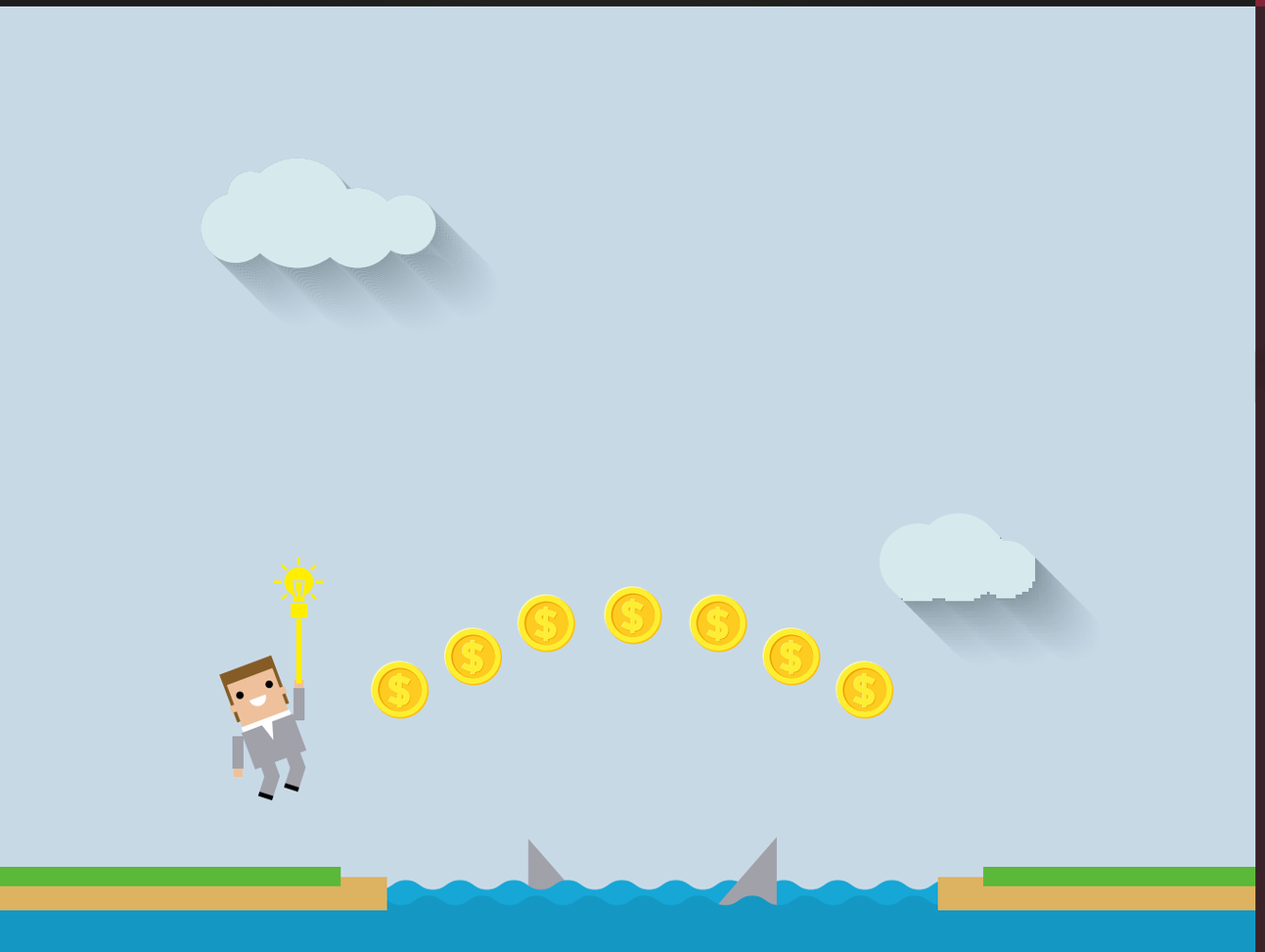
Juni 2019, Edition #15

ORAWORLD

e-Magazine for Oracle Users published by the EOUC

Gamification:
a Game-Changer for Modern Societies?



EOUC
EMEA ORACLE USERGROUP COMMUNITY

A Game-Changer for Modern Societies?

page 6

Code You Should Never See in PL/SQL – Part 3

page 15

App Development. An Update

page 33

Have a comment? Suggestion? Want to submit an article proposal? **Here's how to do it.**

Editorial

Dear Oracle User Group Community Members

In this issue of OraWorld magazine you can read about gamification explained by Dr. Mathias Fuchs, who loves playing Assassin's Creed. I understand him very well because thanks to Assassin's Creed my boys, Patrik and Matias, have been able to guide me in many places: after playing the game, the maps of Venice, Florence, Paris, or even the Castel Sant'Angelo in Rome are in their heads forever. Or you can read how to program games with Java explained by Annette Godtland, who really knows what she is talking about after writing several books on the topic.

I am very pleased to say that also in this issue two Oracle world heroes, whom I really appreciate, Richard Foote and Steven Feuerstein, are sharing their knowledge with our readers. Steven will continue his series of "Code You Should Never See in PL/SQL" with Part 3, and Richard completes his "Importance of Data Clustering" with Part 2. Congratulation to Richard for his renewal of Oracle ACE Directorship! Extremely well deserved!

In the Number of the Month section we will have quite a serious topic this time: layoffs. This has been worrying many people in the Oracle community. But happier topics are a nice interview with Paolo Kreth from SOUG, the Oracle user group in Switzerland, or the list of all the user group events in EMEA.

These are just some examples of the great articles in ORAWORLD issue 15. I hope you enjoy reading it as much as I did! Please remember to submit your content for the upcoming issue online on our website: www.ORAWORLD.org!

Heli Helskyaho
Ambassador, EOUC

Yours,
Heli Helskyaho
Ambassador, EOUC

Submit Your Article!

You have an interesting topic to publish in ORAWORLD Magazine? Then submit your article and be part of it!

Please e-mail us your article via the online form at **www.oraworld.org**.

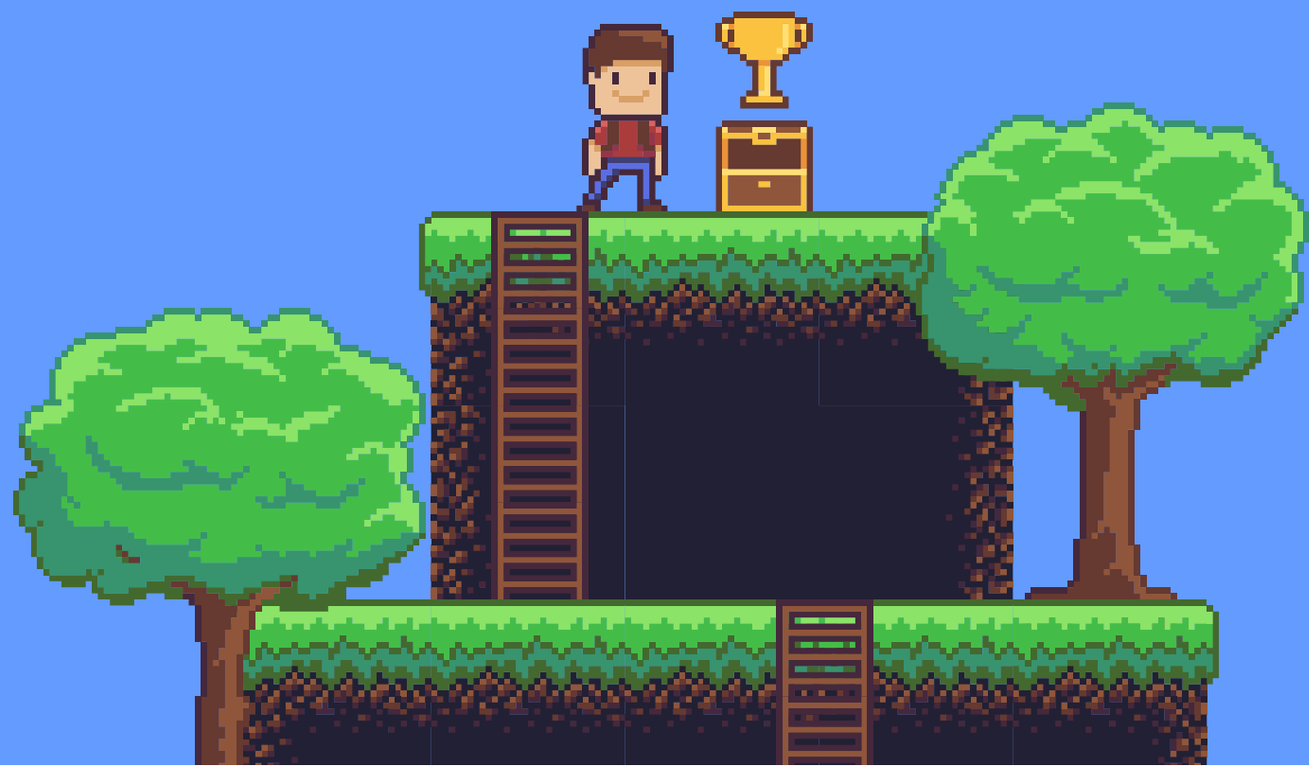
When the Words Write Themselves

www.commitstrip.com

CommitStrip is a daily strip recounting funny anecdotes of life as a coder mixed up with a dash of topical tech news. Find more comics here: ***www.commitstrip.com***

Gamification: a Game-Changer for Modern Societies?

Sanela Lukavica



Dr. Mathias Fuchs is a lecturer and researcher at the Leuphana University of Lüneburg, Germany. With his current research project "The Staging of Computer Simulations", he primarily investigates gamification from a cultural studies, media-theoretical, and political perspective. Sanela Lukavica has asked Dr. Fuchs what gamification actually is, and where the limits to gamification lie.

How does gamification work, Dr. Fuchs?
(Laughs) Well, does it work at all? The thing is that there are famous examples for gamification that have been well received, such as the musically animated stairs in Stockholm or running apps like Nike Plus. However, there are also gamification apps that, possibly, do not work at all. They may even be counter-productive because users reject or ignore them after some time.

How do you evaluate if gamification works?
We can ask for the individual user: does this add value for them? Do they really get smarter, healthier, do they have better grades in school? In many cases, it may seem that this type of self-optimization works. The important question is what it means in this context for a society as a whole, how does it change our understanding of the world or ourselves? In a mere game, it is enough to just feel well. The game works then. When a child swings on a swing, there is no payout. It is only about the fun. Gamification, however, always asks for the profit. How does the customer become a more loyal customer, or stay loyal. Gamification is always designed for an objective.

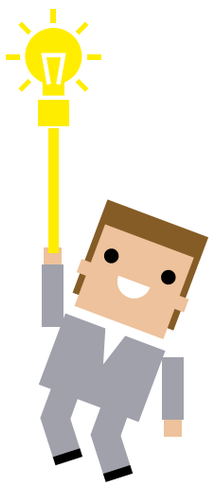
And the balance of the two factors fun and seriousness plays an important role.
Absolutely. When it gets too serious, then there is not enough game in it. When it gets too funny, it will not lead to the desired success.

Is gamification suitable for each work and every professional field? Or is there work that is too monotonous or – on the contrary – too complex for gamification?
For complexity, that is certainly relevant. When we have complex tasks, the actual question is if gamification can work and or it would be better to just focus and try to make it. In that case, the fun comes after the end of the work day. A top class pianist or a top class violinist cannot only handle their instruments in a playful manner. To really master the instrument is hard backbreaking work.

Let's approach the topic "heretically". Some gamification supporters would love to see a completely gamified world. But with gamification following certain patterns that are easily learned, would this not pose the risk that many of us would eventually only develop a kind of guided problem-solving capability?
I am on the side of the heretics. I think it would be terrible if we lived in a world where things are only done by some installed gaming mechanisms. And I think that this two-sidedness of game and serious interaction is very important for us to be reasonable persons. Johan Huizinga, the Dutch anthropologist, phrased that very well in his 1938 book "Homo Ludens": A prescribed game is not a game anymore. This is the whole point. When I tell someone: You must play; play at work, even play at war – then this is not fun anymore.

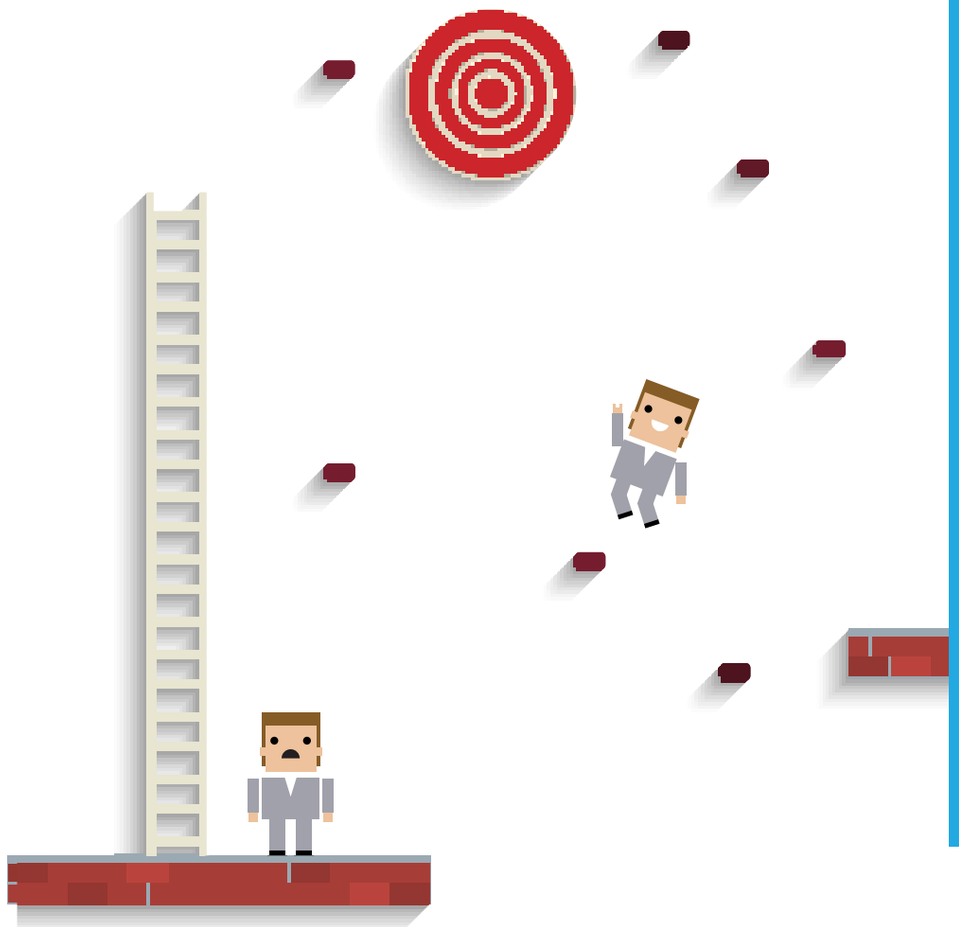
Dr. Fuchs, you have comprehensively researched on gamification in general. Could you outline the meaning of gamification for the average European from a historic perspective?
It actually seems to have changed significantly compared to, for instance, the second half of the 18th century. In Mozart's time, they played like crazy: Billards, card games, composition games, music games – everything was declared a game all of a sudden. And also on a horizontal level, it seems that different cultures accept the gaming element differently. We actually know that Koreans play exceedingly and it seems crazy to many Europeans. In these huge e-sports events: Chinese and Korean gamers are at the very front, while other cultures are not as enthusiastic. E-sports is a professional competition where games like Dota 2 or League of Legends are played against each other in front of an audience of millions. 17 and 18 year old Koreans who simply play incredibly well receive prizes in the range of millions. Today, there are academies in Korea where you can learn how to play these e-sports games. These places are like our universities and we Europeans are still in doubt. We cannot imagine to push young people completely into this gaming world and that is why we are rather careful.

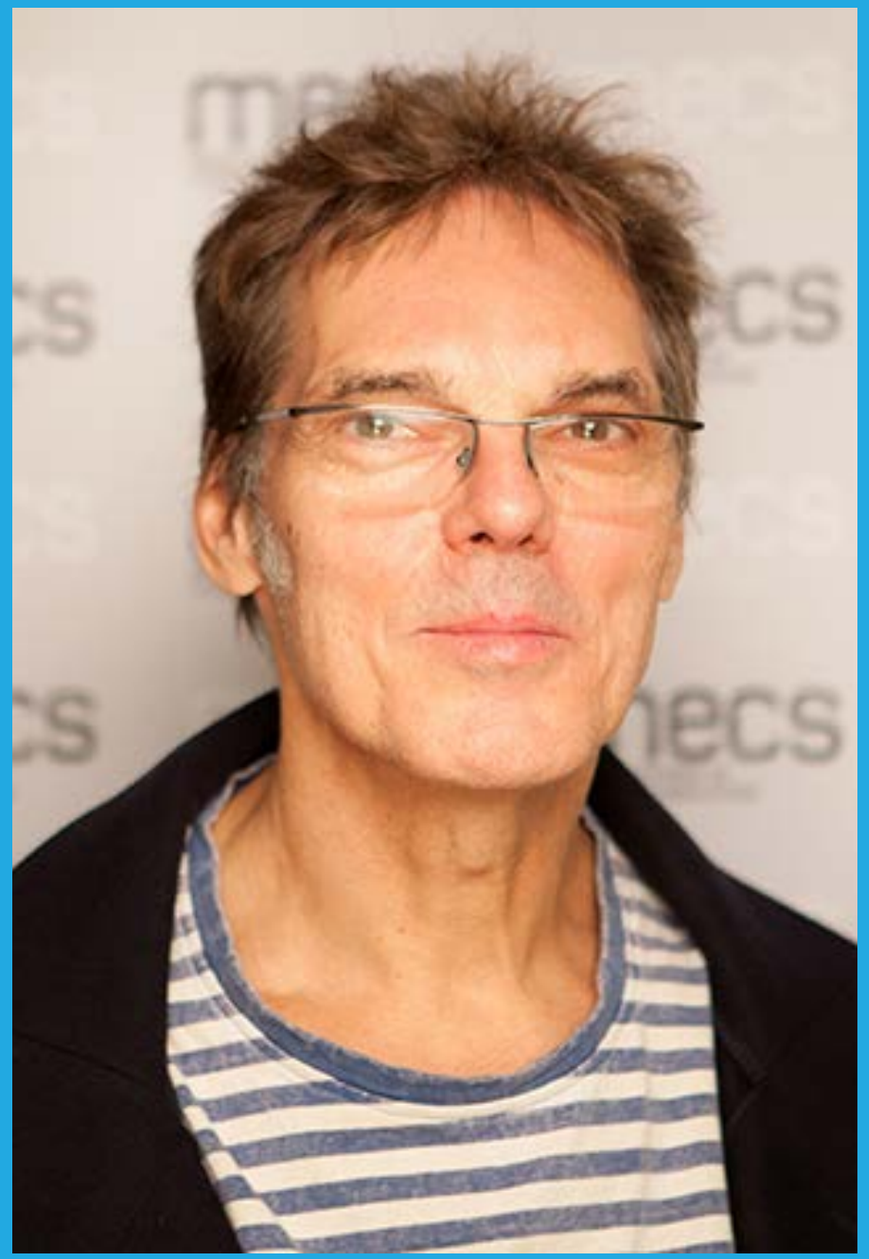
What about countries that are quite similar, let's take the Western European countries for instance. Can gamification in Germany look identical to gamification in England?
That is a good question. I think most gamification developers are not that aware that there are cultural differences. Currently, most gamification applications are designed for US-American or even Californian audiences and then also used for Germans, Italians, Ukrainians, and so on. But, in real life, we are different, and we know that when it comes to humor, for example: There is the English humor that is completely different from the Sicilian humor. Certain things that make the English smile are offending to a Sicilian. And something that is funny for a Sicilian is not that well



received in London. Actually, we should address this in the development of gamification applications. We should clarify the target audience and react to their cultural sensitivities and cultural imprint.

Back to the roots: What is your favorite game?
I really like to play Assassin's Creed because, on one hand, the challenge appeals to me, and, on the other hand, I have the chance to visit all these beautiful cities and can be a virtual tourist in Havana or Boston.





Dr. Mathias Fuchs
Lecturer and researcher at the Leuphana University of Lüneburg, Germany

Annette Godtland

Stop Playing Games — Create Your Own Games Instead

Approximately 2.2 billion people worldwide regularly play video games.[1] The average gamer spends 7.11 hours playing video games each week, with 9.2% of gamers playing more than 20 hours per week.[2] That's a lot of time spent playing games! I enjoy programming and I would like to encourage others to try it too. I wrote a series of books, Do-It-Yourself Java Games, that use a unique "discovery learning" approach to teach computer programming through creating computer games!

What Creating Your Own Computer Games Can Do For You

I believe that creating your own computer games can give you challenges, enjoyment, and satisfaction similar to playing many computer games. But creating your own computer games will give you more to show for your efforts.

Similar to playing games, programming teaches you to think logically and systematically. It teaches you planning skills and problem solving skills. Programming a computer gives you power and control: a program will do exactly what you program it to do. Programming also teaches you to be precise and clear in your instructions: a program will do exactly what you program it to do, whether that is what you intended or not.

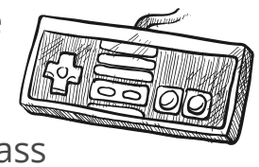
Discovery Learning as a Way to Understand Java Programming Techniques Better

Discovery learning, defined by Jerome Bruner in the 1960s, is "a method of inquiry-based instruction. Discovery learning is based on a belief that it is best for learners to discover facts and relationships for themselves. The learner draws on his or her own past experience and existing knowledge to discover facts and relationships and new truths to be learned. Students interact with the world by exploring and manipulating objects, wrestling with questions and controversies, or performing experiments. As a result, students may be more likely to remember concepts and knowledge discovered on their own."

I feel a person will come to understand Java programming techniques more by doing Java programming than by reading about those techniques. I start the reader writing code in the very first lesson of each book. I use fill-in-the-blank code listings with instructions and hints to guide the reader to write complete programs himself. The answers are at the back of the book. The reader will learn more as he must think about each line of code needed to do the task at hand than he would if he just copied code from a book. By creating a variety of games in each book, the reader will discover what is common across all the games, and what is unique to a particular game. Through it all, he will discover how, when, and why Java programs are written the way they are.

Why Java Makes for a Great First Language

I have to admit that I initially put off writing a book teaching Java as a first programming language. The easiest way to learn the fundamentals of Java programming is by writing programs that print to the console. But no one wants games that run in a command prompt window, nor is a beginning programmer ready to deal with the complexities of writing even a simple "Hello World" Java program with a window interface.



So I decided to bridge the gap by providing a DIYWindow class for readers to use with the first book of my *Do-It-Yourself Java Games* series. DIYWindow is a helper class that allows the reader to focus on learning Java programming fundamentals while creating window-based programs that are easy to export and share.

By removing the need to understand how the window-interface code works from the first book, my books have shown that beginning programmers can easily learn Java

programming. Readers get excited about the great variety, quantity, and quality of games they write as they work through these books. High school teachers praise the books for making the students think about the code they are writing and for allowing the students to work at their own pace without much assistance. College professors confirm that these books give the students a firm foundation before moving on to other college-level computer classes. And in one case a college professor shared with me that after completing a beginner exploratory computer programming class in which he used my books, a couple nursing students switched their majors to computer science. So yes, I believe that if taught correctly, Java can easily be a learned as a first programming language.

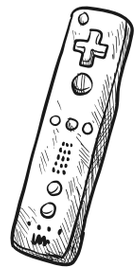
Practice is Key to Becoming a Good Programmer

Key to learning programming is simply to write a lot of programs. As you work through my books, you'll complete 39 programs, a variety of text-based games, window-based games, video games, and network games. Each book in the series builds on skills taught in the previous book, yet each book is completely independent. A reader may start anywhere in the series, based on his Java programming experience and current needs.

Start Easy with Text-Based Games

Beginning programming books must start simply, usually with programs that just print text. But text-based games can be fun to play, too. Remember, the first popular computer games were all text-based.

Text-based games are easiest to write because they use only text as input or output. The program pauses whenever it is ready for more data and it knows what kind of input to expect at every pause. Create *ASCII Art*, *Name Game*, *Silly Story*, and *Choose Your Own Adventure* to learn how to use the development environment, and how to work with strings, variables, conditional statements, and methods. Learn how to manipulate numerical data, generate random numbers, handle exceptions, and repeat loops of code, by creating *Calculator*, *Temperature Converter*, and *Ten Little Chocolates*. Learn about arrays and how to read and write to a file as you create *Word Scramble*, *Secret Code*, *Word Mastermind*, and *Hangman*. Finally, come to better understand object oriented programming as you put together everything you learned so far to create *Crazy Eights*.



Action with Window-Based Games!

A window-based program creates an initial screen, then waits for the program user to take an action. The user may click a button, click on an image, drag something, or press a key on the keyboard. The program then responds to that action, whatever the action may be. This is referred to as event-driven programming.

For event-driven programming you customize existing components – like buttons, input fields, menus, and panels – that create events when the program user interacts with them, and you add listener code to the components to respond to those events. Learn about window layout and button events as you create *Wizard of Yes/No*, Guess *My Color*, *Framed*, and *Watch Your Step*. Learn about images, drawing on the screen,

keyboard input, and how to create dialogs as you create *Sliding Tiles*, *Maze Generator*, *Greedy*, *Word Builder*, and *Image Resizer*. Lastly, combine everything you've learned so far and learn how to move objects around on the screen with the mouse as you create *Baker's Dozen Solitaire*.

Of Collisions and Threads: Animated Video Games

Unlike the window-based games that create an initial screen and wait for a user to take an action, animated video games use threads to perform automated activities, whether the user takes an action or not.

Activity within an animated video game is easier to manage using state-driven programming. Various threads of the program may change the game state at any time. Another thread checks the game state at regular intervals and determines which methods to call, based on the current state of the game.

Write *My Timer* and *Speed Words* to learn how to start and stop a timer and create a timed game. Write *Match Three* and *Baby Bird* to learn to detect collisions and how to use threads to automatically move objects. Learn to use abstract classes in *Falling Bricks* and to use state-driven programming in *Gravity Well* and *Cat and Mouse*. And finally, wrap up all the concepts learned so far and add model/view/controller design in *Blitz*.

Add Complexity to the Mix: Networked Games

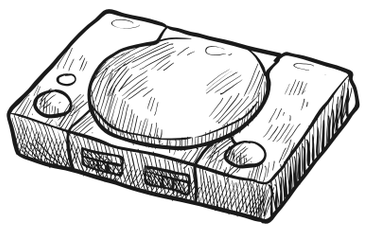
Networked games are usually written with two programs: a client program that each player runs on his own computer, and a server program that is run on another computer. The server program controls who can join the game and manages all the socket communication between the computers.

Create *Hit Counter* to learn how to communicate across a private home network or over the public internet, and how to reduce security risks. Learn to maintain connections between one server and many clients in *Chat*. Learn to write a turn-based game, to pair up any two players, and to use client-side game logic in *Dots and Boxes*. In *Word Hunt*, learn to restrict who can play the game, how to use server-side game logic, and how to control a timed game. In *Snakes*, learn to control continuous client animation from the server. In *Double Solitaire*, learn to control a fast-paced action game that gets non-stop input from the clients and manages all the activity using queued data, timer threads, and synchronized processes. And finally, learn a little about streaming audio in *Voice Chat* so you can talk to other players as you play these games.

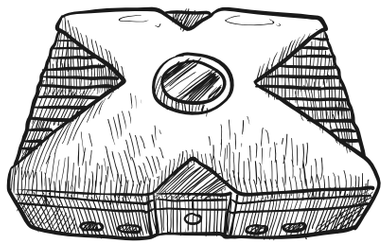
What You Need for a Future Career in Computer Programming

Computers touch nearly every aspect of modern life. Job opportunities in computer programming are many and varied. Indeed.com shows more job openings for Java programmers than for programmers of any other programming language.

Computer programmers need problem solving and logical thinking skills. They must be creative, think analytically,



and be detail oriented. Careers in computer programming include software application developers, web developers, computer system engineers, computer systems analysts, network administrators, and more. Have fun creating your own games while you develop the same skills that can start you on a promising computer programming career path. Happy programming!



[1] Global Games Market Report https://newzoo.com/insights/articles/newzoo-2017-report-insights-into-the-108-9-billion-global-games-market

[2] The State of Online Gaming 2019 research report based on responses from 4,500 consumers in France, Germany, India, Italy, Japan, Singapore, South Korea, the United Kingdom, and the United States age 18 and older who play video games at least once a week. https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019

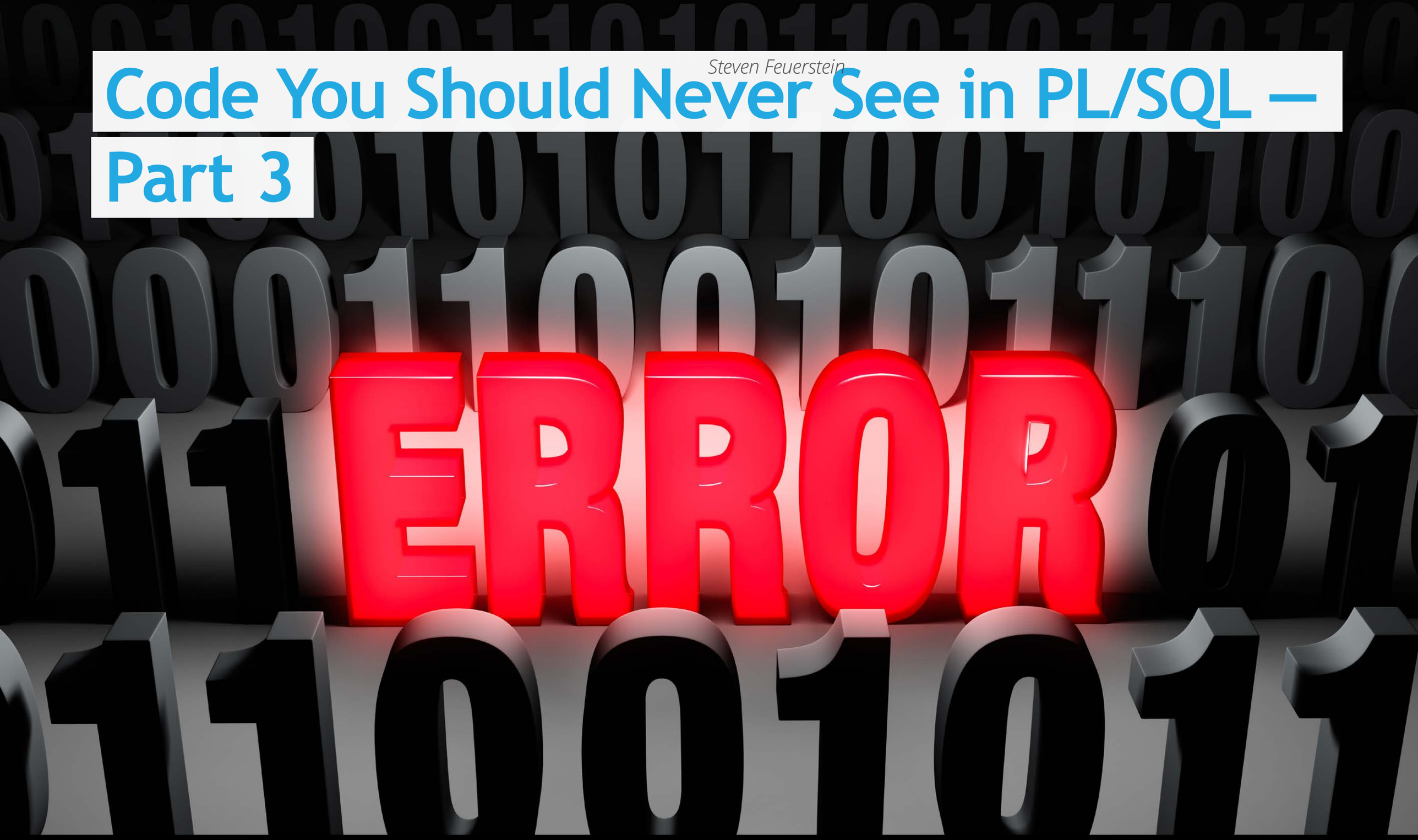
Annette Godtland

Annette Godtland worked professionally as a programmer, 20 years for IBM and 5 years for Kingland Systems. She is the author of the Do-It-Yourself Java Games series of books:

- *Do-It-Yourself Java Games: An Introduction to Java Computer Programming* – learn the fundamentals of Java programming as you create fourteen different text-based games. No previous programming experience is required.
- *More Do-It-Yourself Java Games: An Introduction to Java Graphics and Event-Driven Programming* – learn how to program windows, icons, and menus as you create ten more colorful, more interactive games.
- *Advanced Do-It-Yourself Java Games: An Introduction to Java Threads and Animated Video Games* – learn to control multiple, simultaneous activities as you create eight more lively, audible games.
- *Do-It-Yourself Multiplayer Java Games: An Introduction to Java Sockets and Internet-Based Games* – learn to communicate across a network as you create seven new games that you can play with friends on other computers.

Code You Should Never See in PL/SQL — Part 3

Steven Feuerstein



Part 3 of our series about mistakes you should (and can) avoid in PL/SQL!

WHEN NO_DATA_FOUND for non-query DML

When you execute an implicit query (SELECT-INTO), PL/SQL raises NO_DATA_FOUND if no rows are returned by the query. As in:

```
DECLARE
  l_id   employees.employee_id%TYPE;
BEGIN
  SELECT employee_id
     INTO l_id
     FROM employees
  WHERE 1 = 2;
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN
     DBMS_OUTPUT.put_line ('Not with that WHERE clause!');
END;
/

Not with that WHERE clause!
```

Bad Code

But when I run this block (changing the SELECT to an UPDATE), no output is displayed.

```
BEGIN
  UPDATE employees
     SET last_name = UPPER (last_name)
  WHERE 1 = 2;
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN
     DBMS_OUTPUT.put_line ('Not with that WHERE clause!');
END;
/
```

That's because the PL/SQL engine does not raise an error for non-query DML that change no rows.

Cleaned Up

If you need to know that a row was modified (or how many rows were modified), use the SQL%ROWCOUNT attribute.

```
BEGIN
  UPDATE employees
     SET last_name = UPPER (last_name)
  WHERE 1 = 2;

  IF SQL%ROWCOUNT = 0
  THEN
     DBMS_OUTPUT.put_line ('Not with that WHERE clause!');
  END IF;
END;
/

Not with that WHERE clause!
```

Loop containing non-query DML inside

Last up, a biggie. That is, bad code with a potentially enormous negative impact on performance. Here goes:

Bad Code

```
CREATE TABLE parts
(
  partnum   NUMBER PRIMARY KEY,
  partname  VARCHAR2 (15) UNIQUE NOT NULL
)
/

BEGIN
  FOR indx IN 1 .. 10000
  LOOP
     INSERT INTO parts (partnum, partname)
       VALUES (indx, 'Part' || indx);
  END LOOP;
END;
/
```

When you execute the same DML inside a loop repeatedly, changing only the variables that are bound into the statement, you are unnecessarily (and often dramatically) slowing down your code. The problem here is that I am switching between the PL/SQL and SQL engines 10,000 times. Those context switches are relatively expensive.

You should avoid this kind of row-by-row processing whenever possible. The key anti-pattern to look for is a loop with non-query DML (insert, update, delete, merge) inside it.

Cleaned Up

There are two ways to clean up this slow code.

1. Write it in “pure SQL” if you can. If you don’t need PL/SQL algorithms to process the data, and can do it all in SQL, then do it that way. For example in the above case, I could have written:

```
BEGIN
  INSERT INTO parts
       SELECT LEVEL, ‘Part ‘ || LEVEL
       FROM DUAL
    CONNECT BY LEVEL <= 1000;
END;
```

If you need PL/SQL or simply cannot figure out how to write it in pure SQL, then use the FORALL statement instead of a FOR loop. This statement results in just one context switch, with PL/SQL sending all the bind variables from the bind array (l_parts) across to SQL with a single INSERT statement (that statement, after all, never changes).

```
DECLARE
  TYPE parts_t IS TABLE OF parts%ROWTYPE;

  l_parts   parts_t := parts_t ();
BEGIN
  l_parts.EXTEND(10000);

  FOR indx IN 1 .. 10000
  LOOP
     l_parts (indx).partnum := indx;
     l_parts (indx).partname := ‘Part’ || indx;
  END LOOP;

  FORALL indx IN 1 .. l_parts.COUNT
     INSERT INTO parts (partnum, partname)
       VALUES (l_parts (indx).partnum, l_parts (indx).partname);
END;
/
```

For a much more detailed demonstration of converting row-by-row processing to bulk processing, check out my **LiveSQL script**.

Check the **doc** for lots more information on PL/SQL bulk processing.

Richard Foote

The Importance of Data Clustering — Part 2

In "The Importance of Data Clustering Part 1", we looked at how when it comes to application performance within the Oracle Database, one of the more important considerations is Data Clustering. In Part 2 we will discuss how bad data clustering can be significantly improved by assigning a table a* Clustering Attribute *and reorganising the table via the* TABLE MOVE *command. Associated indexes based on the Clustering Attribute will have a significantly improved Clustering Factor and results in much more efficient index-based accesses.

Depending on the number of distinct column values, it might be possible to improve the Clustering Factor of multiple columns concurrently. Providing there are enough repeated values to allow reasonable sub-groupings of additional columns, it might be possible to get good clustering characteristics of more than one column.

To illustrate, a simple example where a table has 2 columns, each with 100 distinct values:

```
SQL> create table ziggy (id number, code number, grade number, date_created date,
name varchar2(42));

Table created.

SQL> insert into ziggy select rownum, mod(rownum, 100)+1, ceil(dbms_random.val-
ue(0,100)), sysdate - mod(rownum,1000), 'ZIGGY STARDUST' from dual connect by level
<=4000000;

4000000 rows created.

SQL> commit;

Commit complete.
```

Both CODE and GRADE columns each have 100 distinct values, but because CODE is populated via MOD(rownum,100) which populates each value throughout the table and GRADE is populated via a random number generator, each have a very poor data clustering within the table.

If we look at the Clustering Factor (CF) of each corresponding index:

```
SQL> create index ziggy_code_i on ziggy(code);
Index created.
SQL> create index ziggy_grade_i on ziggy(grade);
Index created.
SQL> select index_name, clustering_factor, num_rows from user_indexes
     where table_name='ZIGGY';

INDEX_NAME                CLUSTERING_FACTOR NUM_ROWS
------------------------- ----------------- ----------
ZIGGY_CODE_I              2194994                4000000
ZIGGY_GRADE_I             1843514                4000000
```

We notice the CF of both indexes are relatively poor at **2194994** for the CODE based index and **1843514** for the GRADE based index.

If we run a query that selects rows based on a predicate on say the GRADE column:

```
SQL> select * from ziggy where grade=42;

39979 rows selected.

Execution Plan
----------------------------------------------------------
Plan hash value: 2421001569

--------------------------------------------------------------------------------
| Id  | Operation         | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |       | 39979 |  1366K|  6105   (1)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| ZIGGY | 39979 |  1366K|  6105   (1)| 00:00:01 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

     1 - filter("GRADE"=42)

Statistics
----------------------------------------------------------
          0      recursive calls
          0      db block gets
      22107      consistent gets
          0      physical reads
          0      redo size
     870268      bytes sent via SQL*Net to client
        685      bytes received via SQL*Net from client
          9      SQL*Net roundtrips to/from client
          0      sorts (memory)
          0      sorts (disk)
      39979      rows processed
```

We notice the GRADE index is deemed too expensive by the CBO because of the poor CF of the associated index and a Full Table Scan is used.

As discussed in Part 1, we can improve the CF of an index by assigning the corresponding Clustering Attribute to the table and reorganising the table. To improve the CF for say the CODE column:

```
SQL> alter table ziggy add clustering by linear order (code);
Table altered.
SQL> alter table ziggy move online;
Table altered.
SQL> select index_name, clustering_factor, num_rows from user_indexes
     where table_name='ZIGGY';

INDEX_NAME             CLUSTERING_FACTOR   NUM_ROWS
---------------------- ----------------- ----------
ZIGGY_CODE_I                       22065    4000000
ZIGGY_GRADE_I                    1849820    4000000
```

We notice the CF has indeed improved significantly for the CODE based index from 2194994 to just **22065**. However, as there's no inherent relationship between the CODE and GRADE columns, the index based on GRADE remains poor and relatively unchanged at **1849820**.

As the CODE column has only 100 distinct values (and therefore 40,000 rows per CODE value), it's quite possible to also effectively sub-cluster the GRADE column as well within each CODE value.

In addition to the CLUSTERING BY LINEAR ORDER clustering attribute used above, which can order columns primarily on the first column listed and then sub-order by subsequently listed columns, there is also the **CLUSTERING BY INTERLEAVED ORDER** clustering attribute. This uses a Z-order curve clustering technique that can potentially provide excellent clustering results for multiple columns that have reasonable numbers of repeated values.

If we look at using CLUSTERING BY INTERLEAVED ORDER on both the CODE and GRADE columns:

```
SQL> alter table ziggy drop clustering;

Table altered.

SQL> alter table ziggy add clustering by interleaved order (code, grade);

Table altered.

SQL> alter table ziggy move online;

Table altered.

SQL> select index_name, clustering_factor, num_rows
     from user_indexes where table_name='ZIGGY';

INDEX_NAME             CLUSTERING_FACTOR   NUM_ROWS
---------------------- ----------------- ----------
ZIGGY_CODE_I                       31809    4000000
ZIGGY_GRADE_I                      27018    4000000
```

We notice we now have excellent, if not quite perfect, CF values for both indexes. While the CF for the CODE index is slightly worse at **31809** from when it was perfect at 22065, the CF for the GRADE index is much improved from 1849820 to just **27018**.

As a result, performance on **both** indexes will now be substantially more efficient than it was before the table reorganisation. If we re-run the previous query based on a GRADE predicate:

```
SQL> select * from ziggy where grade=42;

39979 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 3241548495

-------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |               | 39979 |  1366K|   352   (1)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID BATCHED| ZIGGY         | 39979 |  1366K|   352   (1)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN                  | ZIGGY_GRADE_I | 39979 |       |    80   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access(“GRADE”=42)


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
        370  consistent gets
          0  physical reads
          0  redo size
    1590788  bytes sent via SQL*Net to client
        684  bytes received via SQL*Net from client
          9  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
      39979  rows processed
```

We note the index is now automatically used and subsequent performance has improved significantly with the resultant reduction in Consistent Gets.

Whether to use LINEAR or INTERLEAVED clustering attributes depends on the data characteristics of the columns and whether it’s necessary to have as perfect as possible clustering on a specific column.

The problem with this approach is that any new rows added with conventional serial DML is NOT clustered based on the Clustering Attribute on a table. This means over time, the CF will degrade and a periodic reorganisation of the table might be necessary to keep index performance optimal.

Rather than reorganising the entire table, this is a good example of where Partitioning can help to make such maintenance tasks more viable and less resource intensive.

Simplistically, if we now alter the table and partition it based on a date column in which all new rows are now only inserted into the last table partition:

```
SQL> alter table ziggy
  modify partition by range (date_created)
     (partition p1 values less than (TO_DATE('01-JAN-2017', 'DD-MON-YYYY')),
     partition p2 values less than (TO_DATE('01-JAN-2018', 'DD-MON-YYYY')),
     partition p3 values less than (maxvalue)) online;

Table altered.

SQL> select table_name, status, partitioned from user_tables
  where table_name='ZIGGY';

TABLE_NAME    STATUS      PAR
------------  --------- ---
ZIGGY            VALID    YES
```

If we now look at the current CF of the indexes after the table is reorganised to be partitioned:

```
SQL> select index_name, clustering_factor, num_rows from user_indexes
  where table_name='ZIGGY';

INDEX_NAME                CLUSTERING_FACTOR NUM_ROWS
------------------------- ----------------- ----------
ZIGGY_CODE_I              47195                4000000
ZIGGY_GRADE_I
```

We notice the CF have deteriorated somewhat as the indexed data is now physically spread across different table partitions.

The Clustering Attribute has no effect with standard serial DML, so when more data is added to the table, it could be inserted in a random manner in relation to the indexes causing the further deterioration of the CF of indexes. However, as the table is partitioned now by a date which results in all new rows being inserted into just the latest partition within the table, only rows in this partition need to be reorganised in order to address the CF issue, rather than the entire table.

To illustrate, we now insert more rows into our ZIGGY table, but all new rows are only being inserted into the last table partition:

```
SQL> insert into ziggy select 4000000+rownum, mod(rownum,100), ceil(dbms_random.val-
ue(0,100)), sysdate, 'DAVID BOWIE' from dual connect by level <= 500000;

500000 rows created.

SQL> commit;

Commit complete.

SQL> exec dbms_stats.gather_index_stats(ownname=>null, indname=>'ZIGGY_CODE_I', esti-
mate_percent=>null);

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.gather_index_stats(ownname=>null, indname=>'ZIGGY_GRADE_I', esti-
mate_percent=>null);

PL/SQL procedure successfully completed.

SQL> select index_name, clustering_factor, num_rows from user_indexes
  where table_name='ZIGGY';

INDEX_NAME             CLUSTERING_FACTOR  NUM_ROWS
---------------------  -----------------  ----------
ZIGGY_CODE_I           302299             4000000
ZIGGY_GRADE_I          255922             4000000
```

We notice the CF of both indexes are much worse than they were previously (CODE index is now **302299** up from 47195 and the GRADE index is **255922** up from 36396) due to the new rows that have been inserted into the table that haven't been clustered effectively.

However, rather than having to reorganise the entire table, we can simply reorganise just the latest partition in the table to improve the CF and efficiency of both indexes:

```
SQL> alter table ziggy move partition p3 update indexes online;

Table altered.

SQL> exec dbms_stats.gather_index_stats(ownname=>null, indname=>'ZIGGY_CODE_I', es-
timate_percent=>null);

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.gather_index_stats(ownname=>null, indname=>'ZIGGY_GRADE_I',
estimate_percent=>null);

PL/SQL procedure successfully completed.

SQL> select index_name, clustering_factor, num_rows from user_indexes
  where table_name='ZIGGY';

INDEX_NAME             CLUSTERING_FACTOR NUM_ROWS
---------------------  ----------------- ----------
ZIGGY_CODE_I           49953             4000000
ZIGGY_GRADE_I          39056             4000000
```

The CF of both indexes are now back to being excellent, only a little higher than before the new rows were inserted because of the impact the actual new rows have in the CF values.

The judicious use of partitioning means the overheads associated with periodic table maintenance and reorganisations to keep the CF of indexes optimal is feasible and can be easily managed.

Of course, the judicious use of a partitioning strategy can result in some indexes being redundant due to the effective use of partition pruning, but that's a discussion for another day...

Sanela Lukavica

From Symbol to Meaning: Python's Logo

Once they decide to embark on the path of entrepreneurs and start a company, even the most down-to-earth techies will eventually have to find the right logo for their new endeavor. The interesting thing about logos is that they belong to the realm of the symbolic, and – provided the business succeeds and grows in importance – even create a narrative of their own. Thus they bring together the business background of companies with symbolic meaning that might gradually spread into different directions. The greatest accolade to a company will probably be exactly this engagement of people with its history, and perhaps especially so if it turns out not to have much in common with the creator's intentions in the first place.

Such is the case for the Python logo, which entails a plethora of different interpretations today. It all started out pretty simple: Dutch born and bred Guido van Rossum was contributing to the efforts of a team building the language "ABC", but he wasn't completely happy with the outcome. His participation had given him enough insight, though, to start a new – his own – programming language: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). (...) I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers." [1]

What ultimately gave the new language its name, though, wasn't Van Rossum's inclination for reptiles, but rather his passion for a good laugh. As a lover of Monty Python, his search for a suitable name almost naturally drew him to the British comedy group – and literally led to an entirely different beast.[2] From the 1990s to 2006, the company used a simple wordmark slightly reminiscent of the leathery skin of a snake.[3] It was in 2006 that the all-rounder Tim Parkins (who didn't belong to the Python staff and soon after became a photographer[4]) designed the two snakes that are so well-known today. Needless to say that the motif of the logo incites endless interpretational efforts: people in forums tend to see all kinds of meanings and symbols in the snakes.[5] It was in one of those virtual discussions that Parkins himself decided to throw a light on the enigma: "p.s. [sic] the logo is actually based on mayan [sic] representations of snakes which very often represent only the head and perhaps a short length of tail. The structure of the snake representations the natural coiling/nesting of a snake (...). The mesoamerican [sic] calendar also represents snake heads in a similar manner (...)." [6] And he makes it clear that a certain openness to interpretation is absolutely wished for: "The abstraction of the snake design used in mayan [sic] culture seemed non-denominational enough to only raise contrived objections. The shapes used (cross/spiral/yin-yang) are also primitive enough that there will always be connotations that can be derived. (...) If you look carefully at the logo, you will also see an indian [sic] symbol of peace.. (I'll leave this one alone as it can also mean something else)."

Thus if you ever wondered what the Python logo actually means, you can be pretty creative. But then again, that goes for (almost) everything in life. Have fun finding your own little interpretation!

[1] https://www.python-course.eu/python3_history_and_philosophy.php
[2] https://www.python-course.eu/python3_history_and_philosophy.php
[3] https://en.wikipedia.org/wiki/History_of_Python#/media/File:Python_logo_1990s.svg
[4] http://www.timparkin.net
[5] https://www.thecodingforums.com/threads/proposed-python-logo.356848
[6] https://www.thecodingforums.com/threads/proposed-python-logo.356848

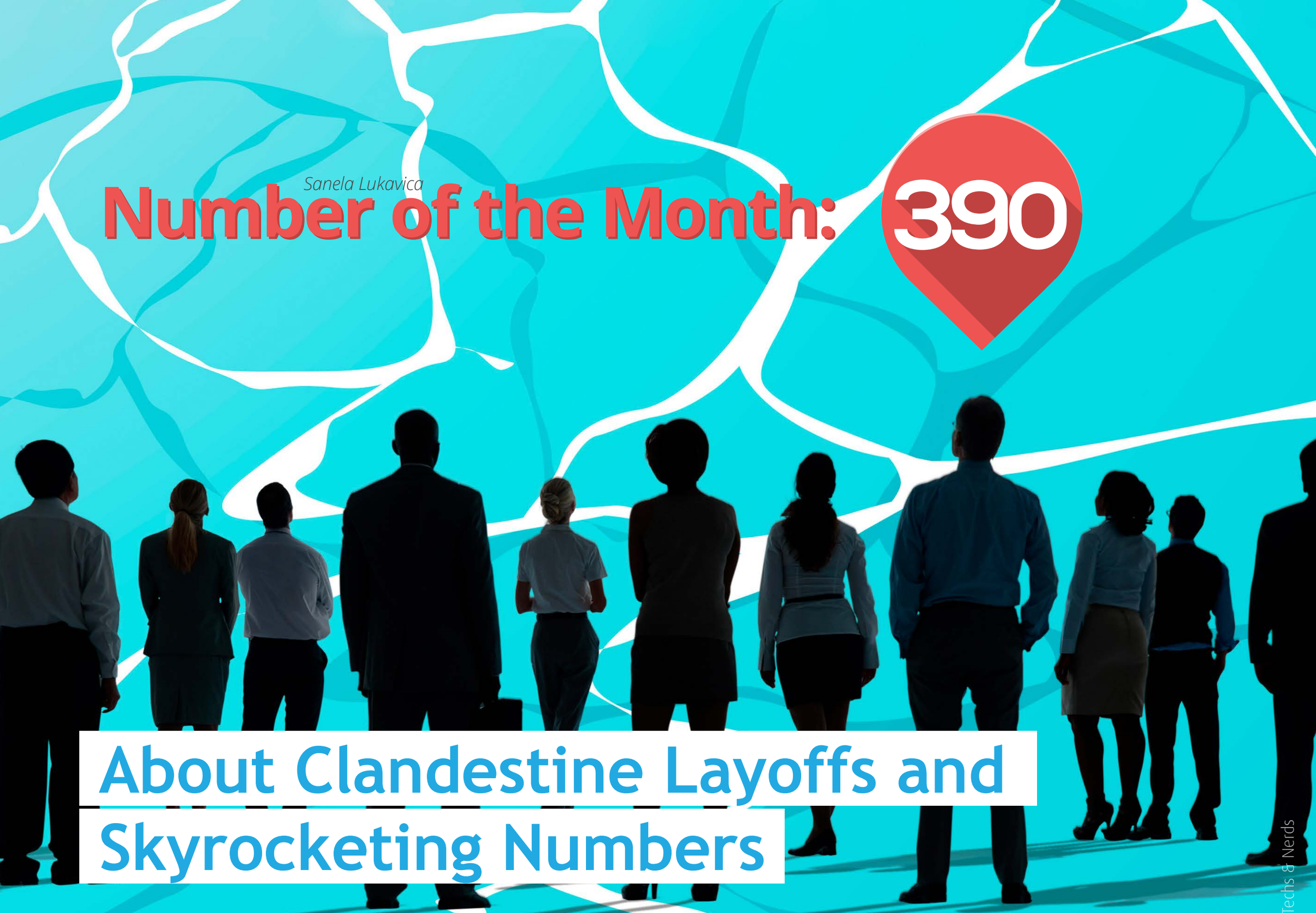
Sanela Lukavica

Empty Me, Damn You! I'm Trash!

Jokes usually have in common that they are made by humans, and apply to certain rules for humor. If you ever wondered what will happen when you let a computer co-write a sketch, you will get the answer in the video. We have to admit that we found the mixture of almost anarchical disregard for all kinds of logic strangely charming, and hope that you enjoy it as much. Meanwhile, we have heard that the algorithm used has been improved, and are terribly sorry to inform you that this will be the computer's first and last guest performance in our magazine.

Sanela Lukavica

Number of the Month: 390



About Clandestine Layoffs and Skyrocketing Numbers

Starting from March, Oracle has been laying off 250 employees in Redwood City, 100 in Santa Clara, as well as 40 people in Seattle. The dismissals were neither confirmed nor denied by the company, with Oracle spokesperson Deborah Hellinger merely publicly commenting: “As our cloud business grows, we will continually balance our resources and restructure our development group to help ensure we have the right people delivering the best cloud products to our customers around the world.” [1]

The procedure only came to the public’s attention in March when Oracle filed California regulatory letters. What Oracle lacks in readiness to comment, its employees make up for by anonymously posting about the situation on various internet forums.[2] It is quite clear that many of the employees who were dismissed are originally from Mexico and India, thus their dismissal may entail the loss of their right of residence.[3] But according to some sources, employees residing in other countries have also been dismissed. Rather concisely, they cite 50 people in Mexico, 50 in Hampshire, UK, and another 100 in India.[4] Some believe that this has only been the first wave of Oracle layoffs, and that at the least 500, at the most 14,000 people will have been dismissed by the end of 2019.[5]

Interestingly enough, a considerable number of people charged with the enhancement of the cloud strategy wasn’t spared. A terminated employee from this group has commented on the events. According to him, two rivalling groups were actually working on the cloud: the Silicon Valley group, and the Seattle group. But the Seattle group has reigned supreme: “Maybe that group in Seattle are the ones that have their game together”.[6] With many former Amazon employees and declared experts on all things cloud among the members, it seems that the decision was not a particularly difficult one. As a matter of fact, the anonymous former employee even believes that the new path makes sense. With the acquisition of new companies, Oracle’s workforce has also grown, but not always to its advantage: “They have a ton of mid-level managers, to me they are mainly clueless.”

No one can blame a company for “streamlining”, if circumstances require such measures. What is important, though, is to be transparent about the goals of the process. If these are left in the dark, it is not only unfair to the employees, but suggests that other motifs are hidden as well. This can only further damage a company’s reputation.

But as long as Oracle is not disclosing the most important goals of its strategy to its employees, one can only speculate on the company’s aims. Meanwhile, the assumed layoff numbers will keep skyrocketing in a manifestation of doubt.

[1] https://www.mercurynews.com/2019/03/27/oracle-cuts-hundreds-of-jobs-in-silicon-valley/
[2] https://www.thelayoff.com/oracle
[3] https://www.theregister.co.uk/2019/03/25/oracle_headcount_cut/
[4] https://en.globes.co.il/en/article-oracle-worldwide-layoffs-not-hitting-israel-yet-1001279666
[5] https://www.theregister.co.uk/2019/03/25/oracle_headcount_cut/
[6] https://www.businessinsider.com/laid-off-oracle-cloud-developer-power-struggle-seattle-silicon-valley-2019-3?IR=T

Sanela Lukavica

Oracle Audits : Lessons Learned and Notes to the Future

In 2015, the working group "Club Utilisateurs Oracle" was created as a subgroup of the French Oracle user group "Association des Utilisateurs Francophones de solutions Oracle (AUFO)". The group provides its members with information on Oracle licensing practices and supports them before, during, and after Oracle audits. Sanela Lukavica has talked to one of the leading members of the Club Utilisateurs Oracle, Frédéric Cessy, about Oracle's aims concerning the audits, as well as the best steps companies can take to prepare for an audit. Frédéric, who is also the founder and director of the company "More Light Consulting", is a leading expert in the field and looks back on a 18-year career with Oracle.

What was the incentive to start a working group that deals with Oracle audits in the first place?
Like other communities or working groups which were created in AUFO, the main goal is to share best practices and exchange feedback.

Regarding the topics around licensing, the incentive was the increasing need of customers to understand what is an audit, the different steps, rights and duties of users, and the reasons leading Oracle to state for a non-compliance issue at the end of an audit. I thought I have some knowledge that could interest Oracle users.

Can you give an insight into the work of the group?
We meet twice a year. During the first two or three meetings of the group, the manager and employees of the Oracle LMS department explained the licensing rules. After that, several subjects have been discussed only with users; starting by workshops on licensing, Discovery and SAM tool efficient on Oracle product, Oracle licensing on the Cloud, The software asset management, best practices for managing an audit...

What are the main goals of the group?
The main goal is to provide Oracle users with an accurate view on their situation in terms of licensing, to exchange on experience and best practices. And outside of the workshops, the group is also an area to ask question on license or to request assistance on a specific topic. From the feedback I received from customers, I would say that we've succeeded in assisting customers to highlight the risks related to their situation and the way to control it.

Your background actually lies with Oracle. What was your main motivation for working in the group?
It came very naturally: having managed several hundred audits of Oracle customers, I know their impact and the difficulty that customers have to face in such circumstances.

With the group I knew I could share this experience and transform it in good practices that could help customers to better manage Oracle audits.

And the audits aside, I think this experience in licensing and Oracle contract could be helpful for customers to better understand their Oracle agreement for using their licenses more efficiently. That is the aim of Oracle software asset management workshops I'm leading for now 5 years.

What do you consider the biggest problem of audits by Oracle at the moment?
VMware of course is a big problem. But contrary to what we hear, Oracle rules didn't change. They just apply the processor rules which are basically the same since 16 march 2009, but very strictly.

Oracle has always considered physical CPU where the software is installed and/or running. The issue is that the flexibility of VMware increased release after release and mathematically it increased the scope of machines where the software is installed and/or running.

And that leads to the following very difficult situation: customers rarely evaluate the impact on Oracle licenses when they upgrade their VMware environment, and there has been so much wrong information on this topic that customers hardly accept Oracle's arguments to present the situation.

So, it's very hard for customers to believe the conclusions presented by Oracle. And they often try to fight uselessly instead of considering a way to control their license consumption on VMware.

There is also another problem with Oracle audit that is related to ethics.

It's the Joint Partner Engagement program. Oracle still has problems auditing small customers. Several programs have been initiated but none of them resolved the issue in the balance between time spent on the audit and the licenses sold at the end.

So, in 2008, Oracle decided to mandate third parties for auditing small customers.

But, contrary to the practice of Microsoft or Adobe, Oracle does not pay the selected companies for that. The agreement is that they will be paid by selling the licenses that will resolve the compliance issue they have highlighted.

In your experience, which circumstances draw Oracle's attention the most and thus lead to an audit?
When I was leading the Oracle audit service in France, half of the nominations for audits came from the sales team. So we can say that the first reason for being audited is probably coming from the relationship with the sales representative. It could be because of a non-compliance issue known by the sales team that they've failed in resolving with a negotiation out of the audit service. Or a situation where the sales rep does not see any opportunity … so he has nothing to lose in requesting LMS to check customer's compliance.

For the audit initiated by the License Management Services, the reasons are purely on facts.

The first criteria is the date of the latest audit: Oracle aims to audit his customer every 3 years.

Then several criteria can put a customer in the LMS radar: such as the customer situation: if something changed in the customer definition (acquisition, merger, divestiture, …), there is often an impact on the compliance situation. The second criteria is the type of license: a customer with no processor license is probably non-compliant, and a customer with old metric is probably non-compliant. Also, the number of licenses compared to the size of the company is important (LMS can check with a similar company in the same industry), the difference in licenses between products (I was checking the ratio versus database to evaluate the level of potential non-compliance), the date of the last purchase, and today, I think a customer known to use VMware could be an interesting target for LMS.

Is there a way to avoid Oracle's attention, and thus avert an audit?
Yes and no, as said before, Oracle aims to conduct an audit every 3 years on all customers. So normally all customer should be audited one day. On one hand, an audit is interesting if it leads to a deal. On the other hand, Oracle sales team prefers to have a good relationship with their customer, so I guess that a customer who makes regular purchases or offers a real opportunity for a good deal will not be on top of the list for being audited.

How important is the proper preparation before an audit, and what can smaller companies do that don't have a whole legal department to deal with Oracle?

The proper preparation is more than important: it's crucial. Whatever the size of the companies, an audit can have huge financial impact so all companies should be prepared for an audit from their main editors.

The legal department is rarely the best entity to prepare the audit, they will be a strong assistance during the audit, but for preparing it, the ball is more in IT department. If needed, they should receive the assistance of the purchase service to understand the licensing rules, to have the number of licenses, and then the IT should be able to evaluate the deployment and the usage in order to control the risk of non-compliance.
Of course, for bigger companies an efficient software asset management system is the best way to prepare an audit.
But it should not be only a SAM tool in place to reassure the management, otherwise there could be the risk that the reality is hidden by the tool, which is the worst situation before an audit.

In brief, for all customers, regardless of the size of their company, the two pillars for preparing the audit are to ensure that the contract and the definition are well understood, and be able to collect the appropriate technical data from measurement.

Around this you can then build a SAM organization or just an audit protocol, but these two pillars should be the solid basis.

How does your cooperation with companies who have to go through an audit look like?
It depends of the situation: If the audit has been done and the conclusions provided, I have to define a strategy in order to reduce the requested amount. If the audit is in progress, I act to slow down the audit process in order to check real usage, and then decide of the best approach with the customer. If the audit has not started yet, it's much easier: I conduct an audit exactly like Oracle would do, and search for any current or past usage that could lead to a non-compliance issue. And then we work to reduce the footprint at the expected level, ideally in accordance with license agreements.

Would it be a smart move to refuse an audit in your opinion after you are asked to go through the procedure?

No, that would not be a good approach. An audit is a contractual right and it's normal for an editor to protect his contractual and intellectual property rights.

This said, an audit should be done accordingly to the agreement, and a customer could reject any action that does not respect the audit clause.

The first important aspect is the notice period, I've seen several audits started before the 45 days while it's important to use that period to verify the real usage.

The second aspect: even if customers must cooperate, an audit should not "unreasonably interfere with business operations". So, with realistic reasons, customers can request to start the audit later to avoid impact on their business.

How do you see the future development of Oracle audits?
Over the last ten years I've seen an evolution in the audits. Before 2002, it was a lax approach: unclear contracts, internal policies not respected (side letters). The audit service's budget was under the local sales team responsibility. From 2002 to 2008, the period of negotiation took place. The audit services became better organized as well as the audit service's budget switched under Finance at Corp level. Value added services have been created, Oracle initiated a software asset management service. As of 2009, Oracle has applied a coercive approach: the non-compliance situation was money belonging to Oracle that customer must give back. Recently, I've seen Microsoft and some other big players focusing more on education and cooperation, hence I guess Oracle will also go in this direction.

What would be the optimal future development in your opinion?
I think audits will always exist as they have become an important revenue source. And Oracle will always have something to leverage the audit: they've used the minima rules to get money from Named User Licences extensively, they've used the data transfer and multiplexing to explain the need of Processor Licences, they've used VMware to ensure ULA was the best solution to ensure compliance, and I know some other topics they could use to push their Cloud solutions. But of course I do not want to give them a good idea.

So, audits will continue, but the form of the audit could change but that will probably depend of the success Oracle will have with their cloud strategy.

From my point of view the optimal development should be the self-audit.

For that Oracle should provide appropriate service, transparency and trust to enable customers to evaluating their own compliancy easily and encourage them to cooperate in an Oracle SAM process. But I don't think that will happen soon.

The obstacles partially lie on Oracle's side: the behavior, the lack of transparency for measuring real usage (for each product line we should have a matrix showing what table, what features correspond to an extra cost option or product). Today, having such information is very difficult and even impossible for some product like Siebel.

Frédéric Cessy
Director "More Light Consulting"

Sanela Lukavica

App Development. An Update

Michal Harakal is a software developer (Telekom) and has been a passionate Android developer from the very beginnings of the operating system. In the following interview, Michal talks about the vicissitudes of app development and the changes he encountered through the years. Sanela Lukavica has conducted the interview.

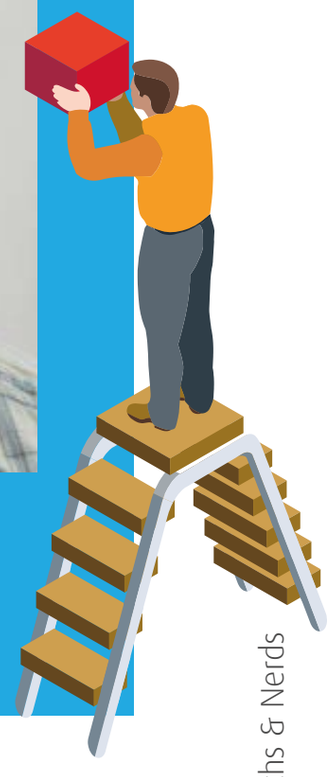
After 13 years in mechanical engineering, you made a professional change in 2010 into the mobile world, with focus on Android. How did app development change in the last years in your opinion?

I was on the Android boat right from the start and followed all the developments. The most important aspect certainly is that development has become professional. In the beginning, it was about the ideas you could implement on mobile devices that opened up completely new possibilities. For example, how the user interface of an app should look like on such a significantly smaller screen. This changed over the years: apps are also developed in larger teams. Quality assurance also became important. What you know from normal software development, that is to say subjects such as software architecture, test coverage, test automation, or static code analyses, the things that were common in case of back end system software with enough tools and the whole ecosystem; these things are now also common in the development of mobile apps. And this is true for both platforms, Android and iOS. From my point of view, we sometimes reinvented the wheel but also searched and found new solutions that have been specific to mobile apps. By now, the requirements are not that different anymore compared to a back end system if you want to write a top-class app.

Principally, people say that the control system of Apple is much stricter compared to Android's. Is this in line with your experience that Google puts up Apps faster than Apple? Can you tell us something about the process?

The handling is indeed different. Apps for the App Store undergo an approval procedure that is not really transparent to the developer, at least in beginning, this was the case. By now, it is known what to do and when an app is at risk of being rejected. Quality assurance and processing time Apple takes have changed significantly. Currently, everything is handled much faster. But publishing in the store really is the last step and could be handled by another team than development.

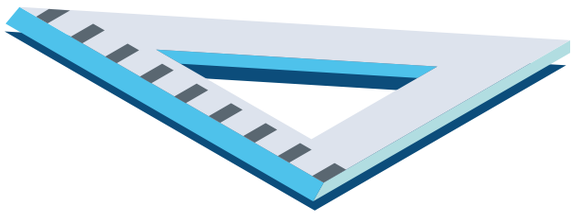
Michal Harakal
Software developer



However, many aspects on Android are still faster. Does this automatically result in a higher level of freedom for the developers?
In case of Android, you are very flexible and free when you put up the apps in the store. This may obviously lead to major bugs slipping through your fingers. For these cases, it is beneficial when you can fix bugs fast and put up the new version in the store. You could regard that as freedom. On the other hand, this leads to a lower quality of the app. In the past, it happened to me that many users have already installed a conference app and absolutely refused to update it – although it was necessary to eliminate some bugs. It also was announced on multiple communication channels that there were fixes. Evaluations showed that half the people kept on using the first flawed version. With that said, it is important to upload apps in the store that do not have bugs. Quality assurance is really important there. The freedom to act fast surely is an advantage, but, in this situation, it did not help me much. For me as developer and consumer, these are two very different processes, and if it is Google or Apple does not play an important role for me as developer. An indie developer who directly sells apps or uses in-app purchases as source of income will surely have a different view on this topic. My perspective is different: Apps that I developed so far have always been a part or an extension of a certain service. Be it a conference app or a festival app where you monetize the service in other ways.

Michal, your focus is on consumer apps. For example, you worked on a wheelchair-accessible map, "Wheelmap.org". If you try to satisfy the need of the consumers, you concentrate on the difficulties the consumer has everyday, or is it more in the nature of your task that this work remains on the surface and technical aspects are in the focus?
The user is indeed in the focus of the apps. The mentioned app was very specific. It was about helping people to find restaurants, offices, etc. barrier-free access. That is some time ago now. The founder of the Wheelmap project is a wheelchair user and he and his friends noticed at some point that they always meet in the same café to exchange ideas because they do not know another with access for wheelchair users. So they thought that the places could be mapped to get other people to go out more often. This did not only share knowledge on suitable places, but it indirectly also put a certain pressure on owners. I got to know the responsible persons back then at an Open Street Maps conference. Not only the app and the website but also accompanying projects have improved the data. For example, they worked with pupils who got smartphones and then mapped many places. From a user perspective, I really liked the indication if a place was accessible or not. In Germany, there is a DIN standard that exactly defines how high a step may be to be referred to as barrier-free. They decided to use a traffic light system instead. Green means: Everything is fine. Yellow: The wheelchair user can come in without difficulties but there may be some obstacles at this place. The restrooms may be only accessible by a stair, for example. And finally red: Access is difficult. This way, it was made clear if you can access the place and, if yes, the circumstances under which you can access it. That is a difference compared to the official, legal regulations. A simple pragmatic solution, not only the operation of the app on mobile use cases to highlight a certain place on your way, but also collecting the necessary data.

Another example of user orientation is the way the data end up in the database in the back end. The data are stored at the Open Street Map project and, to prevent data vandalism, they cannot be manipulated that easily. For users who wanted to start directly without registering with a new account, there was a generated de-facto collective account that collected, evaluated, and synced the data with the database. Later on, there was an actual option to get credits for these tasks for people who did that seriously and found it important. They could finally log on and their entries were then



published with their own names. That was very important from a user point of view. And so, an app keeps evolving. You see what users need and what is important to them.

The number of global app usage went down from 2013 to today. Germany is not an exception, but in other countries, especially in the USA, apps do not play the same role as they did. How do you see future developments?
In view of native apps, this is a question that I actually thought of quite a while recently because, as a developer, this affects me from various perspectives. How will we develop apps? On the one hand, only high-quality apps can be successful in stores. On the other hand, the effort used to develop a really good app becomes more and more important. How fast can you react on customer requests and changes? And are apps delivered without bugs or does the customer have to put up with bugs?

If you want to write a native app, you need specialized knowledge. The operating system of Android is very strange when it comes to handling background processes or data, for example. There are some general philosophic decisions that have been made at the beginning, on purpose, to achieve certain objectives. Many decisions still roll out technical basics today that are from a time when we did not have these strong CPUs and batteries were smaller, but most decisions are still valid.

When I work on a screen as a developer, I always have to consider that there may be an incoming phone call I want to answer immediately as a user. I have to make sure that a half-written text in an input field does not disappear when the user ends the call. These are the small details you only notice as a user when you are confronted with a poor app.

Native app is not the only option to write a mobile app. Web-based (html5) apps are scheduled from the start. Before iOS, this actually was the only option at the start of the iPhone. With Android, you could really observe how Google invested strongly in the Web, e.g., with the development of the Chrome browser or that web-oriented talks were as common as Android itself at Android developer conferences. Hybrid frameworks are now used to try hiding the details of a mobile platform with abstraction. As soon a new framework arrives, the motto is: You have to write it once, and then it runs everywhere. Typical cross-platform solutions such as Xamarin may have found their niche, but they did not achieve a widespread use. Anyhow, Android is still developed natively on a large scale. This is simply due to the technical basis, the decisions that have been made in the beginning. These decisions are, for example, that Android apps are written in Java, or in Kotlin now. However, we can actually see that, with the new generation of cross-platform solutions such as Flutter or React Native, you can build good enough apps. But I am still critical of these technologies, also after experiences others made and the fact that complexities and differences of mobile platforms cannot simply be solved with an additional layer, framework, or the usage of another programming language. I see the solution in code sharing and that is what I expect to see in the future. By using a suitable software architecture, you can write domain and business logic in a platform-neutral way and reuse it on both platforms. Google already declared Kotlin as number 1 language. We can really feel massive investments in terms of tools, libraries, blogs, example, and the whole ecosystem with third party libraries is developing at an incredible speed. With Kotlin Multiplatform and Kotlin Native, there is another important component for code sharing.

What is wrong with web-based solutions?
To me, web-based solutions, html5 apps, that run in the browser only work to a limited extent or not at all. There are various reasons: The first one are the web standards

themselves. Particularly older Android versions are way behind in the support of modern web standards and this excludes many users or additional effort is required to support them. Another reason is the UI. A web app either feels strange or, to get it nicely integrated into the operating system, a very high effort is required. However, the start time is important to me. If you build an app with a web browser, you have to start a web browser. But a web browser is a heavy-weight component that consumes many resources. Storage and CPU. A web browser always needs some hundred milliseconds to start. This means you take your smartphone out of your pocket, start the phone, start the app, and wait for half a second to start the whole thing.

How high is the effort to optimize the app starts?
And then the app itself needs one or two seconds to start. But for apps you use on your way every second counts. We have switched complete libraries to accelerate the app start when it was about improving the software development. Even back then, people realized that clean code and SOLID principles and "dependency injection" also led to source code quality for mobile apps. We then used a dependency injection framework in Android development. "Roboguice" was an Android-specific variant of "Guice" that came from the Java world. Due to technical reasons, how the introspection ("reflection") was implemented in the runtime environment on Android, it took very long until all instances were generated by the necessary objects and the whole dependency tree was built. This resulted in a very clean-written code, but the app took too long to start. When you actually reach the point to consider such things as start pages of screens or the amount of data transmitted, it is just too much. That is why we always kept looking for new solutions that not only carried out the tasks of dependency injection but also were suitable for Android. A huge effort was made only for this aspect. Another example is the date/time-API: With JSR 310, a new date/time-API was introduced in Java 8. To use this on older Java systems or older Androids without Java 8 support, there is a port (backport). In addition to the Java port, there is a variant specific to Android. The only reason for the Android version was to improve the app start. This is a huge effort because someone actually has to maintain, release, and test it. But it is worth it for these few milliseconds!

There is often a certain hype for programming languages. What are your experiences?
I can still remember that Scala was really popular 4 or 5 years ago. It is a modern language, something new, something different. It led to better code with fewer crashes. The Android community, with top developers, made a huge effort to bring Scala on Android. But we all noticed soon enough that it became difficult without official support. There were several reasons that made it difficult to really develop apps productively. Not much happened with Scala then. I attended my first presentation on Kotlin in fall 2017. Will history repeat itself with Scala or is it something new? I just waited for some time. Until I tried out Kotlin myself. Since then, Kotlin also became the official language of Google for Android. And this was not only declared by Google but also confirmed with various steps. By now, I am myself an enthusiastic Kotlin user and I prefer projects written in Kotlin. And it is simply fun to me, personally. In the meantime, Kotlin developed a life of its own. Kotlin itself as language is very stable. New libraries are added daily, although certain areas are still developing in great leaps (Kotlin Scripting, Multiplatform, or WebAssembly). Kotlin/Native and Kotlin Multiplatform are of major importance in my opinion, particularly the support for iOS is essential. A had a meetup presentation in Munich lately and reported on my experiences on Kotlin Multiplatform.

How do you write a good app?
There are many criteria to rate an app as good. At the end, it is the user who rates the app. I am convinced that a good app should be based on a good technical foundation. A

clear software architecture that suits the development team, component-based, with a distinct separation of responsibilities, presentation of layer and domain and business logic. The practices you develop are just as important: how you develop, starting with code reviews, test automation to a high test coverage. Initially, you have to consider what has to be done.

A crucial feature of a good app is that it reacts smoothly. Many long-lasting operations generally take place in the background to not block the UI (to prevent the slowdown when scrolling long lists, for example) and we, as developers, spend much time on syncing the UI with the operations in the background. If you want to download a file, for example, the download itself is handled in a parallel thread and not in the main UI thread. But I still want to display the upload progress. This means that these two have to communicate somehow. That makes the whole thing complex.

Another attribute that is often underestimated is the app size. You are not really confronted with that as a user, but there is a certain limit of usable numbers of methods of approximately 64,000 on Android. That is the DEX limit. The complexity of apps increased over time so that this limit has been exceeded. Obviously, solutions from Google have been added. These led to further challenges, however. I always try to check the impact of libraries on the app size. Libraries have an impact on the start time or also the size of the APK. And the size, in turn, is the reason for native apps and web apps becoming blurred. The concept of instant apps for Android is already two years old. You can install a native app without the PlayStore. Theoretically, you could have the link on the website and this is immediately loaded onto the phone and works instantly without going to the store. Obviously, there is a limit for the app size. Because they are so small, they are downloaded and started immediately.

These are just examples for all the factors that have to be considered in the background of app development.

Thank you very much for the interview, Michal!

Sanela Lukavica

Questions and Answers with Paolo Kreth from SOUG Switzerland

Are you curious about members of the other Oracle user groups? We have you covered! With Paolo Kreth you will get to know a member of Swiss SOUG and take a look behind the curtains of the perspectives as well as the challenges of a user group. Sanela Lukavica has interviewed Paolo.

What are your tasks in the SOUG?
I have been an assessor on the board of the SOUG for approximately one year. Following this test year, I will now candidate as a chairman in the upcoming general meeting. On the board, I support Jürgen Vitek with organizing the events in Switzerland, and I promote the regional meetups. Currently, we have a very well-attended meetup in the Bern region that has been established by our sponsor, Edorex. A second regional meetup has been established in the region of Lake Constance. As SOUG, we immediately felt the potential of these events and decided to actively support them. The meetups should, however, live on their own and be convened and organized by the responsible persons. For Zürich, we have also found a sponsor now and we are currently looking for members to set up a regional meetup in the Zürich region.

What is the most fun part for you?
I try to strengthen the thought of community in the SOUG. The SOUG is a small association where you know each other. Due to this, there is a risk that you are not that open to new persons and are rather noticed as a closed group. Furthermore, the SOUG was, up until now, regarded as an association in that can "consume". Meaning "I go to an event, everything is organized, I even get a nice aperitif at the end, what else could you ask for?" Association members were completely in their "comfort zone". But an association lives on the commitment of several members, everyone can contribute, be it a presentation, a new idea, or support for implementation, or only welcoming a new member and including a member in an interesting discussion while having the aperitif. I live this philosophy and I am very happy that this is more and more well-received by the members.

What are the current main topics in the SOUG?
The SOUG has to develop further to satisfy the needs of the members and the market. In the last years the number of members decreased and the average member age increased. You can feel that Oracle as the single topic does not appeal that much to young people anymore compared to roughly 24 years ago when I started in the IT sector. In order to reach new target audiences and to attract new members, we have to open the subject areas around Oracle. Particularly with developers and students, we see great potential. However, in cooperation with our sponsors, we also want to offer new approaches in the field of education and training to our long-term members. By these means, we want to strengthen our partnership with the companies and position the SOUG as a strong partner. That is why we created a new position in the SOUG that is responsible for company support, in addition to member support.

The cloud topic was of major importance for the German DOAG. What role did the cloud have in Switzerland, and what was the SOUG's position?
The role of the cloud becomes more and more significant in Switzerland. We may have started a bit later in Switzerland compared to Germany (mainly due to the problem of geographic data storage) but, today, an increasing number of companies in Switzerland are starting to operate applications in the cloud. Although the majority of DBAs is not strongly affected, the work of a database administrator will change with time. We want to support our members to discuss the topic. What are the technical skills you need to operate a database in the cloud? What will be the new tasks for DBAs in the near-term and long-term? Will we need less engineering skills and more managing skills instead (e.g. will cost control be a new topic for us)? We will increasingly address these topics at our events to support the community and to stay on the pulse of time.

Switzerland takes a leading role in blockchain development. How important is this topic for the SOUG?
Currently, this topic is not that important. We already had some presentations on this topic during our SOUG Days, but they

were more like “What is blockchain in general?”. In our surveys, we also notice that our members are not very interested in this topic, until now. But we keep our ears open and proactively involve speakers on this for our upcoming events.

Do you cooperate with other Oracle User Groups?
With the DOAG and the AOUG, obviously. Regionally, we are connected with the DOAG regional group Lake Constance and we launch meetups with them in this region.

Are there upcoming SOUG events in Switzerland we should not miss?
Yes, on May 21st, there is the half-day SOUG Day Romandy in Lausanne with two parallel tracks and very good speakers beyond the Swiss borders. On May 22nd, there is the full day event SOUG Day in Olten. This time, the location is the University of Applied Sciences of Northwest Switzerland. There will be five parallel tracks and at least one presentation will always be in English. It is not only about technical topics around the Oracle database but also open technologies such as Kafka, graph databases, Docker, microservices and others. Soft skills are also topics for DBAs and managers. We are looking forward to many new speakers presenting for their first time at the SOUG and therefore lowering the average age.

What are the future developments in the SOUG you are already looking forward to?
To any possible development. You can sense the dynamics and the drive for change, currently, but we do not exactly know the direction yet. There are many discussions and that is the most important thing in our opinion: When there is movement, there is life. What delights me most is the slow but increasing active participation of the SOUG members. I hope for great participating commitment of the association members in the next years, and therefore an active co-creation in the events of the association.

Paolo Kreth
Database specialist

Heli Helskyaho

Ambassador's Corner

Dear user group leaders,
I hope you and your user groups are doing well!

EOUC Leaders' meeting in Riga

Our next EOUC leaders' face-to-face meeting will be in Riga, Latvia May 29, combined with the amazing Riga Dev Days 2019. This meeting is 100% organized by user groups for user groups. I am sure this meeting will be very interesting, and if you are not able to attend, we will try to arrange the possibility to attend remotely. If you have any questions/concerns/ideas please do not hesitate to contact Heli and/or Ami.

Oracle OpenWorld 2019

The preparation for OOW 2019 is on. The election for EOUC speakers was conducted and seven speakers were elected to represent EOUC speakers at the Oracle OpenWorld. At this very moment we are not yet allowed to announce the names since Oracle is still in the middle of the approval process.

EOUC ORACLE Support Survey 2019

The EOUC ORACLE Support Survey 2019 is on, led by the German Oracle user group, DOAG. The results of this survey will be presented at the EOUC Leaders' meeting in Riga.

Your Ambassadors:

If you have anything we can help with, please do not hesitate to contact us!

Ami Aharonovich
ilOUG (Israel)
ami@dbaces.com

Heli Helskyaho
OUGF (Finland)
heli@miracleoy.fi

I hope to see many of you in Riga!

Best regards,
Heli

Events

DOAG 2019 Datenbank

June 3-4, 2019
Düsseldorf, Germany
https://datenbank.doag.org/de/home

BGOUG Spring Conference 2019

June 7 - 9, 2019
Borovets, Bulgaria
http://website.bgoug.online/en/events/details/103.html

OUG Scotland

June 12, 2019
Glasgow
https://www.ukoug.org/page/ougscotland2019

Kscope19

June 23-27, 2019
Seattle, Washington, USA
https://kscope19.odtug.com

JD Edwards Summit

June 11-13, 2019
The Omni Interlocken Hotel- Broomfield, Colorado
http://www.questoraclecommunity.org/summit

PeopleSoft RECONNECT

July 16-18, 2019
Hyatt Regency O'Hare- Rosemont, IL
https://www.questoraclecommunity.org/reconnect

JD Edwards INFOCUS

August 20-22, 2019
Sheraton Downtown Denver- Denver, CO
http://www.questoraclecommunity.org/infocus

POUG 2019

September 6-7, 2019
Wroclaw
https://www.poug.org/en/about

Oracle Open World

September 16 - 19, 2019
San Francisco, California
https://www.oracle.com/openworld

HrOUG 2019

October 15-18, 2019
Rovinj, Croatia
http://2019.hroug.hr

DOAG 2019 Conference + Exhibition

November 19 - 22, 2019
Nuremberg, Germany
https://2019.doag.org/en/home

QXW- Quest Experience Week

November 12-15, 2019
Virtual event
https://www.questoraclecommunity.org/qxw

ORAWORLD is a publication of the EOUC — EMEA ORACLE USERGROUP COMMUNITY
The following user groups belong to EOUC:

Angola Oracle User Group, Oracle User Group Armenia, Austrian Oracle User Group, Azerbaijan Oracle User Group, Bulgarian Association of Software Developer, Bulgarian Oracle User Group, Hrvatska udruga Oracle korisnika, Czech Oracle Applications User Group, Danish Oracle User Group, Egypt Oracle Users Group, Oracle User Group Estonia, Oracle User Group Finland, Club Français des Utilisateurs JD Edwards, Association des Utilisateurs Francophones d'Oracle, Club des Utilisateurs PeopleSoft, Oracle User Group Georgia, Deutsche Oracle Anwendergruppe, PeopleSoft Germany, Hungarian Oracle User Group, Israel Oracle User Group, Taranta Valley Oracle User Group, Italian Oracle User Group, Jordan Amman Oracle User Group, Latvian Oracle Users Group, Lithuanian Oracle Users Group, Mauritius Oracle User Group, Oracle Gebruikersclub Holland, Oracle Benelux User Group, Oracle User Group Norway, Polish Oracle Users Group, Oracle Users Group Portugal, Romanian Oracle User Group, Russian Oracle User Group, EBS Finance Special Interest Group Russia, Arab Oracle User Group, Serbia and Montenegro, Slovenian Oracle User Group, South African Oracle User Group, Spanish Oracle User Group, Swedish Oracle User Group, Swiss Oracle User Group, Tajikistan Oracle User Group, Turkey Oracle Users Group, Ukraine Oracle User Group, Middle East Oracle User Group, United Kingdom Oracle User Group, Zimbabwe Oracle User Group.

Editorial board:

Registered office: DOAG Dienstleistungen GmbH
Tempelhofer Weg 64, 12347 Berlin, Germany
www.doag.org,
Director Fried Saacke,
AG Berlin Charlottenburg HRB 95694B,
VAT ID DE240700058
Contact: redaktion@doag.org
Editor-in-chief (ViSdP): Dr. Dietmar Neugebauer

Editorial team:

Ami Aharonovich, Jean-Jacques Camps, Heli Helskyaho, Dr. Dietmar Neugebauer, Ann-Sofie Vikström Often, Andrejs Vorobjovs

Other editors: Lisa Damerow, Mylène Diacquenod, Christian Luda, Sanela Lukavica, Martin Meyer

Other authors are credited by name with their article. They are subject to the terms and conditions for authors: **www.oraworld.org/terms-and-conditions-for-authors**

Graphic design:

Alexander Kermas, DOAG Dienstleistungen GmbH,
Tempelhofer Weg 64, 12347 Berlin, Germany

Article submission:

If you are interested in submitting an article, please e-mail us your article via the online form at **www.oraworld.org**. Deadline for #15 edition: **August 15, 2019**.

Photo credits:

Title: © vectorstory/www.fotolia.com
P. 4: © floralset/www.123rf.com
P. 5: © Comic: www.commitstrip.com
P. 6: © vlasdv/www.fotolia.com
P. 7-9:© vectorstory/www.fotolia.com
P. 10: © pikisuperstar/www.freepik.com
P. 11-14: Icons © freepik/www.freepik.com
P. 15: © Mark Carrel/www.123rf.com
P. 18: © Richard Foote
P. 23: © www.python.org
P. 26: © rawpixel/www.123rf.com
P. 28: © Valerii Matviienko/www.123RF.com
P. 33: © aurielaki/www.123RF.com
P. 39: © skouatroulio/www.123RF.com
P. 43: © Schuchrat Kurbanov/DOAG

Disclaimer:

The ORAWORLD e-magazine reports on events in the Oracle and IT world. The publication covers current topics in the international user group network, as well as news items on products and technologies and their use. The purpose of the magazine is to foster the sharing of knowledge and experience among readers. ORAWORLD is independent of Oracle and does not represent its commercial interests either directly or indirectly.

ORAWORLD is published by DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Germany, legally represented by director Fried Saacke, the nature and purpose of whose business is management of the group, organising events and publishing.

DOAG Deutsche Oracle Anwendergruppe e.V. holds 100 percent of the capital invested in DOAG Dienstleistungen GmbH. DOAG Deutsche Oracle Anwendergruppe e.V. is legally represented by the managing committee; Chair: Stefan Kinnen.