# Exercício Banco de dados SQL - Itnã

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)

);

1. 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);

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

Resposta: SELECT \* FROM EBAC ORDER BY Nome;

1. Selecione Todos os alunos do curso de QA.

Resposta: SELECT \* FROM EBAC WHERE Curso = 'QA';

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

Resposta: SELECT \* FROM EBAC WHERE Nota >= 6;

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

Resposta: SELECT \* FROM EBAC WHERE Nome LIKE '%Pedro%';

# Exercício Banco de dados MongoDB

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

use EBAC

1. 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 });

1. Selecione todos os alunos do curso de SQL.

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

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

Resposta: db.alunos.find({

$and: [

{ nota: { $gte: 6 } }, // Nota maior ou igual a 6

{ curso: "QA" } // Curso é "QA"

]

});

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

Resposta: db.alunos.find({

nome: { $regex: "Pedro", $options: "i" }

});