

Tecnologia de Bases de Dados

Mestrado Integrado em
Engenharia Informática e Computação



NOSQL ASSIGNMENT

Trabalho realizado por:
Fabiola Figueira da Silva - UP201502850
Pedro Filipe Agrela Faria - UP201406992
Pedro Manuel Monteiro Albano - UP201008982

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

5 de Junho de 2018

1 Conteúdo

2 Objetivos do trabalho	2
3 Modelo do documento em Mongo	2
4 Criação/Exportação do XML e conversão para MongoDB	2
5 Execução das Queries	2
5.1 Query (a)	3
5.1.1 SQL	3
5.1.2 NoSQL (MongoDB)	3
5.1.3 Resultados nos ambientes SQL e NoSQL	4
5.2 Query (b)	4
5.2.1 SQL	5
5.2.2 NoSQL (MongoDB)	6
5.1.3 Resultados nos ambientes SQL e NoSQL	7
5.3 Query (c)	7
5.3.1 SQL	7
5.3.2 NoSQL (MongoDB)	8
5.3.3 Resultados nos ambientes SQL e NoSQL	8
5.4 Query (d)	9
5.4.1 SQL	9
5.4.2 NoSQL (MongoDB)	10
5.4.3 Resultados nos ambientes SQL e NoSQL	10
5.5 Query (e)	11
5.5.1 SQL	11
5.5.2 NoSQL (MongoDB)	12
5.5.3 Resultados nos ambientes SQL e NoSQL	13
5.6 Query (f)	13
5.6.1 SQL	14
5.6.2 NoSQL (MongoDB)	15
5.6.3 Resultados nos ambientes SQL e NoSQL	15
6 Comparações do Mongo e Oracle	16
7 Anexos	16
7.1 Anexo 1 - Estrutura do modelo do documento em Mongo.	16
7.2 Anexo 2 - Código em SQL usado para criação das vistas	17
7.3 Anexo 3 - Código em JS usado para inserir os dados no MongoDB	19

2 Objetivos do trabalho

O objetivo principal deste trabalho é o desenvolvimento de uma base de dados em NoSQL, dando uso das capacidades da base de dados em MongoDB, comparando depois com a convencional base de dados em SQL. Um outro objetivo deste trabalho é popular a nossa base de dados com os dados existentes na base de dados relacional fornecida pelo docente, disponíveis em GTD10.

3 Modelo do documento em Mongo

O grupo decidiu criar uma tabela apenas em Mongo, usando o princípio do menos abrangente para o mais abrangente. A estrutura usada foi a seguinte:

```
XOCORRENCIAS  
XUCS  
[  
  XDOCENTES  
  [  
    XDSD  
    XTIPOSAULA  
  ]  
]
```

Esta abordagem permite-nos ter os dados todos numa mesma tabela, no qual obtemos as ocorrências e ucs em primeiro nível, em segundo nível um vetor de docentes presentes na uc dessa ocorrência, já em terceiro nível, obtemos os tipos de aula e horários do docente. O anexo 1 representa as variáveis e tipos usadas, como a estrutura exata do modelo usada.

4 Criação/Exportação do XML e conversão para MongoDB

Começamos por criar vistas no Oracle SQL para a base de dados original, o código em anexo 2, ilustra o código executado na criação destas vistas. Através do Oracle SQL criamos o XML que contém todos os dados provenientes de GTD10. Para converter para MongoDB usamos um script em NodeJS, o código em anexo 3, ilustra o código usado para esta conversão.

5 Execução das Queries

Descrição, código, resultado e comparação dos tempos de resposta entre os ambientes SQL e NoSQL (MongoDB)

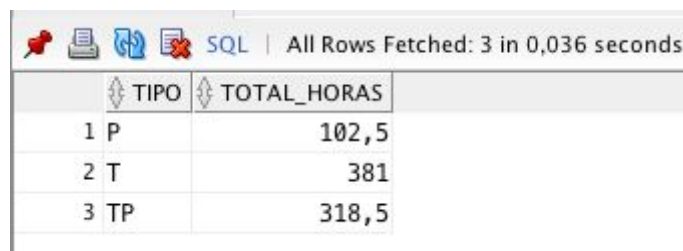
5.1 Query (a)

" How many class hours of each type did the program 233 got in year 2004/2005? "

5.1.1 SQL

```
SELECT gtd10.xtiposaula.tipo, sum(gtd10.xtiposaula.n_aulas * gtd10.xtiposaula.horas_turno) as total_horas
FROM gtd10.xtiposaula
JOIN gtd10.xucs on gtd10.xucs.codigo = gtd10.xtiposaula.codigo
WHERE gtd10.xtiposaula.ano_letivo = '2004/2005' and gtd10.xucs.curso = 233
GROUP BY gtd10.xtiposaula.tipo;
```

Figura 1 - SQL query (a)



	TIPO	TOTAL_HORAS
1	P	102,5
2	T	381
3	TP	318,5

Figura 2 - Resultado da SQL query (a)

5.1.2 NoSQL (MongoDB)

```
db.occurencias.aggregate([
  { $match: { "uc.curso":233 , "ano_lectivo":"2004/2005"}},
  { "$unwind": "$docentes"},
  { "$unwind": "$docentes aulas"},
  { $project: { _id: { ocorrencia: "$codigo", tipo: "$docentes aulas.tipo", turnos: "$docentes aulas.turnos",
    n_aulas: "$docentes aulas.n_aulas", horas_turno: "$docentes aulas.horas_turno" } }},
  { $group: { _id: "$_id", total: { $first: { $multiply: ["$_id.n_aulas", "$_id.horas_turno"] } } }},
  { $group: { _id: "$_id.tipo", total: { $sum: "$total" } }},
]);
```

Figura 3 - NoSQL Query (a)

```
1 {  
2   "_id" : "TP",  
3   "total" : 299.5  
4 }  
5 {  
6   "_id" : "P",  
7   "total" : 94.5  
8 }  
9 {  
10  "_id" : "T",  
11  "total" : NumberInt(361)  
12 }  
13
```

0 documents selected 0.013s

Figura 4 - Resultado da NoSQL query (a)

5.1.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.036	0.013

Tabela 1 - Resultados obtidos das queries em cada ambiente

5.2 Query (b)

" Which courses (show the code, total class hours required, total classes

assigned) have a difference between total class hours required and the service actually assigned in year 2003/2004? ”

5.2.1 SQL

```
with valores as(SELECT gtd10.xocorrencias.codigo, gtd10.xocorrencias.periodo, sum(gtd10.xdsd.horas) as atribuidas,
sum(gtd10.xtiposaula.turnos * gtd10.xtiposaula.horas_turno) as necessarias
FROM gtd10.xocorrencias
JOIN gtd10.xtiposaula on (gtd10.xtiposaula.ano_letivo = gtd10.xocorrencias.ano_letivo and
gtd10.xtiposaula.periodo = gtd10.xocorrencias.periodo and
gtd10.xtiposaula.codigo = gtd10.xocorrencias.codigo)
JOIN gtd10.xdsd on gtd10.xdsd.id = gtd10.xtiposaula.id
WHERE gtd10.xocorrencias.ano_letivo = '2003/2004'
GROUP BY gtd10.xocorrencias.codigo, gtd10.xocorrencias.periodo
ORDER BY gtd10.xocorrencias.codigo
), diferentes as(
SELECT atribuidas, necessarias from valores where atribuidas != necessarias
)
SELECT DISTINCT valores.codigo, valores.periodo, valores.atribuidas, valores.necessarias
FROM valores
INNER JOIN diferentes ON valores.atribuidas = diferentes.atribuidas and valores.necessarias = diferentes.necessarias
ORDER BY valores.codigo;
```

Figura 5 - SQL query (b)

SQL Fetched 50 rows in 0,044 seconds				
	CODIGO	PERIODO	ATRIBUIDAS	NECESSARIAS
1	CI028	2S	4	6
2	EC1101	1S	32	80
3	EC1103	1S	28	50
4	EC1107	1S	48	180
5	EC1108	1S	60	404
6	EC1207	2S	30	102
7	EC1209	2S	28	100
8	EC1211	2S	26	52
9	EC1214	2S	20	40
10	EC2102	1S	50	320
11	EC2103	1S	24	68
12	EC2104	1S	26	72
13	EC2106	2S	28	52
14	EC2107	1S	20	40
15	EC2108	2S	34	124
16	EC2208	2S	46	252
17	EC2209	2S	36	126
18	EC2211	1S	36	132

Figura 6 - Resultado da SQL query (b)

5.2.2 NoSQL (MongoDB)

```
db.occurencias.aggregate([
  { $match: { "ano_lectivo": "2003/2004" } },
  { $unwind: "$docentes" },
  { $unwind: "$docentes.aulas" },
  { $project: { _id: { codigo: "$codigo", periodo: "$periodo" }, aulas: "$docentes.aulas" } },
  { $group: { _id: { codigo: "$_id", t_aula: "$aulas.tipo" }, sum_dsd: { $sum: "$aulas.dsd.horas" },
    sum_horas: { $max: { $multiply: [ "$aulas.horas_turno", "$aulas.turnos" ] } } } },
  { $group: { _id: "$_id.codigo", atribuidas: { $sum: "$sum_dsd" }, necessarias: { $sum: "$sum_horas" } } },
  { $project: { _id: "$_id", atribuidas: "$atribuidas", necessarias: "$necessarias", diferente: { $cmp: [ "$necessarias", "$atribuidas" ] } } },
  { $match: { diferente: { $ne: 0 } } },
  { $project: { _id: "$_id", atribuidas: "$atribuidas", necessarias: "$necessarias" } },
  { $sort: { "_id" : 1 } }
]);
```

Figura 7 - NoSQL query (b)

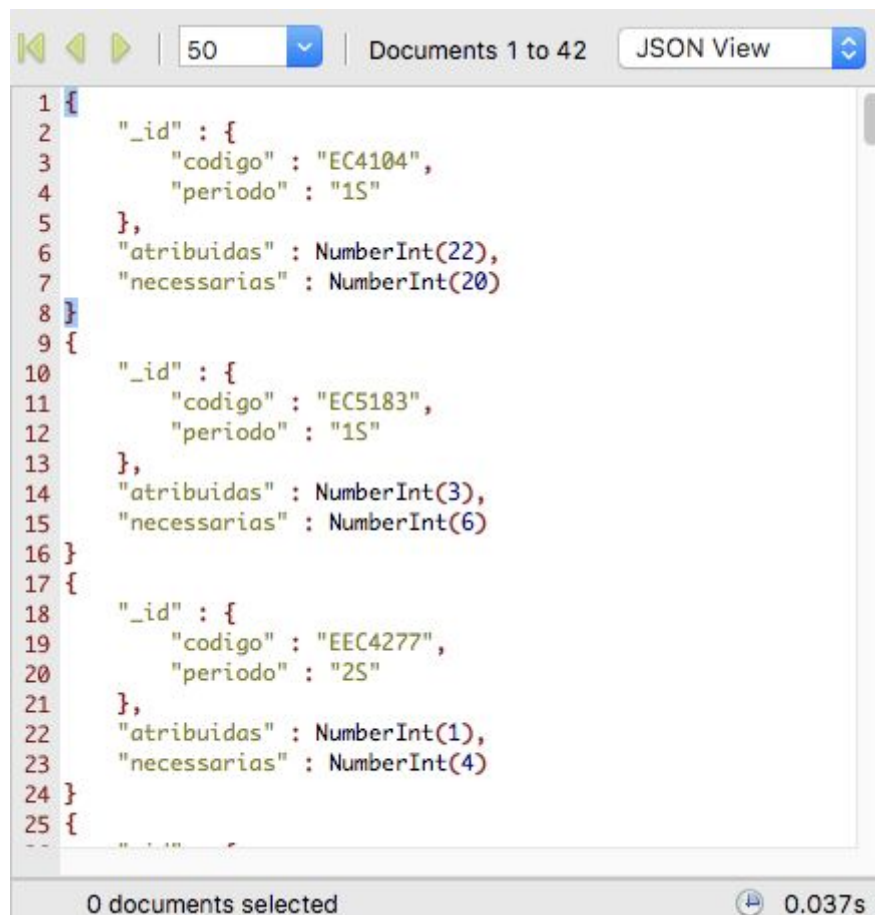


Figura 8 - Resultado da NoSQL query (b)

5.1.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.044	0.037

Tabela 2 - Resultados obtidos das queries em cada ambiente

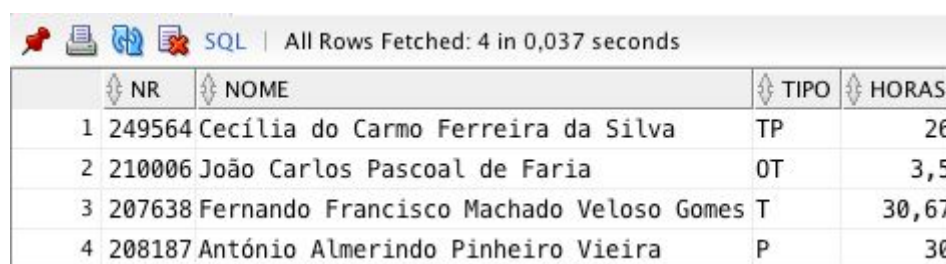
5.3 Query (c)

" Who is the professor with more class hours for each type of class, in the academic year 2003/2004? Show the number and name of the professor, the type of class and the total of class hours times the factor. "

5.3.1 SQL

```
with soma_horas as(
  SELECT gtd10.xdocentes.nr, gtd10.xdocentes.nome, gtd10.xtiposaula.tipo, sum(gtd10.xdsd.horas)as horas
  FROM gtd10.xdocentes
  JOIN gtd10.xdsd on gtd10.xdsd.nr = gtd10.xdocentes.nr
  JOIN gtd10.xtiposaula on gtd10.xtiposaula.id = gtd10.xdsd.id
  WHERE gtd10.xtiposaula.ano_letivo = '2003/2004'
  GROUP BY gtd10.xtiposaula.tipo, gtd10.xdocentes.nr, gtd10.xdocentes.nome
), media_horas as (
  SELECT tipo, max(horas) as horas from soma_horas group by tipo
)
select soma_horas.nr, soma_horas.nome, soma_horas.tipo, soma_horas.horas
from soma_horas
inner join media_horas on media_horas.tipo = soma_horas.tipo and media_horas.horas = soma_horas.horas;
```

Figura 9 - SQL query (c)



	NR	NOME	TIPO	HORAS
1	249564	Cecília do Carmo Ferreira da Silva	TP	26
2	210006	João Carlos Pascoal de Faria	OT	3,5
3	207638	Fernando Francisco Machado Veloso Gomes	T	30,67
4	208187	António Almerindo Pinheiro Vieira	P	30

Figura 10 - Resultado da SQL query (c)

5.3.2 NoSQL (MongoDB)

```
db.occurencias.aggregate([
  { $match: { "ano_lectivo": "2003/2004" } },
  { "$unwind": "$docentes" },
  { "$unwind": "$docentes.aulas" },
  { "$unwind": "$docentes.aulas.dsd" },
  { $group: { _id: { Tipo_Aula: "$docentes.aulas.tipo", prof_Nome: "$docentes.nome" }, soma_Aulas_Tipo: { $sum: "$docentes.aulas.dsd.horas" } } },
  { $sort: { "soma_Aulas_Tipo": -1 } },
  { $group: { _id: "$_id.Tipo_Aula", prof_Nome: { $first: "$_id.prof_Nome" }, soma_Aulas_Tipo: { $first: "$soma_Aulas_Tipo" } } }
])
```

Figura 11 - NoSQL Query (c)

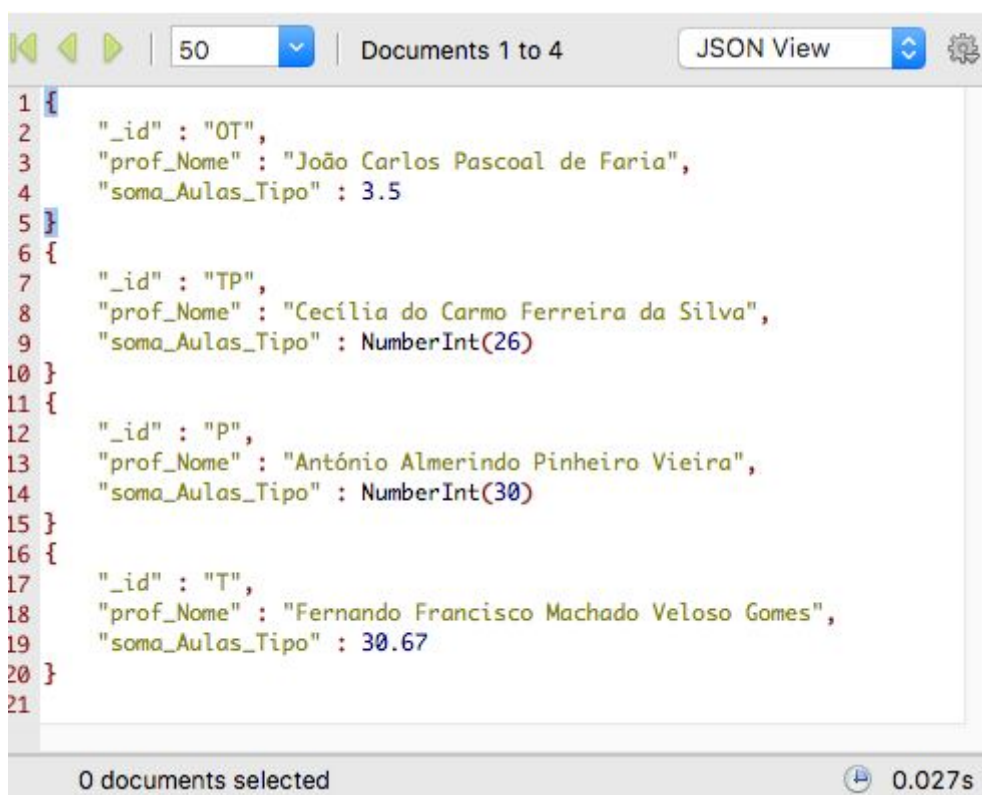


Figura 12 - Resultado da NoSQL query (c)

5.3.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.037	0.027

Tabela 3 - Resultados obtidos das queries em cada ambiente

5.4 Query (d)

" Which is the average number of hours by professor by year in each category, in the years between 2001/2002 and 2004/2005? "

5.4.1 SQL

```
SELECT gtd10.xdocentes.nome, gtd10.xdocentes.categoria, avg(gtd10.xdsd.horas) as horas, gtd10.xtiposaula.ano_letivo
FROM gtd10.xdocentes
JOIN gtd10.xdsd ON gtd10.xdsd.nr = gtd10.xdocentes.nr
JOIN gtd10.xtiposaula ON gtd10.xtiposaula.id = gtd10.xdsd.id
WHERE gtd10.xtiposaula.ano_letivo IN ('2001/2002', '2002/2003', '2003/2004', '2004/2005')
group by gtd10.xdocentes.nome, gtd10.xdocentes.categoria, gtd10.xtiposaula.ano_letivo
ORDER BY gtd10.xtiposaula.ano_letivo desc, gtd10.xdocentes.categoria desc;
```

Figura 13 - SQL query (d)

SQL Fetched 50 rows in 0,033 seconds				
NOME	CATEGORIA	HORAS	ANO_LE...	
1 Diogo de Matos Graça Aires de Campos	(null)	1,2	2004/2005	
2 Mário Abel dos Santos Moura	(null)	3	2004/2005	
3 Cândida Fernanda Antunes Ribeiro	19999	1	2004/2005	
4 Ariel Ricardo Negrão da Silva Guerreiro	19999	4	2004/2005	
5 Isabel Pinto	19999	1	2004/2005	
6 António Carvalho Pedrosa	19999	3	2004/2005	
7 Maria Manuela de Castro e Silva Ferreira	19999	3	2004/2005	
8 Joao Carlos Paiva	19999	3	2004/2005	
9 Álvaro Domingues	19999	1,553333...	2004/2005	
10 Maria Cristina Barbot Campos e Matos	19999	3	2004/2005	
11 João Armando Lobo de Sousa Couto	19999	2	2004/2005	
12 Adriano Joaquim Carvalho Barbosa Nazareth	19999	3	2004/2005	
13 Duarte José Vasconcelos da Costa Pereira	19999	3	2004/2005	
14 Álvaro Jorge da Maia Sêco	19999	1,25	2004/2005	
15 António Lacerda	19999	0,5	2004/2005	
16 José Miguel do Carmo Nunes da Silva	19999	3	2004/2005	
17 José Manuel Miguez Araújo	19999	2,75	2004/2005	
18 Tom Wilson	19999	3	2004/2005	

Figura 14 - Resultado da SQL query (d)

5.4.2 NoSQL (MongoDB)

```
db.ocorrencias.aggregate([
  { $match: { "ano_lectivo": { $in : ["2001/2002", "2002/2003", "2003/2004", "2004/2005"] } } },
  { "$unwind": "$docentes" },
  { "$unwind": "$docentes.aulas" },
  { "$unwind": "$docentes.aulas.dsd" },
  { $group: { _id: { ano: "$ano_lectivo", docente: "$docentes.nome", categoria: "$docentes.categoria" }, horas: { $avg: "$docentes.aulas.dsd.horas" } } },
  { $group: { _id: { ano: "$_id.ano", categoria: "$_id.categoria" }, professores: { $push: { nome: "$_id.docente", horas: "$horas" } } } },
  { $sort: { "_id.ano" : -1, "_id.categoria": -1 } }
]);
```

Figura 15 - NoSQL Query (d)

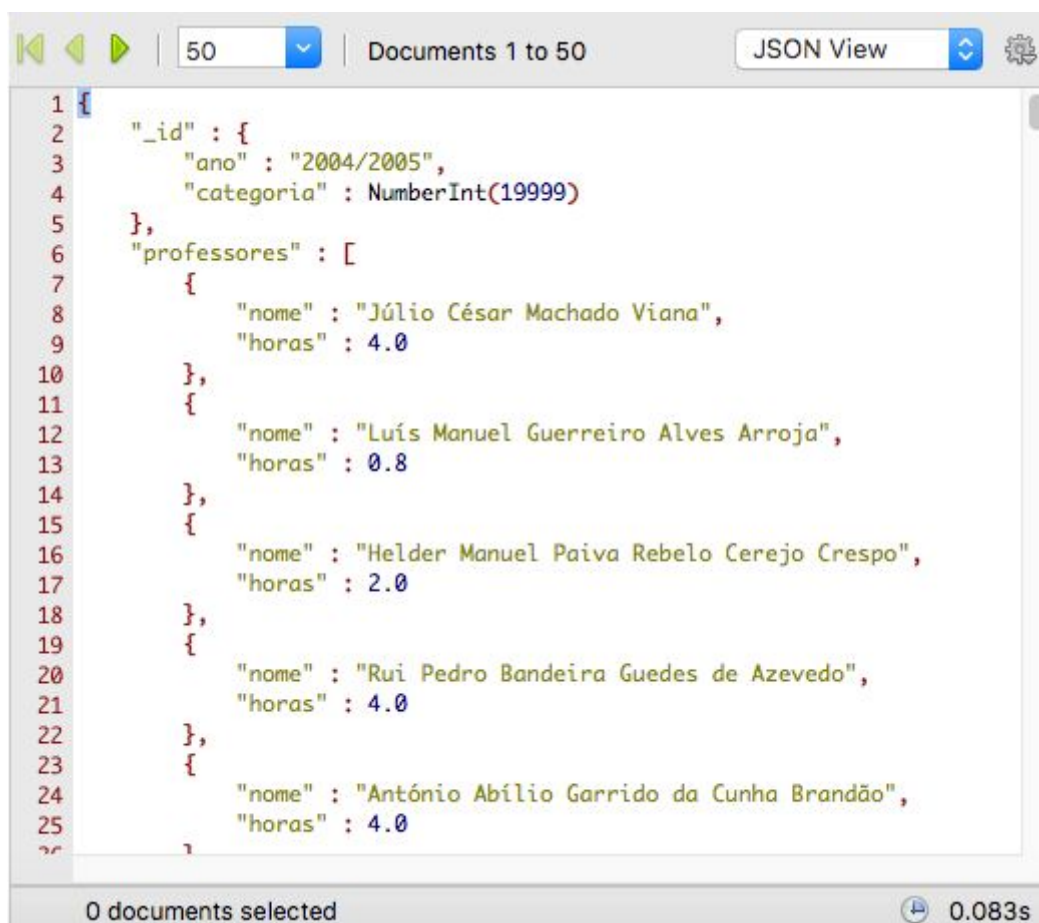


Figura 16 - Resultado da NoSQL query (d)

5.4.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.033	0.083

Tabela 4 - Resultados obtidos das queries em cada ambiente

5.5 Query (e)

" Which is the total hours per week, on each semester, that an hypothetical student enrolled in every course of a single curricular year from each program would get. "

5.5.1 SQL

```
SELECT gtd10.xucs.curso, gtd10.xtiposaula.periodo, gtd10.xtiposaula.ano_letivo,
sum(gtd10.xtiposaula.horas_turno * gtd10.xtiposaula.turnos) as horas
FROM gtd10.xtiposaula
JOIN gtd10.xocorrencias on (gtd10.xtiposaula.ano_letivo = gtd10.xocorrencias.ano_letivo and
gtd10.xtiposaula.periodo = gtd10.xocorrencias.periodo and
gtd10.xtiposaula.codigo = gtd10.xocorrencias.codigo)
JOIN gtd10.xucs on gtd10.xucs.codigo = gtd10.xocorrencias.codigo
group by gtd10.xucs.curso, gtd10.xtiposaula.periodo, gtd10.xtiposaula.ano_letivo
ORDER BY gtd10.xucs.curso, gtd10.xtiposaula.periodo, gtd10.xtiposaula.ano_letivo;
```

Figura 17 - SQL query (e)

SQL | Fetched 50 rows in 0,051 seconds

	CURSO	PERIODO	ANO_LE...	HORAS
1	1	2S	2008/2009	2
2	100	1S	1998/1999	2
3	100	1S	1999/2000	2
4	100	1S	2000/2001	2
5	100	1S	2002/2003	6
6	100	1S	2003/2004	36
7	100	1S	2004/2005	40
8	100	1S	2005/2006	72
9	100	1S	2006/2007	56
10	100	1T	1998/1999	6
11	100	1T	1999/2000	10
12	100	1T	2000/2001	6
13	100	1T	2001/2002	12
14	100	12	1998/1999	(null)
15	100	12	1999/2000	5
16	100	2S	1998/1999	6
17	100	2S	1999/2000	6
18	100	2S	2000/2001	4

Figura 18 - Resultado da SQL query (e)

5.5.2 NoSQL (MongoDB)

```
db.occurencias.aggregate([
  { $unwind: "$docentes" },
  { $unwind: "$docentes.aulas" },
  { $project: { _id: { codigo: "$codigo", periodo: "$periodo", curso: "$uc.curso", ano: "$ano_lectivo" }, aulas: "$docentes.aulas" } },
  { $group: { _id: { codigo: "$_id", t_aula: "$aulas.tipo" }, horas: { $sum: "$aulas.horas_turno" } } },
  { $group: { _id: "$_id.codigo", horas: { $sum: "$horas" } } },
  { $group: { _id: { curso: "$_id.curso", periodo: "$_id.periodo", ano: "$_id.ano" }, horas: { $sum: "$horas" } } },
  { $sort: { "_id.curso": 1, "_id.periodo": 1, "_id.ano": 1 } },
  // Está a separar por anos lectivos/curso/semestre, curriculares não há nada que permita (?)
  // Teria que adicionar essa separação algures, mas acho que não há
]);
```

Figura 19 - NoSQL Query (e)

```

1 {
2   "_id" : {
3     "curso" : null,
4     "periodo" : "1S",
5     "ano" : "2006/2007"
6   },
7   "horas" : NumberInt(1950)
8 }
9 {
10  "_id" : {
11    "curso" : null,
12    "periodo" : "1S",
13    "ano" : "2007/2008"
14  },
15  "horas" : NumberInt(2500)
16 }
17 {
18  "_id" : {
19    "curso" : null,
20    "periodo" : "1S",
21    "ano" : "2008/2009"
22  },
23  "horas" : NumberInt(1850)
24 }
25 {

```

0 documents selected 0.363s

Figura 20 - Resultado da NoSQL query (e)

5.5.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.051	0.363

Tabela 5 - Resultados obtidos das queries em cada ambiente

5.6 Query (f)

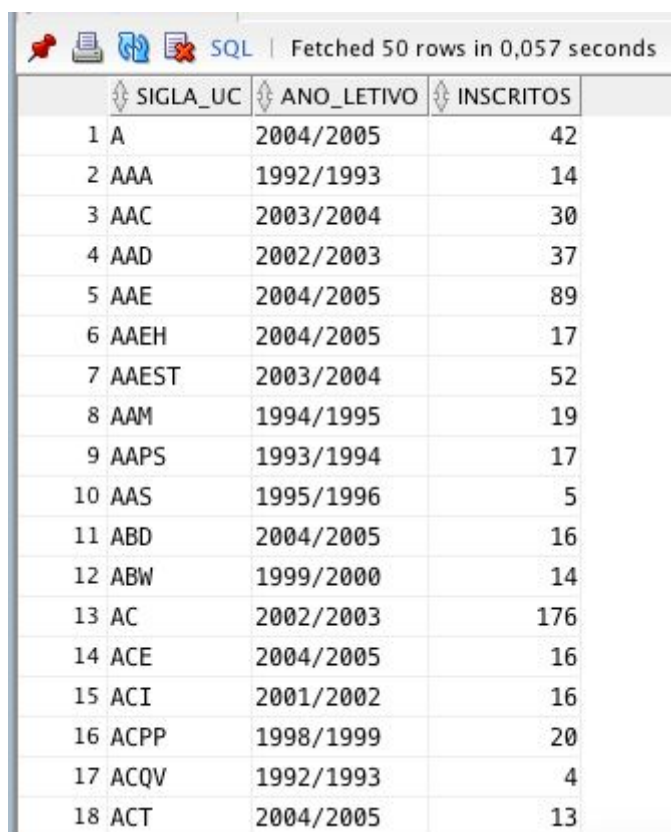
"Ask the database a query you think is interesting."

For each course, what is the year with more enrolled students. Showing the course, year and number of enrolled students, ordered by course.

5.6.1 SQL

```
with todos as(
  SELECT gtd10.xucs.sigla_uc, gtd10.xocorrencias.ano_letivo, gtd10.xocorrencias.inscritos
  FROM gtd10.xocorrencias
  JOIN gtd10.xucs on gtd10.xucs.codigo = gtd10.xocorrencias.codigo
  WHERE gtd10.xocorrencias.inscritos > 0
  ORDER BY gtd10.xucs.sigla_uc
), maximo_todos as(
  SELECT sigla_uc, max(inscritos) as inscritos from todos group by sigla_uc
)
select todos.sigla_uc, todos.ano_letivo, todos.inscritos
from todos
inner join maximo_todos on maximo_todos.sigla_uc = todos.sigla_uc and maximo_todos.inscritos = todos.inscritos
ORDER BY todos.sigla_uc;
```

Figura 21 - SQL query (f)



	SIGLA_UC	ANO_LETIVO	INSCRITOS
1	A	2004/2005	42
2	AAA	1992/1993	14
3	AAC	2003/2004	30
4	AAD	2002/2003	37
5	AAE	2004/2005	89
6	AAEH	2004/2005	17
7	AAEST	2003/2004	52
8	AAM	1994/1995	19
9	AAPS	1993/1994	17
10	AAS	1995/1996	5
11	ABD	2004/2005	16
12	ABW	1999/2000	14
13	AC	2002/2003	176
14	ACE	2004/2005	16
15	ACI	2001/2002	16
16	ACPP	1998/1999	20
17	ACQV	1992/1993	4
18	ACT	2004/2005	13

Figura 22 - Resultado da SQL query (f)

5.6.2 NoSQL (MongoDB)

```
// f) para cada uc, qual o ano com mais alunos inscritos! Mostrar a UC, ano e nr de inscritos, ordenado por UC.
db.occurencias.aggregate([
  { $group: { _id: { UC: "$uc.sigla", Ano: "$ano_lectivo", Inscritos: "$inscritos" } } },
  { $match: { "_id.Inscritos": { $gt: 1 }, "_id.UC": { $exists: 1 } } },
  { $sort: { "_id.Inscritos": -1 } },
  { $group: { _id: "$_id.UC", Inscritos: { $max: "$_id.Inscritos" }, Ano: { $first: "$_id.Ano" } } },
  { $sort: { "_id": 1 } },
])
```

Figura 23 - NoSQL Query (f)

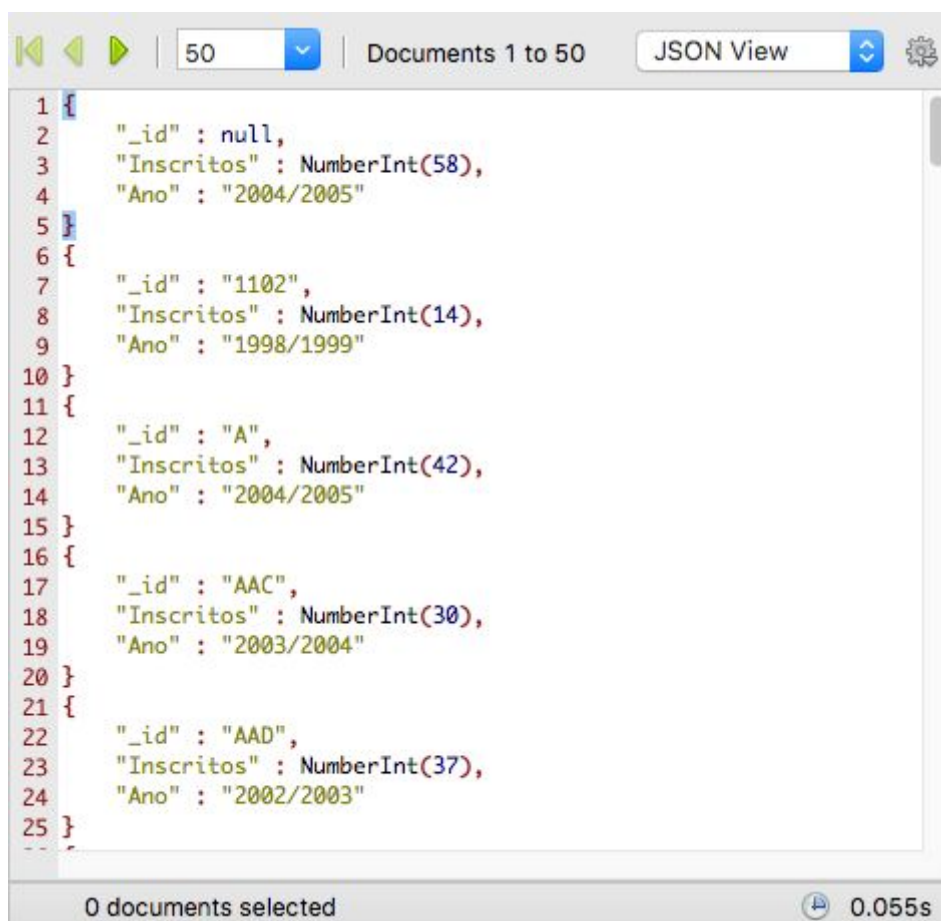


Figura 24 - Resultado da NoSQL query (f)

5.6.3 Resultados nos ambientes SQL e NoSQL

	SQL	NoSQL
Tempos (SI)	0.057	0.055

Tabela 6 - Resultados obtidos das queries em cada ambiente

6 Comparações do Mongo e Oracle

O Oracle SQL e MongoDB implementam ambas tecnologias de base dados, embora em estruturas diferentes. O grupo até este trabalho tinha apenas trabalhado com sintaxe de SQL, não tendo qualquer conhecimento da sintaxe em Mongo, tornando a adaptação um pouco difícil. O mongo disponibiliza mais que uma sintaxe para efetuar a query, sendo estas o *find*, *aggregate* e entre outras, sendo um pouco mais difícil de aprender, uma vez que a mesma query pode ter sintaxes e métodos diferentes.

Quanto ao tamanho dos dados, ambos têm desvantagens. Com o SQL conseguimos usar a operação *join* que facilita a agregação das tabelas, o que por outro lado acarreta um custo computacional no uso desta operação. Já o Mongo uma vez que não usa agregações, os dados têm de estar presentes na mesma tabela, criando assim redundância nos dados. Tem a vantagem de conseguir diminuir o espaço de armazenamento nos atributos a nulo, uma vez que estes não são guardados.

Quanto aos tempos de processamento, o grupo não detetou diferenças consideráveis, tendo os dois métodos tempos parecidos.

7 Anexos

7.1 Anexo 1 - Estrutura do modelo do documento em Mongo.

```
// Dados Ocorrencia
codigo: String,
ano_lectivo: String,
periodo: String,
inscritos: Number,
com_frequencia: Number,
aprovados: Number,
objectivos: String,
conteudo: String,
departamento: String,

// Unidade Curricular
uc : {
  designacao: String,
```

```

        sigla: String,
        curso: Number
    },

    // Docentes
    docentes : [
        {
            nr : Number,
            nome: String,
            sigla: String,
            categoria: Number,
            proprio: String,
            apelido: String,
            estado: String,

            // Aulas
            aulas : [
                {
                    dsd: {
                        horas: Number,
                        fator: Number,
                        ordem: Number
                    },
                    tipo: String,
                    turnos: Number,
                    n_aulas: Number,
                    horas_turno: Number,
                }
            ]
        }
    ]
}
]

```

7.2 Anexo 2 - Código em SQL usado para criação das vistas

```

create or replace view ocs_ucs_view as (
    select
        gtdl0.xocorrencias.codigo as ocorrencia_codigo,
        gtdl0.xocorrencias.ano_letivo as ocorrencia_ano_letivo,
        gtdl0.xocorrencias.periodo as ocorrencia_periodo,
        gtdl0.xocorrencias.inscritos as ocorrencia_inscritos,
        gtdl0.xocorrencias.com_frequencia as ocorrencia_com_frequencia,
        gtdl0.xocorrencias.aprovados as ocorrencia_aprovados,
        gtdl0.xocorrencias.objetivos as ocorrencia_objetivos,
        gtdl0.xocorrencias.conteudo as ocorrencia_conteudo,
        gtdl0.xocorrencias.departamento as ocorrencia_departamento,
        gtdl0.xucs.designacao as uc_designacao,
        gtdl0.xucs.sigla_uc as uc_sigla_uc,
        gtdl0.xucs.curso as uc_curso
    from

```

```

        gtd10.xocorrencias
    join
        gtd10.xucs
    on
        gtd10.xocorrencias.codigo = gtd10.xucs.codigo
);

create or replace view doc_dsd_view as (
    select
        gtd10.xdocentes.nome,
        gtd10.xdocentes.sigla,
        gtd10.xdocentes.categoria,
        gtd10.xdocentes.proprio,
        gtd10.xdocentes.apelido,
        gtd10.xdocentes.estado,
        gtd10.xdsd.horas,
        gtd10.xdsd.fator,
        gtd10.xdsd.ordem,
        gtd10.xdsd.id,
        gtd10.xdocentes.nr
    from
        gtd10.xdsd
    join
        gtd10.xdocentes
    on
        gtd10.xdsd.nr = gtd10.xdocentes.nr
);

create or replace view doc_dsd_taulas_view as (
    select
        doc_dsd_view.nome as docente_nome,
        doc_dsd_view.sigla as docente_sigla,
        doc_dsd_view.categoria as docente_categoria,
        doc_dsd_view.proprio as docente_proprio,
        doc_dsd_view.apelido as docente_apelido,
        doc_dsd_view.estado as docente_estado,
        doc_dsd_view.horas as dsd_horas,
        doc_dsd_view.fator as dsd_fator,
        doc_dsd_view.ordem as dsd_order,
        gtd10.xtiposaula.tipo as aula_tipo,
        gtd10.xtiposaula.turnos as aula_turnos,
        gtd10.xtiposaula.n_aulas as aula_n_aulas,
        gtd10.xtiposaula.horas_turno as aula_horas_turno,
        gtd10.xtiposaula.ano_letivo as j_ano_letivo,
        gtd10.xtiposaula.codigo as j_codigo,
        gtd10.xtiposaula.periodo as j_periodo,
        doc_dsd_view.nr as docente_nr
    from
        doc_dsd_view
    join
        gtd10.xtiposaula
    on
        doc_dsd_view.id = gtd10.xtiposaula.id
);

create or replace view master_view as (
    select

```

```

        ocorrencia_codigo,
        ocorrencia_ano_letivo,
        ocorrencia_periodo,
        ocorrencia_inscritos,
        ocorrencia_com_frequencia,
        ocorrencia_aprovados,
        ocorrencia_objetivos,
        ocorrencia_conteudo,
        ocorrencia_departamento,
        uc_designacao,
        uc_sigla_uc,
        uc_curso,
        docente_nome,
        docente_sigla,
        docente_categoria,
        docente proprio,
        docente_apelido,
        docente_estado,
        dsd_horas,
        dsd_fator,
        dsd_order,
        aula_tipo,
        aula_turnos,
        aula_n_aulas,
        aula_horas_turno,
        docente_nr
    from
        ocs_ucs_view
    join
        doc_dsd_taulas_view
    on
        ocs_ucs_view.ocorrencia_codigo = doc_dsd_taulas_view.j_codigo and
        ocs_ucs_view.ocorrencia_ano_letivo = doc_dsd_taulas_view.j_ano_letivo

        ocs_ucs_view.ocorrencia_periodo = doc_dsd_taulas_view.j_periodo
);

```

7.3 Anexo 3 - Código em JS usado para inserir os dados no MongoDB

```

var rows = j.RESULTS.ROW;
var totalRows = rows.length - 1;

var obj = {};
var obj_d = {};

for(var i in rows){
    var row = rows[i].COLUMN;
    console.log("Processing row %d / %d", i, totalRows);
}

```

```

var ocorrencia_codigo          = (row[0]._ || null);
var ocorrencia_ano_letivo      = (row[1]._ || null);
var ocorrencia_periodo        = (row[2]._ || null);
var ocorrencia_inscritos      = (row[3]._ || "").replace(",", ".");
var ocorrencia_com_frequencia = (row[4]._ || "").replace(",", ".");
var ocorrencia_aprovados      = (row[5]._ || "").replace(",", ".");
var ocorrencia_objetivos      = (row[6]._ || null);
var ocorrencia_conteudo       = (row[7]._ || null);
var ocorrencia_departamento   = (row[8]._ || null);
var uc_designacao             = (row[9]._ || null);
var uc_sigla_uc               = (row[10]._ || null);
var uc_curso                  = (row[11]._ || "").replace(",", ".");
var docente_nome              = (row[12]._ || null);
var docente_sigla             = (row[13]._ || null);
var docente_categoria         = (row[14]._ || "").replace(",", ".");
var docente proprio          = (row[15]._ || null);
var docente_apelido           = (row[16]._ || null);
var docente_estado            = (row[17]._ || null);
var dsd_horas                 = (row[18]._ || "").replace(",", ".");
var dsd_fator                 = (row[19]._ || "").replace(",", ".");
var dsd_ordem                 = (row[20]._ || "").replace(",", ".");
var aula_tipo                 = (row[21]._ || null);
var aula_turnos               = (row[22]._ || "").replace(",", ".");
var aula_n_aulas              = (row[23]._ || "").replace(",", ".");
var aula_horas_turno          = (row[24]._ || "").replace(",", ".");
var docente_nr                = (row[25]._ || "").replace(",", ".");

var key_oc = ocorrencia_ano_letivo + ":@" + ocorrencia_codigo + ":@" +
encia_periodo;

if(obj[key_oc] == null){
    obj[key_oc] = {
        "codigo": ocorrencia_codigo,
        "ano_letivo": ocorrencia_ano_letivo,
        "periodo": ocorrencia_periodo,
        "inscitos": ocorrencia_inscritos,
        "com_frequencia": ocorrencia_com_frequencia,
        "aprovados": ocorrencia_aprovados,
        "objetivos": ocorrencia_objetivos,
        "conteudo": ocorrencia_conteudo,
        "departamento": ocorrencia_departamento,

        "uc" : {
            "designacao": uc_designacao,
            "sigla": uc_sigla_uc,
            "curso": uc_curso
        },

        "docentes" : []
    };
}

var key_dc = docente_sigla;

if(obj_d[key_oc] == null){
    obj_d[key_oc] = {};
    obj_d[key_oc]["docente_list"] = [];
}

```

```

    }

    if(obj_d[key_oc][key_dc] == null){
        obj_d[key_oc][key_dc] = {
            "nr" : docente_nr,
            "nome" : docente_nome,
            "sigla" : docente_sigla,
            "categoria" : docente_categoria,
            "proprio" : docente_proprio,
            "apelido" : docente_apelido,
            "estado" : docente_estado,

            "aulas" : []
        };

        obj_d[key_oc]["docente_list"].push(key_dc);
    }

    obj_d[key_oc][key_dc].aulas.push({
        "dsd" : {
            "horas": dsd_horas,
            "fator": dsd_fator,
            "ordem": dsd_ordem
        },
        "tipo": aula_tipo,
        "turnos": aula_turnos,
        "n_aulas": aula_n_aulas,
        "horas_turno": aula_horas_turno
    });
}

var docs = [];

console.log("Building documents...");
for(var ii in obj){
    var o = obj[ii];

    var k_ano = o.ano_lectivo || "";
    var k_cod = o.codigo || "";
    var k_per = o.periodo || "";

    var k = k_ano + "::-" + k_cod + "::-" + k_per;
    var l = obj_d[k]["docente_list"];

    for(var iii in l){
        o.docentes.push(obj_d[k][l[iii]]);
    }

    docs.push(new dbp.Ocorrencia(o));
}

dbp.Ocorrencia.collection.insert(docs, (err, docs) => {
    console.log("Inserted documents into MongoDB");
    dbp.db.close();
});

```