

IN DEPTH

Is your DBMS really relational?

Rule Zero: For any system that is advertised as, or claimed to be, a relational data base management system, that system must be able to manage data bases entirely through its relational capabilities.

By E. F. Codd

The originator of the relational model for data base management presents basic principles for determining how relational a DBMS product is — a question that faces many buyers today because almost every vendor claims its DBMS is relational. Some vendors may not realize how far from the mark they are.

Part 1

In recent years, the data base management system market has undergone a very rapid swing in favor of products that take the relational approach to data base management. It is hard to find a vendor that does not claim its DBMS is relational. This swing has been so extensive that some vendors of nonrelational DBMS have quickly (and recently) added a few relational features — in some, cases, very few features — in order to be able to claim their systems are relational, even though they may not meet the simple requirements for being rated "minimally relational." We shall refer to this kind of DBMS as "born again."

It is a safe bet that these Johnny-come-lately vendors have not taken the time or manpower to investigate optimization techniques needed in relational DBMS to yield good performance. This is the principal reason they continue to proclaim the "performance myth" — namely, that relational DBMS must perform poorly because they are relational!

One consequence of this rapid swing of the market to the relational approach is that products that are claimed by their vendors to be relational DBMS range from those that support the relational model with substantial fidelity to those that definitely do not deserve the label "relational," because their support is only token.

Some vendors claim that fourth-generation languages will provide all the productivity advantages. This claim conveniently overlooks the fact that most of these languages do little or nothing for shared data (the programming language fraternity

E. F. Codd is the originator of the relational model for data base management. He was the leader of the team that designed and implemented the first operating system with multiprogramming capability. Currently he is president of The Relational Institute and the Codd & Date Consulting Group, both based in San Jose, Calif.

IN DEPTH/RELATIONAL DBMS

still does not appear to realize that support for the dynamic sharing of data is an absolute requirement). In addition, there is no accepted theoretical foundation for fourth-generation languages and not even an accepted, precise definition.

This article outlines a technique that should help users determine how relational a DBMS really is. Accordingly, I shall discuss the following:

- The fidelity of DBMS to the relational model.

- The fidelity of the proposed Ansi SQL standard to the relational model.

- Conclusions regarding choosing a DBMS product.

I shall not attempt a complete description of the relational model here — a relatively brief and concise definition appears in the article "RM/T:

"

The fidelity of the proposed Ansi standard to the relational model is even less than that of some relational DBMS products. However, the standard could be readily modified to be more faithful to the model, and pressure should be brought on Ansi to do so.

Extending the Relational Model to Capture More Meaning," (Chapter 2, "The Basic Relational Model") in the Association for Computing Machinery's "Transactions on Data Base Systems" (December 1979). It is, however, vitally important to remember that the relational model includes three major parts: the structural part, the manipulative part and

the integrity part — a fact that is frequently and conveniently forgotten.

In this paper, I supply a set of rules with which a DBMS should comply if it is claimed to be fully relational. **No existing DBMS product that I know of can honestly claim to be fully relational**, at this time.

The proposed Ansi standard does not fully comply with the relational model, because it is based heavily on that nucleus of SQL that is supported in common by numerous vendors. Moreover, it takes a static, schema-based approach to data base description — reminiscent of Codasyl — instead of specifying a comprehensive, dual-mode data sublanguage that provides the powerful yet easy access to relational data bases and that is unique to the relational approach. Thus, the fidelity of the proposed Ansi standard to the relational model is even less than that of some relational DBMS products.

However, the standard could be readily modified to be more faithful to the model, and pressure should be brought on Ansi to do so. In fact, vendors are advised to extend their products soon in these respects so that they support customers' DBMS needs more fully and avoid possibly large customer expenses in application program maintenance at the time of the improvement.

The 12 rules

Twelve rules are cited below as part of a test to determine whether a product that is claimed to be fully relational is actually so. Use of the term "fully relational" in this report is slightly more stringent than in my Turing paper (written in 1981). This is partly because vendors in their ads and manuals have translated the term "minimally relational" to "fully relational" and partly because in this report, we are dealing with relational DBMS and not relational systems in general, which would include mere query-reporting systems.

However, the 12 rules tend to explain why full support of the relational model is in the users' interest. No new requirements are added to the relational model. A grading scheme is later defined and used to measure the degree of fidelity to the relational model.

First, I define these rules. Although I have defined each rule in earlier papers, I believe this to be the first occurrence of all 12 of them together.

In rules eight through 11, I specify and require four different types of independence aimed at protecting customers' investments in application programs, terminal activities and training. Rules eight and nine — physical and logical data independence — have been heavily discussed for many years.

Rules 10 and 11 — integrity independence and distribution independence — are aspects of the relational approach that have received inadequate attention to date but are likely to become as important as eight and nine.

These rules are based on a single foundation rule, which I shall call Rule Zero:

For any system that is advertised as, or claimed to be, a relational data base management system, that system must be able to manage data bases entirely through its relational capabilities.

This rule must hold whether or not the system supports any non-relational capabilities of managing data. Any DBMS that does not satisfy this Rule Zero is not worth rating as a relational DBMS.

One consequence of this rule: Any system claimed to be a relational DBMS must support data base insert, update and delete at the relational



"Is there a software solution that gives me control?"

Absolutely. It's BPCS.

Business Planning and Control System

BPCS is the solution. It gives your company firm control of its entire manufacturing/distribution operation. From inventory management and order entry to shop floor control and full financial reporting, BPCS coordinates every operational function... with ease!

BPCS is written in native code to maximize IBM's System/38 architecture. So with BPCS you get the power to take control of your entire operation today. And the flexibility you'll need to grow along with your business tomorrow.

Local BPCS affiliates provide technical support for easy installation and implementation. The affiliate network, composed of 43 professional consulting organizations in 23 countries worldwide, guarantees you a trusted partner in the search for a real solution.

BPCS is indeed that real solution. It's designed to accommodate the broadest scope of business operations but can be tailored to meet each individual user's needs. BPCS affirms your control in the most demanding situations.

Over 700 worldwide users attest to BPCS quality. Get the complete software solution designed to coordinate all your information needs. Get BPCS.

BPCS...Absolute Control For Your System/38.



System Software Associates, Inc.
200 West Madison Street
Chicago, Illinois, U.S.A. 60606
Telephone: (312) 641-2900
Telex: 280379 SSA CGO

©System Software Associates, Inc. 1985

gray wolf
Canis lupus
40-58 in. 27-175 lb.

SURVIVAL STRATEGY FOR SERIOUS DATA: PC/T

Could you survive the loss of your PC databases? Protect your future with Tallgrass. It's lonely out there for the corporate PC user, isolated from the company's database.

It's lonely for the DP/MIS pack, too, charged with ensuring that the proliferation of PCs promotes order, not chaos.

Tallgrass mass storage systems offer a three-point data survival strategy.

First: Backup PC data as religiously as you backup your mainframes and minis.

Second: Standardize on tape to provide PC users with easy access to centralized databases. After all, tape can be mailed cross-country, locked in a drawer and can handle far more capacity than the average PC.

Third: Most important, demand Tallgrass mass storage systems, with PC/T[®] to be sure that you get the file addressability, error correction capability and user-friendly simplicity only Tallgrass provides.

Call 1-800-228-DISK for the Tallgrass dealer nearest you. And come in from the cold.

For a framable print of the Canis lupus Gray Wolf, send a check for \$3 to: The Tallgrass Collection, Dept. W, 11100 W. 82nd St., Overland Park, KS 66214. Proceeds, plus an additional donation from Tallgrass, will be forwarded to the World Wildlife Fund.



**TALLGRASS
TECHNOLOGIES**

WHEN IT'S WORTH SAVING.

Tallgrass[®] and PC/T[®] are registered trademarks of Tallgrass Technologies Corporation © 1985 Tallgrass



Choose tape only or tape plus 25, 35, 50 or 80 megabytes hard disk.

IN DEPTH/RELATIONAL DBMS

level (multiple-record-at-a-time). Another consequence is the necessity of supporting the information rule and the guaranteed access rule.

"Multiple-record-at-a-time" includes as special cases those situations in which zero or one record is retrieved, inserted, updated or deleted. In other words, a relation (table) may have either zero tuples (rows) or one tuple and still be a valid relation.

Any statement in the manuals of a system claimed to be a relational DBMS that advises users to revert to some nonrelational capabilities "to achieve acceptable performance" — or for any reason other than compatibility with programs written in the past on nonrelational data base systems — should be interpreted as an apology by the vendor. Such a statement indicates the vendor has

not done the work necessary for achieving good performance with the relational approach.

What is the danger to buyers and users of a system that is claimed to be a relational DBMS and that fails on Rule Zero? Buyers and users will expect all the advantages of a truly relational DBMS, and they will fail to get these advantages.

Now I shall describe the 12 rules that, together with the nine structural, 18 manipulative and three integrity features of the relational model, determine in specific detail the extent of validity of a vendor's claim to have a "fully relational DBMS."

All 12 rules are motivated by Rule Zero defined above, but a DBMS can be more readily checked for compliance with these 12 than with Rule Zero.

"

Rule 1: All information in a relational data base is represented explicitly at the logical level and in exactly one way — by values in tables.

The information rule.

Rule 1: All information in a relational data base is represented explicitly at the logical level and in exactly one way — by values in tables.

Even table names, column names

and domain names are represented as character strings in some tables. Tables containing such names are normally part of the built-in system catalog. The catalog is accordingly a relational data base itself — one that is dynamic and active and represents the metadata (data describing the rest of the data in the system).

The information rule is enforced not only for user productivity but also to make it a reasonably simple job for software vendors to define additional software packages (such as application development aids, expert systems and so on) that interface with relational DBMS and, by definition, are well integrated with the DBMS.

That is, these packages retrieve information already existing in the catalog and, as needed, put new information in the catalog by the very act of using the DBMS.

An additional reason to enforce this rule is to make the data base administrator's task of maintaining the data base in a state of overall integrity both simpler and more effective. There is nothing more embarrassing to a data base administrator than being asked if his data base contains certain specific information and his replying after a week's examination of the data base that he does not know.

Guaranteed access rule.

Rule 2: Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

Clearly, each datum in a relational data base can be accessed in a rich variety — possibly thousands — of logically distinct ways. However, it is important to have at least one way, independent of the specific relational data base, that is guaranteed, because most computer-oriented concepts (such as scanning successive addresses) have been deliberately omitted from the relational model.

Note that the guaranteed access rule represents an associative addressing scheme that is unique to the relational model. The rule does not depend at all on the usual computer-oriented addressing. However, the primary key concept is an essential part of it.

Systematic treatment of null values.

Rule 3: Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information in a systematic way, independent of data type.

To support data base integrity, it must be possible to specify "nulls not allowed" for each primary key column and for any other columns where the data base administrator considers it an appropriate integrity constraint (for example, certain foreign key columns).

Past techniques entailed defining a special value (peculiar to each column or field) to represent missing information. This would be most unsystematic in a relational data base because users would have to employ different techniques for each column or domain — a difficult task because of the high level of language in

What looks like a neighborhood, acts like a hotel, and feels just like home?

Answer:

As close to home as we can make it.



The Residence Inn, of course. For no other place combines so many of the comforts of home with so many of the amenities of a fine hotel, so affordably. In a one-bedroom studio suite, or a two-bedroom, two-story penthouse suite. For a more detailed answer, or reservations, call 1-800-331-3131.



IN DEPTH/RELATIONAL DBMS

use (and a task that I believe would decrease user productivity).

Dynamic on-line catalog based on the relational model.

Rule 4: The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

One consequence of this is that each user (whether an application programmer or end user) needs to learn only one data model — an advantage that nonrelational systems usually do not offer (IBM's IMS, together with its dictionary, requires the user to learn two distinct data models).

Another consequence is that authorized users can easily extend the catalog to become a full-fledged, active relational data dictionary whenever the vendor fails to do so.

Comprehensive data sub-language rule.

Rule 5: A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items:

- Data definition.
- View definition.
- Data manipulation (interactive and by program).
- Integrity constraints.
- Authorization.
- Transaction boundaries (begin, commit and rollback).

The relational approach is intentionally highly dynamic — that is, it should rarely be necessary to bring the data base activity to a halt (in contrast to nonrelational DBMS). Therefore, it does not make sense to separate the services listed above into distinct languages.

In the mid-'70s, the Ansi Standards Planning and Requirements Committee generated a document advocating 42 distinct interfaces and (potentially) 42 distinct languages for DBMS. Fortunately, that idea has apparently been abandoned.

View updating rule.

Rule 6: All views that are theoretically updatable are also updatable by the system.

Note that a view is theoretically updatable if there exists a time-independent algorithm for unambiguously determining a single series of changes to the base relations that will have as their effect precisely the requested changes in the view. In this

regard, "update" is intended to include insertion and deletion as well as modification.

High-level insert, update and delete.

Rule 7: The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.

This requirement gives the system much more scope in optimizing the efficiency

of its execution-time actions. It allows the system to determine which access paths to exploit to obtain the most efficient code.

It can also be extremely important in obtaining efficient handling of transactions across a distributed data base. In this case, users would prefer that communications costs are saved by avoiding the necessity of transmitting a separate request for each record obtained from remote sites.

”

Rule 4: The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

Looking for the first CICS applications development tool that's as versatile as you are?



As a data processing manager, you wear a lot of different hats. Constantly balancing programmer resources, machine resources and time demands to get your different jobs done.

That's why you need CONSENSUS™ from Martin Marietta Data Systems. The first on-line applications development tool to let you develop applications three different ways and in whatever operating environment you choose.

With CONSENSUS, you can develop applications in COBOL and in 4GL procedural and non-procedural languages. CONSENSUS accesses almost every popular database and lets you develop in CICS, CMS, VM/PC or batch environments. And, most importantly, all CONSENSUS components are compatible with one another and with existing applications.

CONSENSUS is truly a breakthrough product. One which can change the way you develop CICS applications. It's another of Martin Marietta's Natural Selection® products—an interrelated family of products that work with an extraordinary variety of machines, environments, applications needs and degree of user sophistication.

If you thought you'd never find an applications development tool that's as versatile as you are, you're in for a nice surprise.

It's ready now.

Call 1-800-257-5171 today!

Martin Marietta Data Systems CW-1014
CONSENSUS Information

P.O. Box 2392, Princeton, NJ 08540

☐ I'd like a Representative to call.

☐ Please send me CONSENSUS literature.

Name _____

Title _____

Company _____

Address _____

City _____

State _____

Zip _____

Phone (_____) _____

MARTIN MARIETTA

Martin Marietta's CONSENSUS. We're ready now.

Nine Ways To Get The Word Processing Advantage.

1.

Use The Best Software.

MultiMate Advantage.[™] State-of-the-art word processing for IBM[®] PCs, XTs, ATs and compatible computers with 320K RAM. MultiMate Advantage builds on America's corporate choice for word processing—MultiMate Professional Word Processor—adding the features business users want most. DCA and WPS file conversion utilities let you exchange documents with IBM and WANG[®] dedicated word processors and mainframes. Plus Advantage includes MultiMate On-File,[™] the versatile information manager, at no extra charge.

New MultiMate Advantage Features:

- | | | |
|--|--|----------------------------|
| ■ Automatic footnoting | ■ Section numbering | ■ Enhanced merge utilities |
| ■ Automated merge data entry, forms generation, forms fill-in. | ■ Table of Contents Generation | ■ Key procedure replay |
| | ■ Line & box drawing for charts and graphs | ■ Text copy into libraries |



Professional
Word Processor

MULTIMATE
ON-FILE
INFORMATION MANAGER
INCLUDED AT
NO EXTRA CHARGE



2.

Make Keying Practically Effortless.

The MultiMate Business Advantage Keyboard brings the features of a dedicated word processing keyboard to your PC. Color-coded keys labeled with MultiMate commands. 22 additional programmable function keys for key procedures or text recall. Separate cursor controls and numeric keypad. The keyboard layout preferred by touch typists.

3.

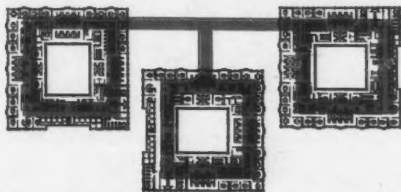
Cross The Integration Barrier.

The MultiMate Business Advantage Communications Board lets MultiMate-equipped PCs exchange documents directly with WANG® dedicated word processors. Switchable asynchronous and bisynchronous mode. Document interchange with full format integrity and error checking. The conversion software is included.

4.

Take Advantage Of Local Area Networks.

LAN versions of MultiMate Advantage and MultiMate 3.3 Series bring word processing to every desk. Hardware specific versions enhance IBM PC Net, 3Com™ and Novell/Netware 86™ LANs. Universal LAN version also available.



5.

Save With Site Licensing.

For large scale users, site licensing brings MultiMate convenience to every desk at substantially lower cost. You can even duplicate your own diskettes and documentation if you choose.



6.

Pick The Total Solution.

MultiMate has products for nearly all word processing needs. Combine MultiMate Advantage with the MultiMate keyboard and communications board, and you've got a word processing system that's hard to beat.

Or choose MultiMate 3.3 Series for heavy duty word processing on 256K computers. MultiMate Executive for the corporate leader who wants word processing in a hurry with full MultiMate compatibility. MultiMate On-File, the versatile information manager that handles almost any filing need. And Just Write™ the word processor for the occasional user.



7.

Look To The Future.

MultiMate is moving up with word processing for the IBM System 36 and mainframes. And MultiMate for minis—all coming soon.

SYSTEM 36 & MAINFRAME

8.

Look For Excellent Support.

MultiMate's technical support is the best in the industry with on-line assistance available every business day. User newsletters and training programs, too.



9.

Call 800-842-8676 ext. 902

You'll get the good word on MultiMate products and services. Or send some words of your own to: MultiMate International Corporation, 52 Oakland Ave., East Hartford, CT 06108. In Connecticut call (203) 522-2116, ext. 902

MultiMate
We're in the write business!

IN DEPTH/RELATIONAL DBMS

Physical data independence.

Rule 8: *Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.*

To handle this, the DBMS must support a clear, sharp boundary between the logical and semantic aspects on the one hand and the physical and performance aspects of the base tables on the other; application programs must deal with the logical aspects only.

Nonrelational DBMS rarely provide complete support for this rule — in fact, I know of none that do.

Logical data independence.

Rule 9: *Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.*

Take the following two examples: splitting a table into two tables, either by rows using row content or by columns using column names, if primary keys are preserved in each result; or combining two tables into one by means of a nonloss join (Stanford University and MIT authors call these joins "lossless").

To provide this service whenever possible, the DBMS must be capable of handling inserts, updates and deletes on *all views* that are theoretically updatable. This rule permits logical data base design to be changed dynamically if, for example, such a change would improve performance.

The physical and logical data independence rules permit data base designers for relational DBMS to make mistakes in their designs without the heavy penalties levied by nonrelational DBMS. This, in turn, means that it is much easier to get started with a relational DBMS because not nearly as much performance-oriented planning is needed prior to "blast-off."

Integrity independence.

Rule 10: *Integrity constraints specific to a particular relational data base must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.*

In addition to the two integrity rules (entity integrity and referential integrity) that apply to every relational data base, there is a clear need to be able to specify additional integrity constraints reflecting either business policies or government regulations.

Assume the relational model is faithfully reflected.

Then, the additional integrity constraints are defined in terms of the high-level data sublanguage and the definitions stored in the catalog, not in the application programs.

Information about inadequately identified objects is never recorded in a relational data base. To be more specific, the following two integrity rules apply to every relational data base:

Entity integrity. No component of a primary key is

allowed to have a null value.

Referential integrity. For each distinct nonnull foreign key value in a relational data base, there must exist a matching primary key value from the same domain.

If, as sometimes happens, either business policies or government regulations change, it will probably become necessary to change the integrity constraints. Normally, this can be accomplished in a fully relational DBMS by changing one or

"

Rule 11: *A relational DBMS has distribution independence.*

more of the integrity statements that are stored in the catalog.

In many cases, neither the application programs nor the

terminal activities are logically impaired.

Nonrelational DBMS rarely support this rule as part of the DBMS engine, where it belongs. Instead, they depend on a dictionary package, which may or may not be present and can readily be bypassed.

Distribution independence.

Rule 11: *A relational DBMS has distribution independence.*

HOW TO MAKE A GREAT IMPRESSION AT THE OFFICE

IN DEPTH/RELATIONAL DBMS

By distribution independence, I mean that the DBMS has a data sublanguage that enables application programs and terminal activities to remain logically unpaired:

- when data distribution is first introduced (if the originally installed DBMS manages nondistributed data only);

- when data is redistributed (if the DBMS manages distributed data).

Note that the definition is

carefully worded so that both distributed and nondistributed DBMS can fully support Rule 11. IBM's SQL/DS and DB2, Oracle Corp.'s Oracle and Relational Technology, Inc.'s Ingres (all nondistributed in present releases) fully support this rule.

This has been demonstrated as follows: SQL programs have been written to operate on nondistributed data (using System R) run correctly on distributed versions of that data (using System R*,

the IBM San Jose Research Laboratory prototype), and the distributed Ingres project at the University of California at Berkeley has shown the same capability for the Quel language of Ingres.

It is important to distinguish distributed processing from distributed data. In the former case, work (for example, programs) is transmitted to the data; in the latter case, data is transmitted to the work. Many nonrelational DBMS support distributed

processing but not distributed data. The only systems that support the concept of making all the distributed data appear to be local are relational DBMS — these are prototypes right now.

In the case of a distributed relational DBMS, a single transaction may straddle several remote sites. Such straddling is managed entirely under the covers — the system may have to execute recovery at multiple sites. Each program or terminal ac-

tivity treats the totality of data as if it were all local to the site where the application program or terminal activity is being executed.

A fully relational DBMS that does not support distributed data bases has the capability of being extended to provide that support while leaving application programs and terminal activities logically unpaired, both at the time of initial distribution and whenever later redistribution is made.

There are four important reasons why relational DBMS enjoy this advantage:

- **Decomposition flexibility** in deciding how to deploy the data.

- **Recomposition power** of the relational operators when combining the results of subtransactions executed at different sites.

- **Economy of transmission** resulting from the fact that there need not be a request message sent for each record to be retrieved from any remote site.

- **Analyzability of intent** (owing to the very high level of relational languages) for vastly improved optimization of execution.

Nonsubversion rule.

Rule 12: If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).

In the relational approach, preservation of integrity is made independent of logical data structure to achieve integrity independence. This rule is extremely difficult for a "born-again" system to obey because such a system already supports an interface below the relational constraint interface. Vendors of "born-again" systems do not appear to have given this problem adequate attention.

(Part two: the practical consequences of the 12 rules and an evaluation of certain products against the relational model.)

With the Hewlett-Packard LaserJet Printer.

Page 2
1985-86 Marketing Report

indicating a lower overall risk than had originally been projected.

Market Penetration

Since introduction in 1976, the product has experienced tremendous growth in all geographical areas. In fact, the only quarter-to-quarter exception occurred Q1-Q2 1980, when the rate of penetration stalled as a result of the \$35 coupon offered by the leading competitor (See Fig. 2a).

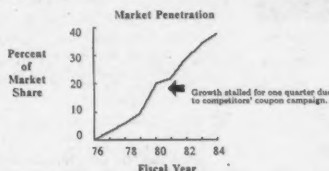


Figure 2a. Market Penetration (all geographic areas)

All regions are contributing to this growth, especially the Southern Region, which is experiencing a growth in market penetration far greater than the industry average. In the last three quarters, the Southern Region has increased at a rate twice that of the same period in the previous year. Figure 2b compares Southern Region and overall company performance with industry growth rates.

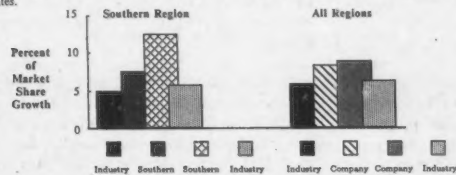


Figure 2b. Market Growth Rate Comparison

This would indicate that the increased effort directed at the dealers in the South has proved successful. No other elements were altered.

Impact on Profitability

After expenses for the new dealer program, profits have increased 29% in the Southern Region. In the other regions, profits have held steady. This indicates that the ROI for dollars allocated

No matter what business you're in, the impressions you make on paper have a lot to do with the impressions you make on people.

And nothing makes a better impression than Hewlett-Packard's LaserJet Printer.

Super sharp, publication-quality printing. The blackest blacks. A wide variety of graphics. All printed in resounding silence. Eight times faster than a daisy wheel for a standard page of text. And completely compatible with all leading PCs and software.

The HP LaserJet Printer will even make a good impression on your ledger sheet. Just \$2995.* For new enhanced graphics: LaserJet PLUS, \$3995.*

Call (800) FOR-HPPC, Dept. 282B, for the Hewlett-Packard dealer nearest you and see an impressive demonstration.



HEWLETT
PACKARD

Print sample produced on an HP LaserJet Printer using Polaris PrintMerge™ Polaris Software, San Diego, California. (619) 471-0922

*U.S. List Price

IBM/38-36 BACKLOG REDUCTION

The world's most successful companies have made Fusion Products Intl. the leading supplier of query/report-processor and spreadsheet software for the IBM/38-36. Call 415 461-4760 or write.

Fusion Products International
900 Larkspur L.C. Suite 295
Larkspur, CA 94939, Telex 176099

FUSION