SuperSQL利用マニュアル

小谷 美紗登　田畑 篤智　藤本 樹　照井 恵太　五嶋 研人
遠山元道

データモデリング　2018年7月5日

目 次

1　はじめに

このプリントでは、遠山研究室で開発した SuperSQL を理工学 ITC で利用するための設定とコマンドについて説明してあります。課題の時などにこのプリントを参考にして下さい。

2　初期設定・確認

SuperSQL を利用するためには以下の設定をする必要があります。

2.1　作業ディレクトリ

各自のホームディレクトリに public_html/ssql が作成されているか確認してください。以降の設定でそのフォルダには生成された HTML ファイルが入ります。また、ホームディレクトリ以下に SQL フォルダを作成しそこに実行する sql ファイルや ssql ファイルを配置するようにしましょう (すでに作成済みだったらそれを使いましょう)。このように実行ファイルを格納するフォルダ (/SQL) と実行の結果生成されるファイルを格納するフォルダ (/public_html/ssql) に区別して作業を進めていきましょう。

2.2　SSedit 設定

SSedit を起動 (3.1 参照) し、タブの一番右端にある 設定 にて各種設定で下記の内容が書かれているか確認して下さい。

- ドライバー名は ‘postgresql’
- データベース名、ユーザ名は自分のアカウント名 (ua123456 等)
- ホスト名は ‘131.113.101.124’
- 出力先は ∼/public_html/ssql

これで、SuperSQL を実行して生成される HTML ファイルが ∼/public_html/ssql 以下に保存されます。また、http://user.keio.ac.jp/∼ 自分のアカウント/ssql/xxxx.html という URL でアクセスできるようになります。

例えば、アカウントが ua012345 のユーザが test.ssql というファイルを ssql コマンドで実行した場合、
http://user.keio.ac.jp/∼ua012345/ssql/test.html
という URL でアクセスできます。

3　利用方法

3.1　SSedit の使い方

(リモートログインの場合は使用できません。)

ua012345% ssedit

上記のコマンドを実行すると、SuperSQL クエリ実行ツールが起動します。

クエリを実行：窓に書き込まれたクエリを実行
フォルダを指定して実行：指定されたフォルダ内の全ファイルを実行
設定：出力先・config ファイルの内容などを指定・変更

実行結果のログは、実行ファイル（.ssql）が保存されているディレクトリに log.txt(または logs.txt) という名前で作成されます。

※ SSedit の詳しい使い方については、別紙の SSedit 利用マニュアルを参照してください。

3.2 ssql コマンドの使い方

(自宅などから Putty でリモートログインして使う場合は、SSedit が使えないので、こちらの方法で行ってください)

設定が終わったら実際にコマンドを入力してみましょう。コマンドの入力方法は、

```
ua012345% ssql 質問ファイル
```

です。このコマンドで、'質問ファイル' を読込み、データベースに問合せをした後、HTML ファイルを ～/public_html/ssql に出力してくれます。
質問ファイルの拡張子は .ssql とします。

4 質問文の書き方

SuperSQL の質問文は、SQL の SELECT 句を GENERATE$< media >< TFE >$ の構文を持つ GENERATE 句で置き換えたものです。ここで $< media >$ は出力媒体を示し、HTML、PDF、Excel、LaTeX などの指定が可能ですが、実習では HTML のみを利用します。$< TFE >$ はターゲットリストの拡張である Target Form Expression を表し、結合子、反復子などのレイアウト指定演算子を持つ一種の式です。

4.1 結合子

結合子はデータベースから得られたデータをどの方向に結合するかを指定する演算子であり、コンマ (,)、感嘆符 (!)、パーセント記号 (%) の 3 種類あります。属性間をこれらの結合子で区切ることによって、それぞれ水平、垂直、深度方向にレイアウトします。 以下は結合子の使用例です。

- 水平結合子 (,)
  データを横に結合して出力します。
  例：Name, Tel

山田花子	03 - 1111 - 2222

- 垂直結合子 (!)
  データを縦に結合して出力します。
  例：Name! Tel

山田花子
03 - 1111 - 2222

- 深度結合子 (%)
  データをリンク方向に結合して出力します。
  例：Name % Tel

山田花子	→	03 - 1111 - 2222

4.2 反復子

反復子は指定する方向に、データベースの値があるだけ繰りかえして表示します。一対の角括弧 ([ ]) に上記の結合子を添えたものが反復子です。以下は反復子の使用例です。

- 水平反復子 ([ ],)
  インスタンスがある限り、データを指定した構造 (この場合は横方向) に反復して出力します。

⟨**Q1.ssql**⟩
```
GENERATE HTML
  [e.name!  e.byear],
FROM employee e
```

鈴木一郎	鈴木二郎	⋯	鈴木十郎
1978	1980	⋯	1990

- 垂直反復子 ([ ]!)
  インスタンスがある限り、データをデータを指定した構造 (この場合は縦方向) に反復して出力します。

⟨**Q2.ssql**⟩
```
GENERATE HTML
  [e.name, e.byear]!
FROM employee e
```

鈴木一郎	1978
鈴木二郎	1980
⋯	⋯
鈴木十郎	1990

4.3 反復子の入れ子

反復子をただ並置した場合には、下記のようにただ各々の一覧が表示されるだけになります。

⟨**Q3.ssql**⟩
```
GENERATE HTML
  [d.name]!  , [i.name]!  , [i.price]!
FROM item i, dept d
WHERE i.dept = d.id
```

駄菓子屋	子供用デニムスーツ	198
子供服	迷路のおもちゃ	800
寝具	300g 入りのあめ	3295
…	…	…

そこで反復子の入れ子を作る事で、その入れ子の関係によって属性間の関連を指定できます。例えば、

⟨**Q4.ssql**⟩
```
GENERATE HTML
  [d.name![i.name, i.price]!]!
FROM item i, dept d
WHERE i.dept = d.id
```

駄菓子屋	
300g 入りのあめ	500
600g 入りのあめ	800
子供服	
ベルボトムジーンズ	3000
子供用デニムスーツ	6000
…	…

とした場合には、売り場ごとの商品の名前と値段の一覧が上記のように表示されます。

4.4 複合反復子

反復子を組み合わせ使用したい場合には上記のような構文があります。
Q5.ssql のようにクエリを書くと、e.name を横に 3 つごとに結合し、改行して出力します。

構文	動作	例
[*TFE*],*number*!	横に *number* 個結合し、次の行に改行	[name],5!
[*TFE*]!*number*,	縦に *number* 個結合し、次の列に改行	[name]!5,
[*TFE*],*number*%	横に *number* 個結合し、ページ切り替え	[name],5%
[*TFE*]!*number*%	縦に *number* 個結合し、ページ切り替え	[name]!5%
[*TFE*],*number1*!*number2*%	横に *number1* 個結合し、次の行に改行し、*number2* 行分出力後、ページ切り替え	[name],5!4%
[*TFE*]!*number1*,*number2*%	縦に *number1* 個結合し、次の列に改行し、*number2* 列分出力後、ページ切り替え	[name]!5,4%

⟨**Q5.ssql**⟩

```
GENERATE HTML
  [e.name],3!
FROM employee e
```

Q6.ssql のようにクエリを書くと、下の表のように横並びした e.name と e.salary を 3 つ横に結合して次の行に改行し、5 行分出力後にページ切り替えすることができます。

⟨**Q6.ssql**⟩

```
GENERATE HTML
  [e.name, e.salary],3!5%
FROM employee e
```

長尾美和子	11985	浅川陽子	1800	広田美智子	1300
江藤俊介	6000	福島裕次郎	12000	岩井直也	17000
平河修	17674	畑千佐子	8000	越智美穂	7500
橋本香澄	12000	清水麻里	12067	高場詩織	11013
会津雅史	19000	金子巧	13000	鈴木紀男	9000

前へ 1 2 3 4 次へ

4.5 文字列の表示

SuperSQL では、文字列を直接表示し、表の見出しなどを作ることができます。表示したい文字列をシングルクオテーションで囲み、TFE と結合子で連結します。以下は使用例です。

例 1 :‘名前’! [Name]!

名前
鈴木一郎
…
鈴木十郎

例 2 : {‘名前’, Name}! {‘電話’, Tel}

名前	山田花子
電話	03 - 1111 - 2222

4.6 FROM 句と WHERE 句

SuperSQL 質問文の FROM 句と WHERE 句の記述方法は SQL と同じで、下記の様になります。

```
FROM dept d, store s
WHERE d.store = s.id
```

5 装飾子

SuperSQL では関係データベースより抽出された情報に、文字サイズ、文字スタイル、横幅などの情報を付加することができます。これらは装飾演算子 (@) によって指定します。

```
<TFE>@{ <装飾指定> }
```

装飾指定は項目 = ‘値’ とし、複数指定するときは各々を ‘,’ で区切ります。指定できる項目は以下の通りです。

- セル幅: width (px)
  例: width=200
- セル高: height (px)
  例: height=20
- パディング (余白): padding (px)
  例: padding=2
- 横位置: align(left/center/right)
  例: align=‘left’
- 縦位置: valign(top/center/bottom)
  例: valign=‘top’
- 背景色: background-color, bgcolor(直接色を指定する方法と、カラーコードを指定する方法の 2 つがあります)
  例: bgcolor=‘red’/bgcolor=‘FF0000’
- 文字色: color, font-color
  例: color=‘red’/color=‘FF0000’
- 文字サイズ: size, font-size(px)
  例: font-size=20
- 文字の太さ: font-weight
  例: font-weight=‘bold’

- 文字のスタイル: font-style
  例: font-style=‘italic’
- CSS の指定（セミコロン区切りで複数指定可能）: style
  例: style=‘font-variant: small-caps; font-weight: 900’
- （CSS や Java Script の）クラス名: class
  例: class=‘item’
- （CSS や Java Script の) id 名: id
  例: id=‘item’
- テーブルの位置 : table-align, talign
  例:table-align=‘center’

以下は複数の項目を指定する場合の例です。

Name@{align=‘center’, font-color=‘blue’, font-size= 20}

スタイルシートのクラスを指定する場合には以下に記述するスタイルシートファイルの指定を行ってください。

※ width を用いる場合の注意

100px と 200px の幅を指定した文字を中括弧で結合したものは、300px の幅と明示的に指定しないと、外側を 300px に合わせる事が出来ません。
例えば [A, B]! のときに、

[A@{width= 100}, B@{width= 200}]!

では、300px より少し大きくなってしまいます。

[{A@{width=100},B@{width=200}}@{width=300}]!

と記述すると、100,200px より少し小さくなりますが、線の幅を合わせ、全体が 300px にそろいます。

上記は属性などに指定する装飾子です。以下の装飾子は「ページ全体」の括弧や「テーブル全体」の括弧に設定する装飾子です。

ページ全体を囲う括弧への装飾子

- スタイルシートファイルの指定: cssfile
  例: cssfile=‘demo.css’
  例のように ‘ ’ 内で CSS ファイルをパスで指定します。例の場合、出力先ディレクトリに CSS ファイルを置いておく必要があります。
  CSS ファイルを複数指定したい場合は
  cssfile=‘demo1.css, demo2.css’
  というように CSS ファイルをカンマでつなげて指定します。
- JavaScript ファイルの指定: jsfile
  例: jsfile=‘demo.js’
  js ファイルも CSS ファイルのようにカンマで区切って複数指定できます。

- 文字コードの指定; charset
  例: charset= ‘utf-8’
- 著者情報: author
  例: author=‘takehiko’
- 背景画像の指定: background
  例: background= ‘neko.gif’
- ページの背景色:page-bgcolor, pbgcolor
  例:page-bgcolor=‘red’

テーブル全体を囲う括弧への装飾子

- 表の位置: table-align
  例: table-align=‘center’
- 表の罫線: tableborder
  例: tableborder=1

以下はその例です。

⟨**Q7.ssql**⟩

```
GENERATE HTML
  {
    {
      [e.name@{class='name'},e.salary@{class='salary'},
      e.byear@{class='birth'}]!
    }@{title='MEMBER',table-align='center'}
  }@{cssfile='demo.css',charset='euc-jp'}
FROM employee e
```

Q7.ssql では css ファイルを cssfile=‘demo.css’ として宣言しています。また出力先フォルダに demo.css を置く必要があります。demo.css は http://ssql.db.ics.keio.ac.jp/demo.css に置いてあります。

5.1 装飾子の値をデータベースの属性名にした場合

属性に格納されている値が HTML のスタイルに直接反映されます。
例えば、以下のクエリのように dept テーブルに格納されている color 属性を装飾子 bgcolor の値に指定した場合、背景色を color 属性に格納されている値 (Yellow など) に変更することができます。

⟨**Q8.ssql**⟩

```
GENERATE HTML
  [(asc)d.floor, d.name@{bgcolor = dc.fcolor}]!
FROM dept d , deptcolor dc
WHERE dc.floor = d.floor
```

※クエリ 2 行目の asc の使用方法については 8 章を参照

6　画像の表示 (image 関数)

image 関数を用いると画像を表示することができます。image 関数を利用する場合は、特定のディレクトリに対象となる画像ファイルをまとめて置き、データベースに値としてファイル名を入力しておき、以下のように記述します。

image(属性名, '画像ファイルの存在するディレクトリのパス')

path は絶対パス、相対パスのどちらでも指定可能ですが、相対パスは SuperSQL を実行したときに生成される HTML ファイルの出力先ディレクトリ (~/public_html/ssql) を基点とします。
image 関数にも width などの装飾指定をすることができます。
画像のパーミッションは 755 としてください。

例: 従業員名と従業員の写真を横に連結し、写真の幅を 200 に指定する (従業員の写真のファイル名を pict という属性に格納し、そのファイルを ~/public_html/ssql/picts/に置いた場合)

⟨**Q9.ssql**⟩

```
GENERATE HTML
  [e.name, image(e.pict,'./picts')@{width=200}]!
FROM employee e
```

と記述します。

7　リンク生成

7.1　分離型リンク生成 (link 関数と FOREACH 句)

link 関数と FOREACH 句を用いると、結合子 (%) と同様にリンクを生成することができます。リンクを生成するために結合子 (%) を用いる場合は全てを一つの質問文で記述しますが、link 関数と FOREACH 句を用いる場合はリンク元のページとリンク先のページを別々の質問文で記述します。リンク元を生成する質問文で link 関数を、リンク先を生成する質問文で FOREACH 句を記述します。記述方法を例を使って説明します。
従業員名とその従業員の給料、入社年をリンク方向に結合する場合、結合子 (%) を用いると Q10.ssql のようになります。

⟨**Q10.ssql**⟩

```
GENERATE HTML
  [e.name % {e.salary, e.syear}]!
FROM employee e
```

これを link 関数と FOREACH 句を用いると、Q11.ssql と Q12.ssql の二つの質問文になります。

⟨**Q11.ssql**⟩

```
GENERATE HTML
  [link(e.name, 'Q12.ssql', e.id)]!
FROM employee e
```

⟨**Q12.ssql**⟩

```
FOREACH e.id
GENERATE HTML
  [e.salary, e.syear]!
FROM employee e
```

※ Q11.ssql と Q12.ssql、両方のクエリを実行してください。

上記のように、link 関数ではペアとなる FOREACH 句を含む質問文のファイル名を第二引数で指定します。ファイル名の指定は相対指定・絶対指定両方が利用可能です。さらに、 link 関数の第三引数と FOREACH 句で同じ属性を指定します。なお、この属性の値を利用して URL を生成するので、指定する属性はリンク先のページを一意に識別できるもの (主キーなど) を選びます。2 つ以上の属性を指定する場合 (主キーが 2 つ以上あるようなとき) は、link 関数では第三引数以降に属性を一つずつ指定し、FOREACH 句では属性を (,) で 区切って指定します。FOREACH 句で指定する属性は link 関数の第三引数以降で指定した順に指定します。値に null がある属性を指定すると、リンク先には NO DATA FOUND と表示されます。

link 関数と FOREACH 句を使った場合は、link を含むクエリと FOREACH を含むクエリの両方を実行して下さい。
また link 関数を使った場合、リンクが生成されますがそのターゲットを指定したい場合（新たにウィンドウを開く、など）は link 関数の「属性部分」の装飾子として指定する必要があります。以下にその例を記述します。

⟨**Q13.ssql**⟩

```
GENERATE HTML
[link(e.name@{target='_new'}, 'Q12.ssql', e.id)]!
FROM employee e
```

link 関数と FOREACH 句を利用した複雑な質問文の例は、

http://ssql.db.ics.keio.ac.jp/DEMO/link/index.html

の下の titles.ssql、film.ssql、actor.ssql、theater.ssql、town.ssql
にアクセスすると見ることができます。生成結果は、

http://ssql.db.ics.keio.ac.jp/DEMO/link/titles.html

です。この例のように、link 関数と FOREACH 句を用いると結合子 (%) では不可能な相互リンクを生成することができます。

※ FOREACH 句を用いたページにおいて、表示に時間が掛かる場合やファイルサイズが 5MB を越える場合は **link** 関数と **FOREACH** 句を **link1** 関数、**FOREACH1** 句に記述し直して実行することをお勧めします。link1 関数、FOREACH1 句を用いることで FOREACH1 句を用いたページが複数 (Q13.ssql の例では e.id のタプル数) に分割されます。見た目上の変化はありませんが、リンク先ページのデータ量が削減される為、挙動が早くなります。

7.2 URL によるリンク生成 (anchor 関数)

anchor 関数を使って、ハイパーリンクを作成することが出来ます。(anchor は a でも可能。また、第 1 引数、第 2 引数には属性名の指定も可能。)

```
anchor(リンク元のタイトル、リンク先 URL)
例  a('SuperSQL','http://ssql.db.ics.keio.ac.jp/')
```

リンク元を画像にしたい場合には、image_anchor 関数を使います。

```
image_anchor( リンク元の画像 PATH, リンク先 URL )
例  img_a('icon/ssql.png', 'http://ssql.db.ics.keio.ac.jp/')
```

8 ソーティング (並べかえ)

8.1 asc と desc

SuperSQL では属性に基づいて、昇順もしくは降順にソーティングすることができます。対象になる属性の前に括弧を書き、その中にソーティングの種類や順序を入れます。ソーティングの種類には asc と desc の二つがあります。ソーティングに順序付けをする場合は種類の右側に数字を書きます。1 からソーティングの優先順位を書き始め、その後は 2、3 のようになります。数字指定がない場合は記述された順番に上から優先順位がついていきます。数字指定のあるものとないものが混在している場合は、数字指定がされているものの順序が優先されます。

- 昇順 (ascending order): (asc) 属性名
  従業員の id の昇順によってソーティングを行います。
  例: (asc)e.id
- 降順 (descending order): (desc) 属性名
  給料の降順によってソーティングを行います。
  例: (desc)e.salary

まず従業員の給料によって昇順に並べ、もし同じ給料の従業員がいれば id の降順によって並べようとする時は以下のようにクエリを書きます。

⟨**Q14-1.ssql**⟩

```
GENERATE HTML
  [(asc)e.salary, (desc)e.id, e.name]!
FROM employee e
```

又は、

⟨**Q14-2.ssql**⟩

```
GENERATE HTML
  [(asc1)e.salary, (desc2)e.id, e.name]!
FROM employee e
```

部署の id によってソーティングを行い、それから各部署に入っている従業員の id によってソーティングをする時は以下の Q15.ssql のようなクエリになります。

⟨**Q15.ssql**⟩

```
GENERATE HTML
  [ (asc)d.id, [ (desc)e.id ,e.name]!  ]!
FROM dept d, employee e
WHERE d.manager=e.manager
```

8.2 null 関数

ある属性を使ってソートを行いたいが、その属性は表示させたくない場合 null 関数を使うことで表現できます。従業員給料の降順でソートしたいが、従業員給料は表示させたくない場合のクエリを以下に示します。

⟨**Q16.ssql**⟩

```
GENERATE HTML
  [null((desc1)e.salary), e.name]!
FROM employee e
```

9 その他の機能

9.1 集約関数

集約関数というのは最大値 (max)、最小値 (min)、平均 (avg)、合計 (sum)、頻度数 (count) を求めるために使います。対象になる属性を括弧で含み、その括弧の前に適用する集約関数を書きます。

- max[属性名], min[属性名], avg[属性名], sum[属性名], count[属性名]
  前から最大値、最小値、平均、合計、頻度数がそれぞれ求められます。

Q17.ssql のようにクエリを書くと、従業員名や彼らの平均給料を得ることができます。avg の代わりに max を書くと給料の中で一番高い値が、min を書くと一番低い値が求められます。

⟨**Q17.ssql**⟩

```
GENERATE HTML
  [ e.name ]!, avg[e.salary]
FROM employee e
```

9.2 四則演算

SuperSQL クエリ内では四則演算を行うことができます。TFE 内に+, -, /, *, %を用いた計算式を記述してください。Q18 では、'e.salary / 113' によって従業員の給料を 113 で割った数字を表示できます。

⟨**Q18.ssql**⟩

従業員の日本円で格納されている給料をドル (1 ドル=113 円で計算) にして表示。

```
GENERATE HTML
[e.id, e.name, (e.salary / 113) || ’ドル’]!
FROM employee e
```

9.3 SQL の関数

SuperSQL クエリ内では DBMS で用意されている関数を使用することが可能です。関数の前に ‘ & ’ を付けて SQL の関数と SuperSQL の関数を区別します。

以下のようにクエリを書くと、REPLACE 関数を使って、従業員の名前の ‘田’ の部分を ‘山’ に置き換えることができます。

⟨**Q19.ssql**⟩

```
GENERATE HTML
[e.name, &replace(e.name,‘田’,‘山’)]!
FROM employee e
```

9.4 コメント

SuperSQL のクエリの一部をコメントアウトしたい場合にはそのほかの多くのプログラミング言語と同様に/* */でコメントアウトしたい箇所を囲うことでコメントアウトできます。

また、1 行だけコメントアウトしたい場合は、--(ハイフン 2 つ) を使用することでその行の--(ハイフン 2 つ) 以降をコメントアウトすることができます。

⟨**Q20.ssql**⟩

```
GENERATE HTML
{
  [e.name /*@{class=‘name’}*/ , e.salary]!
}/*@{cssfile=‘demo.css’}*/
FROM employee e --,shop s
```

以上のように書くことで、/* */で ‘@{cssfile=‘demo.css’}’ の部分を、--で ‘,shop s’ の部分をコメントアウトすることができます。

10 最後に

分からないことがあれば

- *db-ta*18@*db.ics.keio.ac.jp*

まで気軽にメールをください。もしくは研究室 (24-205,209) に直接来てくださっても結構です。

また、SuperSQL の仕様に基づいたチュートリアルが、
http://ssql.db.ics.keio.ac.jp/tutorial/
にあります。この Tutorial の SuperSQL 質問文には今では使えない古い構文や関数が多数含まれているので、あくまで参考に留めて下さい。