

1. Record Nr.	UNINA990006496130403321
Autore	Shembesh, Ali Muhammad
Titolo	The analysis of Libya's foreign policy : 1962-1973 : A study of the impact of environmental and leadership factors / AliMuhammad Shembesh
Pubbl/distr/stampa	Ann Arbor : University Microfilms Int., 1975
Descrizione fisica	VIII, 262 p. ; 20 cm
Disciplina	327.61
Locazione	FSPBC
Collocazione	XIV E 2306
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Dissertazione accademica mcrf.

2. Record Nr.	UNINA9910581796403321
Autore	Cannito, Maddalena
Titolo	Fare spazio alla paternità : essere padri in Italia tra nuovi modelli di welfare lavoro e maschilità / Maddalena Cannito
Pubbl/distr/stampa	Bologna, : il Mulino, 2022
ISBN	9788815295996
Descrizione fisica	254 p. ; 21 cm
Collana	Culture, politica e società
Disciplina	306.8742
Locazione	BFS FSPBC
Collocazione	306.8742 CAN 1 SOC 9
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia

3. Record Nr.	UNINA9910300467003321
Autore	Kyte Thomas
Titolo	Oracle Database Transactions and Locking Revealed / / by Thomas Kyte, Darl Kuhn
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2014
ISBN	9781484207604 1484207602
Edizione	[1st ed. 2014.]
Descrizione fisica	1 online resource (179 p.)
Collana	Expert's Voice in Oracle
Disciplina	004 005.11 005.74
Soggetti	Database management Computer programming Database Management Programming Techniques
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"Independent Oracle Users Group"--Cover. "Building Successful Concurrency"--Cover. Includes index.
Nota di contenuto	Contents at a Glance; Introduction; Chapter 1: Getting Started; Background; Locking; Concurrency Control; Multiversioning; Transactions; Redo and Undo; Summary; Chapter 2: Locking and Issues; What Are Locks?; Lost Updates; Pessimistic Locking; Optimistic Locking; Optimistic Locking Using a Version Column; Optimistic Locking Using a Checksum; Optimistic or Pessimistic Locking?; Blocking; Blocked Inserts; Blocked Merges, Updates, and Deletes; Deadlocks; Lock Escalation; Summary; Chapter 3: Lock Types; DML Locks; TX (Transaction) Locks; TM (DML Enqueue) Locks; DDL Locks; Latches Latch " Spinning "Measuring the Cost of Latching a Shared Resource; Setting Up for the Test; Without Bind Variables; With Bind Variables; Performance/Scalability Comparison; Mutexes; Manual Locking and User-Defined Locks; Manual Locking; Creating Your Own Locks; Summary; Chapter 4: Concurrency and Multiversioning; What Are Concurrency Controls?; Transaction Isolation Levels; READ

UNCOMMITTED; READ COMMITTED; REPEATABLE READ; Getting a Consistent Answer; Lost Updates: Another Portability Issue; SERIALIZABLE; READ ONLY; Implications of Multiversion Read Consistency

A Common Data Warehousing Technique That FailsAn Explanation for Higher Than Expected I/O on Hot Tables; Write Consistency; Consistent Reads and Current Reads; Seeing a Restart; Why Is a Restart Important to Us?; Summary; Chapter 5: Transactions; Transaction Control Statements; Atomicity; Statement-Level Atomicity; Procedure-Level Atomicity; Transaction-Level Atomicity; DDL and Atomicity; Durability; WRITE Extensions to COMMIT; COMMITS in a Nondistributed PL/SQL Block; Integrity Constraints and Transactions; IMMEDIATE Constraints; DEFERRABLE Constraints and Cascading Updates

Bad Transaction HabitsCommitting in a Loop; Performance Implications; Snapshot Too Old Error; Restartable Processes Require Complex Logic; Using Autocommit; Distributed Transactions; Autonomous Transactions; How Autonomous Transactions Work; When to Use Autonomous Transactions; Summary; Chapter 6: Redo and Undo; What Is Redo?; What Is Undo?; How Redo and Undo Work Together; Example INSERT-UPDATE-DELETE-COMMIT Scenario; The INSERT; Hypothetical Scenario: The System Crashes Right Now; Hypothetical Scenario: The Buffer Cache Fills Up Right Now; The UPDATE Hypothetical Scenario: The System Crashes Right NowHypothetical Scenario: The Application Rolls Back the Transaction; The DELETE; The COMMIT; Commit and Rollback Processing; What Does a COMMIT Do?; What Does a ROLLBACK Do?; Summary; Chapter 7: Investigating Redo; Measuring Redo; Can I Turn Off Redo Log Generation?; Setting NOLOGGING in SQL; Setting NOLOGGING on an Index; NOLOGGING Wrap-up; Why Can't I Allocate a New Log ?; Block Cleanout; Log Contention; Temporary Tables and Redo/Undo; Prior to 12c; Starting with 12c; Summary; Chapter 8: Investigating Undo What Generates the Most and Least Undo?

Sommario/riassunto

Oracle Database Transactions and Locking Revealed provides much-needed information for building scalable, high-concurrency applications and deploy them against the Oracle Database. Read this short, 150-page book that is adapted from Expert Oracle Database Architecture to gain a solid and accurate understanding of how locking and concurrency are dealt with by Oracle Database. Also learn how the Oracle Database architecture accommodates user transactions, and how you can write code to mesh with how Oracle Database is designed to operate. Good transaction design is an important facet of highly-concurrent applications that are run by hundreds, even thousands of users who are all executing transactions at the same time. Transaction design in turn relies upon a good understanding of how the underlying database platform manages of the locking of resources so as to prevent access conflicts and data loss that might otherwise result from concurrent access to data in the database. Oracle Database Transactions and Locking Revealed covers in detail the various lock types, and also different locking schemes such as pessimistic and optimistic locking. Then you'll learn about transaction isolation and multiversion concurrency, and how the various lock types support Oracle Database's transactional features. You'll learn some good tips for transaction design, as well as some bad practices and habits to avoid. Coverage is also given to redo and undo, and their role in concurrency. This is an important book that anyone developing highly-concurrent applications will want to have handy on their shelf.
