

package.json > ...

```
1 {
2   "name": "nodejs",
3   "version": "1.0.0",
4   "main": "index.js",
5   "scripts": {
6     "test": "echo \"Error: no test specified\" && exit 1"
7   },
8   "keywords": [],
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "@types/node": "^18.0.6",
13    "knex": "^2.4.2",
14    "node-fetch": "^3.2.6",
15    "sqlite3": "^5.1.6"
16  },
17  "description": ""
18 }
19
```



 Invite



Deploy



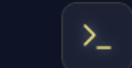
171735_...

js

Authentic



Console



>_



>_



Storage



Storage



Storage

×

? Help

JS index.js

```
1 const Alunos = require('./model/alunos');
2 async function run() {
3   try {
4     // Listar todos os alunos
5     const allAlunos = await Alunos.getAll();
6     console.log('Todos os alunos:', allAlunos);
7     const newAluno = { nome: 'Pedro', email:
      'pedro@example.com'};
8     const createdAluno = await Alunos.create(newAluno);
9     console.log('Aluno criado:', createdAluno);
10
11
12   } catch (error) {
13     console.error('Erro ao listar os alunos:', error);
14   }
15 }
16
17 run();
18
```

Ln 7, Col 5 History 🕒

```
Hint: hit control+c anytime to enter
REPL.
```

```
Todos os alunos: [
  { id: 5, nome: 'João', email: 'joao@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' }
]
Aluno criado: [ 7 ]
```

avaliacao-10

SophiaMello

Stop

Search

Files

index.js

migrations

20230626171735_...

model

alunos.js

disciplinas.js

professores.js

seeds

Tools

Deployments

Ghostwriter

Authenticati...

Chat

Code Search

Console

Database

Debugger

Docs

CPU

RAM

Storage

Try Ghostwriter

Help

index.js

package.json

knexfile.js

2023062617: +

index.js

```
1  const Alunos = require('./model/alunos');
2  async function run() {
3    try {
4      // Listar todos os alunos
5      /*const createdAluno = await Alunos.create(newAluno);
6      console.log('Aluno criado:', createdAluno);*/
7      const updatedAluno = { nome: 'Pedro Silva', email:
'pedro.silva@example.com' };
8
9      const idToUpdate = 5;
10     const result = await Alunos.update(idToUpdate,
updatedAluno);
11     console.log('Aluno atualizado:', result);
12
13     const allAlunos = await Alunos.getAll();
14     console.log('Todos os alunos:', allAlunos);
15     const newAluno = { nome: 'Pedro', email:
'pedro@example.com'};
16
17   } catch (error) {
18     console.error('Erro ao listar os alunos:', error);
19   }
20 }
21 run();
22
```

Console

Hint: hit control+c anytime to enter REPL.

Aluno atualizado: 1

Todos os alunos: [

{ id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },

{ id: 6, nome: 'Maria', email: 'maria@example.com' },

{ id: 7, nome: 'Pedro', email: 'pedro@example.com' }

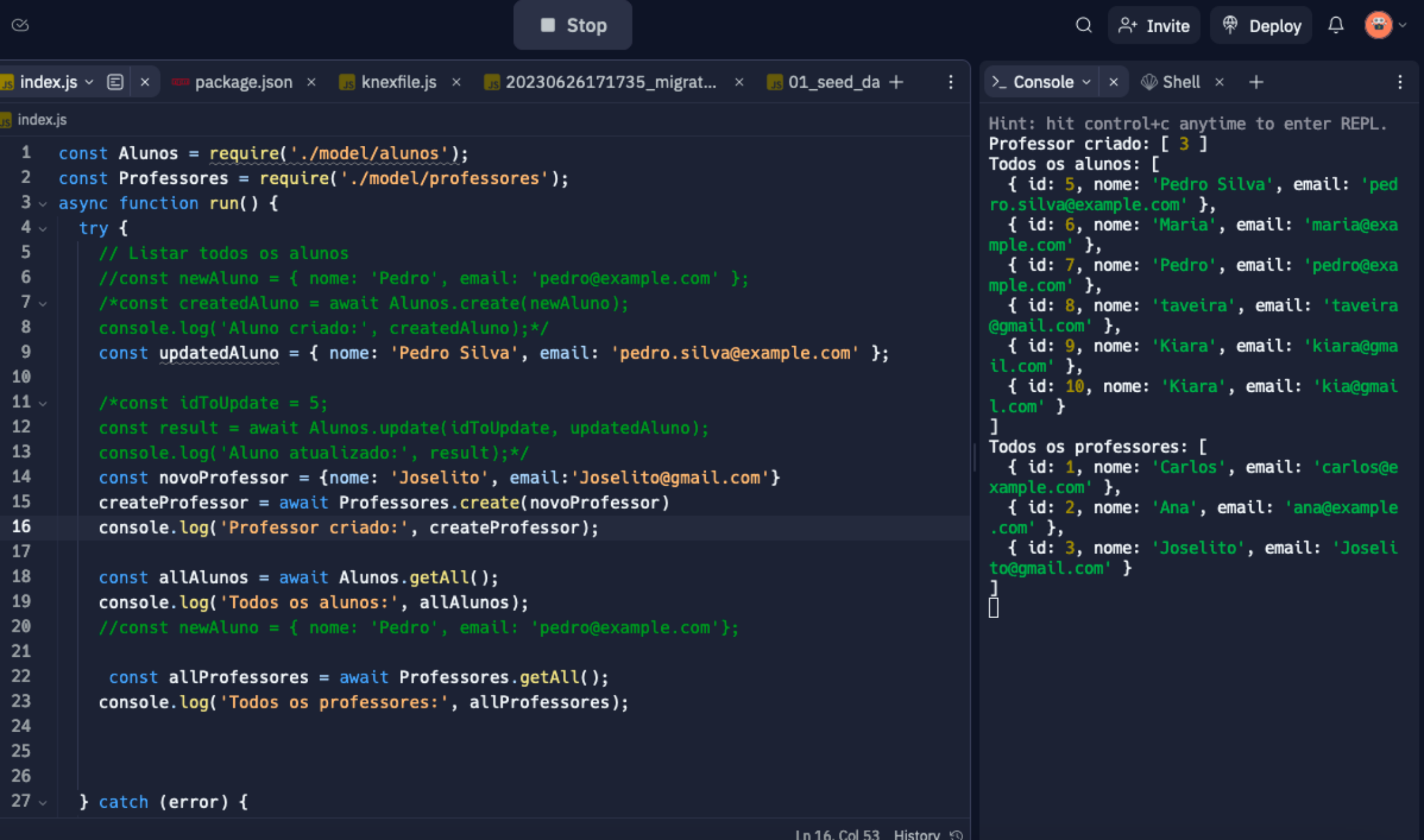
]

```
const Alunos = require('./model/alunos');
const Professores = require('./model/professores');
const Disciplinas = require('./model/disciplinas');
async function run() {
  try {
    // Listar todos os alunos
    /*const newAluno = { nome: 'Pedro', email: 'pedro@example.com' };
    const createdAluno = await Alunos.create(newAluno);
    console.log('Aluno criado:', createdAluno);

    /*
    const updatedAluno = { nome: 'Pedro Silva', email: 'pedro.silva@example.com' };
    const idToUpdate = 5;
    const result = await Alunos.update(idToUpdate, updatedAluno);
    console.log('Aluno atualizado:', result);*/
    const idToDelete = 5;
    const deleteResult = await Alunos.delete(idToDelete);
    console.log('Aluno deletado:', deleteResult);
    const allAlunos = await Alunos.getAll();
    console.log('Todos os alunos:', allAlunos);

  } catch (error) {
    console.error('Erro', error);
  }
}
run();
```

```
Aluno deletado: 1
Todos os alunos: [
  { id: 6, nome: 'Maria', email: 'maria@example.co
m' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.co
m' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.
com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com'
},
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com'
}
]
```



index.js

```
1  const Alunos = require('./model/alunos');
2  const Professores = require('./model/professores');
3  async function run() {
4    try {
5      // Listar todos os alunos
6      //const newAluno = { nome: 'Pedro', email: 'pedro@example.com' };
7      /*const createdAluno = await Alunos.create(newAluno);
8      console.log('Aluno criado:', createdAluno);*/
9      const updatedAluno = { nome: 'Pedro Silva', email: 'pedro.silva@example.com' };
10
11     /*const idToUpdate = 5;
12     const result = await Alunos.update(idToUpdate, updatedAluno);
13     console.log('Aluno atualizado:', result);*/
14     const allAlunos = await Alunos.getAll();
15     console.log('Todos os alunos:', allAlunos);
16
17     const newProfessor = {nome: 'Mario', email:'Mario@gmail.com'}
18     createdProfessor = await Professores.create(newProfessor)
19     console.log('Professor criado:', createdProfessor);
20
21     //const newAluno = { nome: 'Pedro', email: 'pedro@example.com'};
22
23     const allProfessores = await Professores.getAll();
24     console.log('Todos os professores:', allProfessores);
25
26
27
```

Hint: hit control+c anytime to enter REPL.

Todos os alunos: [

- { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
- { id: 6, nome: 'Maria', email: 'maria@example.com' },
- { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
- { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
- { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
- { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }

]

Professor criado: [4]

Todos os professores: [

- { id: 1, nome: 'Carlos', email: 'carlos@example.com' },
- { id: 2, nome: 'Ana', email: 'ana@example.com' },
- { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
- { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }

]

```
6 //const newAluno = { nome: 'Pedro', email: 'pedro@example.com' };
7 /*const createdAluno = await Alunos.create(newAluno);
8 console.log('Aluno criado:', createdAluno);*/
9 const updatedAluno = { nome: 'Pedro Silva', email: 'pedro.silva@example.com' };
10
11 /*
12 const updatedAluno = { nome: 'Pedro Silva', email: 'pedro.silva@example.com' };
13 const idToUpdate = 5;
14 const result = await Alunos.update(idToUpdate, updatedAluno);
15 console.log('Aluno atualizado:', result);*/
16 const allAlunos = await Alunos.getAll();
17 console.log('Todos os alunos:', allAlunos);
18
19 /*const newProfessor = {nome: 'Mario', email:'Mario@gmail.com'}
20 createdProfessor = await Professores.create(newProfessor)
21 console.log('Professor criado:', createdProfessor);*/
22
23 const updatedProfessor = { nome: 'Aline', email: 'Aline@example.com' };
24 const idToUpdate = 1;
25 const result = await Professores.update(idToUpdate, updatedProfessor);
26 console.log('Aluno atualizado:', result);
27
28
29 const allProfessores = await Professores.getAll();
30 console.log('Todos os professores:', allProfessores);
```

Hint: hit control+c anytime to enter REPL.

```
Todos os alunos: [
  { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }
]
Aluno atualizado: 1
Todos os professores: [
  { id: 1, nome: 'Aline', email: 'Aline@example.com' },
  { id: 2, nome: 'Ana', email: 'ana@example.com' },
  { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
  { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }
]
```



```
16 const allAlunos = await Alunos.getAll();
17 console.log('Todos os alunos:', allAlunos);
18
19 /*const newProfessor = {nome: 'Mario', email: 'Mario@gmail.com'}
20 createdProfessor = await Professores.create(newProfessor)
21 console.log('Professor criado:', createdProfessor);
22
23 const updatedProfessor = { nome: 'Aline', email: 'Aline@example.com' };
24 const idToUpdate = 1;
25 const result = await Professores.update(idToUpdate, updatedProfessor);
26 console.log('Aluno atualizado:', result);*/
27
28 const idToDelete = 2;
29 const deleteResult = await Professores.delete(idToDelete);
30 console.log('Professor deletado:', deleteResult);
31
32
33
34 const allProfessores = await Professores.getAll();
35 console.log('Todos os professores:', allProfessores);
36
37
38
39
40
41 } catch (error) {
42 console.error('Erro ao listar:', error);
```

```
Hint: hit control+c anytime to enter REPL.
Todos os alunos: [
  { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }
]
Professor deletado: 1
Todos os professores: [
  { id: 1, nome: 'Aline', email: 'Aline@example.com' },
  { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
  { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }
]
[]
```


.js > f run > ...

```
console.log('Professor criado:', createdProfessor);

const updatedProfessor = { nome: 'Aline', email: 'Aline@example.com' };
const idToUpdate = 1;
const result = await Professores.update(idToUpdate, updatedProfessor);
console.log('Aluno atualizado:', result);

const idToDelete = 2;
const deleteResult = await Professores.delete(idToDelete);
console.log('Professor deletado:', deleteResult);*/

const allProfessores = await Professores.getAll();
console.log('Todos os professores:', allProfessores);

const newDisciplina = {nome:'Física', professor_id:3}
createdDisciplina= await Disciplinas.create(newDisciplina)
console.log('Disciplina criada:', createdDisciplina);

} catch (error) {
  console.error('Erro ao listar:', error);
}
}
run();
```

Hint: hit control+c anytime to enter REPL.

```
Todos os alunos: [
  { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }
]
Todos os professores: [
  { id: 1, nome: 'Aline', email: 'Aline@example.com' },
  { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
  { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }
]
Disciplina criada: [ 3 ]
[]
```

```
index.js x package.json x knexfile.js x 20230626171735_migrat... x 01_seed_da +
index.js > f run > ...
29 const deleteResult = await Professores.delete(idToDelete);
30 console.log('Professor deletado:', deleteResult);*/
31
32 const allProfessores = await Professores.getAll();
33 console.log('Todos os professores:', allProfessores);
34
35
36 const newDisciplina = {nome:'Física', professor_id:3}
37 createdDisciplina= await Disciplinas.create(newDisciplina)
38 console.log('Disciplina criada:', createdDisciplina);
39
40 const allDisciplinas = await Disciplinas.getAll();
41 console.log('Todas as disciplinas:', allDisciplinas);
42
43
44
45
46 } catch (error) {
47   console.error('Erro ao listar:', error);
48 }
49 }
50 run();
51
```

```
>_ Console x Shell x +
Hint: hit control+c anytime to enter REPL.
Todos os alunos: [
  { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }
]
Todos os professores: [
  { id: 1, nome: 'Aline', email: 'Aline@example.com' },
  { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
  { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }
]
Disciplina criada: [ 4 ]
Todas as disciplinas: [
  { id: 1, nome: 'Matemática', professor_id: 1 },
  { id: 2, nome: 'História', professor_id: 2 },
  { id: 3, nome: 'Física', professor_id: 3 },
  { id: 4, nome: 'Física', professor_id: 3 }
]

```

```

30 console.log('Professor deletado:', deleteResult);*/
31
32 const allProfessores = await Professores.getAll();
33 console.log('Todos os professores:', allProfessores);
34
35
36 /*const newDisciplina = {nome:'Física', professor_id:3}
37 createdDisciplina= await Disciplinas.create(newDisciplina)
38 console.log('Disciplina criada:', createdDisciplina);
39
40
41
42 const updatedDisciplina = { nome: 'Ulisses', professor_id: 'Ulisses@example.com' };
43 const idToUpdate = 4;
44 const result = await Disciplinas.update(idToUpdate, updatedDisciplina);
45 console.log('Disciplina atualizada:', result);*/
46
47
48 const idToDelete = 4;
49 const deleteResult = await Disciplinas.delete(idToDelete);
50 console.log('Disciplina deletada:', deleteResult);
51
52 const allDisciplinas = await Disciplinas.getAll();
53 console.log('Todas as disciplinas:', allDisciplinas);
54
55
56
57 } catch (error) {

```

```

Hint: hit control+c anytime to enter REPL.
Todos os alunos: [
  { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
  { id: 6, nome: 'Maria', email: 'maria@example.com' },
  { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
  { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
  { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
  { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }
]
Todos os professores: [
  { id: 1, nome: 'Aline', email: 'Aline@example.com' },
  { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
  { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }
]
Disciplina deletada: 1
Todas as disciplinas: [
  { id: 1, nome: 'Matemática', professor_id: 1 },
  { id: 2, nome: 'História', professor_id: 2 },
  { id: 3, nome: 'Física', professor_id: 3 }
]

```

```

index.js
8 console.log('Disciplina criada:', createDisciplina);
9 const idToDelete = 4;
10 const deleteResult = await Disciplinas.delete(idToDelete);
11 console.log('Disciplina deletada:', deleteResult);*/
12
13
14
15 const updatedDisciplina = { nome: 'Portugues', professor_id:1 };
16 const idToUpdate = 3;
17 const result = await Disciplinas.update(idToUpdate, updatedDisciplina);
18 console.log('Disciplina atualizada:', result);
19
20
21
22 const allDisciplinas = await Disciplinas.getAll();
23 console.log('Todas as disciplinas:', allDisciplinas);
24
25
26
27 } catch (error) {
28 console.error('Erro', error);
29 }
30 }
31 run();
32

```

Hint: hit control+c anytime to enter REPL.

Todos os alunos: [

- { id: 5, nome: 'Pedro Silva', email: 'pedro.silva@example.com' },
- { id: 6, nome: 'Maria', email: 'maria@example.com' },
- { id: 7, nome: 'Pedro', email: 'pedro@example.com' },
- { id: 8, nome: 'taveira', email: 'taveira@gmail.com' },
- { id: 9, nome: 'Kiara', email: 'kiara@gmail.com' },
- { id: 10, nome: 'Kiara', email: 'kia@gmail.com' }

]

Todos os professores: [

- { id: 1, nome: 'Aline', email: 'Aline@example.com' },
- { id: 3, nome: 'Joselito', email: 'Joselito@gmail.com' },
- { id: 4, nome: 'Mario', email: 'Mario@gmail.com' }

]

Disciplina atualizada: 1

Todas as disciplinas: [

- { id: 1, nome: 'Matemática', professor_id: 1 },
- { id: 2, nome: 'História', professor_id: 2 },
- { id: 3, nome: 'Portugues', professor_id: 1 }

]

Files

index.js

migrations

202306261717...

model

alunos.js

disciplinas.js

professores.js

seeds

01_seed_data.js

knexfile.js

Tools

Deployments Ghostwriter Authentication...

Chat

Code Search

Console

Database

Debugger

Docs

Extensions

Git

Markdown

CPU RAM Storage

Try Ghostwriter

migrations > 20230626171735_migration_name.js > f up > f up > f up > ...

```
2  * @param { import("knex").Knex } knex
3  * @returns { Promise<void> }
4  */
5  exports.up = function(knex) {
6    return knex.schema
7      .createTable('tb_alunos', function(table) {
8        table.increments('id').primary();
9        table.string('nome').nullable();
10       table.string('email').nullable().unique();
11      })
12      .createTable('tb_professores', function(table) {
13        table.increments('id').primary();
14        table.string('nome').nullable();
15        table.string('email').nullable().unique();
16      })
17      .createTable('tb_disciplinas', function(table) {
18        table.increments('id').primary();
19        table.string('nome').nullable();
20        table.integer('professor_id').unsigned().references('id').inTable('tb_professores');
21      });
22  };
23
24  /**
25   * @param { import("knex").Knex } knex
26   * @returns { Promise<void> }
27   */
28  exports.down = function(knex) {
29    return knex.schema
30      .dropTableIfExists('tb_disciplinas')
31      .dropTableIfExists('tb_alunos')
32      .dropTableIfExists('tb_professores');
```

```
4  */
5  exports.seed = async function(knex) {
6    return knex('tb_alunos').del()
7      .then(function () {
8        // Inserts seed entries
9        return knex('tb_alunos').insert([
10          { nome: 'João', email: 'joao@example.com' },
11          { nome: 'Maria', email: 'maria@example.com' },
12        ]);
13      })
14      .then(function () {
15        return knex('tb_professores').del()
16          .then(function () {
17            return knex('tb_professores').insert([
18              { nome: 'Carlos', email: 'carlos@example.com' },
19              { nome: 'Ana', email: 'ana@example.com' },
20            ]);
21          });
22      })
23      .then(function () {
24        return knex('tb_disciplinas').del()
25          .then(function () {
26            return knex('tb_disciplinas').insert([
27              { nome: 'Matemática', professor_id: 1 },
28              { nome: 'História', professor_id: 3 },
29            ]);
30          });
31      });
32  };
3
```

```
const knex = require('knex');
```

```
const db = knex({  
  client: 'sqlite3',  
  connection: {  
    filename: './SCA.db'  
  },  
  useNullAsDefault: true  
});
```

```
const Alunos = {  
  getAll: async () => {  
    return db('tb_alunos');  
  },  
  
  getById: async (id) => {  
    return db('tb_alunos').where({ id }).first();  
  },  
  
  create: async (aluno) => {  
    return db('tb_alunos').insert(aluno);  
  },  
  
  update: async (id, aluno) => {  
    return db('tb_alunos').where({ id }).update(aluno);  
  },  
  
  delete: async (id) => {  
    return db('tb_alunos').where({ id }).del();  
  }  
};
```



```
1  const knex = require('knex');
2  const database = knex({
3    client: 'sqlite3',
4    connection: {
5      filename: './SCA.db'
6    },
7    useNullAsDefault: true
8  });
9
10 const Disciplinas = {
11   getAll: async () => {
12     return database('tb_disciplinas');
13   },
14
15   getById: async (id) => {
16     return database('tb_disciplinas').where({ id }).first();
17   },
18
19   create: async (disciplina) => {
20     return database('tb_disciplinas').insert(disciplina);
21   },
22
23   update: async (id, disciplina) => {
24     return database('tb_disciplinas').where({ id }).update(disciplina);
25   },
26
27   delete: async (id) => {
28     return database('tb_disciplinas').where({ id }).del();
29   }
30 };
31
32 module.exports = Disciplinas;
```

```
const database = knex({
  client: 'sqlite3',
  connection: {
    filename: './SCA.db'
  },
  useNullAsDefault: true
});
```

```
const Professores = {
  getAll: async () => {
    return database('tb_professores');
  },

  getById: async (id) => {
    return database('tb_professores').where({ id }).first();
  },
```

```
  create: async (professor) => {
    return database('tb_professores').insert(professor);
  },
```

```
  update: async (id, professor) => {
    return database('tb_professores').where({ id }).update(professor);
  },
```

```
  delete: async (id) => {
    return database('tb_professores').where({ id }).del();
  }
}
```

```
};
```

```
module.exports = Professores;
```