

Exercício Banco de dados SQL

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

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

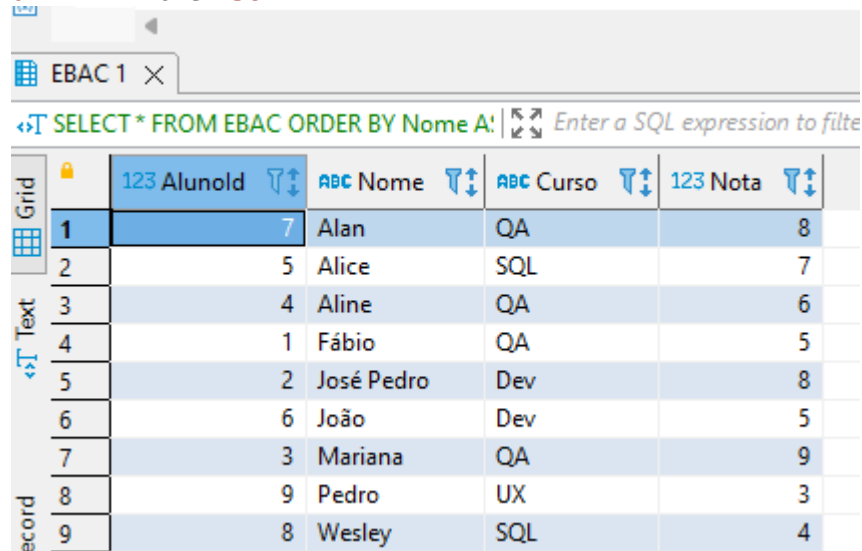
2. Na sequência insira os seguintes dados:

```
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
```



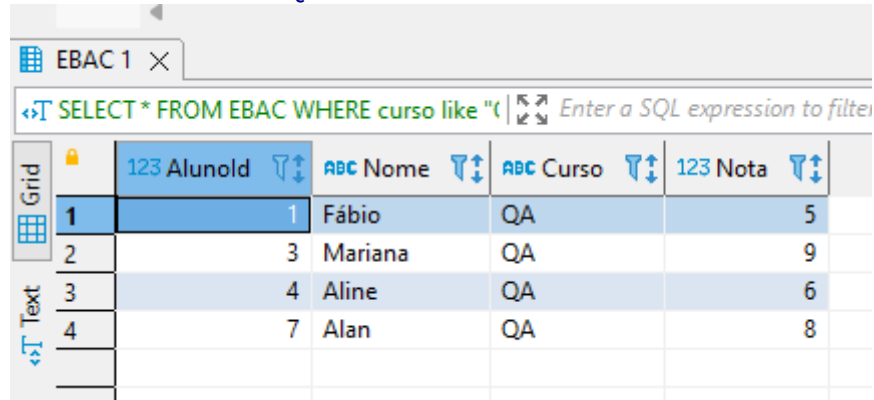
The screenshot shows the DBeaver SQL editor with the query `SELECT * FROM EBAC ORDER BY Nome ASC` entered. Below the query, a data grid displays the results, sorted by name (Nome) in ascending order. The grid has columns for Alunold, Nome, Curso, and Nota. The data is as follows:

	Alunold	Nome	Curso	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 like "QA"
```

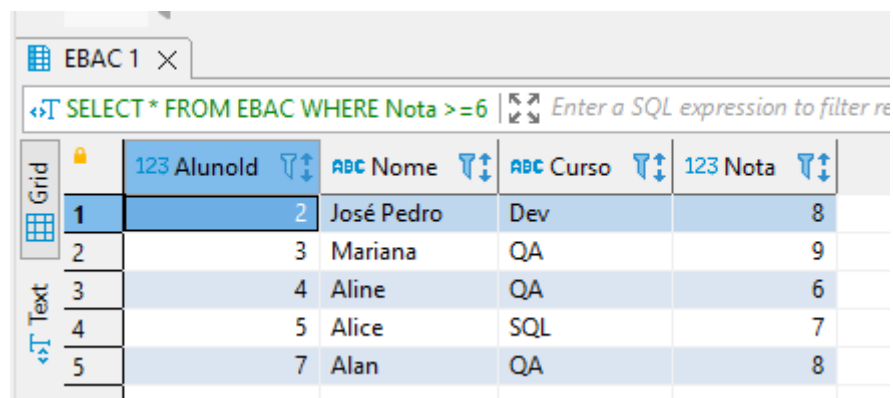


Grid	123 Alunold	ABC Nome	ABC Curso	123 Nota
1	1	Fábio	QA	5
2	3	Mariana	QA	9
3	4	Aline	QA	6
4	7	Alan	QA	8

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

Resposta:

```
SELECT *  
FROM EBAC  
WHERE Nota >=6
```

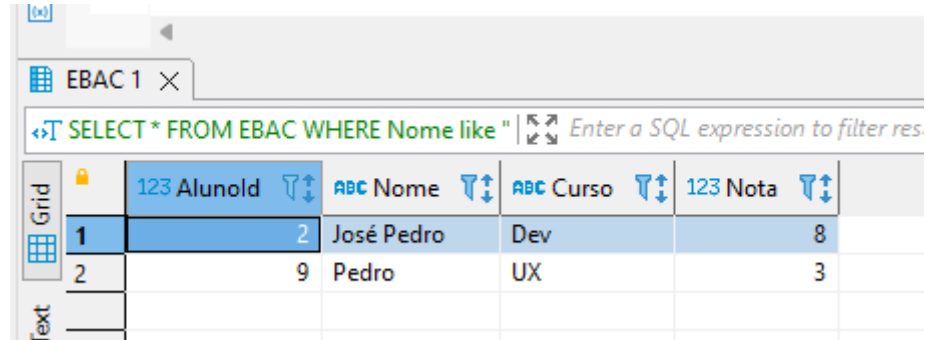


Grid	123 Alunold	ABC Nome	ABC Curso	123 Nota
1	2	José Pedro	Dev	8
2	3	Mariana	QA	9
3	4	Aline	QA	6
4	5	Alice	SQL	7
5	7	Alan	QA	8

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

Resposta:

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



Grid	123 Alunold	ABC Nome	ABC Curso	123 Nota
1	2	José Pedro	Dev	8
2	9	Pedro	UX	3

Exercício Banco de dados MongoDB

1. Execute o docker e abra 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.alunos.find({}).sort({nome: 1})
```

```

< { _id: ObjectId("62698da359ef74567dd4ee57"),
  nome: 'Alice',
  nota: 9,
  curso: 'SQL' }
{ _id: ObjectId("62698da359ef74567dd4ee5c"),
  nome: 'Ana',
  nota: 9,
  curso: 'QA' }
{ _id: ObjectId("62698da359ef74567dd4ee56"),
  nome: 'Fabio',
  nota: 6,
  curso: 'QA' }
{ _id: ObjectId("62698da359ef74567dd4ee5b"),
  nome: 'José',
  nota: 4,
  curso: 'SQL' }
{ _id: ObjectId("62698da359ef74567dd4ee59"),
  nome: 'João',
  nota: 7,
  curso: 'QA' }
{ _id: ObjectId("62698da359ef74567dd4ee5d"),
  nome: 'João Pedro',
  nota: 7,
  curso: 'UX' }
{ _id: ObjectId("62698da359ef74567dd4ee55"),
  nome: 'Maria',
  nota: 7,
  curso: 'QA' }
{ _id: ObjectId("62698da359ef74567dd4ee58"),
  nome: 'Mariana',
  cargo: 'Professora',
  curso: [ 'QA', 'FrontEnd', 'MongoDB' ] }

```

2. Selecione todos os alunos do curso de SQL.

Resposta:

```
db.alunos.find({curso: "SQL"})
```

```
< { _id: ObjectId("62698da359ef74567dd4ee57"),  
  nome: 'Alice',  
  nota: 9,  
  curso: 'SQL' }  
{ _id: ObjectId("62698da359ef74567dd4ee5b"),  
  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.alunos.find({  
  $and: [  
    {nota: {$gte: 6}},  
    {curso: "QA"}  
  ]  
})
```

```
< { _id: ObjectId("62698da359ef74567dd4ee55"),  
  nome: 'Maria',  
  nota: 7,  
  curso: 'QA' }  
{ _id: ObjectId("62698da359ef74567dd4ee56"),  
  nome: 'Fabio',  
  nota: 6,  
  curso: 'QA' }  
{ _id: ObjectId("62698da359ef74567dd4ee59"),  
  nome: 'João',  
  nota: 7,  
  curso: 'QA' }  
{ _id: ObjectId("62698da359ef74567dd4ee5c"),  
  nome: 'Ana',  
  nota: 9,  
  curso: 'QA' }
```

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

Resposta:

```
db.alunos.find({nome: /Pedro/})
```

```
< { _id: ObjectId("62698da359ef74567dd4ee5d"),  
  nome: 'João Pedro',  
  nota: 7,  
  curso: 'UX' }
```