The Effective Extraction and Manipulation of Data
Abstract

• Data mining is the process of extracting raw data to produce useful information.
• A data-mining solution can increase the efficiency of IT and end users by
  ✓ Filtering excess data from files and reports
  ✓ Building subsets of files for testing
  ✓ Empowering more people to take advantage of centralized data—and with minimal interaction with IT
Abstract

• Data mining allows users to extract data, produce a variety of reports, and change data in the source

• A data-mining solution allows users to create sophisticated reports without advanced programming knowledge
Accessing company data—what are the issues?

• Data storage practices have changed, but the need to access company data quickly is greater than ever.
• Centralized storage is popular again—users rely on mainframe systems.
• Many users are dependent on IT professionals to search for information from large volumes of data for and depending on business needs.
BUT...

• Data usually belongs to the line of business or department that maintains it.
• Even though IT’s responsibilities (like data security) keep expanding, IT owns only a fraction of the data.
• IT often is unable to service information requests as quickly as needed.
• A solution is needed for both users and IT.
Data-Mining Challenges

• Both end users and IT professionals have the same basic requirement—to transform lots of data into valuable information.

• This transformation is called **data mining**. Data-mining (DM) tasks often are characterized by
  ✓ **Urgency**. Many data-mining requests are urgent, which means jobs must be run right away, possibly on files that are in use.
Data-Mining Challenges

✓ **Short shelf life.** Today’s businesses cannot wait for information. Information that is only a few hours old may not accurately represent key elements of the business.

✓ **Acceptance of incomplete results.** This may seem counterintuitive, but often some output data is considered better than no output data.
Data-Mining Challenges

✓ **Ad hoc scheduling.** Companies often produce scheduled reports. In contrast, DM tasks are usually unscheduled. This means that DM software must enable key files to be available 24/7.
Data-Mining Challenges

✔ Unconnected data sources. When application files are designed, there often are no virtual pairings or logical relationships between related files. Yet a DM task may require searching such files to compare information. DM software must be able to build data tables dynamically from different types of files.
✓ Data security. Protecting sensitive information is paramount. In some cases, the output from a DM task must be obfuscated. In other cases, the output must be restricted such that only certain fields of certain records are included.
A Good Data-Mining Solution…

• Serves the needs of IT professionals
• Is simple enough for end users to learn
• Allows users to fix corrupted records
• Accesses data in datasets or databases
• Can join unconnected data sources
• Produces reports of varying complexity
• Allows for ad hoc reporting
• Provides security for the output data
Data-Mining Software Must Provide…

• A method to define
  ✓ Input and output data sources
  ✓ Required data fields
  ✓ User variables
Data-Mining Software Must Provide...

• A simple scripting language that can
  ✓ Read and write to multiple data sources
  ✓ Process conditions and flow control
  ✓ Process record-selection criteria
  ✓ Manipulate record fields and user variables
  ✓ Sort datasets and reports
Data-Mining Software Must Provide…

• Basic functions that can easily
  ✓ Print simple or complex reports
  ✓ Copy partial or entire datasets
  ✓ Extract selected data fields
  ✓ Update fields in selected records
  ✓ Delete selected records
Data Mining by IT Professionals

Test-Data Generation

- Test data that reflects current conditions and software is critical
- Test data must contain a sampling of many records
- Test data cannot contain sensitive information
Data Mining by IT Professionals

Test-Data Generation (continued)

• Data-mining software can
  ✓ Copy any number of records, at any interval (frequency), from production files
  ✓ Limit the selection of record fields
  ✓ Cloak or randomize sensitive fields

• To change copied data for testing, a script could include one of the following data-manipulation routines (Example 1):
Example 1

IF FIRST-NAME > "THOMAS"
   LAST-NAME = "LINCOLN"
ENDIF

–OR–

ACCOUNT-BALANCE = ACCOUNT-BALANCE + (AGE * CHILDREN)

• These script commands use unrelated fields to change potentially sensitive values
Test-Data Generation (continued)

- If the script commands are kept in the member CLOAKIT, the entire copy job could look like this:

```
Example 2
COPY
  INPUT=VSAM FILENAME=PRODFIL
  OUTPUT=VSAM FILENAME=TESTFIL
  INCLUDE LAYOUT
  INCLUDE CLOAKIT
```
Test-Data Generation (continued)

- Commands like ONLY and SKIP can be used to select certain records, as follows:

Example 3

```
IF ZIP > '40000'
  BALANCE = 0
ENDIF
LAST = 500
ONLY RECTYPE = "D"
SKIP BALANCE < 0
```
Data Mining By IT Professionals

Problem Determination

• Finding problem records can be time consuming.

• A data-mining program can save time by producing a smaller extract file containing relevant records.

• For example, an application has abended because a numeric BALANCE field contains a character value (“Buckeye”). A data-mining script (Example 4) can be used to find the offending record.
This simple data-mining job dumps only records where BALANCE = “BUCKEYE”—in this case, the 72,138th record:

```plaintext
RECORD NUMBER - IN= 72,138, OUT= 1 LRECL= 223
  0 F1F8F6F8 <snip> D46ED1D6 C8D54040 *18625...eJOHN *
 20 40404040 <snip> C2E4 C3D2C5E8 C540 * ...cBUCKEYE *
 40 40404040 40404040 40404040 [etc.]
```
Data Mining by IT Professionals

Status & Performance Reporting (IT)

• Many reports are generated in IT that often contain more data than is needed.
• For example, a network outage report can contain columns of zeros because failures occur so infrequently. This can make it harder to identify problem values.
• A data-mining job can easily run these reports and omit lines that have no useful information. ONLY & SKIP commands can be used as a filter.
Data Mining by IT Professionals

Sorting Data In Datasets

• Data-mining software can provide a way to sort datasets. The software must
  ✓ Use a simple syntax
  ✓ Allow for multiple keys
  ✓ Sort in ascending or descending order
  ✓ Provide an easy method for both input and output processing (similar to E15/E35 exits)
• Example 5 sorts data in a VSAM dataset.
Example 5

INPUT=VSAM FILENAME=ACCOUNT /* Input VSAM account dataset
NAME 9 32 C
BALANCE 52 10 P

OUTPUT=DISK FILENAME=OUTFILE /* Output SAM dataset
DEFINE TOTAMT 12 P /* User var to hold total bal

*** Sort in ascending order by name
SORT ACCOUNT INTO NEWACCT BY NAME
DURING INPUT
SKIP BALANCE < 0 /* Skip all negative balances
DURING OUTPUT
ADD BALANCE TO TOTAMT /* Keep running total of bal
AFTER OUTPUT

*** Display total balance when finished
DISPLAY 'TOTAL BALANCE IS ' TOTAMT
Calling External Programs

- Often some existing logic must be incorporated into your data-mining script.
- This logic might exist in the form of a load module written in COBOL or assembler.
- A data-mining solution must be able to interface with these existing programs.
- Example 6 shows how a data-mining script can call a decompression routine.
Example 6

INPUT=VSAM FILENAME=COMPFIL /* Define compressed input dataset
COMPREC 1 1000 C
OUTPUT=VSAM FILENAME=OUTFILE /* Define uncompressed output DSN
DECOMPREC 1 4000 C
READ COMPFIL /* Read compressed record
DO WHILE COMPFIL:IORESULT !='EOF' /* Do while not end of file
   /* Call module DECOMP to decompress the record
   CALL DECOMP USING COMPREC UNCOMPREC
   <rest of script>
   WRITE OUTFILE /* Write to the output dataset
   READ COMPFIL /* Read another compressed record
ENDDO
Data Mining by IT Professionals

Creating Extract Datasets for External Uses

• Sensitive information must be protected. This is especially true when it leaves the control of the data center.

• Data-mining software can quickly create a dataset with only necessary fields. All sensitive fields can be excluded.

• Example 7 creates a dataset containing data from a customer file on severely delinquent customers. Only selected fields are included.
Example 7

INPUT=VSAM FILENAME=CUSTOMER
INCLUDE CUSTREC /* Include field descrip from copybook
OUTPUT=VSAM FILENAME=DEBTEXT
EXTRACT
ONLY BAL120 > 0 /* Only those 120 days past due
SELECT NAME, PHONE, BAL120 /* Fields to extract
Troubleshooting Data-Mining Scripts

• Data-mining software must provide commands to assist in debugging problems with user scripts.

• Commands are needed to
  ✓ Display the value of record fields and user variables.
  ✓ Dump the contents of record fields and user variables.
  ✓ Trace the command flow in a script.
  ✓ Cancel a program and set a return code.
Data Mining for Business Users

Extracting Data from Datasets—Simple Example

• A basic data-mining job is shown in Example 8.
• A request is made for a list of telephone numbers on accounts that are delinquent. The requester is perfectly happy to receive the information in the order that it is stored on the file.
Data Mining for Business Users

Extracting Data from Datasets (continued)

• Example 8 prints selected record fields defined in the copybook CUSTREC. The PRINT command automatically reads each record and prints it if BALANCE is greater than 0.

Example 8

```
INPUT=VSAM
FILENAME=CUSTOMER
INCLUDE CUSTREC
PRINT
ONLY BALANCE > 0
SELECT ACCT, PHONE, BALANCE
```
Extracting Data from Datasets (continued)

Example 8 output is shown below. The DM software used the names of the selected fields as column headings.

<table>
<thead>
<tr>
<th>ACCT</th>
<th>PHONE</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10272626</td>
<td>555-329-1221</td>
<td>138.62</td>
</tr>
<tr>
<td>10692630</td>
<td>555-395-1350</td>
<td>8192.96</td>
</tr>
<tr>
<td>10782628</td>
<td>555-350-3009</td>
<td>1057.95</td>
</tr>
<tr>
<td>12132631</td>
<td>555-406-5611</td>
<td>88758.40</td>
</tr>
<tr>
<td>12212627</td>
<td>555-340-8501</td>
<td>3.64</td>
</tr>
<tr>
<td>12412629</td>
<td>555-361-0280</td>
<td>6536.70</td>
</tr>
</tbody>
</table>
Data Mining for Business Users

Extracting Data from Datasets (continued)

• The output could contain thousands of entries when only several dozen were anticipated. Thus, the decision is made to focus on accounts where the amount due is greater than $5000.

• This is accomplished by using

  \text{ONLY \text{BALANCE} > 5000}

• Data-mining software can increase the efficiency of IT staff and end users by filtering excess data from files and reports.
Data Mining for Business Users

Formatting Reports—Simple to Complex

• Data-mining software can format the output to make reports more meaningful. The following four examples show how increasingly complex reports can be created.

• Example 9 creates a basic report from records in an employee VSAM file.

• Subsequent examples show the effect of adding formatting options to this job.
Example 9

INPUT=VSAM  FILENAME=EMPVSAM
EMPNO  1,6,C
LASTNAME  22,13,C
DEPT  35,3,C
SALARY  73,5,P,2
COMM  83,5,P,2

AUTO EMPVSAM      /* Auto-read input file*/
PRINT EMPRPT       /* Write a line to named report */
* DEFINE THE REPORT
REPORT EMPRPT
LINE 1 DEPT EMPNO LASTNAME SALARY COMM
### Example 9 output:

<table>
<thead>
<tr>
<th>DEPT</th>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>SALARY</th>
<th>COMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>D11</td>
<td>000170</td>
<td>YOSHIMURA</td>
<td>46726.40</td>
<td>1974.00</td>
</tr>
<tr>
<td>D11</td>
<td>000160</td>
<td>PIANKA</td>
<td>42144.29</td>
<td>1780.00</td>
</tr>
<tr>
<td>D11</td>
<td>000150</td>
<td>ADAMSON</td>
<td>47857.78</td>
<td>2022.00</td>
</tr>
<tr>
<td>C01</td>
<td>000140</td>
<td>NICHOLLS</td>
<td>29946.00</td>
<td>2274.00</td>
</tr>
<tr>
<td>C01</td>
<td>000130</td>
<td>QUINTANA</td>
<td>25095.00</td>
<td>1904.00</td>
</tr>
<tr>
<td>A00</td>
<td>000120</td>
<td>O'CONNELL</td>
<td>169515.15</td>
<td>2340.00</td>
</tr>
<tr>
<td>A00</td>
<td>000110</td>
<td>LUCCHESI</td>
<td>269424.05</td>
<td>3720.00</td>
</tr>
</tbody>
</table>
Data Mining for Business Users

Formatting Reports (continued)

• Example 10 adds a report title, sorts by department, and creates summary totals. (Record field definitions are shown in Example 9 only.)

Example 10

```
AUTO EMPVSAM     /* Auto-read input file */
PRINT EMPRPT     /* Write a line to named report */
* DEFINE THE REPORT
REPORT EMPRPT
  LINE 1 DEPT EMPNO LASTNAME SALARY COMM
  TITLE 1 'EMPLOYEE LIST'
  ORDER DEPT   /* Sort by department */
  SUM SALARY COMM /* Create summary totals */
```
Data Mining for Business Users
Formatting Reports (continued)

- Output from Example 10 job:

<table>
<thead>
<tr>
<th>DEPT</th>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>SALARY</th>
<th>COMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>00010</td>
<td>HAAS</td>
<td>305634.48</td>
<td>4220.00</td>
</tr>
<tr>
<td>A00</td>
<td>000110</td>
<td>LUCCHESI</td>
<td>269424.05</td>
<td>3720.00</td>
</tr>
<tr>
<td>A00</td>
<td>000120</td>
<td>O'CONNELL</td>
<td>169515.15</td>
<td>2340.00</td>
</tr>
<tr>
<td>B01</td>
<td>00020</td>
<td>THOMPSON</td>
<td>43417.50</td>
<td>3300.00</td>
</tr>
<tr>
<td>C01</td>
<td>00030</td>
<td>Kwan</td>
<td>40267.50</td>
<td>3060.00</td>
</tr>
<tr>
<td>C01</td>
<td>00130</td>
<td>QUINTANA</td>
<td>25095.00</td>
<td>1904.00</td>
</tr>
<tr>
<td>C01</td>
<td>00140</td>
<td>NICHOLLS</td>
<td>29946.00</td>
<td>2274.00</td>
</tr>
</tbody>
</table>

---

1326550.86  39753.00
Example 11

AUTO EMPVSAM     /* Auto-read input file */
PRINT EMPRPT     /* Write a line to named report */
* DEFINE THE REPORT
REPORT EMPRPT
   LINE 1 DEPT EMPNO LASTNAME SALARY COMM
   TITLE 1 'EMPLOYEE LIST'
   ORDER DEPT        /* Sort by department */
   BREAK DEPT       /* Break by department */
   SUM SALARY COMM  /* Create summary totals */
Data Mining for Business Users

- Output from Example 11:

<table>
<thead>
<tr>
<th>Date</th>
<th>EMPLOYEE LIST</th>
<th>PAGE</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/03/2011</td>
<td>EMPLOYEE LIST</td>
<td>PAGE</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPT</th>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>SALARY</th>
<th>COMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>000010</td>
<td>HAAS</td>
<td>305634.48</td>
<td>4220.00</td>
</tr>
<tr>
<td>A00</td>
<td>000110</td>
<td>LUCCHESI</td>
<td>269424.05</td>
<td>3720.00</td>
</tr>
<tr>
<td>A00</td>
<td>000120</td>
<td>O'CONNELL</td>
<td>169515.15</td>
<td>2340.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>744573.68</td>
<td>10280.00</td>
</tr>
<tr>
<td>B01</td>
<td>000020</td>
<td>THOMPSON</td>
<td>43417.50</td>
<td>3300.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>43417.50</td>
<td>3300.00</td>
</tr>
<tr>
<td>C01</td>
<td>000030</td>
<td>KWAN</td>
<td>40267.50</td>
<td>3060.00</td>
</tr>
<tr>
<td>C01</td>
<td>000130</td>
<td>QUINTANA</td>
<td>25095.00</td>
<td>1904.00</td>
</tr>
<tr>
<td>C01</td>
<td>000140</td>
<td>NICHOLLS</td>
<td>29946.00</td>
<td>2274.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>95308.50</td>
<td>7238.00</td>
</tr>
</tbody>
</table>
Data Mining for Business Users

Formatting Reports (continued)

- Example 12 uses the NEWPAGE option to force a page break after each dept. summary. The dept. number is added to a second TITLE, and employees are ordered by descending salary.

Example 12

REPORT EMPRPT
LINE 1 DEPT EMPNO LASTNAME SALARY COMM
TITLE 1 'EMPLOYEE LIST'
TITLE 2 'FOR DEPARTMENT' DEPT
ORDER DEPT -SALARY /* Order by dept., salary */
BREAK DEPT NEWPAGE /* Break by department */
SUM SALARY COMM /* Create summary totals */
### Output from Example 12:

<table>
<thead>
<tr>
<th>Date</th>
<th>EMPLOYEE LIST</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/03/2011</td>
<td>FOR DEPARTMENT A00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEPT</td>
<td>EMPNO</td>
</tr>
<tr>
<td>A00</td>
<td>000010</td>
<td>HAAS</td>
</tr>
<tr>
<td>A00</td>
<td>000110</td>
<td>LUCCHESI</td>
</tr>
<tr>
<td>A00</td>
<td>000120</td>
<td>O'CONNELL</td>
</tr>
<tr>
<td>***</td>
<td>======</td>
<td>=========</td>
</tr>
<tr>
<td>A00</td>
<td></td>
<td>744573.68</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>EMPLOYEE LIST</th>
<th>PAGE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/03/2011</td>
<td>FOR DEPARTMENT B01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEPT</td>
<td>EMPNO</td>
</tr>
<tr>
<td>B01</td>
<td>000020</td>
<td>THOMPSON</td>
</tr>
<tr>
<td>***</td>
<td>======</td>
<td>=========</td>
</tr>
<tr>
<td>B01</td>
<td></td>
<td>43417.50</td>
</tr>
</tbody>
</table>

[Additional lines not shown]
Data Mining for Business Users

Relating Unconnected Data Sources

• To search unconnected data sources, data-mining software can dynamically build a data table from different types of files. This table can then be searched to compare or match data.

• Example 13 is a simple query to search for phantom employees. These employees have ID numbers but are not assigned to a specific department.
Example 13

** Define employee VSAM dataset, record fields
INPUT=VSAM   FILENAME=EMPFILE
EMPNO 1 6 C  
EMPDEPT 135 5 C  
SALARY 312 6 P 2

** Define department sequential dataset; load table
INPUT=DISK   FILENAME=DEPFILE
TABLE DEPTTAB FILE DEPFILE
DEPNO 1 5 C

** Locate & print emp. with no matching department number
PRINT        /* Print basic report
FIND DEPTTAB WHERE DEPNO = EMPDEPT  /* Search dept table
ONLY DEPTTAB:IO-RESULT = 'NFD'    /* Only no-matching dept
SELECT EMPNO EMPDEPT SALARY       /* Select fields to print
Example 13 contains the entire query needed for the desired report.

The primary input is a VSAM KSDS (read sequentially). The sequential file (DEPFILE) is used to load a searchable table of departments.

Only records from EMPFILE that do not have a matching entry in the department table are printed. The table has a single column; it could be defined with any number of columns.
Data Mining and Databases

• Data-mining software can update or extract record fields from a database and print them to a report.

• Data-mining software should be able to
  ✓ Use a dynamic SQL interface
  ✓ Work without a pre-compile or a bind
  ✓ Support SELECT, INSERT, UPDATE, DELETE, COMMIT, and ROLLBACK
  ✓ Allow for multiple cursors
Example 14 produces a simple list of employees earning over $50K.

Example 14
CONNECT SSID(DB9G) PLAN(DATAM71C)  /* Connect to DB

***Define the SQL statement:
INPUT=SQL(SELECT * FROM EMPLOYEE WHERE SALARY > 50000)
PRINT /* Print each row

***Print only selected columns:
SELECT WORKDEPT,EMPNO,LASTNAME,SALARY
### Output from Example 14:

<table>
<thead>
<tr>
<th>WORKDE</th>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>000010</td>
<td>HAAS</td>
<td>196947.05</td>
</tr>
<tr>
<td>D21</td>
<td>000070</td>
<td>PULASKI</td>
<td>70485.07</td>
</tr>
<tr>
<td>A00</td>
<td>000110</td>
<td>LUCCHESI</td>
<td>173605.49</td>
</tr>
</tbody>
</table>
Data Mining and Databases

Example 15 produces an ordered and organized report of all employees.

**Example 15**

```
CONNECT SSID(DB9G) PLAN(DATAM71C)  /* Connect to DB
***Define the SQL statement
INPUT=SQL STATEMENT=EMPSEL(SELECT * FROM EMPLOYEE)
AUTO EMPSEL                      /* Auto read the table
PRINT SALRPT                     /* Print a line to the report
***Layout the report
REPORT SALRPT
LINE WORKDEPT,EMPNO,LASTNAME,SALARY /* Specify columns
ORDER WORKDEPT, -SALARY          /* Order by dept., then salary
BREAK WORKDEPT                   /* Organize by dept.
SUM SALARY                       /* Summarize salaries by dept.
```
### Data Mining and Databases

- **Output from Example 15:**

<table>
<thead>
<tr>
<th>WORK</th>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>000100</td>
<td>HAAS</td>
<td>196947.05</td>
</tr>
<tr>
<td></td>
<td>000110</td>
<td>LUCCHESI</td>
<td>173605.49</td>
</tr>
<tr>
<td></td>
<td>000120</td>
<td>O'CONNELL</td>
<td>109203.32</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>A00</td>
<td></td>
<td></td>
<td>479755.86</td>
</tr>
<tr>
<td>C01</td>
<td>000030</td>
<td>KWAN</td>
<td>38250.00</td>
</tr>
<tr>
<td></td>
<td>000140</td>
<td>NICHOLLS</td>
<td>28420.00</td>
</tr>
<tr>
<td></td>
<td>000130</td>
<td>QUINTANA</td>
<td>23800.00</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>C01</td>
<td></td>
<td></td>
<td>90470.00</td>
</tr>
</tbody>
</table>
Example 16 increases employee salaries by 5 percent in department A00.

Example 16
CONNECT SSID(DB9G) PLAN(DATAM71C) /* Connect to DB

***Define SQL statement to update employee records
EXECUTE SQL(UPDATE EMPLOYEE SET SALARY=SALARY+SALARY*0.05 WHERE WORKDEPT='A00')
Example 17 extracts selected columns to a sequential dataset.

**Example 17**

**EXTRACT**

*** Connect to DB
CONNECT SSID(DB9G) PLAN(DATAM71C)

*** Define the SQL statement
INPUT=SQL (SELECT * FROM EMPLOYEE)

*** Define the output SAM dataset
OUTPUT=DISK  FILENAME=SQLEMP

*** Select the columns to extract
SELECT WORKDEPT,EMPNO,LASTNAME,SALAR
Example 18 inserts a new employee into a VSAM employee dataset.

Example 18

```sql
INSERT INTO EMPFILE (EMPNO, LASTNAME, FIRSTNME, DEPT, HIREDATE, SALARY)
VALUES('123610', 'SMITH', 'JOHN', 'SALES', '2012-05-17', 50000)
```
Example 19 updates employee bonuses.

Example 19

INPUT=VSAM  FILENAME=EMPFILE  /* Update input file
INCLUDE EMPREC  /* Record field defs

UPDATE
IF DEPT = 'SALES'
   BONUS = SALARY * 0.10  /* Change bonus value
ELSE
   BONUS = SALARY * 0.20  /* Use different value
/* for non-sales staff
ENDIF
Example 20 deletes all employees from department A00 only.

Example 20

INPUT=VSAM  FILENAME=EMPFILE /* Update input file
INCLUDE EMPREC        /* Record field defs

DELETE
ONLY DEPT = 'A00'
Highlights

• Data mining extracts raw data to produce useful information.

• An effective data-mining solution increases the efficiency of IT staff and end users by
  ✓ Filtering excess data from files or reports
  ✓ Building subsets of files for testing
  ✓ Empowering people to take advantage of centralized data—with minimal help from IT.