Codes for Creating Views in Couch DB

Author: Hao DUAN<548771> Yu SUN<629341>
Date: 29-OCT-2015

Notice:

Please using below codes to create views in couch DB, and stored views below document name "mapreduce" in corresponding DB, and make sure name view as same as this document.

List of Views in couch DB

No.	View Name	DB Name	Function Description
1.	weathersearch	weatherdb	Offer weather data to web site
2.	cityDateText	weatherdb	Offer all cities' weather data to integration program
3.	Melbourne	weatherdb	Offer Melbourne weather data to integration program
4.	cityDate	cities	Offer each cities' daily emotion to integration program
5.	australia	cities	Offer 8 cities' tweets' geo-info on specific day to website
6.	heatmap	melbourne_suburb	Offer elbourne tweets' geo-info on specific day to website
7.	coastshop	melbourne_suburb	Offer daily emotion on specific areas (coast/shopping mall) to integration program
8.	coastshophour	melbourne_suburb	Offer hourly emotion on specific areas (coast/shopping mall) to integration program
9.	cityDatePNN	citysummary	Offer each cities' daily emotion to web site
10.	nega_con	citysummary	Offer 8 cities' total negative emotion data on specific weather to website for drawing pie graph
11.	posi_con	citysummary	Offer 8 cities' total positive emotion data on specific weather to website for drawing pie graph
12.	tweet_con	citysummary	Offer 8 cities' total tweets number to website for drawing bar graph
13.	emotionTemp	citysummary	Offer 8 cities' emotion data, temperature data and tweets number to website
14.	region	coastshop	Offer Melbourne suburbs' emotion data, temperature data and tweets number to website
15.	nega_aver	coastshop	Offer Melbourne suburbs' total negative emotion data on specific weather to website for drawing pie graph
16.	posi_aver	coastshop	Offer Melbourne suburbs' total

			positive emotion data on specific weather to website for drawing pie graph
17.	tweet_average	coastshop	Offer Melbourne suburbs' total tweets number to website for drawing bar graph

1.View 1:

Description	
Name	weathersearch
Purpose	Offer weather data to web site
Belonging DB Name	weatherdb
Code	
Map Function	

function(doc) {
  var cityName = doc['city']
  emit(cityName,[doc['date'],doc['high'],doc['low'],doc['text']]);
}	

Reduce Function	
-	

2. View 2:

Description	
Name	cityDateText
Purpose	Offer all cities' weather data to integration program
Belonging DB Name	weatherdb
Code	
Map Function	

function(doc) {
  var cityName = doc['city']
  emit([cityName,doc['date'],doc['text'],doc['high'],doc['low']],null);
}	

Reduce Function	
-	

3. View 3:

Description	
Name	Melbourne
Purpose	Offer Melbourne weather data to integration program
Belonging DB Name	weatherdb
Code	
Map Function	

function(doc) {
  var cityName = doc['city'];
if(cityName == 'Melbourne') {
  emit([cityName,doc['date'],doc['text'],doc['high'],doc['low']],null);}}	

Reduce Function	
-	

4. View 4:

Description	
Name	cityDate
Purpose	Offer each cities' daily emotion to integration program
Belonging DB Name	cities

Code

Map Function

```
function(doc) {
    var list = doc.TweetTime.split(" ");
    var a = list[2].split("");
    var b = parseInt(a[0])
    if(b==0)
    {
        var timeday = a[1].concat(" ",list[1]," ",list[5])
        emit([timeday,    doc.city, doc.prediction] , 1);
    }
    else
    {
        var timeday = list[2].concat(" ",list[1]," ",list[5])
        emit([timeday,    doc.city, doc.prediction] , 1);

    }
```

Reduce Function

```
function (key, values, rereduce) {
    if (!rereduce){
        var length = values.length
        return [sum(values), length]
    }else{
        var length = sum(values.map(function(v){return v[1]}))
        var result = sum(values.map(function(v){
            return v[0]
            }))
        return [result, length]
    }
}
```

5. View 5:

Description	
Name	austalia
Purpose	Offer 8 cities' tweets' geo-info on specific day to website
Belonging DB Name	cities
Code	
Map Function	

```
function(doc) {
    var list = doc.TweetTime.split(" ");
    var a = list[2].split("");
    var b = parseInt(a[0])
    if(b==0)
    {
        var timeday = a[1].concat(" ",list[1]," ",list[5])
        emit(timeday, doc.geoInfo.locations);
    }
    else
    {
        var timeday = list[2].concat(" ",list[1]," ",list[5])
        emit(timeday, doc.geoInfo.locations);

    }
}
```

Reduce Function
-

6. View 6:

Description	
Name	heatmap
Purpose	Offer melbourne tweets' geo-info on specific day to website
Belonging DB Name	melbourne_suburb
Code	
Map Function	

```
function(doc) {
    var list = doc.TweetTime.split(" ");
    var a = list[2].split("");
    var b = parseInt(a[0])
    if(b==0)
    {
        var timeday = a[1].concat(" ",list[1]," ",list[5])
        emit(timeday,doc.geoInfo.locations);
    }
```

```
    else
    {
        var timeday = list[2].concat(" ",list[1]," ",list[5])
        emit(timeday, doc.geoInfo.locations);

    }
```

Reduce Function

-

7. View 7:

Description	
Name	coastshop
Purpose	Offer daily emotion on specific areas (coast/shopping mall) to integration program
Belonging DB Name	melbourne_suburb

Code

Map Function

```
function(doc) {
    var list = doc.TweetTime.split(" ");
    var a = list[2].split("");
    var b = parseInt(a[0]);
var myarr = ["Altona","Altona Meadows","Seaholme","Williamstown","Port Melbourne","Newport","Albert Park","St Kilda West","St Kilda","Elwood","Brighton","Middle Park"];
var shop = ["Maribyrnong","South Wharf","Melbourne","Malvern East","Essendon Fields"];
if(b==0)
    {
var timeday = a[1].concat(" ",list[1]," ",list[5])
if(myarr.indexOf(doc.suburb) > -1)
{
emit([timeday,"coast", doc.prediction], 1);
}
else if(shop.indexOf(doc.suburb) > -1)
{
emit([timeday,"shop", doc.prediction], 1);
}
else
{
emit([timeday,"other", doc.prediction], 1);
}

    }
```

```
      else
      {
            var timeday = list[2].concat(" ",list[1]," ",list[5])
if(myarr.indexOf(doc.suburb) > -1)
{
emit([timeday,"coast", doc.prediction], 1);
}
else if(shop.indexOf(doc.suburb) > -1)
{
emit([timeday,"shop", doc.prediction], 1);
}
else
{
emit([timeday,"other", doc.prediction], 1);
}

      }


}
```

Reduce Function

```
function (key, values, rereduce) {
      if (!rereduce){
            var length = values.length
            return [sum(values), length]
      }else{
            var length = sum(values.map(function(v){return v[1]}))
            var result = sum(values.map(function(v){
                  return v[0]
                  }))
            return [result, length]
      }
}
```

8. View 8:

Description	
Name	coastshophour
Purpose	Offer hourly emotion on specific areas (coast/shopping mall) to integration program
Belonging DB Name	melbourne_suburb

Code

Map Function

```
function(doc) {
    var list = doc.TweetTime.split(" ");
    var a = list[2].split("");
    var hour = list[3].split(":");
    var b = parseInt(a[0]);
var month_names_short = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
var myarr = ["Altona","Altona Meadows","Seaholme","Williamstown","Port Melbourne","Newport","Albert Park","St Kilda West","St Kilda","Elwood","Brighton","Middle Park"];
var shop = ["Maribyrnong","South Wharf","Melbourne","Malvern East","Essendon Fields"];
if(b==0)
    {
var newmonth = month_names_short.indexOf(list[1])+1;
var timeday = list[2].concat(" ",newmonth," ",list[5]," ",hour[0]," ","00"," ","00")
if(myarr.indexOf(doc.suburb) > -1)
{
emit([timeday,"coast", doc.prediction], 1);
}
else if(shop.indexOf(doc.suburb) > -1)
{
emit([timeday,"shop", doc.prediction], 1);
}
else
{
emit([timeday,"other", doc.prediction], 1);
}

    }
    else
    {
var newmonth = month_names_short.indexOf(list[1])+1;
var timeday = list[2].concat(" ",newmonth," ",list[5]," ",hour[0]," ","00"," ","00")
if(myarr.indexOf(doc.suburb) > -1)
{
```

```
emit([timeday,"coast", doc.prediction], 1);
}
else if(shop.indexOf(doc.suburb) > -1)
{
emit([timeday,"shop", doc.prediction], 1);
}
else
{
emit([timeday,"other", doc.prediction], 1);
}

    }


}
```

Reduce Function

```
function (key, values, rereduce) {
    if (!rereduce){
        var length = values.length
        return [sum(values), length]
    }else{
        var length = sum(values.map(function(v){return v[1]}))
        var result = sum(values.map(function(v){
            return v[0]
            }))
        return [result, length]
    }
}
```

9.View 9:

Description	
Name	cityDatePNN
Purpose	Offer each cities' daily emotion to web site
Belonging DB Name	citysummary
Code	
Map Function	

```
function(doc) {

  var cityName = doc['city']
  emit(cityName,[doc['date'],doc['positive'],doc['netrual'],doc['negative']]);
}
```

Reduce Function

-

10. View 10:

Description	
Name	nega_con
Purpose	Offer 8 cities' total negative emotion data on specific weather to website for drawing pie graph
Belonging DB Name	citysummary
Code	
Map Function	

```
function(doc) {

var positive;
var netrual;
var negative;
var tweet;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
{
```

```
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else                    if(doc['condition'].indexOf('hower')>-1                    ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}
   emit(condition,negative);
}
```

Reduce Function

```
function (key, values, rereduce) {
	if (!rereduce){
		var length = values.length
		return [sum(values)/length, length]
	}else{
		var length = sum(values.map(function(v){return v[1]}))
		var result = sum(values.map(function(v){
			return v[0] * (v[1] / length)
			}))
		return [result, length]
	}
}
```

11. View 11

Description	
Name	posi_con
Purpose	Offer 8 cities' total positive emotion data on specific weather to website for drawing pie graph
Belonging DB Name	citysummary
Code	
Map Function	

```
function(doc) {

var positive;
var netrual;
var negative;
var tweet;
var condition;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
```

```
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else                    if(doc['condition'].indexOf('hower')>-1                    ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}

   emit(condition,positive);
}
```

Reduce Function

```
function (key, values, rereduce) {
      if (!rereduce){
            var length = values.length
            return [sum(values)/length, length]
      }else{
            var length = sum(values.map(function(v){return v[1]}))
            var result = sum(values.map(function(v){
                  return v[0] * (v[1] / length)
                  }))
            return [result, length]
      }
```

```
}
```

12.View 12

Description	
Name	tweet_con
Purpose	Offer total tweets number to website for drawing bar graph
Belonging DB Name	citysummary

Code

Map Function

```
function(doc) {

var positive;
var netrual;
var negative;
var tweet;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}
```

```
if(isNaN(doc['tweets']))
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}
if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else if(doc['condition'].indexOf('hower')>-1 ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}
   emit(condition,tweet);
}
```

Reduce Function

```
function (key, values, rereduce) {
      if (!rereduce){
            var length = values.length
            return [sum(values)/length, length]
      }else{
            var length = sum(values.map(function(v){return v[1]}))
            var result = sum(values.map(function(v){
                  return v[0] * (v[1] / length)
                  }))
            return [result, length]
      }
}
```

13. View 13

Description	
Name	emotionTemp
Purpose	Offer 8cities' emotion data, temperature data and tweets number to website
Belonging DB Name	citysummary
Code	
Map Function	

```
function(doc) {

   var cityName = doc['city']
var positive;
var netrual;
var negative;
var tweet;
var diff = parseFloat(doc['high'])-parseFloat(doc['low']);
if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
```

```
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else if(doc['condition'].indexOf('hower')>-1 ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}

emit(cityName,[doc['date'],positive,netrual,negative,doc['high'],doc['low'],diff,do
c['tweets'],condition]);
}
```

Reduce Function

-

14.View 14

Description	
Name	region
Purpose	Offer Melbourne suburbs' emotion data, temperature data and tweets number to website
Belonging DB Name	coastshop
Code	
Map Function	

```
function(doc) {

   var cityName = doc['region'];
var positive;
var netrual;
var negative;
var tweet;
var diff = parseFloat(doc['high'])-parseFloat(doc['low']);
if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
```

```
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

emit(cityName,[doc['date'],positive,netrual,negative,doc['high'],doc['low'],diff,doc['tweets'],doc['condition']]);
}
```

Reduce Function

-

15. View 15

Description	
Name	nega_aver
Purpose	Offer Melbourne suburbs' total negative emotion data on specific weather to website for drawing pie graph
Belonging DB Name	coastshop

Code

Map Function

```
function(doc) {

   var cityName = doc['region'];
var positive;
var netrual;
var negative;
var tweet;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
else
```

```
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else                    if(doc['condition'].indexOf('hower')>-1                    ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}
   emit(condition,negative);
```

```
}
```

Reduce Function

```
function (key, values, rereduce) {
	if (!rereduce){
		var length = values.length
		return [sum(values)/length, length]
	}else{
		var length = sum(values.map(function(v){return v[1]}))
		var result = sum(values.map(function(v){
			return v[0] * (v[1] / length)
			}))
		return [result, length]
	}
}
```

16. View 16

Description	
Name	posi_aver
Purpose	Offer Melbourne suburbs' total positive emotion data on specific weather to website for drawing pie graph
Belonging DB Name	coastshop

Code

Map Function

```
function(doc) {

   var cityName = doc['region'];
var positive;
var netrual;
var negative;
var tweet;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
}
```

```
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else                    if(doc['condition'].indexOf('hower')>-1                    ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
}
```

```
    emit(condition,positive);
}
```

Reduce Function

```
function (key, values, rereduce) {
        if (!rereduce){
                var length = values.length
                return [sum(values)/length, length]
        }else{
                var length = sum(values.map(function(v){return v[1]}))
                var result = sum(values.map(function(v){
                        return v[0] * (v[1] / length)
                        }))
                return [result, length]
        }
}
```

17. View 17

Description	
Name	tweet_average
Purpose	Offer Melbourne suburbs' total tweets number to website for drawing bar graph
Belonging DB Name	coastshop

Code

Map Function

```
function(doc) {

    var cityName = doc['region'];
var positive;
var netrual;
var negative;
var tweet;

if(isNaN(doc['positive']))
{
  positive = 0;
}
else
{
positive = doc['positive'];
}

if(isNaN(doc['netrual']))
{
  netrual = 0;
```

```
}
else
{
netrual = doc['netrual'];
}

if(isNaN(doc['negative']))
{
  negative = 0;
}
else
{
negative = doc['negative'];
}

if(isNaN(doc['tweets']))
{
  tweet = 0;
}
else
{
tweet = doc['tweets'];
}

if(doc['condition'].indexOf('loud')>-1)
{
condition = "Cloud";
}
else if(doc['condition'].indexOf('lear')>-1)
{
condition = "Clear";
}
else                    if(doc['condition'].indexOf('hower')>-1                    ||
doc['condition'].indexOf('hunder')>-1 || doc['condition'].indexOf('ain')>-1)
{
condition = "Rain";
}
else if(doc['condition'].indexOf('ind')>-1)
{
condition = "Wind";
}
else if(doc['condition'].indexOf('unny')>-1)
{
condition = "Sunny";
```

```
}
   emit(condition,doc['tweets']);
}
```

Reduce Function

```
function (key, values, rereduce) {
      if (!rereduce){
            var length = values.length
            return [sum(values)/length, length]
      }else{
            var length = sum(values.map(function(v){return v[1]}))
            var result = sum(values.map(function(v){
                  return v[0] * (v[1] / length)
                  }))
            return [result, length]
      }
}
```