Macmillan Encyclopedia of Computers

Macmillan Publishing Company
A Division of Macmillan, Inc.
866 Third Avenue, New York, NY 10022

Maxwell Macmillan Canada, Inc.
1200 Eglinton Avenue East, Suite 200, Don Mills, Ontario M3C 3N1

Macmillan, Inc., is part of the Maxwell Communication Group of Companies.

Library of Congress Catalog Card Number: 91-45339

Printed in the United States of America

Printing number
3 4 5 6 7 8 9 10

Library of Congress Cataloging-in-Publication Data

Macmillan encyclopedia of computers / Gary G. Bitter, editor in chief.
p. cm.
ISBN 0-02-897045-4 (set). — ISBN 0-02-897046-2 (vol. 1). — ISBN 0-02-897047-0 (vol. 2)
1. Computers—Encyclopedias. I. Bitter, Gary G.
QA76.15.M33 1992
004'.03—dc20 91-45339
CIP

The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences—Permanence of Paper for Printed Library Materials. ANSI Z39.48–1984.

input? What happens if a lightning bolt strikes a thermocouple? Will it destroy the data acquisition hardware? Will it also destroy the computer? Both can happen.

The remote instrument approach is the best way to allow for the kind of high reliability just discussed, as the data acquisition hardware is physically and electrically separated from the computer. The board approach is the most risky, but may be entirely acceptable if the installer is careful and knowledgeable.

Price

Little needs to be said about price, except that each application has its price constraints, and each different hardware approach has a different price. Price must be weighed along with all the other factors to arrive at the best balance. In general, the boards are the least expensive, the remote instruments are the most expensive, and the proprietary board/remote instrument combinations are intermediate. This is because a remote instrument must contain its own power supply and also a communication subsystem to talk to the computer.

Software

Software is an important part of the total picture, as none of these hardware devices can be used without it. Many of the aforementioned characteristics, such as speed, throughput, channel type capability, channel number capacity, and (especially) ease of use, depend on the software as well as the hardware. This article is mainly about hardware, however.

Customer Support

It is likely that sometime in the life use of any data acquisition equipment, the manufacturer will have to be called for service or support. This important factor should not be overlooked. One good way to evaluate companies in this regard is to pose some good hard application questions to them before buying. Another method is to ask for and follow up on references.

SELECTING A DATA ACQUISITION BOARD

To match application requirements to data acquisition boards, first collect the literature on likely candidates. Then consider each of the aforementioned factors in the order of their importance to the application at hand, discarding candidates that do not meet the requirements. The list of candidates will narrow to two or three by the time this process is finished. If this is the case, consult a colleague if possible on the relative merits of the remaining candidates. Review the list one more time, rating each candidate on each factor with a score of 1 to 10, and then total the scores. The highest final score wins!

Frederick A. Putnam

DATABASE DESIGN, AUTOMATED

A database system is a collection of related records stored in a manner that makes the storage and retrieval of the data very efficient. The four well-known data models for databases are the hierarchical, network, relational, and object-oriented models.

The oldest of these models, the hierarchical model, was established out of necessity in the early 1960s without any prior formal study or definitions. Theoretically, the many-to-many relationship type between records (as when a student takes many courses and a course has many students) is not permitted in the hierarchical model. The IMS Data Base Management System (DBMS) package is the oldest and most dominant hierarchical DBMS package.

The network model is the product of the Database Task Group Committee (CODASYL 1975). Its first version appeared in 1961. This model permits all different types of connectivity: one-to-one, one-to-many, and many-to-many relationships (through the composition of two one-to-many relationships). Some network database packages are IDMS and IDMS/R of Cullinet Software, IDS of Honeywell, DMS II of Unisys, DBMS 10 and 11 of Digital Equipment, and Image of Hewlett-Packard.

The theoretical foundation for the relational database model was established by E. F. Codd in 1970 (Codd 1970). In this model, information is stored in a two-dimensional table called a relation. Each column represents a field of the record and is called an attribute. Each row of the table represents a data record of the file and is called a tuple. All the elements of the table must be simple. In other words, no element of the table can be a table on its own. The reservoir from which the values of an attribute are drawn is called the domain of that attribute. A process called normalization is used for grouping information in these tables so that duplicate values of attributes are eliminated. These tables are constructed in such a manner that

1. All the requirements of the client for whom the system is designed are satisfied.

2. The relationship between attributes within a table or between attributes of different tables or between tables themselves enables the user to add new data to the databases or to delete data from the databases or to update data without causing any anomaly (inconsistency) within a database.

Some well-known relational database packages are DB2 of IBM; Ingres of Relational Technology; Oracle of Oracle, Inc.; Unify of Unify, Inc.; dBASE of Ashton-Tate; Rdbase 5000 of Microrim; Informix of Informix Software, Inc.; and Paradox of Borland, Inc.

The object-oriented data model uses the concepts of entities and objects. An object is a representation of an entity in the database system environment. An entity has an infinite number of properties, but an object will take only a finite subset of these properties to represent the entity in the system. An object encapsulates both the data and the operations that can be performed on them. The operations are known as methods. Some examples of object-oriented database packages are Vbase Integrated Object Oriented System of Ontologic, Inc.; GEMSTONE/OPAL of Serviologic; and ORION of Micro-Electronic and Computer Technology Corporation.

In database terminology a record type means a file. In relational databases, record types are called relations, and fields are known as attributes. The record types in object-oriented databases are called objects, and the fields are called properties. In general, a file has one or more candidate keys. Each candidate key is a field or a group of fields that identifies a unique record within the file. Any of the candidate keys can be chosen as the primary key of the file; thus the primary key of a file identifies a unique record of the file. A foreign key is a field or a group of fields within a file that is the primary key of another file. The primary key is used to access a specific record from a file, and the foreign keys establish the links between different files.

A database management system (DBMS) is a software package. Its main functions are (1) to provide the facility to set up the database, (2) to retrieve and store source data (actual data in the database), (3) to retrieve and store the data about the structure of the database (data dictionary), (4) to provide the facilities to enforce security rules, (5) to back up the database, and (6) to control the concurrent transactions so that one user's environment is protected from others.

DATABASE DESIGN

Before discussing automated database design, it is essential to understand the meaning of database design and the processes involved. Basically, database design means identifying all the needed files, the fields of their records, and all candidate and foreign keys. Databases, like any other software system, have their own software design life cycle. The process involved in the design of database systems are:

1. Analysis of the required information. In this phase, like any other software system, a precise definition of the problem at hand is established from interviews and examination of the organizational documents. Also in this phase, the description of the system constraints, requirements, queries, transactions, and needed reports are expressed in short, concise sentences, usually

Teorey, J. T. 1990. *Database modeling and design—The entity-relationship approach.* San Mateo, Calif.: Morgan Kaufmann.

Welzer, T., I. Rozman, and J. Gyorkos. 1989. Automated normalization tool. *Microprocessing and Microprogramming* 25(1):375–80.

Asad Khailany

DATABASE MANAGEMENT

As mass storage prices fall and computers are used extensively throughout businesses and other organizations, vast amounts of valuable information are piling up in electronic repositories. Finding ways to organize this information and to allow convenient access to it is steadily becoming more important. The emergence of advanced concepts based on artificial intelligence will make the storage and management of such vast amounts of information even more critical.

Software developers have come up with a variety of database management systems, which accept, organize, store, and retrieve information quickly and efficiently. A *database* is a collection of related data that contains information about an enterprise such as a university or an airline. *Data* include facts and figures that can be represented as numbers, text strings, images, or voices stored in files on disk or other media. A *database management system* (DBMS) is a set of programs (a software package) that allows accessing and/or modification of the database. A major goal of the DBMS is to allow the user to deal with the data in abstract terms, rather than as the computer stores the data.

Before databases were introduced, users had to write programs to manipulate data stored in files (conventional file systems environment). Database management systems differ from the conventional file systems as follows.

Data dependence. File systems exhibit data dependence, whereas a DBMS exhibits data independence. That is, changes in the organization of physical data and/or storage device parameters are absorbed by the DBMS and, therefore, do not affect the user of the database. If data were stored in a conventional file, changes in the record length, for example, would affect the program that manipulates (or uses) those data.

Data redundancy. In file systems, data may be duplicated. For example, the address of a customer may appear in a file representing the data for orders received and in a file representing orders shipped. An objective of a DBMS is to avoid this type of redundancy by placing common information in a single file and then, as users request the information, applying a certain set of operations on stored data to obtain the necessary information.

Inconsistency. In file systems, there is no technique for checking whether various copies of data are consistent. For example, if the customer address changes, then the user might change that address in the order-received file and not in the order-shipped file; the customer address therefore becomes inconsistent. It is the responsibility of the programmer to manipulate the data so that they are consistent; if he or she fails to do so, the program may produce incorrect results. In DBMSs, the database is always in a consistent state because changes to one piece of data are automatically absorbed throughout the entire system. Note that in databases, for efficiency purposes, the data might be duplicated; however, this redundancy is controlled.

Lack of data integrity. Constraints may be imposed on certain data. For example, a customer can charge a sum greater than her or his allowed credit limit. In file systems, it is the responsibility of the user to determine if information entered violates the specific constraints imposed, in this case, a credit limit. In DBMSs, however, constraints can be handled automatically. If the data entered into the system violate the constraints, the system rejects the data and/or warns the user about them.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.