

Prática de Projeto de Banco de Dados com MER

Introdução

O MER – Modelo Entidade Relacionamento é a ferramenta usada para desenvolver um projeto de banco de dados. Existem 3 níveis de modelagem: conceitual, lógico e físico. Cada um desses níveis mostra uma parte de como o projeto de banco de dados deve ser desenvolvido.

Objetivos da aula

- Desenvolver a modelagem Entidade Relacionamento no nível conceitual;
- Realizar a engenharia de requisitos para fazer a definição do escopo;
- Implementar A modelagem Entidade Relacionamento no nível lógico com visualização de Primarys Keys, Foreigns keys e cardinalidades;
- Desenvolver os scripts para implementação do banco de dados com o modelo físico.

Resumo

Modelagem Conceitual

A modelagem conceitual é uma importante fase do projeto de banco de dados e modelar corretamente pode influenciar diretamente na qualidade. É preciso entender bem o conceito além de aprender como modelar corretamente um contexto.

Uma das fases do projeto de um banco de dados é a modelagem conceitual, que ocorre logo em seguida após o levantamento de requisitos. Essa fase é de importância ímpar para o projeto pois poderá influenciar diretamente a qualidade do projeto final.

O modelo conceitual foi proposto por Peter Chen em 1976, mas sofreu muitas atualizações ao longo do tempo. Não é incomum encontrar uma modelagem diferente da que aprenderemos aqui, o importante é entender os conceitos de que cada símbolo representa. Lembre-se, também, que essa fase de modelagem é totalmente independente da tecnologia, ou seja, não importa qual banco de dados será utilizado no projeto, mas sim as entidades, atributos e relacionamentos em questão.

Essa fase gera um artefato (documento) para o projeto que é o diagrama entidade-relacionamento e gerar este diagrama corretamente é uma tarefa importante desta etapa do projeto. Entretanto, antes de conhecer a simbologia específica para isso, é preciso lembrar alguns conceitos importantes.

Essa modelagem baseia-se em entidades, seus atributos e como cada entidade relaciona-se com as demais. Podemos considerar que as entidades se tornarão tabelas nos modelos que derivam desta fase (modelagem lógica e física). Pode-se, ainda, observar entidades como os objetos do “mundo real”, ou seja, os objetos que são importantes para o sistema que estamos construindo. Não se esqueça que as entidades devem ser distintas, ou seja, não podem se repetir no universo modelado.

No “mundo real”, ou seja, onde nosso projeto está sendo empregado, os objetos possuem, essencialmente, três coisas: nome; características; e comportamentos. O nome é, de fato, o nome do objeto modelado (por exemplo: Livro, Aluno, Médico, Produto, Personagem etc.).

Já os atributos são as características relevantes que nossas entidades podem ter. Por exemplo, num sistema de bibliotecas, o livro pode possuir atributos como “título”, “resumo”, “ISBN”, “autor(es)”, “localização na biblioteca”, etc. Já o mesmo livro, numa loja tem, além destes atributos, o “preço”. Por isso o contexto é muito importante, pois os atributos são apenas os que fazem sentido para tal.

Os comportamentos dos objetos (os métodos) não são tão importantes para o mundo do banco de dados, embora eles possam influenciar diretamente como cada objeto se relaciona com os demais. Por exemplo, se um livro pode ser emprestado num sistema de biblioteca, então sabemos que existe um relacionamento direto de livro com alguma entidade que o recebe, como por exemplo, “aluno” e este relacionamento possui uma série de características, como prazo de empréstimo, multa por atraso etc.

Outro ponto importante aqui é que no modelo conceitual os atributos não possuem um tipo específico, ou seja, não importa se o atributo é do tipo texto ou número inteiro, além das características da segurança do banco de dados também não serem modeladas aqui.

Simbologia da Modelagem Conceitual

Como apresentado anteriormente, essa fase do projeto do banco de dados resulta em uma modelagem conceitual completa do domínio da aplicação que está sendo desenvolvida. Com isso, é preciso aprender a simbologia específica para se representar cada item.

A figura 1 mostra como devem ser modeladas as entidades. Note que apenas um retângulo com o nome da entidade ao centro basta para essa tarefa.

Figura 1 - SEQ Figura * ARABIC 1 - Representação de entidade na modelagem conceitual



Fonte: autoral, 2023.

Agora os atributos devem ser modelados conforme exibidos na figura 2. Note que há dois tipos de atributos, um identificador e um comum. O atributo identificador é aquele cujos dados devem ser únicos (nunca se repetem) e nunca podem ser nulos (sem dados). Por exemplo: a placa de um carro, o CPF de uma pessoa, o ISBN de um livro etc. Já os atributos do tipo “comum” são aqueles que representam dados convencionais dessa entidade. Por exemplo, o nome de uma pessoa (pode se repetir), o modelo de um carro etc.

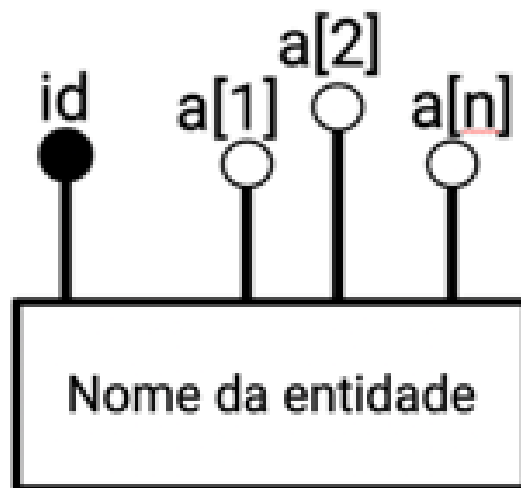
Figura 2 - Atributos na modelagem conceitual



Fonte: autoral, 2023.

Uma vez compreendida as simbologias principais, agora é preciso juntar. A figura 3 representa como os atributos ficam em relação à entidade. Repare que o exemplo genérico mostra que há apenas um único atributo identificador, o que ocorre comumente.

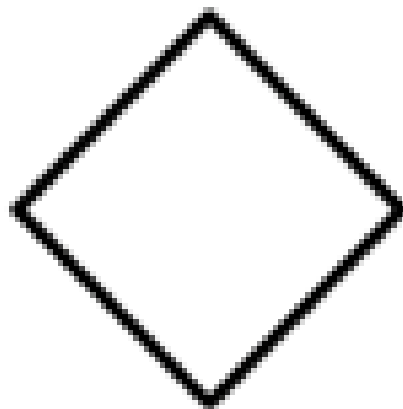
Figura 3 - Atributos na entidade no modelo conceitual



Fonte: autoral, 2023.

Ainda falta mais um símbolo importante, o de relacionamento, neste caso representado na figura 4. Este símbolo ocorre entre duas entidades e denota que entre elas existe um grau de existência de uma em relação à outra. Por exemplo, um livro possui suas próprias características, assim como autor e podemos, portanto, ver estes como entidades distintas. De qualquer maneira, temos que relacionar um livro e um autor e, para isso, utilizamos este símbolo.

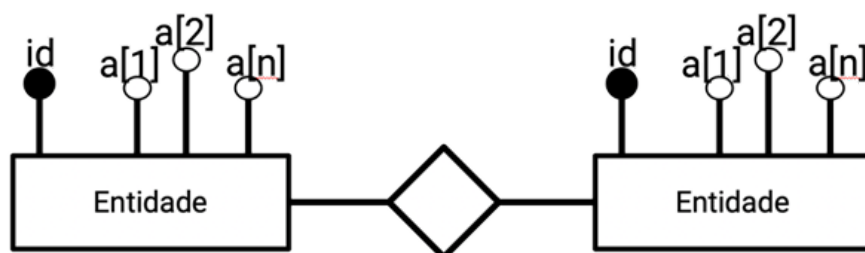
Figura 4 - Relacionamento no modelo conceitual



Fonte: autoral, 2023.

Finalmente, compreendida a simbologia principal da modelagem conceitual, pode-se ver na figura 5 como todos os elementos se conectam, de forma generalista.

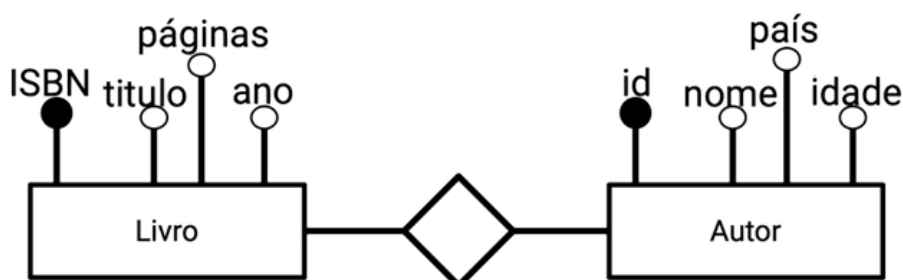
Figura 5 - Entidades, atributos e relacionamentos no modelo conceitual



Fonte: autoral, 2023.

Para ficar um pouco mais claro, vamos usar o mesmo exemplo do livro em relação autor, mostrado na figura 6. O livro, que possui suas próprias características, está associado ao autor que, por sua vez, também possui suas próprias características. Entretanto, ainda falta uma coisa: O grau de relação, ou seja, quantos autores um livro pode ter? Quantos livros um autor pode ter? Isso chama-se cardinalidade e veremos a seguir.

Figura 6 - Exemplo do livro com autor numa modelagem conceitual



Fonte: autoral, 2023.

Cardinalidade

O termo cardinalidade vem da matemática que representa o número de elementos em um determinado conjunto. Por exemplo, o conjunto $X = \{2, 4, 6, 8, 20\}$, então dizemos que a cardinalidade é 5, pois este é o número de elementos dentro deste conjunto.

Na modelagem conceitual o conceito é o mesmo. Quantos elementos de uma entidade referem-se à outra entidade. Toda entidade que se relaciona com outra deve ter uma cardinalidade e, aqui, é representada em mínimo e máximo.

Usando o mesmo exemplo para explicar, de livro, temos a relação de ambos os lados, ou seja, do lado do livro e do lado do autor, conforme:

- O livro, para existir, precisa ter ao menos 1 autor e no máximo vários (indiferente) autores, ou seja, "N".
- O autor, para ser considerado autor de livro, deve ter pelo menos 1 livro e no máximo vários livros, ou seja, "N".

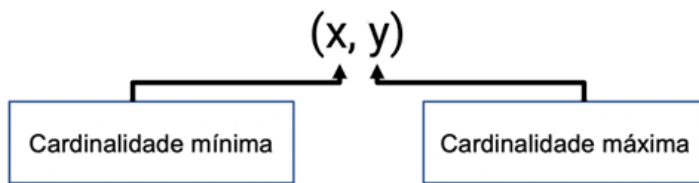
Isso quer dizer que estamos falando de uma relação "1 para N", onde "N", pode ser qualquer número inteiro natural. Lembre-se que na cardinalidade máxima não expressamos um número diferente de 1 ou N.

Para melhorar o entendimento, vamos a outro exemplo: Relação médico X paciente.

- O médico pode ter vários pacientes;
- Cada paciente pode ter vários médicos.

A cardinalidade na modelagem conceitual deve ser representada conforme a figura 7, ou seja, um parêntese, a cardinalidade mínima, uma "vírgula", a cardinalidade máxima e o parêntese de fechamento.

Figura 7 - Representação da cardinalidade no modelo conceitual



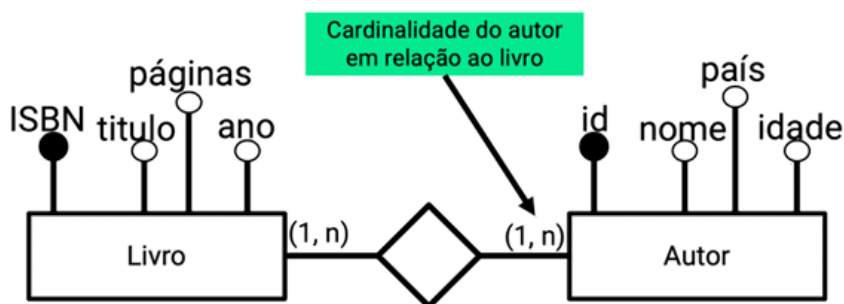
Fonte: autoral, 2023.

A cardinalidade mínima pode ser 0 (zero) ou 1 (um), enquanto a cardinalidade máxima pode ser 1 (um) ou N (“ene”), portanto as possibilidades são:

- **(0, 1)** Indica que a entidade pode ter nenhum ou apenas um de outra
- **(0, n)** Indica que a entidade pode ter nenhum ou vários (mais de 1) de outra
- **(1, 1)** Indica que a entidade precisa ter ao menos um e não mais que isso
- **(1, n)** Indica que a entidade precisa ter ao menos um e no máximo vários

Para ilustrar melhor o conceito e como incluir este grau de relação entre as entidades na modelagem conceitual, vamos continuar explorando o exemplo do livro X autor, conforme demonstrado na figura 8. Note que a cardinalidade ocorre de ambos os lados.

Figura 8 - Exemplo de modelagem conceitual com as cardinalidades.



Fonte: autoral, 2023.

A cardinalidade que está próxima ao livro refere-se a do livro em relação ao autor e que está ao lado do autor refere-se à do autor em relação ao livro. Uma forma bastante usual de descobrir as cardinalidades é fazendo uma pergunta para a entidade e imaginando o que ela responderia: “Quantos você pode ter ao menos e no máximo da sua vizinha?”, por exemplo:

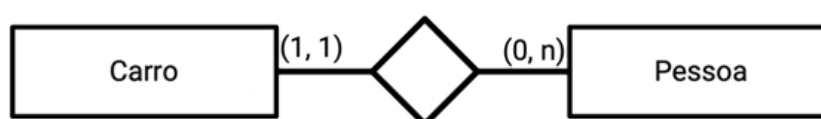
- Livro, você precisa ter quantos autores, pelo menos? Um!
- Livro, você pode ter até quantos autores? Vários

Do outro lado agora...

- Autor, você precisa ter ao menos quantos livros? Um!
- Autor, quantos livros você pode ter? Vários

Outro pequeno exemplo. Carro X Proprietário (chamado de pessoa aqui), mostrado na figura 9. Os atributos aqui foram suprimidos apenas para mostrar o grau de relação entre essas entidades. Note que o carro pode ter no mínimo um proprietário e no máximo um proprietário, também. Por outro lado, a pessoa (proprietário), neste exemplo, pode não ter nenhum carro ou ter vários carros, nesse caso representado por “N”

Figura 9 - Exemplo de cardinalidade entre Carro e Pessoa



Fonte: autoral, 2023.

Técnica de Ida e Volta para estabelecer a cardinalidade dos relacionamentos

Identificar a cardinalidade do relacionamento entre as tabelas é fundamental para determinar a localização da chave estrangeira, lembrando que a chave estrangeira sempre fica do lado do “n”. Para essa tarefa vamos utilizar a técnica de Ida e Volta na leitura do relacionamento. Vamos analisar o exemplo das tabelas Empregado e Departamento.

Por força do negócio, podemos verificar que a tabela Empregado está relacionada com a tabela Departamento, ou seja, um empregado trabalha no departamento e o departamento possui empregados. Mas o problema é a identificação correta da cardinalidade entre essas tabelas. Abaixo, temos os exemplos das tabelas empregado e departamento.

Tabela 1 - Tabelas de Empregados e Departamentos

Empregado			Departamento	
cod_func (pk)	nome_func	sal	cod_depto (pk)	nome_depto
1000	Jose	100	10	comercial
1001	João	120	20	financeiro
1002	Jose	140	30	diretoria
1003	Maria	160		
1004	Joaquim	180		

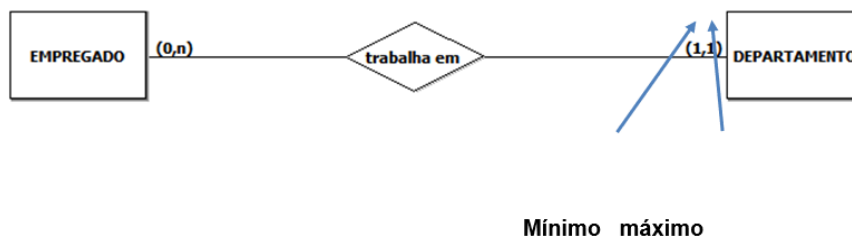
Fonte: Autor, 2023.

Vamos ler o relacionamento no sentido da Ida, ou seja, do empregado para o departamento.

IDA ----->

1 Empregado no mínimo trabalha em 1 Departamento e no máximo em 1 Departamento

Figura 10 - Estabelecendo Relacionamento



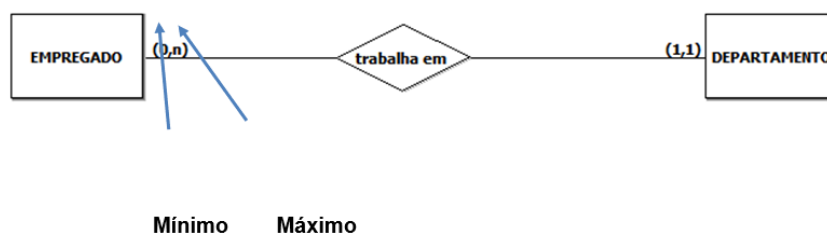
Fonte: autoral, 2023.

Agora, vamos ler o relacionamento no sentido da Volta, ou seja, do departamento para o empregado.

<-----VOLTA

1 Departamento no mínimo tem 0 Empregados e no máximo N Empregados

Figura 11 - Estabelecendo Relacionamento



Fonte: autoral, 2023.

Perceba que o “n” ficou do lado da entidade empregado, portanto é nessa tabela que a chave estrangeira deverá ficar, conforme exemplo abaixo.

Figura 12 - Localização da Chave Estrangeira

Empregado				Departamento	
chave primária			chave estrangeira	chave primária	
cod_func	nome_func	sal	cod_depto	cod_depto	nome_depto
1000	jose	100	10	10	comercial
1001	joao	120	10	20	financeiro
1002	jose	140	20	30	diretoria
1003	maria	160	20		
1004	joaquim	180	30		

Fonte: autoral, 2023.

Note, no exemplo acima, que a função da chave estrangeira é fazer o relacionamento entre as tabelas, daí a importância de estabelecer corretamente a cardinalidade do relacionamento.

Vamos, agora, analisar outro exemplo. A área de negócio é uma escola, abaixo temos a especificação dos requisitos funcionais.

RF01- MANTER DADOS DO PROFESSOR

O sistema deve permitir a inclusão, alteração, exclusão e consulta dos dados dos professores. Sendo que estas operações serão realizadas pela secretária.

RF02- MANTER DADOS DO ALUNO

O sistema deve permitir a inclusão, alteração, exclusão e consulta dos dados dos alunos. Sendo que estas operações serão realizadas pela secretária.

RF03- MANTER DADOS DO CURSO

O sistema deve permitir a inclusão, alteração, exclusão e consulta dos dados dos cursos. Sendo que estas operações serão realizadas pela secretária.

RF04- MANTER DADOS DA MATRÍCULA

O sistema deve permitir a inclusão, alteração e consulta dos dados das matrículas.

RF05- MANTER DADOS DA DISCIPLINA

O sistema deve permitir a inclusão, alteração, exclusão e consulta dos dados das disciplinas. Sendo que estas operações serão realizadas pela secretária.

O primeiro passo é identificar as entidades, como já sabemos, uma entidade é uma representação de um conjunto de dados sobre determinado conceito do sistema. Algo que seja relevante para o negócio e que tenha mais de uma ocorrência. Toda entidade possui atributos, que são os dados que referenciam a entidade. Exemplos de Entidades: CLIENTE, PROFESSOR, ALUNO, CONSULTA, PACIENTE, PEDIDO, etc.

No exemplo da escola, devemos analisar o requisito funcional e identificar as entidades, realizando as seguintes perguntas:

- Para a escola é relevante armazenar os dados dos alunos? Existe mais de um aluno para ser armazenado?
- Para a escola é relevante armazenar os dados dos cursos? Existe mais de um curso para ser armazenado?
- Para a escola é relevante armazenar os dados dos professores? Existe mais de um professor para ser armazenado?
- Para a escola é relevante armazenar os dados das matrículas? Existe mais de uma matrícula para ser armazenada?
- Para a escola é relevante armazenar os dados das disciplinas? Existe mais de uma disciplina para ser armazenada?

Se as respostas dessas perguntas forem “sim”, significa que se trata de uma entidade. Portanto, chega-se à conclusão que ALUNO, PROFESSOR, CURSO, DISCIPLINA e MATRÍCULA são entidades da Escola.

Depois de identificar as entidades, agora é necessário encontrar os atributos de cada entidade, lembrando que os atributos são características que descrevem, identificam, qualificam e/ou quantificam a entidade. São os dados que precisam ser armazenados sobre a entidade”. Exemplos de atributos: código, nome, endereço, sexo, data de nascimento, cpf, RA, valor unitário, etc.

No exemplo do cenário exemplo escola, devemos analisar o requisito funcional e identificar os atributos, realizando a seguinte pergunta:

- Quais dados devem ser armazenados para essa entidade? A resposta dessa pergunta foram os atributos da entidade.

No mínimo, cada entidade deve ter 2 atributos. Então podemos ter os seguintes dados:

Na entidade Aluno, encontramos os atributos: RA (Registro de Aluno), nome do aluno, data de nascimento e gênero.

Na entidade Professor, encontramos os atributos: RP (Registro de Professor), nome do professor, data de nascimento, gênero e titulação.

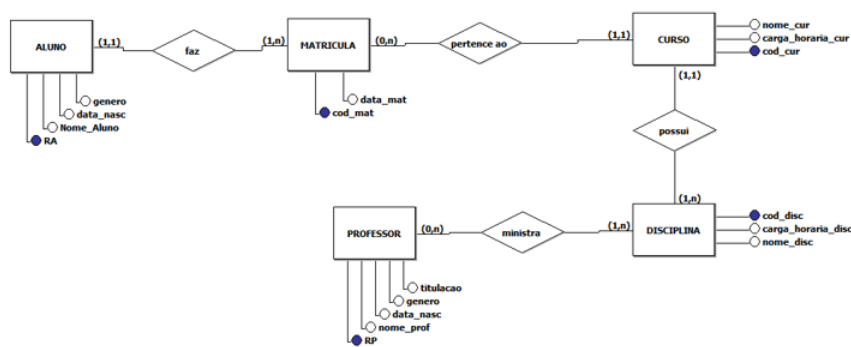
Na entidade Curso, encontramos os atributos: código do curso, nome do curso, carga horário do curso.

Na entidade Disciplina, encontramos os atributos: código da disciplina, nome da disciplina, carga horária da disciplina.

Na entidade Matrícula, encontramos os atributos: código da matrícula, data da matrícula.

Depois de identificar as entidades e os atributos, chegamos ao momento de realizar o modelo conceitual, ou seja, representar as entidades e os seus relacionamentos.

Figura 13 - Modelo Conceitual da Escola



Fonte: autoral, 2023.

Leitura dos Relacionamentos

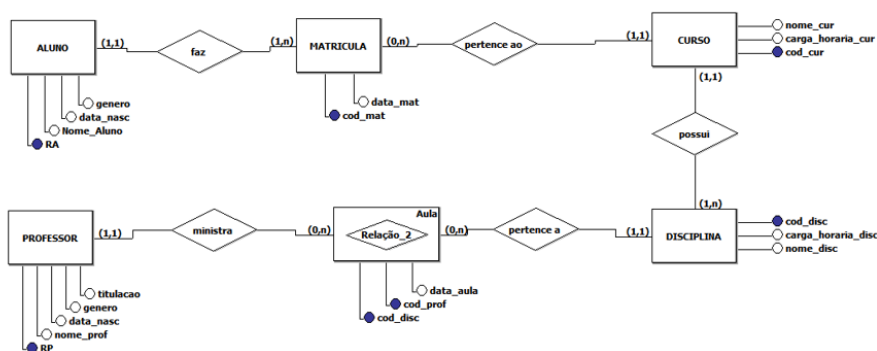
Tabela 2 - Leitura dos Relacionamentos

ALUNO<- >MATRICULA	IDA	VOLTA
	1 ALUNO faz no mínimo 1 MATRICULA e no máximo N MATRICULAS	1 MATRICULA tem no mínimo 1 ALUNO e no máximo 1 ALUNO
MATRICULA<- >CURSO	IDA	VOLTA
	1 MATRICULA pertence no mínimo a 1 CURSO e no máximo 1 CURSO	1 CURSO tem no mínimo 0 MATRICULAS e no máximo N MATRICULAS
CURSO >DISCIPLINA	IDA	VOLTA
	1 CURSO possui no mínimo 1 DISCIPLINA e no máximo N DISCIPLINAS	1 DISCIPLINA no mínimo pertence a 1 CURSO e no máximo 1 CURSO
DISCIPLINA<-> PROFESSOR	IDA	VOLTA
	1 DISCIPLINA é ministrada por no mínimo 0 PROFESSORES e no máximo N PROFESSORES	1 PROFESSOR no mínimo ministra 1 DISCIPLINA e no máximo N DISCIPLINAS

Fonte: autoral, 2023.

Note, que o relacionamento entre as entidades DISCIPLINA e PROFESSOR é do tipo N:M, isso significa, que será necessário fazer um ajuste no modelo conceitual com a criação da entidade associativa, conforme se vê abaixo.

Figura 14 - Modelo Conceitual



Fonte: autoral, 2023.

Modelagem Lógica

Chegamos finalmente na modelagem lógica. Considerando as fases do projeto de banco de dados, a modelagem lógica sucede a modelagem conceitual e antecede a última fase, a física. Nessa fase o artefato gerado é um modelo lógico e pode-se dizer que ele é derivado do modelo conceitual.

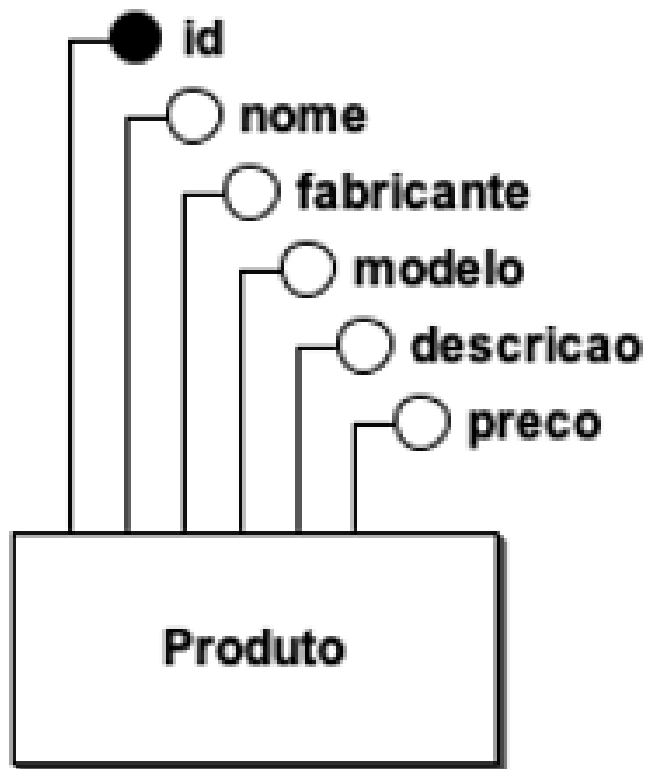
Antes de começar essa modelagem, é muito importante entender algumas coisas que serão utilizadas. Primeiro é preciso relembrar o conceito de entidade e atributo. As entidades representam os objetos do mundo real, enquanto seus atributos representam as características destes objetos. Por exemplo, num sistema de e-commerce, você pode ter a entidade chamada “Produto”, enquanto seus atributos possíveis podem ser “nome”, “fabricante”, “modelo”, “descrição”, “preço”, entre outros.

Não podemos esquecer também que existem essencialmente dois tipos de atributos: comum (ou tipo “dado”) e identificador (ou tipo “chave”).

O atributo do tipo comum é aquele que não possui nenhuma restrição importante na modelagem conceitual e representa um dado normalmente, daquela entidade. Já o atributo do tipo chave é aquele que identifica o conjunto de dados. Em uma tabela um atributo tipo chave não pode ser nulo e nem mesmo se repetir nunca. Por exemplo, no mesmo cenário de um e-commerce, como cada produto possui características que podem se repetir, podemos criar um novo atributo chamado “id_produto” para considerá-lo como chave. Neste caso, este “id” é único de um produto (regra de nunca se repetir) e nunca podem existir produtos sem este dado (regra de nunca ser nulo).

O exemplo da tabela de produtos pode ser visto na figura 1, onde a modelagem conceitual deste simples objeto é descrita.

Figura 15 - Modelo conceitual da entidade "Produto" com seus respectivos atributos.



Fonte: autoral, 2023.

Para ilustrar como ficaria a tabela deste exemplo, alguns dados arbitrários e fictícios podem ser vistos na tabela 1. Note que a tabela se chama “Produto” e cada um dos atributos descritos no modelo conceitual se tornaram as colunas. Note, também, que nessa tabela de exemplo os nomes das colunas estão todos em letras minúsculas e sem quaisquer acentuação ou caracteres especiais. Embora seja apenas um exemplo, já está sendo exibido no padrão lógico. Note, também, que a coluna “id” possui uma cor diferente para representá-la como chave.

Tabela 3 - Exemplo de tabela de produtos

Produto					
id	nome	fabricante	modelo	descrição	preço
1	Discman Legal	DiscLegal	DM2020	Discman moderno e portátil	R\$ 100,00
2	Walkman	WalkMusic	WM2020	Walkman retro com rebobinagem automática	R\$ 80,00
3	MP4 player	MusicPlayer	MP42020	MP4 player com tela de 5 polegadas	R\$ 200,00
4	MP4 Player	MusicPlayer	MP32020	MP3 player portátil estilo pendrive	R\$ 400,00
5	Fone sem fio	MusicPlayer	FONBLU21	Fone de ouvido bluetooth com graves ótimos	R\$ 1000,00

Fonte: autoral, 2023.

Lembre-se que a tabela anterior contém apenas exemplos arbitrários e está sendo usada apenas para referência. Note uma coisa importante nessa tabela: Cada coluna possui um tipo de dados específico. Isso é algo que devemos prestar bastante atenção na hora da modelagem lógica, pois os tipos de dados dependem da tecnologia de banco de dados que será utilizada. É por isso que durante a modelagem lógica já é preciso saber qual tecnologia de banco de dados será utilizada. Para todos os exemplos a partir de agora, será considerado o *Oracle Database* como foco e, por isso, toda a modelagem será baseada nele, inclusive realizada com ferramentas da própria Oracle, a partir de agora.

Tipos de Dados

No exemplo mostrado anteriormente as colunas possuem tipos específicos de dados. Se estivéssemos olhando do ponto de vista de uma linguagem de programação, saberíamos que estes atributos são, do tipo: “id” é um inteiro, “nome”, “fabricante”, “modelo” e “descricao” são do tipo texto (String para muitas linguagens) e “preco” do tipo número real, ou float (variáveis de ponto flutuante) para muitas linguagens. Entretanto, no mundo dos bancos de dados, existem tipos específicos para se representar cada tipo em cada tecnologia utilizada. De forma resumida, a tabela 2 mostra os principais tipos de dados em bancos de dados Oracle, mas muito comuns, também em outros bancos de dados como, MySQL, MariaDB, SQL Server etc.).

Tabela 4 - Tipos de dados mais usados em bancos Oracle

Tipo	Descrição
CHAR(n)	Representa uma cadeia de caracteres. Se o “n” (tamanho) não for informado, representará apenas um único caractere. O tamanho máximo é 2000. Quando um dado não atinge o tamanho máximo uma cadeia de espaços em branco é gerada.
VARCHAR2 (n)	Representa um texto, onde n pode ser de até 4000 caracteres. Quando um dado é inserido sem ocupar o tamanho máximo, o banco de dados não acrescenta espaços em branco.
NUMBER(n)	Representa um número de até “n” dígitos. Se o “n” não for informado, possuirá uma precisão de 38 dígitos significativos.
NUMBER(n, d)	Representa um número decimal de até “n” dígitos e precisão “d”, onde “d” é o número máximo de casas decimais após a vírgula.
DATE	Representa um dado do tipo data, com horas, minutos e segundos, de 1 Jan de 4712 AC até 31 de Dez de 9999 DC
BLOB	“Binary Large Object”, pode armazenar até 4GB de texto binário. Normalmente utilizado para arquivos (imagens, vídeos etc.)
CLOB	“Character Large Object”, pode armazenar até 4GB de texto.
BOOLEAN	O Oracle não implementa nativamente o tipo boolean. É preciso utilizar algum mecanismo como um char(1) ou um varchar2(1) ou number(1).

Fonte: autoral, 2023.

É muito importante salientar que há muitos outros tipos de dados, que serão, neste material, explorados aos poucos e conforme a necessidade. Os tipos de dados mostrados na tabela 2 representam os mais comuns, que podem resolver a maior parte dos casos.

No caso do exemplo da tabela produto, podemos notar que o “id” pode ser do tipo “NUMBER(5)”, ou seja, um número de até 5 dígitos enquanto os atributos “nome” pode ser um VARCHAR2(40), ou sejam um texto de até quarenta caracteres, o “fabricante” pode ser tipo VARCHAR2(15), o “modelo” pode ser um VARCHAR2(30), a “descricao” pode ser um VARCHAR2(255) e, finalmente, o atributo “preço” pode ser um NUMBER(5,2), ou seja um número real de até 5 dígitos com duas casas decimais de precisão.

Chave Estrangeira (FK)

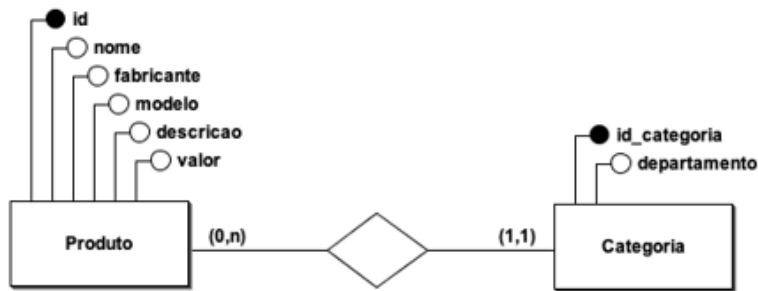
Na modelagem lógica um novo tipo de coluna aparece. As tão esperadas chaves estrangeiras ou, em sua nomenclatura original, *foreign keys*, as famosas FKs. É muito comum tratarmos esse tipo de coluna com o nome original em inglês ou, simplesmente, pela sigla dela.

As chaves estrangeiras são importantes pois são elas que representarão os relacionamentos entre as entidades (que agora começamos a chamar de tabelas). Sempre que estamos derivando o modelo conceitual para o lógico, teremos uma série de chaves estrangeiras em nosso modelo.

Conceito importante sobre chave estrangeira: Todas as colunas marcadas como “FK”, sempre apontam para uma coluna de chave primária (primary key – PK), ou seja, os dados que são inseridos dentro das colunas do tipo “FK” são dados que apontam para outros dados, sempre identificadores (chave) de alguma outra entidade (relacionada) com a tabela que está recebendo essa FK. Eventualmente apontam para a mesma tabela, em auto relacionamentos.

Para melhor entendimento, vamos ao exemplo original do “produto”, mas agora será acrescido uma nova tabela chamada “categoria”, conforme descrito pelo modelo conceitual da figura 2.

Figura 16 - Modelo conceitual de exemplo, com uma nova entidade



Fonte: autoral, 2023.

Neste exemplo, foi adicionada uma entidade chamada categoria, onde o produto deve possuir no mínimo uma categoria e no máximo uma categoria, também. Por outro lado, a categoria pode não conter nenhum produto, mas também pode conter vários produtos. Para o relacionamento em questão, uma FK deve ser gerada na tabela de produto, ou seja, uma nova coluna deve ser incluída na tabela de produto indicando a qual categoria ele pertence, mas através de seu "id_categoria". Para ilustrar essa ideia, veja a tabela categoria implementada com dados arbitrários.

Tabela 5 - Exemplo de dados na tabela "categoria", do exemplo

Categoria	
id_categoria	departamento
100	Tocadores de CD

Isso é uma FK!

101	Tocadores de Fita K7
102	Tocadores Digitais
103	Fones
104	Baterias

Fonte: autoral, 2023.

Tabela 6 - Tabela de produtos com relacionamento para categoria:

Produto						
Id	nome	fabricante	modelo	descrição	preço	Id_categoria
1	Discman Legal	DiscLegal	DM2020	Discman moderno e portátil	R\$ 100,00	100
2	Walkman	WalkMusic	WM2020	Walkman retro com rebobinagem automática	R\$ 80,00	101
3	MP4 player	MusicPlayer	MP42020	MP4 player com tela de 5 polegadas	R\$ 200,00	102
4	MP4 Player	MusicPlayer	MP32020	MP3 player portátil estilo pendrive	R\$ 400,00	102
5	Fone sem fio	MusicPlayer	FONBLU21	Fone de ouvido bluetooth com graves ótimos	R\$ 1000,00	103

Fonte: autoral, 2023.

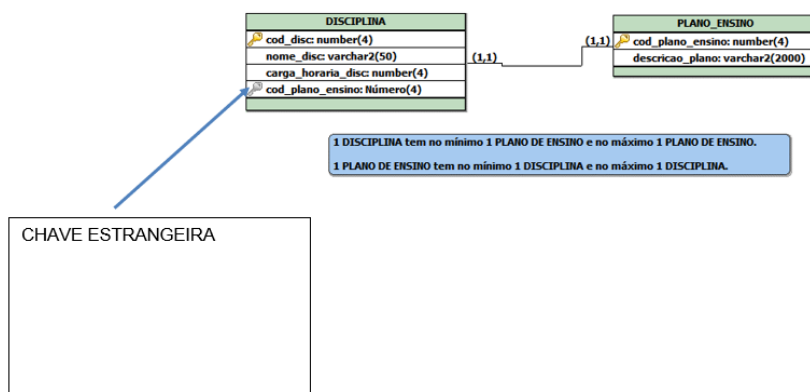
Note que cada categoria, neste caso, possui um “id” que é único da categoria e, para satisfazer esse tipo de relacionamento, o “id_categoria” deve ser adicionado na tabela de produtos, conforme exemplificado com dados arbitrários da tabela 4. Isso quer dizer que a “id_categoria” na tabela de produtos é um FK para tabela “Categoria”. Neste modelo foi adicionada a FK na tabela “produto” pois cada produto possui uma categoria, ou seja, o entendimento do contexto também é necessário para a inclusão dessa FK no local correto e indicada pela modelagem lógica, vista a seguir.

Tipos de Relacionamentos que podemos encontrar no MER Lógico

Exemplo de Relacionamento 1:1

No relacionamento 1:1, geralmente a chave estrangeira (FK) fica na entidade que contém (entidade forte), ou seja, do lado da entidade que existe sem depender da outra entidade relacionada. No exemplo abaixo, a disciplina existe independente do plano de ensino. Já, no caso, do plano de ensino, não faz sentido ter um plano de ensino sem a disciplina, por isso nesse exemplo a FK fica do lado da entidade forte, ou seja, na disciplina.

Figura 17 - Relacionamento 1:1

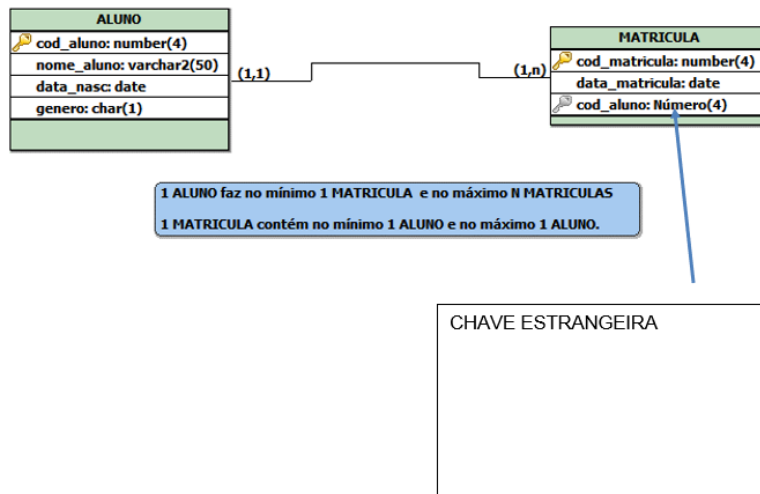


É importante ressaltar que em alguns casos de relacionamento 1:1, não se faz a separação em entidades diferentes, mas isso depende muito das regras específicas de funcionamento de cada negócio.

Exemplo de Relacionamento 1:N ou N:1

No relacionamento 1:N, a chave estrangeira sempre fica do lado do N.

Figura 18 - Relacionamento 1:N - Fonte: Autoral, 2023.



Exemplo de Relacionamento N:M

No relacionamento N:M, é necessário criar uma entidade associativa, que via de regra, é formada pela composição das chaves primárias das entidades envolvidas no relacionamento, que vem como chaves estrangeiras formando uma chave primária composta, além dos atributos próprios dessa entidade.

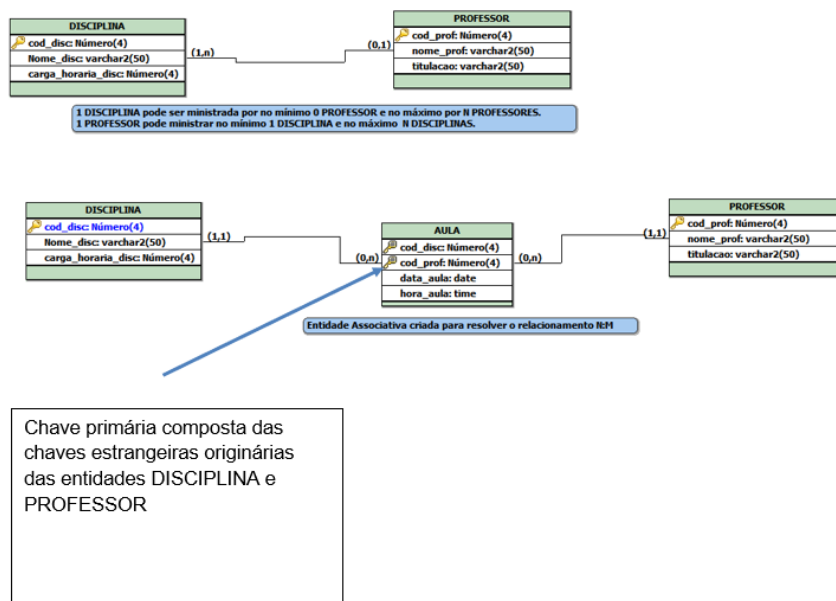


Figura 19 - Relacionamento N:M - Fonte: autoral, 2023.

Modelagem Física

O modelo físico contém todos os scripts de comandos, que permitem a implementação física do banco de dados. Esse nível de modelagem é completamente dependente do SGBD.

Vamos ver um exemplo prático da derivação de Modelo lógico e físico. Analise o modelo entidade relacionamento lógico da fábrica de autopeças, gere e implemente o modelo físico, além de fazer a população das tabelas criadas.

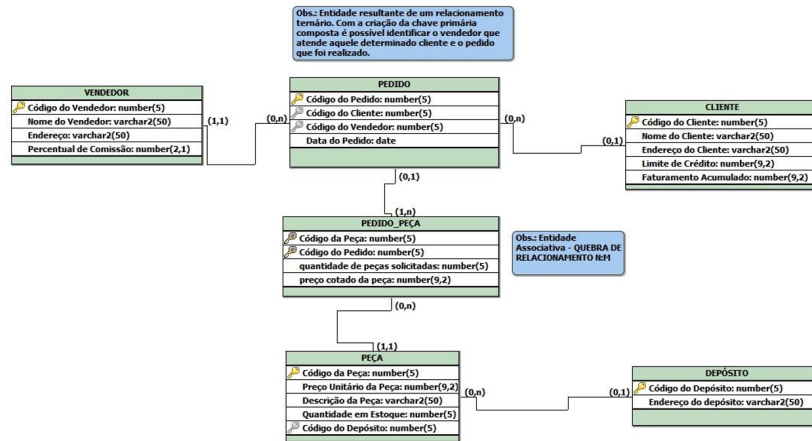


Figura 20 - Modelo Lógico - Fonte: Autoral, 2023.

Ao fazer o modelo físico, é necessário conhecer as regras de nomenclatura do SGBD e, se for o caso, ajustar os nomes dos atributos do modelo lógico para o modelo físico. As regras de nomenclatura, mudam de um SGBD para outro, mas as principais regras são:

- nome de objeto não pode começar com número;
- não pode haver objetos com nomes duplicados;
- não pode usar palavras reservadas para nomear objetos;
- somente os caracteres especiais _, \$ e # podem ser utilizados nos nomes;
- nomes dos objetos podem ter letras e números;
- quanto ao tamanho, depende muito do SGBD, mas por exemplo no Oracle, a partir da versão 10g, é possível criar objetos com 128 caracteres.

Ao fazer o modelo físico, primeiro crie as entidades fortes, depois as entidades fracas e por fim as entidades associativas. Então, no exemplo, da fábrica de autopeças, a ordem de criação deve ser : Cliente, Vendedor, Depósito, Pedido, Peça e Pedido_Peca.

Uma técnica utilizada, para não se preocupar com a ordem de criação das tabelas, é criar as tabelas sem nenhuma constraint e depois de criar todas as tabelas executar o comando alter table e adicionar as constraints de pk's e fk's.

Abaixo, temos um exemplo de modelo lógico, dentro do BRmodelo, do estudo de caso da fábrica de autopeças.

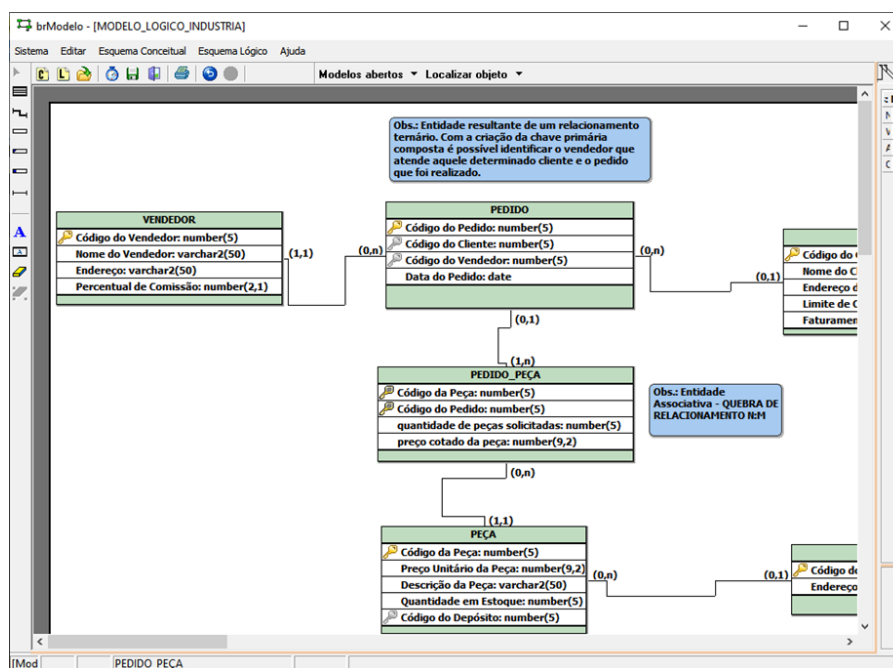


Figura 21 - Modelo Lógico AutoPeças - Fonte: autoral, 2023.

Depois clique no menu esquema lógico e selecione a opção gerar esquema físico, conforme figura abaixo.

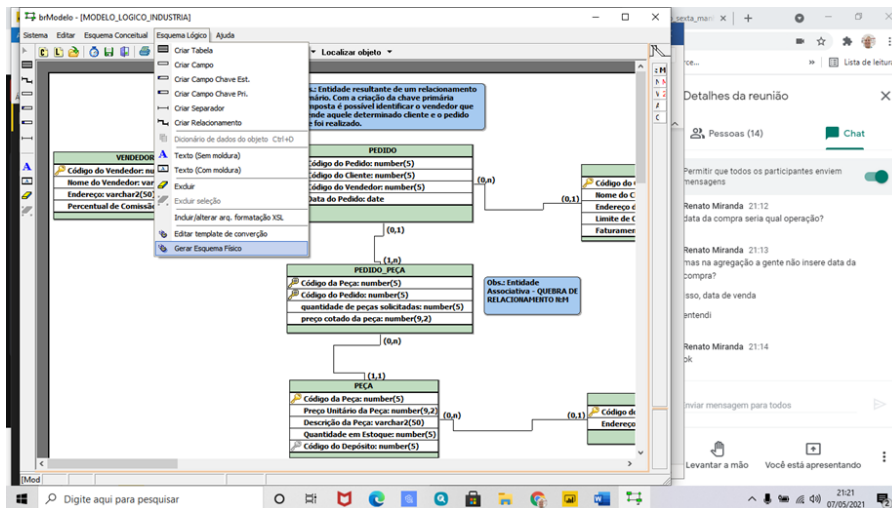


Figura 22 - Conversão do Modelo - Fonte: Autoral, 2023.

Clique na opção de conversão, conforme figura abaixo.

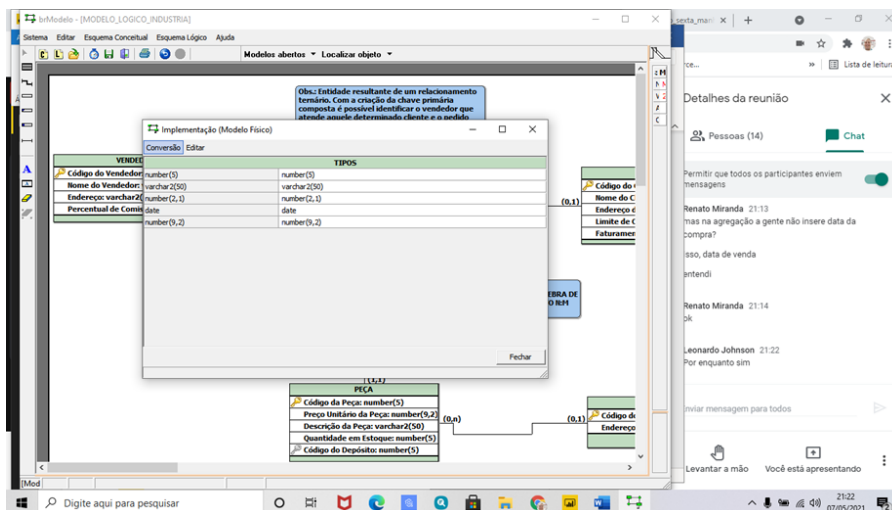


Figura 23 - Conversão dos tipos de dados - Fonte: autoral, 2023.

Verifique que o esquema físico foi gerado com os comandos de criação de todas as tabelas que constavam no modelo lógico.

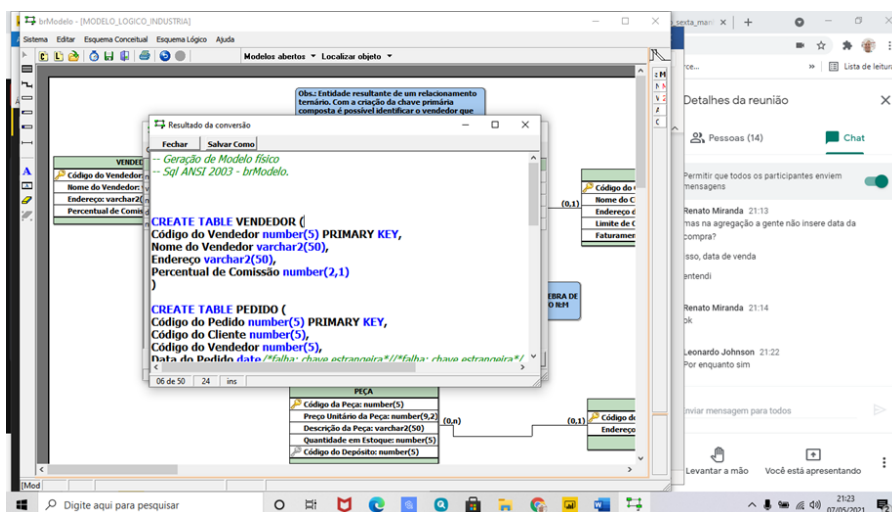


Figura 24 - Geração do Modelo Físico - Fonte: autoral, 2023.

Exemplo de Geração de Modelo físico

Observe que primeiro são criadas as entidades fortes, depois as entidades fracas e por fim as entidades associativas. É muito importante, respeitar essa ordem na criação das tabelas, para não ter erros de violação de constraints (restrições de integridade).

```
CREATE TABLE VENDEDOR (
```

```
Cod_do_Vendedor number(5) PRIMARY KEY,
```

```
Nome_do_Vendedor varchar2(50),
```

```
Endereco varchar2(50),
```

```
Percentual_de_Comissao number(2,1)
```

```
);
```

```
CREATE TABLE CLIENTE (
```

```
Cod_do_Cliente number(5) PRIMARY KEY,
```

```
Nome_do_Cliente varchar2(50),
```

```
Endereco_do_Cliente varchar2(50),
```

```
Limite_de_Credito number(9,2),
```

```
Faturamento_Acumulado number(9,2)
```

```
);
```

```
CREATE TABLE DEPOSITO (
```

```
Cod_do_Deposito number(5) PRIMARY KEY,
```

```
Endereco_do_deposito varchar2(50)
```

```
);
```

```
CREATE TABLE PECA (
```

```
Cod_da_Peca number(5) PRIMARY KEY,
```

```
Preco_Unitario_da_Peca number(9,2),
```

```
Descricao_da_Peca varchar2(50),
```

```
Quantidade_em_Estoque number(5),
```

```
Cod_do_Deposito number(5) REFERENCES DEPOSITO (Cod_do_Deposito)
```

```
);
```

```
CREATE TABLE PEDIDO(
```

```
Cod_do_Pedido number(5) PRIMARY KEY,
```

```
Cod_do_Cliente number(5) references Cliente (cod_do_cliente),
```

```
Cod_do_Vendedor number(5) references Vendedor (cod_do_vendedor),
```

```
Data_do_Pedido date
```

```
);
```

```
CREATE TABLE PEDIDO_PECA (
```

```
Cod_da_Peca number(5) references peca(cod_da_peca),
```

```
Cod_do_Pedido number(5) references pedido (cod_do_pedido),
```

```
Qtde_de_Pecas_solicitadas number(5),
```

```
Preco_cotado_da_Peca number(9,2),  
PRIMARY KEY(Cod_da_Peca,Cod_do_Pedido)  
);
```

Verifique que o esquema físico foi gerado com os comandos de criação de todas as tabelas que constavam no modelo lógico.

Conteúdo bônus

Tópicos avançados

Depois de realizar a criação do modelo físico, significa que o projeto banco de dados está implementado e terminado. E, a partir desse momento, o banco de dados está pronto para o processo de população das tabelas, ou seja, sofrer as operações de inserções de registros das tabelas e, por consequência as demais operações de manipulação de dados como consultas, alterações e exclusões de registros, ou seja, temos o banco de dados disponível para leitura e gravação. Então, vamos ver um exemplo da população das tabelas da fábrica de autopeças, explanada no texto.

```
Insert into vendedor values (10,'Jose','R.X,100',0.1);
```

```
Insert into vendedor values (11,'Joao','R.Y,200',0.2);
```

```
Insert into vendedor values (12,'Antonio','R.Z,300',0.1);
```

```
Insert into vendedor values (13,'Maria','R.W,400',0.1);
```

```
Insert into vendedor values (14,'Joaquim','R.K,500',0.2);
```

```
Commit;
```

```
Select * from vendedor;
```

```
Insert into cliente values (100,'Joana','R.X,1000',10000,1548);
```

```
Insert into cliente values (110,'Josafa','R.Y,2000', 20000,5874);
```

```
Insert into cliente values (120,'Andre','R.Z,3000', 30000,9856);
```

```
Insert into cliente values (130,'Mariana','R.W,4000', 40000,4512);
```

```
Insert into cliente values (140,'Manuel','R.K,5000', 50000,7541);
```

```
Commit;
```

```
Select * from cliente;
```

```
Insert into deposito values (1,'R.A,10');
```

```
Insert into deposito values (2,'R.B,11');
```

```
Insert into deposito values (3,'R.C,12');
```

```
Insert into deposito values (4,'R.D,13');
```

```
Insert into deposito values (5,'R.E,14');
```

```
Commit;
```

```
Select * from deposito;
```

```
Insert into peca values(1000,10,'martelo',15,1);
```

```
Insert into peca values(1001,11,'machado',20,3);
```

```
Insert into peca values(1002,12,'prego',1,1);
```

```
Insert into peca values(1003,13,'parafuso',1,2);
```

```
Insert into peca values(1004,14,'chave de fenda',20,1);
```

Commit;

Select * from peca;

Insert into pedido values (201,100,10,'01/02/2021');

Insert into pedido values (202,100,10,'01/03/2021');

Insert into pedido values (203,110,11,'10/02/2021');

Insert into pedido values (204,120,12,'11/05/2021');

Insert into pedido values (205,120,12,'21/04/2021');

Commit;

Select * from pedido;

Insert into pedido_peca values (1000,201,100,10);

Insert into pedido_peca values (1001,201,200,10);

Insert into pedido_peca values (1002,202,150,11);

Insert into pedido_peca values (1003,203,120,12);

Insert into pedido_peca values (1004,204,130,13);

Commit;

Select * from pedido_peca;

É importante observar que ao inserir os códigos, é necessário prestar atenção na colocação dos códigos corretos nas chaves estrangeiras, ou seja, insira os códigos que já foram criados como chave primária, para não ter erros de violação de chave estrangeira.

Referências Bibliográficas

DATE, C. J. **Introdução a sistemas de banco de dados**. Rio de Janeiro. Ed. Campus, 1991.

CHEN, Peter. **Modelagem de dados: a abordagem entidade-relacionamento para projeto lógico**. São Paulo: Makron Books, 1990.

MEDEIROS, L. F. **Banco de dados, princípios e práticas**, 1ª. ed., Ed. Intersaberes, 2013.

PUGA, S., França E., GOYA M., **Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g**, Ed. Pearson, 2013.

ELMASRI R., NAVATHE, S. **Sistemas de Banco de Dados**, 4ª ed. Ed. Pearson, 2005.