

Aluna: Daniella Maleski

Exercício Banco de dados SQL

1. Abra o IDE DBeaver (ou o que você tiver acesso) e crie a seguinte tabela:

```
CREATE TABLE EBAC(  
  Alunoid INTEGER PRIMARY KEY AUTOINCREMENT,  
  Nome VARCHAR(30),  
  Curso VARCHAR (20),  
  Nota INTEGER(2)  
);
```

2. Na sequência insira os seguintes dados:

```
INSERT INTO EBAC (Nome, Curso, Nota)  
VALUES  
  ("Fábio", "QA", 5),  
  ("José Pedro", "Dev", 8),  
  ("Mariana", "QA", 9),  
  ("Aline", "QA", 6),  
  ("Alice", "SQL", 7),  
  ("João", "Dev", 5),  
  ("Alan", "QA", 8),  
  ("Wesley", "SQL", 4),  
  ("Pedro", "UX", 3);
```

3. Selecione todos os dados da tabela EBAC, ordenando o nome em ordem alfabética.

Resposta:

```
SELECT * FROM EBAC  
order by Nome ASC
```

	123 Alunoid ▼	A-Z Nome ▼	A-Z Curso ▼	123 Nota ▼
1	7	Alan	QA	8
2	5	Alice	SQL	7
3	4	Aline	QA	6
4	1	Fábio	QA	5
5	2	José Pedro	Dev	8
6	6	João	Dev	5
7	3	Mariana	QA	9
8	9	Pedro	UX	3
9	8	Wesley	SQL	4

4. Selecione Todos os alunos do curso de QA.

Resposta:

```
SELECT * FROM EBAC
WHERE Curso = "QA"
order by Nome ASC
```

	123 Alunold ▼	A-Z Nome ▼	A-Z Curso ▼	123 Nota ▼
1	7	Alan	QA	8
2	4	Aline	QA	6
3	1	Fábio	QA	5
4	3	Mariana	QA	9

5. Selecione todos os alunos com nota maior e igual a 6.

Resposta:

```
SELECT * FROM EBAC
WHERE Nota >= 6
ORDER BY Nota ASC
```

	123 Alunold ▼	A-Z Nome ▼	A-Z Curso ▼	123 Nota ▼
1	4	Aline	QA	6
2	5	Alice	SQL	7
3	2	José Pedro	Dev	8
4	7	Alan	QA	8
5	3	Mariana	QA	9

6. Selecione todos os alunos que tem a palavra "Pedro" no nome.

```
SELECT * FROM EBAC
where Nome like "%Pedro%"
```

	123 Alunold ▼	A-Z Nome ▼	A-Z Curso ▼	123 Nota ▼
1	2	José Pedro	Dev	8
2	9	Pedro	UX	3

Exercício Banco de dados MongoDB

1. Execute o docker e abara o MongoDB Compass e crie o seguinte banco:

```
use EBAC
```

2. Crie a seguinte coleção:

```
db.alunos.insertMany([
{
  "nome": "Fábio",
  "nota": 7,
```

```

    "curso": "QA"
  },
  {
    "nome": "Alice",
    "nota": 9,
    "curso": "SQL"
  },
  {
    "nome": "Mariana",
    "cargo": "Professora",
    "curso": ["QA", "FrontEnd", "MongoDB"]
  },
  {
    "nome": "João",
    "nota": 7,
    "curso": "QA"
  },
  {
    "nome": "Paulo",
    "nota": 5,
    "curso": "Dev"
  },
  {
    "nome": "Maria",
    "nota": 8,
    "curso": "QA"
  },
  {
    "nome": "José",
    "nota": 4,
    "curso": "SQL"
  },
  {
    "nome": "Ana",
    "nota": 9,
    "curso": "QA"
  },
  {
    "nome": "José Pedro",
    "nota": 7,
    "curso": "UX"
  }
]

```

1. Selecione todos os dados da Collection Alunos, ordenando o nome em ordem alfabética.

Resposta:

db.EBAC.find({}).sort({nome: 1})

```
> db.EBAC.find({}).sort({nome: 1})
< [
  {
    _id: ObjectId('67a546d4100bea8d9c9dc00e'),
    nome: 'Alice',
    nota: 9,
    curso: 'SQL'
  },
  {
    _id: ObjectId('67a546d4100bea8d9c9dc014'),
    nome: 'Ana',
    nota: 9,
    curso: 'QA'
  },
  {
    _id: ObjectId('67a546d4100bea8d9c9dc00d'),
    nome: 'Fábio',
    nota: 7,
    curso: 'QA'
  },
  {
    _id: ObjectId('67a546d4100bea8d9c9dc013'),
    nome: 'José',
    nota: 4,
    curso: 'SQL'
  },
  {
    _id: ObjectId('67a546d4100bea8d9c9dc015'),
    nome: 'José Pedro',
    nota: 7,
    curso: 'UX'
  }
]
```

```
{
  _id: ObjectId('67a546d4100bea8d9c9dc010'),
  nome: 'João',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc012'),
  nome: 'Maria',
  nota: 8,
  curso: 'QA'
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc00f'),
  nome: 'Mariana',
  cargo: 'Professora',
  curso: [
    'QA',
    'FrontEnd',
    'MongoDB'
  ]
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc011'),
  nome: 'Paulo',
  nota: 5,
  curso: 'Dev'
}
```

2. Selecione todos os alunos do curso de SQL.

Resposta:

db.EBAC.find({curso: "SQL"})

```
> db.EBAC.find({curso: "SQL"})
< [
  {
    _id: ObjectId('67a546d4100bea8d9c9dc00e'),
    nome: 'Alice',
    nota: 9,
    curso: 'SQL'
  },
  {
    _id: ObjectId('67a546d4100bea8d9c9dc013'),
    nome: 'José',
    nota: 4,
    curso: 'SQL'
  }
]
```

3. Selecione todos os alunos com “nota maior e igual a 6” e “do curso de QA”.

Resposta:

db.EBAC.find({ \$and: [{nota: {\$gte: 6}}, {curso: "QA"}]})

```
> db.EBAC.find({
  $and: [
    {nota: {$gte: 6}},
    {curso: "QA"}
  ]
})
< {
  _id: ObjectId('67a546d4100bea8d9c9dc00d'),
  nome: 'Fábio',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc010'),
  nome: 'João',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc012'),
  nome: 'Maria',
  nota: 8,
  curso: 'QA'
}
{
  _id: ObjectId('67a546d4100bea8d9c9dc014'),
  nome: 'Ana',
  nota: 9,
  curso: 'QA'
}
```

4. Selecione todos os alunos que tem a palavra “Pedro” no nome.
Resposta:

db.EBAC.find({nome: /Pedro/})

```
> db.EBAC.find({nome: /Pedro/})
< {
  _id: ObjectId('67a546d4100bea8d9c9dc015'),
  nome: 'José Pedro',
  nota: 7,
  curso: 'UX'
}
```