



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e
Computação

Tecnologias de Bases de Dados

NoSQL: Instalações Culturais

Ana Rita Torres - up201406093@fe.up.pt
Catarina Correia - up201405765@fe.up.pt
José Oliveira - up201406208@fe.up.pt

5 de junho de 2018

Conteúdo

1	Introdução	2
2	Modelo	3
3	População	5
4	<i>Queries</i>	6
4.1	<i>Query</i> 3.a	6
4.1.1	<i>Query</i>	6
4.1.2	Resultado	6
4.1.3	<i>Query</i> em SQL	7
4.2	<i>Query</i> 3.b	7
4.2.1	<i>Query</i>	8
4.2.2	Resultado	8
4.2.3	<i>Query</i> em SQL	9
4.3	<i>Query</i> 3.c	9
4.3.1	<i>Query</i>	9
4.3.2	Resultado	10
4.3.3	<i>Query</i> em SQL	10
4.4	<i>Query</i> 3.d	10
4.4.1	<i>Query</i>	11
4.4.2	Resultado	11
4.4.3	<i>Query</i> em SQL	12
4.5	<i>Query</i> 3.e	13
4.5.1	<i>Query</i>	13
4.5.2	Resultado	13
4.5.3	<i>Query</i> em SQL	19
4.6	<i>Query</i> 3.f	20
4.6.1	<i>Query</i>	20
4.6.2	Resultado	20
4.6.3	<i>Query</i> em SQL	20
5	Mongo vs Oracle implementations	21
5.1	Tempo de Processamento	21
5.1.1	Conclusão	21
5.2	Tamanho da informação	21
5.2.1	Conclusão	22
5.3	Facilidade da <i>Query</i>	22

1 Introdução

No âmbito da disciplina de "Tecnologias e Bases de Dados", foi pedido ao grupo que elaborasse um trabalho em NoSQL, relativamente ao tema "Instalações Culturais".

Num primeira fase, vai ser mostrado o modelo implementado pelo grupo para solucionar o problema proposto. Numa segunda fase, será mostrado como foi feita a população da nova base de dados. Numa fase posterior, serão mostradas as *queries* elaboradas pelo grupo para dar resposta às perguntas do enunciado. Por último, será feita uma comparação entre as implementações em Mongo e em Oracle. Mais propriamente, será feita uma análise das vantagens e desvantagens do uso de SQL e NoSQL.

2 Modelo

O grupo optou por criar as seguintes coleções: "FACILITIES", "MUNICIPALITIES", "DISTRICTS" e "REGIONS".

As escolhas foram baseadas em diversos fatores. Primeiro, o grupo tentou agrupar tanta informação quanto possível num só documento. Tal escolha deve-se ao facto de não fazer sentido ter dados numa coleção à parte que só são usados quando o documento "pai" é chamado. Assim, evita-se uma série de joins que em SQL seriam inevitáveis. Segundo, o grupo tentou separar os dados que são referidos múltiplas vezes numa coleção, para garantir consistência dos dados.

Deste modo, é possível ver várias tabelas numa só coleção (exemplo: "FACILITIES" contém "ACTIVITIES").

O grupo optou pelo seguinte modelo de documentos:

```
//facilities definition
{
  id: {type: Number, required: true, unique: true},
  nome: {type: String, required: true},
  lotacao: {type: Number, required: true},
  tipo: {type: String, required: true},
  morada: {type: String, required: true},
  concelho: [
    id: {type: Number, required: true, unique: true},
    designacao: {type: String, required: true},
    distrito: [
      id: {type: Number, required: true, unique: true},
      designacao: {type: String, required: true},
    ],
    regiao: [
      id: {type: Number, unique: true},
      designacao: {type: String},
      NUT: {type: String}
    ]
  ],
  actividades: [
    actividade: {type: String, required: true}
  ]
}

//municipalities definition
{
  id: {type: Number, required: true, unique: true},
  designacao: {type: String, required: true},
  distrito: [
    id: {type: Number, required: true, unique: true},
```

```

        designacao: {type: String, required: true},
    ],
    regiao: [
        id: {type: Number, unique: true},
        designacao: {type: String},
        NUT: {type: String}
    ]
)

//districts definition
{
    id: {type: Number, required: true, unique: true},
    designacao: {type: String, required: true}
}

//regions definition
{
    id: {type: Number, unique: true},
    designacao: {type: String},
    NUT: {type: String}
}

```

3 População

Para popular a base de dados, exportou-se no formato *json* os dados da tabela GDT8 necessários (através do *sqldeveloper* e com *encoding "utf8"*). De seguida, usaram-se *scripts* em *Python* desenvolvidos para este propósito que processam o *json* obtido e o colocam num formato concordante com o modelo que se definiu no ponto 2. São também usados alguns ficheiros *batch* para automatizar a chamada dos *scripts*. O ficheiro *Python* assim como um *readme* seguem em anexo.

4 *Queries*

As *queries* abaixo representadas foram usadas para obter os resultados relativas às questões propostas no enunciado.

4.1 *Query 3.a*

Which are the facilities where the room type description contains ‘touro’ and have ‘teatro’ as one of their activities? Show the id, name, description and activity.

4.1.1 *Query*

```
db.getCollection('recintos').find(
  {
    "tipo":
    {
      $regex : "touro"
    },
    "atividades": "teatro"
  },
  {
    "nome": 1,
    "tipo": 1,
    "atividades": 1
  }
)
```

4.1.2 *Resultado*

```
/* 1 */
{
  "_id" : 916,
  "nome" : "COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS",
  "tipo" : "Praça de touros multiusos",
  "atividades" : [
    "dança",
    "música",
    "tauromaquia",
    "teatro"
  ]
}

/* 2 */
{
```

```

    "_id" : 940,
    "nome" : "ARENA DE ÉVORA - EX PRAÇA DE TOIROS",
    "tipo" : "Praça de touros multiusos",
    "atividades" : [
        "dança",
        "música",
        "tauromaquia",
        "teatro"
    ]
}

/* 3 */
{
    "_id" : 957,
    "nome" : "COLISEU DE REDONDO - EX PRAÇA DE TOIROS",
    "tipo" : "Praça de touros multiusos",
    "atividades" : [
        "dança",
        "música",
        "tauromaquia",
        "teatro"
    ]
}

```

4.1.3 *Query* em SQL

```

select distinct gtd8.facilities.id, name, description, activity
from gtd8.facilities
join gtd8.roomtypes
on gtd8.facilities.roomtype = gtd8.roomtypes.roomtype
join gtd8.uses
on gtd8.uses.id = gtd8.facilities.id
join gtd8.activities
on gtd8.activities.ref = gtd8.uses.ref
where gtd8.roomtypes.description like '%touros%'
and gtd8.activities.activity = 'teatro';

```

4.2 *Query* 3.b

How many facilities with ‘touros’ in the room type description are there in each region?

4.2.1 Query

```
db.getCollection('recintos').aggregate(
{
    $match:
    {
        "tipo":
        {
            $regex : "touros"
        }
    }
},
{
    $group:
    {
        _id: "$concelho.regiao.designacao",
        quantidade:
        {
            $sum: 1
        }
    }
}
)
```

4.2.2 Resultado

```
/* 1 */
{
    "_id" : "Algarve",
    "quantidade" : 1.0
}

/* 2 */
{
    "_id" : "Norte",
    "quantidade" : 3.0
}

/* 3 */
{
    "_id" : "Centro",
    "quantidade" : 11.0
}

/* 4 */
{
```

```

    "_id" : "Alentejo",
    "quantidade" : 43.0
  }

/* 5 */
{
    "_id" : "Lisboa",
    "quantidade" : 6.0
}

```

4.2.3 Query em SQL

```

select region, count(*) as nr_facilities
from gtd8.regions
join gtd8.municipalities
on gtd8.regions.cod = gtd8.municipalities.region
join gtd8.facilities
on gtd8.facilities.municipality = gtd8.municipalities.cod
join gtd8.roomtypes
on gtd8.facilities.roomtype = gtd8.roomtypes.roomtype
where gtd8.roomtypes.description like '%touros%'
group by region;

```

4.3 Query 3.c

How many municipalities do not have any facility with an activity of 'cinema'?

4.3.1 Query

```

db.concelhos.count() - db.getCollection('recintos').aggregate(
  {
    $match:
    {
      "atividades":
      {
        $in: ["cinema"]
      }
    }
  },
  {
    $group:
    {
      _id: "$concelho.designacao",
      quantidade:

```

```

        {
            $sum: 1
        }
    }
).toArray().length

```

4.3.2 Resultado

100

4.3.3 *Query* em SQL

```

create or replace view count_municipalities_cinema as
select count(*) as val
from (
select distinct cod
from gtd8.municipalities
join gtd8.facilities
on gtd8.facilities.municipality = gtd8.municipalities.cod
join gtd8.uses
on gtd8.uses.id = gtd8.facilities.id
join gtd8.activities
on gtd8.activities.ref = gtd8.uses.ref
where activity = 'cinema'
order by cod);

```

```

create or replace view count_total as
select count(*) as total
from gtd8.municipalities;

```

```

select total-val as nr
from count_municipalities_cinema, count_total;

```

4.4 *Query* 3.d

Which is the municipality with more facilities engaged in each of the six kinds of activities? Show the activity, the municipality name and the corresponding number of facilities.

4.4.1 Query

```
db.getCollection('recintos').aggregate([
  {
    $unwind: "$atividades"
  },
  {
    $group:
    {
      _id: { concelho: "$concelho.designacao", atividade: "$atividades"},
      quantidade:
      {
        $sum: 1
      }
    }
  },
  {
    $group:
    {
      _id: "$_id.atividade",
      quantidade: { $max: "$quantidade"}
    }
  }
])
```

4.4.2 Resultado

```
/ 1 /
{
  "_id" : "cinema",
  "quantidade" : 96.0
}

/ 2 /
{
  "_id" : "dança",
  "quantidade" : 47.0
}

/ 3 /
{
  "_id" : "circo",
  "quantidade" : 2.0
}
```

```

/ 4 /
{
    "_id" : "música",
    "quantidade" : 77.0
}

/ 5 /
{
    "_id" : "tauromaquia",
    "quantidade" : 4.0
}

/ 6 /
{
    "_id" : "teatro",
    "quantidade" : 66.0
}

```

4.4.3 *Query* em SQL

```

create or replace view count_facilities_municipality as
select cod, activity, count(*) as nr_facilities
from gtd8.municipalities
join gtd8.facilities
on gtd8.facilities.municipality = gtd8.municipalities.cod
join gtd8.uses
on gtd8.uses.id = gtd8.facilities.id
join gtd8.activities
on gtd8.activities.ref = gtd8.uses.ref
group by cod, activity;

```

```

create or replace view max_facilities_activity as
select activity, max(nr_facilities) as nr_facilities
from count_facilities_municipality
group by activity;

```

```

select cod, max_facilities_activity.activity, max_facilities_activity.nr_facilities
from count_facilities_municipality, max_facilities_activity
where max_facilities_activity.nr_facilities = count_facilities_municipality.nr_facilities
and max_facilities_activity.activity = count_facilities_municipality.activity;

```

4.5 Query 3.e

Which are the codes and designations of the districts with facilities in all the municipalities?

4.5.1 Query

```
db.getCollection('concelhos').aggregate([
  {
    $lookup:
    {
      from: 'recintos',
      localField: '_id',
      foreignField: 'concelho._id',
      as: 'recintos'
    }
  }, {
    $group:
    {
      _id: {_id:"$distrito._id", designacao:"$distrito.designacao"},
      concelhos: { $push: { nome: "$designacao", temRecintos: {$gt: [{ $size: "$recintos"
    }
  }, {
    $match:
    {
      "concelhos" : {"$not":{"$elemMatch":{"temRecintos":false}}}
    }
  }
}]
])
```

4.5.2 Resultado

```
/* 1 */
{
  "_id" : {
    "_id" : 12,
    "designacao" : "Portalegre"
  },
  "concelhos" : [
    {
      "nome" : "Alter do Chão",
      "temRecintos" : true
    },
    {
      "nome" : "Arronches",
```

```

        "temRecintos" : true
    },
    {
        "nome" : "Avis",
        "temRecintos" : true
    },
    {
        "nome" : "Campo Maior",
        "temRecintos" : true
    },
    {
        "nome" : "Castelo de Vide",
        "temRecintos" : true
    },
    {
        "nome" : "Crato",
        "temRecintos" : true
    },
    {
        "nome" : "Elvas",
        "temRecintos" : true
    },
    {
        "nome" : "Fronteira",
        "temRecintos" : true
    },
    {
        "nome" : "Gavião",
        "temRecintos" : true
    },
    {
        "nome" : "Marvão",
        "temRecintos" : true
    },
    {
        "nome" : "Monforte",
        "temRecintos" : true
    },
    {
        "nome" : "Nisa",
        "temRecintos" : true
    },
    {
        "nome" : "Ponte de Sor",
        "temRecintos" : true
    },

```

```

        {
            "nome" : "Portalegre",
            "temRecintos" : true
        },
        {
            "nome" : "Sousel",
            "temRecintos" : true
        }
    ]
}

/* 2 */
{
    "_id" : {
        "_id" : 11,
        "designacao" : "Lisboa"
    },
    "concelhos" : [
        {
            "nome" : "Alenquer",
            "temRecintos" : true
        },
        {
            "nome" : "Arruda dos Vinhos",
            "temRecintos" : true
        },
        {
            "nome" : "Azambuja",
            "temRecintos" : true
        },
        {
            "nome" : "Cadaval",
            "temRecintos" : true
        },
        {
            "nome" : "Cascais",
            "temRecintos" : true
        },
        {
            "nome" : "Lisboa",
            "temRecintos" : true
        },
        {
            "nome" : "Loures",
            "temRecintos" : true
        },
    ],
}

```



```

        {
            "nome" : "Lourinhã",
            "temRecintos" : true
        },
        {
            "nome" : "Mafra",
            "temRecintos" : true
        },
        {
            "nome" : "Oeiras",
            "temRecintos" : true
        },
        {
            "nome" : "Sintra",
            "temRecintos" : true
        },
        {
            "nome" : "Sobral de Monte Agraço",
            "temRecintos" : true
        },
        {
            "nome" : "Torres Vedras",
            "temRecintos" : true
        },
        {
            "nome" : "Vila Franca de Xira",
            "temRecintos" : true
        },
        {
            "nome" : "Amadora",
            "temRecintos" : true
        },
        {
            "nome" : "Odivelas",
            "temRecintos" : true
        }
    ]
}

/* 3 */
{
    "_id" : {
        "_id" : 7,
        "designacao" : "Évora"
    },
    "concelhos" : [

```

```

{
  "nome" : "Alandroal",
  "temRecintos" : true
},
{
  "nome" : "Arraiolos",
  "temRecintos" : true
},
{
  "nome" : "Borba",
  "temRecintos" : true
},
{
  "nome" : "Estremoz",
  "temRecintos" : true
},
{
  "nome" : "Évora",
  "temRecintos" : true
},
{
  "nome" : "Mora",
  "temRecintos" : true
},
{
  "nome" : "Mourão",
  "temRecintos" : true
},
{
  "nome" : "Portel",
  "temRecintos" : true
},
{
  "nome" : "Redondo",
  "temRecintos" : true
},
{
  "nome" : "Reguengos de Monsaraz",
  "temRecintos" : true
},
{
  "nome" : "Vendas Novas",
  "temRecintos" : true
},
{
  "nome" : "Montemor-o-Novo",

```

```

        "temRecintos" : true
    },
    {
        "nome" : "Viana do Alentejo",
        "temRecintos" : true
    },
    {
        "nome" : "Vila Viçosa",
        "temRecintos" : true
    }
]
}

/* 4 */
{
    "_id" : {
        "_id" : 15,
        "designacao" : "Setúbal"
    },
    "concelhos" : [
        {
            "nome" : "Alcácer do Sal",
            "temRecintos" : true
        },
        {
            "nome" : "Alcochete",
            "temRecintos" : true
        },
        {
            "nome" : "Almada",
            "temRecintos" : true
        },
        {
            "nome" : "Barreiro",
            "temRecintos" : true
        },
        {
            "nome" : "Grandola",
            "temRecintos" : true
        },
        {
            "nome" : "Moita",
            "temRecintos" : true
        },
        {
            "nome" : "Montijo",

```

```

        "temRecintos" : true
    },
    {
        "nome" : "Palmela",
        "temRecintos" : true
    },
    {
        "nome" : "Seixal",
        "temRecintos" : true
    },
    {
        "nome" : "Sesimbra",
        "temRecintos" : true
    },
    {
        "nome" : "Setúbal",
        "temRecintos" : true
    },
    {
        "nome" : "Sines",
        "temRecintos" : true
    },
    {
        "nome" : "Santiago do Cacém",
        "temRecintos" : true
    }
]
}

```

4.5.3 *Query* em SQL

```

create or replace view nr_municipalities_per_district as
select district, count(*) as nr_municipalities
from gtd8.districts
join gtd8.municipalities
on gtd8.municipalities.district = gtd8.districts.cod
group by district
order by district;

```

```

create or replace view aux as
select distinct district, gtd8.municipalities.cod
from gtd8.districts
join gtd8.municipalities
on gtd8.municipalities.district = gtd8.districts.cod
join gtd8.facilities
on gtd8.facilities.municipality = gtd8.municipalities.cod

```

```
order by district;
```

```
create or replace view nr_mun_with_fac_per_district as
select district, count(*) as nr_municipalities
from aux
group by district;
```

```
select nr_municipalities_per_district.district, designation
from nr_municipalities_per_district, nr_mun_with_fac_per_district, gtd8.districts
where nr_mun_with_fac_per_district.nr_municipalities = nr_municipalities_per_district.nr_mun
      and nr_mun_with_fac_per_district.district = nr_municipalities_per_district.district
and gtd8.districts.cod = nr_municipalities_per_district.district;
```

4.6 Query 3.f

Ask the database a query you think is interesting.

4.6.1 Query

```
db.getCollection('concelhos').find
(
{
"regiao.designacao": "Norte"
}
).count()
```

4.6.2 Resultado

86

4.6.3 Query em SQL

```
select regions.designation, count(*)
from regions
join municipalities
on municipalities.region = regions.cod
where regions.designation = 'Norte'
group by regions.designation;
```

5 Mongo vs Oracle implementations

Nesta secção, vai ser feita uma comparação entre as implementações em Mongo e as implementações em Oracle. Vai ser comparado o tamanho da informação guardada (ou uso do disco), tempo de processamento e facilidade de fazer as queries.

5.1 Tempo de Processamento

Os tempos de processamento das *queries* demonstradas na secção 4 podem ser observados na seguinte tabela:

Query	Tempo Mongo (s)	Tempo Oracle (s)
3.a)	0.004	0.022
3.b)	0.006	0.037
3.c)	0.005	0.404
3.d)	0.007	0.282
3.e)	0.306	0.251
3.f)	0.001	0.2

5.1.1 Conclusão

Pode concluir-se que as *queries* em Mongo são significativamente mais rápidas do que as *queries* implementadas em Oracle.

5.2 Tamanho da informação

Com o uso da seguinte query foi possível ver o tamanho que cada uma das tabelas ocupa em disco:

```
select owner, table_name, round((num_rows*avg_row_len)) B
from all_tables
where num_rows > 0
and owner = 'GTD8';
```

Os resultados podem ser observados na tabela que se segue:

Tabela	Tamanho (B)
FACILITIES	87804
REGIONS	147
DISTRICTS	260
MUNICIPALITIES	10472
ROOMTYPES	378
ACTIVITIES	132
USES	12354
TOTAL	111547

O tamanho das coleções em Mongo é o seguinte:

Coleção	Tamanho (B)
FACILITIES	408280
REGIONS	404
DISTRICTS	1691
MUNICIPALITIES	61164
TOTAL	471539

5.2.1 Conclusão

É possível concluir que os dados em Mongo ocupam mais espaço do que os dados em Oracle.

5.3 Facilidade da *Query*

Relativamente à facilidade de realizar as *queries*, pode observar-se, pelas queries mostradas na secção 4, que as *queries* em SQL usam um número elevado de joins que são desnecessários em NoSQL devido à estruturação dos documentos. As queries em SQL são, portanto, mais extensas e ilegíveis quando comparadas com as queries em NoSQL.