

## リアルタイム監視

### パイプコマンド “|” が行う機能

### WSH シェルから SQLPLUS の起動方法

### WSH シェルから SQL コマンドの発行

## パイプコマンド “|” が行う機能

続・門外不出の Oracle 現場ワザ

プログラム A | プログラム B > 出力ファイル名

- ・ パイプコマンド（|）は、左側プログラムでは、標準出力の宛先がスプールに指定されたことになる  
なので、標準出力に出力した内容がスプールへ貯めこまれていく
- ・ パイプコマンド（|）は、右側プログラムでは、標準入力宛先の宛先がスプールに指定されたことになる  
なので、標準入力からの入力指示は、貯めこまれているスプールの内容がとりこまれていく

### 【重要】

プログラム A とプログラム B は、**並行して動作し、同時に処理が行われている**

プログラム A が終了してから、プログラム B が動作を開始するのではない

よって、プログラム A がスプールへ出力された内容は、**随時にプログラム B へ受け渡され**て順次処理されていくことになる

- ・ 不等号コマンド（>）は、右側プログラムでは、標準出力がファイル宛てに指定されたことになる  
これにより、ファイルへ標準出力への出力内容が書き込まれていく

> : 上書きモード

>> : 追加書き込みモード

## WSH シェルから SQLPLUS の起動方法

### WSH シェルから SQL コマンドの発行

```
cscript.exe //NOLOGO スクリプト・ファイル名.vbs |  
    sqlplus.exe -s /nolog > .¥出力ファイル名.txt
```

パイプコマンドの左側では、

`cscript.exe` プログラムが、コマンドライン引数で指定されたファイルのスクリプトの内容で、1 ステップずつ処理を実行する

ここで、**WScript.echo** で出力された内容が、その都度スプールへ出力されていく（1 ステップごと）

なお、スクリプト・ファイルは、`cscript.exe` が処理する内容を記述してある設計書の役割である。（実行モジュールではない）

パイプコマンド（`|`）の右側では、

`sqlplus.exe` プログラムが、コマンドライン引数で指定された接続情報に基づき Oracle に接続したのち、上記の処理によってスプールで受けわたってきた内容を Oracle サーバー（データベース）へ、随時 SQL コマンドとして送信し、データベースでの処理を進めていくことになる

そして、`DBMS_OUTPUT.PUT_LINE(・・・)` コマンドの内容が、スプールへ出力され、スプール経由でファイルへ書き出されていく

#### 【 重要 】

`cscript.exe` プログラムと `sqlplus.exe` プログラムは、**並行して動作し、同時に処理が行われている**

`cscript.exe` プログラムが終了してから、プログラム B が動作を開始するのではない

使用例)

```
cscript.exe //NOLOGO RealTime_Check.vbs |  
    sqlplus.exe -s /nolog > .¥RealTime_Check_Start_DATA.txt
```

## リアルタイム監視

データベースの稼働状況（リソース負荷）をリアルタイムに監視する方法

### 概要

V\$動的ビューの内容を SQL 文にて調査



テキストファイルへ書き出し



テキストファイルに新しく追加された行をリアルタイムに表示する



Excel にて、グラフ表示を行う

### サンプルファイルの保存場所

¥Oracle¥Book\_Data¥続・門外不出の Oracle 現場ワザ¥script  
¥リアルタイム監視スクリプト¥

### 説明

実行した SQL 文の内容を、画面に随時出力していく

### ファイル構成とファイル関連

\_1 リアルタイム監視の開始.bat

実行 SQL 文ファイル

```
cscript.exe //NOLOGO RealTime_Check.vbs |  
sqlplus.exe -s /nolog >  
.¥RealTime_Check_Start_DATA.txt
```

結果出力ファイル

入力ファイル

```
start PowerShell.exe Get-Content ".¥RealTime_Check_Start_DATA.txt"  
-wait -tail 10
```

画面出力表示を行う