

CompAct

● Inside

Chairperson's Corner _____ 3
by David Snell

Surprising Results from Curriculum
Survey _____ 4
by Randall Kaye

Computer Science Sessions in Anaheim &
San Antonio _____ 5
by Phil Gold

Standards: XTbML Tables Beyond
Mortality _____ 7
by Dean Slyter

P versus NP, or How Long Will This
Take? _____ 10
by Carol Marler

Using Excel to Prototype Relational
Database Applications _____ 12
by Dave Snell

```
<?xml version="1.0" encoding="UTF-8" ?>
- <XTbML id="Table1" Version="XTbML2.9.91"
  xmlns="http://ACORD.org/Standards/Life/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ACORD.org/Standards/Life/2
  XtbML2.9.91.xsd">
- <ContentClassification>
  <ContentType tc="45">Cash Value</ContentType>
  <TableName>Tabular Cash Value</TableName>
</ContentClassification>
- <Table>
  - <MetaData>
    <ScalingFactor>0</ScalingFactor>
    <DataType tc="5">Currency</DataType>
    <CurrencyType code="USD">US Dollar</CurrencyType>
    <TableDescription>Issue Age
  Duration</TableDescription>
    - <AxisDef id="A_1">
      <ScaleType tc="3">Age</ScaleType>
      <AxisName>Issue Age</AxisName>
      <MinScaleValue>35</MinScaleValue>
      <MaxScaleValue>45</MaxScaleValue>
      <Increment>1</Increment>
    </AxisDef>
    - <AxisDef id="A_2">
      <ScaleType tc="4">Services</ScaleType>
      - <!--
        also known as Duration
      -->
      <AxisName>Duration</AxisName>
      <MinScaleValue>1</MinScaleValue>
      <MaxScaleValue>20</MaxScaleValue>
      <Increment>1</Increment>
    </AxisDef>
  </MetaData>
  - <Values>
    - <Axis AxisDefID="A_1" T="35">
      - <Axis AxisDefID="A_2" T="1">
        <V>0</V>
      </Axis>
    - <Axis AxisDefID="A_2" T="2">
      <V>0.39</V>
    </Axis>
    - <Axis AxisDefID="A_2" T="3">
      <V>14.81</V>
    </Axis>
    - <Axis AxisDefID="A_2" T="4">
      <V>29.81</V>
    </Axis>
    <Axis AxisDefID="A_2" T="6">
      <V>
    </V>
  </Values>
</Table>
```



SOCIETY OF ACTUARIES

CompAct Electronic Newsletter • Issue No. 17 • May 2004 • Published in Schaumburg, IL by the Society of Actuaries

Computer Science Section Newsletter
Issue Number 17 • May 2004

Published quarterly by the Computer Science
Section of the Society of Actuaries

475 N. Martingale Road, Suite 600
Schaumburg, IL 60173
phone: 847.706.3500
fax: 847.706.3599

World Wide Web: www.soa.org

This newsletter is free to section members.

Dave Snell
Computer Science Section Editor
RGA Reinsurance Company, Inc.
1370 Timberlake Manor Parkway
Chesterfield, MO 63017-6309
phone: 636.736.7345
fax: 636.736.7745
dsnell@rgare.com

Computer Science Section
Section Council
David L. Snell, Chairperson
Brian S. Septon, Vice-Chairperson
Hal S. Tepfer, Treasurer/Secretary
William R. Aguayo, Council Member/
Web Liaison
Charles S. Fuhrer, Council Member
Philip Gold, Council Member
Kok Bin Liew, Council Member
Frank G. Reynolds, Council Member
Michael K. Rigby, Council Member
Charles S. Linn, Immediate Past Chairperson

Lois Chinnock, Sections Manager
lchinnock@soa.org

Clay Baznik, Publications Director
cbaznik@soa.org

Erica Barraca, Graphic Designer
ebarraca@soa.org

Facts and opinions contained in these pages
are the responsibility of the persons who
express them and should not be attributed to
the Society of Actuaries, its committees,
the Computer Science Section or the
employers of the authors. Errors in fact,
if brought to our attention, will be promptly
corrected.

Copyright © 2004 Society of Actuaries.
All rights reserved.
Printed in the United States of America.

Articles Needed for the CompAct Electronic Newsletter

Your help and participation is needed and welcomed. All articles will include a byline to give you full credit for your effort. CompAct is pleased to publish articles in a second language if a translation is provided by the author. For those of you interested in working on the CompAct, several associate editors are needed to handle various specialty areas such as meetings, seminars, symposia, continuing education meetings, new research and studies by SOA committees and so on. If you would like to submit an article or be an associate editor, please call Dave Snell, editor, at 636.736.7345.

The CompAct is published as follows:

Publication Date	Submission Deadline
September 15	July 15
March 15	January 15

Preferred Format

In order to efficiently handle articles, please use the following format when submitting material:

Please e-mail your articles as attachments in either MS Word (.doc) or Simple Text (.txt) files. We are able to convert most PC-compatible software packages. Headlines are typed upper and lower case. Please use a 10 point Times New Roman font for the body text. Carriage returns are put in only at the end of paragraphs. The right-hand margin is not justified.

If you must submit articles in another manner, please call Erica Barraca, 847.706.3549 at the Society of Actuaries for assistance.

Please send electronic copies of the articles to:

Dave Snell
Computer Science Section Editor
1370 Timberlake Manor Parkway
Chesterfield, MO 63017-6309
phone: 636.736.7345
fax: 636.736.7745
dsnell@rgare.com

Thank you for your help.

Chairperson's Corner

by David Snell

Welcome to this issue of CompAct. Personal thanks are due to my immediate predecessor, Charlie Linn, who restarted the newsletter last fall, after a long hiatus. We plan to make this a more frequent way of communicating with you. Of course, the communication process should occur in both directions, so let us know what you think about the articles, and even better, contribute some articles for the next issue.

A few decades ago, when I was a fledgling actuarial student, I read about an SOA study that asked people on the street to explain the term "Actuary." Some of us laughed at some of the answers. I thought the most creative response was that an Actuary was "a place where you bury dead actors."

Back then, of course, the insurance industry knew of actuaries as the ultimate gurus of financial calculations and as natural leaders of their companies. Many, if not most, insurance companies were managed by an actuary. I remember hearing then that if you liked mathematics, and you liked money (i.e., having some), this was the profession of choice.

Last year the SOA conducted another study, this time asking financial services professionals, and the results were markedly different. We are still viewed as very good with mathematics, but now we have a lot of competition on that front. Communication skills and business savvy are also important and in those areas, the CFA and MBA are considered stronger than the FSA. The role of actuaries as company leaders has diminished.

One thought is that the actuarial academic training curricula may be a bit out of touch with current needs. We decided to investigate this from a computer science perspective, and Randall Kaye summarizes the thought-provoking survey results for you in this issue of CompAct.

Also in this issue, we have an article by Dean Slyter about XTbML, a standard we, along with ACORD (an industry standards organization), are promoting to facilitate the sharing of tabular data between different programs, operating systems and even hardware platforms. Cindy Jeness and Jacques Rioux introduced us to XTbML in the last issue of CompAct, and Dean's article continues with an example beyond just mortality tables.

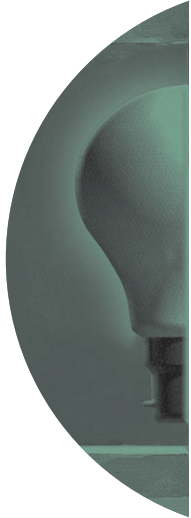
We even have an article about a mathematics contest with a \$1,000,000 prize (mathematics and money, revisited). Carol Marler describes this for us.

You can read about the exciting Computer Science Section sessions planned for this year's Spring and Annual SOA Meetings, and about how to use Excel, a tool as indispensable to actuaries now as the calculator was decades ago, in new ways, to prototype relational databases.

We think we have an assortment of articles you might find interesting. I assure you, we have buried no dead actors in this issue. 🖥️



Dave Snell is chairperson of the Computer Science Section. He is the technology evangelist for RGA Reinsurance Company, and has a lot of fun designing customized solutions for RGA's Asia-Pacific operations. His current hobby is studying Mandarin Chinese. Dave can be reached at dsnell@rgare.com



Surprising Results from Curriculum Survey

by Randy Kaye

In performing their work, Computer Science Section members utilize a wide variety of technical skills. Many useful “computer science”-related subjects are not covered by the SOA exams, though they may be considered. The Computer Science Section recommends that colleges and universities develop curricula for actuarial science programs that reflect the results of this survey and which enable college and university students interested in pursuing an actuarial career to select appropriate courses.

The Computer Science Section membership was surveyed to help us understand which topics are important for actuarial staff to have a working knowledge of, at either the “early career” or “mature career” stage. Over 250 members completed the survey.

The results below are based on a scoring system from 0 = Unimportant, to, 5 = Must Have, then converted to a percentage. A score of 3.5 or more is significant, since it would imply a level of importance somewhere between:


1. 70 percent of the respondents felt that it was 5 = Must Have, and,
2. 100 percent of the respondents felt that it was 3.5 = Important.

A score of 3.5 or more was achieved in the following areas:

While the importance of system-related topics such as spreadsheets, programming languages, database concepts, documentation and project management were expected, the surprising result from this survey is that the respondents felt so strongly that numerical algorithms (especially basic calculus, interpolation, extrapolation, estimation and errors) are still considered a Must Have in importance.

Yet the traditional actuarial skills of calculus, numerical analysis and graduation methods have been removed from the E&E syllabus over the last decade or, if not removed, mentioned as “prerequisites”. This survey shows that the membership still judges these skills as extremely important, and the Computer Science Section council encourages the E&E Committee to consider reinstating numerical analysis and graduation on both the E&E syllabus and examinations.

Colleges and universities are also encouraged to review these results and interpret them as a “call to arms”, or at least, reinstatement, for some of the more traditional actuarial skills.

Complete survey results can be found in the Computer Science Section of the SOA Web site at www.soa.org/ccm/content/areas-of-practice/special-interest-sections/computer-science/compact-newsletter/ 



Randall A. Kaye ASA, FLMI/M is a principal with arc360, a consulting firm providing systems for the life insurance industry. While working with illustration and admin systems, he also helps companies with their universal life product features and policyholder taxation interpretations. He can be reached at RAKaye@compuserve.com.

Score	Early Career
4.7	Mathematical applications, especially spreadsheets (including macros).
3.5	Programming languages, especially procedural languages (C, Fortran, Basic).
3.5	Statistical analysis, especially regression, distributions, standard deviation and Monte Carlo methods.
3.8	Numerical algorithms, especially basic calculus, interpolation, extrapolation, estimation and errors.
3.5	Database Concepts, such as hierarchical & relational, queries, unions and intersections.
3.9	Documentation, such as program comments, system documentation and user documentation.

Score	Mature Career
4.0	Mathematical applications, especially spreadsheets (including macros).
3.8	Project management & planning, such as steering committee, project sponsor, critical path, dependencies and Gantt charts.
3.6	Financial aspects of information systems, such as total cost of ownership, business risk and return on investment.

Computer Science Sessions in Anaheim & San Antonio

by Phil Gold

Now's your chance to sign up for some special sessions at the upcoming spring meetings.

We are sponsoring two sessions in Anaheim in May:

Hot Technologies

Wednesday May 19, 10:30 a.m.

Moderator: Charles S. Linn.

Panel: Neil Raden, Jit Sinha, Jerry Goedicke

This session covers a variety of hot technologies, including:

- Wireless business solutions
- Data management for analytical applications
- Grid computing

Attendee benefits include:

- All the new buzzwords
- Ideas for revolutionizing your own work environment

Nontraditional Computer Science Roles for Actuaries

Thursday, May 20, 8:30 a.m.

Moderator: Brian Reid.

Panel: Lyle S. Semchyshyn, Nancy Walczak, Emily B. Kraft

This forum highlights:

- The career paths of the panelists
- The experience of other session attendees.

Attendee benefits include ideas for job opportunities in less conventional areas of actuarial employment.

We are co-sponsoring a panel discussion with the Health Section:

Data Warehousing Wednesday, May 19, at 4:00 p.m.

• continued on page 6 •



Phil Gold is a partner in GGY Inc., the developers of the AXIS Actuarial System. Phil holds three passports, writes on music and stereo, and has at one time or another been a shepherd, a teacher, a tank gunner, a programmer, a VP of underwriting, an actuary, and an object of derision. Phil lives in Toronto and is deteriorating rapidly. He can be reached at pg@ggy.com.

We are sponsoring two sessions in San Antonio in June:

Actuaries versus IT – The Perpetual Dilemma

Monday, June 14, 10:30 a.m.

Moderator: Michael Hartfield.

Panel: Frank Reynolds, Susie Lee, Emil B. Kraft

Actuaries and IT people rarely see eye-to-eye. Neither understands the other's perspective, with frustrating results for both sides. Yet, with some tolerance and understanding, both can profit from the other's approach.

Actuaries expect IT to:

- Adjust priorities to quickly implement important changes
- Help fix program bugs and technical difficulties
- Design programs for easy modification with little verification
- Provide error-free systems that are already tested

IT expects actuaries to:

- Write specs from which include all reasonably foreseeable modifications
- Write specs IT can program
- Rigorously test all system calculations

Business Intelligence for Actuaries

Tuesday, June 15, 10:30 a.m.

Moderator: Michael K. Rigby

Lecturer: Michael Levine

This session covers:

- Context and general background on business intelligence (BI) and online analytical processing (OLAP)
- How to build a successful application
- How to integrate with other applications

Attendee benefits include:

- A survey of products and approaches
- Examples of actuarial/financial applications

We are also co-sponsoring a workshop with the Financial Reporting Section:

Applications of Stochastic Modeling in Financial Reporting

Tuesday, June 15, 2:00 p.m. 

Standards: XTbML Tables Beyond Mortality

By Dean Slyter

If you have been a member of the Computer Science Section for a while you know about the Table Manager tool. You may have used it to access a storehouse of mortality, morbidity and lapse tables. You probably know this section worked with ACORD, the life insurance standards organization, to generalize the table manager format into XML (eXtensible Markup Language). This new standard is called XTbML (sometimes pronounced as “x-table”).

Moving the table format to ACORD XML is sort of like sending our child into the world. The little being we nurtured to handle our mortality/morbidity tables is being asked to carry all kinds of data. Insurance companies are trying to use XTbML to carry premiums, cash values, commission rates, modal factors, dividend rates and interest rates. And these are just the uses I know about.

Beginning with this issue of CompAct, I’m going to highlight business use cases where XTbML can define the tabled data.

My first example is from the life insurance practice area. If you’re willing to e-mail me a table data business case from your practice area, I would be happy to represent it using XTbML in a future CompAct newsletter.

Business Use Case: Tabular Cash Values

This table provides cash values for a 20 pay whole life insurance plan. After paying 20 annual premiums, no additional premiums are paid. The cash values are based on issue age and duration (during the premium paying period) or just attained age (after the premium paying period). Duration 1 is the first year from the policy’s issue date and the cash values in this table are end of duration cash values. If you have a whole insurance contract, you will find that it contains a table of cash values. Below you’ll find versions examples of these cash value tables truncated by issue age or attained age to save space.

Now let’s look at how this table might be represented in XTbML. Keep in mind that XTbML is a self-defined machine-readable

Premium Payment Period Cash Values										
	Duration									
Issue Age	1	2	3	4	5	6	7	8	9	10
	0	0.39	14.81	29.81	45.38	61.56	78.35	95.79	113.92	132.77
	0	0.7	15.63	31.15	47.25	63.97	81.33	99.36	118.11	137.6
	0	0.97	16.43	32.47	49.12	66.39	84.35	103	122.38	142.54
	Duration									
Issue Age	11	12	13	14	15	16	17	18	19	20
	152.38	172.78	194.01	216.12	239.13	263.09	288.02	313.98	340.98	369.11
	157.88	178.98	200.93	223.78	247.56	272.3	298.03	324.8	352.66	381.7
	163.51	185.32	208.01	213.62	256.17	281.69	308.22	335.83	364.57	394.56
Paid-Up Period Cash Values										
Attained Age	55	56	57	58	59	60	61	62	63	64
	381.7	394.56	407.75	421.26	435.09	449.23	463.64	478.26	493.03	507.91

• continued on page 8 •


Dean Slyter is actuary and account manager, LIDS Consulting in Woodridge, Ill. He can be reached at slytde@lidp.com.

format for transferring and storing tabled data. If a machine can read this format, so can you. I'll give you a few XML reading hints to help you read it:

1. XML provides opening (e.g.: <Table>) and closing (e.g.: </Table>) for each data element. A lot of child data can be represented between the opening and closing of a data element.
2. The content classification defines the type of values contained in the table. It uses a typecode (tc) lookup table where typecode '45' is cash value.
3. The cash values are represented in two stacked tables, first the premium paying period table, then the paid-up period table.

4. A low duration of 21 has been defined for the paid-up period table to clarify the first duration where it should be used. The data element <Continuous> indicates that the duration 21 cash value is used until a higher duration cash value is given.

Cash value tables are used for product pricing, financial reporting, sales and illustration and policy administration. If each such system could import and export data in the XTbML format, data transfer between any two systems would be possible.

Please send your tabled data business cases and XTbML questions to me at slytde@lidp.com. 

```
<?xml version="1.0" encoding="UTF-8" ?>
- <XTbML id="Table1" Version="XTbML2.9.91"
  xmlns="http://ACORD.org/Standards/Life/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ACORD.org/Standards/Life/2
  XTbML2.9.91.xsd">
- <ContentClassification>
  <ContentType tc="45">Cash Value</ContentType>
  <TableName>Tabular Cash Value</TableName>
</ContentClassification>
- <Table>
  - <MetaData>
    <ScalingFactor>0</ScalingFactor>
    <DataType tc="5">Currency</DataType>
    <CurrencyTypeCode tc="840">USD</CurrencyTypeCode>
    <TableDescription>Issue Age -
      Duration</TableDescription>
  - <AxisDef id="A_1">
    <ScaleType tc="3">Age</ScaleType>
    <AxisName>Issue Age</AxisName>
    <MinScaleValue>35</MinScaleValue>
    <MaxScaleValue>45</MaxScaleValue>
    <Increment>1</Increment>
  </AxisDef>
  - <AxisDef id="A_2">
    <ScaleType tc="4">Service</ScaleType>
  - <!--
    also known as Duration
    -->
    <AxisName>Duration</AxisName>
    <MinScaleValue>1</MinScaleValue>
    <MaxScaleValue>20</MaxScaleValue>
    <Increment>1</Increment>
  </AxisDef>
  </MetaData>
- <Values>
  - <Axis AxisDefID="A_1" T="35">
  - <Axis AxisDefID="A_2" T="1">
```



```

    <V>0</V>
  </Axis>
- <Axis AxisDefID="A_2" T="2">
  <V>0.39</V>
</Axis>
- <Axis AxisDefID="A_2" T="3">
  <V>14.81</V>
</Axis>
- <Axis AxisDefID="A_2" T="4">
  <V>29.81</V>
</Axis>
- <Axis AxisDefID="A_2" T="5">
  <V>45.38</V>
</Axis>
- <Axis AxisDefID="A_2" T="6">
  <V>61.56</V>
</Axis>
- <Axis AxisDefID="A_2" T="7">
  <V>78.35</V>
</Axis>
- <!--
  ....Continued....
-->
</Axis>
- <Axis AxisDefID="A_1" T="36">
- <Axis AxisDefID="A_2" T="1">
  <V>0</V>
</Axis>
- <Axis AxisDefID="A_2" T="2">
  <V>0.7</V>
</Axis>
- <Axis AxisDefID="A_2" T="3">
  <V>15.63</V>
</Axis>
- <Axis AxisDefID="A_2" T="4">
  <V>31.15</V>
</Axis>
- <Axis AxisDefID="A_2" T="5">
  <V>47.25</V>
</Axis>
- <Axis AxisDefID="A_2" T="6">
  <V>63.97</V>
</Axis>
- <Axis AxisDefID="A_2" T="7">
  <V>81.33</V>
</Axis>
<!--
  ....Continued....
-->
</Axis>
<!--
  ....Continued....
-->
</Values>
</Table>
- <Table>
  - <MetaData>
    <ScalingFactor>0</ScalingFactor>
    <DataType tc="5">Currency</DataType>
    <CurrencyTypeCode
tc="840">USD</CurrencyTypeCode>
    <TableDescription>Attained
Age</TableDescription>
    - <AxisDef id="A_3">
      <ScaleType tc="3">Age</ScaleType>

```

```

    <AxisName>Attained Age</AxisName>
    <MinScaleValue>55</MinScaleValue>
  <MaxScaleValue>99</MaxScaleValue>
  <Increment>1</Increment>
</AxisDef>
- <AxisDef id="A_4">
  <ScaleType tc="4">Service</ScaleType>
  - <!--
    also known as Duration
  -->
  <AxisName>Duration</AxisName>
  <MinScaleValue>21</MinScaleValue>
  <Increment>1</Increment>
  <Continuous>1</Continuous>
</AxisDef>
</MetaData>
- <Values>
  - <Axis AxisDefID="A_3" T="55">
  - <Axis AxisDefID="A_4" T="21">
    <V>381.70</V>
  </Axis>
</Axis>
- <Axis AxisDefID="A_3" T="56">
- <Axis AxisDefID="A_4" T="21">
  <V>394.56</V>
</Axis>
</Axis>
- <Axis AxisDefID="A_3" T="57">
- <Axis AxisDefID="A_4" T="21">
  <V>407.75</V>
</Axis>
</Axis>
- <Axis AxisDefID="A_3" T="58">
- <Axis AxisDefID="A_4" T="21">
  <V>421.26</V>
</Axis>
</Axis>
<!--
  ....Continued....
-->
</Values>
</Table>
</XTbML>

```



P versus NP, or How long will this take?

by Carol Marler

For a new and creative solution to this question, you could win a prize of \$1,000,000.

On May 24, 2000, the Clay Mathematics Institute offered prizes of \$1 million each for seven unsolved problems in mathematics. Landon Clay (whose major at Harvard was not in mathematics, but in English) wanted to get more media attention for the field of mathematics. His personal fortune was used to found the Clay Mathematics Institute (www.claymath.org).

The idea for these "millennium problems" came from an address given by mathematician David Hilbert in 1900, in which he listed the most significant mathematics problems at that time. One problem on his list was Fermat's Last Theorem, which was solved recently.

Although the solution to Fermat's Last Theorem made extensive use of computers, the theorem itself does not refer in any way to computers. Very few problems in theoretical math, even today, fall under the heading of computer science. Most computer science questions are practical, not theoretical. But looking back in history, some theory was developed even before any modern computers were built.

The theoretical basis of computer science was developed by a mathematician named A.M. Turing. His model of a computer was a black box of sorts with n different internal states, and an input "tape" consisting of zeros and ones. The operation of the black box was defined by a transition matrix in which the current state and the current value on the input tape determined the next state.

The amazing thing about this model is that any real-world computer program could be mimicked with a suitable set of states and a transition matrix. The theoreticians can then analyze whole families of programs. The P versus NP millennium problem considers a grouping of programs based on a formula for how long (at most) it takes to process an input string of length n .

Programs whose time formula is a polynomial in n are denoted as P, for polynomial time. For example, consider a program that sorts N values in ascending order. A simple way to do this is to loop through the list, comparing each pair of numbers ($N-1$ comparisons), and switching them if the second number is smaller than the first. This loop would then be repeated $N-1$ times, until all the numbers had been sifted through to their proper position in the sorted list. If each compare and switching took k units of time, the formula is $k*(N-1)*(N-1)$ plus some small amount of time to set things up, etc. The time required is clearly a second degree polynomial. Improvements are possible, such as noting that each iteration of the outer loop reduces by one the number of comparisons required, since the last $i+1$ entries are now determined.

Incidentally, changing a list of n zeros and ones into N values is also known to be a polynomial process. But some processes simply cannot be done in polynomial time. For example, consider the "traveling salesman" problem. The salesman (or woman, but let's use male pronouns for simplicity) has a list of N destinations that he wants to visit. There is an $N*N$ matrix of the distances from each destination to any other. The salesman wants to minimize

Carol Marler is a former editor of the Speculative Fiction contest. She is currently unaffiliated. She can be reached at 704-948-0545.

time on the road, and this can be programmed by listing all possible routes, calculating the length of each route, and then taking the shortest. If there are only five or so cities, this can even be accomplished by hand. But it is apparent that as N increases, the program will require $N!$ distance calculations. And $N!$ for large N grows exponentially. You don't want to kick off a calculation for 50 destinations if you expect the answer in a reasonable length of time.

The next step in analyzing this kind of problem is to generalize the computer model. One very useful variation, and one that is rather actuarial in nature, is to replace the deterministic computer with a non-deterministic one. Instead of defining the resulting state, the transition matrix produces a probability distribution of states. Nobody has yet built a non-deterministic computer, but if we had one, the traveling salesman problem could be solved on it in polynomial time.

In fact, the traveling salesman problem is just one of a whole family of problems for which the individual calculations are simple enough, but the number of possibilities to be tested grow exponentially. These problems are denoted as NP-complete.

In 1971, Stephen Cook proved that if any NP-complete problem could be solved on a deterministic computer in polynomial time, then all of them could be similarly solved. The millennium problem then asks whether these NP-complete problems are a subset of P, meaning that they could be solved deterministically with the right kind of algorithm, or are they inherently not solvable in that way?

In the real world, the Cook analysis leaves just two possibilities. First, if a problem is known to be NP-complete, and a software developer doesn't want to waste time looking for a solution that has already been

sought so many times without success, the problem can be discarded and the software can be built for some other problem instead. A great many problems of importance to business and industry have been dropped for this reason. Some other problems of less practical value also fall into this category—you may want to Google “NP-complete” just to see how many times the minesweeper game pops up. Second, if some genius inventor in a garage can find the answer to any particular NP-complete problem, he will soon be richer than Bill Gates (even without the millennium prize).

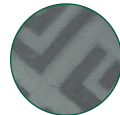
However, if you think it is important to have the ability to do secure online transactions, you had better hope that the genius is not successful, since the most commonly used security method is based on a known NP-complete process—factoring a very large number into its prime factors. If NP-complete turns out to be a subset of P, the methodology will no longer be secure.

If, in fact, NP-complete is not a subset of P, the millennium prize will go to the person who can prove that fact. A great many attempts have been made to do this, and none have yet been successful. It could be another century before this problem gets solved. Some even theorize that it is true but cannot be proven.

One intermediate possibility also exists. Even if NP-complete is a subset of P, the power of the required polynomial may prove to be so large that no practical solution is available. A 50-degree polynomial is not much of an improvement on an exponential formula if the traveling salesman has no more than 50 destinations.

Do you and your PC have the insight and CPU cycles to attack the problem? If so, see <http://www.claymath.org/millennium/> to get entry information. 🏠

Nobody has yet built a non-deterministic computer, but if we had one, the traveling salesman problem could be solved on it in polynomial time.



Using Excel to Prototype Relational Database Applications

by Dave Snell

Excel is often dismissed by information technology (IT) professionals because it is viewed by them as a toy—suitable only for single-user applications. Actuaries know the value of spreadsheets in rate calculations and what-if scenario testing; but most actuaries have never realized that their convenient little spreadsheet tool is also capable of lifting some rather heavy weights in the database environment.

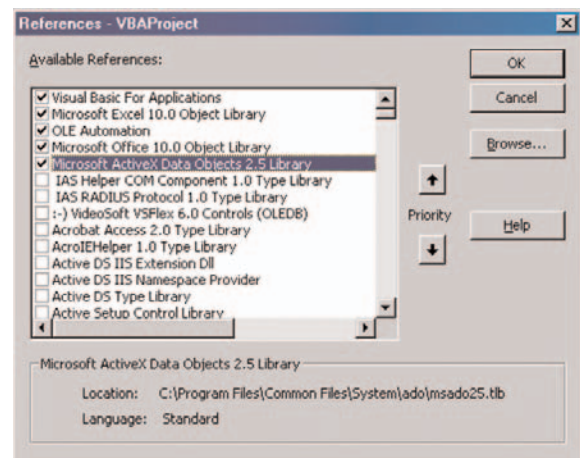
In my opinion, Microsoft Excel is a highly functional programming language that happens to come with an awesome built-in grid control. It is the one of the most cost-effective tools I know of for making high-quality small applications that can be used on almost any corporate desktop. It can also be great for prototyping larger database applications.

I propose to show in this article that you can develop and test relational database projects from your spreadsheets. You can do it rapidly, and with less angst or cost than you might feel by going through the learning curve of the big database tools, like Microsoft's SQL Server or Oracle. I'll assume you have some familiarity with Structured Query Language (SQL), which we won't cover here. I also assume you've some exposure to Visual Basic for Applications (VBA) and that you may have even heard of Active Data Objects (ADO); but this can be minimal. Once you follow the instructions for entering the code below, you can embellish it at your leisure. This article is to help get you started using SQL from within Excel.

Using SQL with Excel sources

Excel has the capability to contain multiple tables and ranges, and it can be used as a source for SQL queries.

An easy way to accomplish this is via ADO. First, you must add a reference to ADO in your project in the VBA mode, which you can reach quickly from the spreadsheet mode via Alt-F11 (hold down the Alt key and press the F11 key). Once in VBA mode, click on Tools, References.



There are several ADO libraries. I recommend using the lowest numbered one that meets your needs so that your workbook will be more transportable. In this case, I chose version 2.5, which probably came with Excel 97.

Next, you are going to need a little VBA help in the background to let you effectively use this conduit to SQL capability. Click Insert, then Module to get a working area. Then copy and paste the code in from this article.

The first item to copy is a Sub to read from an Excel database via Active Data Objects (ADO).

```

Sub CopyFromXLDatabaseADO( _
    sDBRangeBook As String, _
    sSQLRangeName As String, _
    sDestinationSheet As String, _
    Optional bolClearSheet As Boolean = True, _
    Optional bolShowFieldNames As Boolean = True, _
    Optional sDestRange As String = "A1")
'written by Dave Snell
'This Sub allows you to perform SQL queries against
'tables (named ranges) in an Excel workbook

Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
Dim fld As ADODB.Field
Dim lngOffset As Long
Dim lngStartTime As Long
Dim cmd As New ADODB.Command
Dim s As String
Dim sSQL As String
Dim l As Integer

    lngStartTime = Timer

    Sheets(sDestinationSheet).Activate
    If bolClearSheet Then
        Cells.Select
        Selection.ClearContents
    End If

    'populate with new information from the Access database extract
    With cn
        'PROVIDER STRING follows (change for different source type e.g., Oracle)
        .Provider = "Microsoft.Jet.OLEDB.4.0" 'If you use a 'different
        'version of 'the Jet engine, change the 4.0 to whatever you use.
        .Properties("Extended Properties").Value = "Excel 8.0" 'If you
        'use Excel '2000, this is correct; Excel 2002 is 9.0, etc.
        .Open ActiveWorkbook.Path & "\ " & sDBRangeBook
    End With
    With cmd
        Set .ActiveConnection = cn
        'build a SQL query from the text on the SQL worksheet
        With Sheets("SQL")
            s = " "
            sSQL = ""
            While Len(s) > 0
                s = .Cells(.Range(Range(sSQLRangeName).Address).Row + l, _
                    .Range(Range(sSQLRangeName).Address).Column).Value
                sSQL = sSQL & " " & s
                l = l + 1
            Wend
        End With
        .CommandType = adCmdText
        .CommandText = sSQL
        Set rs = .Execute
    End If
End With

If Not rs.EOF Then
    With ActiveSheet.Range(sDestRange)
        If bolShowFieldNames Then
            'write field names as column headers
            For Each fld In rs.Fields
                .Offset(0, lngOffset).Value = fld.Name
                lngOffset = lngOffset + 1
            Next fld
        End If
    End With
    'copy the rest of the records one row below the headers

```

• continued on page 14 •

```

        ActiveSheet.Range(sDestRange).Offset(1, 0).CopyFromRecordset rs
        ActiveSheet.UsedRange.EntireColumn.AutoFit
    Else
        MsgBox "Error: No records read", vbCritical
    End If

    Range(sDestRange).Select
    MsgBox "Done!" & vbCrLf & "Elapsed time = " & Timer - lngStartTime & " seconds"

ExitRoutine:
    rs.Close
    cn.Close
    Set cn = Nothing
    Set rs = Nothing
    Set cmd = Nothing

End Sub 'CopyFromXLDatabaseADO

```

This Sub should be usable for many situations. Note however that for readability of the sample code, I removed a lot of error checking. You will want to add error checking in various portions of the Sub. Since this approach uses ADO for the data access, you can also use it to query Access databases, Oracle databases, etc. with a mere change of the Provider string. In practice, I have written a longer

version of this macro that determines the source type from parameters I set and changes the provider string accordingly (a future article). Rather than get complicated here though, let's just go ahead and try an application showing how you might make use of the supplied Sub.

Let's create a very simple calling routine:

```

SubTest_CopyFromXLDatabase()
    CopyFromXLDatabaseADOThisWorkbook.Name, "SQL_Example", "Example_Output"
End Sub 'Test_CopyFromXLDatabase

```

This tells the Sub CopyFromXLDatabase to look in the current workbook (we could also refer to any other workbook) for the Range Name "SQL_Example." Then, it reads down the column from cell SQL_Example and builds the SQL query until it encounters a cell that is blank. The final set of results is then copied to the Excel spreadsheet "Example_Output."

All of this looks OK, but obviously nothing is going to happen unless we actually have a range named "SQL_Example" somewhere in the workbook. Let's assume we have a simple workbook with just three worksheets in it. I'll name my three sheets Example_Source, SQL and Example_output.

In Example_Source we'll place our tables. Assume we have two tables of data as shown in Figure 1 on page 15.

Note that I have the Applicants table highlighted. Likewise, I named the Apps table (cells G4 through I25) "Apps." The easy way to accomplish this naming process is to high-

light the desired area with your mouse, then type the desired Range name in the input box just above cell A1. In this screen shot you can see Applicants in this input box.

Next, we look at the SQL sheet, Figure 2 on page 15.

Cell A4 on this sheet is the one I named SQL_Example. I also made sure I have another sheet for my output. This one is named Example_Output.

Everything is in place. Now, let's run the macro (Figure 3, page 15).

Clicking on Tools, Macro then Macros (or just pressing the shortcut key Alt-F8), we see a list of the macros available (in this case just the one test macro).

Run the macro Test_CopyFromXLDatabase and voila! You have the SQL query output right away on the Example_Output sheet. (Figure 4, page 16).

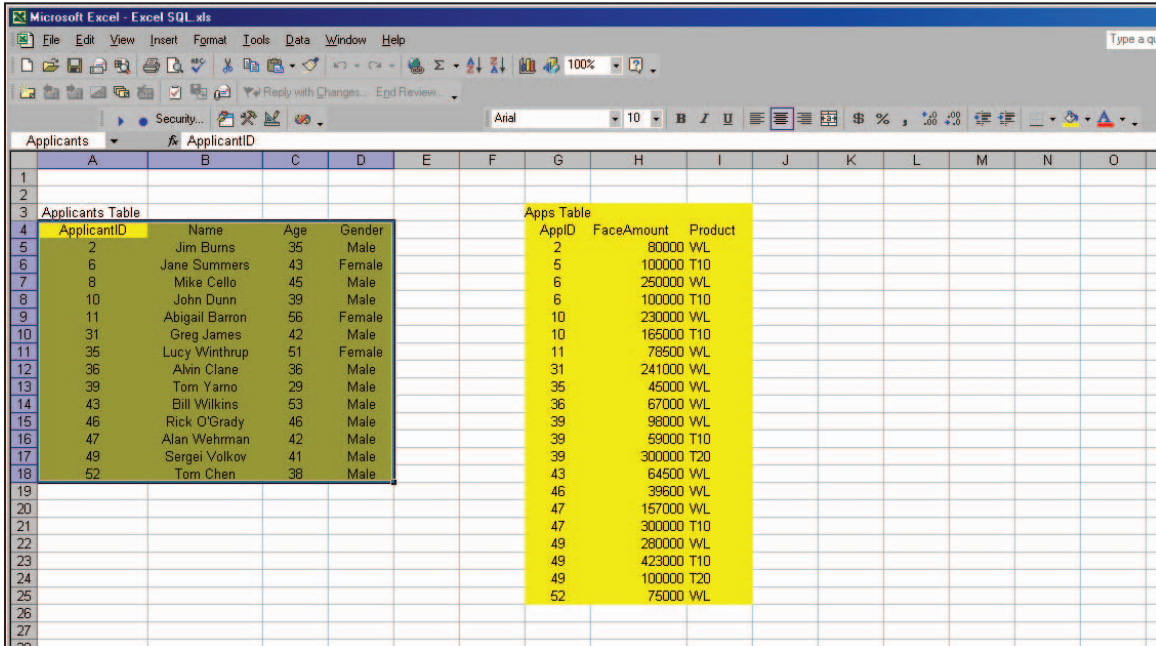


Figure 1

Note the Range Name "SQL_Example" assigned to cell A4 of this sheet.

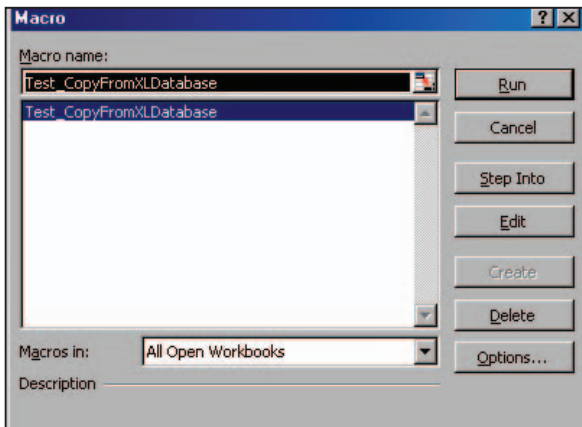


Figure 3

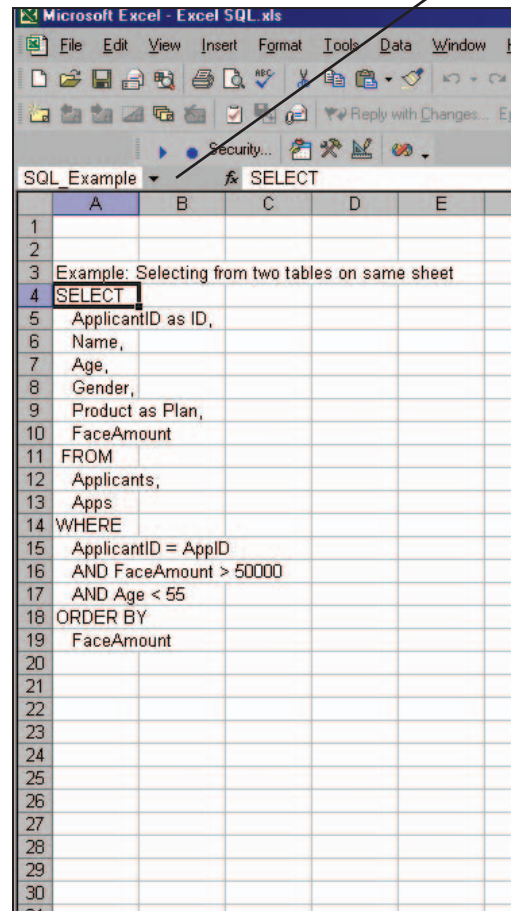


Figure 2

Bigger and Better

Let's face it. This is a very simple example. It's hardly worth writing a VBA Sub to accomplish this task. You could have obtained similar results with Excel's Data, Filter, Advanced Filter feature or with its built-in QueryBuilder (accessed via Data, Import External Data, New Database Query) capability.

I wrote (and use) the VBA approach for a couple of reasons. First, it allows me to get very flexible with my SQL statements—beyond the limits of the built-in Excel tools. In a report I wrote

• continued on page 16 •

ID	Name	Age	Gender	Plan	FaceAmount
39	Tom Yarno	29	Male	T10	59000
43	Bill Wilkins	53	Male	WL	64500
36	Alvin Clane	36	Male	WL	67000
2	Jim Burns	35	Male	WL	80000
39	Tom Yarno	29	Male	WL	98000
6	Jane Summers	43	Female	T10	100000
10	John Dunn	39	Male	T10	165000
10	John Dunn	39	Male	WL	230000
31	Greg James	42	Male	WL	241000
6	Jane Summers	43	Female	WL	250000
39	Tom Yarno	29	Male	T20	300000

Figure 4

for one of our foreign offices in Asia, I use a SQL query more than 100 lines long that involves a 16-table join with dozens of special criteria. Second, I like the fact that once I debug my routine on a small scale against an Excel or Access database, I can just change the provider string in my Sub and use exactly the same SQL and VBA code to scale this up to a huge Oracle database.

In my presentation on relational databases at the SOA Annual Meeting in Orlando last year, I showed a video of how much faster a big chainsaw can cut a big black walnut log than an electric circular saw or a hand-held scroll saw can. In the beginning of the video though, I show that if you use the chainsaw the same

way you use a handsaw, by rubbing it back and forth against the wood, it gives very disappointing results. My point here is that the bigger database tools, like Oracle and SQL Server, have a serious learning curve associated with their efficient usage. The actuary does not have to become a database administrator (DBA) in order to intelligently prototype a large database project. Excel, supplemented with VBA and ADO, provides a very convenient tool for trying out your ideas, and also gives you a scalable solution you can later turn over to your IT associates as a working set of specifications for the big tool implementation.

Happy prototyping! 



SOCIETY OF ACTUARIES

475 N. Martingale Suite 600 • Schaumburg, IL • 60173 • www.soa.org