

# NoSQL



Mestrado Integrado em Engenharia Informática e Computação

Tecnologias de Bases de Dados

Ano Letivo 2018/19, 2º Semestre

Francisco Tuna de Andrade - 201503481  
João Paulo Madureira Damas - 201504088

Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

28 de Maio de 2019

# Conteúdo

<b>1</b>	<b>Sumário</b>	<b>4</b>
<b>2</b>	<b>Modelo NoSQL</b>	<b>4</b>
<b>3</b>	<b>Migração de dados</b>	<b>5</b>
<b>4</b>	<b>Perguntas Mongo</b>	<b>9</b>
4.1	Pergunta a . . . . .	9
4.1.1	Interrogação Mongo . . . . .	9
4.1.2	Resultado . . . . .	9
4.2	Pergunta b . . . . .	9
4.2.1	Interrogação Mongo . . . . .	10
4.2.2	Resultado . . . . .	10
4.3	Pergunta c . . . . .	10
4.3.1	Interrogação Mongo . . . . .	10
4.3.2	Resultado . . . . .	11
4.4	Pergunta d . . . . .	11
4.4.1	Interrogação Mongo . . . . .	11
4.4.2	Resultado . . . . .	11
4.5	Pergunta e . . . . .	12
4.5.1	Interrogação Mongo . . . . .	12
4.5.2	Resultado . . . . .	12
4.6	Pergunta f . . . . .	13
4.6.1	Interrogação Mongo . . . . .	13
4.6.2	Resultado . . . . .	13
<b>5</b>	<b>Perguntas Oracle</b>	<b>15</b>
5.1	Pergunta a . . . . .	15
5.1.1	Interrogação Oracle . . . . .	15
5.1.2	Resultado . . . . .	15
5.2	Pergunta b . . . . .	15
5.2.1	Interrogação Oracle . . . . .	15
5.2.2	Resultado . . . . .	16
5.3	Pergunta c . . . . .	16
5.3.1	Interrogação Oracle . . . . .	16
5.3.2	Resultado . . . . .	16
5.4	Pergunta d . . . . .	16
5.4.1	Interrogação Oracle . . . . .	17
5.4.2	Resultado . . . . .	17
5.5	Pergunta e . . . . .	17
5.5.1	Interrogação Oracle . . . . .	17
5.5.2	Resultado . . . . .	18
5.6	Pergunta f . . . . .	18
5.6.1	Interrogação Oracle . . . . .	18
5.6.2	Resultado . . . . .	19

<b>6</b>	<b>MongoDB vs Oracle</b>	<b>19</b>
<b>7</b>	<b>Conclusões</b>	<b>20</b>

# 1 Sumário

No âmbito da cadeira de Tecnologia de Bases de Dados do Mestrado Integrado em Engenharia Informática e Computação, no ano letivo 2018/2019, este trabalho tem como objetivo implementar um modelo numa base de dados com um paradigma não relacional e realizar interrogações de média complexidade sobre o mesmo, de forma a poder comparar com a versão análoga já implementada num modelo relacional em termos de espaço ocupado e facilidade de interrogação. Mais especificamente, será usado o modelo baseado em documentos oferecido pela tecnologia MongoDB. Neste relatório, é inicialmente apresentado um esquema comum que todos os documentos terão de seguir. De seguida, é apresentado o processo de migração da informação da BD Oracle para a BD Mongo. Por fim, são feitas algumas interrogações a ambas as BDs e comparadas as suas execuções, bem como apresentada uma reflexão acerca das vantagens e desvantagens de cada perspetiva.

## 2 Modelo NoSQL

Apesar de uma das vantagens das bases de dados não relacionais ser a possibilidade de ter esquemas flexíveis, é útil definir um conjunto de propriedades e restrições comuns a todos os documentos. Devido à inexistência de operações nativas de *join* (visto que, neste paradigma, geralmente se opta por alguma redundância, em troca de consultas mais rápidas), é comum associações serem implementadas via subdocumentos embedidos no documento original. Uma outra opção é na mesma a criação de várias coleções: uma principal com a informação essencial e coleções secundárias com informações detalhadas sobre entidades específicas. Neste caso em particular, com o esquema adotado, a redundância de informações específicas é mínima e, por isso, não compensaria o *tradeoff* de optar por esta estratégia.

Optou-se, então, por mapear o modelo numa única coleção. Em termos de mapeamento das associações, de forma a decidir a hierarquia dos sub-documentos, duas associações saltam imediatamente à vista: as associações entre Facilities/Roomtypes e Facilities/Uses/Activities. Em ambas, as propriedades que ajudam a descrever o espaço cultural são representadas por chaves estrangeiras na tabela, com a verdadeira informação guardada numa outra tabela. No entanto, esta informação não passa de simples strings, pelo que se torna muito mais vantajoso embeber esta informação, ainda que de forma redundante, no subdocumento de Facilities (desta forma, a tabela Uses cai por completo no mapeamento). Se o contrário se verificasse (guardar os espaços culturais associados a cada tipo de sala e/ou atividade), verificar-se-ia uma redundância bastante mais significativa nos dados e talvez já não representasse uma alternativa tão vantajosa.

A associação entre Facilities/Municipalities representa um caso semelhante: de forma a evitar maior redundância, guardar-se-ão, dentro de cada concelho, os espaços culturais presentes. Já no caso da associação Municipalities/Districts/Regions, a situação requer uma análise mais cuidada. Qualquer uma destas entidades é passível de representar o documento base, embebendo as outras em sub-documentos. Regiões será o candidato mais fraco, visto que, como um distrito não tem de ter necessariamente uma região associada, obrigaria a que os concelhos representassem o segundo nível de profundidade, formando uma hierarquia pouco intuitiva região ->concelho ->distrito. Por outro lado, utilizar os concelhos como documento base obriga a redundância de informação, quer ao nível da região, quer ao nível do distrito. Desta forma, optou-se pela terceira opção, utilizando os distritos como o nível mais exterior dos documentos. Desta forma,

a única redundância presente é ao nível da região nos concelhos (que não terá possibilidades de crescer futuramente, visto que representam algo estável). Apresentando o melhor balanço entre redundância e proximidade com a intuição de entre as opções, aparenta ser a melhor escolha para modelar. A coleção terá, assim, 20 documentos (20 distritos) com uma estrutura semelhante a esta:

```
{
  "_id" : 4,
  "designation" : "Bragança",
  "municipalities" : [
    {
      "cod" : 409,
      "designation" : "Torre de Moncorvo",
      "region" : {
        "cod" : 1,
        "designation" : "Norte",
        "nut1" : "Continente"
      },
      "facilities" : [
        {
          "id" : 242,
          "name" : "CINE TEATRO DE TORRE DE MONCORVO",
          "capacity" : 293,
          "roomtype" : "Cine-Teatro",
          "address" : "AVENIDA ENGENHEIRO DUARTE PACHECO",
          "activities" : [
            "cinema",
            "dança",
            "música",
            "teatro"
          ]
        }
      ]
    }
  ],
  ...
}
```

De notar que a informação da região não está (diretamente) presente no distrito, contrariamente à versão relacional. Realmente, essa informação é derivada da informação da região em cada um dos concelhos do distrito: caso seja igual em todos, será essa a região do distrito, caso contrário, não terá região associada. Assim, a informação continua toda presente, mas consegue-se encurtar os documentos sem comprometer nada.

No caso de (sub)documentos não possuírem certos atributos, nomeadamente um concelho não ter espaços culturais associados, os campos respetivos podem ser facilmente omitidos sem comprometer a intuição do esquema ou a integridade da informação.

### 3 Migração de dados

De forma a migrar os dados para a base de dados NoSQL, recorreu-se ao seguinte método:

## 1. Criar um *package* PL/SQL que exporta os dados, em formato XML, para uma tabela

Primeiramente, definiu-se uma estrutura XML que fosse facilmente processada mais tarde para inserção na BD NoSQL. Desta forma, definiu-se o procedimento *export\_db* que constroi o XML, recorrendo a funções *package private* auxiliares (e.g., para cada distrito, obter todos os concelhos, para todos os concelhos, obter todos os espaços culturais) e guarda o seu valor num CLOB. No fim, insere essa informação na tabela *db\_export*, cujas linhas contêm apenas uma coluna do mesmo tipo.

```
create or replace package export_cultural_facilities as
procedure export_db;
end export_cultural_facilities;

create or replace package body export_cultural_facilities as
function get_activities(facility_id gtd8.facilities.id%type) return clob is
v_clob clob := '';
begin
for activity in (select a.activity as a_name from gtd8.uses u inner join gtd8.activities a on
↳ u.ref = a.ref where u.id = facility_id) loop
v_clob := v_clob || to_clob('<ACTIVITIES>' || activity.a_name || '</ACTIVITIES>');
end loop;
return v_clob;
end get_activities;

function get_facilities(municipality_cod gtd8.municipalities.cod%type) return clob is
v_clob clob := '';
begin
for facility in (select f.id as f_id, f.name as f_name, f.capacity as f_cap, r.description as
↳ r_des, f.address as f_addr from gtd8.facilities f inner join gtd8.roomtypes r on
↳ f.roomtype = r.roomtype where f.municipality = municipality_cod) loop
v_clob := v_clob || to_clob('<FACILITIES>');
v_clob := v_clob || to_clob('<ID>' || facility.f_id || '</ID>');
v_clob := v_clob || to_clob('<NAME>' || facility.f_name || '</NAME>');
v_clob := v_clob || to_clob('<CAPACITY>' || facility.f_cap || '</CAPACITY>');
v_clob := v_clob || to_clob('<ROOMTYPE>' || facility.r_des || '</ROOMTYPE>');
v_clob := v_clob || to_clob('<ADDRESS>' || facility.f_addr || '</ADDRESS>');
v_clob := v_clob || get_activities(facility.f_id);
v_clob := v_clob || to_clob('</FACILITIES>');
end loop;
return v_clob;
end get_facilities;

function get_municipalities(district_cod gtd8.districts.cod%type) return clob is
v_clob clob := '';
begin
for municipality in (select m.cod as m_cod, m.designation as m_des, r.cod as r_cod,
↳ r.designation as r_des, r.nut1 as r_nut1 from gtd8.municipalities m inner join
↳ gtd8.regions r on m.region = r.cod where m.district = district_cod) loop
v_clob := v_clob || to_clob('<MUNICIPALITIES>');
v_clob := v_clob || to_clob('<COD>' || municipality.m_cod || '</COD>');
v_clob := v_clob || to_clob('<DESIGNATION>' || municipality.m_des || '</DESIGNATION>');
v_clob := v_clob || to_clob('<REGION>');
v_clob := v_clob || to_clob('<COD>' || municipality.r_cod || '</COD>');
v_clob := v_clob || to_clob('<DESIGNATION>' || municipality.r_des || '</DESIGNATION>');
```

```

v_clob := v_clob || to_clob('<NUT1>' || municipality.r_nut1 || '</NUT1>');
v_clob := v_clob || to_clob('</REGION>');
v_clob := v_clob || get_facilities(municipality.m_cod);
v_clob := v_clob || to_clob('</MUNICIPALITIES>');
end loop;
return v_clob;
end get_municipalities;

procedure export_db is
v_clob clob := '';
v_offset number default 1;
v_chunk_size number := 3500;
begin
v_clob := v_clob || to_clob('<?xml version="1.0" encoding="iso-8859-1" ?>');
v_clob := v_clob || to_clob('<DATA>');
for district in (select d.cod as d_cod, d.designation as d_des from gtd8.districts d) loop
v_clob := v_clob || to_clob('<DISTRICTS>');
v_clob := v_clob || to_clob('<_ID>' || district.d_cod || '</_ID>');
v_clob := v_clob || to_clob('<DESIGNATION>' || district.d_des || '</DESIGNATION>');
v_clob := v_clob || get_municipalities(district.d_cod);
v_clob := v_clob || to_clob('</DISTRICTS>');
end loop;
v_clob := v_clob || to_clob('</DATA>');
loop
    exit when v_offset > dbms_lob.getlength(v_clob);
    insert into db_export values (to_clob(dbms_lob.substr(v_clob, v_chunk_size, v_offset)));
    v_offset := v_offset + v_chunk_size;
end loop;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line(SQLERRM);
end export_db;
end export_cultural_facilities;

```

De notar a decomposição do CLOB em *chunks* de 3500 carateres em vez da inserção completa duma só vez na tabela. Isto deve-se ao facto da operação de exportação não suportar a exportação direta de objetos tão grandes (limitado ao tamanho máximo de *strings*, 4000 carateres). Apesar do limite de 4000, a decomposição foi em segmentos de 3500 carateres, visto que nalguns segmentos existem *multi-byte characters*.

## 2. Exportar a tabela para um ficheiro

Para exportar a informação, utilizou-se o utilitário do SQL developer para exportar uma tabela, exportando apenas os dados, em formato de texto, como visível na figura abaixo.

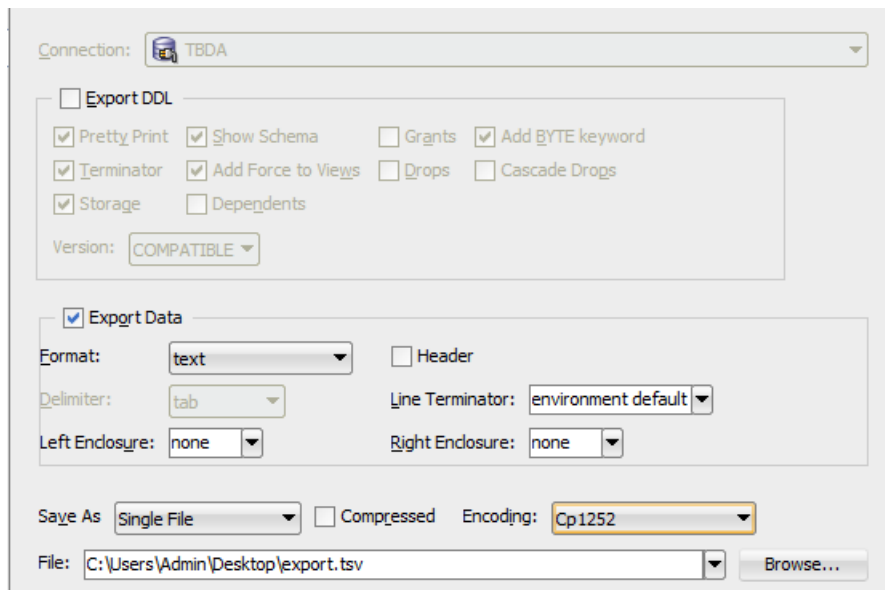


Figura 1: Exportação dos dados do SQL Developer

### 3. Fornecer o ficheiro como input a um *script* de Python que povoa a BD NoSQL

Por fim, através de um *script* em Python, fazendo uso das ferramentas *xmldict* e *pymongo*, o XML presente no ficheiro é convertido em formato de documento e importado para a BD NoSQL.

```
import xmldict
import pymongo

# Parses key to lowercase strings (more conventional for mongo) and parses int values as int
↳ (when applicable)
def postprocessor(path, key, value):
    try:
        return key.lower(), int(value)
    except (ValueError, TypeError):
        return key.lower(), value

XML_DATA = ''.join([line.rstrip('\n\t') for line in open('export.tsv')])
my_dict = xmldict.parse(XML_DATA, postprocessor=postprocessor)

client =
↳ pymongo.MongoClient("mongodb://tbda:grupoa@vdbase.inesctec.pt:27017/tbda?authSource=admin")
db = client["tbda"]
coll = db["districts"]
try:
    coll.insert_many(my_dict['data']['districts'])
except pymongo.errors.BulkWriteError as e:
    print(e.details)
```



## 4 Perguntas Mongo

### 4.1 Pergunta a

*Quais os espaços culturais em que a descrição do tipo de sala contém 'touros' e tem 'teatro' como uma das suas atividades? Mostre o id, nome, descrição e atividade.*

#### 4.1.1 Interrogação Mongo

```
db.districts.aggregate([
  {$unwind: "$municipalities"},
  {$unwind: "$municipalities.facilities"},
  {$match: {"municipalities.facilities.roomtype": /touros/,
    ↳ "municipalities.facilities.activities": "teatro"}},
  {$project: {_id: 0, "Facility ID": "$municipalities.facilities.id", "Facility Name":
    ↳ "$municipalities.facilities.name", "Room Type Description":
    ↳ "$municipalities.facilities.roomtype", "Activity": "teatro"}},
  {$sort: {"Facility ID":1}}
]);
```

#### 4.1.2 Resultado

```
[
  {
    "Facility ID" : 916,
    "Facility Name" : "COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS",
    "Room Type Description" : "Praça de touros multiusos",
    "Activity" : "teatro"
  },
  {
    "Facility ID" : 940,
    "Facility Name" : "ARENA DE ÉVORA - EX PRAÇA DE TOIROS",
    "Room Type Description" : "Praça de touros multiusos",
    "Activity" : "teatro"
  },
  {
    "Facility ID" : 957,
    "Facility Name" : "COLISEU DE REDONDO - EX PRAÇA DE TOIROS",
    "Room Type Description" : "Praça de touros multiusos",
    "Activity" : "teatro"
  }
]
```

### 4.2 Pergunta b

*Quantos espaços culturais com 'touros' na descrição do tipo de sala existem em cada região?*

### 4.2.1 Interrogação Mongo

```
db.districts.aggregate([
  {$unwind: "$municipalities"},
  {$unwind: "$municipalities.facilities"},
  {$match: {"municipalities.facilities.roomtype": /touro/}},
  {$group: {_id: "$municipalities.region.designation", "NoFacilities": {$sum: 1}}},
  {$project: {"_id": 0, "Region": "$_id", "NoFacilities": "$NoFacilities"}},
  {$sort: {"Region": 1}}
]);
```

### 4.2.2 Resultado

```
[
  {
    "Region" : "Alentejo",
    "NoFacilities" : 43.0
  },
  {
    "Region" : "Algarve",
    "NoFacilities" : 1.0
  },
  {
    "Region" : "Centro",
    "NoFacilities" : 11.0
  },
  {
    "Region" : "Lisboa",
    "NoFacilities" : 6.0
  },
  {
    "Region" : "Norte",
    "NoFacilities" : 3.0
  }
]
```

## 4.3 Pergunta c

*Quantos concelhos não têm nenhum espaço cultural com uma atividade 'cinema'?*

### 4.3.1 Interrogação Mongo

```
db.districts.aggregate([
  {$unwind: "$municipalities"},
  {$match: {"municipalities.facilities.activities": {$nin: ["cinema"]}}},
  {$count: "NoMunicipalities"}
]);
```

### 4.3.2 Resultado

```
{
  "NoMunicipalities" : 100
}
```

## 4.4 Pergunta d

*Qual o concelho com mais espaços culturais envolvidos em cada um dos seis tipos de atividade? Mostre a atividade, o nome do concelho e o número correspondente de espaços culturais.*

### 4.4.1 Interrogação Mongo

```
db.districts.aggregate([
  {$unwind: "$municipalities"},
  {$unwind: "$municipalities.facilities"},
  {$unwind: "$municipalities.facilities.activities"},
  {$group: {_id: {"activity": "$municipalities.facilities.activities", "municipality":
    ↳ "$municipalities.designation"}, count: {$sum: 1}}},
  {$sort: {"count": -1}},
  {$group: {_id: "$_id.activity",
    "municipalities":{
      $push: {"municipality": "$_id.municipality", "count": "$count"}
    }
  }},
  {$project: {
    "_id": 0,
    "Activity": "$_id" ,
    "Municipality": {$arrayElemAt: ["$municipalities.municipality", 0]},
    "NoFacilities": {$arrayElemAt: ["$municipalities.count", 0]}
  }},
  {$sort: {"Activity": 1}}
]);
```

### 4.4.2 Resultado

```
[
  {
    "Activity" : "cinema",
    "Municipality" : "Lisboa",
    "NoFacilities" : 96.0
  },
  {
    "Activity" : "circo",
```

```

    "Municipality" : "Lisboa",
    "NoFacilities" : 2.0
  },
  {
    "Activity" : "dança",
    "Municipality" : "Lisboa",
    "NoFacilities" : 47.0
  },
  {
    "Activity" : "música",
    "Municipality" : "Lisboa",
    "NoFacilities" : 77.0
  },
  {
    "Activity" : "tauromaquia",
    "Municipality" : "Moura",
    "NoFacilities" : 4.0
  },
  {
    "Activity" : "teatro",
    "Municipality" : "Lisboa",
    "NoFacilities" : 66.0
  }
]

```

## 4.5 Pergunta e

*Qual o código e designação dos distritos com espaços culturais em todos os seus concelhos?*

### 4.5.1 Interrogação Mongo

```

db.districts.aggregate([
  {$match: {"municipalities.facilities": {$ne: null}}},
  {$project: {_id:0, "Code": "$_id", "Designation": "$designation"}},
  {$sort: {"Designation": 1}}
]);

```

### 4.5.2 Resultado

```

[
  {
    "Code" : 11,
    "Designation" : "Lisboa"
  },
  {
    "Code" : 12,
    "Designation" : "Portalegre"
  },
]

```

```

{
  "Code" : 15,
  "Designation" : "Setúbal"
},
{
  "Code" : 7,
  "Designation" : "Évora"
}
]

```

## 4.6 Pergunta f

*Para cada distrito, indique a percentagem de municípios onde existem espaços culturais com a atividade de cinema.*

### 4.6.1 Interrogação Mongo

```

db.districts.aggregate([
  {$addFields: {"size_mun": {$size: "$municipalities"}}},
  {$unwind: "$municipalities"},
  {$match: {"municipalities.facilities.activities": {$in: ["cinema"]}}},
  {$group: {_id: {"designation": "$designation", "size_mun": "$size_mun"}, count: {$sum: 1}}},
  {$project: {_id: 0, "Distrito": "$_id.designation", "Percentage": {$divide: [{$multiply:
    ↪ ["$count", 100]}, "$_id.size_mun"] }}},
  {$project: {"Distrito": 1,
    "Percentage":
      {$subtract: [
        {$add: ['$Percentage', 0.049999999999999999]},
        {$mod: [{$add: ['$Percentage', .049999999999999999]}, 0.1]]}}
      }
    },
  {$sort: {"Distrito": 1}}
]);

```

### 4.6.2 Resultado

```

[
  {
    "Distrito" : "Aveiro",
    "Percentage" : 89.5
  },
  {
    "Distrito" : "Beja",
    "Percentage" : 78.6
  },
  {
    "Distrito" : "Braga",

```

```

    "Percentage" : 57.1
  },
  {
    "Distrito" : "Bragança",
    "Percentage" : 66.7
  },
  {
    "Distrito" : "Castelo Branco",
    "Percentage" : 63.6
  },
  {
    "Distrito" : "Coimbra",
    "Percentage" : 58.8
  },
  {
    "Distrito" : "Faro",
    "Percentage" : 75.0
  },
  {
    "Distrito" : "Guarda",
    "Percentage" : 85.7
  },
  {
    "Distrito" : "Leiria",
    "Percentage" : 75.0
  },
  {
    "Distrito" : "Lisboa",
    "Percentage" : 100.0
  },
  {
    "Distrito" : "Portalegre",
    "Percentage" : 73.3
  },
  {
    "Distrito" : "Porto",
    "Percentage" : 77.8
  },
  {
    "Distrito" : "Santarém",
    "Percentage" : 71.4
  },
  {
    "Distrito" : "Setúbal",
    "Percentage" : 100.0
  },
  {
    "Distrito" : "Viana do Castelo",
    "Percentage" : 60.0
  },
  {
    "Distrito" : "Vila Real",
    "Percentage" : 50.0
  },
  {
    "Distrito" : "Viseu",

```

```

    "Percentage" : 66.7
  },
  {
    "Distrito" : "Évora",
    "Percentage" : 92.9
  }
]

```

## 5 Perguntas Oracle

### 5.1 Pergunta a

*Quais os espaços culturais em que a descrição do tipo de sala contém 'touros' e tem 'teatro' como uma das suas atividades? Mostre o id, nome, descrição e atividade.*

#### 5.1.1 Interrogação Oracle

```

SELECT f.id AS "Facility ID", f.name AS "Facility Name", r.description AS "Room Type
↪ Description", a.activity AS "Activity"
FROM GTD8.facilities f JOIN GTD8.roomtypes r ON f.roomtype = r.roomtype
JOIN GTD8.uses u ON f.id = u.id JOIN GTD8.activities a ON u.ref = a.ref
WHERE r.description LIKE '%touros%' AND a.activity = 'teatro'
ORDER BY f.id;

```

#### 5.1.2 Resultado

	Facility ID	Facility Name	Room Type Description	Activity
1	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
2	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
3	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro

Figura 2: Resultado da aléna a)

### 5.2 Pergunta b

*Quantos espaços culturais com 'touros' na descrição do tipo de sala existem em cada região?*

#### 5.2.1 Interrogação Oracle

```

SELECT reg.designation AS "Region", COUNT(DISTINCT f.id) AS "No. Facilities"
FROM GTD8.facilities f JOIN GTD8.roomtypes r ON f.roomtype = r.roomtype
JOIN GTD8.uses u ON f.id = u.id JOIN GTD8.activities a ON u.ref = a.ref
JOIN GTD8.municipalities m ON f.municipality = m.cod JOIN GTD8.regions reg ON m.region =
↪ reg.cod

```

```
WHERE r.description LIKE '%touros%'
GROUP BY reg.designation
ORDER BY reg.designation;
```

## 5.2.2 Resultado

Region	No. Facilities
1 Alentejo	43
2 Algarve	1
3 Centro	11
4 Lisboa	6
5 Norte	3

Figura 3: Resultado da alínea b)

## 5.3 Pergunta c

*Quantos concelhos não têm nenhum espaço cultural com uma atividade 'cinema'?*

### 5.3.1 Interrogação Oracle

```
SELECT COUNT(*) AS "No. Municipalities" FROM GTD8.municipalities m
WHERE NOT EXISTS (
  SELECT * FROM GTD8.facilities f JOIN GTD8.uses u ON f.id = u.id
  JOIN GTD8.activities a ON u.ref = a.ref WHERE m.cod = f.municipality AND a.activity =
    ↪ 'cinema'
);
```

## 5.3.2 Resultado

No. Municipalities
1 100

Figura 4: Resultado da alínea c)

## 5.4 Pergunta d

*Qual o concelho com mais espaços culturais envolvidos em cada um dos seis tipos de atividade? Mostre a atividade, o nome do concelho e o número correspondente de espaços culturais.*



### 5.4.1 Interrogação Oracle

```
WITH act_facilities AS (  
  SELECT a.activity AS activity, m.designation AS municipality, COUNT(DISTINCT f.id) AS  
    ↪ nr_facilities  
  FROM GTD8.facilities f JOIN GTD8.roomtypes r ON f.roomtype = r.roomtype  
  JOIN GTD8.uses u ON f.id = u.id JOIN GTD8.activities a ON u.ref = a.ref  
  JOIN GTD8.municipalities m ON f.municipality = m.cod  
  GROUP BY a.activity, m.designation )  
SELECT aux.activity AS "Activity", aux.municipality AS "Municipality", aux.nr_facilities As  
  ↪ "No. Facilities"  
FROM act_facilities aux  
WHERE NOT EXISTS  
  (SELECT * FROM act_facilities aux2  
   WHERE aux2.activity = aux.activity AND aux2.nr_facilities > aux.nr_facilities)  
ORDER BY aux.activity;
```

### 5.4.2 Resultado

Activity	Municipality	No. Facilities
1 cinema	Lisboa	96
2 circo	Lisboa	2
3 dança	Lisboa	47
4 música	Lisboa	77
5 tauromaquia	Moura	4
6 teatro	Lisboa	66

Figura 5: Resultado da alínea d)

## 5.5 Pergunta e

*Qual o código e designação dos distritos com espaços culturais em todos os seus concelhos?*

### 5.5.1 Interrogação Oracle

```
SELECT d.cod AS "Code", d.designacao AS "Designation"  
FROM GTD8.distritos d  
WHERE NOT EXISTS (  
  SELECT * FROM GTD8.municipalities m WHERE d.cod = m.district AND NOT EXISTS (  
    SELECT * FROM GTD8.facilities f WHERE m.cod = f.municipality  
  )  
)  
ORDER BY d.cod;
```

### 5.5.2 Resultado

Code	Designation
1	7 Évora
2	11 Lisboa
3	12 Portalegre
4	15 Setúbal

Figura 6: Resultado da alínea e)

## 5.6 Pergunta f

*Para cada distrito, indique a percentagem de municípios onde existem espaços culturais com a atividade de cinema.*

### 5.6.1 Interrogação Oracle

```
WITH dist_cinema AS (  
  SELECT d.cod AS dist, COUNT(DISTINCT m.cod) AS count  
  FROM GTD8.activities a, GTD8.uses u, GTD8.facilities f, GTD8.municipalities m, GTD8.distritos d  
  WHERE a.activity = 'cinema' AND a.ref = u.ref AND u.id = f.id AND f.municipality = m.cod AND  
    ↪ m.district = d.cod  
  GROUP BY d.cod),  
dist_mun AS (  
  SELECT d.cod AS dist, COUNT(m.cod) AS count  
  FROM GTD8.municipalities m, GTD8.distritos d  
  WHERE m.district = d.cod  
  GROUP BY d.cod  
)  
SELECT d.designacao AS "Distrito", round(100*(aux1.count/aux2.count), 1) AS "Percentage"  
FROM dist_cinema aux1, dist_mun aux2, GTD8.distritos d  
WHERE aux1.dist = aux2.dist AND aux1.dist = d.cod  
ORDER BY d.designacao;
```

### 5.6.2 Resultado

Distrito	Percentage
1 Aveiro	89.5
2 Beja	78.6
3 Braga	57.1
4 Bragança	66.7
5 Castelo Branco	63.6
6 Coimbra	58.8
7 Faro	75
8 Guarda	85.7
9 Leiria	75
10 Lisboa	100
11 Portalegre	73.3
12 Porto	77.8
13 Santarém	71.4
14 Setúbal	100
15 Viana do Castelo	60
16 Vila Real	50
17 Viseu	66.7
18 Évora	92.9

Figura 7: Resultado da alínea f)

## 6 MongoDB vs Oracle

No que diz respeito aos tempos de execução, podemos constatar que as queries em *NoSQL* são processadas mais rapidamente do que as queries na base de dados relacional. A tabela abaixo mostra a discriminação de cada um destes tempos para cada uma das queries:

Query	Mongo (s)	Oracle (s)
a)	0.01	0.012
b)	0.009	0.011
c)	0.007	0.008
d)	0.012	0.017
e)	0.003	0.007
f)	0.009	0.018

Tabela 1: Tempos de execução das queries em Mongo e Oracle

Em certas queries, foi possível alcançar em *Mongo* uma diminuição de mais de 50% do tempo de execução comparativamente com *Oracle*, pelo que em sistemas onde a execução temporal é uma componente crítica, as bases de dados *NoSQL* devem ser consideradas. De realçar que em nenhum dos modelos (relacional e não relacional) faz uso de índices.

Quanto à facilidade de escrita das queries, verificámos que para queries mais simples como a **a)** ou a **c)**, se revelou mais ingénuo escrever as queries em *SQL*. Por outro lado, para queries

mais complexas como a **d)** ou a **f)**, usar *Mongo* revelou-se menos complexo. Em particular, na query **f)**, foi possível, através do operador *\$addFields*, fazer, numa linha, o trabalho, para o qual, em *SQL*, necessitámos de criar uma tabela auxiliar.

Já quanto ao tamanho ocupado pelos dados, não é possível identificar sempre um vencedor claro, visto que a flexibilidade do esquema dos documentos no modelo não relacional traz bastante variabilidade ao espaço ocupado por uma base de dados deste tipo. Tipicamente, com a introdução de redundância, é esperado que o modelo não relacional seja mais volumoso que o seu análogo relacional, constituindo um dos *trade-offs* deste paradigma. No entanto, para o caso em concreto, o modelo não relacional definido contém o mínimo de redundância possível (ao nível de atividades, tipos de sala e regiões) e, por conseguinte, não deverá ocupar significativamente mais espaço.

## 7 Conclusões

Com a introdução do paradigma não relacional, surge uma alternativa que resolve bastantes problemas decorrentes do uso de bases de dados relacionais, nomeadamente uma solução distribuída, capaz de lidar com largos volumes de informações, de rápida consulta e com esquemas flexíveis e adaptáveis aos requerimentos hoje em dia em constante mudança, permitindo um início e iterações consequentes de desenvolvimento bastante mais rápidas. Tudo isto não vem sem custo, já que frequentemente a rápida consulta resulta da introdução de redundância no modelo a fim de evitar operações de junção. A flexibilidade de esquema traz a necessidade de um forte suporte para interrogações, o que foi de certa forma corroborado com este trabalho. No entanto, a heterogeneidade das mesmas é um aspeto menos positivo (não existe uma estrutura bem definida para interrogar como em *SQL*). Por outro lado, vários motivos não permitiram aproveitar as vantagens do paradigma não relacional: o reduzido volume de dados e a estrutura relativamente consistente dos mesmos acabam por mascarar o propósito de adotar esta estratégia. Assim, reconhece-se o potencial deste paradigma numa era de grande volume e constante diversidade de informação, no entanto, para casos de dimensão reduzida, como é o caso, é necessário uma análise mais cuidada para verificar se, de facto, as vantagens compensam o que se cede ao sair do paradigma relacional.