Disciplina Laboratório de Banco de Dados

2021/1

RELATÓRIO FINAL

Arthur Battistel Ilha, Caroline Lewandowski, Pedro Zart, João Narciso

Escola Politécnica - PUCRS

21 de maio de 2021

Resumo

O presente trabalho apresenta a análise e construção da solução para o modelo proposto na disciplina de Laboratório de Banco de Dados no 2° semestre. O mesmo consiste na interpretação de um enunciado e de parte de um modelo de dados, de sua implementação nos SBGDs Oracle e MongoDB, na implementação de consultas e na criação de um relatório.

1. Esquema de Dados

1.1. Imagens

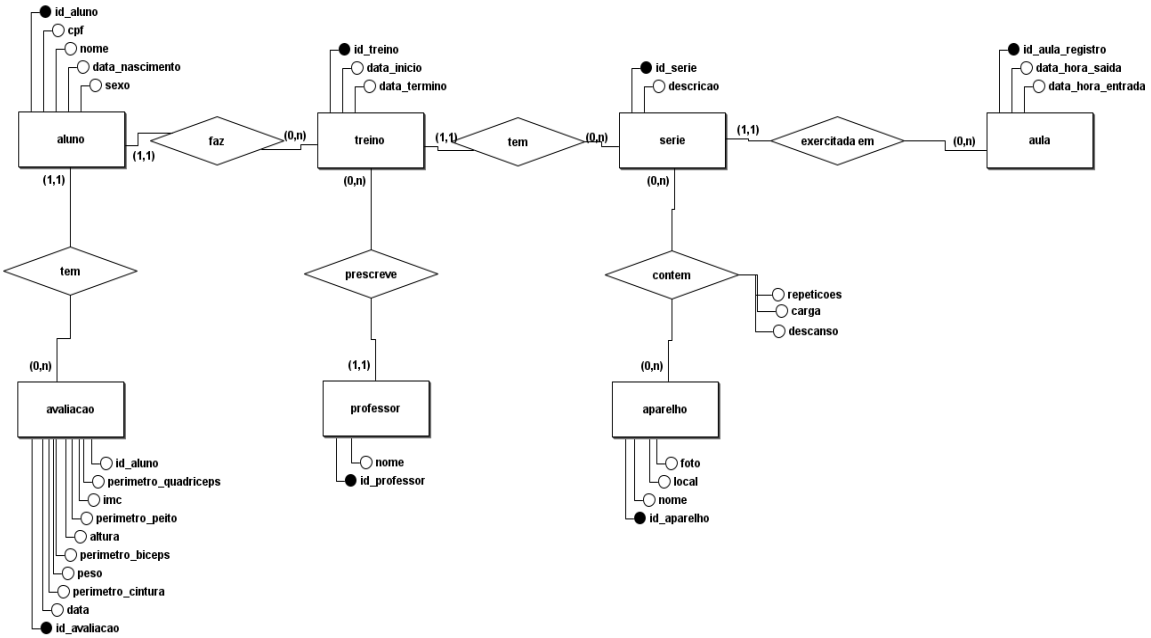

Figura 1: Representação do Esquema Conceitual

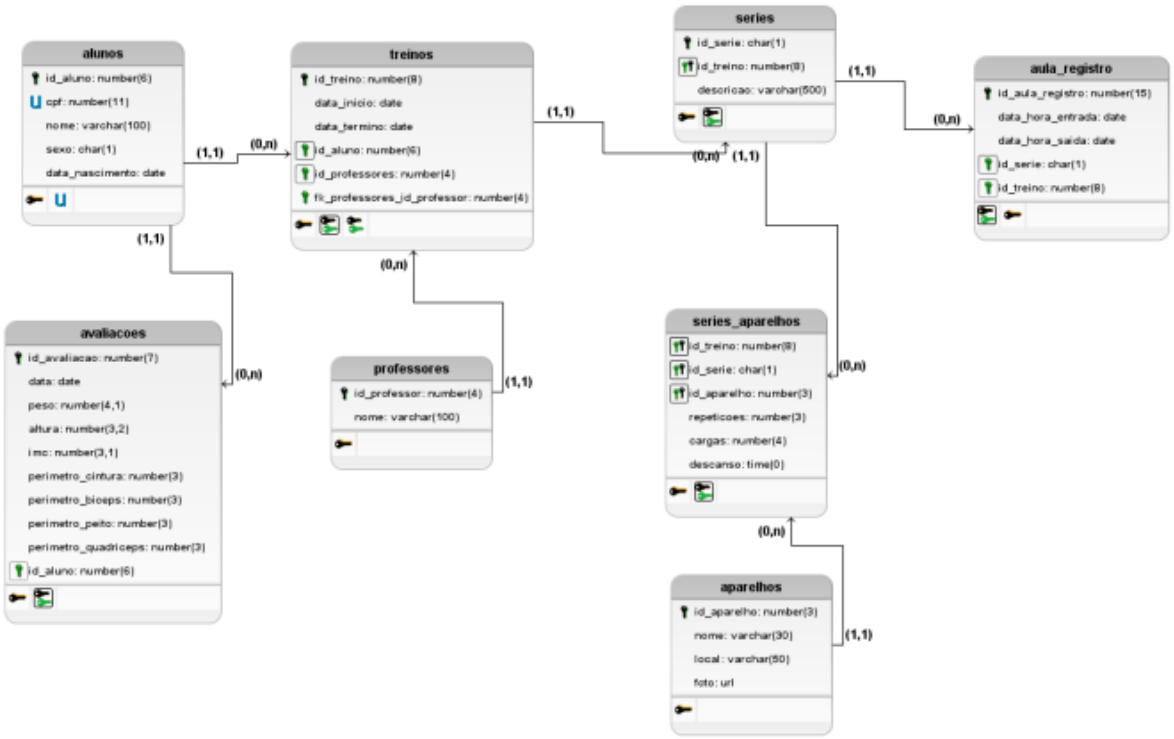

Figura 2: Representação do Esquema Lógico

1.2. Tomadas de Decisões

A partir da tabela "`series`", o grupo analisou os seguintes requisitos para continuar a construção do modelo de dados:

4. Cada treino é composto por uma ou mais séries de exercícios em aparelhos, que devem ser realizadas em dias alternados.

5. Cada série prevê exercícios em um ou mais aparelhos.

6. Para cada aparelho previsto na série o professor indica uma carga, o número de repetições e o tempo de descanso.

7. Os aparelhos são previamente cadastrados com seu nome, uma descrição de sua localização na academia e uma foto.

Dessa forma foi observado que cada série prevê exercícios em um ou mais aparelhos, foi avaliada que essa é uma relação *muitos para muitos* entre "`series`" e "`aparelhos`", portanto a tabela "`series_aparelhos`" foi criada.

Primeiro a tabela "`aparelhos`" foi criada, o 7º requisito informa que todos aparelhos são previamente cadastrados com nome, local e uma foto do aparelho, decidimos ter como chave primária dessa tabela "`id_aparelho`" e o restante dos

elementos descritos no 7° requisito. Os elementos "*nome*" e "*local*" tem o tipo *VARCHAR* e "*foto*" também do tipo *VARCHAR* para suportar *URL*.

Para a tabela resultante da relação entre "*series*" e "*aparelhos*", o grupo decidiu manter como chave primária os elementos: "*id_treino*" e "*id_serie*" vindo como foreign key da tabela "*series*" e o elemento "*id_aparelho*", foreign key da tabela "*aparelhos*". Para o restante dos elementos descritos no 6° requisito, foi criado os elementos "*repeticoes*" e "*cargas*" com o tipo *NUMBER* e "descanso" com o tipo *TIME*, que pode representar melhor o tempo de descanso pois imprime o formato "*hh:mm:ss*".

O 8º requisito informa que toda vez que um aluno comparece na academia um registro é realizado contendo a data e hora de entrada, a série executada e, ao final, a data e hora de saída para isso foi criado uma tabela "*aula_registro*", sendo assim decidimos ter como chave primária dessa tabela um "*id_aula_registro*" e vindo como foreign key da tabela "series" o "*id_série*" do tipo CHAR e o "*id_treino*" do tipo NUMBER. Para o restante dos elementos descritos no 8° requisito, foi criado os elementos "data_hora_entrada" e "data_hora_saida" com o tipo *timestamp default current_timestamp*, que pode representar melhor a data e hora pois imprime o formato "YYY-MM-DD HH:MM:SS".

2. Criação de consultas, gatilhos e utilização do MongoDB

2.1. Consultas SQL

Para realizar as consultas utilizamos os comandos que se encontram entre parênteses referenciando a consulta, quantidade de registros(COUNT(*)), todos os pesos(DISTINCT), o número de identificação dos treinos e suas datas de início(ORDER BY), cpf dos alunos(IN), nome completo do professor que possui as letras inicial tal(LIKE), Consultar o nome do aluno, id do treino realizado e a descrição da série somente dos treinos que tiveram início entre(FULL OUTER JOIN),nome a identificação do treino dos alunos que possuem aulas com o professor fulano(JOIN),Consultar o nome e peso do aluno que contém IMC entre 20 e 25(INNER JOIN),Consultar a data da avaliação o peso, altura, imc e nome do aluno que está vinculado ao professor com ID tal(INNER JOIN, RIGHT JOIN),Consultar a quantidade de alunos que possui o professor de identificação tal(HAVING,GROUP BY). Consultar

a quantidade de avaliações que possui o aluno de identificação tal(HAVING,GROUP BY),Consultar o nome de todos alunos e de todos professores(UNION), Consultar todas as datas de nascimento e todas as datas de avaliações encontradas(UNION), consultar o número de identificação do aluno e data de sua avaliação para o aluno que possua altura de 1.90 e peso acima do peso do aluno que contenha o código de identificação tal(SUB-CONSULTAS),Consultar a quantidade de alunos da tabela alunos que tem data de nascimento maior do que a data de nascimento do aluno de id tal(SUB-CONSULTAS).

2.2. Calcular IMC

Para calcular automaticamente o IMC, inicialmente será necessário criar um *trigger* que, antes de inserir uma nova *query* para a tabela *AVALIACOES*, confere o valor do `peso` e `altura` e substitui o campo imc com o resultado da conta `imc = peso / altura²`. O seguinte parágrafo apresenta o código usado pelo grupo para criação do gatilho responsável pela funcionalidade mencionada:

```
create or replace TRIGGER calcular_imc_auto
BEFORE INSERT OR UPDATE OF peso, altura ON AVALIACOES
FOR EACH ROW
BEGIN
    :NEW.imc := (:NEW.peso/POWER(:NEW.altura, 2));
END;
```

Para testar a criação do gatilho, será inserida uma nova query nas tabelas `ALUNOS` e `AVALIACOES`, sem inserir informações no campo imc:

```
INSERT INTO ALUNOS(id_aluno,cpf,nome,sexo,data_nascimento)
VALUES(11,82564539812,'Novo Aluno', 'F', to_date('18/05/1999','dd-mm-yy'));

INSERT INTO
AVALIACOES(id_avaliacao,data_avaliacao,peso,altura,perimetro_cintura,perimetro
_biceps,perimetro_peito,perimetro_quadriceps,id_aluno)
VALUES(11,to_date('21/05/2021','dd-mm-yy'), 61, 1.70, 62, 39.1, 86, 42.9, 10);
```

Como resultado teremos:

1	1	11/05/21	70	1,78	22,1	66	38	82	42	1
2	2	13/05/21	78	1,86	22,5	63	39	78	46	2
3	3	14/05/21	83	1,9	23	72	43	96	52	3
4	4	15/05/21	80	1,73	22,1	66	38	82	42	4
5	5	16/05/21	55	1,67	19,7	51	36	80	47	5
6	6	17/05/21	63	1,54	26,6	52	41	86	49	6
7	7	18/05/21	77	1,83	23	55	41	89	45	7
8	8	19/05/21	90	1,79	28,1	68	39	86	44	8
9	9	20/05/21	79	1,87	22,6	68	39	85	43	9
10	11	21/05/21	61	1,7	21,1	62	39	86	43	11

2.3. Coleção MongoDB

Foi criado um array de objetos onde cada aluno é representado por um objeto. Neste objeto consta o oid, chave gerada automaticamente na criação da collection, e atributos que referenciam outros objetos, como o do treino e professor. Esses id`s levam a outros objetos, que representam os respectivos atributos. Caso estivéssemos utilizando uma API Rest, por exemplo, faríamos uma requisição e a partir desse JSON retornado faríamos uma outra requisição passando o id do professor ou treino para seus respectivos endpoints e recebendo seus dados.

3. Conclusão

Durante o semestre desenvolvemos esse trabalho de forma coletiva referentes aos aprendizados passado pelo professor Azriel Majdenbaum na cadeira de Laboratório de Banco de Dados.

Pontua-se que a utilização da linguagem SQL satisfaz as condições para a elaboração do trabalho sem que houvesse problemas para representar qualquer proposta exigida pelas atividades.