R.V. COLLEGE OF ENGINEERING
BANGALORE

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

PROJECT REPORT

DESIGN AND IMPLEMENTATION OF A RELATIONAL DATABASE MANAGEMENT SYSTEM

Developed by:

Rahul Ravindran	1RV98CS066
Srivas N. Chennu	1RV98CS086
Sumanth G.	1RV98CS089
Vishwas N.	1RV98CS105

6th Semester CSE RVCE

R.V. COLLEGE OF ENGINEERING
Department of Computer Science & Engineering
Bangalore - 560 059.

CERTIFICATE

This is to certify that the mini project titled –

Design and Implementation of a Relational Database Management System

has been successfully completed by Srivas N. Chennu (Roll No: 1RV98CS086) in partial fulfillment of the Mini Project in the 6th semester B.E. (CSE) during the period March - August 2001 as prescribed by VTU.

Examiner 1 Guide

Examiner 2: HoD

Table of Contents

1. Synopsis
2. Introduction
3. Software Requirement Specification
4. Design
    a. High Level Design
    b. Low Level Design
5. Implementation
6. Testing
7. Conclusion
8. Bibliography

Synopsis

InfoBASE is a relational database management system designed and implemented for single user computer systems. It is intended for deployment in IBM – Compatible Personal Computer Systems.

The following are some of the features and characteristics of InfoBASE.

- InfoBASE has been implemented using the C programming Language. It's user interface has been designed using the Visual C++ Development Environment.

- The InfoBASE DBMS package consists of a set of generic library of software functions, which can be deployed by application layer software tools for database management.

- The InfoBASE package provides the following functionality:

  - Database Schema Definition

  - Database Creation and Deletion

  - Creation and Deletion of Database Tables

  - Efficient Storage and Retrieval of Data

- InfoBASE further incorporates advanced database management concepts, some of which are listed below.

  - Index Management

  - Database Recovery

Introduction

A Database Management System is a collection of software tools intended for the purpose of efficient storage and retrieval of data in a computer system. Some of the important concepts involved in the design and implementation of a Database Management System are discussed below.

The Database

A database is an integrated collection of automated data files related to one another in the support of a common purpose.

Each file in a database is made up of data elements – numbers, dates, amounts, quantities, names, addresses and other identifiable items of data.

The smallest component of data in a computer is the bit, a binary element with the values 0 and 1. Bits are used to build bytes, which are used to build data elements. Data files contain records that are made up of data elements and a database consists of files. Starting from the highest level, the hierarchy is as follows:

1. Database
2. File
3. Record
4. Data element
5. Character (byte)
6. Bit

The Data Element

A data element is a place in a file used to store an item of information that is uniquely identifiable by its purpose and contents. A data value is the information stored in a data element.

The data element has functional relevance to the application being supported by the database.

The Data Element Dictionary

A data element dictionary is a table of data elements including at least the names, data types and lengths of every data element in the subject database.

The data element dictionary is central to the application of the database management tools. It forms the basic database schema or the meta-data, which is the description of the database. The DBMS constantly refers to this Data Element Dictionary for interpreting the data stored in the database.

Data Element Types

Relevant to the database management system, there are a variety of data types that are supported. Examples of common data element types supported are numeric, alphanumeric, character strings, date and time.

Files

A database contains a set of files related to one another by a common purpose. A file is collection of records. The records are alike in format but each record is unique in content, therefore the records in a file have the same data elements but different data element values.

A file is a set of records where the records have the same data elements in the same format.

The organization of the file provides functional storage of data , related to the purpose of the system that the data base supports. Interfile relationships are based on the functional relationships of their purposes.

Database Schemas

A schema is the expression of the data base in terms of the files it stores, the data elements in each file, the key data elements used for record identification , and the relationships between files.

The translation of a schema into a data base management software system usually involves using a language to describe the schema to the data base management system.

Key Data Elements

The primary key data element in a file is the data element used to uniquely describe and locate a desired record. The key can be a combination of more that one data element.

The definition of the file includes the specification of the data element or elements that are the key to the file. A file key logically points to the record that it indexes

Interfile Relationships

In a database, it is possible to relate one file to another in one of the following three ways:

1. One to one
2. Many to one
3. Many to many

In such interfile relationships, the database management system may or may not enforce data integrity called referential integrity.

Data Models

The data in a database may be organized in 3 principal models:

1. **Hierarchical Data Model**: The relationships between the files form a hierarchy.
2. **Network Data Model**: This model is similar to hierarchical model except that a file can have multiple parents.
3. **Relational Data Model**: Here, the files have no parents and no children. They are unrelated. Here the relationships are explicitly defined by the user and maintained internally by the database

Data Definition Language

The format of the database and the format of the tables must be in a format that the computer can translate into the actual physical storage characteristics for the data. The Data Definition Language (DDL) is used for such a specification.

Data Manipulation Language

The Data Definition Language is used to describe the database to the DBMS; there is a need for a corresponding language for programs to use to communicate with the DBMS. Such a language is called the Data Manipulation Language (DML). The DDL describes the records to the application programs and the DML provides an interface to the DBMS. The first used the record format and the second uses the external function calls.

Query Language

The Query Language is used primarily for the process of retrieval of data stored in a database. This data is retrieved by issuing query commands to DBMS, which in turn interprets and appropriately processes them.

Software Requirements Specification

- The DBMS should be a generic software system, independent of any database instance.
- The DBMS should have a user friendly and interactive interface.
- The DBMS should support base DDL and DML operations.
- The DBMS should incorporate sufficient transaction control.
- The DBMS should have features for backup and recovery of data stored in the database.
- The DBMS should provide for efficient retrieval of required data elements.
- The DBMS should allow for constraints enforcement like data integrity, and referential integrity.

- The DBMS should allow for the definition of primary keys and indexing on the basis of keys.
- The DBMS should allow for the definition of dependencies between tables.
- The DBMS should allow for the primary data operations like select, project etc.
- The database tables must be modifiable with ease.
- The DBMS must allow for selection of the specific database instance.
- The DBMS must ensure portability without any re-compilation requirements.
- The DBMS must ensure that its operations are performed within reasonable limits of time and resources.

InfoBASE
The GUI-Based Database Management System

Design

The overall design description of InfoBASE can considered at two different levels of abstraction.

High Level Design

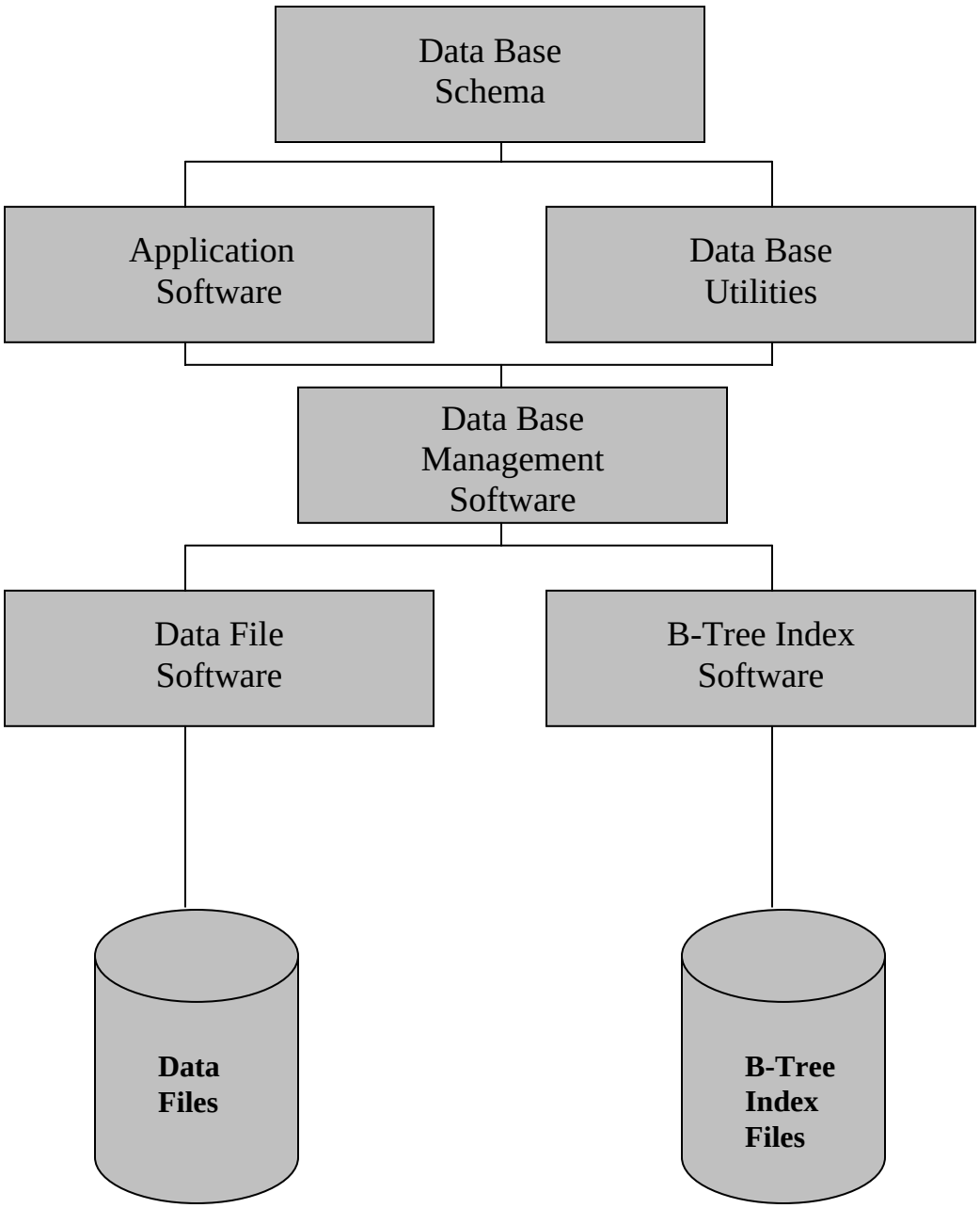
The concepts used in the design of InfoBASE extend the capabilities of the C programming language to construct the Data Definition Language and the Data Manipulation Language. These languages include three components of the automated data base environment:

- The schema
- The application software
- The DBMS

The schema is defined in a special file called the Schema file (.sch file), which incorporates the details of the various files making up the database. The information regarding the names of the tables, the names of the attributes, the types and lengths of the attributes etc. are stored in the schema file.

The DBMS is basically collection of software functions that provide the application software with a generic interface to the underlying data. By doing this, the DBMS abstracts the stored binary data to collections of records and files. The DBMS accepts and processes requests for data storage, retrieval and manipulation, coming from the application layer. The database schema, or meta-data is a dictionary used by the DBMS to interpret the data stored in a database.

High Level Design



InfoBASE Architecture

System Architecture of InfoBASE

InfoBASE manages a database consisting of **data files** and a **b-tree indexes**. The structure of InfoBASE is shown in the above diagram.

At the bottom of the figure are the data files and the index files. Data files are managed by a set of software functions that are dedicated to file management. Index files are managed by a different function set. The two sets of functions are unrelated and the files themselves are unconnected.

Above the data file and index file software is a library of functions whose purpose is to manage the database. Since the database is a collection of data and index files, the data base management functions call the data files and index file management functions directly. The database maintains the relationship that logically exists among the many files that constitute the database. The set of functions represent the InfoBASE Data Manipulation Language (DML).

Above the data base management software are two sets of programs, the application software and any utility programs. These programs are concerned with the files in the database.

Data Files

Each data base file described in InfoBase DDL has a corresponding disk file. The file name comes from the first eight characters of the data base file. The file consists of a header record and a series of fixed length data records. The header record is of the same length and format for all the files. This header record contains information that is used to add and delete records from the data file. The data file essentially consists of a sequential collection of fixed length records.

Index Files

InfoBASE supports the relational data model with inverted indexed into data files. The inverted index processes use B-tree algorithms.

The B-tree is a balanced tree of key values used to locate the data file record that matches a specified key argument. The tree is a hierarchy of nodes where each node contains from one to a fixed number of keys.

A B-tree consists of a root node and two or more lower nodes. If the total number of keys in the tree is equal to or less that the number that a node can contain, then only the root node exists. When that number exceeds, the capacity of a node, the root node splits into two lower nodes, retaining the key that is logically between the key values of the two new nodes. Higher nodes are parents of the lower nodes. Nodes store keys in a key value sequence. When the tree has multiple levels, each key in a parent node points to the lowed node that contains keys greater than the parent key and less than the next adjacent key in a leaf node. The nodes at the lowest level are called leaves. The keys in a leaf node point to the file records that match the indexed values. Since values occur at all levels in the tree, the first key in a leaf is preceded by a pointer to the record of a key value from a higher node.

Low Level Design

The following is a list of the important functions that form the library of software functions making up the InfoBASE DBMS.

Database Management Functions

db_open : Opens the database, populating the various array elements by either reading from the appropriate schema or by obtaining values from the user.

add_rcd: Adds a record to a file, checking for entity and refererntial intergrity constraints.

find_rcd: Finds a record in a file based on a specified key number and value.

first_rcd: Retrieves the first record from a file based on the sequence of the index specified by the key number.

last_rcd: Provides reverse functionality of first_rcd.

next_rcd: Retrieves the next record from a file based on the sequence of the index specified by the key number.

prev_rcd: Retrieves the previous record from a file based on the sequence of the index specified by the key number.

rtn_rcd: Returns a record previously retrieved by one of the previous record processing functions.

del_rcd: Deletes a record that was previously retrieved using any of the previous functions.

db_cls – Closes the currently open database.

Data File Management Functions

file_create : Used to create database and called by db_init

file_open : Used to open an existing data file

file_close: Used to close an existing data file

get_record: Used to retrieve a record that was previously stored in the data file

put_record: Used to rewrite a record to a file and is used when the record has been changed

delete_record: Used to delete a record located at logical record position

B-Tree Index File Management Functions

build_b: Used to establish a new b-tree

btree_init: Used to initialize the processing for an existing B-tree index file

btree_close: Used to close a B-tree that was opened by b-tree

insert_key: Adds a key to the b-tree

locate: Used to find a key value in the b-tree

deletekey: Used to delete a key from the b-tree

firstkey, lastkey, nextkey, prevkey, currkey: Used similarly to above, but returning the associated record for the corresponding collating key sequence.

keyval : Used to retrieve the key value associated with the current key pointer as positioned by insertion, deletion or search functions.

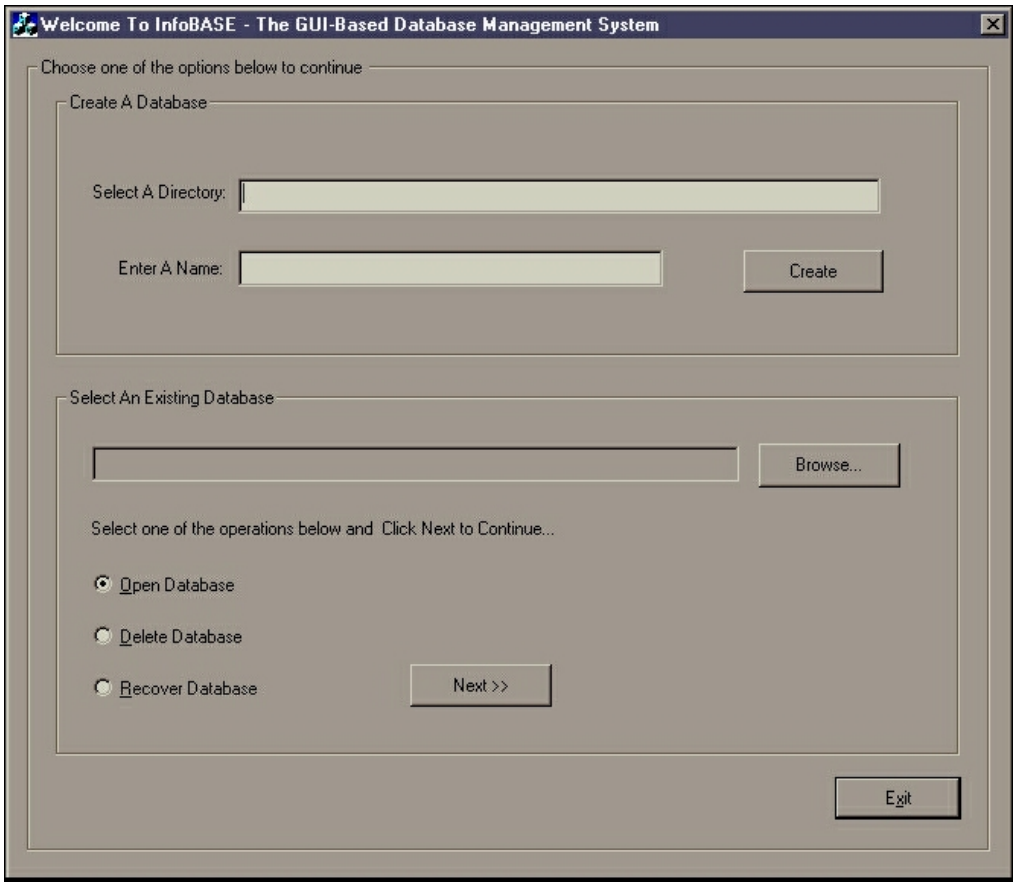
Testing

InfoBASE has been fully tested on the Windows 9x Operating System, installed on an IBM-PC Compatible System.

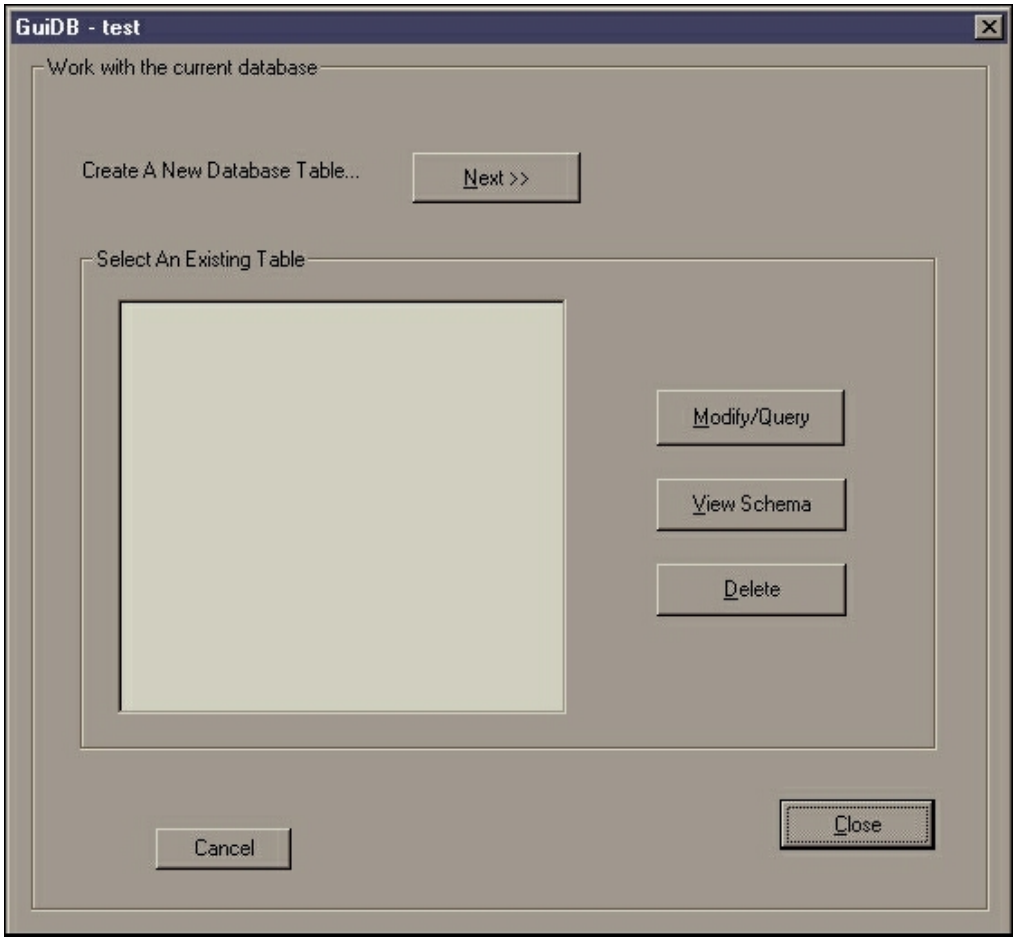
Performance restrictions and efficiency considerations of InfoBASE were affected by the performance of the Operating System on which it was deployed. The following issues were observed while testing of InfoBASE.

Below are test screens from InfoBASE.

The InfoBASE Main Window



The InfoBASE Current Database View



Conclusion

The InfoBASE Relational Database Management System was successfully designed, implemented, tested and deployed.

The InfoBASE system was found to provide satisfactory performance when working with reasonable size databases.

It was able to efficiently store, retrieve data from database tables stored on disk files as and when requested.

The InfoBASE front – end Application was able to suitably format and display the retrieved database information to the user in a comprehensive manner.

Bibliography

- Fundamentals of Database Systems – R. Elmasri and S. Navathe
- Introduction to Database Systems – C. J. Date
- The C Programming Language – B. W. Kerninghan and D. M. Ritchie
- Mastering Visual C++

The End