
# White Paper
# Data Mining—Rapid Access to Buried Information

What is data mining? In layman's terms it is simply a systematic rummaging though potentially unrelated files of data to retrieve some type of information–preferably information that is of use to you. It is important to remember, though, that producing a message saying, "`NO MATCHES FOUND,`" is often more valuable than producing an unmanageably large listing of matching information.

## Common Characteristics of Data-Mining Jobs

Data-mining jobs have very few common characteristics aside from being expected to return relevant data. And as previously mentioned, the data returned may be a simple "`NO MATCHES FOUND`" message. You are probably delighted with such a response if the query is "Which customer has owed us more than a million dollars for more than six months?" Below we'll address other traits that data-mining jobs often share. Consider this a buffet table from which you can pick freely.

### *Urgency*

Most data-mining jobs have a high degree of urgency about them. If the average IT job request has a due date of yesterday, the information expected from a data-mining request is usually required as of last week. This typically means that the job needs to run immediately, and because it may be a long-running job, it may need to access files that are being read and updated in real-time by a wide range of users.

### *Short Shelf Life*

Often coupled with the urgency of data-mining jobs is a short shelf life for the results. A search firm may be looking for a candidate with an engineering degree who speaks French fluently and has ten years of experience. If the firm is unable to locate a satisfactory candidate today, the contract will be awarded to another firm and the immediate relevancy of this information goes away.

The short shelf life of the information and the short time allotted for producing results usually creates a willingness to compromise on the look and feel of the final output. If a request is made for a list of telephone numbers where the account is delinquent, a user may be perfectly happy to get the information in the same order that it is stored on the file (in this case the information is stored in account number sequence). Here is a typical data-mining job to produce such a report:

Example 1

```
INPUT=VSAM
PRINT
ONLY BALANCE > 0
SELECT ACCT, PHONE, BALANCE
```

We have left it to the data-mining software to format the report, which looks like this:

```
ACCT            PHONE                BALANCE

10272626     555-329 1221             138.62
10692630     555-395 1350            8192.96
10782628     555-350 3009            1057.95
12132631     555-406 5611           88758.40
12212627     555-340 8501               3.64
12412629     555-361 0280            6536.70
13412626     555-329 2350               4.94
13792630     555-395 1574               6.65
13932628     555-350 9233             250.30
etc.
```

It is possible that this request produced a list of customers numbering in the thousands, when the submitter was anticipating only a few hundred customers. Thus, the decision is made to focus on the most important accounts, which will better match the results of the query with the resources available to process them. By changing the `ONLY` command to `ONLY BALANCE > 5000` and running the job again, you shift the focus to only those accounts that are delinquent by more than $5,000.00.

## *Incomplete Results May Be Acceptable*

It may seem shocking that we are sometimes prepared to accept incomplete (or overly complete) information; however, such data is frequently preferable to getting no information at all. For example, we may want to offer a new credit card to all our wealthy customers. Unfortunately, we do not keep our customers' annual income or net worth on file. But we do maintain zip codes for their current residences. If we were to select everyone who resides in what we know to be exclusive zip codes (for example, Beverly Hills 90210, Oyster Bay 11771, Aspen 81611), we would get a higher percentage of success than if we merely offered cards to everyone in the country, or to everyone with a surname beginning with the letter *A*. We will miss the owner of the Greenville shopping mall and have included the person living in the 8'x8' room behind the gas station in Beverly Hills, yet the results are acceptable–despite being incomplete.

## *Unscheduled and Irregular*

Most organizations have regular production cycles (daily, weekly, monthly, etc.) that produce standard reports–invoices, statements, and so on. The typical data-mining activity or ad hoc examination of data is unscheduled and not required on a regular cycle. This generally increases urgency and requires it to run without disrupting production online systems. It is not acceptable to inform a CEO that the report he needs has to run at 3:00 A.M. when the online systems are down for scheduled maintenance. If he is willing to make online access unavailable for approximately an hour, he can have the report faster though

this obvious loss of productivity would also be deemed unacceptable. It is advisable to be able to access online CICS files without needing to take them offline for data-mining jobs.

## *Connecting the Unconnected*

When application files and systems are designed, it is often the case that no virtual pairings or logical relationships exist between related files. An example could be the Employee file and the Sales file. And yet that is exactly the kind of connection that may need to be made when running a query. A relational database manager like DB2 can accomplish the task with the help of a database administrator and a programmer, but the turnaround time may be too long to meet our urgency requirements. If the data lives in sequential and KSDS files, the process is even longer. Data-mining tools avoid these pitfalls by dynamically building tables of data from any type of file.

Here is a simple query to detect "phantom employees." Phantom employees typically have employee numbers but are not assigned to any specific department. This is one method used by employees to fraudulently supplement their salary.

Example 2

```
INPUT=VSAM      FILENAME=EMPFILE
EMPNO       1     6     C
EMPDEPT     135   5     C
SALARY      312   6     P     2
 TABLE DEPTTAB      DEPFILE
 DEPNO    1  5     C
PRINT                   [what we print is named in the SELECT statement at the end]
 FIND DEPTTAB  WHERE DEPNO = EMPDEPT
 ONLY DEPTTAB:IO-RESULT = 'NFD'                [Only those with no matching dept]
SELECT EMPNO EMPDEPT SALARY                    [Fields to print]
```

This is the entire program to produce the required report. The primary input is a KSDS, which gets read sequentially. A sequential file (DEPFILE) is used to build a table of departments which can then be searched, and only those records from EMPFILE that do not have a matching entry in the department table get printed. The table has only one column; it could be defined with any number of columns and have any number of rows, limited only by the amount of storage available to the job. Any column can be searched, and values from the table can be used in calculations, reports, etc. An important point here is that we have given database facilities to datasets that are not part of a database without having to import them into a database. So there is no conversion effort required.

## *Data Security*

Every effort must be made to protect sensitive information, especially when it leaves the control of the data center. For example, you may need to send a file of severely delinquent customers to your overseas call center for follow-up. The only information required to be sent is the name, phone number, and

the amount due. Instead of copying the entire customer file, you can select the exact information (certain fields of pertinent records) to be sent to the call center.

Example 3

```
INPUT=VSAM      FILENAME=CUSTOMER
INCLUDE CUSTREC.C                    [Get field descriptions from copy book]
OUTPUT=VSAM FILENAME=DEBTEXT
EXTRACT                              [What we extract is named in the SELECT later]
 ONLY BAL120 > 0                     [Only those 120 days past due]
SELECT ACCTNO, PHONE, BAL120         [Fields to print]
```

In this example, we extracted only the desired fields to an ESDS (VSAM file) to be sent to the call center.

## *Verification of the Data-Mining Job*

Some data-mining jobs can run for several hours–especially those involving a large number of files and tables. So a limiting mechanism is needed to allow a check on the output before committing a lot of machine resources. Although you could write your own logic (for instance, add `1` to record count, and if the record count is greater than `1,000`, it's end of job), data-mining tools have one or more ways of limiting the amount of work they do. We've already examined the `ONLY` facility, which lets us select records that meet certain criteria. Jobs can also be limited by giving a maximum number of records to read from or write to a file or by giving a starting key for a KSDS. (This is also useful if you want only to report on newer accounts, which presumably have higher account numbers.) These limiting factors are set with simple parameters on commands (such as `MAX=500` on the `INPUT` command) or by simple individual commands such as `STOP ACCTNO > '54321'`.

## *Going Beyond Simple Extraction*

Although detailed reports may be appropriate for simple call center activities like contacting delinquent accounts, you may need to get a bit more sophisticated to achieve even better results. An obvious thing you will want to do is sort your output so records are grouped together (for example, by zip code) or get the most important records to the top of the listing (for example, the people who owe you the most). It is also possible that some information may be captured from the files only by performing one or multiple calculations. Again using our debtors example, we may think that someone who has owed us $1,000 for six months needs chasing more than someone who has owed as $5,000 for 30 days. In the following example, we have put that lot together, and for good measure, we look up which of our offices is responsible for collecting the debt, based on zip code.

Example 4

```
INPUT=DISK FILENAME=CUSTOMER
INCLUDE CUSTREC
TABLE OFFICE FROM OFFFILE
OFFID     1     6     N
OFFNAME   56    20    C
OFFZIP    132   5     N
* Lay out the debtor report, sorted by office & weighted debt
* Break & total for each office
REPORT       DEBTREP
SORT   OFFZIP,WTAMT
LINE   DEBTID,DEBTNAME,DEBTAMT,OFFID,OFFNAME
BREAK        OFFZIP
SUM          DEBTAMT
* End of the report layout
DEFINE   WTAMT   8    P   2                          [Weighted amount owed]
AUTO INPUT DEBTORS
FIND   OFFICE WHERE OFFZIP = DEBTZIP
* Compute a weighted debt value
WTAMT = DEBTAMT * (DEBTAGE - 120)
ONLY WTAMT > 1000                                    [Ignore small debts]
PRINT DEBTREP
```

## *Other Uses for Data Mining*

We have concentrated largely on debt management as a use for data mining, but the possibilities and applications are endless. Employee addresses can be matched with branch locations to help people work closer to home; social security and other identifying numbers can be checked for duplicates to ensure they are valid; old inactive accounts can be identified for purging or for contacting to ask why they are inactive. The possibilities are bound only by the collective imagination of everybody in your organization and the many reporting requirements of the government.

## Conclusion

It is possible with very little effort to join normally unrelated files together in a data-mining operation to extract valuable business information in a number of different forms. The job can be simple enough to be created by non-technical users from outside the IT department on an as-required basis, or incorporated into a regular schedule of work. And it's easy to get as much or as little information as you want, sorted in the sequence that you need.

## A Word About the Examples in This White Paper

All the examples in this white paper were created with Data-Miner from CSI International. For the sake

of brevity, field definitions have been left out of the examples; they can be supplied either in shorthand form (for example, `CUSTNO 1 8 C`, meaning an eight-character field starting at byte `1`) or included in the job from a COBOL copybook.

For more information, call 800.795.4914 or visit the Website: www.CSI-International.com