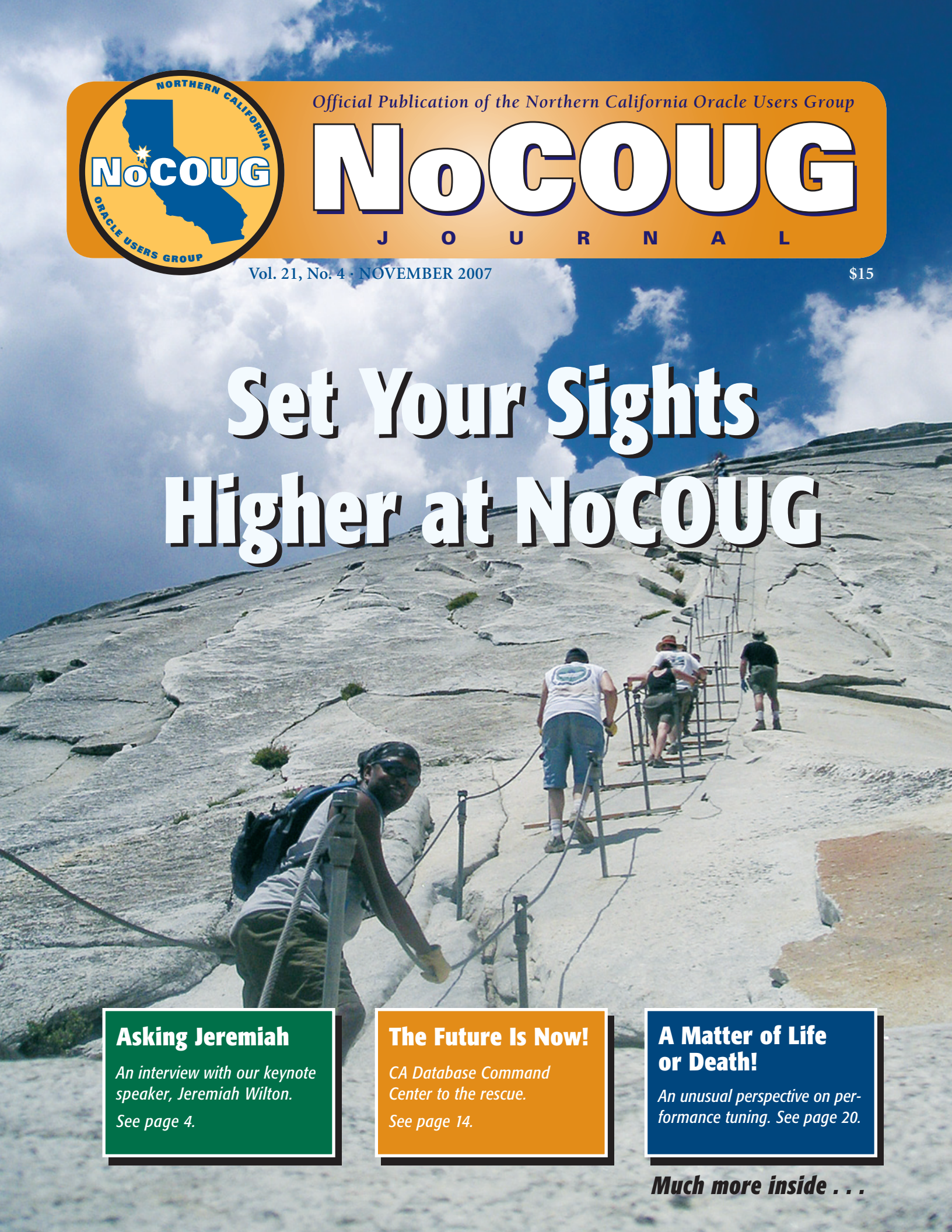
Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

Vol. 21, No. 4 · NOVEMBER 2007 $15

Set Your Sights Higher at NoCOUG



Asking Jeremiah

An interview with our keynote speaker, Jeremiah Wilton.

See page 4.

The Future Is Now!

CA Database Command Center to the rescue.

See page 14.

A Matter of Life or Death!

An unusual perspective on performance tuning. *See page 20.*

Much more inside . . .

Oracle Annoyances for Geeks: Deadlocks

by Iggy Fernandez

Iggy Fernandez

"It's kind of hard to understand the rules. It's like you can't do nothing no more."
—Football star Terrell Owens, after being fined for a flamboyant celebration after scoring a touchdown.

Case Study

I came across a book called *Windows XP Annoyances for Geeks* and wondered why there wasn't a book about Oracle annoyances. Nothing's perfect, Oracle included, and it always helps to know exactly where the deficiencies lie and how to work around them. Perhaps somebody at Oracle Corporation with the power to change things might read the book and the problems would get fixed some day! How about the inability of SQL*Plus to handle blank lines in long SQL statements? Did you know that the SET SQLBLANKLINES ON command forces SQL*Plus to process blank lines correctly?

I would definitely write extensively about deadlocks. They are annoying because Oracle could do a better job of generating the information needed to find out what is causing them; in fact, Oracle contributes to the misinformation about them. They are also annoying because applications don't usually handle them gracefully, perhaps because they are not well understood. The end result is endless nagging by pointy-haired managers and countless hours wasted by database administrators.

I'd need lots and lots of space—space that I don't have—to write all that it is possible to write on deadlocks, so I thought I might try the approach of a quiz today. Don't worry if you can't answer all of the questions; I learned the answer to some of them only when doing the research for this article. Detailed answers are provided after the quiz.

Deadlock Quiz

Here are a few questions to test how well you understand deadlocks. Pick the best answer in each case.

1. Which of the following situations is an example of an Oracle deadlock? Pick all that apply.
   (a) A situation in which a blocking session becomes inactive but does not release its locks.
   (b) A situation in which a blocking session is itself blocked by another session.
   (c) A situation in which two sessions are trying to modify a row of data that has previously been modified by the other.
   (d) A cycle in the "wait-for" diagram.
   (e) A situation in which a session is blocking itself, i.e., "self-deadlock."

2. What kind of situation cannot result in an Oracle deadlock? Pick all that apply.
   (a) A situation involving a single table, because deadlocks cannot occur if everybody is modifying the same table.
   (b) A situation involving a single user, because sessions owned by the same user cannot deadlock with each other.
   (c) A situation involving a single session, because a session cannot deadlock with itself.
   (d) A situation not involving any locking activity whatsoever.

3. Who receives the "ORA-00060: deadlock detected while waiting for resource" error message?
   (a) The participant that is blocking the others.
   (b) Only one of the participants in the deadlock, randomly picked by Oracle.
   (c) All the participants in the deadlock.

4. What is the effect on any session that receives the "ORA-00060: deadlock detected while waiting for resource" error message?
   (a) The statement that it was executing is rolled back.
   (b) Its transaction is rolled back.
   (c) It is terminated.
   (d) It depends on the circumstances.

5. What action must the database administrator take to resolve the deadlock?
   (a) Terminate the session that is blocking the others.
   (b) Terminate any one of the sessions participating in the deadlock.
   (c) Terminate all the sessions participating in the deadlock.
   (d) None of the above.

6. What is the recommended action that a session should take when it receives an ORA-00060 error message:
   (a) Issue the ROLLBACK TO SAVEPOINT command and repeat the statement that it was executing when it received the ORA-00060 error message.
   (b) Repeat its entire transaction; there is no need to issue an explicit ROLLBACK command because

Oracle has already rolled back the entire transaction.
(c) Repeat the entire transaction; the session must first issue an explicit ROLLBACK command because Oracle has not rolled back the transaction.

7. What causes a deadlock in Oracle? Pick all that apply.
   (a) Contention for the same rows of data by multiple sessions.
   (b) Physical database design issues such as unindexed foreign keys.
   (c) Architectural issues such as block-level locking.

8. How does Oracle detect that a deadlock has occurred?
   (a) It checks for deadlocks whenever a session cannot acquire a lock.
   (b) It checks for deadlocks at regular intervals.
   (c) It checks for deadlocks whenever the wait time of a session has exceeded a certain threshold.
   (d) It checks for deadlocks whenever the number of waiting sessions has exceeded a certain threshold.

9. How can you obtain a list of all the rows locked by a transaction? Pick all that apply.
   (a) By querying V$LOCK in order to identify the locked rows.
   (b) By querying V$LOCKED_OBJECT in order to identify the locked rows.
   (c) By querying DBA_LOCK in order to identify the locked rows.
   (d) None of the above.

10. A transaction issues a SAVEPOINT command and then locks a data row that is subsequently required by another transaction. What happens if the blocking transaction issues a ROLLBACK TO SAVEPOINT command?
   (a) The blocking transaction reverses the changes made to the data row but does not release its lock on the row.
   (b) The blocking transaction reverses the changes made to the data row and releases its lock on the row; the blocked transaction is automatically unblocked.
   (c) The blocking transaction reverses the changes made to the data row and releases its lock on the row, but the blocked transaction remains blocked.

Answers

1. The correct answers are (c), (d), and (e). Deadlock is often confused with regular blocking situations, but a situation in which a blocking session becomes inactive without releasing its locks is not a deadlock. Neither is a situation in which a blocking session is itself locked by another session or an even more complicated logjam involving multiple sessions. The technical definition of a deadlock is a cycle in the "wait-for" diagram (e.g., A waiting for B, B waiting for C, C waiting for A). A situation in which two sessions are each trying to modify a row of data that has previously been modified by the other is obviously an example of a deadlock. Less obvious examples are situations in which a deadlock arises even though sessions do not require the same data rows or when a session

Es-Cue-El Challenge

Be It Known By These Presents that the great Wizard of Odds at Hogwash School of Es-Cue-El needs your help in solving the riddle of the ancient jade icosahedron found in the secret chamber of mystery. A great tournament has been organized and all practitioners of the secret art of Es-Cue-El have been invited to demonstrate their prowess.

The best entry will win the Wooden Pretzel award, instituted in honor of another great wizard, who famously observed that "some people can perform seeming miracles with straight Es-Cue-El, but the statements end up looking like pretzels created by somebody who is experimenting with hallucinogens." As if that singular honor is not enough, a magnificent iPod Shuffle will also be bestowed on the champion.

Unsolvable Riddle

An ancient twenty-sided die (icosahedron) was discovered in the secret chamber of mystery at Hogwash School of Es-Cue-El. A mysterious symbol was inscribed on each face of the die. The great Wizard of Odds discovered that each symbol represented a number, and he also discovered that the die was biased—some numbers were more likely to be displayed than others when the die was used in a game of chance. The great wizard then recorded all this information in tabular fashion as described below.

```
SQL> describe die
 Name            Null?        Type
 --------------  -----------  --------------
 FACE_ID         NOT NULL     NUMBER(2)
 FACE_VALUE      NOT NULL     NUMBER(2)
 PROBABILITY     NOT NULL     NUMBER(10,10)
```

*The great wizard now implores you to create an Es-Cue-El spell that displays the probabilities of obtaining various sums when the die is thrown N times in succession.**

May the best wizard win!

**N is a "substitution variable" or "bind variable."*

deadlocks with itself. Examples are provided in the references listed in the Further Reading section.

2. The correct answer is (d). The only listed situation that cannot result in a deadlock is a situation that does not involved any locking activity at all. Deadlocks can certainly occur if everybody is modifying the same table. Further, the fact that two sessions are owned by the same user does not prevent them from blocking each other and, when two sessions block each other, a deadlock can result. It is not, however, obvious that a session can deadlock with *itself*. Documented examples include parallel query processing, autonomous transactions, and recursive SQL.

3. The correct answer is (b). Only one of the participants in the deadlock receives an error message. This is the one that Oracle arbitrarily picks as a victim in order to resolve the deadlock. For example, if A is waiting for B, B is waiting for C, and C is waiting for A, then Oracle may arbitrarily pick B as the victim; B will then receive the error message "ORA-00060: deadlock detected while waiting for resource." The other sessions do not receive an error message. Option (a) is not the best answer because all the participants in the deadlock are blocking someone else.

4. The correct answer is (a). Oracle performs statement-level rollback on a session that receives the "ORA-00060: deadlock detected while waiting for resource" error message.

5. The correct answer is (d). The database administrator does not have to resolve a deadlock because Oracle automatically does all that is necessary—let no one tell you otherwise! However, if deadlocks happen frequently and the applications do not handle them gracefully, the database administrator might well be drawn into the investigation of their cause.

6. The correct answer is (c). Oracle resolves a deadlock by performing a statement-level rollback operation on the victim but all the modifications made by the victim in previous statements, as well as all the locks previously acquired by it in the course of those statements, are still in effect.

Let's suppose that A is waiting for B, B is waiting for C, and C is waiting for A: a deadlock situation. If Oracle picks C as a victim and performs statement-level rollback, then B will still be unable to proceed. If C submits its last statement again, then the deadlock situation will be created again and C may be picked as the victim again. In an automated situation, the correct thing for C to do is to issue the ROLLBACK command before retrying its entire transaction.

In the following example, each session receiving a deadlock error retries without rolling back the rest of its work. Deadlocks happen repeatedly until one session finally decides to roll back its work and the other sessions are then able to complete their work. The session that rolled back then resubmits all its statements for processing and successfully completes its work.

```
SESSION A> UPDATE dummy SET status = 1 WHERE dummy = 1;
SESSION A> 1 row updated.

SESSION B> UPDATE dummy SET status = 1 WHERE dummy = 2;
SESSION B> 1 row updated.

SESSION C> UPDATE dummy SET status = 1 WHERE dummy = 3;
SESSION C> 1 row updated.

SESSION A> -- A tries to update the row locked by B
SESSION A> UPDATE dummy SET status = 1 WHERE dummy = 2;

SESSION B> -- B tries to update the row locked by C
SESSION B> UPDATE dummy SET status = 1 WHERE dummy = 3;

SESSION C> -- C tries to update the row locked by A
SESSION C> UPDATE dummy SET status = 1 WHERE dummy = 1;

SESSION A> ORA-00060: deadlock detected while waiting for resource

SESSION A> -- A retries without issuing a rollback command
SESSION A> UPDATE dummy SET status = 1 WHERE dummy = 2;

SESSION B> ORA-00060: deadlock detected while waiting for resource

SESSION B> -- B retries without issuing a ROLLBACK command
SESSION B> UPDATE dummy SET status = 1 WHERE dummy = 3;

SESSION C> ORA-00060: deadlock detected while waiting for resource

SESSION C> -- C retries without issuing a ROLLBACK command
SESSION C> UPDATE dummy SET status = 1 WHERE dummy = 1;

SESSION A> ORA-00060: deadlock detected while waiting for resource

SESSION A> -- A decides to issue a ROLLLBACK
SESSION A> ROLLBACK;
SESSION A> Rollback complete.

SESSION C> -- C is no longer blocked by A
SESSION C> 1 row updated.
SESSION C> COMMIT;
SESSION C> Commit complete.

SESSION B> -- B is no longer blocked by C
SESSION B> 1 row updated.
SESSION B> COMMIT;
SESSION B> Commit complete.

SESSION A> -- A retries its entire transaction and is successful
SESSION A> UPDATE dummy SET status = 1 WHERE dummy = 1;
SESSION A> 1 row updated.
SESSION A> UPDATE dummy SET status = 1 WHERE dummy = 2;
SESSION A> 1 row updated.
SESSION A> COMMIT;
SESSION A> Commit complete
```

7. The correct answers are (a), (b), and (c). Oracle contributes to the misinformation about deadlocks by always using the following language in the trace file where the deadlock is recorded: "The following deadlock is not an ORACLE error. It is a deadlock due to user error in the design of an application or from issuing incorrect ad-hoc SQL." In fact, deadlocks can result from no fault of the application.

In particular, physical database design issues such as un-

Oracle contributes to the misinformation about deadlocks by always using the following language in the trace file where the deadlock is recorded: "The following deadlock is not an ORACLE error. It is a deadlock due to user error in the design of an application or from issuing incorrect ad-hoc SQL."

indexed foreign keys and indexing choices such as bitmap indexes can make the database prone to deadlock. Also, block-level locking can be encountered in an Oracle database in the guise of "ITL waits." The details can be found in the references listed in the Further Reading section.[1]

> ***"The database administrator does not have to resolve a deadlock because Oracle automatically does all that is necessary—let no one tell you otherwise!"***

8. The correct answer is (b). It is computationally too expensive for Oracle to check whether a deadlock has occurred every time a transaction is blocked. Instead, Oracle invokes its deadlock detection algorithm at regular intervals.

9. The correct answer is (d). When a deadlock occurs, it would be helpful if Oracle would list all the locks held by the participants. However, row locks are not held in memory but in the data blocks themselves. The advantage of this approach is that Oracle does not have to place artificial limits on the number of rows that can be locked by one transaction; the disadvantage is that it is not easy to produce a list of rows locked by a transaction.

10. The correct answer is (c). When the blocking transaction requested a ROLLBACK TO SAVEPOINT, it released the row lock acquired since the save point, but the blocked transaction is *still blocked*. This is because the blocked transaction is waiting for the "TX enqueue" (visible in V$LOCK) which represents the blocking transaction, not any particular row lock.

Further Reading

Mark Bobak's paper "Understanding and Interpreting Deadlocks or What to Do When You Encounter ORA-00060" is a good introduction to the topic. It is available for download at the Oak Table website **www.oaktable.net/userFiles.jsp**.

The following Metalink documents are also good sources of information on deadlocks: Metalink document 164661.1: ORA-60/Deadlocks Most Common Causes; Metalink document 62365.1: What to do with "ORA-60 Deadlock Detected" Errors. If you'd like to discuss this topic in a public forum, you can do so at **oracleannoyances.blogspot.com**. ▲

Iggy Fernandez is a database administrator for Verizon. Please send any comments or corrections to **iggy_fernandez@hotmail.com**.

[1] My colleague Junho Whang points out that even ordinary indexing choices can increase the probability of deadlocks. If appropriate indexes are not available or the query optimizer does not use them appropriately, queries may hold locks longer than necessary, which increases the probability of deadlocks. The greater the locking activity, the longer the locks are held, and the wider the scope of the locks; the higher the chances are of a deadlock occurring. Batch jobs, long transactions, table locks, etc., increase the chances of deadlocks.