

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação

SCC0640 - Bases de Dados
Prof. Elaine Parros M. de Sousa
PAE André Moreira Souza

V.U.E.I. D.E.M.A.I.S.
Veículo Universal de Expedições Intergalácticas
Desbravando Espaços Muito Além das Infinitudes Siderais

Dikson Ferreira dos Santos - 11275147
Felipe Cadavez Oliveira - 11208558
Gabriela Rodrigues do Prado - 11892917
Luca Maciel Alexander - 11219175
Victor Gomes de Carvalho - 11275168

São Carlos, 2023

Sumário

1. Introdução.....	3
2. Modelagem.....	4
2.1. Notação.....	4
2.2. Requisitos de Dados.....	4
2.3. Principais Funcionalidades.....	6
2.4. Restrições de Integridade.....	7
2.5. Modelo Entidade-Relacionamento.....	9
2.6. Anotações para o MER-X.....	10
2.7. Correções da Entrega 1.....	12
3. Projeto Lógico.....	13
3.1. Esquema Lógico.....	13
3.2. Justificativas.....	14
3.3. Restrições.....	18
3.4. Correções da Entrega 2.....	18
4. Implementação (github.com/diksown/vuei).....	19
4.1. Consultas.....	19
4.2. Aplicação.....	21
5. Conclusão.....	26
5.1. Análise.....	26
5.2. Críticas.....	26
5.3. Sugestões.....	27

1. Introdução

“Se você não está quebrando coisas, você não está inovando.”

– **Stockton Rush**, CEO da OceanGate

Por muito tempo, a exploração espacial permaneceu um sonho inalcançável para a maioria, reservada apenas aos bolsos mais abastados da sociedade. As viagens espaciais eram um privilégio exclusivo dos bilionários, um deleite para poucos sortudos que podiam arcar com os custos *astronômicos* envolvidos.

No entanto, essa realidade mudou drasticamente quando um grupo de alunos do ICMC introduziram o revolucionário **V.U.E.I. (Veículo Universal de Expedições Intergalácticas)**. Eles substituíram os sistemas de navegação espaciais complicados por controles de videogame sem fio e simplificaram os dispendiosos e restritivos processos de segurança, impulsionando o avanço tecnológico e democratizando o acesso cósmico.

As incríveis jornadas lideradas pela empresa criada por esses estudantes, agora líder no turismo espacial, tornaram-se um fenômeno mundial. Elas conduzem aventureiros em emocionantes expedições pelo cosmos, explorando as trilhas que ligam os mais distantes planetas e estrelas. Assim, as naves baseadas no **V.U.E.I.** se transformaram no ícone da aventura intergaláctica acessível, abrindo as portas do espaço para uma variedade de viajantes, desde famílias em busca de férias memoráveis até aventureiros solitários em busca de experiências únicas.

O usuário-alvo desse projeto é composto por três grupos principais: turistas em busca de experiências interplanetárias, pilotos responsáveis por conduzir as naves espaciais, e administradores encarregados de manter e gerenciar os aspectos operacionais. O objetivo do sistema é dar suporte à empresa nas suas principais atividades, especialmente as relacionadas à organização de suas expedições cósmicas.

2. Modelagem

2.1. Notação

Na seção a seguir, para facilitar o acompanhamento do texto, **entidades** serão representadas em **itálico e negrito**, enquanto **atributos** serão destacados em **negrito**. Os **atributos** que fazem parte da chave primária serão adicionalmente **sublinhados**.

2.2. Requisitos de Dados

Uma **pessoa** no sistema possui informações como **nome**, **email**, **senha**, **telefone** e **data de nascimento**. A depender de seu **tipo**, essas **pessoas** podem ser **turistas** e/ou **pilotos**. Os **pilotos** possuem um **número de licença**, enquanto os **turistas** têm o **país de origem** registrado. **Turistas** podem participar de **expedições**, enquanto **pilotos** (um por **expedição**) guiam as **naves** que irão nessas **expedições**.

As **expedições** possuem **data/hora de início** e **data/hora de fim**, e seguem **rotas** pré-determinadas. Cada **expedição** será guiada por uma única **nave**, identificada por seu **número de registro**. Todas as **naves** são de algum **modelo** e possuem uma **capacidade** máxima de pessoas.

Cada **rota** possui um **nome** e passa por diversos **corpos celestes**. A depender de seu **tipo**, esses **corpos** podem ser divididos em **planetas** e **estrelas**. Todos os **corpos** possuem as informações de **nome**, **raio**, **massa**, **distância da terra**, **temperatura** e o valor de sua **gravidade** (tal que nome é único dentro de uma especificada galáxia). Os **planetas** possuem um atributo adicional para indicar a **presença de água** e podem ser classificados entre **habitáveis** ou **não habitáveis**, enquanto as **estrelas** possuem uma determinada **luminosidade**.

Além disso, cada **planeta** orbita uma **estrela**, e todos os **corpos** pertencem a uma **galáxia**, identificadas por seu **nome**.

A humanidade também descobriu e **nomeou** diversas **raças alienígenas**. Após anos de pesquisas realizadas por alunos de estatística do ICMC, temos informações como **tamanho** e **peso** médio dos indivíduos de cada **raça**, assim como uma **descrição** textual de suas principais características.

Ao longo do tempo, alienígenas se uniram em **facções aliens**. Cada **facção** possui um **nome** e é constituída de diferentes **raças**. Alunos de estatística do ICMC foram capazes de calcular um **número estimado de membros** dessas **facções**, que podem ter colonizado **planetas habitáveis**, formando **colônias**.

Cada **colônia** possui um **nome** e **data/hora de início** e possivelmente uma **data/hora de término**, caso algum infortúnio tenha encerrado elas em algum momento. Como nem tudo no mundo (nem no universo...) são flores, as **colônias** podem ter entrado em **guerras**.

Uma **guerra** ocorre entre duas **colônias**. A **colônia** que iniciou o ataque é considerada a agressora, enquanto a que foi atacada é considerada a defensora. Cada **guerra** tem uma **data/hora de início**, um **número de órbitos** e pode eventualmente ser finalizada em uma **data/hora de término**.

2.3. Principais Funcionalidades

O sistema tem as seguintes principais funcionalidades, divididas por tipo de usuário:

- Turista:
 - Cadastro de perfil;
 - Consulta de expedições;
 - Pesquisa de expedições;
 - Inscrição em uma expedição;
 - Pesquisa por dados de corpos celestes;
 - Consulta de expedições que não passam por nenhum planeta em guerra;
 - Consulta do histórico de planetas visitados e raças encontradas;
- Piloto
 - Pesquisa por veículo espacial;
 - Consulta de expedições envolvendo naves que vai pilotar, suas características e os usuários que estão embarcando;
 - Consulta do histórico de guerras de planetas;
 - Consulta de raças e colônias atualmente presentes em planetas;
- Administrador
 - Cadastro de veículos espaciais;
 - Pesquisa por dados cadastrais de usuários;
 - Criação, alteração e remoção de guerras espaciais;
 - Consulta de viagens de um usuário;
 - Pesquisa por rota;
 - Consulta de raças de criaturas espaciais identificadas e seus atributos;
 - Consulta de galáxias identificadas e seus atributos;
 - Consulta de colônias identificadas e seus atributos;
 - Consulta de expedições registradas, seus atributos e os usuários que estão embarcando;
 - Consulta de usuários que já visitaram determinado corpo ou galáxia;

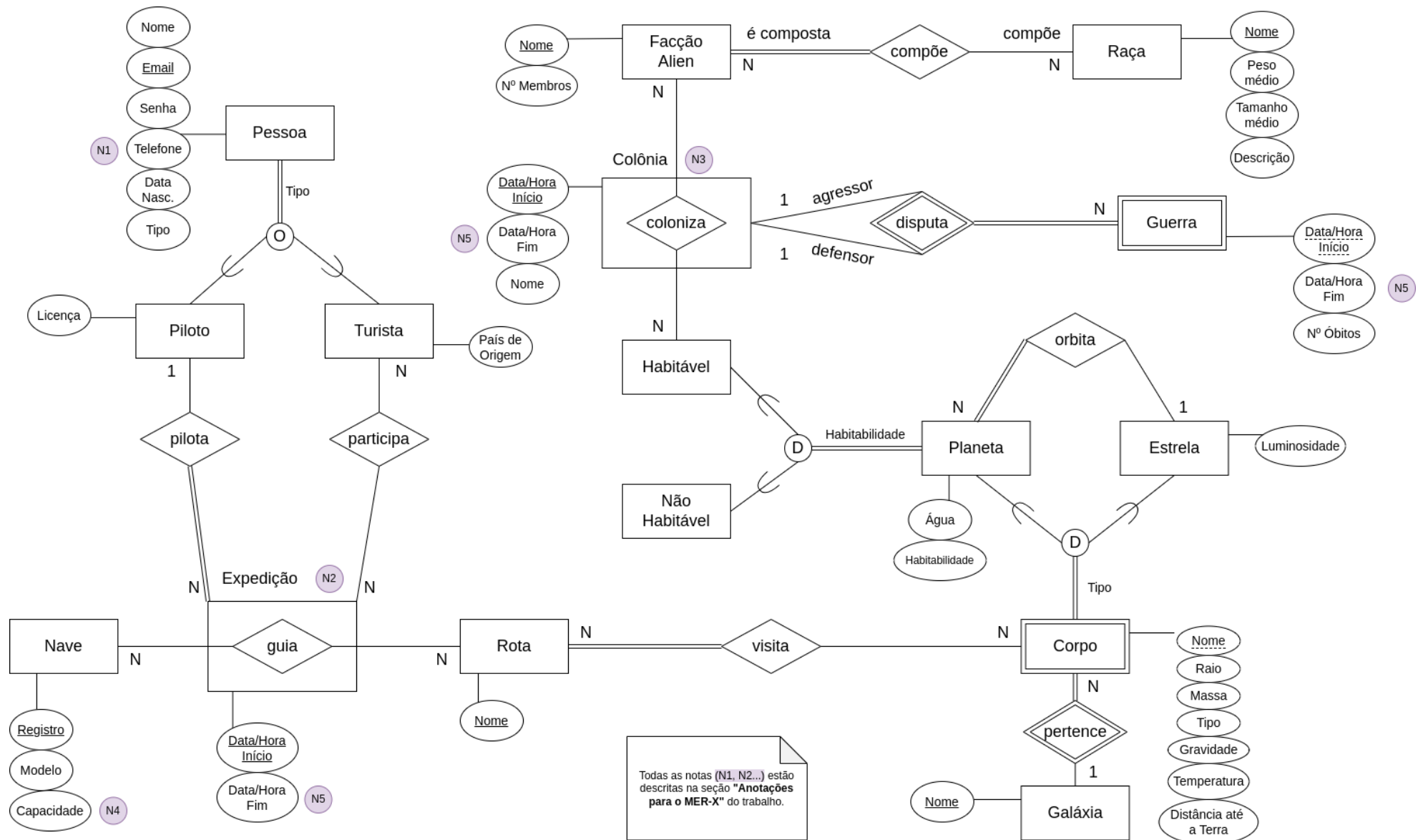
2.4. Restrições de Integridade

- Ciclo: Pessoa - Expedição
 - Este ciclo de dependência é necessário, pois tanto Turistas quanto Pilotos podem se relacionar com uma Expedição, e ambos Turistas e Pilotos são Pessoas, sendo que um Turista pode também ser Piloto (especialização de sobreposição). Como não é tratado na estrutura, a aplicação deve garantir que um Piloto não participe como Turista de uma Expedição que ele mesmo pilota.
- Potencial inconsistência no CR disputa
 - O auto-relacionamento com Colônia é necessário, pois para definir uma Guerra, é preciso especificar quais duas Colônias estão em guerra. Uma Colônia pode estar em guerra com ela mesma (i.e. guerra civil). Como não é tratado na estrutura, a aplicação não deve permitir que sejam criadas duas Guerras simultâneas (com intersecção não nula de intervalos de horário) entre as mesmas duas Colônias, independente de qual das duas foi a agressora.
- Potencial inconsistência no CR orbita
 - A estrutura modelada não especifica que um Planeta deve orbitar uma Estrela de sua mesma Galáxia, embora isso não faça sentido em nossa semântica. Nesse caso, a aplicação fica responsável por garantir essa restrição.
- Potenciais inconsistências em Expedição
 - Toda Nave possui uma capacidade máxima de passageiros. A estrutura não garante que esta capacidade será respeitada. A aplicação deve se responsabilizar por garantir que a soma da quantidade de Turistas que participam de uma Expedição com a quantidade de Pilotos (um único) não exceda a capacidade máxima da Nave.
 - A modelagem não impossibilita a criação de duas Expedições simultâneas com a mesma Nave, o que não faz sentido no mundo real, pois uma única Nave não pode guiar duas Expedições ao mesmo tempo. A aplicação deve garantir que isso não ocorra. Para exemplificar um caso

inconsistente permitido pela estrutura (assumindo nenhuma restrição por parte da aplicação): como uma Expedição é identificada por um número de registro de Nave e por uma data/hora de início, poderiam existir duas instâncias de Expedição identificadas por {123, 24-05-2025/10:00} e {123, 25-05-2025/10:00}, com ambas terminando no dia 26 de maio de 2025. Nesse caso, apesar de suas chaves serem distintas, ambas as Expedições estariam ocorrendo na tarde do dia 25, portanto não poderiam ser guiadas pela mesma Nave de registro 123, demonstrando a inconsistência.

- O mesmo problema exemplificado no item acima ocorre entre Expedição e Piloto, pois a modelagem não garante que um mesmo Piloto não pilote Expedições simultâneas. A aplicação deve garantir que se duas Expedições possuem intersecção de horários, então não podem ser Pilotadas pela mesma Pessoa.
- Potencial inconsistência em Colônia
 - É suposto que não faz sentido uma única Facção Alien possuir duas Colônias em um mesmo Planeta ao mesmo tempo. A modelagem não garante essa suposição semântica, portanto a aplicação não deve permitir que existam duas Colônias com horários de existência intersectantes, da mesma Facção Alien, no mesmo Planeta.
- Potencial inconsistência em Guerra
 - Esta inconsistência potencialmente atinge os valores de data/hora de início e fim de uma Guerra. Uma Guerra só pode existir enquanto ambas as Colônias participantes existem. Assim, a aplicação deve consultar os valores de data/hora de início e fim tanto de Guerra quanto das Colônias envolvidas, para garantir que o intervalo de horários em que ocorre uma Guerra está contido na intersecção dos intervalos de horário de existência das Colônias envolvidas.

2.5. Modelo Entidade-Relacionamento



2.6. Anotações para o MER-X

- Foi utilizada a notação tradicional para indicar chaves primárias, com anotações indicando possíveis chaves secundárias.
- Chaves candidatas
 - **(N1)** O atributo telefone também identifica Pessoa univocamente.
- Chaves de agregações
 - **(N2)** A chave primária da agregação Expedição é composta pela chave da Nave e por seu atributo data/hora de início.
 - **(N3)** A chave primária da agregação Colônia é definida pela chave do Planeta Habitável, chave da Facção Alien e pelo seu atributo data/hora de início.
- Atributos
 - **(N4)** É importante que o atributo Nave.capacidade seja não nulo, pois uma restrição de integridade depende de seu valor.
 - **(N5)** As entidades Expedição/Colônia/Guerra possuem atributos de 'data/hora fim'. Se estes valores existirem no mundo real, é importante que sejam inseridos na base de dados, pois algumas restrições de integridade dependem deles. No caso da Expedição, podemos impor essa restrição no futuro com um *not null*, mas não podemos fazer isso com Colônia/Guerra, já que precisamos de um valor nulo para indicar que a Colônia/Guerra ainda não acabou. Além disso, se esse atributo não for nulo, ele não pode estar antes da data/hora de início da entidade correspondente. Por exemplo, uma Guerra não pode começar em 2025 e terminar em 2024.
- Demais suposições semânticas
 - Um Turista pode participar de duas Expedições cujos horários de viagem se intersectam. Esse cenário é interpretado como o Turista ter comprado dois ingressos para viagens separadas, sabendo que só participará de uma.
 - O atributo Colônia.data/hora_fim pode ter valor nulo, indicando que a correspondente Colônia ainda existe.
 - O atributo Guerra.data/hora_fim pode ter valor nulo, indicando que a correspondente Guerra ainda ocorre.
 - Duas Colônias de Planetas distintos podem guerrear entre si.

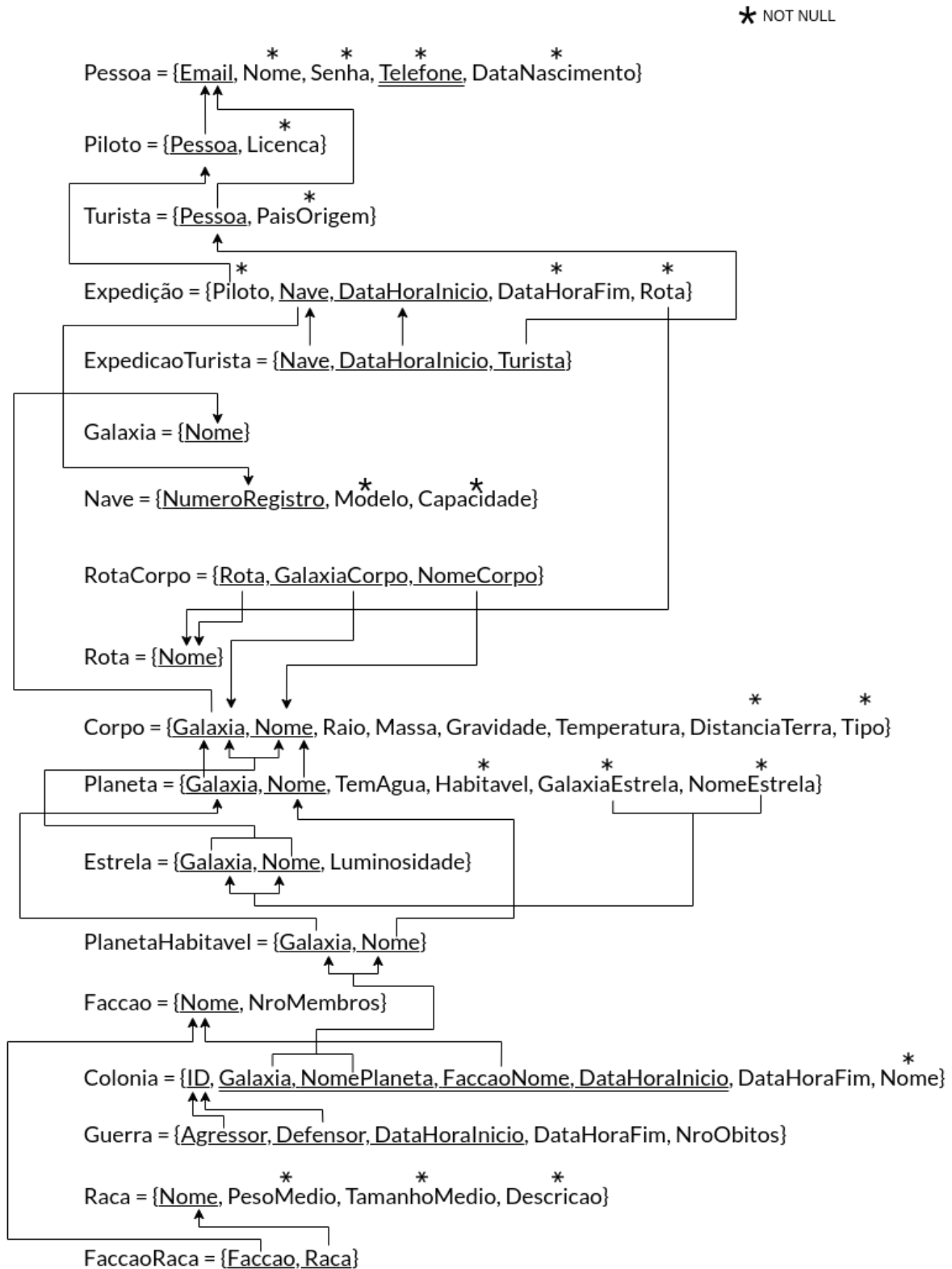
- A presença de água é independente do fato de um Planeta ser Habitável ou não.
- Nas Expedições, a ordem dos Corpos a se visitar será crescente na distância até a terra desses Corpos.

2.7. Correções da Entrega 1

Todos os erros identificados pelo monitor foram corrigidos. Especificamente: foi deixado mais claro nos requisitos de dados que um corpo deve ser entidade fraca de galáxia, impondo a restrição de que o nome de um corpo é único dentro de uma determinada galáxia; a terminologia 'x é chave secundária de y' foi substituída por 'x identifica y univocamente'; os atributos de critério (tipo) foram mencionados nos requisitos de dados.

3. Projeto Lógico

3.1. Esquema Lógico



3.2. Justificativas

- **PKs compostas**

- **Observação:** Em toda ocorrência de PK composta, a busca pelo primeiro campo é priorizada (pela estrutura de indexação).

- **ID sintético em 'Colônia'**

- **Solução adotada:** Um atributo que representa um identificador sintético identifica uma Colônia univocamente.
- **Vantagens:** A tabela de Guerra tem o número de colunas reduzido drasticamente, poupando espaço. Um alto volume de redundâncias é evitado, pois é esperado que o número de Guerras cresça mais rapidamente que o número de colônias.
- **Desvantagens:** Algumas consultas, por exemplo sobre o atributo de uma Colônia envolvida em uma determinada Guerra, terá de ser feita com mais junções, pois as características de Colônia não são armazenadas na tabela de Guerra.
- **Alternativas:** Não utilizar ID sintético, replicando a os identificadores de Colônia na tabela de Guerra.

- **Especialização de 'Pessoa'**

- **Solução adotada:** Foi realizado o mapeamento pela alternativa 1 (vista em aula), isto é, o CEG e os CEEs foram mapeados em tabelas separadas. A aplicação deve garantir a participação total.
- **Vantagens:** Note que esta especialização é de sobreposição, portanto ao separar o CEG em sua própria tabela, evitamos a replicação dos atributos genéricos de Pessoa, e evitamos que a aplicação tenha que constantemente garantir a consistência destes atributos, causando maior eficiência de tempo e espaço.
- **Desvantagens:** A participação total fica a cargo da aplicação garantir. Em contrapartida, o mapeamento pela alternativa 3 (vista em aula) garante a participação total na estrutura. Além disso, a separação dos atributos genéricos em uma tabela separada pode causar mais junções em algumas

consultas, como por exemplo a recuperação do nome de um Piloto, dado o número de sua licença.

- **Alternativas:** Mapear pela alternativa 3, isto é, criar tabelas apenas para os CEEs.

- **Especialização de ‘Corpo’**

- **Solução adotada:** Foi realizado o mapeamento pela alternativa 1 (vista em aula), isto é, o CEG e os CEEs foram mapeados em tabelas separadas. A aplicação deve garantir a participação total, a disjunção e a consistência dos atributos de critério.
- **Vantagens:** Esta especialização é de disjunção, portanto ao separar o CEG em sua própria tabela, não estamos evitando a replicação dos atributos genéricos, pois cada CEE possui apenas um tipo. As vantagens, portanto, são diferentes das citadas no caso da especialização de Pessoa. Especificamente, note que nesta especialização, temos relacionamentos com o CEG. Caso o CEG não fosse mapeado em sua tabela própria, estes relacionamentos teriam que ser redirecionados para cada tabela de CEE, causando mais restrições de integridade referencial para o SGBD garantir, bem como mais possíveis junções em consultas. Além disso, a busca por uma instância do CEG é mais eficiente nesta alternativa, pois se não tivéssemos uma tabela separada para o CEG, o SGBD teria que varrer múltiplas tabelas (dos CEEs), as quais podem estar fisicamente distantes em memória.
- **Desvantagens:** A participação total, a disjunção e a consistência dos atributos de critério ficam a cargo da aplicação garantir. Em contrapartida, o mapeamento pela alternativa 3 (vista em aula) garante a participação total na estrutura, e não mapeia os atributos de critério. Além disso, a separação dos atributos genéricos em uma tabela distinta pode causar mais junções em algumas consultas, como por exemplo a recuperação do nome das Estrelas com determinada luminosidade.
- **Alternativas:** Mapear pela alternativa 3, isto é, criar tabelas apenas para os CEEs.

- **Especialização de ‘Planeta’**

- **Solução adotada:** Foi realizado o mapeamento pela alternativa 1 (vista em aula), isto é, o CEG e os CEEs foram mapeados em tabelas separadas.

A aplicação deve garantir a participação total, a disjunção e a consistência dos atributos de critério.

- **Vantagens:** Esta especialização é de disjunção, portanto ao separar o CEG em sua própria tabela, não estamos evitando a replicação dos atributos genéricos, pois cada CEE possui apenas um tipo. As vantagens, portanto, são diferentes das citadas no caso da especialização de Pessoa. Especificamente, note que ao criar uma nova Colônia (ou buscar por Colônias em Planetas) associada a um Planeta, não é necessário varrer a tabela toda de Planeta, conferindo em toda tupla o atributo de critério habitabilidade, pois já existe uma tabela exclusivamente para Planetas habitáveis onde a busca pode ser realizada.
 - **Desvantagens:** A participação total, a disjunção e a consistência dos atributos de critério ficam a cargo da aplicação garantir. Além disso, a separação dos atributos genéricos em uma tabela distinta pode causar mais junções em algumas consultas.
 - **Alternativas:** Mapear pela alternativa 2, isto é, criar apenas uma tabela para o CEG, com um atributo não nulo de critério 'habitável', garantindo a disjunção em aplicação.
- **Agregações 'Expedição' e 'Colônia'**
 - **Solução adotada:** Como o CR gerador não possui atributos próprios, o CR não foi mapeado em uma tabela própria (e sim, mapeado na mesma tabela que a agregação).
 - **Vantagens:** Consultas mais eficientes, pois descartamos uma tabela intermediária, implicando em menos junções.
 - **Desvantagens:** Nenhuma encontrada.
 - **Alternativas:** Criar tabelas separadas para o CR e para a agregação.
- **CRs 1:N 'pilota' e 'orbita'**
 - **Solução adotada:** Como em nenhum CR 1:N houve caso de pouca participação (pelo contrário, todos possuem participação total), optamos por não criar uma tabela separada para o CR. Garantimos a participação total na estrutura, com o uso de not nulls.

- **Vantagens:** Menos junções em consultas, portanto consultas mais eficientes. Participação total garantida na estrutura com restrição not null.
- **Desvantagens:** Caso não houvesse participação total do CE do lado de cardinalidade N, uma desvantagem deste mapeamento seria a possibilidade de eventuais valores nulos na tabela do CE do lado de cardinalidade N. Como não é o caso, não foram encontradas desvantagens.
- **Alternativas:** Mapear o CR em uma tabela separada, com referência para ambos os CEs envolvidos, garantindo participação total na aplicação.
- **CR N:N 'participa'**
 - **Solução adotada:** O CR foi mapeado em sua própria tabela, com PK composta por chaves estrangeiras referenciando as CEs participantes do relacionamento.
 - **Vantagens:** A cardinalidade do relacionamento é preservada na estrutura.
 - **Desvantagens:** Não encontrada.
 - **Alternativas:** Não encontrada.
- **CRs N:N 'visita' e 'compõe'**
 - **Solução adotada:** O CR foi mapeado em sua própria tabela, com PK composta por chaves estrangeiras referenciando as CEs participantes do relacionamento. A participação total deve ser garantida na aplicação.
 - **Vantagens:** A cardinalidade do relacionamento é preservada na estrutura.
 - **Desvantagens:** A participação total não é garantida na estrutura.
 - **Alternativas:** Não encontrada.
- **Entidades fracas 'Guerra' e 'Corpo'**

- **Solução adotada:** A solução usual: o CE forte e fraco foram mapeados em tabelas separadas, com a PK do CE fraco composta parcialmente por chave estrangeira = PK do CE forte.
- **Vantagens:** O mapeamento preserva as cardinalidades, a participação total, e a unicidade da chave fraca, dado um valor de chave do owner.
- **Desvantagens:** Nenhuma encontrada.
- **Alternativas:** Nenhuma encontrada.

3.3. Restrições

- Todas as restrições de integridade identificadas na modelagem conceitual ainda precisam ser tratadas em aplicação (ciclos e potenciais inconsistências).
- Em entidades com atributos data/hora de início e fim, deve ser garantido que $D/H-I < D/H-F$.
- Os valores de raio, massa, gravidade, temperatura e luminosidade de um corpo devem pertencer a um intervalo fisicamente razoável de valores, a ser estabelecido por cientistas qualificados.

3.4. Correções da Entrega 2

Todos os erros identificados pelo monitor foram corrigidos. Especificamente:

- A estética do modelo relacional foi alterada, eliminando as várias setas curvadas e intersectantes que dificultavam a leitura
- Na justificativa do CR N:N 'participa', foi eliminada a menção de desvantagem, como sugerido (não há alternativas viáveis, portanto não há desvantagens);
- Na justificativa das especializações de 'Corpo' e de 'Planeta', foi acrescentado às desvantagens que a aplicação também deve garantir a disjunção;
- Na justificativa das especializações de 'Corpo' e de 'Planeta' foram separadas em duas justificativas, pois as de Corpo diferem das de Planeta, apesar do mapeamento proposto ser o mesmo;

Além disso, foram feitas correções que passaram despercebidas, como fazer o telefone ser chave secundária de pessoa e fazer o atributo Nome de Colônia ser NOT NULL.

4. Implementação (github.com/diksown/vuei)

4.1. Consultas

- **Consulta 1:** Expedições disponíveis para reserva.

```
SELECT EX.NAVE, EX.DH_INICIO, ((NV.CAPACIDADE - 1) - COUNT(ET.TURISTA)) AS VAGAS_RESTANTES
FROM EXPEDICAO EX JOIN NAVE NV
  ON EX.NAVE = NV.NUMERO_REGISTRO
LEFT JOIN EXPEDICAO_TURISTA ET
  ON ET.NAVE = EX.NAVE AND ET.DH_INICIO = EX.DH_INICIO
GROUP BY EX.NAVE, EX.DH_INICIO, NV.CAPACIDADE
HAVING COUNT(ET.TURISTA) < NV.CAPACIDADE - 1
AND CURRENT_DATE < EX.DH_INICIO;
```

Esta consulta seleciona todas as expedições cadastradas que ainda não partiram e que não possuem capacidade lotada. É uma consulta necessária para determinar quais expedições estão disponíveis para um turista participar. Adicionalmente, para cada expedição disponível para reserva, é exibida a quantidade de vagas restantes. O uso de uma consulta aninhada correlacionada, varrendo a tabela expedição turista para cada expedição é possível, mas uma forma mais eficiente dessa consulta é dada acima.

- **Consulta 2:** Rotas seguras.

```
SELECT DISTINCT RC.ROTA
FROM GUERRA GR JOIN COLONIA CL
  ON GR.AGRESSOR = CL.ID OR GR.DEFENSOR = CL.ID
RIGHT JOIN ROTA_CORPO RC
  ON RC.GALAXIA_CORPO = CL.GALAXIA AND RC.NOME_CORPO = CL.NOME_PLANETA
WHERE GR.AGRESSOR IS NULL
ORDER BY RC.ROTA;
```

Esta consulta seleciona todas as rotas que visitam apenas corpos que nunca estiveram em guerra. É uma consulta útil para determinar os itinerários recomendados ao turista preocupado com a preservação de sua vida.

- **Consulta 3:** Raças mais perigosas.

```

SELECT RC.NOME, COUNT(GR.AGRESSOR) AS QUANTIDADE_GUERRAS, ROUND(AVG(GR.OBITOS)) AS MEDIA_OBITOS,
CASE
    WHEN AVG(GR.OBITOS) > 1000 THEN 'ALTO'
    WHEN AVG(GR.OBITOS) <= 1000 AND AVG(GR.OBITOS) > 500 THEN 'MEDIO'
    WHEN AVG(GR.OBITOS) <= 500 AND AVG(GR.OBITOS) > 0 THEN 'BAIXO'
    ELSE 'INEXISTENTE'
END AS PERIGO
FROM GUERRA GR JOIN COLONIA CL
ON GR.AGRESSOR = CL.ID
JOIN FACCAO_RACA FR
ON FR.FACCAO = CL.FACCAO_NOME
RIGHT JOIN RACA RC
ON RC.NOME = FR.RACA
GROUP BY RC.NOME
HAVING ROUND(AVG(GR.OBITOS)) >= 0 OR AVG(GR.OBITOS) IS NULL
ORDER BY AVG(GR.OBITOS) DESC NULLS LAST;

```

Esta consulta seleciona a quantidade de guerras e o número médio de óbitos em guerras das quais cada raça participou ou participa como agressora (inclusive para as raças que nunca participaram ou foram agressoras em alguma guerra). Há a possibilidade de filtrar o resultado por nível mínimo de perigo (i.e. o valor mínimo permitido para a média de óbitos em guerras que a raça em questão começou), que é fixado em 0 por default, de modo a incluir até as raças mais pacíficas. Além disso, as raças são classificadas de acordo com 4 possíveis níveis de perigo: alto, médio, baixo ou inexistente.

- **Consulta 4:** Estatísticas para o setor de marketing.

```

SELECT EX.ROTA,
MAX(EXTRACT(YEAR FROM AGE(CURRENT_DATE, PS.DATA_NASC))) AS IDADE_MAX,
MIN(EXTRACT(YEAR FROM AGE(CURRENT_DATE, PS.DATA_NASC))) AS IDADE_MIN,
ROUND(AVG(EXTRACT(YEAR FROM AGE(CURRENT_DATE, PS.DATA_NASC)))) AS IDADE_MEDIA
FROM EXPEDICAO EX JOIN EXPEDICAO_TURISTA ET
ON EX.NAVE = ET.NAVE AND EX.DH_INICIO = ET.DH_INICIO
JOIN PESSOA PS
ON PS.EMAIL = ET.TURISTA
GROUP BY EX.ROTA
ORDER BY IDADE_MEDIA;

```

Esta consulta seleciona as faixas etárias (máximo, mínimo e média entre as idades) dos turistas que já viajaram por cada rota, para que o setor de marketing possa futuramente realizar melhores recomendações de rotas a cada indivíduo (e.g. evitar recomendar a “rota da vida selvagem” a turistas acima de 60 anos de idade). Obs. rotas que ainda não foram desbravadas por turistas não são retornadas.

- **Consulta 5:** Pilotos experientes em uma rota.

```
SELECT DISTINCT PS.NOME, PS.EMAIL
FROM PESSOA PS
JOIN EXPEDICAO EX ON PS.EMAIL = EX.PILOTO
WHERE NOT EXISTS (
    (
        SELECT EX1.NAVE, EX1.DH_INICIO
        FROM EXPEDICAO EX1
        WHERE (UPPER(EX1.ROTA) = 'ROTA DA AVENTURA') AND
              (EX1.DH_INICIO BETWEEN TIMESTAMP '2021-01-01' AND TIMESTAMP '2023-12-31')
    )
    EXCEPT
    (
        SELECT EX2.NAVE, EX2.DH_INICIO
        FROM EXPEDICAO EX2
        WHERE EX2.PILOTO = EX.PILOTO
    )
);
```

Esta consulta seleciona todos os pilotos que, em um dado período de tempo especificado, pilotaram todas as expedições para uma determinada rota. Esta consulta pode ser usada para melhor informar (ou automatizar) as criações das escalas de trabalho dos pilotos, indicando os pilotos mais qualificados para guiar cada expedição. Obs. esta consulta implementa a divisão relacional.

4.2. Aplicação

A aplicação foi escrita na linguagem Python na versão 3.10 e para o SGBD utilizou-se o PostgreSQL. Para realizar a conexão com a base de dados foi utilizada a biblioteca de python [Psycopg2](#) e a biblioteca [Rich](#) para a formatação das tabelas no console, além da utilização da biblioteca nativa [cmd](#) para facilitar a leitura e a navegação pelo terminal. As funções implementadas pela aplicação são:

- Listar expedições

```
def do_listar_expedicoes(self, arg):
    """Lista expedições que ainda não partiram
    e que possuem vagas. (1º do consultas.sql)"""
    try:
        table = self.pretty_query(
            """
            SELECT
                EX.ROTA,
                EX.NAVE,
                EX.DH_INICIO,
                ((NV.CAPACIDADE - 1) - COUNT(ET.TURISTA))
                AS VAGAS_RESTANTES
            FROM EXPEDICAO EX JOIN NAVE NV
            ON EX.NAVE = NV.NUMERO_REGISTRO
            LEFT JOIN EXPEDICAO_TURISTA ET
            ON ET.NAVE = EX.NAVE AND ET.DH_INICIO = EX.DH_INICIO
            GROUP BY EX.NAVE, EX.DH_INICIO, NV.CAPACIDADE
            HAVING COUNT(ET.TURISTA) < NV.CAPACIDADE - 1
            AND CURRENT_DATE < EX.DH_INICIO;"""
        )
        console.print(table)
    except Exception as e:
        print(e)
```

Essa função encaminha uma query de seleção de todas as expedições disponíveis, isto é as expedições que ainda não saíram e não estão lotadas, dessa forma o usuário pode escolher de qual expedição ele deseja participar e também se alguma expedição que ele desejava participar ainda está disponível.

- Listar turistas

```
def do_listar_turistas(self, arg):  
    """Lista todos os turistas."""  
    try:  
        table = self.pretty_query(  
            """  
            SELECT NOME, TURISTA.EMAIL  
            FROM TURISTA JOIN PESSOA  
            ON TURISTA.EMAIL = PESSOA.EMAIL  
            ORDER BY NOME;  
            """  
        )  
        console.print(table)  
    except Exception as e:  
        print(e)
```

Essa função retorna uma tabela com o nome e o email de todos os turistas cadastrados na base de dados.

- Inserir turista em expedição

```
def do_inserir_turista_em_expedicao(self, arg):
    """Insere um turista em uma expedição."""
    try:
        turista = questionnaire.text("Email do turista:").unsafe_ask()
        nave = questionnaire.text("Número de registro da nave:").unsafe_ask()
        dh_inicio = questionnaire.text(
            'Data e hora de início da expedição (formato "YYYY-MM-DD HH24:MI:SS"):'
        ).unsafe_ask()
    except KeyboardInterrupt:
        return

    data = rr(nave, dh_inicio, turista)

    try:
        self.cur.execute(
            """
            INSERT INTO EXPEDICAO_TURISTA (NAVE, DH_INICIO, TURISTA)
            VALUES (%s, %s, %s)
            """,
            data,
        )
        self.conn.commit()
        print(
            f"[bold green]Turista {turista} registrado na expedição "
            + "(NAVE='{nave}', DH_INICIO='{dh_inicio}').[/bold green]"
        )
    except psycopg2.Error as error:
        a = f"Erro ao inserir turista na expedição: {error}"
        print(a)
        self.conn.rollback()
```

Essa função cadastrar os turistas em uma expedição. O funcionamento dela consiste em pedir para o usuário inserir os dados de **email do turista**, o **número de registro da nave** e a **data e hora de início da expedição**, após isso é realizado um insert, com os dados fornecidos. Se qualquer dado estiver inconsistente ou inválido a função retornará um erro e fará um rollback.

- Função auxiliar

```
def query_to_rich_table(cursor: cursor, query: str):
    cursor.execute(query)
    columns = [desc[0] for desc in cursor.description]
    data = cursor.fetchall()

    table = Table(show_header=True, header_style="bold magenta")
    for col in columns:
        table.add_column(col)

    for row in data:
        table.add_row(*[str(item) for item in row])

    return table
```

A forma como as consultas ocorrem é chamando a função *pretty_query* que por sua vez chama a função *query_to_rich_table* (mostrada na figura abaixo), que faz uso das funções **cursor** para navegar pela base de dados e **table** para formatar as saída. Essa última parte é apenas para deixar as tabelas mais bonitas.

5. Conclusão

5.1. Análise

Nossa experiência com o projeto foi geralmente positiva. Tivemos uma ótima introdução à disciplina de base de dados, e os monitores se destacaram por sua competência e disposição para esclarecer dúvidas. Eles também foram extremamente eficientes na correção dos trabalhos, fornecendo feedbacks construtivos e rápidos.

No entanto, os membros do grupo tiveram **diversos pontos de desgaste** com a disciplina em si, o que influenciou nossa motivação com o projeto e na matéria como um todo.

Para contexto, todos os cinco membros do nosso grupo eram veteranos do 8º e 10º período do BCC com um semestre relativamente leve, alguns tendo somente essa matéria. Mesmo considerando isso, a disciplina ainda foi exaustiva para os integrantes.

No 5º período, quando a matéria é usualmente cursada, a carga de outras disciplinas é significativamente maior, e quatro dos cinco membros foram reprovados nesse período (dois por 0.2 pontos...), realçando sua dificuldade.

5.2. Críticas

- **Falta de transparência com as notas**

As notas da P1 (outubro) não saíram antes da P2 (dezembro). Isso gerou ansiedade e incerteza, pois não sabíamos nossa situação quanto à aprovação. Esse acontecimento é recorrente, tendo acontecido em anos anteriores e colaborado para a reprovação prévia de membros do grupo.

- **Inflexibilidade**

Não foram oferecidas provas substitutivas nem atividades para ganho de pontos extras.

- **Material incompleto**

Os slides são incompletos ou confusos quando não acompanhados da aula e as listas de exercícios não possuem solução. É importante ressaltar que isso afeta até os alunos que vão em todas as aulas e nos atendimentos.

- **Extensão**

As provas eram extremamente longas e cansativas.

- **Aulas fora do horário da disciplina**

A matéria já é extensa, ter que assistir aula gravada no tidia dificulta ainda mais o gerenciamento de tempo entre as outras disciplinas do semestre, especialmente de um assunto exigido na prova.

5.3. Sugestões

- **Agilizar a correção das provas**
- Dar atividades / questões valendo ponto extra para as provas
- Aplicar uma “sub do bem”
- Melhorias no material de aula