A file is also called a data set [see file]

DBMS: DBMS stands for database management system. It is software which is used to manipulate the

Data dictionary: Most database management systems (DBMS) use a file to store the data definitions or a description of the structure of data used in the database which is called data dictionary.

Dependent table: The table in which theforeign key is found is caned dependent table.

Data administrator (DA): A data administrator (DA) is responsible for the entire data Of an organization. He normally develops the overall functional requirements for the databases being used in the office,

DBA: A database administrator (DBA) is responsible for the design, implementation, operation, management and maintenance of the database. He/She must technically expert on the overall intricacies of the database & DBMS.

Degree of a relation: The number of fields in a relation is called the degree of a relation or table.

Delete query: A delete query deletes a group of records from one or more tables.

DFD: Data Flow Diagram

Database Integrity: It referees to the correctness and consistency of data in the database.

Data anomalies or Database anomalies: These are certain situations created When one or more records are deleted, modified or inserted in the database and the database goes into an inconsistent



Entity: An entity is any thing about which you want to keep information in the database.

ERD: Entity Relationship Diagram

Entity Integrity Constraint: It states that primary key of a relation can not be null. EOF: End Of File



File: A collection Of related records treated as a single unit is called a file

Field: Each column Of the table in relational database is called a field. A field represents a characteristic of the entity.

Foreign key; A foreign key is an attribute in a table whose values must match a primary key in another table.

Functional dependency: It is a relationship between two attribute Of the same relation. It states that for any relation R. attribute B is functionally on attribute A, if for every valid instance Of A, the value of A uniquely determines the value of B.

First Normal Form (I-NE): A relation R is in I-NF. if and only if all underlying domains contain atomic values only.

Format specifiers: Format specifiers descritr the format in which the value of a variable should be displayed on the screen.

Field-width specifiers: Field-width specifiers describe the number of columns that should be used to print a value on the Screen.

Function: A function is a self-contained piece of code which performs a specific task.

Function prototype: A function prototype is a statement that provides the basic information that the Compiler needs to check and use•a function correctly.

Function call: Function call is a mechanism that is used to invoke a function to perform a specific

Function arguments: Variables through which the data is provided to a function are known as function arguments, These are specified in function header.



Global variables: The variables which are declared outside all blocks i.e. outside the main( ) and all Other functions are called global variables



Hierarchical Model: It is a database model in which the data is organized like an organizational chart. Every node in the chart represents an entity and its subordinate entities are described at next level of the hierarchical tree.

Hybrid Distribution: The database is partitioned into critical and non-critical fragrnents. Non-critical fragments are stored at one site while critical fragments are stored at multiple sites.

Homonym: In a database, a homonym is created when same name is used for two different attributes. High level languages: programming languages whose instructions resemble the English language are called high level languages.



Information: The manipulated and processed data is called information

Index: It is another table created by the system developer/DBA containing the key attributes Of the table for which the Index is created.

Indexed sequential file: The data in this type of file can be accessed sequentially as well as randomly based on a key value which is stored separately along with the address of each record in file.

Input mask: An input mask controls the value of a record and sets itin a specific format.

Identifiers: Identifiers are the names used to represent variables, constants, types, functions, and labels in the program.

Iteration: A loop is also called iteration. [see loop]



Keywords: Keywords are the words, which have predefined meaning in C



Linker: The linker is a program that combines the Object program with additional Object files that may be needed for the program to execute. It combines different library files to the object file and produces an executable file with .exe extension

Loader: Loader is a program that places executable file in memory.

Logical errors: Logical error occurs when a program follows a faulty algorithm. The compiler can not detect logical errors; therefore no error message is reported from the compiler.

Loop: It is a control structure. which repeats a statement or a group of statetnents in a program upto specified number of times or until a given condition is true.

Lifetime Of the variable: The duration in which a variable exists in memory is called lifetime Of the variable.



Master File: -mese are the latest updated files which never become empty. ever since they are created.

Macro: A macro is used to perform the same sequence of steps or automating tasks repeatedly.

Modality: Modality defines whether the participation of an entity in a relationship is mandatory or optional.

Mutually exclusive data: The data which dcrs not have overlapping information is known as mutually exclusive data.

Machine language: Machine language is the native language of the computer. The computer does ncn need any translator to understand this language.

Net-work database model: In this model, subordinate entities may participate in as many subordinate relationships as desired. It is more flexible than hierarchical database model.

Normalization: It is the process of converting complex data structures into simple data structures by following certain rules.

Nested if: An if statement within the body of another if statement is referred to as nested if statement. Nested loop: A loop within the of another is called nested loop.

o

Occurrence: In relational database, the tuple or record or row of the table is also called occurrence in that table.

OLE: Object linking and embedding.

Object File: It is binary file which the compiler produces.



executable

primary Key: In a relation, the attribute (column) or a combination of attributes (columns) that uniquely identifies a row or a record.

parent table: The table to which theforeign key refers is called as parent table.

Partitioned distribution: The database is divided into partitions. Each partition is assigned a particular site.

Partial functional dependency: A partial functiona] dependency exists if one or more non-key attributes are functionally dependent on part of the primary key.

Preprocessor: The preprcxessor is a program that modifies the C program (source program) prior to its compilation.



Query: Query is a statement that extracts specific information from database.



Record: A collection ofrelated fields (facts about something) treated as a single unit is called a record. Random file: Each record in this type Of file is accessed directly without going through the preceding records. 

Relational database model: This system consists of a collection of simple files/relations (Entities), each of which has no structural or physical connection such as those typically used in hierarchical or network systems.

Report generator: A report generator is a program that is used to produce an on-screen or printed document from the database.

Relation: In relational database, the table in Which data is stored is also called a relation.

Redundancy: Redundancy means duplication of data in multiple files.

Referential Integrity Constraint: This rule states that the foreign key value must match the primary key value in the other relation or the foreign key value must null.

- Relationship: Relationship indicates how the entities are connected or related to each other in the system.

Replicated distribution: In this data distribution strategy, multiple copies of the database are stored at different sites on the network.

Runtime error: A runtime error occurs when the program directs the computer to perform an illegal operation, such ms dividing a number by zero.

Reserved words: keywords are also called reserved words. [see keywords]



Repetition structure: A loop is also called repetition structure. [see loop]



Secondary key: A secondary key is non-unique field that is used as a secondary (alternate) key.

Sequential files: The data stored in these files are accessed sequentially i.e. to access a record from the file all preceding records must be accessed.

SQL: It stands for structured query language. It is used to create table structures, to enter data into them and to retrieve/update the selected records. based on the particular criteria and format indicated in the database.

Sort key: A sort key is used to physically sequence the stored date according to our need. Multiple attributes can be used as sort fields.

Scroll bars: Scroll bars are used to move around the window if its contents do not fit on-screen.

Select query: A select query gathers, collates and presents information in usable forms.

Synonym: In a database, a synonym is created when two different names are used for the same information (attribute).

Second Normal Form (2-NF): A relation is in 2-NF, if it is in I-NF and all non-key attributes of the relation are fully functionally dependent on primary key.

Source program: The program written in any high level programming language, such as C, is called

source program.

Syntax error: A syntax error cwcurs when the program violates one or more grammar rules of C language. Syntax: Every high level language defines a set of rules for writing programs called syntax of the language.

Sequence structure: In case of sequence structure, instructions are executed in the same order in which they are specified in the program.

Selection structure: A selection structure chooses which statement or a block of statements is to execute.

Scope of a variable: The of a variable refers to the region of a program in which it is accessible. Stream: Stream is a logical interface to a file which is associated to a file using an open operation.

Transaction File: These are those files in which data prior to the stage of processing is recorded. It may be temporary file, retained till the master file is updated.

Table: In relational database, the data is stored in the form of table (a two-dimensional array).

Tuple: In relational database, a row of the table represents a record which is also called a tuple.

Toolbars: Toolbars contain icon button that are shortcuts to the command in the menu.

Third Normal Form (3-NF): A relation is in 3-NF, if it is in 2-NF and no transitive deFndency exists.

Transitive dependency: It states that in a relation R, if an attribute B is functionally dependent on an attribute A. and the attribute C is functionally dependent on the attribute B. This implies that the attribute C is functionally dependent on attribute A.

Turbo C++: Compiler for C language

Text stream: A text stream is a sequence of characters.

Text file: A text file is a named collection of characters saved in secondary storage.

User: The user or end-user is simply a computers for his specific need. Update query: An update query makes changes to a group of records in one or more tables. Unary operators: The operators which have just one operand are called unary operators.



the

uses



View: The view is a virtual or temporary relation created by using SQL.

Variables: Variables are named memory locations (memory cells), which are used to store program's input data andits computational results during program execution.

INDEX



ASCII. 4

Attribute. 18.24

Alternate key, 19

Auto ccrrect, 79

Assembly language, 140

Arithmetic operator, 152

Actual argunrMs. 217

Array, 229



Backup file, 3

[Backup 10](#_Toc1176770)

[Break 185](#_Toc1176771)

[Built-in function 208](#_Toc1176772)

[Binary sueamc 223](#_Toc1176773)

Candidate key. 19 

Composite key, 20,

Cardinality, 25

&tatuse &sign, 28

Centralized distribution. 30

Computer program, 131

Compiler. 133-134

COBOL 141

Ccmstants. 149

Charxter cmstaMS. 149

Comments, 159

Control



175

Data,

Data file. 3

Database, 4

Data Integrity, 6.31. 

Data inderrndence, 6, 8

DBMS. 8, 27

Data dictionary. 10

DA, 20 DBA, 20

DFD, 23

Data Modeling. 23

[kletion aty•naly. 42.44

Database wizard. 49 Design view. 61-62

Datastret view. 62

Delete query, 85

Delimiter, 138

 Do-while. 195-198



Entity, 16, 23

ERD, 26,

Entity Integrity, 35

Escape sapence, 167-168



File. 2. 14.223

File mamgement system, 13

Field. 14

Fcreign key. 20

Functional dependency; 35

Form. 55

FORTRAN. 141 Format specifier. 164 field-wi'fth sFcifier. 165 Flowchart, 178

Function, 207

Functi«t 210

Formal arguments, 217 fputs(). 232

fÄ), 232



getch(). 170-171 getche(). 170-171

Statenrnt. 202

Hardware, 5

Hierarchical 6

Hy&id 30

Homonym. 36

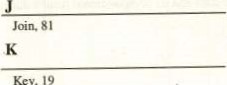
Hyperlink, 67 • High level language. 140



Inf«mation. I Index sequential file, 4 Index. 18, 31.

Inserticm anomaly, 42-43 Input mask. 71 inclu& directive. 137 l&ntifier, 145 if 176-178 if-else. 178 if-else-if, 183

Iterati«m. 193

Key, 19

Keywcrd, 146

Linker. 134

Loader. 134

Logical errrr,

Low level language: 140

Logical 153

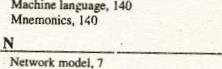
193

Lifetime Of a variable. 211 Lcxal variable. 211



i.-laster file, 3 Mcdality, 25 

Mutu.•'l Exclusive data, 36 arwnaly. 42, 44



Macro,

137

Norrnalization. 35-37

I-NF. 38



2.NF. 41-43

3-NF.43

Numeric constant. 149

Nested-if, 10182



Nested

198

23

OLE. 67 ObÉct



134

program file. 3 Primary key. 19.72

Physical database design. 29 Partitioæd data distribuim, 30

Prewcxesscr. 136 PreprcxessKX directive, 136 printf(). 163 Poimer. 226



Record. 2. 14

file. 4

Relatimal 7

Recovery. I Relational database. 15 Relatim. 15-16.

24.81

Replicated distributim, 30

Referemial 35.82

Redundancy. 36

Reixm. 56

Runtime error, 139

Rexrved WMds. 146

Relational %rratcr, 153

ReFtitim stnxtur. 176

s

Software, I

Squential file. 4 SQL 10 Secondary key.

Scyt key. 20

Synonym. 36 Select qtrry.

Source program, 134

Strwnzed Frogramming. 136

Syntax error. 139

Stan&rd i&.ntifier, 145 Signature, 163 ran(). 169 Sequence Structure. 175

Selection structure, 175

Switch. 185 SCOR 211 Stream. 223

stmM). 231



TutSoC, 132

•rent cream. 223

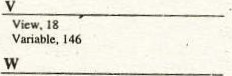
Update query. 85

Unix. 131

Unstructured programming, 136, 207

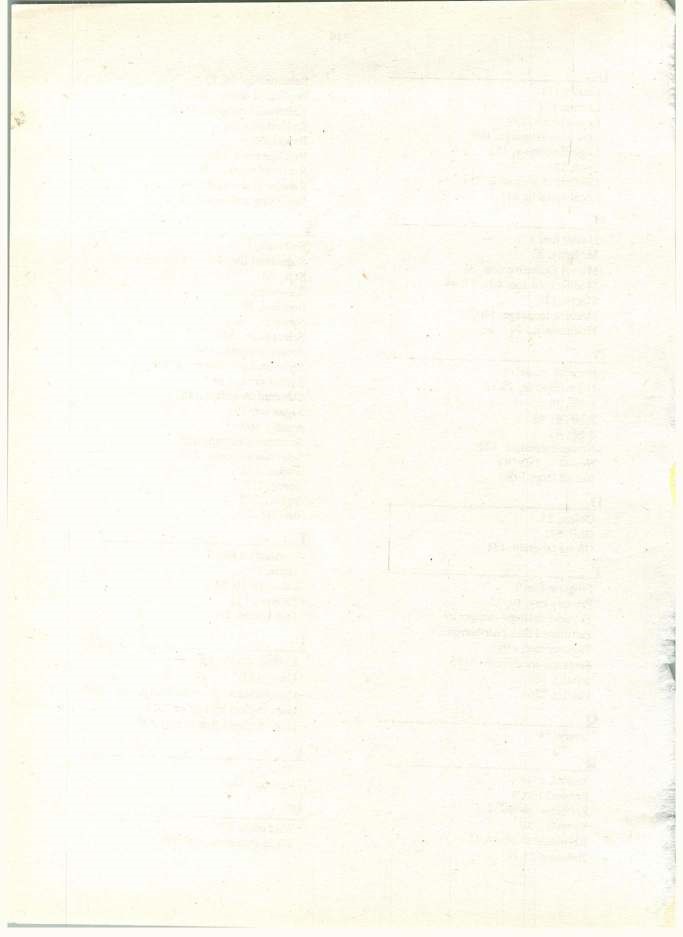
User-define ülentif•.er. 145

User-defined functions. 209



Wild cards, 88

While statement. 193-195



Smoking can cause a slow and painful death.



Punjab Curriculum and Textbook Board provides standard textbooks at low price according to the approved curricula. Suggestions are requested for improvement of these books by pointing out any error in spellings, contents, etc.

Fax No: oa2-gqz306tg Managing Director

E•man: chairman@ptb.gop.pk Punjab Curriculum and Textbook Board Website: 21-E-II, Gulberg-lll, Lahore.

Printed by UBC Convertec (Put) Lid  co

