



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Bases de datos

Grupo 7

Integrante	LU	Correo electrónico
Lavia, Alejandro	43/11	lavia.alejandro@gmail.com
Simón, Jorge		jorgesm595@gmail.com
Rey, Esteban	657/10	estebanlucianorey@gmail.com
Tripodi, Guido	843/10	guido.tripodi@hotmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

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

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

Índice

1. Descripción de problema	2
2. Modelo conceptual y modelo de interacción de documentos	3
2.1. Modelo conceptual	3
2.2. Modelo de interrelación de documentos	3
3. Diseño lógico	4
4. Queries	8
4.1. Lista de atracciones visitadas por un cliente dado	8
4.2. Para un parque dado el ranking de atracciones por cantidad de visitas.	8
4.3. Top 5 de Eventos con mayor facturación	8
4.4. Total de importes por atracción en un parque dado.	8
4.5. Para un parque dado el ranking de atracciones por cantidad de visitas utilizando Map-Reduce	9
5. Sharding	10
5.0.1. Experimentación	10
6. Aclaraciones para correr las implementaciones	13

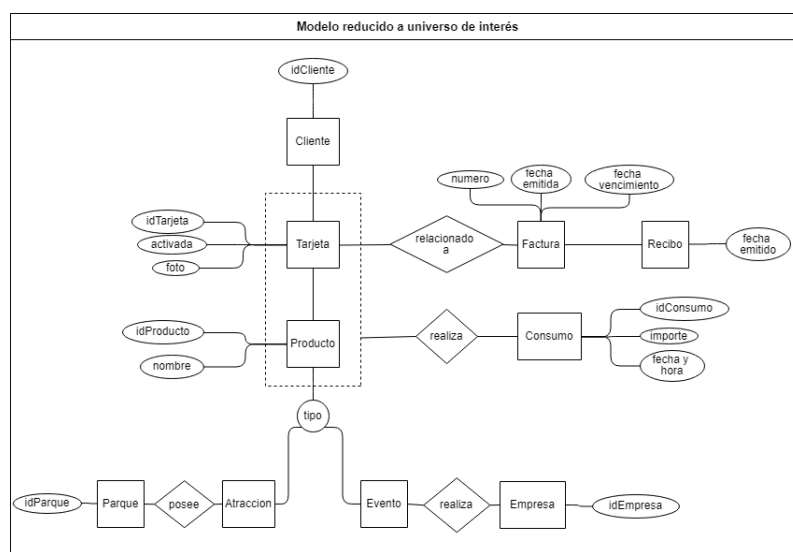
1. Descripción de problema

Tomando en cuenta el crecimiento de la base de datos por la enorme cantidad de usuarios de las tarjetas, se desea guardar el historial de visitas a parques, atracciones y eventos evitando afectar la base de datos relacional de transacciones implementada en el TP1. Se desea, además, guardar además el histórico de facturas.

2. Modelo conceptual y modelo de interacción de documentos

2.1. Modelo conceptual

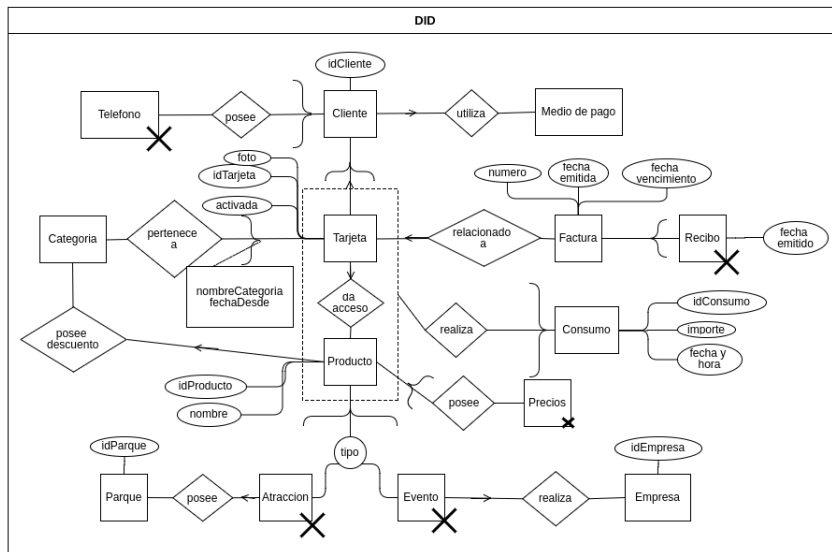
En esta primera etapa realizamos un diseño conceptual del problema utilizando la herramienta de Modelo de Entidad Relación. Para simplificar el diseño, a partir del MER obtuvimos un submodelo del mismo para poder clarificar el diseño del modelo de interrelación:



MER acotado al problema

2.2. Modelo de interrelación de documentos

Este lo construimos con la técnica de Diagrama Entidad relación, a continuación presentamos el diagrama obtenido:



Entidad Relación

Diagrama

3. Diseño lógico

```
{
  "Parque": {
    "type": "object",
    "properties": {
      "parqueId": { "type": "string" },
      "nombre": { "type": "string" },
      "ubicacion": { "type": "string" },
      "atracciones": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "productoId": { "type": "integer" },
            "edadDesde": { "type": "integer" },
            "edadHasta": { "type": "integer" }
          }
        }
      },
    },
  },
}

{
  "Producto": {
    "type": "object",

```

```
    "properties":{
      "tipo": {"type" : "string"},
      "empresa": {"type":"string"},
      "parque": {"type":"string"},
      "nombre": {"type":"string"},
      "categoriasDescuentos": {
        "type":"array"
        "items":{
          "type":"object",
          "properties":{
            "nombreCategoria":{"type":"string"},
            "descuento":{"type":"float"}
          }
        }
      },
      "precios": {
        "type":"array"
        "items":{
          "type":"object",
          "properties":{
            "dia":{"type":"string"},
            "valor":{"type":"float"}
          }
        }
      },
      "edadDesde":{"type":"string"},
      "edadHasta":{"type":"string"},
      "fechaDesde":{"type":"string"},
      "fechaHasta":{"type":"string"}
    }
  }
}
{
  {
    "Empresa":{
      "type":"object",
      "properties":{
        "empresaId": {"type":"string"},
        "pais": {"type":"string"},
        "provincia": {"type":"string"},
        "direccion": {"type":"string"},
        "razonSocial": {"type":"string"}
      }
    }
  }
}
{
  {
    "Categoria":{
      "nombre": {"type":"string"},
      "x": {"type":"float"},
      "y": {"type":"float"}
    }
  }
}
{
```

```
"Cliente":{
  "ClienteId":{"type": "integer"},
  "nombre":{"type": "string"},
  "apellido":{"type": "string"},
  "domicilioFacturacion":{"type": "string"},
  "foto":{"type": "string"},
  "nombreMedioPago":{"type": "string"},
  "tarjetas":{
    "type": "array",
    "items": {
      "type":"object",
      "properties":{"
        "numero":{"type":"integer"},
      }
    },
  },
  "telefonos":{
    "type": "array",
    "items": {
      "type":"object",
      "properties":{"
        "numero":{"type":"string"}
      }
    },
  },
}
}
{
  "Tarjeta": {
    "type":"object",
    "properties":{"
      "numero":{"type":"string"},
      "categoria":{"type":"string"},
      "activada":{"type":"boolean"},
      "idCliente":{"type":"integer"},
      "productos":{"
        "type": "array",
        "items": {
          "type":"object",
          "properties":{"
            "productoId":{"type":"integer"},
            "nombre":{"type":"string"}
          }
        }
      }
    }
  }
}
{
  "Factura":{
    "facturaId":{"type": "integer"},
    "fechaEmitida":{"type": "string"},
    "fechaVencimiento":{"type": "string"},
    "importe":{"type": "float"},
  }
}
```

```
        "tarjeta":{"type": "integer"}
    }
}

{
  "Consumo":{
    "idConsumo":{"type": "integer"},
    "importe":{"type": "float"},
    "fechaHora":{"type": "string"},
    "tarjeta":{"
      "type": "object",
      "propeties":{"
        "idCliente":{"type":"integer"},
        "numero":{"type":"integer"},
        "categoria":{"type":"string"}
      }
    },
    "producto":{
      "type": "object",
      "propeties":{"
        "tipo": {"type" : "string"},
        "parque":{"type":"integer"},
        "empresa":{"type":"integer"},
        "idProducto":{"type":"integer"},
        "nombre":{"type":"string"}
      }
    },
    "facturaId":{"type": "integer"}
  }
}
```


4. Queries

4.1. Lista de atracciones visitadas por un cliente dado

```
r.db('tp2').table('Consumo')  
.filter({tarjeta: {"idCliente": 1}, producto: {"tipo": "atraccion"}})
```

4.2. Para un parque dado el ranking de atracciones por cantidad de visitas.

```
r.db('tp2').table('Consumo')  
.filter({"producto": {"parque": 2}})  
.group("producto")  
.count()  
.ungroup()  
.orderBy(r.desc("reduction"))  
.map(function(consumoR){  
    return {"producto": consumoR('group'), "visitas": consumoR('reduction')}  
})
```

4.3. Top 5 de Eventos con mayor facturación

```
r.db('tp2').table('Consumo')  
.filter({"producto": {"tipo": "evento"}})  
.group("producto")  
.sum("importe")  
.ungroup()  
.orderBy(r.desc("reduction"))  
.map(function(consumo){  
    return {"producto": consumo('group'), "facturacion": consumo('reduction')}  
})  
.limit(5)
```

4.4. Total de importes por atracción en un parque dado.

```
r.db('tp2').table('Consumo')  
.filter({"producto": {"parque": 5}})  
.group("producto")  
.sum("importe")  
.ungroup()  
.map(function(consumo){  
    return {"producto": consumo('group'), "sumasImportes": consumo('reduction')}  
})
```

4.5. Para un parque dado el ranking de atracciones por cantidad de visitas utilizando Map-Reduce

```
r.db('tp2').table('Consumo')
  .filter({'producto':{'parque':2}})
  .map(function(consumo){
    return {'producto':consumo('producto'),'visitas':1}
  })
  .group('producto')
  .reduce(function(left, right){
    return {'visitas':left('visitas').add(right('visitas'))}
  })
  .ungroup()
  .map(function(o){
    return {"producto":o("group"),"visitas":o("reduction")("visitas")}
  })
  .orderBy(r.desc("visitas"))
```

5. Sharding

5.0.1. Experimentación

A continuación se muestran los gráficos obtenidos habiendo trabajado con **4 sharding y 1 réplica** por shard cargando registros en la tabla **Consumos**.

Se realizaron diversos experimentos cargando 1000, 5000, 10000 y 50000 registros, realizando gráficos cada 100.

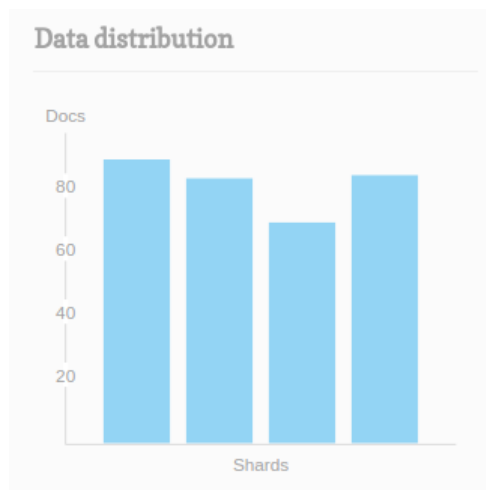


Grafico 5.1 - Sharding 4 shard - 1 Replica

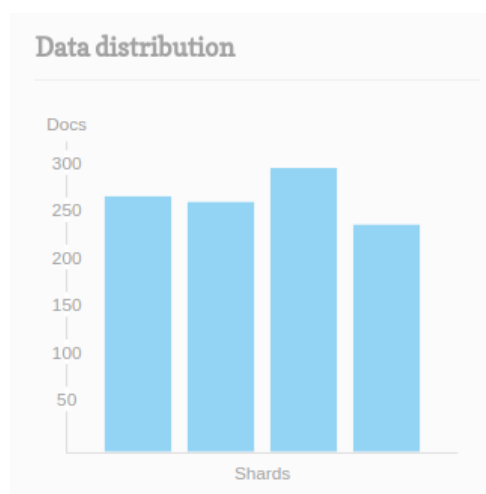


Grafico 5.2 - Sharding 4 shard - 1 Replica

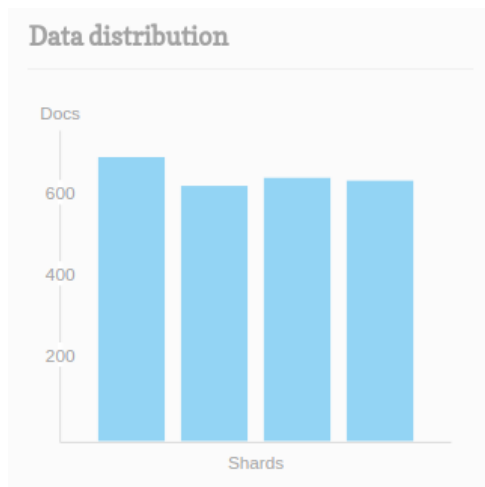


Grafico 5.2 - Sharding 4 shard - 1 Replica

Ademas realizamos para una mejor observación el comportamiento de la carga de registros con 4 réplicas por shard.

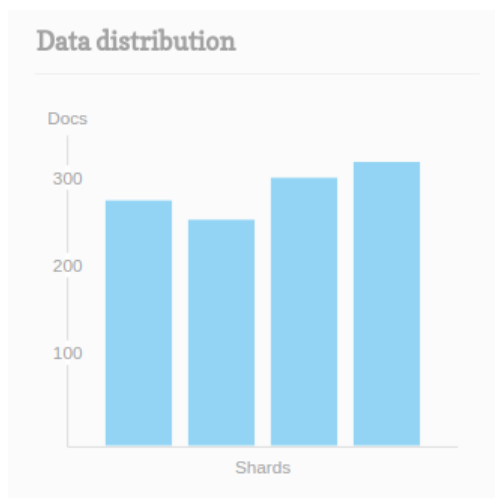


Grafico 5.1 - Sharding 4 shard - 4 Replica

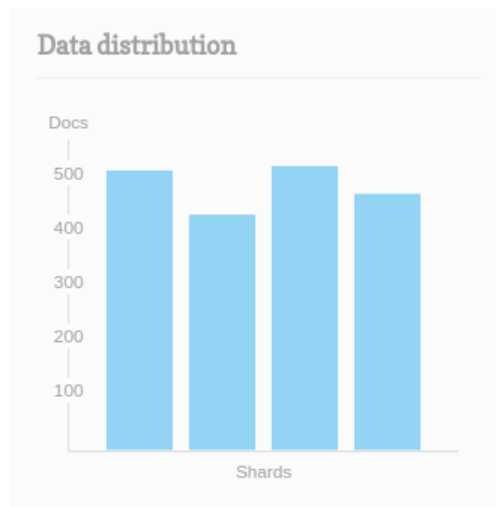


Grafico 5.2 - Sharding 4 shard - 4 Replica

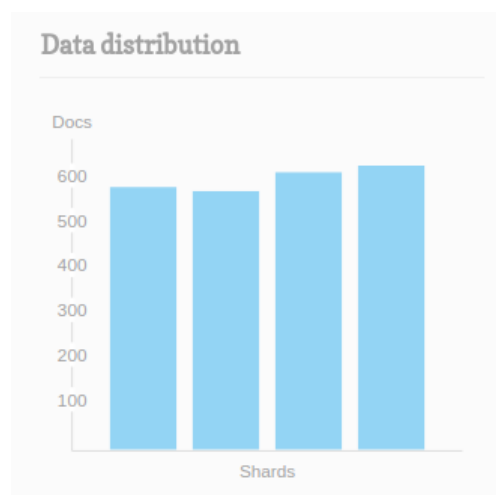


Grafico 5.2 - Sharding 4 shard - 4 Replica

Luego de la carga de todos los experimentos notamos que con más réplicas por shard la tasa de write/sec disminuía. Si se agregaban más réplicas mientras se iba realizando la carga y se despreciaba el tiempo en el que la ejecución de la misma se llevaba a cabo, la tasa nombrada disminuía, pero si se retiraban dichas réplicas, no solo volvía a aumentar la tasa sino que además se perdían datos.

6. Aclaraciones para correr las implementaciones

Para montar el proyecto se deberá tener instalado el motor de base de datos rethinkdb.

Para crear las tablas se genero un archivo en python **tp2.py** el cual se conecta al servidor de rethinkdb, crea la base "tp2" y carga los respectivos documentos.