

ライブツアーのセットリストを公演毎に横に並べる

2015/12/07

とりあえず公開

2015/12/08

某『手を抜いたでしょ?』

私『...はい』

という事でテーブル構成等々を手直し。

こちらのサイト <http://www.seven-ripe.sakura.ne.jp/nml/> の作成者と思われる方のツイッターアカウントをたまたま見つけつづやきを見てしまいました。

うちのドキュメント [■■■■でデータベーステーブルの縦・横変換](#) を見て、おそらく各ツアーの公演毎のセットリストを横に並べて表示させたかったんじゃないかな、と想像しました。

公演日: 2003-07-19	公演日: 2003-07-21	公演日: 2003-07-26	公演日: 2003-07-27	公演日: 2003-08-03	公演日: 2003-08-22	公演日: 2003-08-24	公演日: 2003-08-25	公演日: 2003-08-27	公演日: 2003-08-31
開演地: 大阪 会場: Zepp Sapporo	開演地: 大阪 会場: Zepp Osaka	開演地: 大阪 会場: Zepp Osaka	開演地: 福岡 会場: Zepp Fukuoka	開演地: 東京 会場: Zepp Tokyo	開演地: 埼玉 会場: Zepp Fuchu	開演地: 東京 会場: 豊洲PARK	開演地: 京都 会場: 京セラドーム大阪	開演地: 東京 会場: 国立競技場	開演地: 東京 会場: 国立競技場
セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト	セトリリスト
1. TRANSDISGRATION 2. アップルパイ 3. 舞臺 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. アップルパイ 3. 舞臺 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. アップルパイ 3. 舞臺 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. アップルパイ 3. 舞臺 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. アップルパイ 3. 舞臺 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. The space of happiness 3. PROTECTION 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. The space of happiness 3. PROTECTION 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. The space of happiness 3. PROTECTION 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. The space of happiness 3. PROTECTION 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する	1. TRANSDISGRATION 2. The space of happiness 3. PROTECTION 4. 777 5. 舞臺 6. 恋する 7. 恋する 8. 恋する 9. 恋する 10. 恋する 11. 恋する 12. 恋する 13. 恋する 14. 恋する 15. 恋する 16. 恋する 17. 恋する 18. 恋する 19. 恋する 20. 恋する
アンコール	アンコール	アンコール	アンコール	アンコール	アンコール	アンコール	アンコール	アンコール	アンコール
1. The space of happiness 2. PROTECTION	1. The space of happiness 2. PROTECTION	1. The space of happiness 2. PROTECTION	1. The space of happiness 2. PROTECTION	1. The space of happiness 2. PROTECTION					

せっかく見てもらったのに解決しなかったのは残念だな、と思いましたので、私なりの実装方法を書いていってみます。なお、私はPHP使いではないのでPerlのCGIで実装します。

データの取得

実装するにしてもデータがないとどうにもなりません。先のサイトを少しの間眺めて、以下のPerlコードを書きました。サイトの左側フレームにはツアータイトルと公演回のリンクがリストになっていて、クリックすると右側フレームにその公演の情報が表示されます。リンクを見た様子ではURLのパラメタcategoryが1~148のコンテンツを表示できるようです。

[getnanadb.pl](#)

```

use Encode;
use WWW::Mechanize;
use Web::Scraper;
use JSON;

my $baseUrl =
'http://www.seven-ripe.sakura.ne.jp/nml/index.php?category=%s';

my $mech = WWW::Mechanize->new(autocheck=>1);
my $json = JSON->new();
my $pageResult;
my @tours = ();

```

```
my $contentParse = scraper {
    process '//div[@id="contents"]/div[@id="main"]/h2[1]',
    'tour_name'=>"TEXT";
    process
    '//div[@id="contents"]/div[@id="main"]/dl[@class="information"]/dd[1]/div', 'date' =>"TEXT";
    process
    '//div[@id="contents"]/div[@id="main"]/dl[@class="information"]/dd[2]/div', 'place' =>"TEXT";
    process
    '//div[@id="contents"]/div[@id="main"]/dl[@class="information"]/dd[3]/div', 'venue' =>"TEXT";
    process '//div[@id="contents"]/div[@id="main"]/ol[1]//li',
    'setlist[]' => "TEXT";
    process '//div[@id="contents"]/div[@id="main"]/ol[2]//li',
    'encorelist[]' => "TEXT";
    process '//div[@id="contents"]/div[@id="main"]/ol[3]//li',
    'wencorelist[]' => "TEXT";
};

$mech->agent_alias("Windows Mozilla");
foreach my $category ( 1 .. 148 )
{
    $mech->get( sprintf($baseUrl,$category) );
    $pageResult = $contentParse->scrape( $mech->content );
    push(@tours, $pageResult);
}

$json->pretty();
print encode('utf-8',$json->encode(\@tours));
```

このコードを

```
$ perl getnanadb.pl > nana.json
```

な感じで実行するとJSONドキュメント nana.json が取得されます。以下はgetnanadb.plの

```
foreach my $category ( 1 .. 148 )
```

を

```
foreach my $category ( 1 .. 3 )
```

にして最初の3つの公演情報を取り出したものです。

nana.json

```
[
  {
    "setlist" : [
```

```

    "Dear. Friend",
    "Send You My Love",
    "小さな秋の恋～コスモス～",
    "Yo□A□Ke",
    "夕立",
    "空と心と...",
    "冬の淡雪(CMソング)",
    "Bird's Eye View",
    "Our Song",
    "かもめ哀歌",
    "Autunm Moon",
    "Twiligt Highway Star",
    "natural",
    "テルミドール"
  ],
  "date" : " 2000-01-23",
  "encorelist" : [
    "つばさを広げて",
    "NANA色のように"
  ],
  "place" : " 東京",
  "venue" : " 銀座ヤマハホール",
  "tour_name" : "20th Birthday Anniversary Live"
},
{
  "venue" : " ヤクルトホール",
  "tour_name" : "水樹奈々 21ANNIVERSARY CONCERT \"HAPPY\"",
  "wencorelist" : [
    "WINDOW OF HEART"
  ],
  "place" : " 東京",
  "date" : " 2001-01-21",
  "setlist" : [
    "Love's Wonderland",
    "Heaven Knows",
    "想い",
    "Summer Sweet",
    "ピアス",
    "水中の青空",
    "夕やけ雲のむこうに",
    "Shining★star",
    "アノネ～まみむめ もがちょ～",
    "Look Away(Acoustic)",
    "七夕の夜(Acoustic)",
    "NANA色のように",
    "空と心と...",
    "VOICE"
  ],
  "encorelist" : [
    "想い",
    "アノネ～まみむめ もがちょ～"
  ]
},

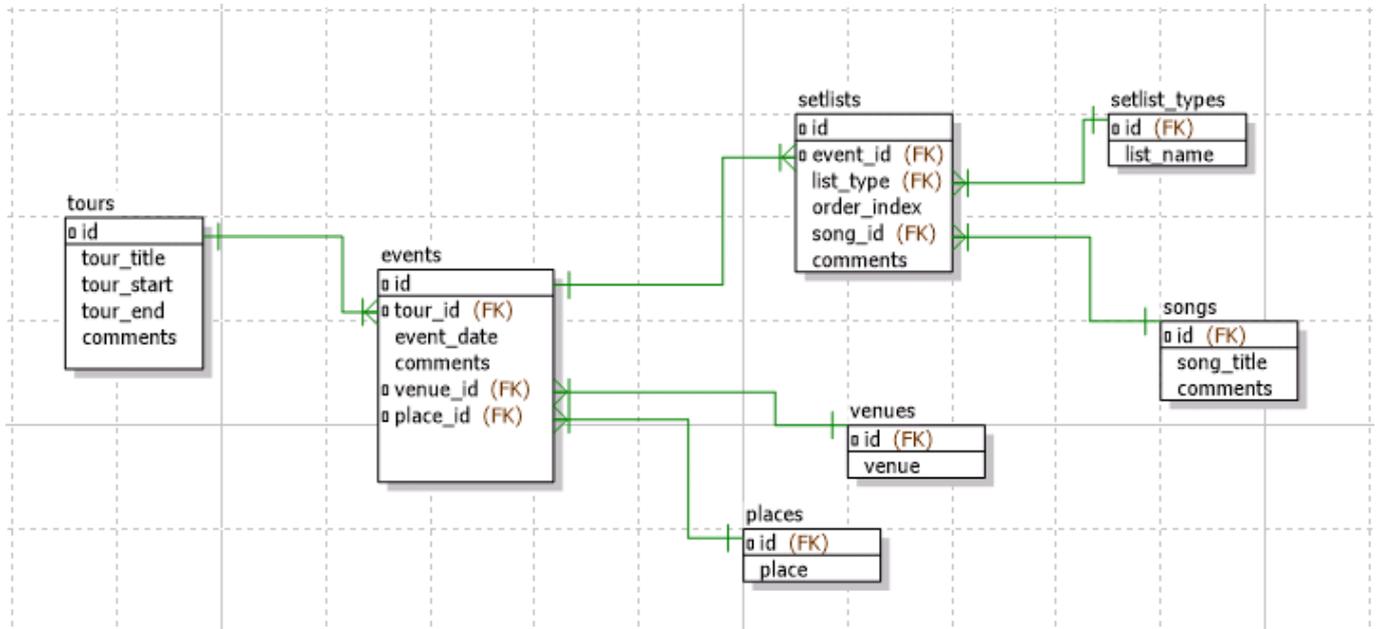
```

```
{
  "encorelist" : [
    "TRANSMIGRATION",
    "アノネ～まみむめ もがちょ～",
  ],
  "date" : " 2001-12-23",
  "setlist" : [
    "Love's Wonderland",
    "Heaven Knows",
    "LOOKING ON THE MOON",
    "オルゴールとピアノと",
    "想い",
    "水中の青空",
    "Lost in Your Eyes",
    "Look Away",
    "WINDOW OF HEART",
    "NANA色のように",
    "シアワセ大将",
    "supersonic girl",
    "The place of happiness"
  ],
  "place" : " 東京",
  "wencorelist" : [
    "真冬の観覧車"
  ],
  "venue" : " 原宿アストロホール",
  "tour_name" : "水樹奈々 X'mas LIVE \"supersonic girl\""
}
]
```

148の公演情報を取り出しできたら、このnana.jsonの内容をデータベースへ格納していきます。

テーブル定義

データベースは PostgreSQL を使っています。PostgreSQL上にデータベース NANAMIZUKI を作成し7つのテーブルを定義しました。



各テーブル定義では項目idに Serial を使っているので、シーケンス(SEQUENCE)オブジェクトも同時に定義されます。

私が使っているpgAdminIIのGUIグリッドでテーブルの編集を行いたいので項目idは主キーに使っていません。

ツアーテーブル

[tours_ddl.sql](#)

```

CREATE TABLE tours(
  id          serial
, tour_name  VARCHAR(200)
, tour_start DATE
, tour_end   DATE
, comments   text
, PRIMARY KEY (id)
);
  
```

公演テーブル

[events_ddl.sql](#)

```

CREATE TABLE events(
  id          serial
, tour_id    INTEGER
, event_date  DATE
, place      VARCHAR(100)
, venue      VARCHAR(100)
, comments   text
, PRIMARY KEY(id)
);
  
```

```
);
```

演目テーブル

[setlists_ddl.sql](#)

```
CREATE TABLE setlists(  
  id          serial  
  ,event_id   INTEGER  
  ,list_type  INTEGER  
  ,order_index INTEGER  
  ,song_id    INTEGER  
  ,comments   text  
  ,PRIMARY KEY(id)  
);
```

演目ラベルテーブル

[setlisttypes_ddl.sql](#)

```
CREATE TABLE setlist_types(  
  id          INTEGER NOT NULL  
  ,list_name  VARCHAR(100)  
  ,PRIMARY KEY(id)  
);
```

曲テーブル

[songs_ddl.sql](#)

```
CREATE TABLE songs(  
  id          serial  
  ,song_title VARCHAR(500)  
  ,comments   text  
  ,PRIMARY KEY(id)  
  ,UNIQUE (song_title)  
);
```

場所テーブル

places_ddl.sql

```
CREATE TABLE places(  
  id      serial  
  ,place  VARCHAR(100)  
  ,PRIMARY KEY(id)  
);
```

会場テーブル

venues_ddl.sql

```
CREATE TABLE venues(  
  id      serial  
  ,venue  VARCHAR(200)  
  ,PRIMARY KEY(id)  
);
```

データ取り込み

nana.jsonはJSON形式データとなっているのでPHPやPerlはじめいろいろな言語やツールで参照できるデータ形式になります。

JSONデータからSQL生成

perlでnana.jsonを読みSQLを生成させました。使ったPerlのコードは泥臭くこんな感じ。

gen.pl

```
use Encode;  
use JSON;  
  
my $d;  
  
while(<STDIN>)  
{  
  $d .= decode('utf-8', $_);  
}  
  
my $json = JSON->new()->decode($d);  
  
foreach my $event ( @{$json} )  
{  
  my $date = $event->{date};
```

```
my $place = $event->{place};
my $venue = $event->{venue};
my $tour_name = $event->{tour_name};

$tour_name =~ s/'//'/g;

my $orderindex = 1;
foreach my $song_title ( @{$event->{setlist}} )
{
    $song_title =~ s/'//'/g;
    print encode('utf-8',sprintf("insert into setlists(list_type,
order_index, comments)
values(1,%d,'1:[%s]2:[%s]3:[%s]4:[%s]5:[%s]');\n", $orderindex,
$song_title, $date, $place, $venue, $tour_name));
    $orderindex++;
}
$orderindex = 1;
foreach my $song_title ( @{$event->{encorelist}} )
{
    $song_title =~ s/'//'/g;
    print encode('utf-8',sprintf("insert into setlists(list_type,
order_index, comments)
values(2,%d,'1:[%s]2:[%s]3:[%s]4:[%s]5:[%s]');\n", $orderindex,
$song_title, $date, $place, $venue, $tour_name));
    $orderindex++;
}
$orderindex = 1;
foreach my $song_title ( @{$event->{wencorelist}} )
{
    $song_title =~ s/'//'/g;
    print encode('utf-8',sprintf("insert into setlists(list_type,
order_index, comments)
values(3,%d,'1:[%s]2:[%s]3:[%s]4:[%s]5:[%s]');\n", $orderindex,
$song_title, $date, $place, $venue, $tour_name));
    $orderindex++;
}
}
```

演目テーブルにデータ取り込み

Perlのコード gen.pl から生成されたSQLはこんな感じ。

```
$ cat nana.json | perl gen.pl > setlists.sql
$ cat setlists.sql | head -30
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,1,'1:[Dear.
Friend]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary
Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,2,'1:[Send
```

```
You My Love]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,3,'1:[小さな秋の恋～コスモス～]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,4,'1:[Yo-A-Ke]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,5,'1:[夕立]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,6,'1:[空と心と...]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,7,'1:[冬の淡雪(CMソング)]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,8,'1:[Bird]'s Eye View]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,9,'1:[Our Song]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,10,'1:[かもめ哀歌]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,11,'1:[Autumn Moon]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,12,'1:[Twiligt Highway Star]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,13,'1:[natural]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,14,'1:[テルミドル]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(2,1,'1:[つばさを広げて]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(2,2,'1:[NANA色のように]2:[ 2000-01-23]3:[ 東京]4:[ 銀座ヤマハホール]5:[20th Birthday Anniversary Live]');
INSERT INTO setlists(list_type, order_index, comments)
VALUES(1,1,'1:[Love]'s Wonderland]2:[ 2001-01-21]3:[ 東京]4:[ ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,2,'1:[Heaven Knows]2:[ 2001-01-21]3:[ 東京]4:[ ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,3,'1:[想い]2:[ 2001-01-21]3:[ 東京]4:[ ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,4,'1:[Summer Sweet]2:[ 2001-01-21]3:[ 東京]4:[ ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
```

```
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,5,'1:[ピアス]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,6,'1:[水中の青空]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,7,'1:[夕やけ雲のむこうに]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,8,'1:[Shining★star]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,9,'1:[アノネ～まみむめ もがちょ～]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,10,'1:[Look Away(Acoustic)]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,11,'1:[七夕の夜(Acoustic)]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,12,'1:[NANA 色のように]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,13,'1:[空と心と...]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
INSERT INTO setlists(list_type, order_index, comments) VALUES(1,14,'1:[VOICE]2:[2001-01-21]3:[東京]4:[ヤクルトホール]5:[水樹奈々 21ANNIVERSARY CONCERT "HAPPY"]');
$
```

このSQLを実行してテーブルsetlists に最初のデータを取り込みします。

演目テーブルから各テーブルへのデータ格納

以下はテーブルsetlistsに格納されたデータを基に残りのテーブルへ格納していくSQLです。

convert.sql

```
-- 演目テーブルの項目commentsから曲タイトルを抜き出して曲テーブルへ格納します。
INSERT INTO songs(song_title)
  SELECT regexp_replace(comments, '^1:\[(.*?)\].+$', '\1') AS
  "song_title"
  FROM setlists
  GROUP BY "song_title"
  ORDER BY "song_title"
;

-- 演目テーブルの項目commentsから場所を抜き出して場所テーブルへ格納します。
INSERT INTO places(place)
  SELECT
```

```
TRIM(regexp_replace(setlists.comments, '^.+3:\[(.*?)\].+$', '\1')) AS
"place"
  FROM setlists
  GROUP BY "place"
  ORDER BY "place"
;

-- 演目テーブルの項目commentsから会場を抜き出して会場テーブルへ格納します。
INSERT INTO venues(venue)
  SELECT
TRIM(regexp_replace(setlists.comments, '^.+4:\[(.*?)\].+$', '\1')) AS
"venue"
  FROM setlists
  GROUP BY "venue"
  ORDER BY "venue"
;

-- 演目テーブルの項目commentsからツアーのタイトルを抜き出してツアーテーブルへ格納しま
す。
INSERT INTO tours(tour_title)
  SELECT TRIM(regexp_replace(setlists.comments, '^.+5:\[(.*?)\]$', '\1'))
AS "tour_title"
  FROM setlists
  GROUP BY "tour_title"
  ORDER BY "tour_title"
;

-- 演目テーブルの項目song_idに曲テーブルの項目idを設定します。
UPDATE setlists
  SET song_id = songs.id
  FROM songs
  WHERE songs.song_title =
regexp_replace(setlists.comments, '^1:\[(.*?)\].+$', '\1')
;

-- 演目テーブルから集約処理で邪魔になる情報(曲タイトル)を項目commentsから取り除きます。
UPDATE setlists
  SET comments =
TRIM(regexp_replace(setlists.comments, '^.+2:\[.*\]$', '\1'))
;

-- 演目テーブルから公演テーブルへ格納します。
-- 演目テーブルの曲情報が無い状態で集約すると、1レコードが1公演の情報、となります。
INSERT INTO events(comments)
  SELECT comments
  FROM setlists
  GROUP BY comments
  ORDER BY comments
;

-- 公演テーブルの項目event_dateに項目commentsの公演日を転記します。
```

```
UPDATE events
  SET event_date =
to_date(regex_replace(comments, '^2:\[(.*?)\].+$', '\1'), 'YYYY-MM-DD')
;

-- 演目テーブルの項目event_idに公演テーブルの項目idを設定します。
UPDATE setlists
  SET event_id = events.id
  FROM events
  WHERE events.comments = setlists.comments
;

-- 公演テーブルの項目commentsを使って公演テーブルのtour_idにツアーテーブルの項目id
を設定します。
UPDATE events
  SET tour_id = tours.id
  FROM tours
  WHERE tour_title =
TRIM(regex_replace(events.comments, '^.+?5:\[(.*?)\]$', '\1'))
;

-- 公演テーブルの項目commentsを使って公演テーブルのplace_idに場所テーブルの項目idを
を設定します。
UPDATE events
  SET place_id = places.id
  FROM places
  WHERE place =
TRIM(regex_replace(events.comments, '^.+?3:\[(.*?)\].+$', '\1'))
;

-- 公演テーブルの項目commentsを使って公演テーブルのvenue_idに会場テーブルの項目idを
を設定します。
UPDATE events
  SET venue_id = venues.id
  FROM venues
  WHERE venue =
TRIM(regex_replace(events.comments, '^.+?4:\[(.*?)\].+$', '\1'))
;

-- このテーブルへは手書きで追加します。このテーブルはなくてもかまいません。
INSERT INTO setlist_types(id, list_name) VALUES(1, 'セットリスト');
INSERT INTO setlist_types(id, list_name) VALUES(2, 'アンコール');
INSERT INTO setlist_types(id, list_name) VALUES(3, 'Wアンコール');

-- テーブルtours にはツアー開始、終了の項目があります。
-- この項目をテーブルeventsのデータで設定します。
UPDATE tours
  SET tour_start = mindate
    , tour_end   = maxdate
  FROM (SELECT tour_id
        , MIN(event_date) AS mindate
        , MAX(event_date) AS maxdate
```

```
        FROM events
        GROUP BY tour_id ) T
WHERE tours.id = T.tour_id
;

-- 各テーブルの項目 comments には取り込みと関連付け作業で使ったデータが残ったままです
-- なのでこれを削除します。
UPDATE tours SET comments = NULL;
UPDATE songs SET comments = NULL;
UPDATE events SET comments = NULL;
UPDATE setlists SET comments = NULL;

-- 結果をコミット
commit;
```

データ取り込みはこれで終わりです。

クエリーの作成

今回のような、横に並ぶ数が可変となるような(ツアー毎に公演回数が異なる)場合は基本的にクエリを書けないです。最近ですとPivotとかcrosstabとかの構文を備えているRDBMSも出てきていますが、そのクエリを発行した側が動的な戻り値に対応できていなくてはなりません。

もし、すべてのクエリの結果項目が同じ型に落とし込めるのであればUNION演算子で複数クエリを縦に繋ぐ方法も考えられなくはないのですが、面倒ばかりが多くあまりいい結果にならないかもしれません。

ここではクエリで横につなげるのではなく、クエリ結果を横につなげる方法で実装してみます。

ツアー一覧

メニューで使うであろうツアー一覧を作成するクエリ。

```
SELECT id
       ,tour_title
       ,(SELECT COUNT(*) FROM events WHERE events.tour_id=tours.id) AS
"stages"
FROM tours
ORDER BY tour_start,id
```

結果はこうなります。

```
nanamizuki=> SELECT id
nanamizuki->       ,tour_title
nanamizuki->       ,(SELECT COUNT(*) FROM events WHERE
events.tour_id=tours.id) AS "stages"
nanamizuki-> FROM tours
nanamizuki-> ORDER BY tour_start,id;
```

id	tour_title	stages
207	20th Birthday Anniversary Live	1
234	水樹奈々 21ANNIVERSARY CONCERT "HAPPY"	1
235	水樹奈々 X'mas LIVE "supersonic girl"	1
208	NANA MIZUKI "LIVE ATTRACTION 2002"	4
227	NANA MIZUKI LIVE SENSATION 2003	10
228	NANA MIZUKI LIVE SKIPPER COUNTDOWN 2003-2004	1
229	NANA MIZUKI LIVE SPARK 2004 -summer-	7
225	NANA MIZUKI LIVE RAINBOW 2004-2005	3
226	NANA MIZUKI LIVE ROCKET 2005□summer□	6
233	NANA MIZUKI LIVEDOM 2006-BIRTH-	1
232	NANA MIZUKI LIVE UNIVERSE 2006 □summer□	8
224	NANA MIZUKI LIVE MUSEUM 2007	1
219	NANA MIZUKI LIVE FORMULA 2007-2008	7
215	NANA MIZUKI LIVE FIGHTER 2008 □BLUE SIDE□	1
216	NANA MIZUKI LIVE FIGHTER 2008 □RED SIDE□	1
214	NANA MIZUKI LIVE FEVER 2009	7
213	NANA MIZUKI LIVE DIAMOND 2009	1
209	NANA MIZUKI LIVE ACADEMY 2010	7
220	NANA MIZUKI LIVE GAMES 2010	2
221	NANA MIZUKI LIVE GRACE 2011 -ORCHESTRA-	2
223	NANA MIZUKI LIVE JOURNEY 2011	14
211	NANA MIZUKI LIVE CASTLE 2011	2
231	NANA MIZUKI LIVE UNION 2012	14
222	NANA MIZUKI LIVE GRACE 2013 -OPUSII-	2
212	NANA MIZUKI LIVE CIRCUS 2013	11
217	NANA MIZUKI LIVE FLIGHT 2014	11
218	NANA MIZUKI LIVE FLIGHT 2014+	3
230	NANA MIZUKI LIVE THEATER 2015 -ACOUSTIC-	2
210	NANA MIZUKI LIVE ADVENTURE 2015	14

(29行)

nanamizuki=>

公演リスト

指定のツアーの公演情報を一覧します。

```

SELECT t1.id
      ,t1.event_date
      ,t2.place
      ,t3.venue
FROM events t1 LEFT OUTER JOIN places t2 ON(t1.place_id = t2.id)
              LEFT OUTER JOIN venues t3 ON(t1.venue_id = t3.id)
WHERE tour_id = ? -- tour_id
ORDER BY event_date, id
;

```

tour_id に 227□NANA MIZUKI LIVE SENSATION 2003 のid□を指定すると、このような出力になります。

```

nanamizuki=> SELECT t1.id
nanamizuki->      ,t1.event_date
nanamizuki->      ,t2.place
nanamizuki->      ,t3.venue
nanamizuki-> FROM events t1 LEFT OUTER JOIN places t2 ON(t1.place_id =
nanamizuki->      t2.id)
nanamizuki->      LEFT OUTER JOIN venues t3 ON(t1.venue_id =
nanamizuki->      t3.id)
nanamizuki-> WHERE tour_id = 227 -- NANA MIZUKI LIVE SENSATION 2003 tour_id
nanamizuki-> ORDER BY event_date, id
nanamizuki-> ;
  id | event_date | place |          venue
-----+-----+-----+-----
 299 | 2003-07-19 | 宮城 | Zepp Sendai
 300 | 2003-07-21 | 北海道 | Zepp Sapporo
 301 | 2003-07-26 | 大阪 | Zepp Osaka
 302 | 2003-07-27 | 福岡 | Zepp Fukuoka
 303 | 2003-08-03 | 東京 | Zepp Tokyo □
 304 | 2003-08-22 | 埼玉 | 川口リリアメインホール
 305 | 2003-08-24 | 愛知 | 愛知県勤労会館
 306 | 2003-08-25 | 京都 | 京都会館第二ホール
 307 | 2003-08-27 | 愛媛 | 松山市民会館中ホール
 308 | 2003-08-31 | 東京 | 渋谷公会堂
(10行)

nanamizuki=>

```

セットリスト

指定の公演のセットリストを一覧します。

```

SELECT t1.id
      ,t2.list_name
      ,t1.order_index
      ,t3.song_title
FROM setlists t1 LEFT OUTER JOIN setlist_types t2 ON( t1.list_type =
t2.id)
      LEFT OUTER JOIN songs t3 ON( t1.song_id = t3.id)
WHERE event_id = ? -- event_id
ORDER BY t1.list_type, t1.order_index
;

```

テーブルsetlist_typesを作成していないならこのクエリで。

```

SELECT t1.id
      ,t1.list_type
      ,t1.order_index
      ,t3.song_title
FROM setlists t1 LEFT OUTER JOIN songs t3 ON( t1.song_id = t3.id)
WHERE event_id = ? -- event_id

```

```
ORDER BY t1.list_type, t1.order_index
```

```
;
```

event_id に 299,303,304 を指定してみます。

```

nanamizuki=> SELECT t1.id
nanamizuki->         ,t2.list_name
nanamizuki->         ,t1.order_index
nanamizuki->         ,t3.song_title
nanamizuki-> FROM setlists t1 LEFT OUTER JOIN setlist_types t2 ON(
nanamizuki-> t1.list_type = t2.id)
nanamizuki->         LEFT OUTER JOIN songs t3 ON(
nanamizuki-> t1.song_id = t3.id)
nanamizuki-> WHERE event_id = 299 -- 2003-07-19 宮城 Zepp Sendai
nanamizuki-> ORDER BY t1.list_type, t1.order_index
nanamizuki-> ;

```

id	list_name	order_index	song_title
6765	セットリスト	1	TRANMIGRATION
6766	セットリスト	2	アノネ～まみむめ もがちょ～
6767	セットリスト	3	Heaven Knows
6768	セットリスト	4	リプレイマシン -custom-
6769	セットリスト	5	HONEY FLOWER
6770	セットリスト	6	恋してる...
6771	セットリスト	7	二人のMemory
6772	セットリスト	8	想い
6773	セットリスト	9	POWER GATE
6774	セットリスト	10	What cheer□
6775	セットリスト	11	YEAR! めっちゃホリディ (松浦亜弥)
6776	セットリスト	12	空と心と...
6777	セットリスト	13	フリースタイル
6778	セットリスト	14	ジュリエット
6779	セットリスト	15	still IN the groove
6780	セットリスト	16	supersonic girl
6781	セットリスト	17	Suddenly□巡り合えて～
6782	セットリスト	18	NEW Sensation
6783	セットリスト	19	refrain
6784	アンコール	1	The place OF happiness
6785	アンコール	2	PROTECTION

(21行)

```

nanamizuki=> SELECT t1.id
nanamizuki->         ,t2.list_name
nanamizuki->         ,t1.order_index
nanamizuki->         ,t3.song_title
nanamizuki-> FROM setlists t1 LEFT OUTER JOIN setlist_types t2 ON(
nanamizuki-> t1.list_type = t2.id)
nanamizuki->         LEFT OUTER JOIN songs t3 ON(
nanamizuki-> t1.song_id = t3.id)
nanamizuki-> WHERE event_id = 303 -- 2003-08-03 東京 Zepp Tokyo □
nanamizuki-> ORDER BY t1.list_type, t1.order_index

```

```

nanamizuki-> ;
  id | list_name | order_index | song_title
-----+-----+-----+-----
6850 | セットリスト | 1 | TRANMIGRATION
6851 | セットリスト | 2 | アノネ～まみむめ もがちょ～
6852 | セットリスト | 3 | Heaven Knows
6853 | セットリスト | 4 | リプレイマシンのcustom-
6854 | セットリスト | 5 | HONEY FLOWER
6855 | セットリスト | 6 | 恋してる...
6856 | セットリスト | 7 | 二人のMemory
6857 | セットリスト | 8 | 想い
6858 | セットリスト | 9 | POWER GATE
6859 | セットリスト | 10 | What cheer☐
6860 | セットリスト | 11 | VALENTI(BoA)
6861 | セットリスト | 12 | 空と心と...
6862 | セットリスト | 13 | フリースタイル
6863 | セットリスト | 14 | ジュリエット
6864 | セットリスト | 15 | still IN the groove
6865 | セットリスト | 16 | supersonic girl
6866 | セットリスト | 17 | Suddenly☐巡り合えて～
6867 | セットリスト | 18 | NEW Sensation
6868 | セットリスト | 19 | refrain
6869 | アンコール | 1 | The place OF happiness
6870 | アンコール | 2 | PROTECTION
6871 | Wアンコール | 1 | NEW Sensation
(22行)

```

```

nanamizuki=> SELECT t1.id
nanamizuki->      ,t2.list_name
nanamizuki->      ,t1.order_index
nanamizuki->      ,t3.song_title
nanamizuki-> FROM setlists t1 LEFT OUTER JOIN setlist_types t2 ON(
t1.list_type = t2.id)
nanamizuki->      LEFT OUTER JOIN songs t3 ON(
t1.song_id = t3.id)
nanamizuki-> WHERE event_id = 304 -- 2003-08-22 埼玉川口リリアメインホール
nanamizuki-> ORDER BY t1.list_type, t1.order_index
nanamizuki-> ;

```

```

  id | list_name | order_index | song_title
-----+-----+-----+-----
6872 | セットリスト | 1 | TRANSMIGRATION
6873 | セットリスト | 2 | The place OF happiness
6874 | セットリスト | 3 | おんなになあれ
6875 | セットリスト | 4 | リプレイマシンのcustom-
6876 | セットリスト | 5 | HONEY FLOWER
6877 | セットリスト | 6 | 恋してる...
6878 | セットリスト | 7 | 二人のMemory
6879 | セットリスト | 8 | STAND
6880 | セットリスト | 9 | POWER GATE
6881 | セットリスト | 10 | What cheer☐
6882 | セットリスト | 11 | 明日へのbrilliant road(angela)

```

6883	セットリスト	12	テルミドール
6884	セットリスト	13	Heaven Knows
6885	セットリスト	14	想い
6886	セットリスト	15	still IN the groove
6887	セットリスト	16	supersonic girl
6888	セットリスト	17	Suddenly□巡り合えて～
6889	セットリスト	18	NEW Sensation
6890	セットリスト	19	refrain
6891	アンコール	1	アノネ～まみむめ もがちょ～
6892	アンコール	2	PROTECTION

(21行)

nanamizuki=>

本家様とも一致していることがわかりました。

CGIの作成

作成したクエリを基に、HTMLコンテンツを生成するCGIを作成します。

ツアーリスト

ツアーのリストをHTMLで出力します。 ツアー名にリンクを張り、b.plを呼び出せるようにします。

a.pl

```
use Encode;
use CGI;
use DBI;

my $dbname="nanamizuki";
my $dbhost="your.database.host.addr";
my $dbuser="db user id";
my $dbpass="db password";

my $encstr = "utf-8";

my $cgi = CGI->new();
my $dbh =
DBI->connect("dbi:Pg:dbname=$dbname;host
=$dbhost",$dbuser,$dbpass) or die
"cannot connect database.";
my $sth = $dbh->prepare('SELECT id,
tour_title FROM tours ORDER BY
tour_start' );
my $id;
my $tour_title;
```

Live

1. [20th Birthday Anniversary Live](#)
2. [水樹宗々 21ANNIVERSARY CONCERT "HAPPY"](#)
3. [水樹宗々 X'mas LIVE "supersonic girl"](#)
4. [NANA MIZUKI "LIVE ATTRACTION 2002"](#)
5. [NANA MIZUKI LIVE SENSATION 2003](#)
6. [NANA MIZUKI LIVE SKIPPER COUNTDOWN 2003-2004](#)
7. [NANA MIZUKI LIVE SPARK 2004 _summer-](#)
8. [NANA MIZUKI LIVE RAINBOW 2004-2005](#)
9. [NANA MIZUKI LIVE ROCKET 2005~_summer~](#)
10. [NANA MIZUKI LIVEDOM 2006-BIRTH-](#)
11. [NANA MIZUKI LIVE UNIVERSE 2006 ~_summer~](#)
12. [NANA MIZUKI LIVE MUSEUM 2007](#)
13. [NANA MIZUKI LIVE FORMULA 2007-2008](#)
14. [NANA MIZUKI LIVE FIGHTER 2008 ~BLUE SIDE~](#)
15. [NANA MIZUKI LIVE FIGHTER 2008 ~RED SIDE~](#)
16. [NANA MIZUKI LIVE FEVER 2009](#)
17. [NANA MIZUKI LIVE DIAMOND 2009](#)
18. [NANA MIZUKI LIVE ACADEMY 2010](#)
19. [NANA MIZUKI LIVE GAMES 2010](#)
20. [NANA MIZUKI LIVE GRACE 2011 _ORCHESTRA-](#)
21. [NANA MIZUKI LIVE JOURNEY 2011](#)
22. [NANA MIZUKI LIVE CASTLE 2011](#)
23. [NANA MIZUKI LIVE UNION 2012](#)
24. [NANA MIZUKI LIVE GRACE 2013 _OPUSII-](#)
25. [NANA MIZUKI LIVE CIRCUS 2013](#)
26. [NANA MIZUKI LIVE FLIGHT 2014](#)
27. [NANA MIZUKI LIVE FLIGHT 2014+](#)
28. [NANA MIZUKI LIVE THEATER 2015 _ACOUSTIC-](#)
29. [NANA MIZUKI LIVE ADVENTURE 2015](#)

```
print
$cgi->header({charset=>'utf-8'});

print $cgi->start_html( {title=>'Live
List', charset=>'utf-8'} );

print $cgi->h1( {class=>"h1"},
"Live");

print $cgi->start_ol({});

$dbh->{AutoCommit}=0;
$sth->execute();
$sth->bind_col( 1, \$id );
$sth->bind_col( 2, \$tour_title );

while( $sth->fetchrow_hashref() !=
undef )
{
    $id = encode($encstr, $id
);
    $tour_title = encode($encstr,
$tour_title);
    print $cgi->li(
$cgi->a({class=>'leftlink',
href=>"b.cgi?id=$id"},$tour_title) );
}
$sth->finish();
$dbh->commit();

print $cgi->end_ol();

print $cgi->end_html();

$dbh->disconnect();
```

公演情報とセットリスト

ツアーの公演会場を横に並べたテーブルを作り、その中にセットリストを埋め込みします。

先のa.pl で生成されたリンクから呼び出される想定で作られていてURLパラメタ id=xxx で示すツアーの公演情報を横並びに表示、情報中にセットリストが配置されます。

例は、id=227の“NANA MIZUKI LIVE SENSATION 2003”を選択した場合の結果です。

[b.pl](#)


```
print
$cgi->header({charset=>'utf-8'});
print $cgi->start_html(
{title=>encode($encstr,decode('euc-jp','
公演情報')), charset=>'utf-8'} );

print $cgi->start_table({border=>1});

##### 開催場所、会場情報展開

print $cgi->start_Tr();

$sth1->bind_param( 1, $cgi_id);
$sth1->execute();
$sth1->bind_col( 1, \$event_id );
$sth1->bind_col( 2, \$event_date );
$sth1->bind_col( 3, \$place );
$sth1->bind_col( 4, \$venue );

while( $sth1->fetchrow_hashref() !=
undef )
{
    $event_id = encode($encstr,
$event_id );
    $event_date = encode($encstr,
$event_date);
    $place = encode($encstr, $place
);
    $venue = encode($encstr, $venue
);

    print $cgi->td( {nowrap=>'nowrap'},
$cgi->p( encode($encstr, decode('euc-
jp','公演日:')).$event_date ),
$cgi->p( encode($encstr, decode('euc-
jp','開催地:')).$place
),
$cgi->p( encode($encstr, decode('euc-
jp','会場:')).$venue
) );
}
$dbh->commit();

print $cgi->end_Tr();

##### セットリストの展開

print $cgi->start_Tr();

$sth1->bind_param( 1, $cgi_id);
$sth1->execute();
```

```
$sth1->bind_col( 1, \$event_id );
$sth1->bind_col( 2, \$event_date );
$sth1->bind_col( 3, \$place );
$sth1->bind_col( 4, \$venue );

while( $sth1->fetchrow_hashref() !=
undef )
{
    $event_id = encode($encstr,
$event_id);

    print
$cgi->start_td({valign=>'top'});

    ## セットリスト
    $sth2->bind_param( 1, $event_id);
    $sth2->bind_param( 2, 1 );
    $sth2->execute();
    $sth2->bind_col( 1, \$song_title );

    print $cgi->p({class=>"list"},
encode($encstr, decode('euc-jp', 'セットリス
ト')));
    print $cgi->start_ol();
    while( $sth2->fetchrow_hashref() !=
undef )
    {
        $song_title = encode($encstr,
$song_title);
        print $cgi->li({}, $song_title);
    }
    print $cgi->end_ol();
    $dbh->commit();

    ## アンコール
    $sth2->bind_param( 1, $event_id);
    $sth2->bind_param( 2, 2 );
    $sth2->execute();
    $sth2->bind_col( 1, \$song_title );

    print $cgi->p({class=>"list"},
encode($encstr, decode('euc-jp', 'アンコー
ル')));
    print $cgi->start_ol();
    while( $sth2->fetchrow_hashref() !=
undef )
    {
        $song_title = encode($encstr,
$song_title);
        print $cgi->li({}, $song_title);
    }
}
```

```
print $cgi->end_ol();
$dbh->commit();

## Wアンコール
$sth2->bind_param( 1, $event_id);
$sth2->bind_param( 2, 3 );
$sth2->execute();
$sth2->bind_col( 1, \$song_title );

print $cgi->p({class=>"list"},
encode($encstr, decode('euc-jp', 'Wアンコー
ル')));
print $cgi->start_ol();
while( $sth2->fetchrow_hashref() !=
undef )
{
    $song_title = encode($encstr,
    $song_title);
    print $cgi->li({}, $song_title);
}
print $cgi->end_ol();
$dbh->commit();

}
$sth1->finish();
$sth2->finish();
$dbh->commit();

print $cgi->end_td();
print $cgi->end_Tr();

print $cgi->end_table();

print $cgi->end_html();

$dbh->disconnect();
```

データメンテナンス

なんで曲や場所、会場、ツアーを別々にしたのか。それはデータのメンテナンスをやりやすくするためです。

たとえば、会場の情報を一覧してみます。

```
nanamizuki=> SELECT id, '['||venue||']' FROM venues ORDER BY 2;
```

```
id | ?COLUMN?
```

```
-----+-----
```

```
179 | [Club Diamond Hall]
```

```
180 | [ON AIR OSAKA]
```

- 181 | [QVCマリンフィールド(千葉マリンスタジアム)]
- 182 | [Resorts World Theatre, Resorts World Sentosa]
- 183 | [Zepp Fukuoka]
- 184 | [Zepp Nagoya]
- 185 | [Zepp Osaka]
- 186 | [Zepp Sapporo]
- 187 | [Zepp Sendai]
- 189 | [Zepp Tokyo □]
- 188 | [Zepp Tokyo]
- 190 | [さいたまスーパーアリーナ]
- 191 | [さぬき市野外音楽広場テアトロン]
- 192 | [とりぎん文化会館]
- 193 | [ひめぎんホール]
- 194 | [びわ湖ホール]
- 195 | [やまぎんホール(山形県県民会館)]
- 196 | [アクトシティ浜松]
- 197 | [アスティとくしま]
- 198 | [エコパアリーナ]
- 199 | [グランキューブ大阪 メインホール]
- 200 | [グリーンドーム前橋]
- 201 | [セキスイハイムスーパーアリーナ(グランディ・21□)]
- 202 | [ゼビオアリーナ仙台]
- 203 | [ダイヤモンドホール]
- 204 | [ニトリ文化ホール]
- 205 | [パシフィコ横浜 国立大ホール]
- 206 | [パシフィコ横浜 国立大ホール]
- 207 | [ヤクルトホール]
- 208 | [リンクステーションホール青森(青森市文化会館)]
- 209 | [三重県営サンアリーナ]
- 210 | [京都会館第二ホール]
- 211 | [仙台サンプラザホール]
- 212 | [佐賀市文化会館]
- 213 | [倉敷市民会館]
- 214 | [北上市文化交流センター さくらホール]
- 215 | [北九州メディアドーム]
- 216 | [原宿アストロホール]
- 217 | [名古屋国際会議場 センチュリーホール]
- 218 | [名古屋国際会議場 センチュリーホール]
- 219 | [名古屋市民会館]
- 220 | [周南市文化会館]
- 221 | [国立代々木競技場第一体育館]
- 222 | [大阪NHKホール]
- 223 | [大阪厚生年金会館]
- 224 | [大阪城ホール]
- 225 | [大阪市中央体育館]
- 226 | [天母体育館]
- 227 | [宇都宮市文化会館]
- 228 | [宮崎市民文化ホール]
- 229 | [富士急ハイランド コニファーフォレスト]
- 230 | [島根県民会館]
- 231 | [川口リリアメインホール]

- 232 | [幕張メッセ イベントホール]
 - 233 | [広島アステールプラザ 大ホール]
 - 234 | [愛媛県武道館]
 - 236 | [愛媛県県民文化会館 サブホール]
 - 235 | [愛媛県県民文化会館]
 - 237 | [愛知厚生年金会館]
 - 238 | [愛知県勤労会館]
 - 239 | [新居浜市市民文化センター大ホール]
 - 240 | [新潟県民会館]
 - 241 | [日本ガイシホール]
 - 242 | [日本武道館]
 - 243 | [日比谷公園大音楽堂(日比谷野外大音楽堂)]
 - 244 | [本多の森ホール]
 - 245 | [東京エレクトロンホール宮城]
 - 246 | [東京ドーム]
 - 247 | [東京厚生年金会館]
 - 248 | [東京国際フォーラム ホールC]
 - 249 | [松山市民会館 大ホール]
 - 250 | [松山市民会館中ホール]
 - 251 | [森のホール2 1 大ホール(松戸市文化会館)]
 - 252 | [横浜アリーナ]
 - 253 | [横浜スタジアム]
 - 254 | [渋谷公会堂]
 - 255 | [福井フェニックスプラザ]
 - 256 | [福岡サンパレス ホール]
 - 257 | [福岡国際会議場メインホール]
 - 258 | [秋田県民会館]
 - 259 | [西武ドーム]
 - 260 | [西武プリンスドーム]
 - 261 | [銀座ヤマハホール]
 - 262 | [長崎ブリックホール]
 - 263 | [長野ビッグハット]
 - 264 | [須賀川市文化センター]
 - 265 | [高岡市民会館]
 - 266 | [鹿児島市民文化ホール 第1ホール]
 - 267 | [□□□大阪ホール]
- (89 行)

nanamizuki=>

以下の組ははたぶん同じ場所ですよ。

189 | [Zepp Tokyo □]

188 | [Zepp Tokyo]

205 | [パシフィコ横浜 国立大ホール]

206 | [パシフィコ横浜 国立大ホール]

217 | [名古屋国際会議場 センチュリーホール]

218 | [名古屋国際会議場 センチュリーホール]

NHK大阪ホールの表記が正しそうだけど実際がわからないので保留。

222 | [大阪NHKホール]
267 | [大阪ホール]

こういうデータの揺れはよくあります。このidがどのくらい使われているか公演テーブルを見てみましょう。

```
nanamizuki=> SELECT events.id,venue_id,venue FROM events LEFT OUTER JOIN
venues ON(venue_id=venues.id) WHERE venue_id IN(189,188,205,206,217,218);
```

id	venue_id	venue
336	217	名古屋国際会議場 センチュリーホール
347	218	名古屋国際会議場センチュリーホール
348	218	名古屋国際会議場センチュリーホール
314	188	Zepp Tokyo
303	189	Zepp Tokyo □
298	188	Zepp Tokyo
313	188	Zepp Tokyo
333	205	パシフィコ横浜 国立大ホール
334	205	パシフィコ横浜 国立大ホール
325	206	パシフィコ横浜 国立大ホール

(10行)

nanamizuki=>

もしID=188,206,217 の記述が正しいなら、それ以外は間違っていることになるので訂正しなくてはなりません。苦労してテーブルを分けた結果がこういう時に報われます。

```
nanamizuki=> UPDATE events SET venue_id = 188 WHERE venue_id = 189;
UPDATE 1
```

```
nanamizuki=> UPDATE events SET venue_id = 206 WHERE venue_id = 205;
UPDATE 2
```

```
nanamizuki=> UPDATE events SET venue_id = 217 WHERE venue_id = 218;
UPDATE 2
```

```
nanamizuki=> SELECT events.id,venue_id,venue FROM events LEFT OUTER JOIN
venues ON(venue_id=venues.id) WHERE venue_id IN(189,188,205,206,217,218);
```

id	venue_id	venue
336	217	名古屋国際会議場 センチュリーホール
314	188	Zepp Tokyo
298	188	Zepp Tokyo
313	188	Zepp Tokyo
303	188	Zepp Tokyo
333	206	パシフィコ横浜 国立大ホール
334	206	パシフィコ横浜 国立大ホール
347	217	名古屋国際会議場 センチュリーホール
348	217	名古屋国際会議場 センチュリーホール
325	206	パシフィコ横浜 国立大ホール

(10行)

```
nanamizuki=>
```

テーブルeventsの項目venue_idの差し替えだけで、公演場所の誤記が訂正されました。

曲テーブルもざっと眺めてみます。

```
nanamizuki=> SELECT id,song_title FROM songs ORDER BY 2;
```

```
id | song_title
```

```
-----+-----  
641 | 1986年のマリリン
```

```
642 | 76th Star
```

```
643 | 7COLORS
```

```
-- 中略--
```

```
791 | Synchrogazer
```

```
792 | TRANMIGRATION
```

```
793 | TRANSMIGRATION
```

```
794 | TRANSMIGRATION 2007
```

```
795 | TRY AGAIN
```

```
796 | Take a Shot
```

```
-- 以下略--
```

```
(320行)
```

```
nanamizuki=>
```

ちょっと怪しげ。

```
792 | TRANMIGRATION
```

```
793 | TRANSMIGRATION
```

```
794 | TRANSMIGRATION 2007
```

誤表記をワザと掲載している訳でないならID=792は誤りでID=793が正解ですね。

```
nanamizuki=> SELECT id, event_id, song_id FROM setlists WHERE song_id=792;
```

```
id | event_id | song_id
```

```
-----+-----+-----  
6723 | 296 | 792
```

```
6704 | 295 | 792
```

```
6744 | 297 | 792
```

```
6764 | 298 | 792
```

```
6765 | 299 | 792
```

```
6786 | 300 | 792
```

```
6808 | 301 | 792
```

```
6829 | 302 | 792
```

```
6850 | 303 | 792
```

```
(9行)
```

```
nanamizuki=>
```

9つの公演のセットリストに誤りがあることがわかります。修正しちゃいましょう。

```

nanamizuki=> UPDATE setlists
nanamizuki->     SET song_id=793
nanamizuki->     ,comments='change 792 to 793 '||now()
nanamizuki->     WHERE song_id=792
nanamizuki-> ;
UPDATE 9
nanamizuki=> SELECT id, event_id, song_id, comments
nanamizuki->     FROM setlists
nanamizuki->     WHERE song_id IN (792,793)
nanamizuki-> ;
  id | event_id | song_id |
-----+-----+-----+-----
-
6744 |      297 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6764 |      298 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6765 |      299 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6786 |      300 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6808 |      301 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6829 |      302 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6850 |      303 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6723 |      296 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6704 |      295 |      793 | CHANGE 792 TO 793 2015-12-08 15:04:12.189903+00
6872 |      304 |      793 |
6893 |      305 |      793 |
6915 |      306 |      793 |
6937 |      307 |      793 |
6959 |      308 |      793 |
6985 |      309 |      793 |
7013 |      310 |      793 |
7033 |      311 |      793 |
7093 |      314 |      793 |
7113 |      315 |      793 |
7133 |      316 |      793 |
7157 |      317 |      793 |
7176 |      318 |      793 |
7198 |      319 |      793 |
7221 |      320 |      793 |
7241 |      321 |      793 |
7261 |      322 |      793 |
7281 |      323 |      793 |
7301 |      324 |      793 |
7321 |      325 |      793 |
7383 |      328 |      793 |
7447 |      331 |      793 |
7467 |      332 |      793 |
7489 |      333 |      793 |
7739 |      344 |      793 |
7961 |      353 |      793 |
7985 |      354 |      793 |

```


-- 中略--

```

233 | NANA MIZUKI LIVEDOM 2006-BIRTH- | 2006-01-21 |
2006-01-21 |
234 | 水樹奈々 21ANNIVERSARY CONCERT "HAPPY" | 2001-01-21 | 2001-01-21 |
|
235 | 水樹奈々 X'mas LIVE "supersonic girl" | 2001-12-23 | 2001-12-23 |
(29行)

```

```

nanamizuki=> SELECT * FROM events WHERE tour_id=235;
id | tour_id | event_date | venue_id | place_id | comments | stage
-----+-----+-----+-----+-----+-----+-----
294 | 235 | 2001-12-23 | 216 | 380 | | 1
(1行)

```

```

nanamizuki=> INSERT INTO events(tour_id, event_date, venue_id, place_id)
nanamizuki-> SELECT tour_id
nanamizuki->      ,event_date
nanamizuki->      ,venue_id
nanamizuki->      ,place_id
nanamizuki-> FROM events
nanamizuki-> WHERE tour_id=235
nanamizuki-> ;
INSERT 0 1

```

```

nanamizuki=> SELECT * FROM events WHERE tour_id=235;
id | tour_id | event_date | venue_id | place_id | comments | stage
-----+-----+-----+-----+-----+-----+-----
437 | 235 | 2001-12-23 | 216 | 380 | | 1
294 | 235 | 2001-12-23 | 216 | 380 | | 1
(2行)

```

```

nanamizuki=> SELECT * FROM setlists WHERE event_id=294 ORDER BY id,event_id,
list_type, order_index;
id | event_id | list_type | order_index | song_id | comments
-----+-----+-----+-----+-----+-----
6654 | 294 | 1 | 1 | 721 |
6655 | 294 | 1 | 2 | 696 |
6656 | 294 | 1 | 3 | 710 |
6657 | 294 | 1 | 4 | 861 |
6658 | 294 | 1 | 5 | 926 |
6659 | 294 | 1 | 6 | 936 |
6660 | 294 | 1 | 7 | 718 |
6661 | 294 | 1 | 8 | 716 |
6662 | 294 | 1 | 9 | 812 |
6663 | 294 | 1 | 10 | 733 |
6664 | 294 | 1 | 11 | 865 |
6665 | 294 | 1 | 12 | 833 |
6666 | 294 | 1 | 13 | 799 |
6667 | 294 | 2 | 1 | 793 |
6668 | 294 | 2 | 2 | 854 |

```

6669	294	3	1	945
6670	294	1	1	721
6671	294	1	2	696
6672	294	1	3	710
6673	294	1	4	861
6674	294	1	5	926
6675	294	1	6	936
6676	294	1	7	718
6677	294	1	8	716
6678	294	1	9	812
6679	294	1	10	733
6680	294	1	11	865
6681	294	1	12	833
6682	294	1	13	799
6683	294	2	1	793
6684	294	2	2	854
6685	294	3	1	945

(32行)

```
nanamizuki=> UPDATE setlists
nanamizuki->   SET event_id = 437
nanamizuki->   WHERE id >= 6670
nanamizuki->   AND id <= 6685
nanamizuki-> ;
```

UPDATE 16

```
nanamizuki=> SELECT * FROM setlists WHERE event_id=294 ORDER BY id,event_id,
list_type, order_index;
```

id	event_id	list_type	order_index	song_id	comments
6654	294	1	1	721	
6655	294	1	2	696	
6656	294	1	3	710	
6657	294	1	4	861	
6658	294	1	5	926	
6659	294	1	6	936	
6660	294	1	7	718	
6661	294	1	8	716	
6662	294	1	9	812	
6663	294	1	10	733	
6664	294	1	11	865	
6665	294	1	12	833	
6666	294	1	13	799	
6667	294	2	1	793	
6668	294	2	2	854	
6669	294	3	1	945	

(16行)

```
nanamizuki=> SELECT * FROM setlists WHERE event_id=437 ORDER BY id,event_id,
list_type, order_index;
```

id	event_id	list_type	order_index	song_id	comments
----	----------	-----------	-------------	---------	----------

6670	437	1	1	721
6671	437	1	2	696
6672	437	1	3	710
6673	437	1	4	861
6674	437	1	5	926
6675	437	1	6	936
6676	437	1	7	718
6677	437	1	8	716
6678	437	1	9	812
6679	437	1	10	733
6680	437	1	11	865
6681	437	1	12	833
6682	437	1	13	799
6683	437	2	1	793
6684	437	2	2	854
6685	437	3	1	945

(16行)

```

nanamizuki=> WITH t AS (
nanamizuki(> SELECT id
nanamizuki(>           ,event_date
nanamizuki(>           ,tour_id
nanamizuki(>           ,place_id
nanamizuki(>           ,venue_id
nanamizuki(>           ,ROW_NUMBER() OVER(partition BY tour_id ORDER BY
event_date, id) AS stage
nanamizuki(> FROM events
nanamizuki(> )
nanamizuki-> UPDATE events
nanamizuki->   SET stage = t.stage
nanamizuki-> FROM t
nanamizuki-> WHERE events.id           = t.id
nanamizuki->   AND events.event_date = t.event_date
nanamizuki->   AND events.tour_id     = t.tour_id
nanamizuki->   AND events.place_id   = t.place_id
nanamizuki->   AND events.venue_id  = t.venue_id
nanamizuki-> ;

```

UPDATE 146

```

nanamizuki=> SELECT * FROM events ORDER BY tour_id, event_date;

```

id	tour_id	event_date	venue_id	place_id	comments	stage
292	207	2000-01-23	261	380		1
295	208	2002-11-23	180	365		1
296	208	2002-11-28	179	378		2
297	208	2002-11-30	248	380		3
298	208	2003-01-05	188	380		4
353	209	2010-02-14	257	386		1
354	209	2010-02-20	239	377		2
355	209	2010-03-07	211	366		3
356	209	2010-03-13	252	384		4
357	209	2010-03-14	252	384		5

358	209	2010-03-20	224	365	6
359	209	2010-03-27	241	378	7

-- 中略 --

326	233	2006-01-21	242	380	1
293	234	2001-01-21	207	380	1
294	235	2001-12-23	216	380	1
437	235	2001-12-23	216	380	2

(146 行)

nanamizuki=>

技術資料, SQL, www, perl, json, スクレイピング

From:
<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:
<https://wiki.hgotoh.jp/documents/other/memo02/other-040>

Last update: **2025/11/20 09:31**

