

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)  
);
```

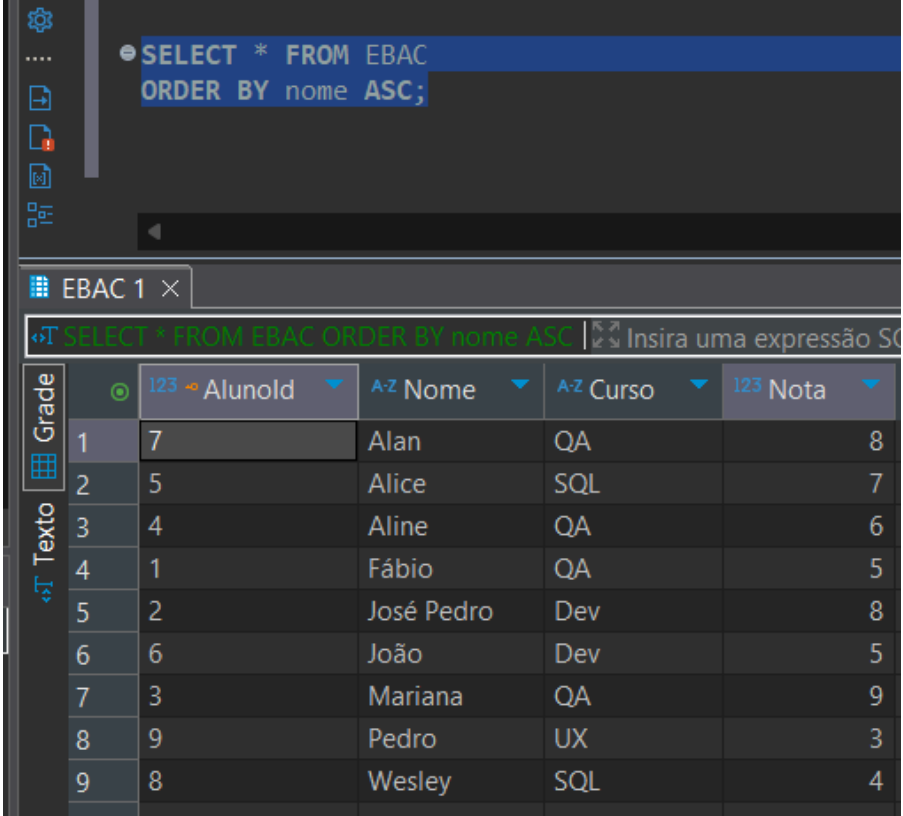
2. Na sequência insira os seguintes dados:

```
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:



The screenshot shows the DBeaver IDE interface. At the top, a SQL editor contains the query: `SELECT * FROM EBAC ORDER BY nome ASC;`. Below the editor, a tab labeled 'EBAC 1' is active, displaying the results of the query in a table view. The table has five columns: 'Alunold' (integer), 'Nome' (text), 'Curso' (text), and 'Nota' (integer). The data is sorted by 'Nome' in ascending order. The results are 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 = 'QA';
```

C 1 x

SELECT * FROM EBAC WHERE Curso = 'QA' | Insira uma expressão SQL

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

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

Resposta:

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

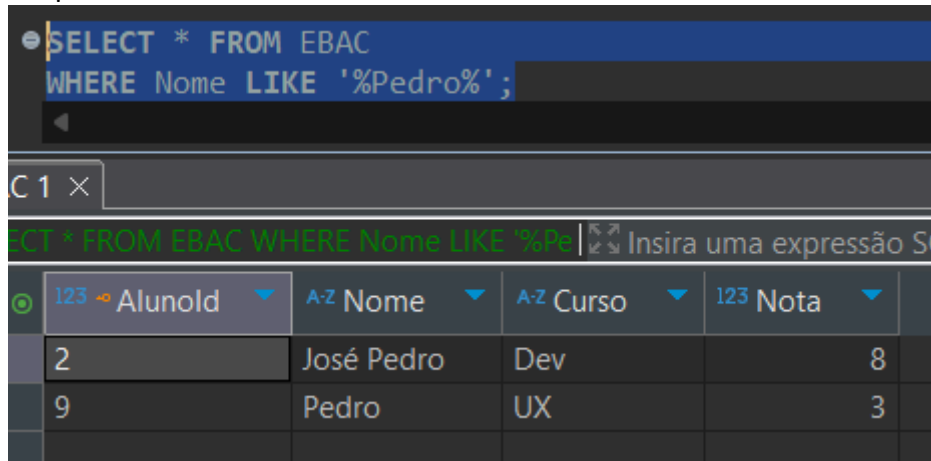
C 1 x

SELECT * FROM EBAC WHERE Nota >= 6 | Insira uma expressão SQL p

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

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

Resposta:



```
SELECT * FROM EBAC
WHERE Nome LIKE '%Pedro%';
```

123	Alunoid	A-Z Nome	A-Z Curso	123	Nota
2		José Pedro	Dev		8
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:

```
>_MONGOSH
> db.alunos.find().sort({ nome: 1 })
< {
  _id: ObjectId('685ec4988d9971efe81d8ce2'),
  nome: 'Alice',
  nota: 9,
  curso: 'SQL'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce8'),
  nome: 'Ana',
  nota: 9,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce1'),
  nome: 'Fábio',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce7'),
  nome: 'José',
  nota: 4,
  curso: 'SQL'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce9'),
  nome: 'José Pedro',
  nota: 7,
  curso: 'UX'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce4'),
  nome: 'João',
  nota: 7,
  curso: 'QA'
}
}
```

```
{
  _id: ObjectId('685ec4988d9971efe81d8ce6'),
  nome: 'Maria',
  nota: 8,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce3'),
  nome: 'Mariana',
  cargo: 'Professora',
  curso: [
    'QA',
    'FrontEnd',
    'MongoDB'
  ]
}
```

2. Selecione todos os alunos do curso de SQL.

Resposta:

```
> db.alunos.find({ curso: "SQL" }, { nome: 1, _id: 0 })
< {
  nome: 'Alice'
}
{
  nome: 'José'
}
```

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

Resposta:

```
> db.alunos.find({
  nota: { $gte: 6 },
  curso: "QA"
})
< {
  _id: ObjectId('685ec4988d9971efe81d8ce1'),
  nome: 'Fábio',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce4'),
  nome: 'João',
  nota: 7,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce6'),
  nome: 'Maria',
  nota: 8,
  curso: 'QA'
}
{
  _id: ObjectId('685ec4988d9971efe81d8ce8'),
  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('685ec4988d9971efe81d8ce9'),
  nome: 'José Pedro',
  nota: 7,
  curso: 'UX'
}
```