

You'll make breakthroughs

INTERSYSTEMS®

MAKING BREAKTHROUGHS
InterSystems Symposia
2014

Caché SQL 最新情報

インターシステムズジャパン株式会社

セールスエンジニア
(テクニカル・コンサルタント)
堀田 稔

A photograph of two mountaineers in red and yellow gear climbing a steep, rocky mountain peak. The background shows a vast, snow-covered mountain range under a blue sky with scattered clouds.

You'll make breakthroughs

アジェンダ

- Caché SQL概要
- インデックスの種類
- SQL統計情報とクエリ・プラン
- チューニングとヒント
- ビットマップインデックスの圧縮
- インデックスの追加・再構築の新機能

INTERSYSTEMS®

1



You'll make breakthroughs

ダイレクト・アクセス vs SQLアクセス

- ダイレクト・アクセス
 - データの取得方法をプログラムで記述する(How)。
 - プログラムのロジックを改善させることで、パフォーマンスを向上させる。
 - プログラマによる柔軟なコントロール。
 - インデックスの一貫性、その他データモデルの一貫性を維持するのはプログラマの責任。
- SQLアクセス
 - どんなデータを取得したいかを、SQLで記述する(What)。
 - オプティマイザが、より良いパフォーマンスの実行プランを計算・推測し、コードを生成。
 - チューニングは、基本的に、オプティマイザに情報を与えることで行う。
 - インデックスや、その他制約の一貫性は、システムが自動的に維持。

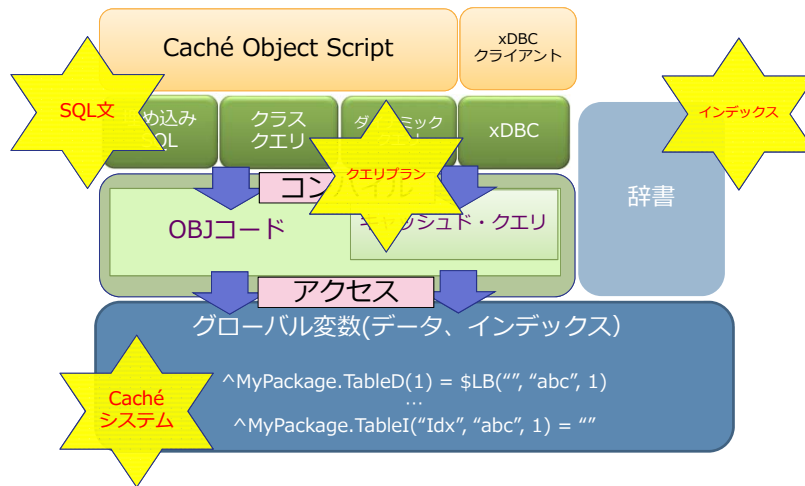
3

INTERSYSTEMS®

You'll make breakthroughs

Caché SQL 概念図

Caché SQL 実行の仕組み



INTERSYSTEMS®

4



You'll make breakthroughs

INTERSYSTEMS®

インデックスの種類



5

You'll make breakthroughs

インデックス

インデックスの種類

- 通常のインデックス
- ビットマップインデックス
- ビットスライスインデックス
- 複合インデックス（複数フィールド）

INTERSYSTEMS®

6

You'll make breakthroughs

通常のインデックス

- ユニークキーなど、どんなフィールドにも使用可能
- その他のフィールドの値（インデックス対象となっているフィールド以外のフィールドの値）を、保持することが可能
- パフォーマンスは安定している

	インデックス名	値	ID	その他フィールドの値 (オプション)
1:	^MyDemo.Table11.SSNIndex	100-10-1138	217474	==
2:	^MyDemo.Table11.SSNIndex	100-10-3795	329642	==
3:	^MyDemo.Table11.SSNIndex	100-10-4043	363974	==
4:	^MyDemo.Table11.SSNIndex	100-10-4213	216331	==
5:	^MyDemo.Table11.SSNIndex	100-10-4799	313071	==
6:	^MyDemo.Table11.SSNIndex	100-10-7541	216824	==
7:	^MyDemo.Table11.SSNIndex	100-10-8444	267897	==
8:	^MyDemo.Table11.SSNIndex	100-10-8620	234322	==
9:	^MyDemo.Table11.SSNIndex	100-10-8713	2493161	==
10:	^MyDemo.Table11.SSNIndex	100-10-9621	358458	==
11:	^MyDemo.Table11.SSNIndex	100-11-1092	192067	==
12:	^MyDemo.Table11.SSNIndex	100-11-1180	2374381	==
13:	^MyDemo.Table11.SSNIndex	100-11-2082	266811	==
14:	^MyDemo.Table11.SSNIndex	100-11-2248	2114433	==
15:	^MyDemo.Table11.SSNIndex	100-11-2586	2358156	==
16:	^MyDemo.Table11.SSNIndex	100-11-3415	2174054	==
17:	^MyDemo.Table11.SSNIndex	100-11-4943	2117668	==
18:	^MyDemo.Table11.SSNIndex	100-11-5062	2438242	==
19:	^MyDemo.Table11.SSNIndex	100-11-5949	2448900	==
20:	^MyDemo.Table11.SSNIndex	100-11-7106	2344573	==
21:	^MyDemo.Table11.SSNIndex	100-11-8037	2133440	==
22:	^MyDemo.Table11.SSNIndex	100-11-8266	2309021	==
23:	^MyDemo.Table11.SSNIndex	100-11-8404	2136384	==
24:	^MyDemo.Table11.SSNIndex	100-11-9145	222521	==

INTERSYSTEMS®

7

You'll make breakthroughs

ビットマップインデックス

```

1: "MyDemo.Table11("StateIdx","AK",1) = 001000100.....10011 (64000ビット)
2: "MyDemo.Table11("StateIdx","AK",2) =
3: "MyDemo.Table11("StateIdx","AK",3) = $zwc(405,7998,37,42,86,89,95,117,154,198,298,375,415,436,487,562,587,591,627,713,953,971,1003,1202,1242,
4: "MyDemo.Table11("StateIdx","AK",4) = $zwc(406,8000,40,48,53,81,140,201,213,241,258,344,470,527,587,588,685,845,983,1106,1167,1204,1212,1296,1
5: "MyDemo.Table11("StateIdx","AK",5) = $zwc(402,7999,8,42,46,59,117,157,210,266,336,440,475,531,619,644,646,671,721,782,786,789,801,930,970,987
6: "MyDemo.Table11("StateIdx","AK",6) = $zwc(405,7990,252,315,444,488,608,612,702,703,705,720,868,1115,1129,1222,1274,1345,1395,1399,1516,1537,1
7: "MyDemo.Table11("StateIdx","AK",7) = $zwc(407,7995,44,247,250,251,268,297,304,337,482,502,531,539,569,625,661,674,720,733,826,846,847,870,967
8: "MyDemo.Table11("StateIdx","AK",8) = $zwc(401,7991,13,39,60,202,217,225,257,264,274,299,421,456,555,578,577,653,663,791,803,823,852,877,879,9
9: "MyDemo.Table11("StateIdx","AK",9) = $zwc(402,7996,75,235,270,285,315,352,357,443,454,469,494,497,531,564,580,682,755,776,1154,1184,1207,1223
10: "MyDemo.Table11("StateIdx","AK",10) = $zwc(407,8000,29,129,181,193,268,278,310,312,392,408,410,437,463,477,548,671,672,794,912,1058,1084,1209,
11: "MyDemo.Table11("StateIdx","AK",11) = $zwc(402,8000,73,78,252,338,352,410,422,513,528,611,757,802,939,976,1046,1067,1127,1179,1242,1252,1259,1
12: "MyDemo.Table11("StateIdx","AK",12) = $zwc(403,7999,14,15,20,58,127,381,449,469,490,511,515,516,670,716,748,779,787,794,889,935,988,991,1030,1
13: "MyDemo.Table11("StateIdx","AK",13) = $zwc(400,7993,8,28,86,155,258,270,310,332,448,473,494,539,625,673,786,1127,1227,1253,1342,1368,1457,1500
14: "MyDemo.Table11("StateIdx","AK",14) = $zwc(402,7995,61,63,306,371,402,440,592,606,699,771,847,890,940,955,971,1033,1095,1205,1240,1354,1380,14
15: "MyDemo.Table11("StateIdx","AK",15) = $zwc(403,7998,7,110,130,171,317,463,464,553,569,572,628,662,733,736,775,807,817,818,826,1066,1149,1157,1
16: "MyDemo.Table11("StateIdx","AK",16) = $zwc(401,5005,23,69,95,141,186,212,255,286,361,391,409,433,503,534,545,561,623,688,727,767,1011,1062,110
17: "MyDemo.Table11("StateIdx","AL",1) = $zwc(406,7981,37,104,216,227,385,398,432,487,553,558,637,663,677,809,943,1020,1030,1081,1234,1252,1313,1
18: "MyDemo.Table11("StateIdx","AL",2) = $zwc(401,7996,23,43,128,156,170,208,271,327,337,394,429,469,556,580,598,636,671,680,741,748,871,981,1004
19: "MyDemo.Table11("StateIdx","AL",3) = $zwc(403,7998,36,129,145,148,209,263,416,425,431,474,475,506,586,672,703,707,727,740,843,903,958,1015,10
20: "MyDemo.Table11("StateIdx","AL",4) = $zwc(400,8000,106,291,300,345,601,640,649,785,813,855,932,993,1013,1016,1029,1037,1060,1087,1088,1193,12
21: "MyDemo.Table11("StateIdx","AL",5) = $zwc(406,7995,38,118,150,202,204,233,258,268,269,299,520,522,553,564,661,752,876,905,960,982,989,1024,11
22: "MyDemo.Table11("StateIdx","AL",6) = $zwc(402,7997,107,217,219,281,288,325,328,411,461,737,768,772,782,793,824,849,852,855,864,969,1071,1188,
23: "MyDemo.Table11("StateIdx","AL",7) = $zwc(405,7997,95,129,157,163,264,300,317,349,382,402,518,555,577,579,624,689,746,812,861,863,929,940,944
24: "MyDemo.Table11("StateIdx","AL",8) = $zwc(405,7993,80,136,141,242,263,521,545,585,657,703,731,736,756,780,816,894,916,939,1024,1145,1283,1357

```

INTERSYSTEMS

8

You'll make breakthroughs

ビットマップインデックス

- コンパクト（1ノードに64000行分の情報）
- フィールドのデータの値が、概ね10000種類以下であれば最も良い選択肢（Uniqueインデックスでは使用できない）
- 複雑な条件の組み合わせを、高速なビット演算に置き換えられる
- データ更新のパフォーマンスに対するインパクトが小さい
- DELETE, INSERTを繰り返すテーブルでは、パフォーマンスが劣化するケースがある
 - 改善の方法に付いては後述
- IDKeyが整数でないテーブルでは使用できない

INTERSYSTEMS

9

You'll make breakthroughs

複合インデックス

複数フィールドに対するインデックス

特定の条件のクエリが
繰り返される場合、非常に有効

```
select SSN from mydemo.table1 where state = 'MA' and gender = 'F'
```

行数: 1000 パフォーマンス: 0.017 秒 2016 グローバル参照 クエリ・キャッシュ: %sqlc.MYSAMPLE.rds1
新 2014-06-09 18:26:59.297

SSN
527-82-7209
916-35-9340
826-11-4079
157-90-6173
923-96-7025
99166-1159

State, Gender		SSN
1: "MyDemo.Table1("StateGenderIdx", "AK", "F"	700108)	= \$1b("680-68-8288")
2: "MyDemo.Table1("StateGenderIdx", "AK", "F"	700348)	= \$1b("921-63-1402")
3: "MyDemo.Table1("StateGenderIdx", "AK", "F"	700499)	= \$1b("949-58-5202")
4: "MyDemo.Table1("StateGenderIdx", "AK", "F"	700633)	= \$1b("789-80-2475")
5: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701099)	= \$1b("915-45-1820")
6: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701155)	= \$1b("237-44-3222")
7: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701327)	= \$1b("536-38-2501")
8: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701332)	= \$1b("601-68-9933")
9: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701339)	= \$1b("738-53-7858")
10: "MyDemo.Table1("StateGenderIdx", "AK", "F"	701801)	= \$1b("740-70-1480")
11: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702118)	= \$1b("932-37-1121")
12: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702178)	= \$1b("853-83-8177")
13: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702191)	= \$1b("821-84-2866")
14: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702233)	= \$1b("230-88-5575")
15: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702671)	= \$1b("274-14-1823")
16: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702676)	= \$1b("578-78-2918")
17: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702689)	= \$1b("495-79-4495")
18: "MyDemo.Table1("StateGenderIdx", "AK", "F"	702871)	= \$1b("192-50-3646")
19: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703109)	= \$1b("193-21-3309")
20: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703313)	= \$1b("400-38-5964")
21: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703495)	= \$1b("939-26-7395")
22: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703643)	= \$1b("524-30-3834")
23: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703885)	= \$1b("220-57-4958")
24: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703889)	= \$1b("814-78-2745")
25: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703975)	= \$1b("984-22-1754")
26: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703982)	= \$1b("365-52-7619")
27: "MyDemo.Table1("StateGenderIdx", "AK", "F"	703997)	= \$1b("299-70-7072")
28: "MyDemo.Table1("StateGenderIdx", "AK", "F"	704020)	= \$1b("719-64-4881")
29: "MyDemo.Table1("StateGenderIdx", "AK", "F"	704056)	= \$1b("984-31-5955")

INTERSYSTEMS®

10

You'll make breakthroughs

INTERSYSTEMS®

SQL統計情報とクエリ・プラン

11

You'll make breakthroughs

SQL実行時統計

%SYS.PTools.SQLStats

- SQLを含むメソッドやルーチンをコンパイルしたOBJや、キャッシュド・クエリに、SELECT文の実行時間に関する統計情報を収集するコードを生成する。
- コードの生成や、収集する統計の詳細レベルを選択できる。
- 時間のかかっているクエリ、グローバル変数の参照が多いクエリを特定できる。
- 以下の数値を集計: Global Refs, Commands Executed, Time, RowCount, クエリの実行回数。
- %SYS_PTools.SQLStatView で、すべての数値を確認できる。

INTERSYSTEMS

12

You'll make breakthroughs

クエリパフォーマンス統計の収集

- 管理ポータル: [システム]→[SQL実行時統計情報]
- \$SYSTEM.SQL.SetSQLStats() もしくは \$SYSTEM.SQL.SetSQLStatsJob()

このページのオプションを使用して、デバッグ用に SQL 実行時統計情報の取得と表示を行います。

プラン表示の統計フラグが変更されました。

設定 クエリのテスト 統計を表示

SQL プラン表示の統計フラグを渡す

- ☐ 統計コード生成を無効にする
- ☐ 統計コードの生成を有効にするが、統計情報は収集しない
- ☐ 統計コード生成を有効にして、クエリの Open と Close で統計情報を収集する
- ☐ 統計コード生成を有効にして、クエリの各モジュールレベルで統計情報を収集する

SQL 統計データを収集するには、統計を有効にしてクエリをコンパイルする必要があります。

ルーチンやクラスに含まれる SQL の統計が必要な場合は、それらのルーチンやクラスをコンパイルする必要があります。

xDBC とダイナミック SQL について、クエリキャッシュの削除でこれを行います。

統計フラグを変更した場合、クエリキャッシュの削除 ボタンをクリックしてクエリキャッシュを削除する必要があります。新しいクエリキャッシュは新しいフラグを使用して生成されます。

INTERSYSTEMS

13

You'll make breakthroughs

クエリパフォーマンス統計の収集

キャッシュド・クエリに収集用のコードが埋め込まれる

```

%OAFirst
; asl MOD# 2
; d: i%%CursorDatad(5)=3 ##class(%SYS.PTools.SQLStats).Start("MYSAMPLE", $zn, "QRS0", 2)
; s i%%CursorDatad(6)=
%OAFirst1 s i%%CursorDatad(6)=($o("MyDemo.TableID(i%%CursorDatad(6))", 1))
; i i%%CursorDatad(6)= g %OAmBdun
; s i%%CursorDatad(7)=($g("MyDemo.TableID(i%%CursorDatad(6))")) s i%%CursorDatad(8)=($lg(i%%CursorDatad(6)))
goto: $zu(115,2)=0 %OAmBuncommitted
if $zu(115,2)=1 [
; lock + "MyDemo.TableID($p(i%%CursorDatad(6), "|", 1))#"S": $zu(115,4)
; if $t { set i%%CursorDatad(4)=$name("MyDemo.TableID($p(i%%CursorDatad(6), "|", 1))", "S1")
; else { s SQLCODE=-114, %msg="Unable to acquire shared lock on table MyDemo.Table1 for RowID "
}
; asl MOD# 3
; d: i%%CursorDatad(5)=3 ##class(%SYS.PTools.SQLStats).Start("MYSAMPLE", $zn, "QRS0", 3)
; s i%%CursorDatad(6)=, $d("MyDemo.TableID(i%%CursorDatad(6))")
; %OAmBdun

```

変数 i%%CursorDatad(5)に、統計情報収集設定の値がセットされている

INTERSYSTEMS

14

You'll make breakthroughs

クエリパフォーマンス統計の収集

統計情報の確認

Menu | ホーム | 開発 | ヘルプ | ロボアクト | システム > SQL 実行時統計情報 | Ensemble
SQL 実行時統計情報 | サーバ: VMWVW07 | キュームスペース: MYSAMPLE | 変更 |
ユーザ: UnknownUser | ライセンス先: InterSystems Sales Engineers | インスタンス: ENSTEST

クエリパフォーマンス統計

SQL 実行時統計情報

このページはオブジェクションを使用して、デバッグ用に SQL 実行時統計情報の取得と表示を行います。

設定 | クエリのテスト | 統計を表示

以下のテーブルには SQL 統計情報を含むすべてのクエリの情報が表示されています。 | テーブルの再読み込み

ページサイズ: 0 | 最大行数: 1000 | 結果: 5 | ページ: 1 | 1 | 0 | 1

クエリ名	カーソル	実行回数	平均行	平均クローバ	平均コマン	平均時間
CachedSQLM04726	SQLSTATS	1	81	201	2123	00199 秒
NavigMYSAMPLE.cis2.1	QRS0	2	100	217	9756	00348 秒
NavigMYSAMPLE.cis3.1	QRS0	1	10	36	1605	00249 秒
NavigMYSAMPLE.cis4.1	QRS0	2	1	20	740	00133 秒
NavigMYSAMPLE.cis5.1	QRS0	2	19844	198821	13672895	2.99929 秒

クエリ文字列

SELECT * FROM mydemo . table1 WHERE state = ?

クエリプラン

相対コスト = 9060812

[stats] Time in Module MAIN = 2.999 Module Execution Count = 1 Global References = 1,996,821
Commands Executed = 13,672,896 Number of Rows = 19,844

- Read master map MyDemo.Table1 IDKEY, looping on ID.
- For each row:
Output the row.

INTERSYSTEMS

15

You'll make breakthroughs

インデックス使用統計

効率の悪いインデックスや必要なインデックスを探す

このページのオプションを使用して、インデックス・アナライザ・レポートをレビューします。

SQL 文のカウント

Cached Query	
Class Method	
Class Query	
Routine	

☐ システムクエリを含む? (%*, DeepSee*, HS*, Ens*)

スキーマ選択

レポートのオプションを選択

- ☐ インデックス使用状況
- ☒ テーブルスキャンを行うクエリ
- ☐ 一時インデックスを使用するクエリ
- ☐ JOIN のインデックスがないクエリ

ページサイズ: 0 最大行数: 1000 結果: 4 ページ: 1 の 1

テーブルスキャンを行うクエリ

SchemaName	TableName	Type	Class/Routine Name	ModuleName	Map Type	Ext
MyDemo	Table1	Cached Query	%sqlq.MYSAMPLE.ds2	B	master map	
MyDemo	Table1	Cached Query	%sqlq.MYSAMPLE.ds3	B	master map	
MyDemo	Table1	Cached Query	%sqlq.MYSAMPLE.ds5	B	master map	
MyDemo	Table1	Routine	CacheSqlStats4736	B	master map	

このページのオプションを使用して、インデックス・アナライザ・レポートをレビューします。

SQL 文のカウント

Cached Query	5
Class Method	409
Class Query	262
Routine	1

☐ システムクエリを含む? (%*, DeepSee*, HS*, Ens*)

スキーマ選択

レポートのオプションを選択

- ☐ インデックス使用状況
- ☐ テーブルスキャンを行うクエリ
- ☐ 一時インデックスを使用するクエリ
- ☒ JOIN のインデックスがないクエリ

ページサイズ: 0 最大行数: 1000 結果: 2 ページ: 1 の 1

JOIN のインデックスがないクエリ

SchemaName	TableName	IndexNeeded	JoinIndexFlag
MyDemo	Table1	State	0 Show Plan
MyDemo	Table2	State	0 Show Plan

INTERSYSTEMS

16

You'll make breakthroughs

クエリ・プラン

```
select * from mydemo.table1 where state = 'MA' and gender = 'F'
```

実行プランが以下に表示されます:

クエリ文字列

```
select * from mydemo.table1 where state = 'MA' and gender = 'F'
```

クエリプラン

Stateのビットマップで絞る

マスタマップをループしながら
Genderをチェック

相対コスト = 3273888

Read bitmap index MyDemo.Table1.StateIdx, using the given %SQLUPPER(State), and looping on ID.

For each row:

Read master map MyDemo.Table1.IDKEY, using the given IdKey value.
Output the row.

INTERSYSTEMS

17

You'll make breakthroughs

クエリ・プラン

ビットマップインデックスの追加後

```
select * from mydemo.table1 where state = 'MA' and gender = 'F'
```

実行プランが以下に表示されます:

クエリ文字列

```
select * from mydemo.table1 where state = 'MA' and gender = 'F'
```

クエリプラン

相対コスト = 2976839

- Generate a stream of idkey values using the multi-index combination:
((bitmap index MyDemo.Table1.StateIdx) INTERSECT (bitmap index MyDemo.Table1.GenderIdx))
- For each idkey value:
Read master map MyDemo.Table1.IDKEY, using the given idkey value.
Output the row.

Stateビットマップと
Genderビットマップの
ANDで絞る

INTERSYSTEMS

18

You'll make breakthroughs

クエリ・プラン

統計情報による比較

SELECT * FROM mydemo . table1 WHERE state = ? AND gender = ?

ルーチン	カーネル	実行回数	平均行	平均グローバル参照	平均コマンド	平均時間
%sqlcq.MYSAMPLE.cls2.1	QRS0	2	1001	2022	89123	05093
%sqlcq.MYSAMPLE.cls6.1	CU	3	101	20144.6	67457.2	75012
%sqlcq.MYSAMPLE.cls7.1	QRS0	3	17	46877.67	243189.67	40452
%sqlcq.MYSAMPLE.cls8.1	QRS0	45	67	0	51	00008
%sqlcq.MYSAMPLE.cls9.1	QRS0	3	2.33	51	1250.33	.00076

クエリ文字列

```
SELECT * FROM mydemo . table1 WHERE state = ? AND gender = ?
```

クエリプラン

相対コスト = 3273888

[stats] Time in Module MAIN = 0.039 Module Execution Count = 1 Global References = 3,948 Commands Executed = 110,263 Number of Rows = 1,001

- Read bitmap index MyDemo.Table1.StateIdx, using the given %SQLUPPER(State), and looping on ID.
- For each row:
Read master map MyDemo.Table1.IDKEY, using the given idkey value.
Output the row.

19

You'll make breakthroughs

クエリ・プラン

複合インデックスとの比較

Index StateGenderIdx On (State, Gender);

ルーチン	カーソル	実行回数	平均行	平均グローバル参照	平均コマンド	平均時間	
%sqlcq.MYSAMPLE.cls2.1	QRS0	3	1001	3019	83757	0.3122	詳細 プラン表示
%sqlcq.MYSAMPLE.cls6.1	C0	5	101	20144.6	87457.2	1.5012	詳細 プラン表示
%sqlcq.MYSAMPLE.cls7.1	QRS0	3	17	46877.67	243189.67	4.0452	詳細 プラン表示
%sqlcq.MYSAMPLE.cls8.1	QRS0	45	67	0	51	0.0008	詳細 プラン表示
%sqlcq.MYSAMPLE.cls9.1	QRS0	3	2.33	51	1250.33	0.0076	詳細 プラン表示

SELECT * FROM mydemo . table1 WHERE state = ? AND gender = ?

クエリプラン

相対コスト = 3068328

[stats] Time in Module MAIN = 0.031 Module Execution Count = 1 Global References = 3,019 Commands Executed = 83,757 Number of Rows = 1,001

- Read index map MyDemo.Table1.StateGenderIdx, using the given %SQLUPPER(State) and %SQLUPPER(Gender), and looping on ID.
- For each row:
Read master map MyDemo.Table1.IDKEY, using the given idkey value.
Output the row.

INTERSYSTEMS

20

You'll make breakthroughs

(参考)

より選択性の高い場合はビットマップが有効

以下のテーブルには SQL 統計情報を含むすべてのクエリの情報が表示されています。 [テーブルの再読み込み](#)

ルーチン	カーソル	実行回数	平均行	平均グローバル参照	平均コマンド	平均時間	
CacheSqlStats4736	SQLStats0	1	6737	13507	337701	1.6578	詳細 プラン表示
CacheSqlStats5796	SQLStats0	1	6737	13507	337701	1.6556	詳細 プラン表示
%sqlcq.MYSAMPLE.cls1.1	1A	1	0	32980	9449853	72563	詳細 プラン表示
%sqlcq.MYSAMPLE.cls2.1	1A	2	999349.5	5997971.6	214079372	33.39958	詳細 プラン表示
%sqlcq.MYSAMPLE.cls3.1	QRS0	7	1	27.14	672.67	0.0332	詳細 プラン表示
%sqlcq.MYSAMPLE.cls5.1	QRS0	2	1	18.5	423	0.0027	詳細 プラン表示
%sqlcq.MYSAMPLE.cls11.1	1A	1	9993	79906	9672947	7922	詳細 プラン表示
%sqlcq.MYSAMPLE.cls12.1	QRS0	5	178.4	401	16671.4	0.0936	詳細 プラン表示

Stateビットマップ & Genderビットマップ

以下のテーブルには SQL 統計情報を含むすべてのクエリの情報が表示されています。 [テーブルの再読み込み](#)

ルーチン	カーソル	実行回数	平均行	平均グローバル参照	平均コマンド	平均時間	
CacheSqlStats4736	SQLStats0	1	6737	13507	337701	1.6578	詳細 プラン表示
CacheSqlStats5796	SQLStats0	1	6737	13507	337701	1.6556	詳細 プラン表示
%sqlcq.MYSAMPLE.cls1.1	QRS0	4	183.25	39746.5	422897.5	0.9984	詳細 プラン表示

State 通常のインデックス

INTERSYSTEMS

21



You'll make breakthroughs

クエリのチューニング

TuneTableユーティリティ

- TuneTableは、テーブルの実データからテーブルのサイズや、データの特性を記録し、クエリ・プラン生成の元となるパラメータを計算する。
- \$SYSTEM.SQL.TuneTable() コマンド、もしくは、管理ポータルから実行
- 実データが存在する状態で実行
- TuneTableの結果をエクスポートし、他の環境にインポートすることも可能
- パラメータ
 - EXTENTSIZE : テーブルに含まれるデータの行数
 - SELECTIVITY : フィールドの値を指定することで選択される行の割合（性別なら50%など..）
 - BLOCKCOUNT: 各マップ(Master, Index1, ...)のブロック数

23

INTERSYSTEMS®

クエリのチューニング

「外れ値」のSELECTIVITY

あるフィールドで、
大きく偏りのある値を検出

右の例では、State='MA'が全体の33%
その他の値のSELECTIVITY=1.37%

SQLテーブル・チューニング

MyDemo.Table1

テーブルチューニングは、テーブルのデータを参照して、エクステントサイズ(テーブルに何行含まれるか)と、各
ロックカウントも計算します。

閉じる

選択性 マップ・ブロックカウント

現在のテーブルのエクステント・サイズ: 999402 編集 テーブルチューニング

クラスを最新状態に保つ: ☐

ページサイズ: 20 最大行数: 1000 結果: 5 ページ: 1 の 1

フィールド名	選択性	備考	外れ値の選択性	外れ値
Gender	50.0000%			
ID	1	RowID field		
SSN	1			
State	1.3705%		32.8443%	"MA"
x_classname	0.0023%	Hidden field	99.9767%	<Null>

INTERSYSTEMS

24

ヒント

- JOINの順序
 - インデックスの選択
 - サブクエリ
 - UNION / OR
-
- 本来は、SQLで書かなくても、Cache Object Scriptで...

INTERSYSTEMS

25

(参考) サブクエリ

```
SELECT g.city, t.month, AVG(s.profit) FROM geog g, sales s, time t
WHERE
    s.gid = g.gid AND s.tid = t.tid AND
    s.pid NOT EXISTS (SELECT p.pid FROM products p
                      WHERE s.pid = p.pid AND p.color = 'Pink')
GROUP BY g.city, t.month;
```

- サブクエリが、salesテーブルに含まれる p.pidの数だけ繰り返される可能性

ヒント

- %ALLINDEX – JOIN順の最初のテーブルに対して、適用可能なすべてのインデックスを使用する。WHERE句に%NOINDEXを指定して、明示的に例外を指定可能。
-
- %NOFLATTEN – サブクエリのFROM句に指定して、サブクエリを“メインクエリ”の条件に変換することを抑止する。
-
- %NOSVSO – サブクエリのFROM句に指定して、“Set-Valued Subquery Optimization”を抑止する。SVSO: [NOT] EXIST または[NOT] IN サブクエリの条件から、一時インデックスを作成して、“メインクエリ”の条件指定に変換する。
- %NOUNIONOROPT – OR ⇔ サブクエリでのUNION 変換による最適化を抑止する。
-
- %NOTOPOPT – TOP ... ORDER BYにおいて、最初の行への時間を優先する最適化を抑止し、全体の結果セット取得時間を優先する。

You'll make breakthroughs

ヒント

- %FIRSTTABLE – 指定したテーブルを最初にJOINする。
- %FULL – すべてのJOIN順を計算し、最適なプランを算出する (コンパイル時間が長くなる)。ストアードプロシージャ内のクエリなどで有効な場合がある。
-
- %IGNOREINDEX – 指定されたインデックスを使わない。
-
- %INORDER – FROM句に指定されたテーブルの順序でJOINする。
- %STARTTABLE – FROM句の最初のテーブルからJOINする。
- %NOINDEX – 指定したWHERE句に対して、インデックスを使用しない。

INTERSYSTEMS®

28



You'll make breakthroughs

INTERSYSTEMS®

ビットマップインデックスの圧縮

29

You'll make breakthroughs

ビットマップインデックスが遅くなる理由

- グローバル変数には、64000行分のビットが含まれる。
- 約8KBを占める。
- 1行削除すると、1のビットを0に変更する。
- すべてのビットが0になれば、その「チャンク」は必要ない。
- 検索実行時、このようなすべてが0の「チャンク」からも、1のビットを探す。
- このような0の「チャンク」は、ビットマップのすべて値について積み重なる。

INTERSYSTEMS

ビットマップインデックス

```

1:  "MyDemo.Table11("StateIdx", "AK", 1) = 001000100.....10011 (64000ビット)
2:  "MyDemo.Table11("StateIdx", "AK", 2) =
3:  "MyDemo.Table11("StateIdx", "AK", 3) = $zwc(405,7998,37,42,66,69,95,117,154,198,298,375,415,436,487,562,587,591,627,713,953,971,1003,1202,1242,
4:  "MyDemo.Table11("StateIdx", "AK", 4) = $zwc(406,8000,40,48,53,81,140,201,213,241,258,344,470,527,587,588,685,845,983,1106,1167,1204,1212,1296,1
5:  "MyDemo.Table11("StateIdx", "AK", 5) = $zwc(402,7999,8,42,46,59,117,157,210,266,336,440,475,531,619,644,646,671,721,782,786,789,801,930,970,987
6:  "MyDemo.Table11("StateIdx", "AK", 6) = $zwc(405,7990,252,315,444,488,608,612,702,703,705,720,868,1115,1129,1222,1274,1345,1395,1399,1516,1537,1
7:  "MyDemo.Table11("StateIdx", "AK", 7) = $zwc(407,7995,44,247,250,251,268,297,304,337,462,502,531,539,569,625,661,674,720,733,826,846,847,870,967
8:  "MyDemo.Table11("StateIdx", "AK", 8) = $zwc(401,7991,13,39,60,202,217,225,257,264,274,299,421,456,555,576,577,653,683,791,803,823,852,877,879,9
9:  "MyDemo.Table11("StateIdx", "AK", 9) = $zwc(402,7996,75,235,270,285,315,352,357,443,454,469,494,497,531,564,580,682,755,776,1154,1184,1207,1223
10: "MyDemo.Table11("StateIdx", "AK", 10) = $zwc(407,8000,29,129,181,193,268,278,310,312,392,408,410,437,463,477,548,671,672,794,912,1058,1084,1209,
11: "MyDemo.Table11("StateIdx", "AK", 11) = $zwc(402,8000,73,78,252,338,352,410,422,513,528,611,757,802,939,976,1046,1067,1127,1179,1242,1252,1259,1
12: "MyDemo.Table11("StateIdx", "AK", 12) = $zwc(403,7999,14,15,20,58,127,381,449,489,490,511,515,516,670,716,748,779,787,794,889,935,988,991,1030,1
13: "MyDemo.Table11("StateIdx", "AK", 13) = $zwc(400,7993,8,28,86,155,258,270,310,332,448,473,494,539,625,673,786,1127,1227,1253,1342,1368,1457,1500
14: "MyDemo.Table11("StateIdx", "AK", 14) = $zwc(402,7995,61,63,306,371,402,440,592,606,699,771,847,890,940,955,971,1033,1095,1205,1240,1354,1380,14
15: "MyDemo.Table11("StateIdx", "AK", 15) = $zwc(403,7998,7,110,130,171,317,463,464,553,569,572,628,662,733,736,775,807,817,818,826,1066,1149,1157,1
16: "MyDemo.Table11("StateIdx", "AK", 16) = $zwc(401,5005,23,69,95,141,186,212,255,286,361,391,409,433,503,534,545,561,623,688,727,767,1011,1062,110
17: "MyDemo.Table11("StateIdx", "AL", 1) = $zwc(406,7981,37,104,216,227,365,398,432,487,553,558,637,663,677,809,943,1020,1030,1081,1234,1252,1313,1
18: "MyDemo.Table11("StateIdx", "AL", 2) = $zwc(401,7996,23,43,128,158,170,208,271,327,337,394,429,469,556,560,598,636,671,680,741,748,871,981,1004
19: "MyDemo.Table11("StateIdx", "AL", 3) = $zwc(403,7998,36,129,145,148,209,263,416,425,431,474,475,506,586,672,703,707,727,740,843,903,958,1015,10
20: "MyDemo.Table11("StateIdx", "AL", 4) = $zwc(400,8000,106,291,300,345,601,640,649,785,813,855,932,993,1013,1016,1029,1037,1060,1087,1088,1193,12
21: "MyDemo.Table11("StateIdx", "AL", 5) = $zwc(406,7995,38,118,150,202,204,233,258,268,269,299,520,522,553,564,661,752,876,905,960,982,989,1024,11
22: "MyDemo.Table11("StateIdx", "AL", 6) = $zwc(402,7997,107,217,219,281,288,325,328,411,461,737,768,772,782,793,824,849,852,855,864,989,1071,1188,
23: "MyDemo.Table11("StateIdx", "AL", 7) = $zwc(405,7997,95,129,157,163,264,300,317,349,382,402,518,555,577,579,624,689,746,812,861,863,929,940,944
24: "MyDemo.Table11("StateIdx", "AL", 8) = $zwc(405,7993,80,136,141,242,263,521,545,595,657,703,731,736,756,780,816,884,916,939,1024,1145,1283,1357

```

INTERSYSTEMS

You'll make breakthroughs

ビットマップインデックス

0ばかりのチャンク

```

1: "MyDemo.Table11("StateIdx","AK",1) = 0000000000.....000000 (64000ビット)
2: "MyDemo.Table11("StateIdx","AK",2) = $zwc(400,7991,15,197,203,225,259,272,362,367,419,521,613,721,754,842,889,904,950,979,1060,1184,1214,1296
3: "MyDemo.Table11("StateIdx","AK",3) = $zwc(405,7998,37,42,66,69,95,117,154,198,298,375,415,436,487,562,587,591,627,713,853,971,1003,1202,1242,
4: "MyDemo.Table11("StateIdx","AK",4) = $zwc(406,8000,40,48,53,81,140,201,213,241,258,344,470,527,587,588,685,845,983,1106,1167,1204,1212,1296,1
5: "MyDemo.Table11("StateIdx","AK",5) = $zwc(402,7999,8,42,46,59,117,157,210,266,336,440,475,531,619,644,646,671,721,782,786,789,801,930,970,987
6: "MyDemo.Table11("StateIdx","AK",6) = $zwc(405,7990,252,315,444,488,608,612,702,703,705,720,868,1115,1129,1222,1274,1345,1395,1399,1516,1537,1
7: "MyDemo.Table11("StateIdx","AK",7) = $zwc(407,7995,44,247,250,251,268,297,304,337,482,502,531,539,569,625,661,674,720,733,826,846,847,870,967
8: "MyDemo.Table11("StateIdx","AK",8) = $zwc(401,7991,13,39,60,202,217,225,257,264,274,299,421,456,555,578,577,653,663,791,803,823,852,877,879,9
9: "MyDemo.Table11("StateIdx","AK",9) = $zwc(402,7996,75,235,270,285,315,352,357,443,454,469,494,497,531,564,580,682,755,776,1154,1184,1207,1223
10: "MyDemo.Table11("StateIdx","AK",10) = $zwc(407,8000,29,129,181,193,268,278,310,312,392,408,410,437,463,477,548,671,672,794,912,1058,1084,1209,
11: "MyDemo.Table11("StateIdx","AK",11) = $zwc(402,8000,73,78,252,338,352,410,422,513,528,611,757,802,939,976,1046,1067,1127,1179,1242,1252,1259,1
12: "MyDemo.Table11("StateIdx","AK",12) = $zwc(403,7999,14,15,20,58,127,381,449,469,490,511,515,516,670,716,748,779,787,794,869,935,988,991,1030,1
13: "MyDemo.Table11("StateIdx","AK",13) = $zwc(400,7993,8,28,86,155,258,270,310,332,448,473,494,539,625,673,786,1127,1227,1253,1342,1368,1457,1500
14: "MyDemo.Table11("StateIdx","AK",14) = $zwc(402,7995,61,63,306,371,402,440,592,606,699,771,847,890,940,955,971,1033,1095,1205,1240,1354,1380,14
15: "MyDemo.Table11("StateIdx","AK",15) = $zwc(403,7998,7,110,130,171,317,463,464,553,569,572,628,662,733,736,775,807,817,818,826,1066,1149,1157,1
16: "MyDemo.Table11("StateIdx","AK",16) = $zwc(401,5005,23,69,95,141,186,212,255,286,361,391,400,423,503,534,545,561,623,688,727,767,1011,1062,110
17: "MyDemo.Table11("StateIdx","AL",1) = 0000000000.....000000 (64000ビット)
18: "MyDemo.Table11("StateIdx","AL",2) = $zwc(401,7996,23,45,126,156,170,208,271,327,337,394,429,469,556,590,596,636,671,680,741,746,671,981,1004
19: "MyDemo.Table11("StateIdx","AL",3) = $zwc(403,7998,36,129,145,148,209,263,416,425,431,474,475,506,586,672,703,707,727,740,843,903,958,1015,10
20: "MyDemo.Table11("StateIdx","AL",4) = $zwc(400,8000,106,291,300,345,601,640,649,785,813,855,932,993,1013,1016,1029,1037,1060,1087,1088,1193,12
21: "MyDemo.Table11("StateIdx","AL",5) = $zwc(406,7995,38,118,150,202,204,233,258,268,269,299,520,522,553,564,661,752,876,905,960,982,989,1024,11
22: "MyDemo.Table11("StateIdx","AL",6) = $zwc(402,7997,107,217,219,281,288,325,328,411,461,737,768,772,782,793,824,849,852,855,864,969,1071,1188,
23: "MyDemo.Table11("StateIdx","AL",7) = $zwc(405,7997,95,129,157,163,264,300,317,349,382,402,518,555,577,579,624,689,746,812,861,863,929,940,944
24: "MyDemo.Table11("StateIdx","AL",8) = $zwc(405,7993,80,136,141,242,263,521,545,595,657,703,731,736,756,780,816,894,916,939,1024,1145,1283,1357

```

INTERSYSTEMS

32

You'll make breakthroughs

ビットマップ圧縮ユーティリティ

%SYS.Maint.Bitmap

- Cache 2014.2の新機能
- 稼働中のシステムでも動作
- ビットマップを圧縮もしくは、不要なチャンクを削除
- メソッド:
 - d ##class(%SYS.Maint.Bitmap).Namespace("Samples",1,1,"2014-01-17 09:00:00")
 - d ##class(%SYS.Maint.Bitmap).OneClass("BitMap.Test",1,1)

INTERSYSTEMS

You'll make breakthroughs

ビットマップ圧縮の例

SAMPLES>d ##class(GS.Compact).Populate(1000000)

Global - GS.CompactI Blocks - 34 Bytes - 250,901

1. DELETE FROM GS.Compact WHERE ID < 750000

Global - GS.CompactI Blocks - 46 Bytes - 344,747

SAMPLES>d ##class(%SYS.Maint.Bitmap).OneClass("GS.Compact",1,1)

Class: GS.Compact Start Time: 2014-03-13 14:45:29

Global: ^GS.CompactI("\$Compact") was compressed: 93.87 %

Global: ^GS.CompactI("GenderIdx") was compressed: 70.48 %

Compression time in seconds: 0

Global - GS.CompactI Blocks - 12 Bytes - 80,066

INTERSYSTEMS®



You'll make breakthroughs



インデックス追加・再構築の新機能

35

You'll make breakthroughs

“ホット インデキシング”

稼働中のシステムにインデックスを追加する

- 稼働中システムにインデックスを追加するのはいくつかの問題がある
 - クエリが有効でなくなる
 - インデックスの再構築には時間がかかる
 - ダウンタイムの必要
- “ホット インデキシング”はこれらの問題を解決する
 - インデックス追加時でも、クエリは有効
 - インデックスの構築を高速に
 - ユーザはシステムを継続的に使用できる
- 2つの新技術
 - マップの選択可・不可の設定
 - 高速なインデックス構築

INTERSYSTEMS

You'll make breakthroughs

マップの選択可・不可を設定する

`$SYSTEM.SQL.SetMapSelectability()`

- コンパイラに、指定インデックスを無視させる
- 間違った検索結果を心配することなく、構築中のインデックスを保持できる
- INSERT, UPDATE, DELETEで、インデックスは最新状態に保たれる

INTERSYSTEMS

You'll make breakthroughs

インデックスを高速に構築する

新しいAPI

- 新しいクラス: %Library.IndexBuilder
 - 新機能を使うために継承する
- パラメータ: INDEXBUILDERFILTER
 - クラスに追加して、新しいインデックス名を指定する
- API:
 - Do ##class(Package.Class).%ConstructIndicesParallel()

INTERSYSTEMS

You'll make breakthroughs

インデックスを高速に構築する

%ConstructIndicesParallel

- インデックスの構築に、複数のジョブを利用する。
- インデックスの構築対象となるIDの範囲を指定できる。
 - 新しい値はインデックスする必要がない
- ジャーナルをOFFにすることにより、インデックス構築のスピードを向上させることができる。

INTERSYSTEMS

You'll make breakthroughs

稼働中システムへの新しいインデックスの追加

2014.1の場合

1. クラスのスーパークラスに%Library.IndexBuilderを追加
 2. INDEXBUILDERFILTERパラメータをクラスに追加
 3. インデックスを追加
 4. このマップのselectabilityを0に
 - w \$SYSTEM.SQL.SetMapSelectability(クラス名,インデックス名,0)
 5. コンパイル
 6. インデックスを構築
 7. このマップのselectabilityを1に
 - W \$SYSTEM.SQL.SetMapSelectability(クラス名,インデックス名,1)
 8. 再コンパイル
- この手順は%Dictionary.ClassDefinitionを使って、プログラムの実行可能

INTERSYSTEMS

You'll make breakthroughs

“ホット インデキシング”の将来

2014.2では

1. %BuildIndices()が、%IndexBuilderの機能を受け継ぐ
2. %IndexBuilderクラスの継承や、INDEXBUILDERFILTERパラメータの追加は必要なくなる

INTERSYSTEMS

You'll make breakthroughs

その他の最新機能 (2014.1)

- Unified trigger
 - SQLトリガー(Insert/Update/Delete)が、Cache Objectアクセス経由でも実行される。
 - %On~ メソッドは引き続きサポートされる。
- INSERT or UPDATE
 - INSERT OR UPDATE文は、レコードが存在しない場合はINSERT、存在する場合はUPDATEと同様の動作を行う。
 - レコードの存在チェックは、Uniqueキーによる判断される。

INTERSYSTEMS

You'll make breakthroughs

その他の最新機能 (2014.1)

- Push Subquery Conditions into UNION Legs

```
SELECT Name, Age FROM
  (SELECT Name, Age FROM Sample.Person
   UNION SELECT Name, Age FROM Sample.Employee)
WHERE Age IN (SELECT TOP 5 Age FROM Sample.Employee WHERE Age > 55 ORDER BY Age)
```

•

INTERSYSTEMS

パラレル クエリ

```

- %PARALLEL
SELECT SUM(amountofsale) FROM %PARALLEL holefoods.salestransaction
Query Plan (Relative cost = 3560400)
Call module B.
Output the row.
module B
Read master map HoleFoods.SalesTransaction.IDKEY, looping on ID. For each row:
  Read master map HoleFoods.SalesTransaction.IDKEY, looping on ID.
  For each row:
    Send query to a separate process, piping results to temp-file A:
    SELECT sum(T1.AmountOfSale) FROM %NOPARALLEL HoleFoods.SalesTransaction T1
    WHERE T1.ID<=?
    Send query to a separate process, piping results to temp-file A:
    SELECT sum(T1.AmountOfSale) FROM %NOPARALLEL HoleFoods.SalesTransaction T1
    WHERE T1.ID>? AND T1.ID<=?
    Send query to a separate process, piping results to temp-file A:
    SELECT sum(T1.AmountOfSale) FROM %NOPARALLEL HoleFoods.SalesTransaction T1
    WHERE T1.ID>? AND T1.ID<=?
    Send query to a separate process, piping results to temp-file A:
    SELECT sum(T1.AmountOfSale) FROM %NOPARALLEL HoleFoods.SalesTransaction T1
    WHERE T1.ID>?
  Read temp-file A, looping on a counter.
  For each row:
    Accumulate the sum([value]).

```

44

INTERSYSTEMS

Q & A

45

INTERSYSTEMS

You'll make breakthroughs

INTERSYSTEMS®

Caché SQL 最新情報

