

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CAMPUS SOROCABA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BANCO DE DADOS

PROF^a. DR^a. SAHUDY MONTENEGRO GONZÁLEZ

PROJETO PRÁTICO - FASE INTERMEDIÁRIA 2

ENTREGA: 21 de dezembro de 2023

Museu de fósseis - grupo 9

Arthur Naoto Nitto
Danillo Santiago Oliveira da Conceição
Thiago Proença Idro

SUMÁRIO

SUMÁRIO.....	1
1. Descrição do Problema.....	2
1.1. Consultas.....	3
2. Projeto Conceitual.....	3
3. Tabela de Metadados.....	4
4. Projeto Lógico.....	6
5. Projeto Físico do Banco de Dados.....	9
6. Especificação de Consultas em Álgebra Relacional e na SQL.....	11
7. Triggers.....	15
8. Considerações finais.....	16
Bibliografia.....	16

1. Descrição do Problema

Para ter informações precisas e seguras tanto de seus itens quanto de suas interações internas, um museu está interessado em um sistema que garanta que essas informações não sejam perdidas, além de facilitar a compreensão de seu funcionamento a fim de realizar melhorias na categorização de seus fósseis.

Para o Paleontólogo é necessário armazenar código de identificação (CPF), nome, data de nascimento, grau de escolaridade (Graduado, Pós-Graduado, Mestre ou Doutor), salário, universidade e equipe. Para o Visitante, é necessário armazenar código de identificação (CPF), nome, data de nascimento, gênero (M, F ou Outro) e idade (calculado a partir da data de nascimento). Visitante pode realizar visitas a uma Coleção de fósseis. Essa visita é gratuita, e para ela é necessário armazenar a data.

Todo Paleontólogo está vinculado a uma Universidade, mas nem toda Universidade possui um Paleontólogo. Para a Universidade, é necessário armazenar sua sigla (identificador), nome e endereço (estado e cidade). Todos os paleontólogos pertencem a uma Equipe de paleontólogos. Toda Equipe possui um nome (identificador), número de integrantes (calculado a partir do número de paleontólogos) e especialidade (Somatofósseis ou Icnofósseis). Uma Equipe é formada de vários paleontólogos e pode catalogar diversos fósseis.

Nesse sistema, é o Departamento que cuida do Fóssil catalogado por uma Equipe. Para o Departamento, é necessário armazenar seu identificador, nome e endereço (estado e cidade). Um Departamento pode possuir diversos fósseis.

Fóssil, por sua vez, possui um número de identificação, nome, descrição, tipo (Somatofósseis ou Icnofósseis), período (Triássico, Jurássico ou Cretáceo) e data de descoberta. Necessariamente, um Fóssil é catalogado por uma Equipe, pertence a um Departamento e faz parte de uma única Coleção.

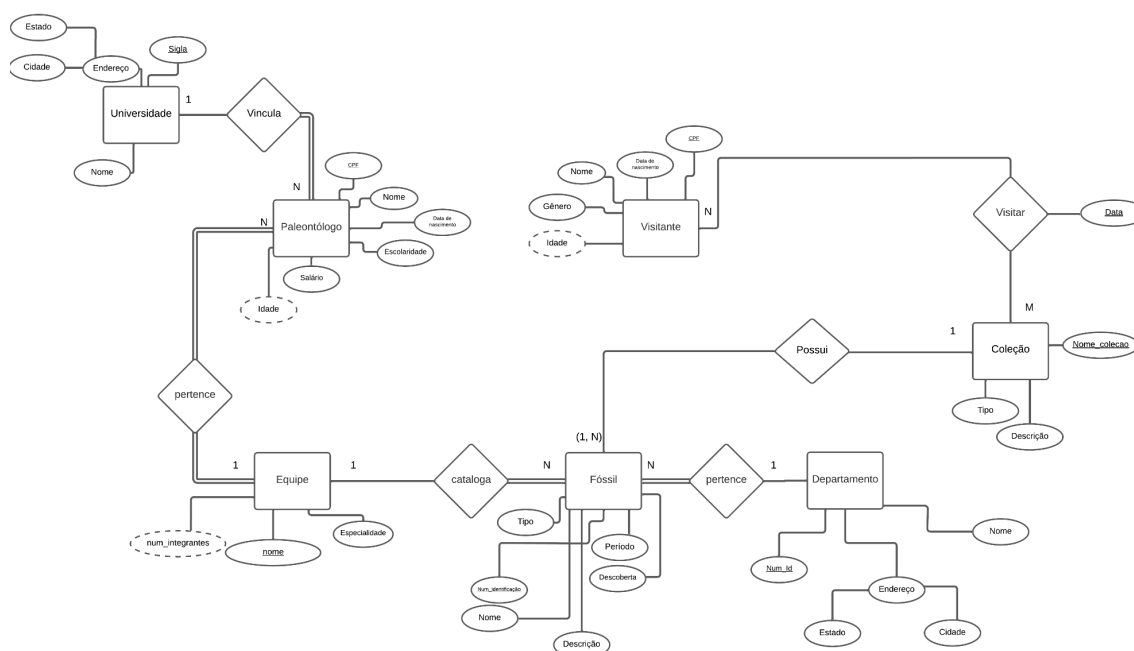
Uma Coleção possui um nome único (identificador), descrição e o tipo da coleção (Somatofósseis, Icnofósseis, Triássico, Jurássico ou Cretáceo). Uma Coleção pode conter vários fósseis, e tem no mínimo um fóssil para apresentar.

1.1. Consultas

1. Qual departamento possui o maior número de fósseis? (Arthur)
2. Quantas equipes possuem pelo menos um paleontólogo Doutor? (Arthur)
3. Que tipo de coleção possui mais fósseis e quantos são? (Arthur)
4. Quais são os paleontólogos de uma dada universidade? (Danillo)
5. Quais são os fósseis do período Jurássico? (Danillo)
6. Qual(is) visitante(s) possui(em) o maior número de visitas à coleção de fósseis Somatofósseis? (Danillo)
7. Qual é a quantidade de equipes com mais de 5 paleontólogos que é especializada em Icnofósseis? (Thiago)
8. Quais coleções possuem fósseis que pertencem a um certo departamento? (Thiago)
9. Quais fósseis foram descobertos antes de 1990? (Thiago)

2. Projeto Conceitual

Na Figura 1 é apresentado o esquema conceitual do banco de dados com base na descrição do problema (minimundo).



Link para melhor visualização:
<https://drive.google.com/file/d/1d5GJnppjOslcgWn7rD7VbWixDBacIlcD/view?usp=sharing>

Figura 1. MER - Museu de Fósseis.

3. Tabela de Metadados

Na Tabela 1 são apresentados os tipo-entidades, seus atributos e suas restrições. Vale destacar que atributos como idade e número de integrantes foram escolhidos como atributos calculáveis como uma oportunidade para implementação de triggers mais à frente no relatório.

Tipo-Entidade	Atributo	Tipo	Restrição
Paleontólogo	CPF	Identificador	Obrigatório
	Nome	Monovalorado	Obrigatório
	Data de nascimento	Monovalorado	Obrigatório
	Idade	Calculado a partir de Data de nascimento	Obrigatório, <= 125 anos
	Grau de escolaridade	Monovalorado	Obrigatório, valores = {Graduado, Pós-Graduado, Mestre, Doutor}
	Salário	Monovalorado	Obrigatório, > 0
Visitante	CPF	Identificador	Obrigatório
	Nome	Monovalorado	Obrigatório
	Data de nascimento	Monovalorado	Obrigatório
	Gênero	Monovalorado	Opcional, valores = {F, M, Outro}
	Idade	Calculado a partir de Data de nascimento	Obrigatório, <= 125 anos
Universidade	Sigla única	Identificador	Obrigatório
	Nome	Monovalorado	Obrigatório
	Endereço (estado e cidade)	Composto	Obrigatório
Equipe	Nome único	Identificador	Obrigatório
	Número de integrantes	Calculado a partir de relacionamento com Paleontólogo	Obrigatório, >=0
	Especialidade	Monovalorado	Obrigatório, valores = {Somatofósseis, Icnofósseis}

Coleção	Nome único	Identificador	Obrigatório
	Descrição	Monovalorado	Obrigatório
	Tipo da coleção	Monovalorado	Obrigatório, valores = {Somatofósseis, Icnofósseis, Triássico, Jurássico, Cretáceo}
Departamento	Identificador único	Identificador	Obrigatório
	Nome	Monovalorado	Obrigatório
	Endereço (estado e cidade)	Composto	Obrigatório
Fóssil	Número de identificação	Identificador	Obrigatório
	Nome	Monovalorado	Obrigatório
	Descrição	Monovalorado	Obrigatório
	Tipo	Monovalorado	Obrigatório, valores = {Somatofóssil, Icnofóssil}
	Período	Monovalorado	Obrigatório, valores = {Triássico, Jurássico, Cretáceo}
	Data de descoberta	Monovalorado	Obrigatório

Tabela 1. Tipos de atributos por tipo-entidade do MER.

4. Projeto Lógico

Para mapear cada entidade modelo Entidade-Relacionamento (MER) para o Modelo Relacional (MR), escolhemos começar mapeando os tipos entidades fortes Universidade, Equipe, Departamento, Coleção e Fóssil. Iremos começar com o mapeamento da Universidade e depois iremos mapear todas os outros tipos entidades, finalizando com as relações que ainda não tiverem sido mapeadas.

O tipo entidade Universidade se trata de um tipo entidade forte, com uma chave primária, um atributo simples e um atributo composto. Para evitar a normalização, optamos por mapear o atributo composto endereço como atributos atômicos, de forma que a relação ficaria assim: **Universidade** (sigla, nome, estado, cidade), com as seguintes dependências funcionais:

- sigla → nome;
- sigla → estado;
- sigla → cidade.

A relação se encontra na primeira forma normal (1FN), pois da forma como foi mapeada, não apresenta nenhum atributo multivalorado. Também se encontra na segunda e terceira forma normal (2FN e 3FN) por não possuir dependência funcional parcial nem dependência funcional transitiva.

A entidade forte Equipe possui uma chave primária, um atributo simples e um atributo calculável, sendo mapeada como: **Equipe** (nome_equipe, especialidade, num_integrantes), com as dependências funcionais:

- nome_equipe → especialidade;
- nome_equipe → num_integrantes.

Por não possuir nenhum atributo multivalorado, a relação já está normalizada na primeira forma normal (1FN). Também está normalizada na segunda e terceira forma normal (2FN e 3FN), por não possuir dependência funcional parcial nem dependência funcional transitiva.

O tipo entidade Departamento também se trata de uma entidade forte, com uma chave primária, um atributo simples e um atributo composto, o qual também será decomposto em vários atributos simples a fim de evitar a normalização. Assim, podemos descrever a relação como: **Departamento** (identificador, nome, estado, cidade), com as seguintes dependências:

- identificador → nome;
- identificador → estado;
- identificador → cidade.

A relação se encontra na primeira forma normal (1FN), pois não possui atributos multivalorados. Ela também se encontra na segunda forma normal (2FN), já que não possui nenhuma dependência parcial. Por fim, também se encontra na terceira forma normal (3FN), uma vez que não possui dependência funcional transitiva.

O tipo entidade forte Coleção possui uma chave primária e dois atributos simples, podendo ser descrita como: **Colecao** (nome_colecao, descricao, tipo), com as dependências funcionais:

- nome_colecao → descricao;
- nome_colecao → tipo.

A relação já se encontra na primeira e segunda forma normal (1FN e 2FN), pois não possui atributos multivalorados e não possui dependência funcional parcial. Também se encontra na terceira forma normal (3FN), visto que não possui dependência funcional transitiva.

O tipo entidade Visitante é um tipo entidade forte e possui uma chave primária, dois atributos simples e um atributo calculável, podendo ser mapeada como: **Visitante** (CPF, nome, data_nascimento, genero, idade), e possui as dependências funcionais:

- CPF-> nome;
- CPF-> data_nascimento;
- CPF-> genero;
- CPF-> idade.

A relação já se encontra na primeira e segunda forma normal (1FN e 2FN), pois não possui atributos multivalorados e não possui dependência funcional parcial. Também se encontra na terceira forma normal (3FN), visto que não possui dependência funcional transitiva.

O tipo entidade Paleontólogo é um tipo entidade forte com uma chave primária, quatro atributos simples, um atributo composto. o qual foi decomposto em atributos simples a fim de evitar a normalização, um atributo calculável e duas chaves estrangeiras (que são as chaves primárias de Universidade e Equipe), e pode ser mapeado como: **Paleontologo** (CPF, nome, data_nascimento, escolaridade, salario, idade, universidade, equipe), com as dependências funcionais:

- CPF-> nome;
- CPF-> data_nascimento;
- CPF-> escolaridade;
- CPF-> salario;
- CPF-> idade;
- CPF-> universidade;
- CPF-> equipe.

A relação já se encontra na primeira e segunda forma normal (1FN e 2FN), pois não possui atributos multivalorados e não possui dependência funcional parcial. Também se encontra na terceira forma normal (3FN), visto que não possui dependência funcional transitiva.

Já o tipo entidade Fóssil possui uma chave primária, cinco atributos simples e três chaves estrangeiras (com Departamento, Equipe e Coleção). O mapeamento fica dessa forma: **Fossil** (num_id, nome, descricao, tipo, periodo, descoberta, departamento, equipe, colecao), com as dependências funcionais:

- num_id-> nome;
- num_id-> descricao;
- num_id-> tipo;
- num_id-> periodo;

- num_id-> descoberta;
- num_id-> departamento;
- num_id-> equipe;
- num_id-> colecao.

A relação já se encontra na primeira e segunda forma normal (1FN e 2FN), pois não possui atributos multivalorados e não possui dependência funcional parcial. Também se encontra na terceira forma normal (3FN), visto que não possui dependência funcional transitiva.

Por fim, apenas o relacionamento Visitar precisa ser mapeado, pois todos os tipos entidades e relacionamentos já foram mapeados. Por se tratar de um relacionamento de vários para vários (n:n), mapeamos como uma tabela única, utilizando as chaves de Coleção e Visitante como chaves primárias, além de utilizar o atributo data também como chave primária, para que o sistema guarde o histórico de visitas de todos os visitantes. Dessa forma, o mapeamento fica como: **Visitar** (CPF, colecao, data), sem dependências funcionais, por possuir uma chave primária composta.

O mapeamento dessa relação já se encontra na primeira forma normal (1FN), por possuir apenas atributos atômicos. Também já está na segunda e terceira forma normal (2FN e 3FN), pois não há dependência funcional parcial ou transitiva.

Com o mapeamento finalizado, temos os seguintes relacionamentos e suas referências a chaves estrangeiras:

Universidade (sigla, nome, estado, cidade)

Equipe (nome_equipe, especialidade, num_integrantes)

Departamento (identificador, nome, estado, cidade)

Colecao (nome_colecao, descricao, tipo)

Visitante (CPF, nome, data_nascimento, genero, idade)

Paleontologo (CPF, nome, data_nascimento, escolaridade, salario, idade, universidade, equipe)
 universidade referencia Universidade;
 equipe referencia Equipe.

Fossil (num_id, nome, descricao, tipo, periodo, descoberta, departamento, equipe, colecao)
 departamento referencia Departamento;
 equipe referencia Equipe;
 colecao referencia Colecao.

Visitar (CPF, colecao, data)

CPF referencia Visitante.

colecacao referencia Colecacao.

5. Projeto Físico do Banco de Dados

Os códigos do banco de dados e a inserção se encontram anexados no arquivo `esquema.sql`. A seguir está a descrição breve de cada tabela:

Para a tabela `visitante`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `CPF`, sendo a chave primária;
- **NOT NULL** para `nome`, `data_nascimento` e `genero`, pois não podem ser nulos;
- **CHECK** para `genero`, para restringir a escolha de opções para esse atributo;
- **DEFAULT** para `genero`, caso o usuário não escolha nenhuma das opções disponíveis.

Para a tabela `universidade`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `sigla`, sendo a chave primária;
- **NOT NULL** para `nome`, `estado` e `cidade`, pois não podem ser nulos.

Para a tabela `equipe`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `nome`, sendo a chave primária;
- **NOT NULL** para `especialidade`, pois não pode ser nulo;
- **CHECK** para `especialidade`, para restringir a escolha de opções para esse atributo.

Não foi escolhido uma opção **DEFAULT** nesse caso, pois não faz sentido ter essa opção, a `equipe` precisa ter uma `especialidade`.

Para a tabela `colecacao`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `nome`, sendo a chave primária;
- **NOT NULL** para `descricao` e `tipo` pois não podem ser nulos;
- **CHECK** para `tipo`, para restringir a escolha de opções para esse atributo.

Não foi escolhido uma opção **DEFAULT** nesse caso, pois não faz sentido ter essa opção, `colecacao` precisa ter um `tipo`.

Para a tabela `departamento`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `id_dep`, sendo a chave primária;
- **NOT NULL** para `nome`, `estado` e `cidade` pois não podem ser nulos.

Para a tabela `paleontologo`, foram usadas as seguintes cláusulas:

- **PRIMARY KEY** para `CPF`, sendo a chave primária;

- *NOT NULL* para `nome`, `data_nascimento`, `escolaridade`, `salario`, `universidade` e `equipe` pois não podem ser nulos;
- *CHECK* para `escolaridade`, para restringir a escolha de opções para esse atributo.
- *FOREIGN KEY* para `universidade`, referenciando o atributo `sigla` da tabela `universidade`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.
- *FOREIGN KEY* para `equipe`, referenciando o atributo `nome` da tabela `equipe`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.

Não foi escolhido uma opção *DEFAULT* nesse caso, pois não faz sentido ter essa opção, `paleontologo` precisa ter uma `escolaridade`.

Para a tabela `fossil`, foram usadas as seguintes cláusulas:

- *PRIMARY KEY* para `id_fos`, sendo a chave primária;
- *NOT NULL* para `nome`, `descricao`, `tipo`, `periodo`, `descoberta`, `departamento`, `equipe` e `colecacao` pois não podem ser nulos;
- *CHECK* para `tipo` e `periodo` para restringir a escolha de opções para esse atributo.
- *FOREIGN KEY* para `departamento`, referenciando o atributo `id_dep` da tabela `departamento`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.
- *FOREIGN KEY* para `equipe`, referenciando o atributo `nome` da tabela `equipe`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.
- *FOREIGN KEY* para `colecacao`, referenciando o atributo `nome` da tabela `colecacao`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.

Assim como foi discutido em `equipe` e `colecacao`, não foi usado *DEFAULT* nos atributos `tipo` e `periodo`.

Para a tabela `visitar`, foram usadas as seguintes cláusulas:

- *PRIMARY KEY* para `cpf`, `colecacao`, `data_vis`, sendo a chave primária;
- *FOREIGN KEY* para `cpf`, referenciando o atributo `cpf` da tabela `visitante`, seguindo de *ON UPDATE CASCADE ON DELETE CASCADE* para atualizar ou deletar caso haja alguma alteração no atributo.
- *FOREIGN KEY* para `colecacao`, referenciando o atributo `nome` da tabela `colecacao`, seguindo de *ON UPDATE CASCADE ON DELETE*

CASCADE para atualizar ou deletar caso haja alguma alteração no atributo.

6. Especificação de Consultas em Álgebra Relacional e na SQL

A inserção está no arquivo `insercao.sql` e as consultas são encontradas em `consultas.sql`. A seguir estão as consultas em AR e SQL:

1. Qual departamento possui o maior número de fósseis? (Arthur)

AR:

$$t1 \leftarrow (fossil \bowtie_{(departamento, id-dep)} departamento)$$

$$t2_{(id, max)} \leftarrow_{(id-dep)} \mathcal{F}_{max(id-fossil)}(t1)$$

$$t3_{(max)} \leftarrow \mathcal{F}_{max(max)}(t2)$$

$$r \leftarrow \Pi(t2 \bowtie t3)$$

Figura 2. AR 1.

SQL:

```
SELECT d.nome, COUNT(f.id_fos)
FROM fossil f INNER JOIN departamento d ON f.departamento =
d.id_dep
GROUP BY d.id_dep
HAVING COUNT(f.id_fos) = (SELECT MAX(cnt) FROM (
    SELECT COUNT(id_fos) AS cnt
    FROM fossil f INNER JOIN departamento d ON
f.departamento = d.id_dep
    GROUP BY d.id_dep
  )
)
```

2. Quantas equipes possuem pelo menos um paleontólogo Doutor? (Arthur)

Nesse caso, a escolaridade do paleontólogo foi definida como “Doutor”, mas na AR está de uma forma genérica. Para realizar a consulta, basta trocar o valor de <escolaridade> para “Doutor”.

AR:

$$t1 \leftarrow \Pi_{(nome)} \left(\sigma_{escolaridade=<escolaridade>} (paleontologo \bowtie_{(equipe,nome)} equipe) \right)$$

$$r \leftarrow \mathcal{F}_{count(*)}(t1)$$

Figura 3. AR 2.

SQL:

```

SELECT count(*) FROM(
    SELECT e.nome
    FROM paleontologo pa INNER JOIN equipe e ON pa.equipe
    = e.nome
    WHERE pa.escolaridade = 'Doutor'
    GROUP BY e.nome
)

```

3. Que tipo de coleção possui mais fósseis e quantos são? (Arthur)

AR:

$$t1 \leftarrow \Pi \left((colecão) \bowtie_{(nome,colecão)} (fossil) \right)$$

$$t2_{(tipo,c)} \leftarrow_{(tipo-colecão)} \mathcal{F}_{count(id-fossil)}(fossil)$$

$$t3_{(max)} \leftarrow \mathcal{F}_{max(c)}(t2)$$

$$r \leftarrow \Pi_{(tipo,max)}(t2 \bowtie_{(c,max)} t3)$$

Figura 4. AR 3.

SQL:

```

SELECT co.tipo, COUNT(f.id_fos)
FROM colecao co INNER JOIN fossil f ON co.nome = f.colecao
GROUP BY co.tipo
HAVING COUNT(f.id_fos) = (
    SELECT MAX(cnt)
    FROM (
        SELECT COUNT(f.id_fos) AS cnt
        FROM colecao co INNER JOIN fossil f ON co.nome =
f.colecao

```

GROUP BY co.tipo
)
)

4. Quais são os paleontólogos de uma dada universidade? (Danillo)

AR:

$$\Pi_{\text{nome}} (\sigma_{\text{nomeuniv}=\langle \text{nomeduniversidade} \rangle} ((\text{Paleontologo}) \bowtie_{(\text{universidade, sigla})} \rho_{\text{nomeuniv/nome}}(\text{Universidade})))$$

Figura 5. AR 4.

SQL:

```
SELECT p.nome
FROM Paleontologo p
JOIN Universidade u ON p.universidade = u.sigla
WHERE u.nome = <nome_da_universidade>;
```

5. Quais são os fósseis do período Jurássico? (Danillo)

AR:

$$\Pi_{\text{nome, tipo}} (\sigma_{\text{periodo}=\text{'Jurássico'}}(\text{Fossil}))$$

Figura 6. AR 5.

SQL:

```
SELECT nome, tipo FROM Fossil WHERE periodo =
'Jurássico';
```

6. Qual(is) visitante(s) possui(em) o maior número de visitas à coleção de fósseis Somatofósseis? (Danillo)

AR:

$$t1_{(visitas)} \leftarrow_{cpf} \mathcal{F}_{count(colecao)} (\sigma_{tipo='Somatofósseis'}((Visitar) \bowtie_{(colecao,nome)} (Colecao)))$$

•

$$t2_{(maxvisitas)} \leftarrow \mathcal{F}_{max(visitas)}(t1)$$

•

$$\Pi_{cpf, \mathcal{F}_{count(colecao)}} (\mathcal{F}_{count(t2)}(\sigma_{tipo='Somatofósseis'}((Visitar) \bowtie_{(colecao,nome)} (Colecao))))$$

Figura 7. AR 6.

SQL:

```
SELECT v.cpf, COUNT(v.colecao) AS numero_visitas
FROM Visitar v
JOIN Colecao c ON v.colecao = c.nome
WHERE c.tipo = 'Somatofósseis'
GROUP BY v.cpf
HAVING COUNT(v.colecao) = (
    SELECT MAX(visitas)
    FROM (
        SELECT COUNT(v2.colecao) AS visitas
        FROM Visitar v2
        JOIN Colecao c2 ON v2.colecao = c2.nome
        WHERE c2.tipo = 'Somatofósseis'
        GROUP BY v2.cpf
    ) AS max_visitas
);
```

7. Qual é a quantidade de equipes com mais de 5 paleontólogos que é especializada em Icnofósseis? (Thiago)

AR:

$$t1 \leftarrow \Pi \left(\sigma_{(especialidade=<especialidade> \wedge num_integrantes > <quantidade>)}(equipe) \right)$$

$$r \leftarrow \mathcal{F}_{count(nome)}(t1)$$

Figura 8. AR 7.

SQL:

```
SELECT COUNT(nome) AS qtd_equipes
FROM equipe
WHERE especialidade = 'Icnofósseis' AND num_integrantes >
```

8. Quais coleções possuem fósseis que pertencem a um certo departamento? (Thiago)

AR:

$$\begin{aligned}
 t1 &\leftarrow \Pi \left((fossil) \bowtie_{(departamento, id_dep)} (departamento) \right) \\
 t2 &\leftarrow \Pi_{(colecacao)} \left(\sigma_{(id_dep = \langle id_dep \rangle)} (t1) \right) \\
 r &\leftarrow \Pi \left((colecacao) \bowtie_{(nome, colecacao)} (t2) \right)
 \end{aligned}$$

Figura 9. AR 8.

SQL:

SELECT *

FROM colecacao cl INNER JOIN

(SELECT f.colecacao

FROM fossil f INNER JOIN departamento d

ON f.departamento = d.id_dep

WHERE d.id_dep=2) AS fd

ON cl.nome = fd.colecacao

ORDER BY cl.nome ASC;

9. Quais fósseis foram descobertos antes de 1990? (Thiago)

AR:

$$\begin{aligned}
 t1 &\leftarrow \Pi \left(\sigma_{(descoberta < \langle descoberta \rangle)} (fossil) \right) \\
 r &\leftarrow \Pi_{(id_fos, nome, periodo, descoberta)} (t1)
 \end{aligned}$$

Figura 10. AR 9.

SQL:

SELECT id_fos, nome, periodo, descoberta

FROM fossil

WHERE date_part('year', descoberta) < 1990;

7. Triggers

Os *triggers* estão no arquivo `triggers.sql`, mas explicando brevemente, há uma função `checa_idade()` que calcula a idade com base na data de nascimento e verifica se é uma idade válida (<125 anos). Os *triggers* que usam essa função são `t_checa_idade_visitante` e `t_checa_idade_paleontologo` (usados para validar o atributo idade nas tabelas `visitante` e `paleontologo`, respectivamente).

Além disso, há uma função `calcula_integrantes()` que atualiza a quantia de paleontólogos na equipe toda vez que é inserido um paleontólogo. O *trigger* que usa essa função é `t_calcula_integrantes`.

8. Considerações finais

As ferramentas disponíveis para criação de Diagrama de Entidade-Relacionamento (LucidChart) e para criação da álgebra relacional das consultas (Latex) foram de grande valia para facilitar o desenvolvimento e entendimento das seções exigidas como Projeto Conceitual e Especificação de Consultas em Álgebra Relacional e na SQL.

Embora o projeto esteja realizando todas as ações esperadas pela descrição do problema, há limitações relacionadas ao escopo pelo qual foi desenvolvido, a visita realizada por um visitante por exemplo, apenas armazena dados básicos sobre essa relação que são quem foi o visitante, a coleção visitada e a data, carecendo de informações mais detalhadas sobre a visita; uma melhoria futura para o projeto do museu de fósseis poderia ser a expansão dos atributos e relacionamentos referentes à visita.

Ademais, o projeto desenvolvido alcançou o objetivo principal idealizado na descrição do mini mundo que consiste no registro das informações de um museu de fósseis.

Bibliografia

- SiBBr, **Base de dados de Paleontologia do MZUSP**. Disponível em: <https://collectory.sibbr.gov.br/collectory/public/show/dr414>. Acesso em: 18 nov, 2023.
- PaleoMundo, **Coleções de Fósseis de A a Z (de Aldrovandi à Zabini)**. Disponível em: <https://www.blogs.unicamp.br/paleoblog/2018/04/10/colecao-de-rochas-minerais-e-fosseis/#:~:text=As%20cole%C3%A7%C3%B5es%20de%20f%C3%B3sseis&text=N%C3%A3o%20cabiam%20mais%20em%20simples,esp%C3%A9cies%20animais%20e%20minerais>. Acesso em: 18 nov, 2023.
- Museu Paraense Emílio Goeldi, **Coleções científicas e a mediação museológica**. Disponível em: <https://www.museu-goeldi.br/noticias/colecoes-cientificas-e-a-mediacao-museologica>. Acesso em: 18 nov, 2023.
- Arqueologia e Pré-história, **Museus de Paleontologia no Brasil**. Disponível em: <https://arqueologiaeprehistoria.com/museus-de-paleontologia-no-brasil/>. Acesso em: 18 nov, 2023.
- Revista Arco, **Coleta de Fósseis**. Disponível em: <https://www.ufsm.br/midias/arco/post471>. Acesso em: 18 nov, 2023.