

```
import mysql.connector
```

```
# Função para conectar ao banco de dados
```

```
def connect():  
    connection = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        password="1234",  
        database="projeto1"  
    )  
    return connection
```

```
# Função para cadastrar um novo Aluno
```

```
def cadastrar_aluno(cursor):  
    nome = input("Digite o nome do Aluno: ")  
    idade = input("Digite a idade do Aluno: ")  
    modalidade_id = input("Digite o ID da Modalidade (ou deixe em branco): ")  
  
    query = "INSERT INTO Aluno (nome, idade) VALUES (%s, %s)"  
    values = (nome, idade)  
    cursor.execute(query, values)  
    connection.commit()  
  
    aluno_id = cursor.lastrowid  
  
    if modalidade_id != "":  
        query = "INSERT INTO Aluno_Modalidade (aluno_id, modalidade_id) VALUES (%s, %s)"  
        values = (aluno_id, modalidade_id)  
        cursor.execute(query, values)  
        connection.commit()  
  
    print("Aluno cadastrado com sucesso!")
```

```
# Função para cadastrar uma nova Modalidade
```

```
def cadastrar_modalidade(cursor):  
    nome = input("Digite o nome da Modalidade: ")  
    query = "INSERT INTO Modalidade (nome) VALUES (%s)"  
    values = (nome,)  
    cursor.execute(query, values)  
    connection.commit()  
    print("Modalidade cadastrada com sucesso!")
```

```
# Função para cadastrar um novo Professor
```

```
def cadastrar_professor(cursor):  
    nome = input("Digite o nome do Professor: ")  
    modalidade_id = input("Digite o ID da Modalidade para vincular ao Professor: ")  
    query = "INSERT INTO Professor (nome) VALUES (%s)"  
    values = (nome,)  
    cursor.execute(query, values)
```

```

modalidade_id = input("Digite o ID da Modalidade para vincular ao Professor: ")
query = "INSERT INTO Professor (nome) VALUES (%s)"
values = [(nome,)]
cursor.execute(query, values)
professor_id = cursor.lastrowid
query = "INSERT INTO Professor_Modalidade (professor_id, modalidade_id) VALUES (%s, %s)"
values = (professor_id, modalidade_id)
cursor.execute(query, values)
connection.commit()
print("Professor cadastrado com sucesso!")

```

Função para editar um Aluno

```

def editar_aluno(cursor):
    aluno_id = input("Digite o ID do Aluno que deseja editar: ")

    query = "SELECT * FROM Aluno WHERE id = %s"
    values = (aluno_id,)
    cursor.execute(query, values)
    aluno = cursor.fetchone()

    if aluno:
        nome = input("Digite o novo nome do Aluno (ou deixe em branco para manter o mesmo): ")
        idade = input("Digite a nova idade do Aluno (ou deixe em branco para manter a mesma): ")
        modalidade_id = input("Digite o novo ID da Modalidade (ou deixe em branco para manter o mesmo): ")

        if nome == "":
            nome = aluno[1]
        if idade == "":
            idade = aluno[2]

        query = "UPDATE Aluno SET nome = %s, idade = %s WHERE id = %s"
        values = (nome, idade, aluno_id)
        cursor.execute(query, values)
        connection.commit()

        if modalidade_id != "":
            query = "INSERT INTO Aluno_Modalidade (aluno_id, modalidade_id) VALUES (%s, %s) ON DUPLICATE KEY UPDATE modalidade_id = %s"
            values = (aluno_id, modalidade_id, modalidade_id)
            cursor.execute(query, values)
            connection.commit()

        print("Aluno atualizado com sucesso!")
    else:
        print("Aluno não encontrado.")

```

Função para excluir um Aluno

```

def excluir_aluno(cursor):
    aluno_id = input("Digite o ID do Aluno que deseja excluir: ")

```

```

else:
    print("Aluno não encontrado.")

# Função para excluir um Aluno
def excluir_aluno(cursor):
    aluno_id = input("Digite o ID do Aluno que deseja excluir: ")

    query = "DELETE FROM Aluno WHERE id = %s"
    values = (aluno_id,)
    cursor.execute(query, values)
    connection.commit()

    print("Aluno excluído com sucesso!")

# Função para buscar Alunos por nome
def buscar_alunos(cursor):
    nome = input("Digite o nome do Aluno que deseja buscar: ")

    query = "SELECT * FROM Aluno WHERE nome LIKE %s"
    values = (f"%{nome}%",)
    cursor.execute(query, values)
    result = cursor.fetchall()

    print("==== Resultado da Busca =====")
    if result:
        for row in result:
            print("ID:", row[0])
            print("Nome:", row[1])
            print("Idade:", row[2])
            print("-----")
    else:
        print("Nenhum resultado encontrado.")

# Função para buscar Professores por nome
def buscar_professores(cursor):
    nome = input("Digite o nome do Professor que deseja buscar: ")

    query = "SELECT * FROM Professor WHERE nome LIKE %s"
    values = (f"%{nome}%",)
    cursor.execute(query, values)
    result = cursor.fetchall()

    print("==== Resultado da Busca =====")
    if result:
        for row in result:
            print("ID:", row[0])
            print("Nome:", row[1])
            print("-----")
    else:

```

```

result = cursor.fetchall()

print("==== Resultado da Busca =====")
if result:
    for row in result:
        print("ID:", row[0])
        print("Nome:", row[1])
        print("-----")
else:
    print("Nenhum resultado encontrado.")

# Função para buscar Modalidades por nome
def buscar_modalidades(cursor):
    nome = input("Digite o nome da Modalidade que deseja buscar: ")

    query = "SELECT * FROM Modalidade WHERE nome LIKE %s"
    values = (f"%{nome}%",)
    cursor.execute(query, values)
    result = cursor.fetchall()

    print("==== Resultado da Busca =====")
    if result:
        for row in result:
            print("ID:", row[0])
            print("Nome:", row[1])
            print("-----")
    else:
        print("Nenhum resultado encontrado.")

# Conecta ao banco de dados
connection = connect()
cursor = connection.cursor()

# Loop do programa
while True:
    print("==== Menu =====")
    print("1. Cadastrar Aluno")
    print("2. Cadastrar Modalidade")
    print("3. Cadastrar Professor")
    print("4. Editar Aluno")
    print("5. Excluir Aluno")
    print("6. Buscar Alunos por nome")
    print("7. Buscar Professores por nome")
    print("8. Buscar Modalidades por nome")
    print("9. Relatório de Alunos por Modalidade")
    print("10. Relatório de Professores por Modalidade")
    print("0. Sair")

```

```
print("10. Relatório de Professores por Modalidade")
print("0. Sair")

opcao = input("Selecione uma opção: ")

if opcao == "1":
    cadastrar_aluno(cursor)
elif opcao == "2":
    cadastrar_modalidade(cursor)
elif opcao == "3":
    cadastrar_professor(cursor)
elif opcao == "4":
    editar_aluno(cursor)
elif opcao == "5":
    excluir_aluno(cursor)
elif opcao == "6":
    buscar_alunos(cursor)
elif opcao == "7":
    buscar_professores(cursor)
elif opcao == "8":
    buscar_modalidades(cursor)
elif opcao == "9":
    relatorio_alunos_por_modalidade(cursor)
elif opcao == "10":
    relatorio_professores_por_modalidade(cursor)
elif opcao == "0":
    break
else:
    print("Opção inválida! Por favor, escolha uma opção válida.")

# Fecha a conexão com o banco de dados
connection.close()
```

Users > Pichau > Downloads > ≡ projetop1.sql > ...

```
-- phpMyAdmin SQL Dump
```

```
-- version 5.1.1deb5ubuntu1
```

```
-- https://www.phpmyadmin.net/
```

```
--
```

```
-- Host: localhost:3306
```

```
-- Tempo de geração: 08/06/2023 às 22:30
```

```
-- Versão do servidor: 8.0.33-0ubuntu0.22.04.2
```

```
-- Versão do PHP: 8.1.2-1ubuntu2.11
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```

```
/*!40101 SET NAMES utf8mb4 */;
```

```
--
```

```
-- Banco de dados: `projetop1`
```

```
--
```

```
-- -----
```

```
--
```

```
-- Estrutura para tabela `Aluno`
```

```
--
```

```
CREATE TABLE `Aluno` (
```

```
  `id` int NOT NULL,
```

```
  `nome` varchar(100) NOT NULL,
```

```
  `idade` int DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
--
```

```
-- Despejando dados para a tabela `Aluno`
```

```
--
```

```
INSERT INTO `Aluno` (`id`, `nome`, `idade`) VALUES
```

```
(1, 'Maria Jotontina', 23),
```

```
(2, 'Roseane Dore', 88),
```

```
(3, 'Sergio Mario', 17),
```

```
(4, 'Tiaga Manoela', 14),
```

```
(5, 'Kiko Tesouro', 28),
```

```
(6, 'Veronica da Silva', 12),
```

```
(7, 'Rolando Lero', 23),
```

```
(8, 'Aldemar Vigario', 12),
```

```
(9, 'Raimundo Nonato', 21)
```

```

(8, 'Aldemar Vigarão', 12),
(9, 'Raimundo Nonato', 21),
(10, 'Nerso da Capetinga', 21),
(16, 'Marcos Mion', 35);

-- -----

--
-- Estrutura para tabela `Aluno_Modalidade`
--

CREATE TABLE `Aluno_Modalidade` (
  `aluno_id` int NOT NULL,
  `modalidade_id` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `Aluno_Modalidade`
--

INSERT INTO `Aluno_Modalidade` (`aluno_id`, `modalidade_id`) VALUES
(6, 1),
(9, 1),
(2, 2),
(3, 2),
(5, 2),
(6, 2),
(8, 2),
(2, 3),
(3, 5),
(2, 9);

-- -----

--
-- Estrutura para tabela `Modalidade`
--

CREATE TABLE `Modalidade` (
  `id` int NOT NULL,
  `nome` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `Modalidade`
--

INSERT INTO `Modalidade` (`id`, `nome`) VALUES
(1, 'Futebol'),

```



```

(1, 'Futebol'),
(2, 'Vôlei'),
(3, 'Surfe'),
(4, 'Handebol'),
(5, 'Boxe'),
(6, 'Kicking Boxe'),
(7, 'Sumô'),
(8, 'Yoga'),
(9, 'Basquete'),
(10, 'Musculação'),
(16, 'Corrida'),
(17, 'Parkour');

-----

--
-- Estrutura para tabela `Professor`
--

CREATE TABLE `Professor` (
  `id` int NOT NULL,
  `nome` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `Professor`
--

INSERT INTO `Professor` (`id`, `nome`) VALUES
(1, 'Rogerio Balboa'),
(2, 'Maguila do Valentina'),
(3, 'Cinziane Barbosa'),
(4, 'Rubao Barrichelo'),
(5, 'Mario Baloteli'),
(6, 'Juninho Baiano'),
(7, 'Joel Santana'),
(8, 'Coach de Mentirinha'),
(9, 'Bruna Surfistinha'),
(10, 'Seu Madruga'),
(16, 'Leao do Ballet'),
(17, 'Marcos'),
(19, 'Kill bill');

-----

--
-- Estrutura para tabela `Professor_Modalidade`
--

```



```
--
-- Estrutura para tabela `Professor_Modalidade`
--

CREATE TABLE `Professor_Modalidade` (
  `professor_id` int NOT NULL,
  `modalidade_id` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `Professor_Modalidade`
--

INSERT INTO `Professor_Modalidade` (`professor_id`, `modalidade_id`) VALUES
(5, 1),
(7, 1),
(10, 1),
(1, 2),
(4, 2),
(6, 2),
(8, 3),
(19, 6);

-----

--
-- Estrutura para tabela `Treino`
--

CREATE TABLE `Treino` (
  `id` int NOT NULL,
  `aluno_id` int DEFAULT NULL,
  `professor_id` int DEFAULT NULL,
  `descricao` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `Treino`
--

INSERT INTO `Treino` (`id`, `aluno_id`, `professor_id`, `descricao`) VALUES
(1, 1, 4, 'Descrição do treino 1'),
(2, 6, 9, 'Descrição do treino 2'),
(3, 7, 8, 'Descrição do treino 3'),
(4, 6, 6, 'Descrição do treino 4'),
(5, 10, 3, 'Descrição do treino 5'),
(6, 5, 3, 'Descrição do treino 6'),
(7, 3, 4, 'Descrição do treino 7');
```

```
-- Despejando dados para a tabela `Treino`
```

```
INSERT INTO `Treino` (`id`, `aluno_id`, `professor_id`, `descricao`) VALUES
(1, 1, 4, 'Descrição do treino 1'),
(2, 6, 9, 'Descrição do treino 2'),
(3, 7, 8, 'Descrição do treino 3'),
(4, 6, 6, 'Descrição do treino 4'),
(5, 10, 3, 'Descrição do treino 5'),
(6, 5, 3, 'Descrição do treino 6'),
(7, 3, 4, 'Descrição do treino 7'),
(8, 10, 10, 'Descrição do treino 8'),
(9, 7, 7, 'Descrição do treino 9'),
(10, 1, 6, 'Descrição do treino 10');
```

```
-- Índices para tabelas despejadas
```

```
-- Índices de tabela `Aluno`
```

```
ALTER TABLE `Aluno`
| ADD PRIMARY KEY (`id`);
```

```
-- Índices de tabela `Aluno_Modalidade`
```

```
ALTER TABLE `Aluno_Modalidade`
| ADD PRIMARY KEY (`aluno_id`,`modalidade_id`),
| ADD KEY `modalidade_id` (`modalidade_id`);
```

```
-- Índices de tabela `Modalidade`
```

```
ALTER TABLE `Modalidade`
| ADD PRIMARY KEY (`id`);
```

```
-- Índices de tabela `Professor`
```

```
ALTER TABLE `Professor`
| ADD PRIMARY KEY (`id`);
```

```
-- Índices de tabela `Professor_Modalidade`
```

```
ALTER TABLE `Professor_Modalidade`
```

```
--
-- Índices de tabela `Professor Modalidade`
--
ALTER TABLE `Professor Modalidade`
  ADD PRIMARY KEY (`professor_id`,`modalidade_id`),
  ADD KEY `modalidade_id` (`modalidade_id`);

--
-- Índices de tabela `Treino`
--
ALTER TABLE `Treino`
  ADD PRIMARY KEY (`id`),
  ADD KEY `aluno_id` (`aluno_id`),
  ADD KEY `professor_id` (`professor_id`);

--
-- AUTO_INCREMENT para tabelas despejadas
--

--
-- AUTO_INCREMENT de tabela `Aluno`
--
ALTER TABLE `Aluno`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

--
-- AUTO_INCREMENT de tabela `Modalidade`
--
ALTER TABLE `Modalidade`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;

--
-- AUTO_INCREMENT de tabela `Professor`
--
ALTER TABLE `Professor`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;

--
-- AUTO_INCREMENT de tabela `Treino`
--
ALTER TABLE `Treino`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- Restrições para tabelas despejadas
--

--
-- Restrições para tabelas `Aluno Modalidade`
```

```

--
-- Restrições para tabelas despejadas
--
--
-- Restrições para tabelas `Aluno_Modalidade`
--
ALTER TABLE `Aluno_Modalidade`
  ADD CONSTRAINT `Aluno_Modalidade_ibfk_1` FOREIGN KEY (`aluno_id`) REFERENCES `Aluno` (`id`),
  ADD CONSTRAINT `Aluno_Modalidade_ibfk_2` FOREIGN KEY (`modalidade_id`) REFERENCES `Modalidade` (`id`);
--
-- Restrições para tabelas `Professor_Modalidade`
--
ALTER TABLE `Professor_Modalidade`
  ADD CONSTRAINT `Professor_Modalidade_ibfk_1` FOREIGN KEY (`professor_id`) REFERENCES `Professor` (`id`),
  ADD CONSTRAINT `Professor_Modalidade_ibfk_2` FOREIGN KEY (`modalidade_id`) REFERENCES `Modalidade` (`id`);
--
-- Restrições para tabelas `Treino`
--
ALTER TABLE `Treino`
  ADD CONSTRAINT `Treino_ibfk_1` FOREIGN KEY (`aluno_id`) REFERENCES `Aluno` (`id`),
  ADD CONSTRAINT `Treino_ibfk_2` FOREIGN KEY (`professor_id`) REFERENCES `Professor` (`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

```
SELECT * FROM treino BETWEEN 4 AND 9;
```

```
SELECT * FROM aluno.modalidade WHERE treino BETWEEN 5 AND 10;
```

```
SELECT * FROM aluno = 'Aldemar';
```



```
SELECT * FROM modalidade_id = 1;
```

```
SELECT * FROM aluno;
```

```
SELECT * FROM modalidade;
```

```
SELECT * FROM aluno.modalidade ORDER BY aluno.idade;
```

```
SELECT * FROM aluno.modalidade;
```

```
SELECT * FROM aluno.modalidade ORDER BY modalidade_id;
```

```
SELECT * FROM modalidade WHERE info_modalidade = 'Surfe' OR modalidade_id = 3;
```

```
SELECT * FROM modalidade WHERE info_modalidade = 'Surfe';
```

```
SELECT * FROM aluno;
SELECT * FROM modalidade;

SELECT * FROM aluno.modalidade ORDER BY aluno.idade;
```

```
SELECT * FROM professor;
```

```
SELECT * FROM professor_id ORDER BY nome;
```



```
SELECT * FROM professor;
```

```
SELECT * FROM professor_id ORDER BY id;
```

```
|
```

```
SELECT * FROM professor;
```

```
SELECT aluno, professor, treino FROM modalidade;
```

```
SELECT * FROM treino BETWEEN 4 AND 9
```

```
SELECT * FROM aluno.modalidade WHERE treino BETWEEN 5 AND 10
```