

MENTORAMA – ESCOLA ONLINE

PROJETO FINAL – LINGUAGEM SQL

MODELO CONCEITUAL

Resumo

A Escola de Artes Marciais "Mentorama Martial Arts", mais conhecida como Mentorama MA, está necessitando mudar seu banco de dados, que atualmente é manual, para um sistema digital. Para isso, contratou um desenvolvedor de banco de dados da Escola Online Mentorama, que acabou de completar seu curso e precisa de oportunidade para começar a sua experiência profissional.

O sistema da Escola de Artes Marciais é para cadastro de funcionários, professores, alunos e sistema de pagamento de mensalidade para os alunos, sendo também de consulta para estes. O sistema financeiro para pagamento dos funcionários será desenvolvido posteriormente. Note que a forma de cadastro no sistema é manual, ou seja, um funcionário da administração é responsável por realizar este cadastro manualmente, um por um, conforme forem chegando mais alunos ou outros funcionários e professores contratados.

A Mentorama MA possui três modalidades de esporte: jiu jitsu, muay thai e MMA (mixed martial arts), esporte que mistura várias artes marciais em apenas uma só, mundialmente conhecido através da empresa UFC, transmitida pelo Canal Combate aqui no Brasil.

As modalidades jiu jitsu e muay thai terão suas tabelas no banco de dados, com os identificadores do cadastro do aluno na turma, identificador do cadastro do aluno na Escola, nomes dos alunos que praticam, a faixa que este praticante é (o nível de proficiência na arte marcial), data de início na turma (para fins de critério para passar de faixa), o plano de assinatura na Escola e o identificador do professor responsável pela turma a qual o aluno faz parte. Já a modalidade de MMA terão os mesmos atributos, porém no lugar da faixa será se é ou não profissional.

Nossa escola então possui o cadastro de alunos e funcionários (podendo ser professores, funcionários administrativos ou funcionários da manutenção da Escola).

As aulas são fornecidas semanalmente, em diversas turmas diferentes, e os planos ou pacotes a serem adquiridos são pacotes mensais, semestrais ou anuais, sendo que os planos mensais devem ser pagos mensalmente, porém o semestral e anual podem ser parcelados em até quatro vezes.

Desta forma, o aluno tem acesso a um portal financeiro, seja pela internet ou pelo celular, por meio do qual é possível saber qual pacote adquiriu, a parcela, o valor e a data de vencimento e pagamento, dentro do ano atual, ou seja, 2022. Os dados financeiros dos demais anos, somente a administração da Escola tem acesso.

Os alunos podem ser cadastrados com os seguintes dados: nome, RG, CPF, data de nascimento, telefone, endereço e cidade. Os alunos terão um número identificador. Não é obrigatório RG, CPF e telefone, pois podemos ter alunos menores de idade que não possuem esse cadastro.

Os professores podem ser cadastrados com os mesmos dados que os alunos, porém como todos são maiores de idade, RG e CPF são obrigatórios. Também terão seus números identificadores.

O sistema de pagamento possui o identificador do pagamento realizado, o identificador do plano ou pacote escolhido pelo aluno, a parcela deste pagamento (se for escolhido pacote que divide as parcelas de pagamento), o valor a ser pago, o identificador do aluno que está realizando o pagamento, a data de vencimento do pagamento e a data que o pagamento foi realizado.

Diagrama entidade-relacionamento

Vamos desta forma, detalhar as entidades do nosso sistema. Temos as entidades das modalidades de artes marciais. Desta forma, teremos três tabelas no nosso sistema para cada modalidade. Estas tabelas se relacionam com a tabela 'aluno' na cardinalidade de 1:1, tanto para a tabela 'aluno', quanto para a tabela de 'jiu_jitsu', por exemplo. Ou seja, cada aluno se relaciona com as tabelas de modalidade uma única vez e vice-versa. Afinal, não tem finalidade cadastrar o mesmo aluno mais de uma vez na mesma modalidade e vice-versa.

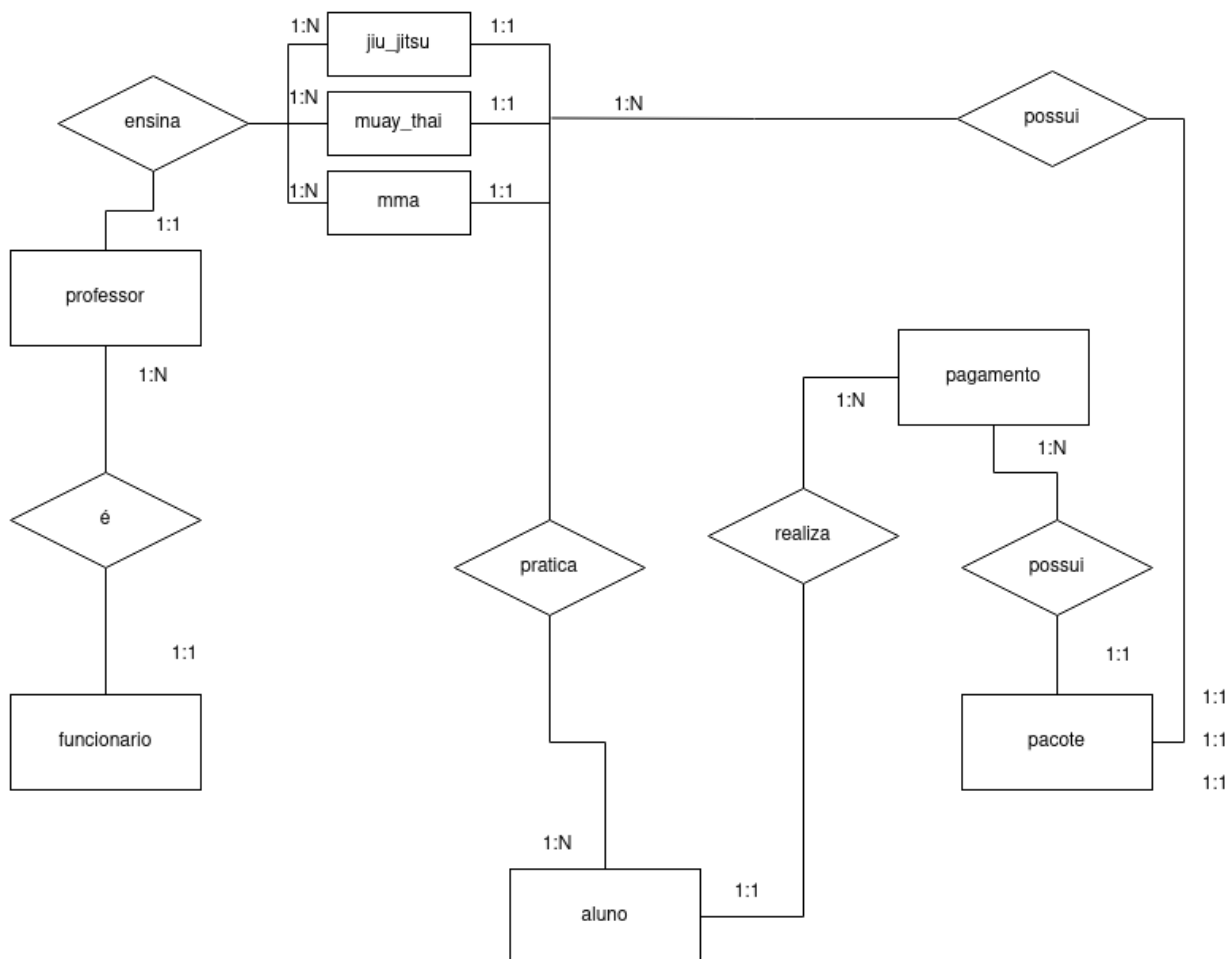
Sendo assim, um aluno da tabela 'aluno' corresponde a apenas um aluno cadastrado na tabela de cada modalidade. O mesmo aluno não aparece duas vezes na tabela 'jiu_jitsu', não faria sentido, já que representam a mesma pessoa praticando o mesmo esporte.

Ainda sobre as tabelas das modalidades, temos o relacionamento com a tabela de professores. Aqui, a cardinalidade do relacionamento não é a mesma para as duas entidades. Cada entidade da tabela 'jiu_jitsu', por exemplo, se relaciona com apenas uma entidade da tabela 'professor', porém, cada professor se relaciona com várias entidades da tabela 'jiu_jitsu'. Sendo assim, a cardinalidade da tabela de cada modalidade para a tabela 'professor' é 1:1, porém da tabela 'professor' para cada modalidade é 1:N.

Sobre a tabela 'professor' e a tabela 'funcionario' é o mesmo raciocínio anterior. Cada entidade da tabela 'funcionario' pode se relacionar com mais de uma entidade da tabela 'professor', mas o inverso não é verdadeiro. Cada entidade da tabela 'professor' se relaciona com apenas uma entidade da tabela 'funcionario'. Sendo assim, da tabela 'professor' para 'funcionario' é de 1:1, porém o inverso é de 1:N. O mesmo raciocínio se aplica para as tabelas das modalidades e a tabela 'pacote'. Cada entidade da tabela 'pacote' pode se relacionar com várias entidades das tabelas de

modalidades, sendo 1:N. Porém o inverso é 1:1, ou seja, cada entidade da tabela 'jiu_jitsu', por exemplo, se relaciona com apenas uma entidade da tabela 'pacote'.

As tabelas 'pagamento' e 'pacote' se relacionam de forma similar a relação anterior. Uma entidade da tabela 'pagamento' possui apenas um relacionamento com uma entidade da tabela 'pacote'. Porém, uma entidade da tabela 'pacote' se relaciona com várias entidades da tabela pagamento. O mesmo raciocínio das tabelas 'aluno' e 'pagamento'. De 'aluno' para 'pagamento', 1:N. De 'pagamento' para 'aluno', 1:1.



MODELO LÓGICO

O modelo lógico do nosso banco de dados contém um plano de tabelas e um diagrama entidade-relacionamento do tipo lógico. Estes diagramas e tabelas contém todas as entidades/tabelas com os atributos e suas restrições/constraints.

jiu_jitsu					
id turma	id_aluno	id_professor	faixa	data_inicio	plano
INT	INT	INT	VARCHAR(20)	DATE	CHAR
AUTO_INCREMENT	FOREIGN KEY	FOREIGN KEY	NOT NULL	NOT NULL	NOT NULL
PRIMARY KEY	NOT NULL	NOT NULL			
NOT NULL					

muay thai					
id_turma	id_aluno	id_professor	faixa	data_inicio	plano
INT	INT	INT	VARCHAR(20)	DATE	CHAR
AUTO_INCREMENT	FOREIGN KEY	FOREIGN KEY	NOT NULL	NOT NULL	NOT NULL
PRIMARY KEY	NOT NULL	NOT NULL			
NOT NULL					

mma						
id_turma	id_aluno	nome	categoria	data_inicio	plano	id_professor
INT	INT	VARCHAR(100)	VARCHAR(20)	DATE	CHAR	INT
AUTO_INCREMENT	FOREIGN KEY	NOT NULL	NOT NULL	NOT NULL	NOT NULL	FOREIGN KEY
PRIMARY KEY	NOT NULL					NOT NULL
NOT NULL						

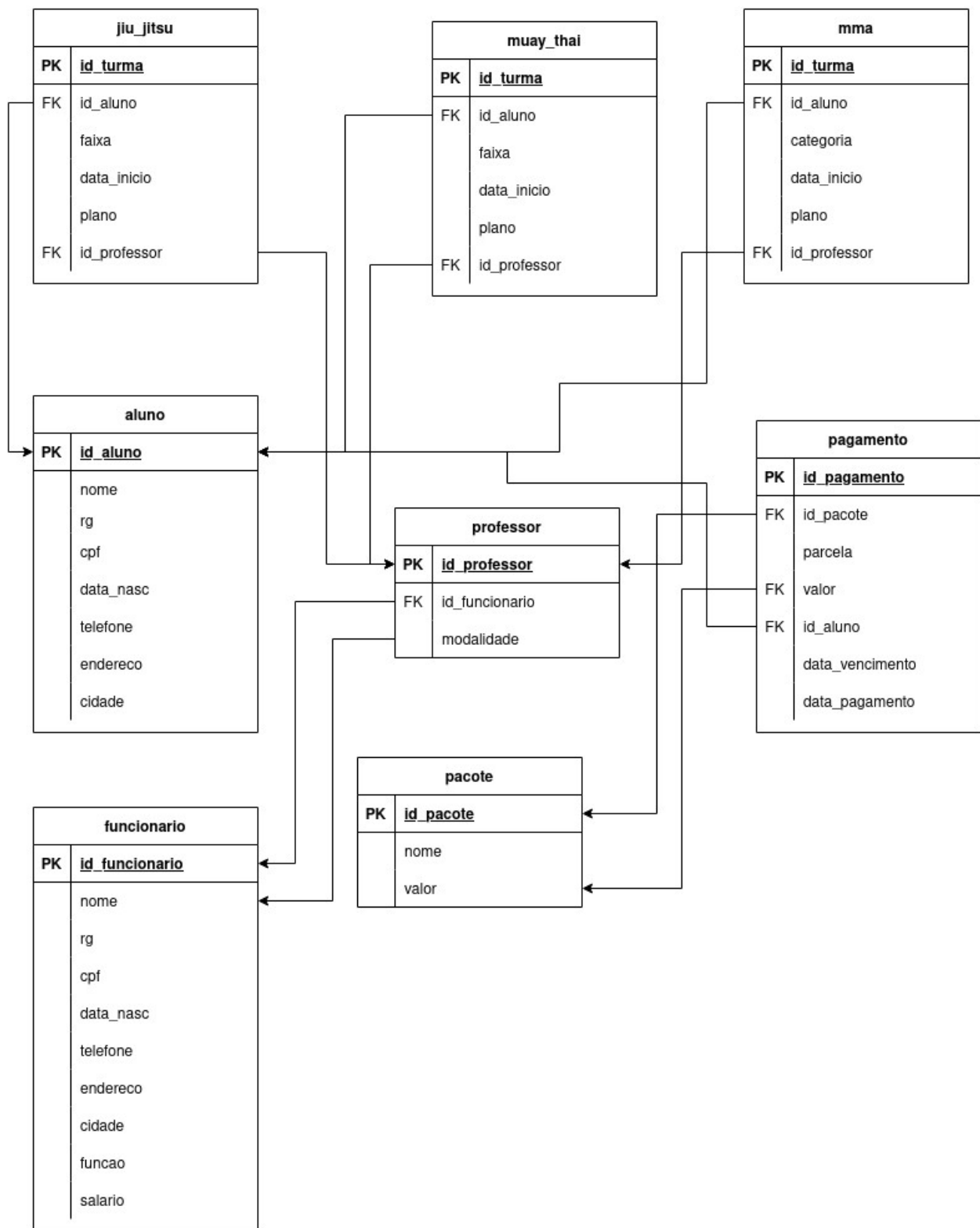
aluno							
id_aluno	nome	rg	cfp	data_nasc	telefone	endereco	cidade
INT	VARCHAR(100)	BIGINT	BIGINT	DATE	VARCHAR(20)	VARCHAR(100)	VARCHAR(50)
AUTO_INCREMENT	NOT NULL	NULL	NULL	NOT NULL	NULL	NOT NULL	NOT NULL
PRIMARY KEY							
NOT NULL							

funcionario									
id_funcionario	nome	rg	cfp	data_nasc	telefone	endereco	cidade	funcao	salario
INT	VARCHAR(100)	BIGINT	BIGINT	DATE	VARCHAR(20)	VARCHAR(100)	VARCHAR(50)	VARCHAR(20)	DECIMAL(6,2)
AUTO_INCREMENT	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL
PRIMARY KEY									
NOT NULL									

pacote				
id_pacote	nome	valor	qtd_parcela	total
INT	VARCHAR(20)	INT	INT	INT
PRIMARY KEY	NOT NULL	NOT NULL	NOT NULL	NOT NULL
NOT NULL		UNIQUE		UNIQUE
AUTO_INCREMENT				

professor		
id_professor	id_funcionario	modalidade
INT	INT	VARCHAR(20)
AUTO_INCREMENT	FOREIGN KEY	NOT NULL
PRIMARY KEY	NOT NULL	
NOT NULL		

pagamento						
id_pagamento	id_pacote	parcela	valor	id_aluno	data_venc	data_pag
INT	INT	INT	INT	INT	DATE	DATE
PRIMARY KEY	FOREIGN KEY	NOT NULL	FOREIGN KEY	FOREIGN KEY	NOT NULL	NULL
NOT NULL	NOT NULL		NOT NULL	NOT NULL		
AUTO_INCREMENT						



MODELO FÍSICO

Podemos então, codificar a estrutura do banco de dados com base nos modelos conceitual e lógico apresentados anteriormente, anexo a documentação em arquivo SQL.

Podemos também esclarecer que o software de gerenciamento de banco de dados a ser utilizado pelo nosso sistema da Mentorama Martial Arts seria o PostgreSQL. Como analisado em aula, o PostgreSQL possui algumas vantagens e desvantagens, como todos analisados, porém uma vantagem merece destaque para a nossa aplicação. O

PostgreSQL apresenta grande apoio da comunidade que o utiliza. Desta forma, suas atualizações de funcionalidades e segurança são mais rápidas a serem desenvolvidas do outros softwares, que apresentam soluções comerciais em seus projetos. O PostgreSQL apresenta outras vantagens em relação aos demais, como maior quantidade de funcionalidades avançadas e maior robustez, capacidade para bancos de dados maiores, etc. Porém, para o objetivo de nossa Academia de Artes Marciais, apenas a vantagem de atualizações do software mais recorrentes para garantir um sistema sempre mais atual e seguro já compensa dar prioridade. Podemos verificar estas diferenças em artigo publicado na comunidade Devmedia (<https://www.devmedia.com.br/postgresql-x-mysql-qual-escolher/3923>), acessado em 01JUL22.

Uma forma de exemplificar como o PostgreSQL pode ter mais funções avançadas ou mais atuais, é a possibilidade de utilizar funções de segurança em nível de linha, ou ‘row level security’. SGBD como o SQL Server e o PostgreSQL possuem essas funções embutidas, como pode-se ver em vários artigos na internet (<https://www.postgresql.org/docs/current/ddl-rowsecurity.html> e <https://docs.microsoft.com/pt-br/sql/relational-databases/security/row-level-security?view=sql-server-ver16>, por exemplo, ambos acessados em 02JUL22). Com o SGBD MySQL teríamos que configurar ‘views’ específicas para restringir este acesso, como veremos no arquivo SQL de definição do banco de dados da Mentorama MA.

CONSULTA E ANÁLISE

“Os dados não agregam valor somente parados e armazenados no banco de dados. É importante sempre ter em mente quais serão as principais **perguntas** a serem respondidas a partir dos dados, seja por meio de um sistema ou interface conectada a esse banco de dados ou por uma conexão direta de alguém que queira fazer uma análise.”

A principal razão de qualquer banco de dados, seja este informatizado ou não, é agregar valor ao negócio em questão, ou seja, que estes dados gerem informações úteis para tomadas de decisão e melhor gerenciamento daquilo que foi ou está sendo realizado pela empresa, seja pública ou privada.

Para a nossa Escola de Artes Marciais podemos destacar entre várias aplicações úteis, necessárias ou importantes para tomadas de decisão, três casos. Podemos evidenciar um caso de funções de agregação ou agrupamento, que apesar de simples de implementar, de suma importância para qualquer empresa se manter em dia com obrigações financeiras: a despesas total gasta com funcionários, o total de salário a ser pago. A partir da tabela ‘pagamento’, podemos usar o comando ‘SELECT’ com a **função de agregação ‘SUM’**.

Outra importante utilidade para nosso banco de dados é o simples fato de armazenar dados relevantes para o negócio, nossa Mentorama MA. Temos uma entidade na nossa escola, uma das principais, que é nosso aluno. Temos a tabela ‘aluno’, que serve para o cadastro de todos os alunos na nossa escola, independente de turma. Porém, em sequência, temos que cadastrar nosso aluno em alguma turma de alguma modalidade. No nosso modelo de negócio, temos o cadastro dos alunos na turmas utilizando apenas o seu identificador. Isso nos proporciona uma maior agilidade e maior confiança na veracidade dos dados, pois, por exemplo, temos que digitar o nome completo do aluno apenas uma vez, no momento de seu cadastro na Escola, e não novamente no cadastro da turma, evitando assim, a chance que ocorram erros. Outro fator importante é evitar a redundância de dados, ou seja, evitar que termos a mesma informação armazenada em duas tabelas diferentes. Assim, podemos recuperar esses dados com **funções de relacionamento**, por exemplo os comandos de ‘**INNER JOIN**’.

Com o objetivo de avaliar os alunos de sua turma, para analisar aqueles que estão aptos a passarem de faixa, um professor deseja visualizar o nome completo de seus alunos (para confeccionar diplomas de graduação a novas faixas), a atual faixa que estão, e desde quando estão treinando na Escola, afinal, um dos critérios de avaliação é o tempo que já estão treinando. Podemos então utilizar os comandos ‘SELECT’ para visualizar os dados, a cláusula ‘WHERE’ para restringir por professor e o comando ‘INNER JOIN’ para voltar o nome completo de cada aluno.

Outro objetivo financeiro e de utilidade estratégica é verificar a quantidade de alunos por turma, separados por planos ou pacotes, sendo necessário ver de forma individual, ou seja, não apenas a quantidade, como seria utilizando funções de agregação ou agrupamento. Para isso podemos utilizar **funções de janela ‘ROW_NUMBER()’** para ter este padrão de análise.