

Trabalho - Etapa 1

INF01145 - FUNDAMENTOS DE BANCO DE DADOS - Professora Karin Becker
Kim Colpo Enghusen e Victor Machado Gonçalves

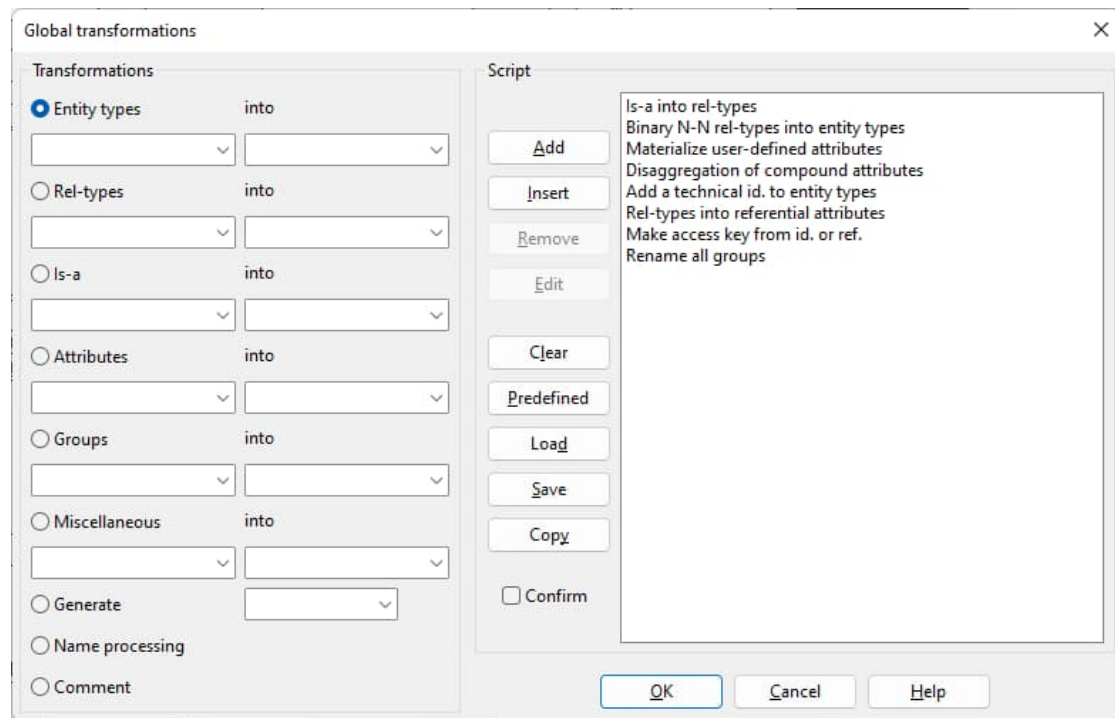
Mapeamentos

O mapeamento foi facilitado com o uso da ferramenta CASE DB-Main. Aprendemos bastante com a ferramenta e aprendemos a como montar um modelo conceitual e lógico, assim como estabelecer restrições básicas pela própria ferramenta.

(Novo) Para explicar como o DB-Main mapeou o modelo conceitual e quais regras foram utilizadas, trazemos o seguinte *workflow*, na mesma ordem:

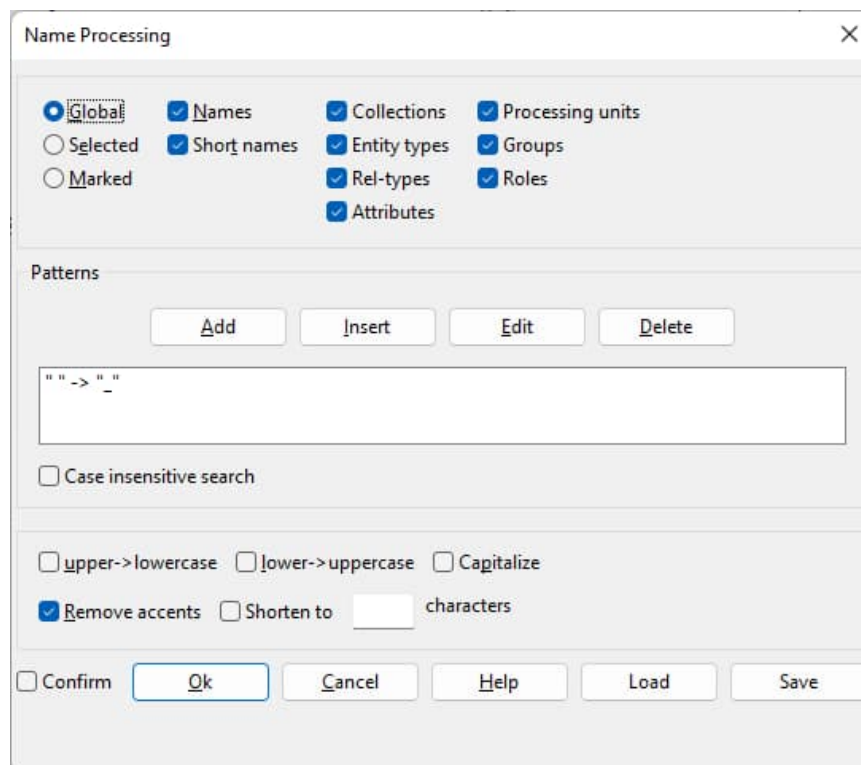
Etapa	Transformação	Explicação
1	Renomeação	Eliminar caracteres especiais e eliminar espaços. Facilita a produção do modelo físico(SQL).
2	HE → Relacionamento	Mapeamento da Hierarquia de Especialização em Relacionamentos.
3	Relacionamentos N-N → Entidade	Transformar relacionamentos com card. máx. N-N em entidades
4	Materializar atributos	Transformar atributos de tipo personalizado em atributos primitivos.
5	Desagregar atributos compostos	Aplainamento de atributos compostos.
6	Adicionar ID técnico	Gerar atributos de identificação para entidades que não possuem chave primária.
7	Relacionamentos 1-N → Atributos referenciais	Transformar relacionamentos card. máx. 1-N(e 1-1) em atributos referenciais.
8	Produzir chaves de acesso	Preparar modelo relacional para o modelo lógico com a definição de chaves de acesso a partir das chaves primárias e estrangeiras.
9	Renomear grupos(restrições)	Renomear restrições em nomes apropriados e únicos.

Enquanto no DB-Main:



1. Renomeação

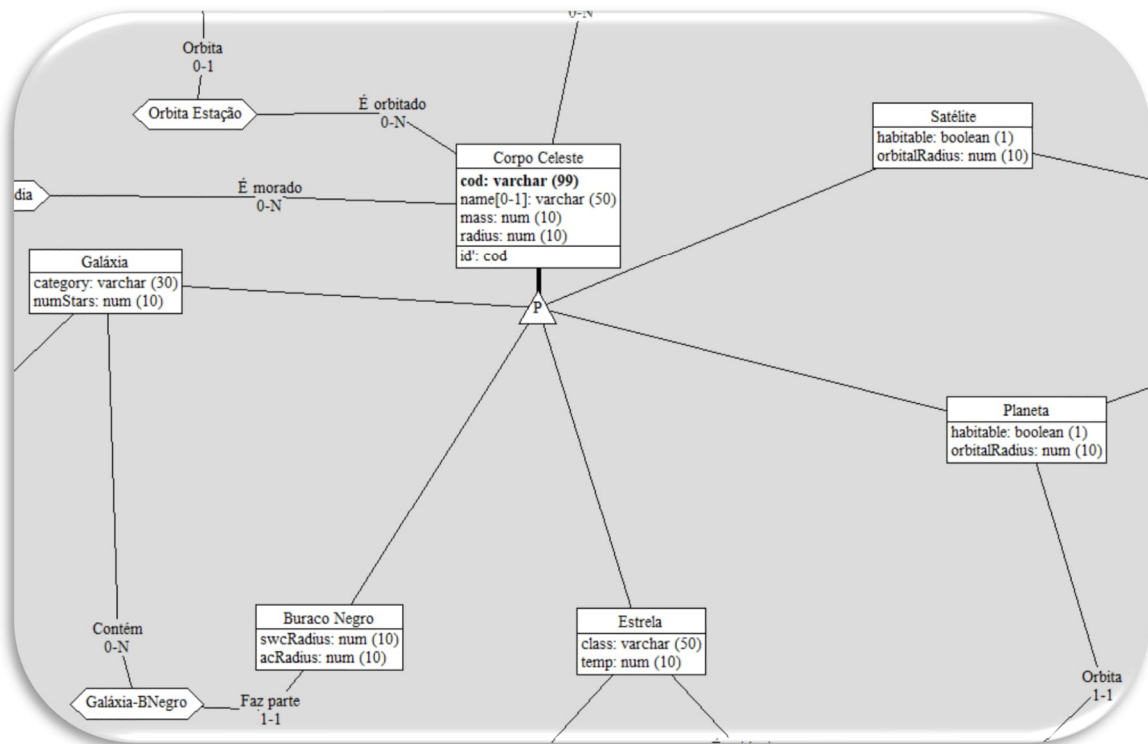
Fizemos uma operação simples. Eliminamos acentos e espaços são substituídos por um símbolo de sublinhado "_".



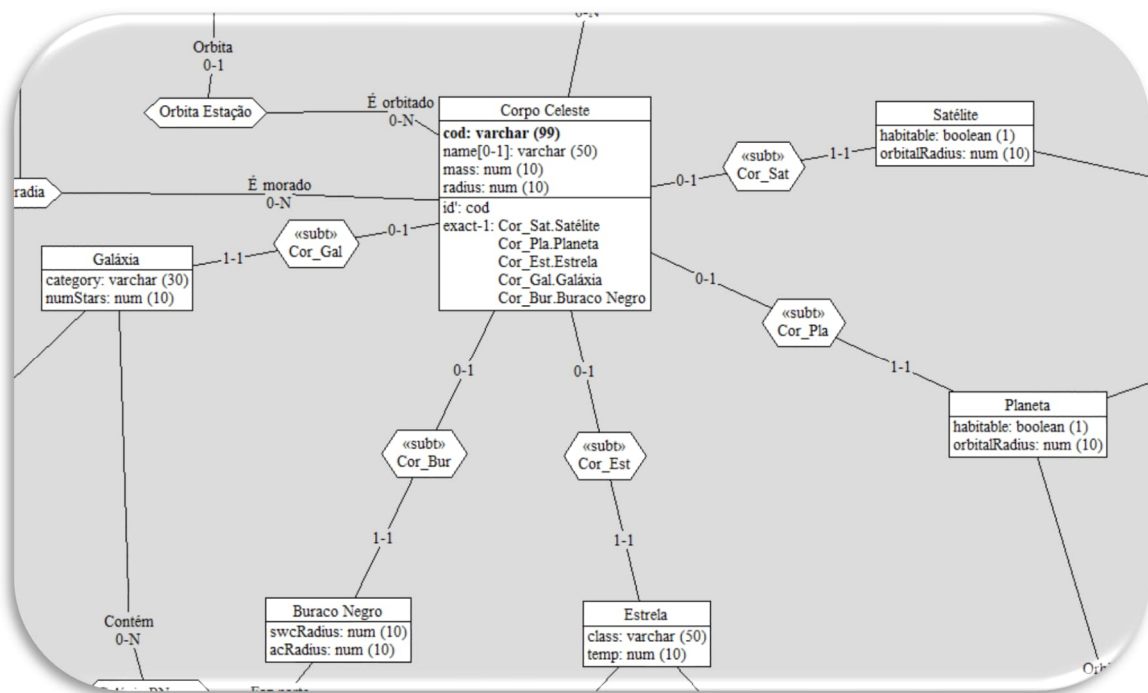
2. HE → Relacionamento

A hierarquia de especialização do tipo partição (total, exclusivo) ou, pela nomenclatura do DB-Main, total e disjunta foi mapeada convertendo em relacionamentos de cardinalidade máxima 1-1.

Visualmente, temos a seguinte transformação, partindo do modelo conceitual:



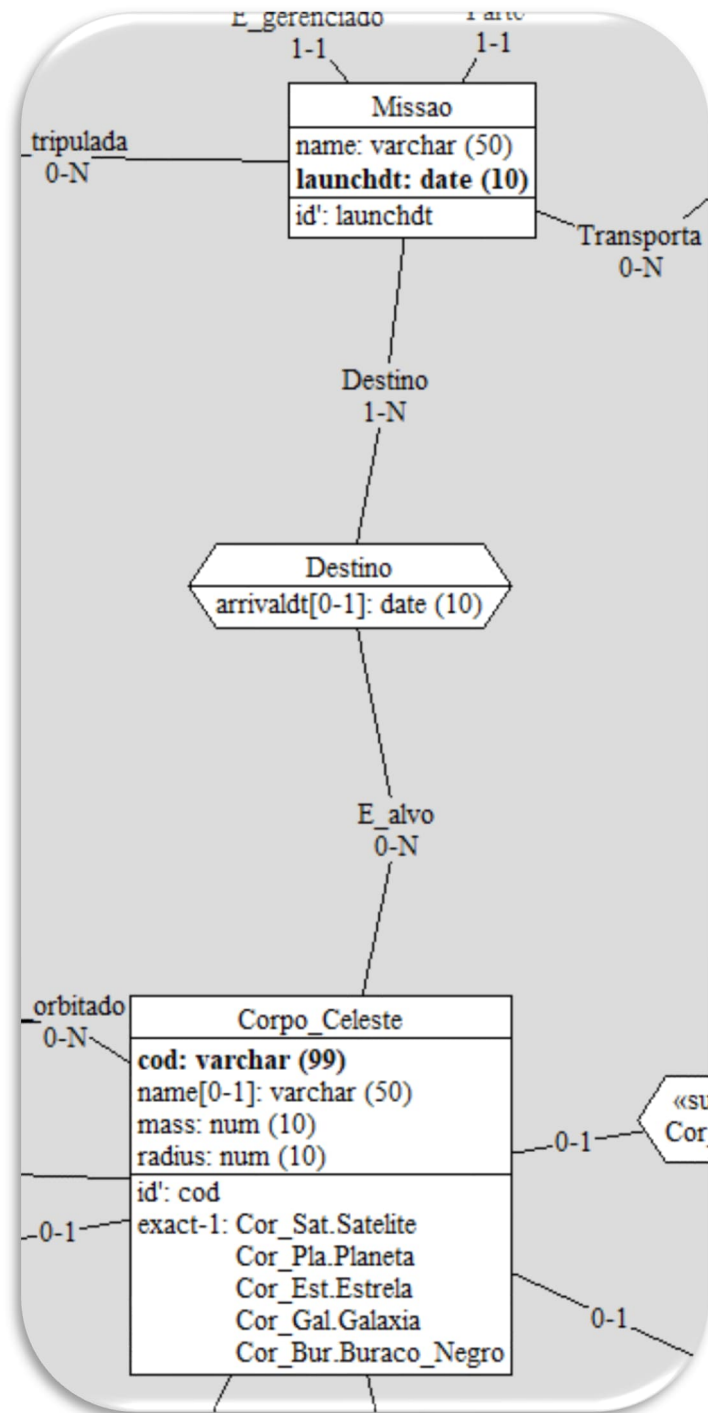
Aplicando:



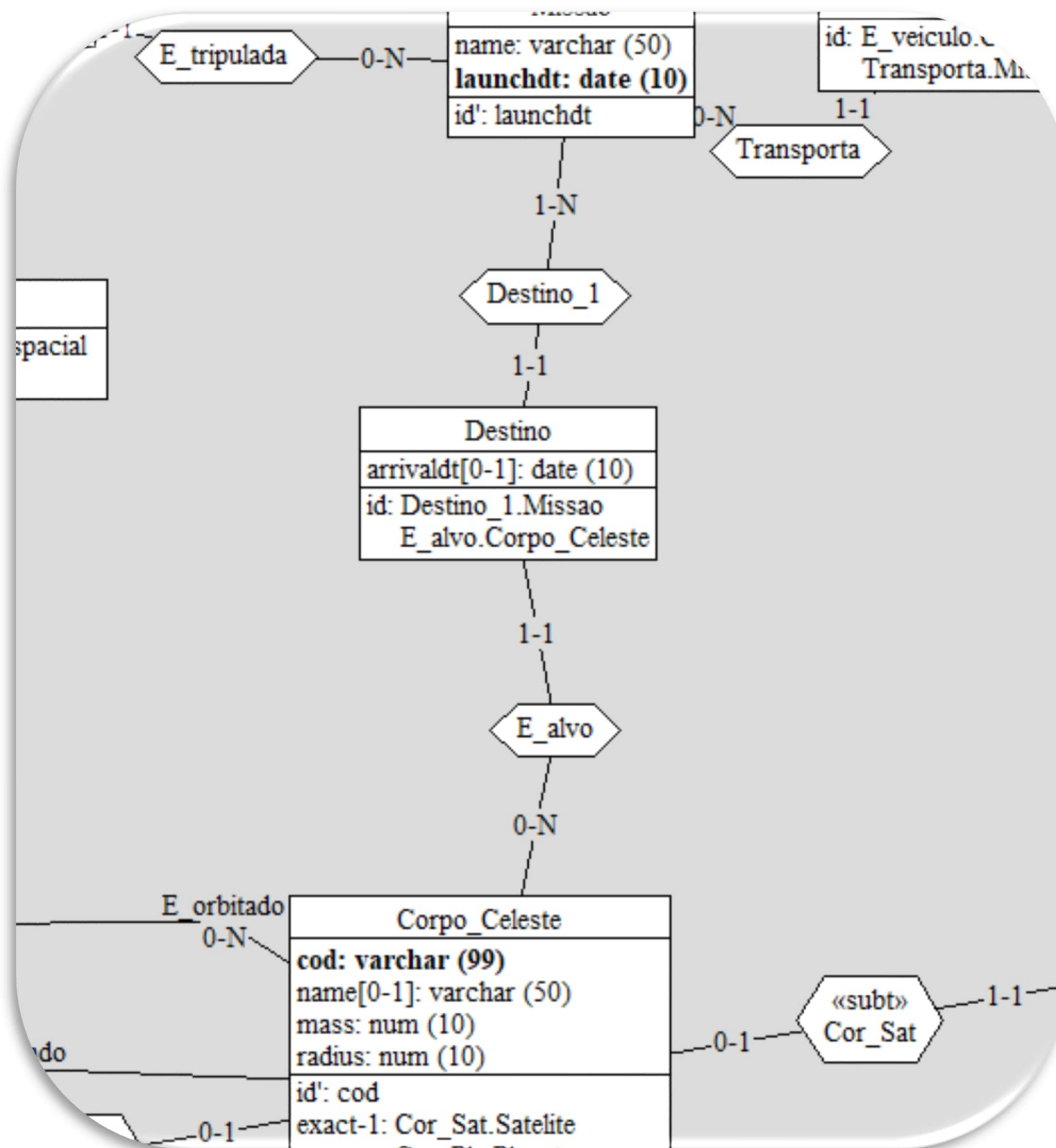
Observação: essas relações serão mapeadas em atributos referenciais como descrito na etapa 7.

3. Relacionamentos N-N → Entidade

Todos os relacionamentos N-N foram mapeados em entidades. Além disso, as relações N-N com atributos terão, na entidade mapeada, os mesmos atributos (somente uma relação, Destino). Visualmente, temos a seguinte operação usando Destino como exemplo.



Mapeando:

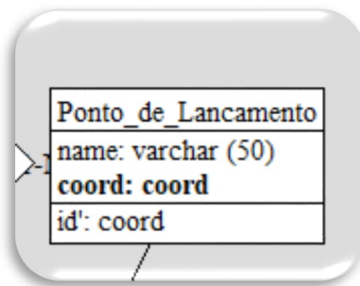


Repare que foram criadas duas relações 1-N que, mais tarde, serão atributos referenciais.

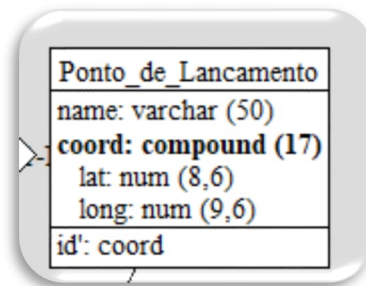
4. Materializar atributos

Esta etapa não é muito importante, mas foi preciso já que foi definido um atributo composto coord. O atributo não estava presente na primeira entrega, já que isso corrige um problema na definição da restrição de unicidade para o conjunto (lat,long). De qualquer jeito, a transformação fez o seguinte:

Antes



Depois



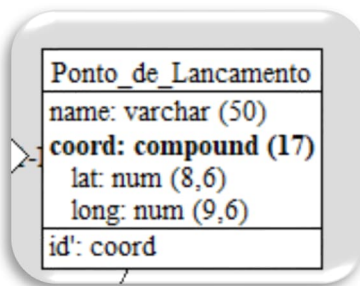
Ou seja, transformamos em um atributo composto.

Observação: Tivemos problema ao criar uma chave secundária(unicidade) para atributos compostos no DB-Main, mas definindo um tipo "coord" tornou possível.

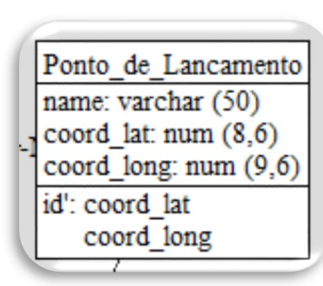
5. Desagregar atributos compostos

Uma sequência natural para a etapa anterior, aqui estamos aplainando o atributo composto. Portanto:

Antes



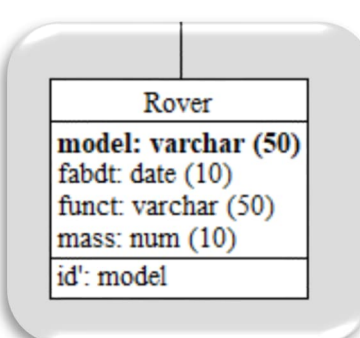
Depois



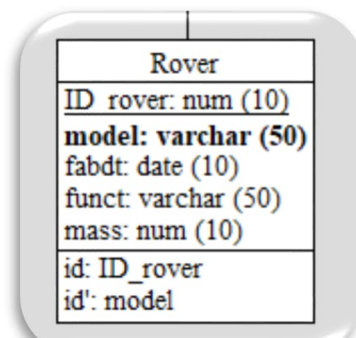
6. Adicionar ID técnico

Como podemos observar pelo modelo ER, não foi proposto nenhum candidato para chave primária. Assim, geramos um atributo de identificação para todas as entidades que não possuem candidatos a chaves primárias. Ou seja, usando a entidade Rover como exemplo:

Antes



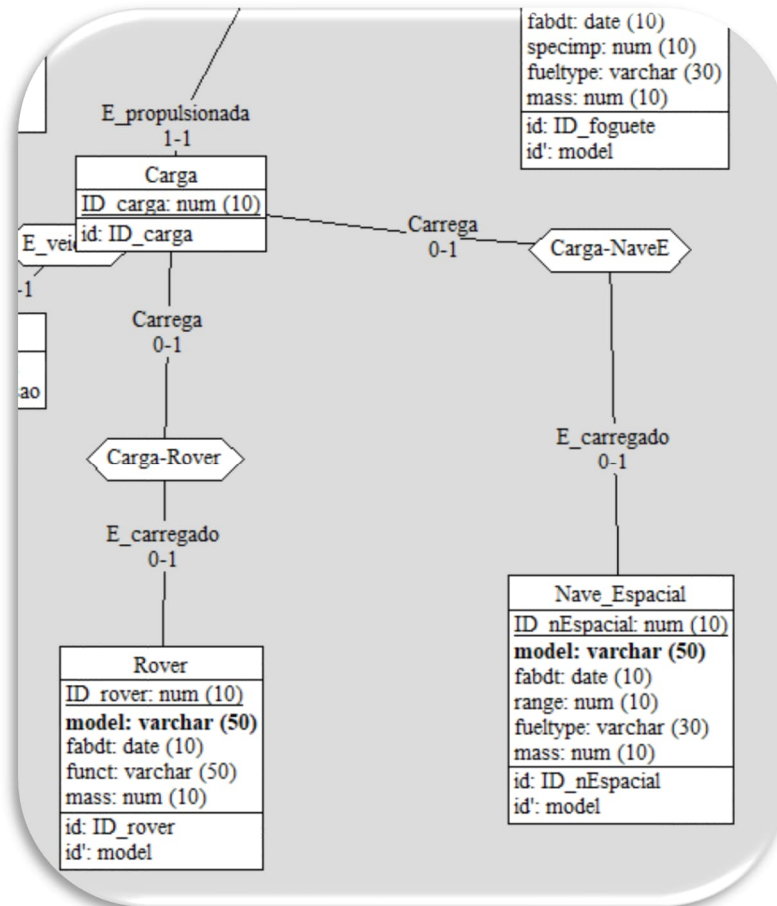
Depois



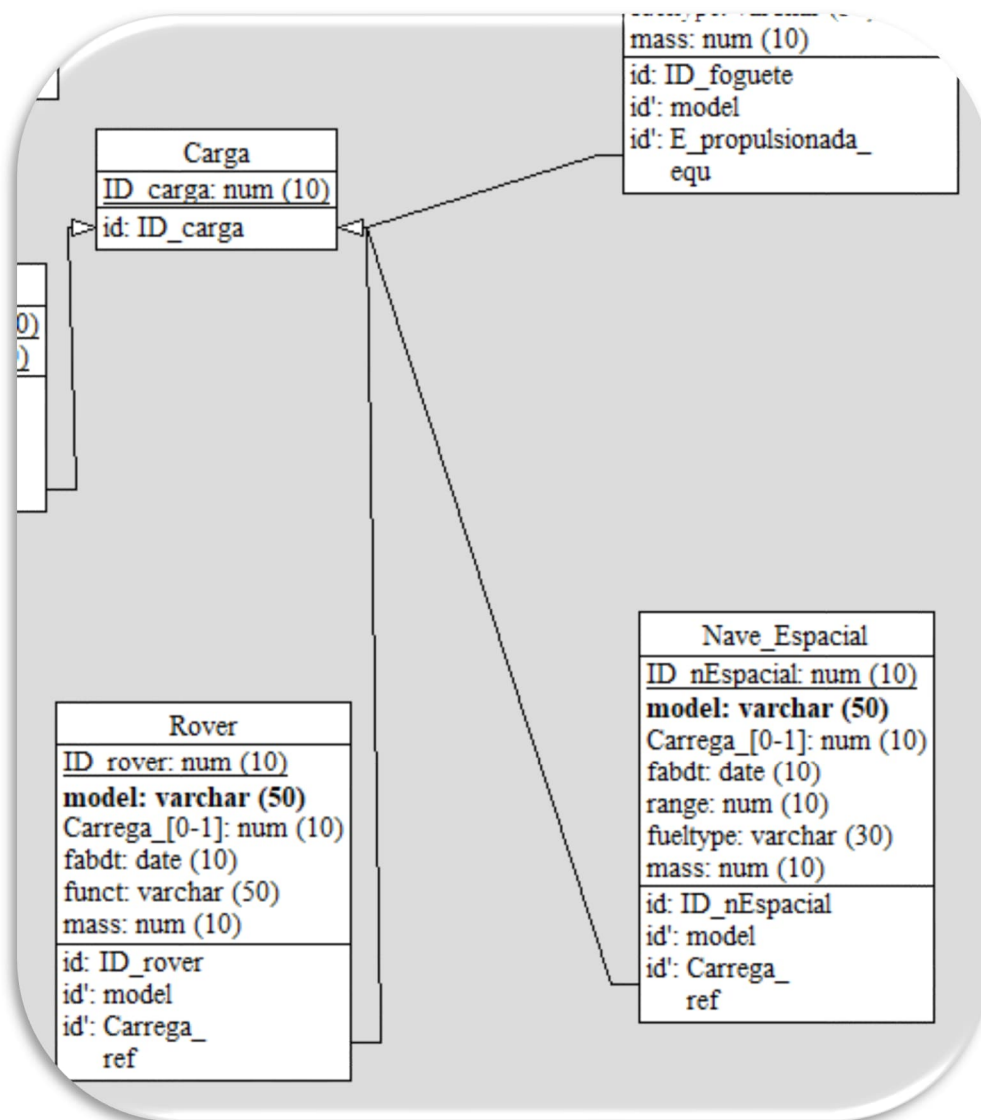
Observação: houve problemas ao usar o tipo seq/serial na geração do código SQL. Assim, utilizamos o tipo num.

7. Relacionamentos 1-N → Atributos referenciais

Para encerrar o mapeamento de relacionamentos, mapeamos relacionamentos 1-N e 1-1 usando atributos referenciais. Ou seja, foi incluído atributos em uma entidade associada a um relacionamento. A entidade escolhida foi ao critério da ferramenta.



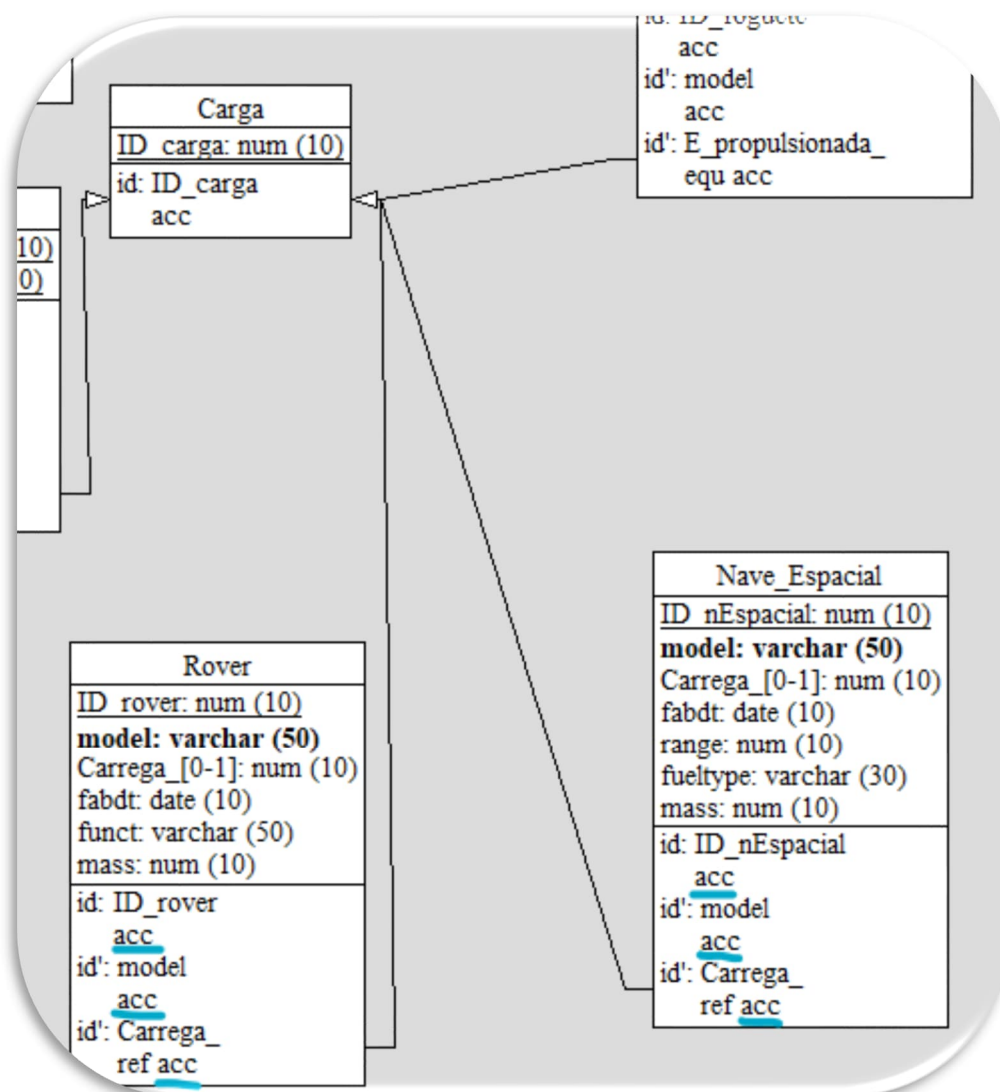
Mapeando:



Observe que foi definido implicitamente chaves estrangeiras.

8. Produzir chaves de acesso

Não é importante para o modelo lógico, mas colocamos para facilitar a conversão em um modelo físico (PostgreSQL, SQLite etc.) pelo DB-Main. Aplicando a transformação na mesma parte exemplificada na etapa 7:



Repare que foi incluído um tag *acc*.

9. Renomear grupos

Por último, renomear os grupos gerados até então pela ferramenta que, para o DB-Main, podemos pensar como a renomeação de restrições. É necessário para evitar conflitos na transformação ao código SQL. Não há uma representação visual da operação.