**Oracle 19c**

**-**

**Performance Tuning Applicativo**

**e Tuning dell’SQL**

Distribuito tramite il sito www.manualioracle.it

Copyright © 2024 Assi Loris

Qualsiasi abuso sarà perseguito e punito secondo i termini di legge.

Version: 1.8

Alcuni termini usati sono trademarks registrati dei rispettivi proprietari.

**Sommario**

Introduzione al Manuale 5

Cap 1 - Introduzione al Performance Tuning 6

1.1 – Strutture di memoria Oracle 6

1.2 – Primi Concetti sul Tuning Oracle 7

1.3 – DWH, OLAP, OLTP e concetti di Normalizzazione 8

1.4 – Approfondimenti su Data Warehouse (DWH) e Data mart 9

1.5 – Performance Tuning Views 10

1.6 – Approccio DTAP nei database 12

Cap 2 - Introduzione all’ SQL Tuning 14

2.1 – Introduzione all’Explain Plan 14

2.2 – Il package DBMS\_XPLAN 14

2.3 – L’Autotrace 16

2.4 – Explain Plan and Autotrace Exceptions 18

Cap 3 - Application Tracing Tools 19

3.1 – Event Tracing e User Trace File 19

3.2 – Trcsess Utility 21

3.3 – Il TKPROF : Introduzione 21

3.4 – Esempi di uso del TKPROF 23

3.5 – TKPROF : il Sort 24

3.6 – TKPROF : il Result File 25

Cap 4 - Il Query Optimizer 27

4.1 – Optimizer e Optimizer Operations 27

4.2 – Componenti dell’Ottimizzatore 28

4.3 – Gli Hints 30

4.4 – Esempi di uso degli Hints 30

4.5 – Optimizer Goal 31

4.6 – Parent and Child cursors 32

4.7 – Il flusso di Esecuzione di un SQL 33

4.8 – Hard Parse e Soft Parse 34

Cap 5 - Gli Indici 35

5.1 – Introduzione 35

5.2 – Indici B-tree 35

5.3 – Indici Function-based e Indici Bitmap 36

5.4 – Indici Reverse Key (RKI) 37

5.5 – Rebuild e Coalesce degli Indici 38

5.6 – Index monitoring 40

5.7 – Create Invisible Indexes 42

5.8 – Altre feature riguardanti gli Indici 43

Cap 6 - Raccolta Statistiche 45

6.1 – Optimizer Statistics e initialization parameter OPTIMIZER\_DYNAMIC\_SAMPLING 45

6.2 – Gli istogrammi 45

6.3 – DBMS\_STATS 47

6.4 – Vedere le statistiche di Tabelle e Indici 48

6.5 – Stale statistics 51

6.6 – STALE\_PERCENT 51

6.7 – Altre features sulla Raccolta delle Statistiche 52

6.8 – Incremental Statistics 53

6.9 – ESTIMATE\_PERCENT e AUTO\_SAMPLE\_SIZE 54

6.10 – Pending Statistics 54

6.11 – Multicolumn Statistics in Oracle 56

6.12 – Altre Features relative alla raccolta delle Statistiche 57

Cap 7 - Altre feature relative al SQL Tuning 60

7.1 – Materialized Views 60

7.2 – Gestione e Tuning delle Materialized view 61

7.3 – Literal e Bind Variable 62

7.4 – Initialization parameter CURSOR\_SHARING 63

7.5 – Adaptive Cursor Sharing (ACS) 64

7.6 – Adaptive Execution Plans 65

Cap 8 - Oracle Partitioning 68

8.1 – Il Partizionamento 68

8.2 – Partitioning methods 69

8.3 – Range Partitioning 70

8.4 – DML operations nelle tabelle partizionate 71

8.5 – List Partitioning 73

8.6 – Hash Partitioning 74

8.7 – Interval Partitioning 74

8.8 – System Partitioning 77

8.9 – Reference Partitioning 78

8.10 – Virtual column-based Partitioning 79

Cap 9 - Partitioning: Altre considerazioni 81

9.1 – System Views riguardanti il Partizionamento 81

9.2 – Subpartitioning 82

9.3 – ALTER TABLE di tabelle partizionate 83

9.4 – Partition Pruning 84

9.5 – Indici nelle tabelle Partizionate 85

9.6 – Local Partitioned index 86

9.7 – Global Partitioned index 87

9.8 – Miscellaneous relativi agli indici partizionati 87

9.9 – Miscellaneous relativi alle tabelle partizionate 89

Cap 10 - Parallelismo in Oracle 91

10.1 – Introduzione 91

10.2 – AutoDOP, parallel\_degree\_policy e parallel\_min\_time\_threshold 92

10.3 – Limitare i Parallel Process 93

10.4 – Riassunto dei parametri associati al Parallelismo 95

Cap 11 - SQL Tuning : Come interpretare i Piani di Accesso 97

11.1 – Piano di accesso di query su una tabella non partizionata (senza WHERE condition) 97

11.2 – Piani di accesso di query su una tabella non partizionata (con WHERE condition) 98

11.3 – Altri piani di accesso su una tabella non partizionata 101

11.4 – Piani di accesso di query su una tabella partizionata (non “composite partitioned”) 102

11.5 – Piani di accesso di query su una tabella sotto-partizionata 106

11.6 – Piani di accesso su Query con Join 109

Cap 12 - AWR e Statspack 111

12.1 – Collecting Performance Statistics tramite l’AWR 111

12.2 – Come usare l’AWR 112

12.3 – La vista DBA\_HIST\_SNAPSHOT 113

12.4 – Gli AWR Reports 114

12.5 – Gli AWR baseline 115

12.6 – Gli AWR baseline template 117

12.7 – Settare le metric Thresholds per le AWR baseline 118

12.8 – Come usare STATSPACK 118

Cap 13 - ADDM 120

13.1 – Diagnosing Performance Statistics: ADDM 120

13.2 – Come usare l’ ADDM 121

13.3 – Findings e DBMS\_ADDM 121

13.4 – Le directives dell’ADDM 123

13.5 – Emergency Monitoring e real-time ADDM: Introduzione 124

13.6 – Emergency Monitoring 125

13.7 – Real-Time ADDM 127

Cap 14 - ASH e Real Time SQL Monitoring 128

14.1 – L’Active Session History (ASH) 128

14.2 – ASH report 128

14.3 – Real Time SQL Monitoring 129

14.4 – Viste associate al SQL Monitoring 130

14.5 – Procedure, Hints e Parametri riguardanti Real-time SQL Monitoring 131

Cap 15 - Space Management e Server Alerts 134

15.1 – Segment Shrink 134

15.2 – Segment Advisor 135

15.3 – Analizzare una tabella e un LOB usando il Segment Advisor 136

15.4 – Introduzione alla reorg delle tabelle 138

15.5 – Server-generated alerts 139

Cap 16 - Cambiare o Fissare il Piano di Accesso di un SQL statement 141

16.1 – Introduzione 141

16.2 – SQL Tuning Set (STS) 141

16.3 – SQL Tuning Advisor (STA) 142

16.4 – Automatic SQL Tuning task 144

16.5 – SQL Profile 146

16.6 – SQL Plan Management (SPM) 148

16.7 – Come Creare ed Eliminare un SQL Plan Baseline 150

16.8 – I SQL Plan Attributes 152

Cap 17 - Comandi SQL Avanzati 154

17.1 – With clause 154

17.2 – Partitioned Outer Join 155

17.3 – Regular Expressions 156

17.4 – La MODEL clause 157

17.5 – Case-Insensitive and Accent-Insensitive Queries 158

17.6 – La WIDTH\_BUCKET function 159

17.7 – Introduzione al MERGE SQL statement 160

17.8 – Row limiting clause 161

17.9 – Miscellaneous relative ai SQL statement 162

Cap 18 - Miscellaneous 165

18.1 – Automatic Big Table Caching 165

18.2 – In-Memory (IM) column store 165

18.3 – Nuove Feature Oracle 19c 166

* 1. Introduzione al Manuale

 **Contenuto**

Il presente manuale tratta gli argomenti applicativi e di sviluppo (SQL e PL/SQL) del *Performance & Tuning* nel database Oracle 19c.

 **Audience**

Il presente manuale è rivolto a chiunque voglia avere una conoscenza specifica sulla *Performance & Tuning* applicativa, avendo già una conoscenza base dell’architettura di *Oracle.*

Dunque affronta:

 - le basi relative al disegno del db

 - le tecniche per velocizzare un codice SQL

 - le tecniche per velocizzare un workload Oracle (SQL e PL/SQL)

 **Particolarità**

Il presente manuale è in italiano ma molti termini tecnici in esso contenuti sono in lingua inglese.

Abbiamo fatto tale scelta perché spesso a lavoro tali termini vengono detti in inglese e perché riteniamo più utile che i lettori del manuale conoscano tali termini nella lingua usata nella documentazione ufficiale Oracle.

Molti argomenti sono schematizzati al fine di fornire una comprensione ed una memorizzazione superiore.

A volte quando nel manuale si parla di una feature particolare, citiamo anche la versione Oracle in cui tale feature è stata creata.

Alcune abbreviazioni usate:

 !!! : Attenzione

 DOP : Degree Of Parallelism

 FTS : Full Table Scan

HWM : High Water Mark

OS o O.S. : Sistema Operativo

 db : database

 init parameter : initialization parameter

 loccare : mettere un lock

 stats : statistiche

 startare il database : fare lo startup del database

 tunare : fare il tuning

 **Principali Versioni**

- 18.04.2024 : version 1.0

- 25.04.2024 : version 1.2 varie modifiche e rilascio al pubblico

- 31.05.2024 : version 1.6 approfondito l’argomento *partitioning* e varie altre modifiche

**Disclaimer**

Non si fornisce alcuna garanzia relativamente al fatto che il presente documento sia privo di errori. Non si assume nessuna responsabilità sugli eventuali errori o danni derivanti dall’uso delle informazioni qui contenute.

* 1. Introduzione al Performance Tuning
		1. 1.1 – Strutture di memoria Oracle

Qui vediamo una veloce review riguardante le strutture di memoria Oracle.

Ulteriori dettagli sono nei seguenti manuali:

 Oracle 19c: Completo e New Features

Oracle 19c Performance Tuning (sistemistico)

distribuiti tramite il sito

 [www.manualioracle.it](http://www.manualioracle.it)

Le strutture di memoria sono allocate dall’istanza Oracle quando l’istanza stessa viene tirata su e vengono deallocate quando si fa lo shutdown.

Le 2 maggiori strutture di memoria sono:

* **SGA :**

*System Global Area* (o anche *Shared Global Area*)

* **PGA :**

*Program Global Area* (o *Private Global Area*)



La **System Global Area (SGA)** è un area di memoria condivisa; ossia tutti gli utenti del DB condividono le informazioni in essa contenute.

Essa contiene le aree indicate nel disegno sopra e descritte in dettaglio sotto.

.................

.................

.................

* 1. Introduzione all’ SQL Tuning
		1. 2.1 – Introduzione all’Explain Plan

Un tuning appropriato dell’applicazione dipende molto dal tipo di sistema in cui si è: come visto, gli *OnLine Transactional Processing systems* (OLTP) hanno differenti SQL e design considerations rispetto ai *data warehouse* (DWH).

Comunque senza preoccuparti del tipo di sistema, è importante misurare le performance degli SQL in esecuzione e capire se si deve intervenire in qualche modo.

Fra i metodi più usati per misurare le performance degli *SQL statement* c’è l’*Explain Plan*, che può essere vistotramite il DBMS\_XPLAN, l’AUTOTRACE, il TKPROF utility o l’AWR SQL report.

L’*explain plan* è un modo per vedere gli *execution plan* (piani di esecuzione) scelti dall’optimizer per SELECT, UPDATE, INSERT e DELETE statements.

Un *execution plan* è la sequenza di operazioni che il db fa per fare girare un *SQL statement*.

Guardando l’*execution plan* puoi cercare di migliorare le performance di un *SQL statement* lento*.*

**Es.:**

Vediamo la seguente query:

SELECT \*

FROM emp, dept

WHERE emp.deptno = dept.deptno;

Ecco una parte del suo piano d'esecuzione che puoi vedere usando i metodi elencati sopra:

Execution Plan

---------------------------------

MERGE JOIN

 SORT JOIN

 TABLE ACCESS (FULL) OF 'DEPT'

 SORT JOIN

 TABLE ACCESS (FULL) OF 'EMP'

Dopo vedremo nel dettaglio come si leggono i piani di esecuzione.

.................

.................

.................

* 1. Application Tracing Tools
		1. 3.1 – Event Tracing e User Trace File

I *trace file* possono avere informazioni utili per il tuning e per il troubleshooting del sistema.

 **EVENT TRACING**

In Oracle esiste una opzione che puoi configurare per permettere di tracciare informazioni dettagliate per particolari *db events*.

Un **db event** è una specifica azione o attività che avviene nel db. A volte, il supporto Oracle può chiederti di attivare tale *trace* per risolvere problemi di performance tuning o per troubleshooting.

Il trace di questi eventi è normalmente fatto ad uno di questi livelli:

* Sessione.

Es.:

ALTER SESSION

SET events '10046 trace name context forever, level 12';

* Istanza, aggiungendo istruzioni nell’*init.ora* o nell’*spfile*.

Attenzione: in questo caso, rischi di produrre tantissimi *trace*.

Es. di *init entry*:

 EVENT="10046 trace name context forever, level 12"

.................

.................

.................

* 1. Il Query Optimizer
		1. 4.1 – Optimizer e Optimizer Operations

Il *Query Optimizer* (chiamato anche *optimizer* o *ottimizzatore*) è un software interno al database Oracle il cui scopo è quello di determinare il modo più efficiente di eseguire un *SQL statement* (spesso una SELECT).

Quando lanci un *SQL statement*, l’ottimizzatore sceglierà come farlo eseguire velocemente: lo fa trovando il piano d’accesso con il costo minore.

Il **costo** è un numero (senza una vera unità di misura) che indica quante risorse l’ottimizzatore si aspetta di utilizzare per eseguire un *SQL statement* seguendo un certo piano di accesso.

In questo paragrafo dunque vediamo come l’ *Optimizer* sceglie uno specifico piano d’accesso per eseguire un *SQL statement*.

 **Optimizer Operations**

Un *SQL statement* può essere eseguito accedendo ai dati in molti modi, come *Table Access Full (TAF)*, *index scan*, ecc…

Quando lanci un *SQL statement*, l’ottimizzatore fa i seguenti steps (qui semplificati):

1. Genera un set di potenziali *execution plans* per eseguire il *SQL statement*
2. Stima il costo di ogni *execution plan* basandosi sulle statistiche presenti nel dizionario dati.

In particolare, fra le statistiche l’ottimizzatore valuterà vari indicatori (numero di righe, distribuzione dei dati, come sono fatti gli indici, ecc..) degli oggetti referenziati.

1. Compara i *plan* e sceglie quello con costo più basso.

Es.:

.................

.................

.................

* 1. Gli Indici
		1. 5.1 – Introduzione

 **Minimizzare gli I/O usando gli Indici**

L’appropriato uso degli indici è probabilmente la cosa più importante (e relativamente semplice) per il tuning applicativo in quanto possono fare evitare tanti I/O non necessari, i quali contribuiscono pesantemente nel *response time* di una query.

In Oracle sono disponibili diversi tipi di indice.

Qui li elenchiamo mettendoli in ordine da quello che reputiamo essere il più comune a quello che è più raro da usare:

* *B-Tree index*
* *Function-based index*
* *Bitmap index*
* *Reverse Key Index (RKI)*
* *Index Organized Table (IOT), Index Clusters* e *Hash Clusters*. Questi 3 oggetti sono poco usati quindi in questo manuale non verranno trattati

L’unica configurazione Oracle in cui l’uso degli indici è molto meno importante è quando si lavora su *Oracle exadata* in quanto, per sua natura, effettua veloci *Table Access Full* anche su tabelle di dimensioni rilevanti.

Le viste di dizionario più importanti che riguardano gli indici sono:

- DBA\_INDEXES :

 Mostra tutti gli indici presenti nel db

.................

.................

.................