



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Base de datos NOSql

Trabajo Práctico 2

Base de datos

Integrante	LU	Correo electrónico
Guillamon, Gonzalo	97/12	gonzaguillamon@gmail.com
Toffoletti, Luis	827/11	luis.toffoletti@gmail.com
Zanollo, Florencia	934/11	florenciazanollo@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



**Facultad de Ciencias Exactas y
Naturales**

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta
Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep.
Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

Contents

1	Introducción	2
2	Consultas SQL	3
3	Map Reduce	4
3.1	El importe total de ventas por usuario	4
3.2	La reputación historica de cada usuario según la calificación . . .	5
3.3	Las operaciones con comisión más alta	5
3.4	El monto total facturado por año	6
3.5	El monto total facturado por año de las operaciones pertenecientes a usuarios con suscripciones Rubí Oriente	7
3.6	El total de las publicaciones por tipo de publicación (productos, servicios, mixtas	7
4	Sharding	9

1 Introducción

2 Consultas SQL

3 Map Reduce

MapReduce es un modelo de programación para el procesamiento y creación de grandes colecciones de datos.

Consiste de dos partes:

- Map() se encarga de filtrar y organizar los datos, produciendo una lista de pares (clave, valor) por cada llamada. Luego de esto el framework agrupa los pares con la misma clave, creando un grupo por cada una de las diferentes claves generadas.
- Reduce(clave, valores) aplica funciones sobre los valores agrupados en el map. Por ejemplo, suma todos los valores, saca el promedio, etc. Retornando un resultado por cada clave.

Notar que estos procedimientos (Map y Reduce) son paralelizables, con lo cual pueden ser distribuidos en diferentes unidades (por ejemplo clusters) para su procesamiento.

A continuación listamos todas las consultas a implementar, utilizando MapReduce, requeridas, con su respectivo código y una pequeña explicación del mismo.

3.1 El importe total de ventas por usuario

```
1 var mapFunctionA = function(){
2   emit(this.vendedor, this.importe)
3 };
4
5 var reduceFunctionA = function(userId, importes){
6   return Array.sum(importes);
7 }
8
9 db.Ventas.mapReduce( mapFunctionA, reduceFunctionA, {out: "
  mapReduceA"} )
10
11 db.mapReduceA.find()
```

línea 9 Se aplica el MapReduce sobre la colección Ventas.

línea 2 Como clave se eligió el id del usuario que realiza la venta (i.e. el e-mail del vendedor) y el valor es el importe de la misma.

línea 5 Con esto se obtiene una lista por vendedor con todos los importes de sus compras.

línea 6 Los cuales son sumados y se retorna dicha suma.

3.2 La reputación historica de cada usuario según la calificación

```
1 var mapFunctionB = function(){
2   emit(this.vendedor, this.calificacionVendedor);
3   emit(this.comprador, this.calificacionComprador);
4 };
5
6 var reduceFunctionB = function(userId, calificaciones){
7   suma = Array.sum(calificaciones);
8   promedio = suma / calificaciones.length
9   return promedio;
10 }
11
12 db.Ventas.mapReduce( mapFunctionB, reduceFunctionB, {out: "
13   mapReduceB"} )
14
15 db.mapReduceB.find()
```

línea 12 Se aplica el MapReduce sobre la colección Ventas.

líneas 2 y 3 Se emiten dos datos, uno que contiene el id del vendedor y otro del comprador, ambos con la calificación correspondiente como valor.

línea 6 Con esto se obtiene una lista por usuario con todas sus calificaciones (tanto de compras como de ventas).

líneas 7 y 8 Se retorna el promedio de las calificaciones (por cada usuario).

3.3 Las operaciones con comisión más alta

```
1 404 NOT FOUND
```

3.4 El monto total facturado por año

```
1 var mapFunctionD = function(){
2   facturaciones = new Array();
3
4   if(this.hasOwnProperty('abonoFijo')){
5     var abono = this.abonoFijo.abono;
6     facturaciones.push(abono);
7   }
8
9   if(this.hasOwnProperty('comisiones')){
10    var comisiones = this.comisiones;
11
12    for (var i = 0; i < comisiones.length; i++) {
13      facturaciones.push(comisiones[i].comision)
14    }
15  }
16
17  emit(this.fecha.slice(this.fecha.length-2, this.fecha.length),
18    facturaciones)
19 };
20 var reduceFunctionD = function(year, facturaciones){
21   facturacionesFlattened = facturaciones.reduce(function(a, b){
22     return a.concat(b)}, []);
23
24   facturacionAnual = Array.sum(facturacionesFlattened);
25
26   return facturacionAnual;
27 }
28
29 db.Facturas.mapReduce( mapFunctionD, reduceFunctionD, {out: "
30   mapReduceD"} )
31 db.mapReduceD.find()
```

línea 29 Se aplica el MapReduce sobre la colección Facturas.

líneas 4 a 7 Si el usuario tiene facturación por abono fijo (i.e. Rubí Oriente) se agrega dicha facturación a la lista de facturaciones.

líneas 9 a 15 Si el usuario tiene una o más facturaciones por comisión se agregan todas ellas a la lista de facturaciones.

línea 17 Se utiliza como clave el año.

línea 20 Con esto se obtiene una lista por año con todas las facturaciones (de comisiones o abonos, de todos los usuarios).

líneas 24 y 26 Se suman todas las facturaciones y se retorna dicha suma.

3.5 El monto total facturado por año de las operaciones pertenecientes a usuarios con suscripciones Rubí Oriente

```
1 var mapFunctionE = function(){
2   facturaciones = new Array();
3
4   if(this.hasOwnProperty('abonoFijo')){
5     var abono = this.abonoFijo.abono;
6     facturaciones.push(abono);
7
8     if(this.hasOwnProperty('comisiones')){
9       var comisiones = this.comisiones;
10
11       for (var i = 0; i < comisiones.length; i++) {
12         facturaciones.push(comisiones[i].comision)
13       }
14     }
15
16     emit(this.fecha.slice(this.fecha.length-2, this.fecha.length),
17         facturaciones)
18   }
19 };
20
21 var reduceFunctionE = function(year, facturaciones){
22   var facturacionesFlattened = 0;
23   facturacionesFlattened = facturaciones.reduce(function(a, b){
24     return a.concat(b)}, []);
25
26   facturacionAnual = Array.sum(facturacionesFlattened);
27   return (facturacionAnual);
28 }
29 db.Facturas.mapReduce( mapFunctionE, reduceFunctionE, {out: "
30   mapReduceE" } )
31 db.mapReduceE.find()
```

La funcionalidad es igual al anterior pero en este caso sólo se emiten todas las facturaciones si el usuario posee un abono fijo (i.e. Rubí Oriente).

3.6 El total de las publicaciones por tipo de publicación (productos, servicios, mixtas)

```
1 var mapFunctionF = function(){
2   emit(this.tipoDePublicacion, this.id);
3 };
4
5 var reduceFunctionF = function(tipoDePublicacion, publicaciones){
6   return publicaciones.length;
7 }
8
9 db.Publicaciones.mapReduce( mapFunctionF, reduceFunctionF, {out: "
  mapReduceF"} )
10
11 db.mapReduceF.find()
```

línea 9 Se aplica el MapReduce sobre la colección Publicaciones.

línea 2 Se toma el tipo de publicación como clave, el valor es el id (único para cada publicación).

línea 5 Con esto se obtiene una lista de todas las publicaciones agrupadas por tipo de publicación.

línea 6 Se retorna la cantidad de publicaciones en cada grupo.

4 Sharding