

# ***KTest***

更に上のクオリティ 更に上のサービス



## **問題集**

<http://www.ktest.jp>

1年で無料進級することに提供する

**Exam** : **1Z0-054**

**Title** : Oracle Database 11g:  
Performance Tuning

**Version** : DEMO

1. SQL パフォーマンス アナライザ (SPA) を実行した後、いくつかは、SPA の出力に低下した SQL 文を確認します。あなたはこれらの低下した SQL 文に対して示唆するという 2 つのアクションを識別します。(2つ選択してください。)

- A. SQL アクセス アドバイザを実行する
- B. SQL 計画ベースラインに追加する
- C. SQL チューニング アドバイザに提出する
- D. 自動データベース診断モニター (ADDM) を実行している

Answer: BC

2. exhibit1 SQL コマンドとパラメータ設定のシリーズを調べるために展示を見る。

SQL> SHOW PARAMETER OPTIMIZER

NAME	TYPE	VALUE
optimizer_capture_sql_plan_baselines	boolean	TRUE
optimizer_dynamic_sampling	integer	2
optimizer_features_enable	string	11.1.0.6
optimizer_index_caching	integer	0
optimizer_index_cost_adj	integer	100
optimizer_mode	string	ALL_ROWS
optimizer_secure_view_merging	boolean	TRUE
optimizer_use_invisible_indexes	boolean	FALSE
optimizer_use_pending_statistics	boolean	FALSE
optimizer_use_sql_plan_baselines	boolean	TRUE

```
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
SQL> ALTER SESSION SET OPTIMIZER_MODE=FIRST_ROWS;
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
```

SQL で使用可能な計画を検討する展示 exhibit2 基準計画を表示します。

Select Name	SQL Text	Enabled	Accepted	Fixed	Auto Purge	Created	Last Modified
<input type="checkbox"/> SYS_SQL_PLAN_89447021cf314e9e	select * from hr.employees where job_id='CLERK'	YES	YES	NO	YES	Jul 20, 2008 7:02:30 PM	Jul 20, 2008 7:16:48 PM
<input type="checkbox"/> SYS_SQL_PLAN_894470210572d2e8	select * from hr.employees where job_id='CLERK'	YES	NO	NO	YES	Jul 20, 2008 7:20:45 PM	Jul 20, 2008 7:20:45 PM
<input type="checkbox"/> SYS_SQL_PLAN_7ed8568135b3cdca	SELECT NAME NAME COL PLUS SHOW PARAM,DECODE (TYPE,1...	YES	YES	NO	YES	Jul 21, 2008 2:40:44 PM	Jul 21, 2008 2:40:44 PM
<input type="checkbox"/> SYS_SQL_PLAN_4698b35ddf463620	select * from table(dbms_xplan.display (null,null,...	YES	YES	NO	YES	Jul 20, 2008 7:04:22 PM	Jul 20, 2008 7:04:22 PM
<input type="checkbox"/> SYS_SQL_PLAN_467a776254bc8843	select * from sh.sales where quantity_sold > 40 cr...	YES	YES	NO	YES	Jul 21, 2008 2:25:42 PM	Jul 21, 2008 2:25:42 PM
<input type="checkbox"/> SYS_SQL_PLAN_467a776211df68d0	select * from sh.sales where quantity_sold > 40 cr...	YES	NO	YES	YES	Jul 21, 2008 2:41:22 PM	Jul 21, 2008 2:41:56 PM

最初の計画は、(赤) OPTIMIZER\_MODE を ALL\_ROWS に設定されており、OPTIMIZER\_MODE を FIRST\_ROWS に設定されているとき、2 番目の計画は (青) が作成されたときに作成されます。

OPTIMIZER\_MODE の値が FIRST\_ROWS に設定されている場合、SQL クエリを再度 exhibit1 で実行された場合、どの SQL 計画ベースラインを使用するでしょうか？

- A. 第二の計画、それは固定されたスケジュールであるため
- B. 最初の計画は、承認の計画であるため
- C. 第二の計画、それは first\_row を、最新のファッションに生成されたスケジュールであるため

D. 新しい計画、first\_row をモードの第二計画が承認の計画ではないので

**Answer: B**

3. あなたは会社のために DBA として働くとは、そのオンライン トランザクション処理 (OLTP) システムのいずれかを管理する責任を持っています。データベースのパフォーマンス関連の問題が発生すると、さらにそれを調査するために自動ワークロード リポジトリ (AWR) レポートで生成された。

展示を見ると、AWR レポートを調べます。

### Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		584		29.08	
library cache: mutex X	14,721	71	5	3.53	Concurrency
latch: shared pool	1,158	55	48	2.76	Concurrency
cursor: pin S wait on X	3,777	50	13	2.50	Concurrency
log file sync	672	17	25	0.83	Commit

### Time Model Statistics

- Total time in database user-calls (DB Time): 2008.5s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % of DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	1,731.94	86.23
DB CPU	584.11	29.08
parse time elapsed	533.72	26.57
hard parse elapsed time	416.43	20.73
connection management call elapsed time	33.26	1.66
PL/SQL compilation elapsed time	10.58	0.53
Java execution elapsed time	8.01	0.40
failed parse elapsed time	5.20	0.26
PL/SQL execution elapsed time	3.66	0.18
hard parse (sharing criteria) elapsed time	1.94	0.10
hard parse (bind mismatch) elapsed time	1.33	0.07
sequence load elapsed time	0.41	0.02
repeated bind elapsed time	0.05	0.00
DB time	2,008.48	
background elapsed time	32.06	
background cpu time	4.79	

**Load Profile**

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	3.8	12.6	0.01	0.00
DB CPU(s):	1.1	3.7	0.00	0.00
Redo size:	6,062.3	20,190.1		
Logical reads:	5,982.5	19,924.3		
Block changes:	25.5	84.9		
Physical reads:	2,778.2	9,252.7		
Physical writes:	2.9	9.7		
User calls:	1,263.4	4,207.7		
Parses:	506.6	1,687.3		
Hard parses:	53.3	177.5		
W/A MB processed:	726,646.9	2,420,040.5		
Logons:	1.1	3.5		
Executes:	513.1	1,708.9		
Rollbacks:	0.1	0.3		
Transactions:	0.3			

## Dictionary Cache Stats

- "Pot Misses" should be very low (<2% in most cases)
- "Final Usage" is the number of cache entries being used

Cache	Get Requests	Pct Miss	Scan Reqs	Pct Miss	Mod Reqs	Final Usage
do_awr_control	13	69.23	0		2	1
do_database_links	1,074	0.56	0		0	0
do_global_oids	15,419	2.87	0		0	13
do_histogram_data	77,565	21.21	0		0	571
do_histogram_defs	168,045	23.16	0		0	1,014
do_object_grants	44,042	4.17	0		0	59
do_objects	358,789	3.30	0		0	398
do_profiles	548	2.19	0		0	1
do_rollback_segments	230	0.00	0		0	38
do_segments	99,805	15.72	0		5	279
do_sequences	25	100.00	0		25	0
do_tablespaces	85,888	0.04	0		0	5
do_users	179,387	0.35	0		0	20
global database name	927	0.11	0		0	1
kg/subheap_object	197	30.48	0		0	0
outstanding_alerts	19	84.74	0		0	1

[Back to Top](#)

## Library Cache Activity

- "Pot Misses" should be very low

Namespace	Get Requests	Pct Miss	Pin Requests	Pct Miss	Reloads	Invali-dations
BODY	1,832	1.36	3,673	1.55	23	0
CLUSTER	2,761	1.81	1,590	3.14	0	0
INDEX	947	35.59	947	35.80	1	0
JAVA DATA	4	75.00	873	0.89	0	0
SQL AREA	340,330	23.79	602,683	12.78	22,142	5,231
TABLE/PROCEDURE	145,489	2.49	191,059	8.55	5,812	0
TRIGGER	5,539	0.23	5,539	0.29	0	0

何がこのデータベースの問題だろうか？

- Java プールが構成されていません。
- システム内の CPU が遅いです。
- 共有プール サイズが不十分である。
- データベース バッファ キャッシュが不十分である。
- OPEN\_CURSORS パラメータは小さな値に設定されています。

**Answer: C**

4. あなたは、オンライントランザクション処理 (OLTP) システムで作業している空きバッファがあなたのデータベース インスタンスに対して、複数の CPU を持つマシンで実行されているイベントを待機が



検出されました。あなたが最初のステップとして、データベース バッファ キャッシュ サイズを増加しました。データベースの作業の数時間後に、さらなる調査は、同じイベントが記録されていることを示しています。

何が将来的にこのイベントを避けるために、次のステップでしょうか？

- A. パラメータの DBWR\_IO\_SLAVES の値を小さくします。
- B. TRUE USE\_INDIRECT\_DATA\_BUFFERS パラメータを設定します。
- C. パラメータ DB\_WRITER\_PROCESSES の値を増やします。
- D. パラメータ DB\_FILE\_MULTIBLOCK\_READ\_COUNT の値を増やします。

**Answer: C**

5. あなたは、Oracle Database9i から Oracle Database 11g にアップグレードされた開発用データベースに取り組んでいます。このデータベースで検索する ADDM は、別紙に示すように、共有プールは、不十分なサイズであることを述べています。



このワークロードのさまざまな種類によるもので、これだけのピーク時に発生することを診断した。あなたは、原因となる問題不十分なデータベース バッファ キャッシュ バッファ キャッシュを縮小ではなく、このサイズを変更しようとしてしました。以下は、関連するパラメータの設定は次のとおりです。

SQL> show parameter sga

name	TYPE	VALUE
lock_sga	boolean	FALSE
pre_page_sga	boolean	FALSE
sga_max_size	big integer	300M
sga_target	big integer	0

SQL> show parameter target

name	TYPE	VALUE
fast_start_mttr_target	integer	0
memory_max_target	big integer	0
memory_target	big integer	0
pga_aggregate_target	big integer	100M
sga_target	big integer	0

あなたは、プログラム グローバル領域 (PGA) のサイズに影響を与えずに、SGA 内のシステム グローバル領域 (SGA) コンポーネント間のメモリのバランスをしたいと思います。

どのアクションは、この問題を解決するでしょうか？

- A. 300M に SGA\_TARGET パラメータを設定します。
- B. 400M を SGA\_MAX\_SIZE は、パラメータを設定します。

C. MEMORY\_TARGET100M にパラメータを設定します。

D. MEMORY\_MAX\_TARGET の 300M にパラメータを設定します。

**Answer: A**

6. あなたは、クエリの一部がデータベース SALES\_RECORDS のテーブルでパフォーマンスが低下していることを観察した。

さらに調査では、一日の終わりには、表の内容を SALES\_RECORDS SALES 表に転送され、テーブル SALES\_RECORDS から削除されていることを見つける。削除された操作は、テーブルがまばらに移入する原因となります。

あなたは、テーブルを縮小するには、ALTER TABLE ... SHRINK SPACE COMPACT コマンドを使用することを決めた。

なぜあなたはこの方法を選びますか？（該当するものすべてを選択します。）

A. それはピーク時に使用することができるので

B. 不要な回避してカーソルを無効化するため

C. それはすぐにハイウォーターマーク（HWM）を調整するため

D. あなたは、運用の縮小をまたぐかもかもしれません実行時間の長いクエリを持っているので

E. それによって高速に運用縮小すること、任意のデータ操作言語（DML）操作を実行できませんので

**Answer: ABD**

7. 展示を見ると、次のクエリから得られる出力の一部を調べる：

STAT_ID	STAT_NAME	VALUE
3649082374	DB time	61021783
2748282437	DB CPU	3890625
4157170894	background elapsed time	42472524
2451517896	background cpu time	2796875
4127043053	sequence load elapsed time	0
1431595225	parse time elapsed	10983653
372226525	hard parse elapsed time	10480831
2821698184	sql execute elapsed time	50353110
1990024365	connection management call elapsed time	855906

SQL> SELECT \* FROM v\$sys\_time\_model;

時間モデル統計の 3 つの正しい解釈を選択します。（3 つ選択してください。）

A. DB 時間は、すべての nonidle と、アイドル状態のユーザーセッションの待機時間が含まれています。

B. SQL は、経過時間を実行するクエリ結果のフェッチを実行するのに費やされる時間が含まれています。

C. DB CPU はデータベース ユーザーレベルの呼び出しとバックグラウンドの CPU 時間に費やされる CPU 時間が含まれています。

D. SQL では、バインドの経過時間のようなハード解析経過時間のコンポーネントが含まれています。経過時間を実行します。

E. DB 時間は、接続管理は、バックグラウンドプロセスの時間を除く経過時間を呼び出すことが含まれています。

**Answer: BDE**



8. 歴史的に午後 10 時と真夜中の間にメンテナンスウィンドウで完了バッチ ワークロードは、現在、パフォーマンスの低下を示すと、午前 2 時に完了している。

パフォーマンス低下の診断に役立つように、組織内の上級 DBA は `awrddrpt.sql` 比較期間のレポートを生成するスクリプトを実行するように求められます。

どの 2 つのステートメントは、このスクリプトによって生成されるレポートについては真ですか？ (2 つ選択してください。)

- A. それは移動ウィンドウ ベースラインに基づいて自動的にリフレッシュされます。
- B. これは、同じ期間の期間の間の任意の 2 つの選択の詳細を比較します。
- C. これは、各期間のデータベースに費やす時間の量によって統計を正規化します。
- D. それは、同じまたは異なる期間の詳細は間の 2 つの連続した期間を比較して、60 分ごとに更新されます。

**Answer: BC**

9. あなたは、パフォーマンス チューニングの活動の一環として、適応しきい値を使用するように計画しています。あなたのデータベース内のすべての観測値との比較のデフォルトメトリックの移動ウィンドウ ベースラインのウィンドウサイズを増やすことにしました。あなたは、Enterprise Manager を使用してウィンドウサイズを増やすしようとすると、次のエラーが発生する:

コミットに失敗しました: **ORA-13541**: システムは、ウィンドウ ベースラインのサイズ (1296000) 保持より大きい (1036800) **ORA-06512** を移動させる: "SYS.DBMS\_WORKLOAD\_REPOSITORY"で、行 601 **ORA-06512**: 行 2 での

どのアクションが正常に前述のタスクを実行することができますか?

- A. フラッシュバックの保存期間を増加させる
- B. SQL 管理ベースの保存期間を増加させる
- C. データベース インスタンスの UNDO 保存期間を増加させる
- D. 自動ワークロード リポジトリ (AWR) の保存期間を増加させる

**Answer: D**

10. アクティブ セッション履歴 (ASH) データの約 2 正しい文を識別します。(2 つ選択してください。)

- A. SGA メモリーの一部は、ASH ローリングバッファとしてデータを格納するために使用されています。
- B. ASH データは、任意の 2 つの小さな時間間隔の間に分析することができます。
- C. 秒 ASH メモリー内のすべてのデータがすべての 3 分の 1 に MMON によってディスクにフラッシュされます。
- D. バッファがいっぱいになるときに、メモリー内のディスクへのすべての ASH データは MMNL プロセスによってフラッシュされます。

**Answer: AB**

11. セッション内のユーザーは、オプティマイザ モードを設定するには、次の SQL ステートメントを実行:

セッションセット `OPTIMIZER_MODE= ALL_ROWS` の ALTER

それは、そのセッションのオプティマイザの目標にどのような影響を及ぼすでしょうか？ (該当するものすべてを選択します。)

- A. 文レベルの `OPTIMIZER_MODE` ヒントは、セッションレベルの設定よりも優先されます。
- B. インスタンス レベルの `OPTIMIZER_MODE` で設定されたパラメータは、セッション レベルの値よりも優先されます。

C. オプティマイザは関係なく、統計の存在の、コストベースのアプローチを使用して、それが最善の応答時間を目標に最適化します。

D. オプティマイザは、セッション内のすべての SQL 文に対してコストベースのアプローチを使用して、かかわらず、統計の存在、それが最高のスループットを目標に最適化します。

**Answer: AD**

12. あなたは、アプリケーション ユーザーのほとんどは、最近追加または変更された行にアクセスするクエリを実行する日までにオンライン トランザクション処理 (OLTP) システムで作業している。アプリケーションが複数のテーブルに基づくクエリのほとんどを持っています。しかし、夜なので、いくつかのバッチ処理が行われます。

つのアクションこれはあなたのアプリケーションのニーズに基づいて、オプティマイザの目標を選択することをお勧めしますか？ (2つ選択してください。)

A. インスタンス レベルで ALL\_ROWS OPTIMIZER\_MODE パラメータを設定

B. インスタンス レベルで OPTIMIZER\_MODE FIRST\_ROWS\_n にパラメータを設定

C. ヒントを追加する開発者を求めて /\* ALL\_ROWS \*/ で長時間実行されるバッチ処理のクエリ

D. ヒントを追加する開発者を求めて /\* FIRST\_ROWS\_n \*/ で長時間実行されるバッチ処理のクエリ

**Answer: BC**

13. SQL 文の最適化アプローチと目標に選んでいる間、どの 3 つの要因は、オプティマイザの動作に影響を与える？ (3つ選択してください。)

A. SQL 文の解析

B. オペレーティングシステム (OS) の統計

C. データ ディクショナリ内のオブジェクトの統計情報

D. 初期化パラメータ OPTIMIZER\_MODE

E. 問合せオプティマイザの目標を変更するための SQL オプティマイザ ヒント

**Answer: CDE**

14. 以下の特定のインスタンスの初期化パラメータの値を調べる:

name	TYPE	VALUE
optimizer_capture_sql_plan_baselines	boolean	FALSE
optimizer_dynamic_sampling	integer	2
optimizer_features_enable	string	11.1.0.6
optimizer_index_caching	integer	0
optimizer_index_cost_adj	integer	100
optimizer_mode	string	ALL_ROWS
db_file_multiblock_read_count	integer	64

インデックスは、クエリの WHERE 句で使用されている列上に作成されます。あなたは、クエリがインデックスを使用していないことに注意してください。代わりに索引スキャンの、全表スキャンが使用されます。

展示を見ると、クエリの自動トレースの出力を調べます。

```
select * from employees where employee_id=107;
```

Execution Plan

Plan hash value: 1601196873

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	71	3 (0)	00:00:01
* 1	TABLE ACCESS FULL	T	1	71	3 (0)	00:00:01

Predicate Information (identified by operation id):

1 - filter("EMPLOYEE\_ID"=107)

何が理由だろうか？（該当するものすべてを選択します。）

- A. 初期化パラメータ OPTIMIZER\_INDEX\_COST\_ADJ は低値を持っています。
- B. 初期化パラメータ DB\_FILE\_MULTIBLOCK\_READ\_COUNT は低値を持っています。
- C. テーブルとテーブルに関連付けられたすべてのインデックスの統計は、現行ではありません。
- D. テーブルは、高水位標の下で DB\_FILE\_MULTIBLOCK\_READ\_COUNT ブロック未満を持っています。

**Answer: CD**

15. 以下の特定のインスタンスの初期化パラメータの値を調べます。

name	TYPE	VALUE
optimizer_capture_sql_plan_baselines	boolean	FALSE
optimizer_dynamic_sampling	integer	2
optimizer_features_enable	string	11.1.0.6
optimizer_index_caching	integer	0
optimizer_index_cost_adj	integer	100
optimizer_mode	string	ALL_ROWS
db_file_multiblock_read_count	integer	64

あなたは、クエリのいずれかではなく、索引の一意スキャン（ビューExhibit2）の全表スキャンを（ビューExhibit1）を使用していることに注意してください。インデックスは、クエリの WHERE 句でアクセスされる列に存在しています。全表スキャンのコストは、索引の一意スキャンのためにそれ以上のものです。

```
select * from employees where employee_id=107;
```

Execution Plan

Plan hash value: 1601196873

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	71	3 (0)	00:00:01
* 1	TABLE ACCESS FULL	T	1	71	3 (0)	00:00:01

Predicate Information (identified by operation id):

1 - filter("EMPLOYEE\_ID"=107)

Plan hash value: 1076294677

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	71	1 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID	T	1	71	1 (0)	00:00:01
* 2	INDEX UNIQUE SCAN	EMP_PK	1		1 (0)	00:00:01

Predicate Information (identified by operation id):

2 - access("EMPLOYEE\_ID"=107)

なぜ、オプティマイザは、索引の一意スキャンを介して、全表スキャンを選択するのでしょうか？（該当するものすべてを選択します。）

- A. 初期化パラメータは、OPTIMIZER\_INDEX\_COST\_ADJ 低い値をに設定されます。
- B. 初期化パラメータは、OPTIMIZER\_INDEX\_COST\_ADJ 高い値に設定されています。
- C. 初期化パラメータが低い値を DB\_FILE\_MULTIBLOCK\_READ\_COUNT に設定されています。
- D. テーブルとテーブルに関連付けられたすべてのインデックスの統計は、現行ではありません。

**Answer: BD**

16. 列 CUST\_CITY、CUST\_STATE\_PROVINCE、頻繁にクエリの WHERE 句で country\_id が一緒に使用されます。CUSTOMERS 表は、データの 20 GB の大きなテーブルがあります。あなたは、これらの 3 つのカラムの選択性は、オプティマイザが計算される選択性によって異なることに注意してください。あなたは何をオプティマイザによって計算された選択性に影響を与えることをお勧めしますか？

- A. すべての列を連結することによって、ファンクション ベース索引を作成する
- B. 手順の DBMS\_STATS.GATHER\_TABLE\_STATS を使用して、これらの列のヒストグラム統計を更新
- C. 仮想 DBMS\_STATS.CREATE\_EXTENDED\_STATS 列を作成し、仮想列にインデックスを作成する関数を使用して
- D. DBMS\_STATS.GATHER\_TABLE\_STATS に仮想列の統計情報を収集するために、仮想 DBMS\_STATS.CREATE\_EXTENDED\_STATS 列およびプロシージャを作成する関数を使用して

**Answer: D**

17. CUSTOMERS 表の説明を調べるために Exhibit1 を表示します。

```
SQL> DESC customers
Name                                         Null?    Type
-----
CUST_ID                                     NOT NULL NUMBER
CUST_FIRST_NAME                             NOT NULL VARCHAR2 (20)
CUST_LAST_NAME                              NOT NULL VARCHAR2 (40)
CUST_GENDER                                 NOT NULL CHAR (1)
CUST_YEAR_OF_BIRTH                           NOT NULL NUMBER (4)
CUST_MARITAL_STATUS                          VARCHAR2 (20)
CUST_STREET_ADDRESS                          NOT NULL VARCHAR2 (40)
CUST_POSTAL_CODE                             NOT NULL VARCHAR2 (10)
CUST_CITY                                    NOT NULL VARCHAR2 (30)
CUST_CITY_ID                                 NOT NULL NUMBER
CUST_STATE_PROVINCE                          NOT NULL VARCHAR2 (40)
CUST_STATE_PROVINCE_ID                       NOT NULL NUMBER
COUNTRY_ID                                   NOT NULL NUMBER
CUST_MAIN_PHONE_NUMBER                       NOT NULL VARCHAR2 (25)
CUST_INCOME_LEVEL                            VARCHAR2 (30)
CUST_CREDIT_LIMIT                            NUMBER
CUST_EMAIL                                    VARCHAR2 (30)
CUST_TOTAL                                   NOT NULL VARCHAR2 (14)
CUST_TOTAL_ID                                NOT NULL NUMBER
CUST_SRC_ID                                  NUMBER
CUST_EFF_FROM                                 DATE
CUST_EFF_TO                                  DATE
CUST_VALID                                    VARCHAR2 (1)
```

あなたは、country\_id が列が使用されるときに、選択性のオプティマイザが正確ではありません観察し、クエリの WHERE 句と一緒に CUST\_STATE\_PROVINCE。

クエリの実行計画や統計情報を収集するために実行したコマンドを調べるために Exhibit2 を表示します。

```
SQL> SELECT COUNT(*) FROM customers WHERE country_id=52790 and cust_state_province='CA';

COUNT(*)
-----
      3341

SQL> exec dbms_stats.gather_table_stats(null,'CUSTOMERS',method_opt=>'for all columns size 1');

PL/SQL procedure successfully completed.

SQL> explain plan for select count(*) from customers where COUNTRY_ID=52790 and CUST_STATE_PROVINCE='CA';

Explained.

SQL> select * from table(dbms_xplan.display())
 2 ;

PLAN_TABLE_OUTPUT
-----
Plan hash value: 296924608

-----
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----
|  0 | SELECT STATEMENT   |               |      1 |    16 |    406 (1)| 00:00:05 |
|  1 |   SORT AGGREGATE   |               |      1 |    16 |           |          |
|* 2 |    TABLE ACCESS FULL| CUSTOMERS     |     20 |    320 |    406 (1)| 00:00:05 |
-----

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT
-----

 2 - filter("CUST_STATE_PROVINCE"='CA' AND "COUNTRY_ID"=52790)

14 rows selected.
```

オプティマイザは、20 行ではなく、テーブルから返される行の実際の数である 3.341 の行よりも処理さ

れることを予測しています。

あなたは、オプティマイザーが実際の行数を検出するために何ができるのでしょうか？

- A. ALL を STATISTICS\_LEVEL するパラメータを設定します。
- B. FALSE optimizer\_use\_pending\_statistics をにパラメータを設定します。
- C. CUST\_STATE\_PROVINCE と country\_id 列の拡張統計を作成します。
- D. DBMS\_STATS.SET\_TABLE\_PREFS を使用してプロシージャの STALE\_PERCENT の Customers テーブルの値を増やします。

**Answer: C**

18. CUSTOMERS 表の説明を調べるために Exhibit1 を表示します。

```
SQL> DESC customers
Name                               Null?    Type
-----
CUST_ID                             NOT NULL NUMBER
CUST_FIRST_NAME                      NOT NULL VARCHAR2 (20)
CUST_LAST_NAME                       NOT NULL VARCHAR2 (40)
CUST_GENDER                          NOT NULL CHAR (1)
CUST_YEAR_OF_BIRTH                   NOT NULL NUMBER (4)
CUST_MARITAL_STATUS                  VARCHAR2 (20)
CUST_STREET_ADDRESS                  NOT NULL VARCHAR2 (40)
CUST_POSTAL_CODE                     NOT NULL VARCHAR2 (10)
CUST_CITY                            NOT NULL VARCHAR2 (30)
CUST_CITY_ID                         NOT NULL NUMBER
CUST_STATE_PROVINCE                   NOT NULL VARCHAR2 (40)
CUST_STATE_PROVINCE_ID               NOT NULL NUMBER
COUNTRY_ID                           NOT NULL NUMBER
CUST_MAIN_PHONE_NUMBER                NOT NULL VARCHAR2 (25)
CUST_INCOME_LEVEL                    VARCHAR2 (30)
CUST_CREDIT_LIMIT                     NUMBER
CUST_EMAIL                            VARCHAR2 (30)
CUST_TOTAL                           NOT NULL VARCHAR2 (14)
CUST_TOTAL_ID                        NOT NULL NUMBER
CUST_SRC_ID                           NUMBER
CUST_EFFECTIVE_FROM                  DATE
CUST_EFFECTIVE_TO                     DATE
CUST_VALID                            VARCHAR2 (1)
```

CUSTOMERS 表は、今日頻繁に更新されています。頻繁に使用される SQL 文では、行の推定値と実際の行数が大幅に差別化フェッチされていることがわかります。 country\_id がカラムにはインデックスがあります。

Exhibit2 を表示し、クエリの実行プランを確認します。

```
SQL> SELECT cust_id, cust_last_name, cust_total
2 FROM customers
3 WHERE country_id = 52790
```

296320 rows selected.

#### Execution Plan

Plan hash value: 2008213504

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		73966	2383K	798 (1)	00:00:10
* 1	TABLE ACCESS FULL	CUSTOMERS	73966	2383K	798 (1)	00:00:10

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

1 - filter("COUNTRY\_ID"=52790)

#### Statistics

```

1 recursive calls
0 db block gets
27844 consistent gets
8597 physical reads
0 redo size
6373693 bytes sent via SQL*Net to client
217714 bytes received via SQL*Net from client
19756 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
296320 rows processed
```

あなたは何を、オブティマイザの推定を改善することをお勧めでしょうか？

- A. ALL\_STATISTICS\_LEVEL パラメータを設定
- B. FALSE optimizer\_use\_pending\_statistics をにパラメータを設定
- C. CUST\_LAST\_NAME、CUST\_ID、と CUST\_TOTAL 列の拡張統計を作成する
- D. 使用して、CUSTOMERS 表の統計情報を更新 DBMS\_STATS.GATHER\_TABLE\_STATS procedure

**Answer: D**

19. あなたは、意思決定支援システム（DSS）に取り組んでいます。インデックスは、CUSTOMERS 表の country\_id が列で使用できます。

展示を見るとパラメータの設定とクエリの実行プランを確認します。



```

NAME                                     TYPE          VALUE
-----
db_file_multiblock_read_count           integer        49

SQL> SELECT BLOCKS, EMPTY_BLOCKS FROM ALL_TABLES
WHERE TABLE_NAME = 'CUST';

   BLOCKS  EMPTY_BLOCKS
-----
   2902         0

SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS('SH','CUST');

PL/SQL procedure successfully completed.

SQL> select cust_id, cust_last_name from sh.cust where country_id=52790;

74080 rows selected.

Execution Plan
-----
Plan hash value: 260468903

-----
| Id | Operation          | Name | Rows  | Bytes | Cost (CPU) | Time |
-----
|  0 | SELECT STATEMENT   |      | 74150 | 1303K | 793 (1)    | 00:00:10 |
|*  1 | TABLE ACCESS FULL| CUST | 74150 | 1303K | 793 (1)    | 00:00:10 |
-----

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

   1 - filter("COUNTRY_ID"=52790)

Statistics
-----
   0 recursive calls
   0 db block gets
  7659 consistent gets
  2848 physical reads
   0 redo size
1584703 bytes sent via SQL*Net to client
  54738 bytes received via SQL*Net from client
   4940 SQL*Net roundtrips to/from client

```

なぜ、フルテーブルスキャンの代わりに索引スキャンを使用したクエリは何ですか？

- A. country\_id が列のヒストグラム統計が更新されていないため
- B. country\_id が列のインデックスのインデックスの統計は、現行ではありませんので
- C. 初期化パラメータが大きな値を DB\_FILE\_MULTIBLOCK\_READ\_COUNT に設定されているため
- D. オプティマイザは、テーブル内のブロックの大部分がアクセスされると予測しているからである。したがって、索引が使用可能であるにもかかわらず、全表スキャンを使用しています。

**Answer: D**

20. Exhibit1 を表示し、Customers テーブルにインデックスを調べます。

```
SQL> SELECT i.table_name, column_name, o.object_name , o.object_id, i.index_type
FROM user_objects o,
user_indexes i,
user_ind_columns c
WHERE o.object_type= 'INDEX'
AND i.table_name LIKE '%CUST%'
AND i.index_name = o.object_name
AND i.index_name = c.index_name
```

TABLE_NAME	COLUMN_NAME	OBJECT_NAME	OBJECT_ID	INDEX_TYPE
CUSTOMERS	CUST_GENDER	CUSTOMERS_GENDER_BIX	70685	BITMAP
CUSTOMERS	CUST_MARITAL_STATUS	CUSTOMERS_MARITAL_BIX	70686	BITMAP
CUSTOMERS	CUST_ID	CUSTOMERS_PK	70473	NORMAL
CUSTOMERS	CUST_YEAR_OF_BIRTH	CUSTOMERS_YOB_BIX	70687	BITMAP
CUSTOMERS	COUNTRY_ID	CUST_COUNTRY	79047	NORMAL

CUSTOMERS 表の統計は、次のコマンドを使用して、最近更新されました。

```
SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS ( 'SH', 'CUSTOMERS', METHOD_OPT=> 'ALL INDEXED COLUMNS SIZE の AUTO' );
```

クエリプランを調べるために Exhibit2 を表示します。インデックスが country\_id がと CUST\_GENDER 列上に存在するにもかかわらず、クエリは、全表スキャンを使用しています。何が理由だろうか？

```
SQL> SELECT cust_id,cust_last_name
FROM customers
WHERE cust_gender='M' AND country_id=52790;
```

49292 rows selected.

Execution Plan

Plan hash value: 2008213504

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		49456	821K	2478 (1)	00:00:30
* 1	TABLE ACCESS FULL	CUSTOMERS	49456	821K	2478 (1)	00:00:30

Predicate Information (identified by operation id):

```
1 - filter("COUNTRY_ID"=52790 AND "CUST_GENDER"='M')
```

Statistics

```
1 recursive calls
0 db block gets
12275 consistent gets
9077 physical reads
0 redo size
1052848 bytes sent via SQL*Net to client
36566 bytes received via SQL*Net from client
3288 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
49292 rows processed
```

- A. country\_id が列のヒストグラム統計が更新されていないため
- B. 初期化パラメータが大きな値を DB\_FILE\_MULTIBLOCK\_READ\_COUNT に設定されているため
- C. 索引が使用可能であっても、オプティマイザは、インデックススキャンに比べて小さくなるように、

全表スキャンを使用してブロックをアクセスするためのコストを計算しますので

D. CUST\_GENDER と country\_id 列に対する索引のタイプが異なるため、列のインデックスは CUST\_GENDER ビットマップ索引である、と country\_id 列に btree インデックスです。

**Answer: C**