ＳＱＬ文の解析数を減らす方法
（類似 SQL 文に対して、同じ SQL 実行計画を使用させる判定基準レベル設定）

――― SQL 文の解析数削減 ―――――

ＳＱＬ文の解析数を減らす方法は、SQL 文のテキストを完全に一致させるしかない。

――― 類似 SQL 文に対して、同じ SQL 実行計画を使用させるための判断基準 ―――――

解析したＳＱＬ文の実行計画はキャッシュに保存されるが、類似した SQL 文に対し同一の実行計画と見なされれば、SQL 文の解析を行わずにキャッシュの実行計画を使った効率的な処理が行われる

また、類似したＳＱＬ文の実行計画の１つだけをキャッシュに保存するようにすれば、キャッシュでの実行計画の登録数の総数は同じでも、登録してある実行計画の種類数 および、対応する SQL 文の種類数（SQL 文→実行計画）が増えることになり、SQL 文の類似~~完全~~一致によるキャッシュ使用のヒット率が向上する

キャッシュでの SQL 文の共有の判断基準

【設定方法】

初期化パラメータ CURSOR_SHARING　=

EXACT　：構文検査を行い、SQL 文が完全に一致するときのみ、実行計画はキャッシュの内容を使用する
デフォルト値
オプティマイザによる実行計画の策定が省略される。
SQL 文のキャッシュでの実行計画の共有は無し

SIMMILAR　：構文検査を行い SQL 文が完全に一致しないときには、常にオプティマイザによる実行計画の策定が行われる
実行計画が一致すれば、既存の SQL 文と同一とみなして、キャッシュでの実行計画を共有する
異なる SQL 文とで実行計画を共有するので、登録してある SQL 文が増加し、次回の SQL 文処理の時に SQL 文の完全一致の割合が高くなる

FORCE　：構文検査を行った段階で SQL 文が似ていれば、キャッシュの実行計画を使うようにして、オプティマイザによる実行計画の策定を行わない
実行計画の共有が無くても、「判断基準が緩い」ので実行計画のキャッシュヒット率が向上し、オプティマイザの処理数は、減少する

――― SQL 文の共有のメリット ―――――

例えば FORCE 指定を行うと、バインド変数を使っていなくて、リテラルだけが異なる SQL を同一の実行計画とみなし、解析処理なしで SQL 文の実行を行います

――― SQL 文の解析処理 ―――

解析手順（SQL 文の使用する実行計画の決定手順）

１）構文の文法検査

２）キャッシュの SQL 文からの検索

３-A）２）の結果が、キャッシュにヒットした場合
ソフト解析：実行計画はキャッシュの内容を使用

３-B）２）の結果が、キャッシュにヒットしなかった場合
ハード解析：オプティマイザによる実行計画の新規作成

※ Force を指定した場合には、ソフト解析の可能性がより高くなる

ライブラリ・キャッシュ中に実行計画が保管されている Select 文の一覧

```
SELECT　SQL_TEXT　FROM　V$SQLAREA　;
```

実行された Select 文の実行計画のキャッシュヒット率

```
SELECT　SUM( pinhits ) / sum ( pins ) * 100　AS　"ヒット率"
　　FROM　v$librarycache　;
```