



Persistência e Pesquisa de Dados

Capítulo 1. Modelagem Relacional de Dados

Prof. Gustavo Aguilar



Aula 1.1. Introdução à Modelagem de Dados

- ☐ Minimundo, Modelo de Dados.
- ☐ Pirâmide do Conhecimento
- ☐ Visão Geral do Processo de Modelagem de Dados
- ☐ Níveis de Modelagem e Abstração

- Representação gráfica



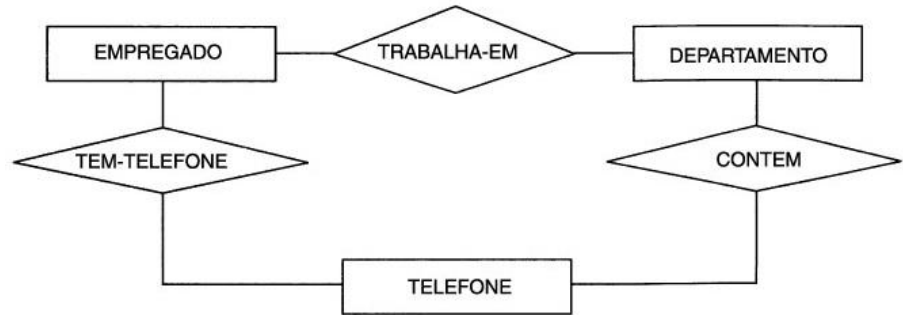
- Modelo:

- Desenho
- Réplica
- Protótipo
- Coisa que serve de exemplo
- Esquema de representação



- “Modelo é a representação abstrata e simplificada de um sistema real, com a qual se pode explicar ou testar o seu comportamento, em seu todo ou em partes” Paulo Cougo (1997, p. 7).

- Representação abstrata dos dados de um sistema, com o qual podemos explicar o seu comportamento.
- Korth (1999):
 - Conjunto de ferramentas conceituais
 - Descrição de dados
 - Relacionamentos entre dados
 - Semântica de dados
 - Regras de consistência



- **Dado:** “matéria-prima”
 - Estado bruto
 - Sem significado de forma isolada
- **Informação:** dados com significado
 - “o quê?”, “quem?”, “onde?” ou “quando?”.
- **Conhecimento:** informação enriquecida
 - “como?” e “por quê?”
- **Sabedoria:** conhecimento aplicado
 - Robotizações e Inteligência Artificial



Pirâmide do Conhecimento, também conhecida como “Pirâmide DIK (Data, Information, Knowledge)”

▪ Dado?

- Matriz de caracteres

▪ Informação?

- Imagem de Zebra

▪ Conhecimento?

- Zebra com machucado

▪ Sabedoria?

- Acionar mecanismos do zoológico para direcionar a zebra para o setor veterinário



- Modelo que explique os relacionamentos entre os dados, para que seja possível prover insumos consistentes para a geração de conhecimento e sabedoria.
- Modelo de dados que preveja todos os dados brutos necessários.



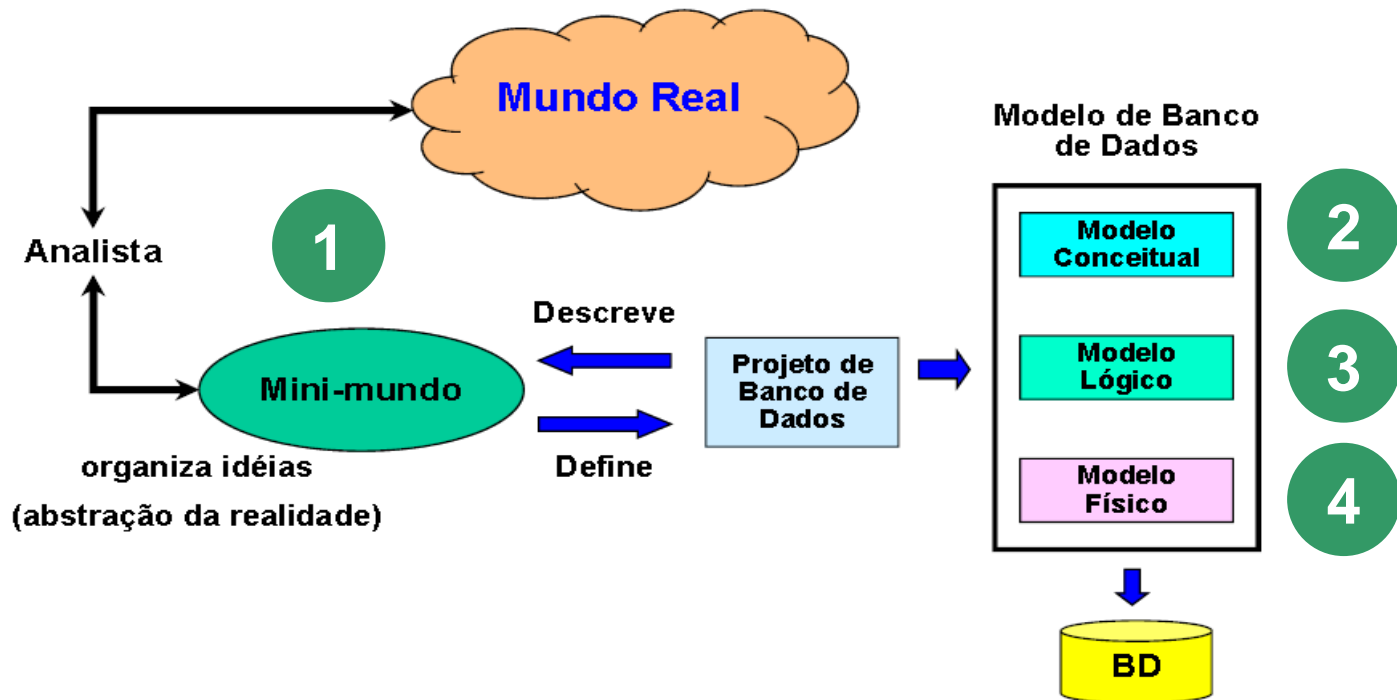
Lembrete muito importante !!



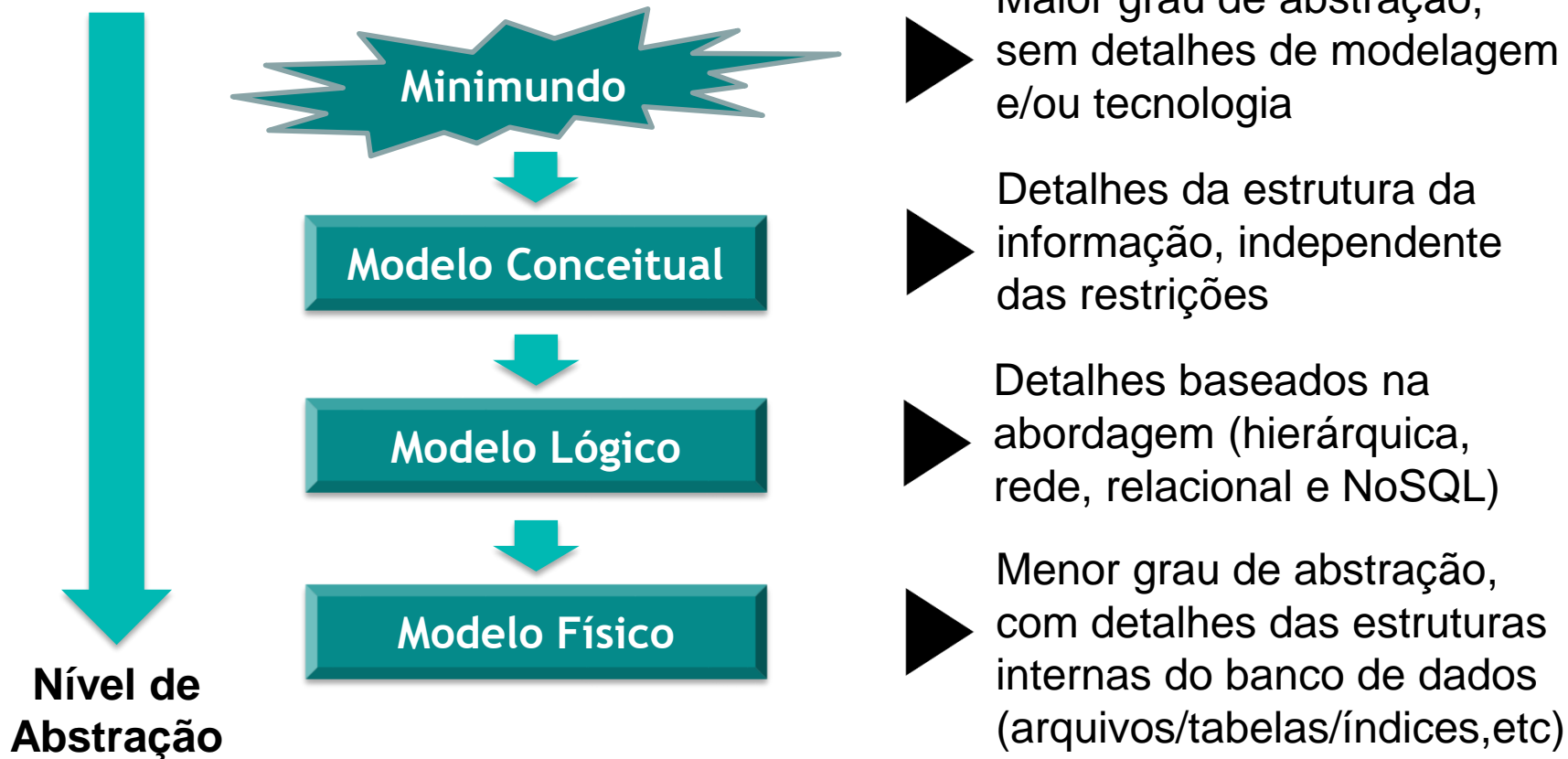
Pirâmide do Conhecimento, também conhecida como "Pirâmide DIK (Data, Information, Knowledge)"

Visão Geral do Processo de Modelagem de Dados iGTi

- Quatro grandes etapas



Níveis de Modelagem e Abstração



- ☑ Minimundo: sua observação como o “pontapé” inicial.
- ☑ Modelo de Dados: representação abstrata dos dados do minimundo.
- ☑ Pirâmide do Conhecimento.
- ☑ Dado → Informação → Conhecimento → Sabedoria
- ☑ Visão geral do processo de Modelagem de Dados.
- ☑ Minimundo → Mod. Conceitual → Mod. Lógico → Mod. Físico
- ☑ Níveis de Modelagem: abstração x níveis de detalhes.

Próxima aula

☐ Modelagem Conceitual.



Aula 1.2. Modelagem Conceitual

- ❑ Introdução à Modelagem Conceitual.
- ❑ O Modelo Conceitual de Dados (MCD).

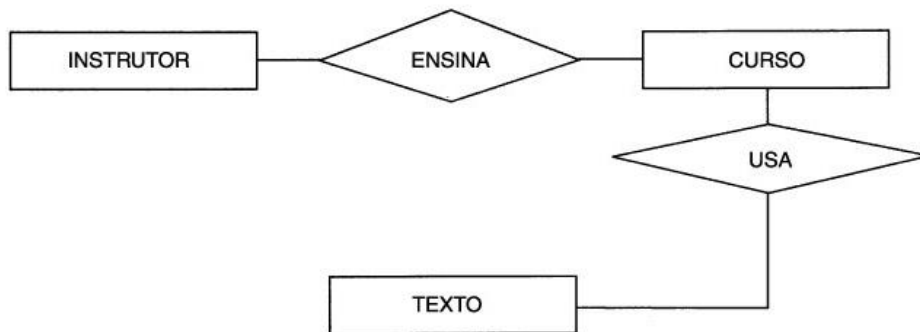
- Objetivo de construir o Modelo Conceitual de Dados (MCD);
- “...representação fiel ao ambiente observado, independente de limitações quaisquer impostas por tecnologias, técnicas de implementação ou dispositivos físicos.” (Paulo Cougo, 1997, p. 28)

➔ ater-se única e exclusivamente aos aspectos conceituais ←

- Utilizado para o nível de conversação, entendimento, transmissão, validação de conceitos, mapeamento do ambiente, etc.
- **Estabilidade ➔** derivação de diferentes modelos lógicos (relacional, hierárquico, de rede, NoSQL) a partir do mesmo MCD.

O Modelo Conceitual de Dados (MCD)

- Visão macro dos dados e os relacionamentos entre si;
- Detalhes das estruturas de informação necessárias do minimundo, que futuramente serão armazenadas em um banco de dados;



- **Reaproveitamento** do MCD para outras finalidades que não a de projetar bancos de dados.

- ☑ Modelagem Conceitual: representação gráfica do minimundo.
- ☑ O Modelo Conceitual de Dados (MCD): estável e reaproveitável.

- ❑ O Diagrama de Entidades e Relacionamentos (DER).

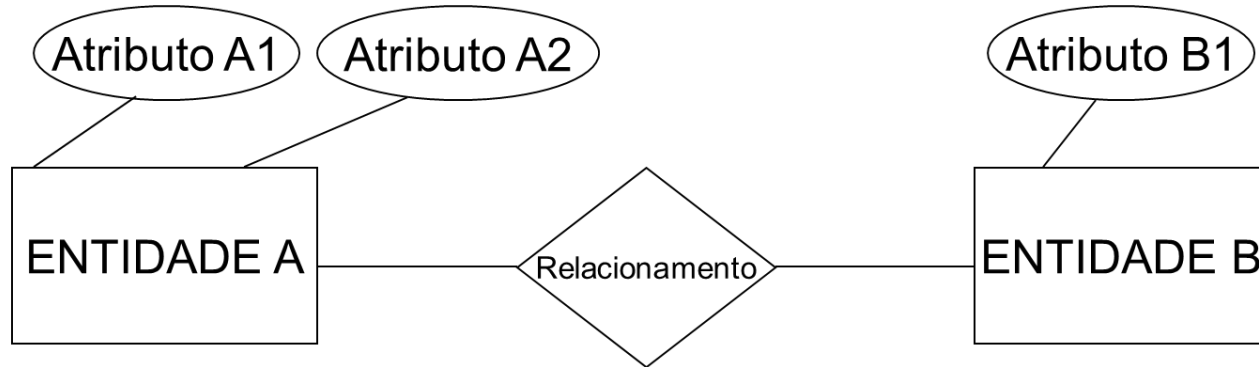


Aula 1.2.1. O Diagrama de Entidades e Relacionamentos (DER)

- ☐ O Diagrama de Entidades e Relacionamentos (DER).
- ☐ Entidade.
- ☐ Atributo.
- ☐ Relacionamento.
- ☐ Especialização.
- ☐ Generalização.
- ☐ Notações gráficas.

O Diagrama de Entidades e Relacionamentos

- É um modelo conceitual → percepção do mundo real (minimundo);
- Representação gráfica → DER;
- Construído com objetos → entidades (+ atributos) e relacionamentos.



- Proposto por Peter Chen em março de 1976
 - *The Entity-Relationship Model - Toward a Unified View of Data*

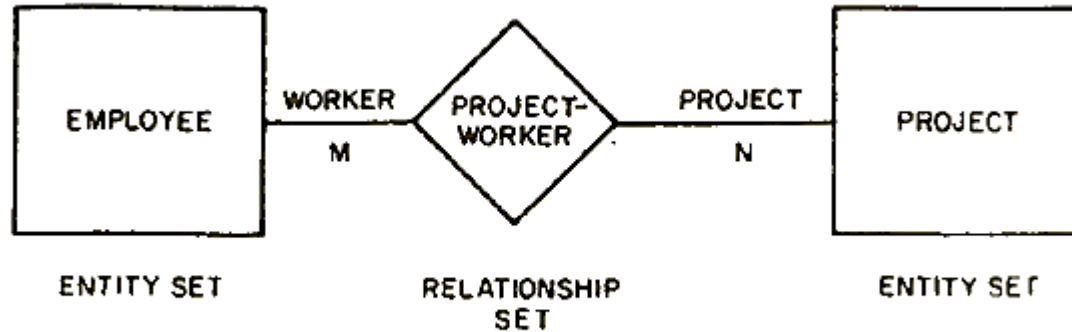


Fig. 10. A simple entity-relationship diagram

- Elaboração simples + Fácil Entendimento → uso em larga escala.

- Representação abstrata de um objeto do mundo real
 - ➔ Tudo aquilo que se deseja armazenar informação a respeito
- Ex.: CLIENTE, FUNCIONARIO, FUNCAO, VENDA, ALUNO, TURMA
- Representação no DER: com **retângulo**.

CLIENTE

PRODUTO

FUNCIONARIO

HISTORICO_ESCOLAR

- Classificadas em dois tipos;
- **Entidade Forte:** existe por si só, não dependendo de outra entidade para existir.
- **Entidade Fraca:** sua existência depende da existência de outra entidade.
 - Representada por um retângulo dentro do outro.



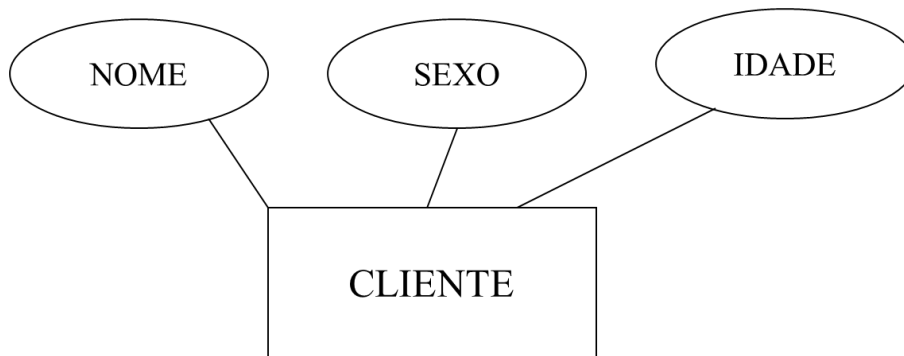
BANCO



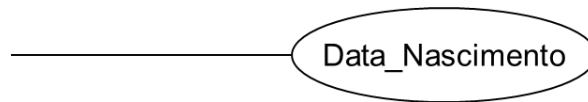
AGENCIA

- Estratégia para identificar entidades (Shlaer & Mellor): observação de **cinco grandes grupos**;
- **Coisas tangíveis**: pode-se tocar. Pessoas, animais, equipamentos, etc.
- **Funções exercidas**: classificações, papéis, atribuições, etc.
- **Eventos ou ocorrências**: ações ou fatos com informações temporais.
- **Interações**: associação de outros elementos. Ex.: venda, inscrição, carrinho de compras, etc.
- **Especificações**: características de outros elementos, não possuindo significado se olhados isoladamente. Ex.: modelo, dimensões, etc.

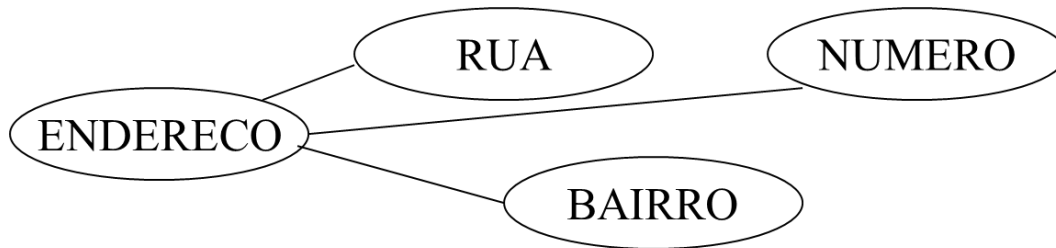
- Propriedade que **identifica** ou **qualifica** uma entidade;
- Ex.: nome, idade, matrícula, código produto, valor, quantidade, etc;
- Representado por uma **elipse** ligada à entidade.



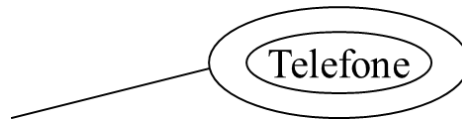
- **Atributo Simples:** não é dividido em partes → atômico.



- **Atributo Composto:** pode ser dividido em partes, ou seja, decomposto em outros atributos.



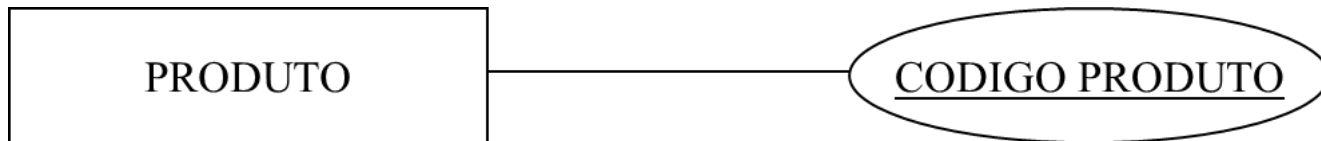
- **Atributo Monovalorado:** assume um único valor para cada elemento da entidade;
 - Ex.: CPF, IDADE, DATA_FABRICACAO, etc.
- **Atributo Multivalorado:** pode assumir um valor dentre um conjunto (faixa) de valores;
 - Ex.: TELEFONE, NACIONALIDADE, etc.;
 - É representado, no DER, por uma **elipse dentro de outra**.



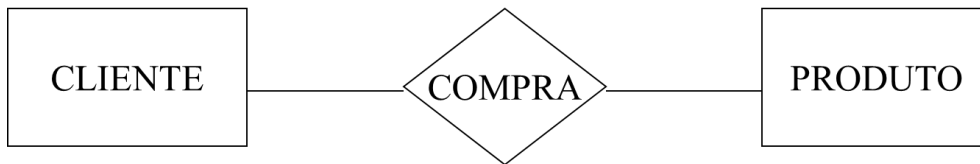
- **Atributo Nulo:** não possui valor ou o valor do mesmo é desconhecido → valor não é obrigatório;
- **Atributo Derivado:** valor obtido através do(s) valor(es) de outro(s) atributo(s);
 - Ex.: IDADE → pode ser obtida através da diferença entre a DATA ATUAL e o atributo DATA NASCIMENTO.
 - É representado, no DER, por uma **elipse pontilhada**.



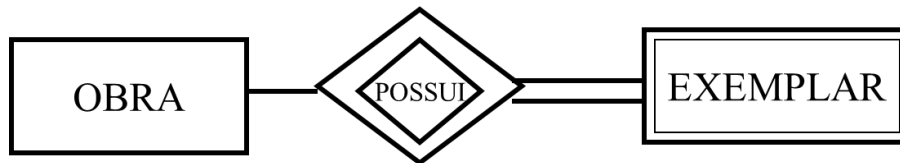
- **Atributo Chave:** identifica unicamente o conjunto de atributos de uma entidade → atributo mais importante da entidade;
- Atributos chaves **não podem ser nulos**;
- Ex.: CODIGO DO PRODUTO, MATRICULA DO FUNCIONARIO, etc.;
- É representado, no DER, com um **sublinhado no nome do atributo**.



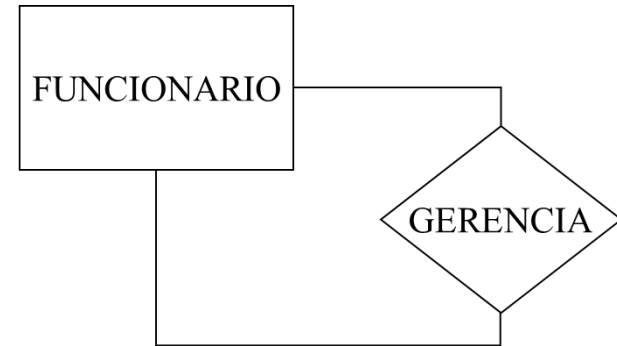
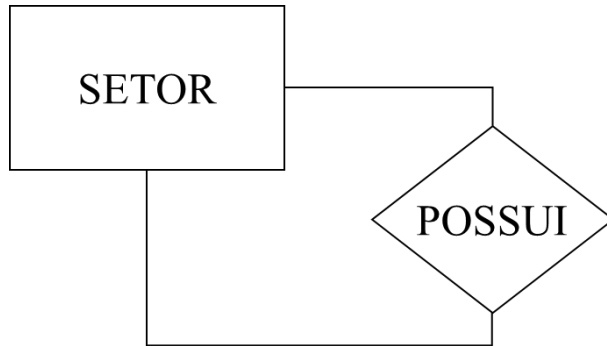
- Associação entre duas ou mais entidades;
- Indica uma correspondência ou dependência entre os elementos;
- **Relacionamento Natural:** entre duas entidades fortes;



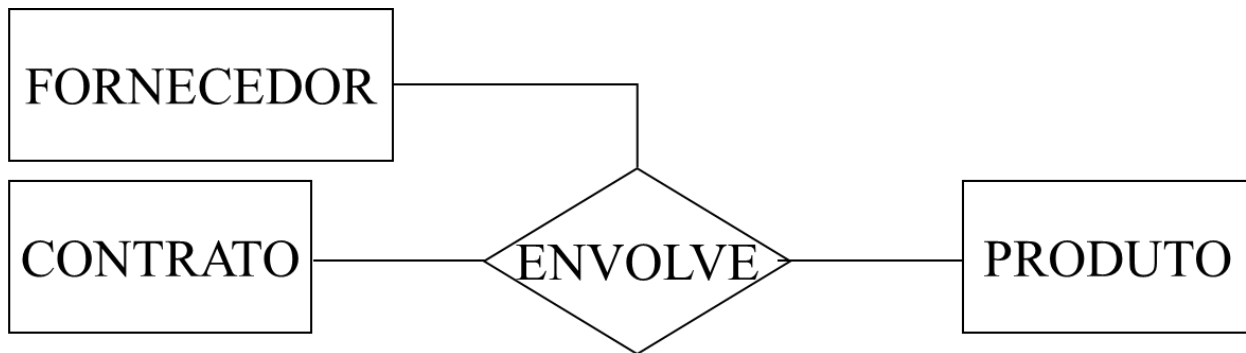
- **Relacionamento Identificador:** associa uma entidade fraca à uma entidade forte;



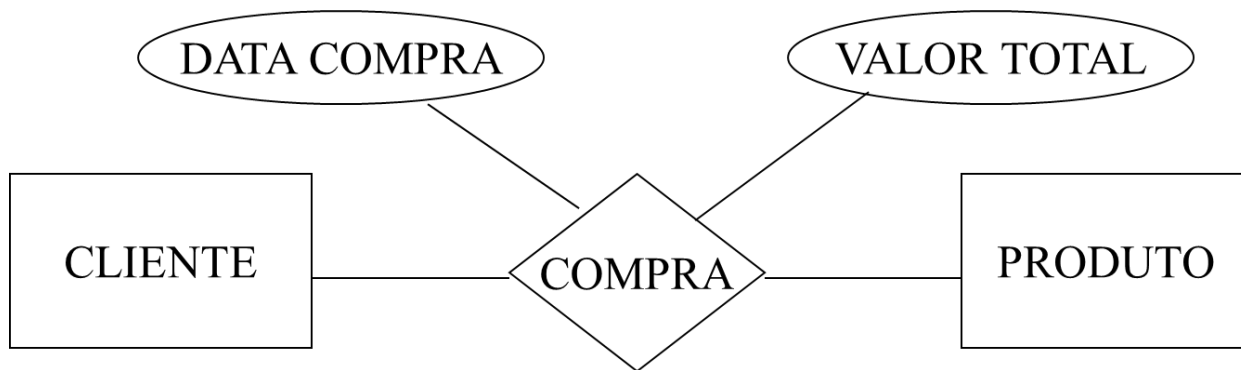
- **Relacionamento Recursivo:** relacionamento de uma entidade com ela mesma.
 - Autorelacionamento.



- **Grau do Relacionamento:** indica o número de entidades que participam de um relacionamento.
 - Binário, ternário, quaternário, etc.

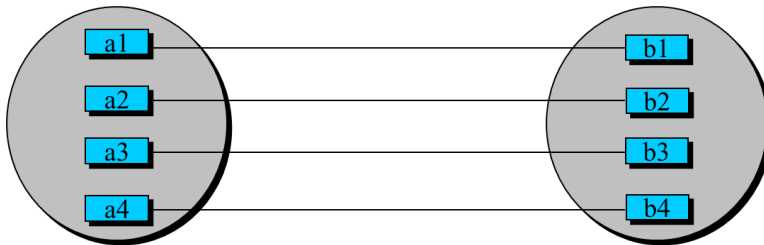


- **Atributos de Relacionamento:** identificam ou qualificam um relacionamento entre duas ou mais entidades.
 - Representados por elipses ligadas ao losango do relacionamento.

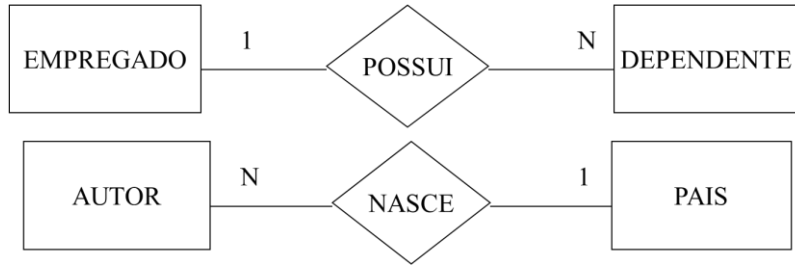
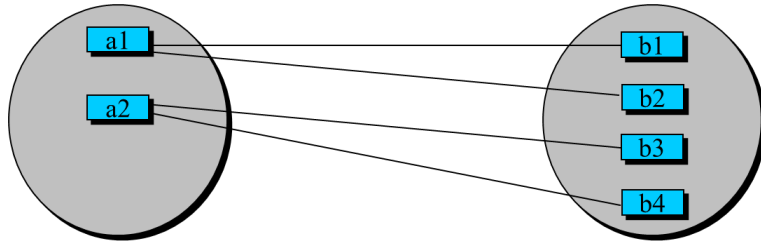


- **Cardinalidade de Relacionamento:** representa a quantidade de elementos de uma entidade A que podem ser associados à elementos de uma entidade B e vice-versa.
 - Também conhecida como Restrições de Mapeamento;
 - Baseada na Teoria de Conjuntos;
 - Pode ser:
 - Relacionamento Um Para Um (1:1)
 - Relacionamento Um Para Muitos (1:N)
 - Relacionamento Muitos Para Muitos (N:N)

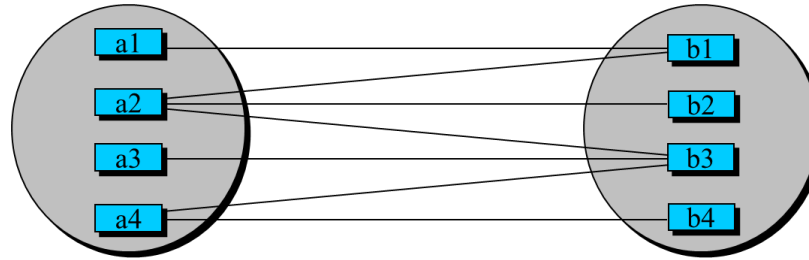
- **Relacionamento Um Para Um (1:1):** um elemento da entidade A pode estar associado no máximo a um elemento da entidade B e vice-versa.



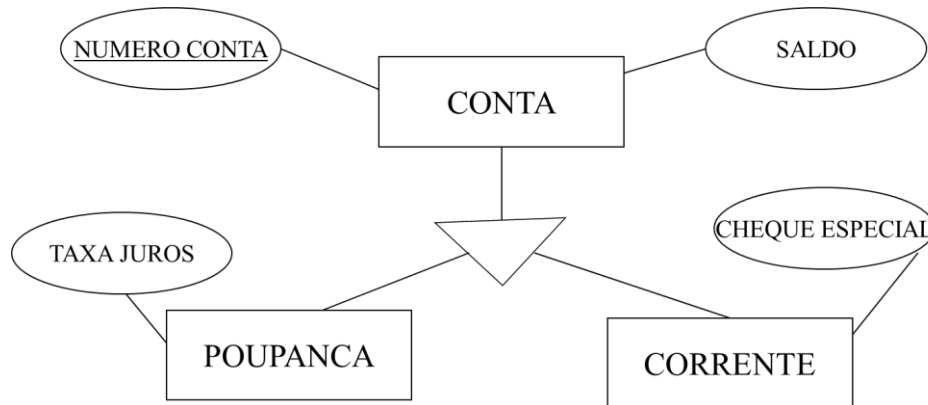
- **Relacionamento Um Para Muitos (1:N):** um elemento da entidade A pode estar associado a N elementos da entidade B, mas um elemento da entidade B pode estar associado no máximo a um elemento da entidade A.



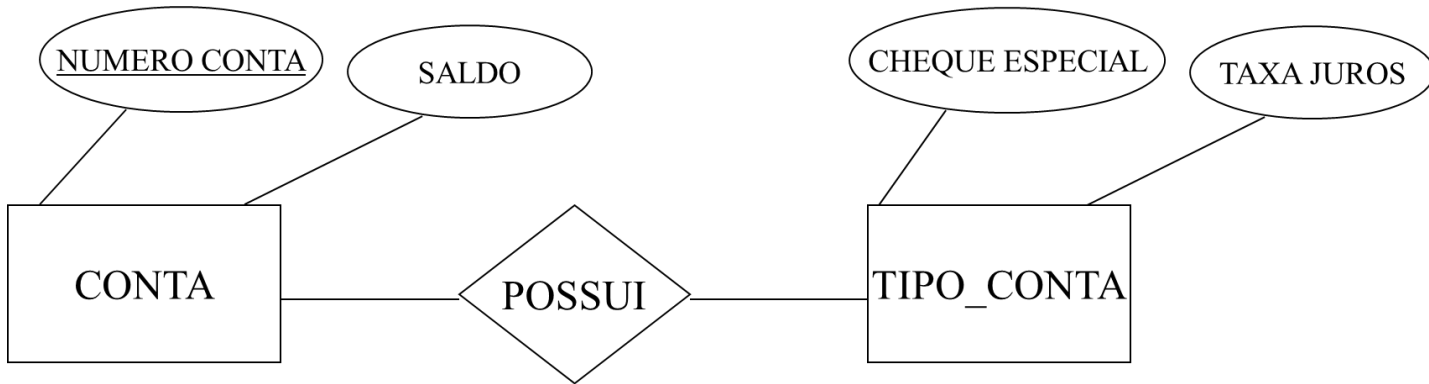
- **Relacionamento Muitos Para Muitos (N:N):** um elemento da entidade A pode estar associado a N elementos da entidade B e vice-versa.



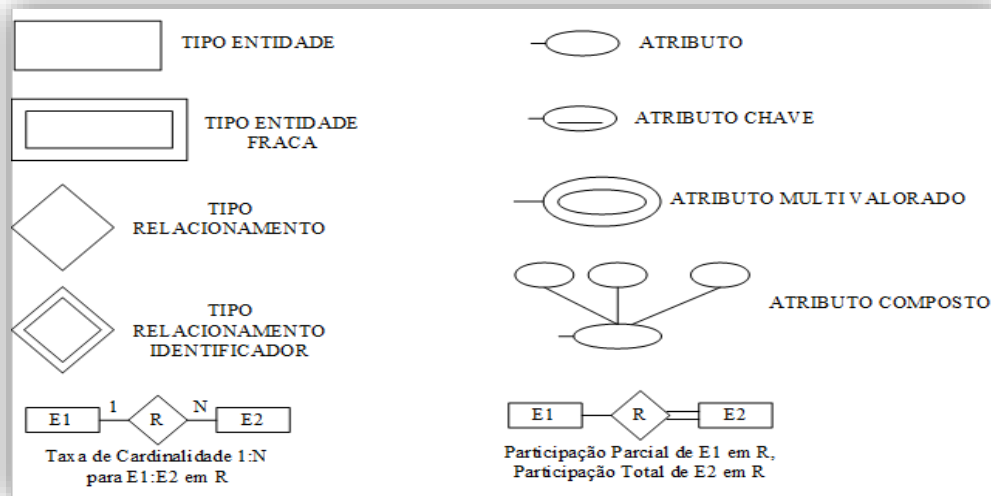
- Recurso que permite a definição de subgrupos para uma determinada entidade;
- Abordagem top-down → dividir uma entidade em outras, de forma a se ter dados (atributos) mais especializados em cada uma delas.



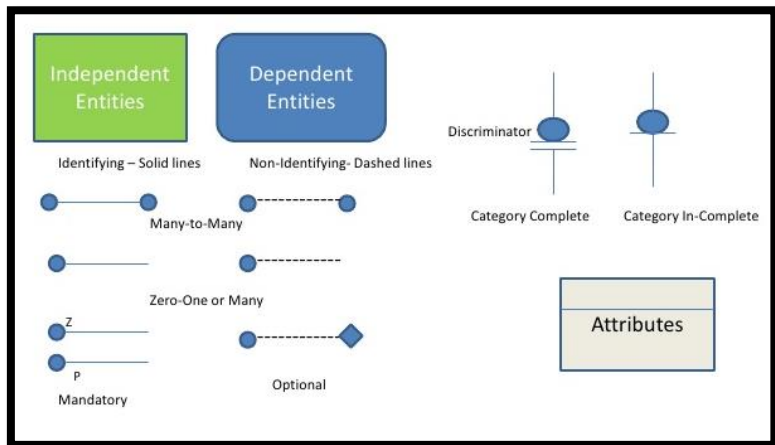
- Contrário da especialização;
- Abordagem bottom-up → parte-se de entidades que formam um subgrupo para definir grupos que englobem e generalizem.



- Várias notações para representação de objetos do modelo conceitual;
- DER → notação de Peter Chen.

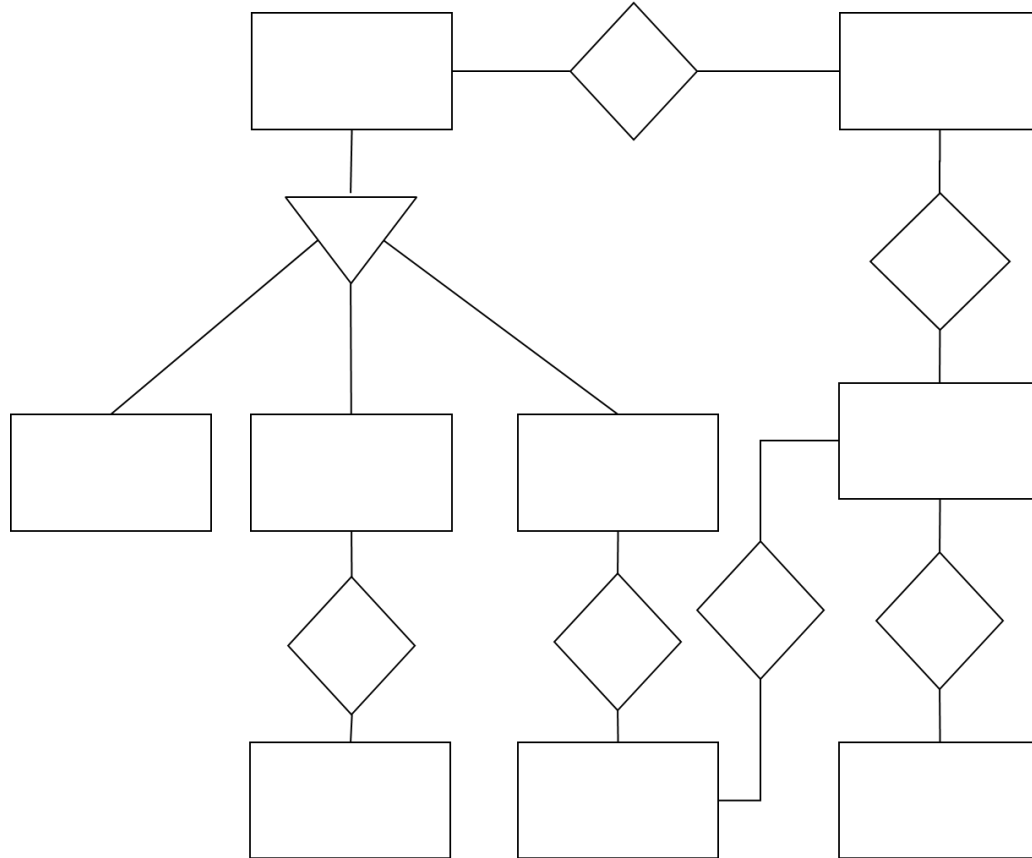


- Notação James Martin
 - “Pé-de-galinha” / “Crow’s foot”



- Notação IDEF1X

- Elaborar um DER para atender as necessidades de uma empresa que deseja controlar os seus equipamentos de informática;
- Existem vários tipos de equipamento, tais como: CPU, impressoras, modem, monitor, teclado, mouse, etc.;
- Deseja-se ter o cadastro dos usuários de cada equipamento e as manutenções solicitadas / realizadas em cada equipamento;
- Cada equipamento está fisicamente em um departamento da empresa e somente o empregado responsável pelo departamento pode solicitar a compra de um novo equipamento.



- ☑ O Diagrama de Entidades e Relacionamentos (DER) ainda usado em larga escala na atualidade;
- ☑ Entidades, atributos, relacionamentos, especialização e generalização são ótimos recursos para representar o MCD;
- ☑ Notações Gráficas: pode-se utilizar a que tiver mais afinidade, mas se precisar construir um DER, obrigatoriamente usa-se a notação de Peter Chen.

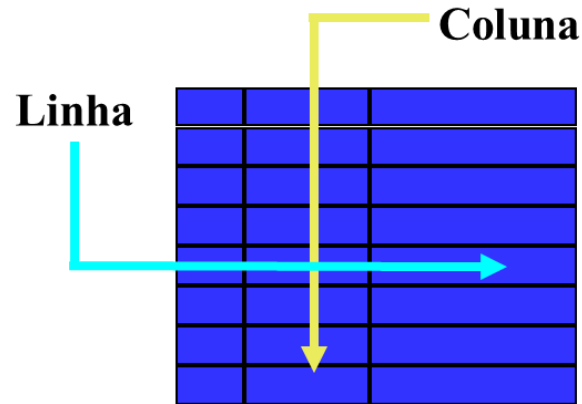
- ❑ O modelo lógico relacional e as 12 regras de Codd.



Aula 1.3.1. O Modelo Lógico Relacional e as 12 Regras de Codd (Parte 1)

- ☐ O Modelo Lógico Relacional.

- Criado por Edgar Frank Codd, em junho de 1970;
 - Artigo *A Relational Model of Data for Large Shared Data Banks*.
- Representação dos dados é independente de como eles são organizados internamente nos servidores;
- Representação tabular dos dados;
 - Linhas e Colunas.
- Baseado na Álgebra Relacional;



O Modelo Lógico Relacional

- Dados sendo vistos como uma tupla → linha formada por uma lista ordenada de colunas;

COD_CLIENTE	NOM_CLIENTE	NUM_TEL_CLIENTE	IND_SEXO_CLIENTE
1	JOÃO	33333333	M
2	MARIA	99999999	F
3	JOSÉ	88888888	M
4	TIAGO	66666666	M
5	PEDRO	44444444	M
6	MATEUS	11111111	M
7	ANA	77777777	F
8	TEREZA	55555555	F

- Facilidade para manipular e visualizar dados → “boom” do modelo;

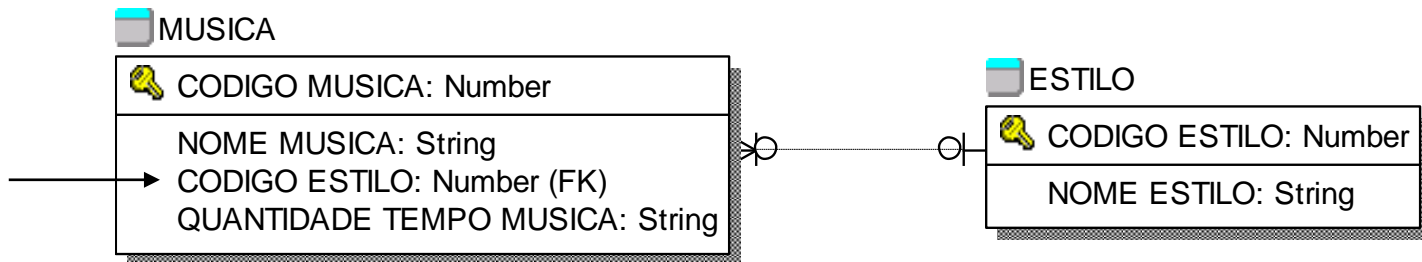
- **Tabela:** objeto correspondente à entidade;
- **Domínio:** natureza dos valores. Exemplos:
 - CODIGO CLIENTE → NUMBER
 - NOME CLIENTE → STRING
 - DATA NASCIMENTO → DATE
- **Coluna (campo):** atributo do modelo conceitual + domínio do dado;

TIPO_PRODUTO



COD_TIPO_PRODUTO: Number
DSC_TIPO_PRODUTO: String

- **Chave Primária (“Primary Key” / “PK”)**: coluna(s) que identificam unicamente uma linha (tupla) em uma tabela ➔ Ex.: CPF;
- **Chave Candidata**: não repetem valor e podem ser PK;
 - Exemplos: TÍTULO DO ELEITOR, RG, CPF.
- **Chave Estrangeira (“Foreign Key” / “FK”)**: elo de ligação;




- **Cardinalidade:** além de representar a quantidade de elementos das entidades envolvidos na relação, indicam também as **restrições de nulidade das chaves estrangeiras**;
 - Notação *Information Engineering (IE)*;

0,1 : 0,N 

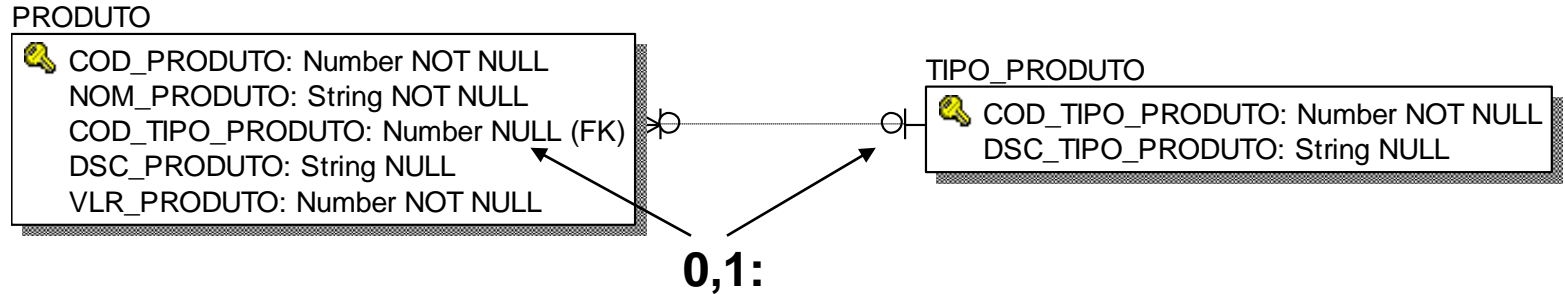
1 : 0,N 

N : N 

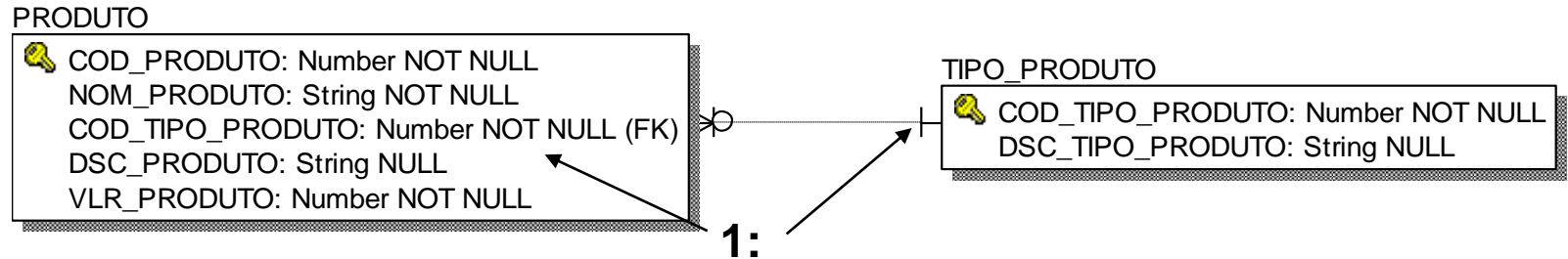
0,1 : 0,1 

1 : 0,1 

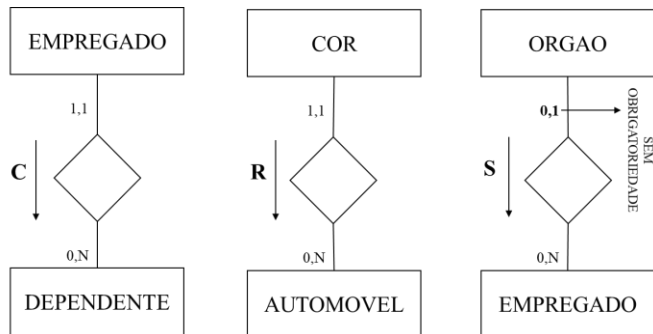
- Cardinalidades **0,1** e **0,N** → chaves estrangeiras nulas;



- Cardinalidades **1:0,N** e **1:0,1** → chaves estrangeiras não nulas;



- **Restrições de Integridade:** regra que garante a consistência dos dados em um relacionamento entre duas tabelas;
 - **Cascade (C):** deleção em cascata;
 - **Restricit (R):** não permite deletar um registro que seja chave estrangeira;
 - **Set Null (S):** seta chave estrangeira para nulo.



- **Restrições de Integridade:** algoritmo para auxiliar a definição.

Posso Eliminar Chave Primária (registro pai)?

Se não puder → RESTRICT

Se puder:

Preciso Manter os Filhos (registros na tabela da FK)?

Se precisar → SET NULL

Se não precisar → CASCADE

☒ O Modelo Lógico Relacional.

- ☐ As Doze Regras de Codd.
- ☐ Derivação do Modelo Lógico Relacional.



Aula 1.3.2. O Modelo Lógico Relacional e as 12 Regras de Codd (Parte 2)

- ☐ As Doze Regras de Codd.
- ☐ Derivação do Modelo Lógico Relacional.

- Publicadas em 1981, com o propósito de definir o que seria necessário para um SGBD ser considerado relacional;

→ Proteger visão relacional dos fornecedores de SGBDs ←

- Numeradas de 1 à 12;
- **REGRA 0:** regra base para toda as outras 12;
 - Gerenciamento exclusivamente pelas capacidades relacionais do SGBD.
- **REGRA 1:** A Regra da Informação
 - Representada explicitamente no nível lógico e por valores em tabelas.

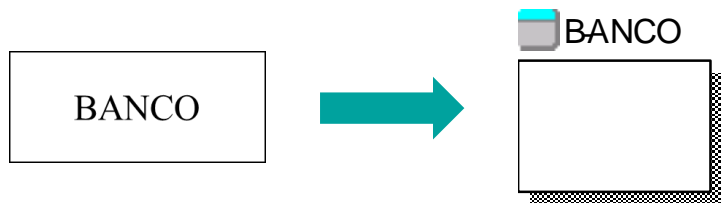
- **REGRA 2:** Regra de Garantia de Acesso:
 - Logicamente acessível com nome da tabela + chave + coluna.
- **REGRA 3:** Tratamento Sistemático de Valores Nulos:
 - Possibilidade de valores nulos para as colunas.
- **REGRA 4:** Catálogo Online e Dinâmico Baseado no Modelo Relacional.
- **REGRA 5:** Regra da Linguagem de Dados Compreensiva:
 - Linguagem de alto nível: DML, DDL, DCL.
- **REGRA 6:** Regra da Atualização de Visões.

- **REGRA 7:** Inserções, Atualizações e Deleções de Alto Nível:
 - Otimização das tarefas do SGBD e melhor experiência do usuário.
- **REGRA 8:** Independência Física dos Dados:
 - Imutabilidade dos programas com alterações no armazenamento dos dados.
- **REGRA 9:** Independência Lógica dos Dados:
 - Imutabilidade dos comandos de manipulação dos dados após alterações de estruturas das tabelas que preservam os dados originais das mesmas.
- **REGRA 10:** Independência de Integridade:
 - Deixa à cargo do banco de dados a garantia da integridade dos dados.

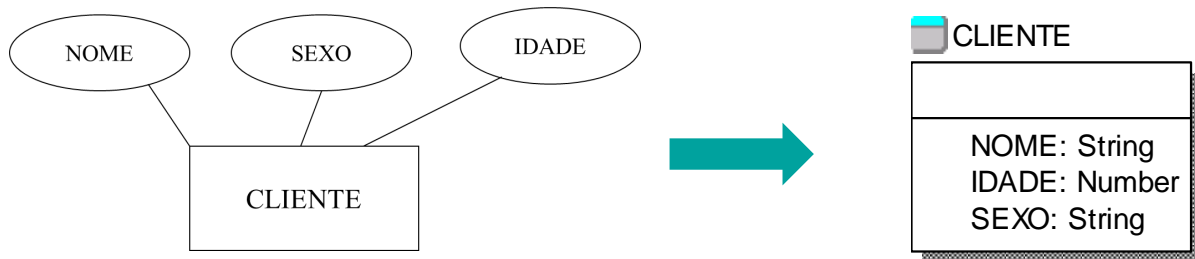
- **REGRA 11:** Independência de Distribuição:
 - Imutabilidade dos programas quando uma distribuição de dados for introduzida no banco de dados ou quando os dados forem redistribuídos.

- **REGRA 12:** Regra de Não Subversão:
 - Bloquear as intervenções diretas nos dados através de interfaces de baixo nível, de forma a preservar as regras relacionais do SGBD.

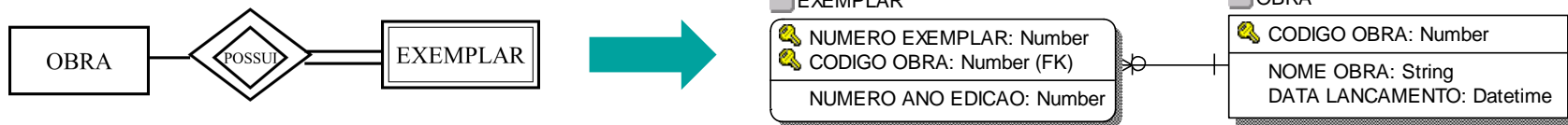
- Modelo de Dados Conceitual (MCD) → **Modelo Lógico Relacional**
- Uma entidade forte normalmente vira uma tabela;



- Os atributos viram campos (colunas) nas tabelas;



- Entidade fraca vira uma tabela → PK composta também FK;

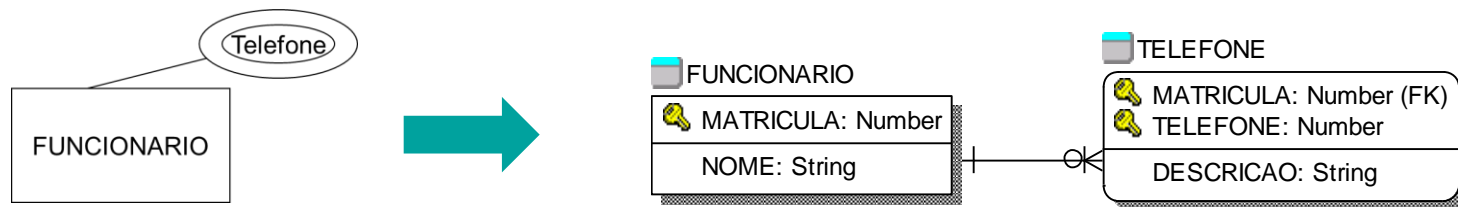


- Atributo chave vira a chave primária da tabela;

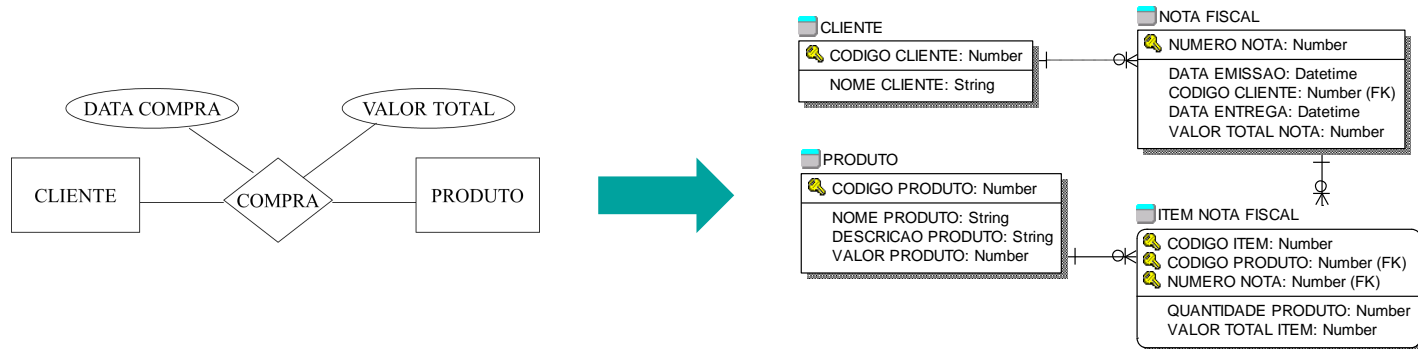


Derivação do Modelo Lógico Relacional

- Atributo Multivalorado → tabela ao invés de campo string (*array*);



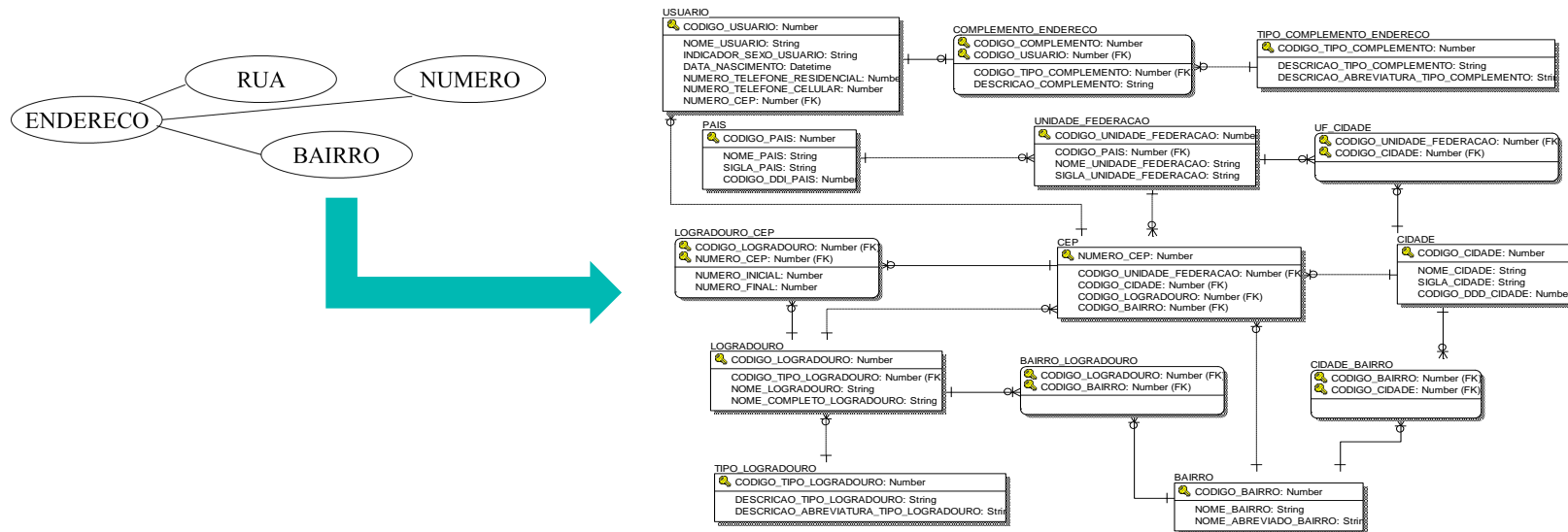
- Relacionamento com atributo vira uma ou mais tabelas;



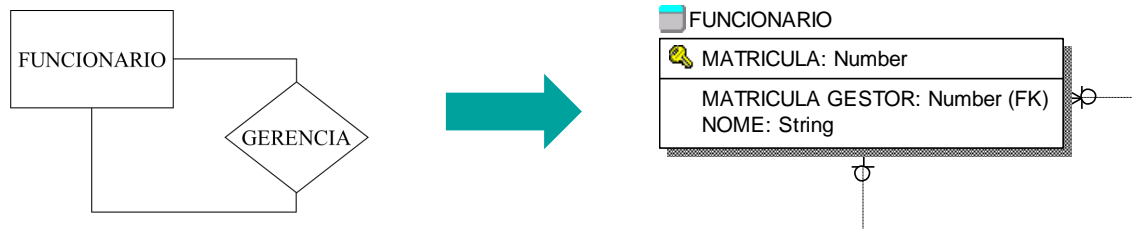
Derivação do Modelo Lógico Relacional

- Atributos Compostos

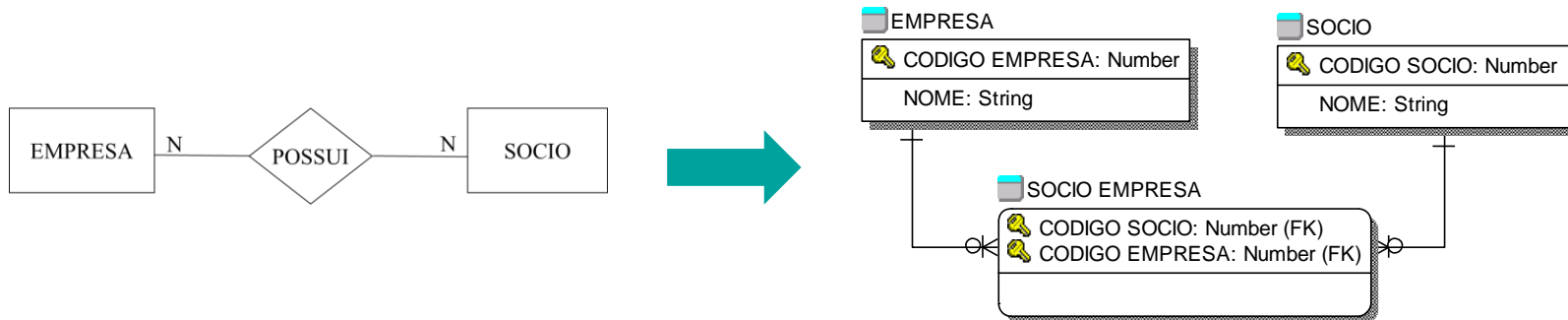
- Parte da tabela ou nova(s) tabela(s)*



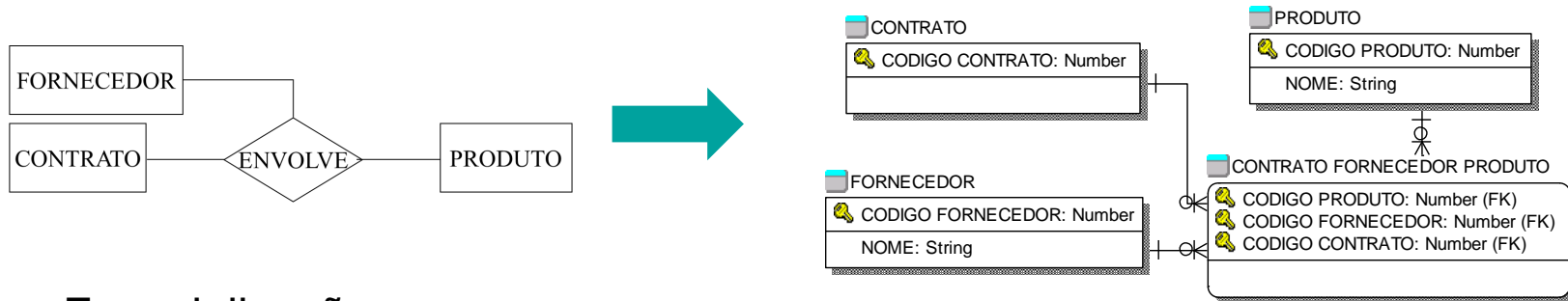
- Relacionamento recursivo → PK será a FK (renomeada);



- Relacionamento N:N → tabela associativa;



- Relacionamento com grau \geq ternário \rightarrow tabela com chave composta;



- Especialização
 - Campos comuns às tabelas especializadas \rightarrow na tabela não especializada;
 - PK das tabelas especializadas composta pelas FKs também.
- Generalização \rightarrow conversão direta.

- ☑ O Modelo Lógico Relacional é a etapa seguinte à construção do modelo conceitual.
- ☑ As Doze Regras de Codd foram publicadas para garantir que a visão relacional não fosse desviada pelos fornecedores da época.
- ☑ Derivação do Modelo Lógico Relacional é feita a partir do modelo conceitual, seguindo algumas regras e recomendações.

Próxima aula

☐ Normalização.



Aula 1.4. Normalização

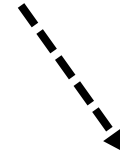
- ☐ Benefícios da Normalização.
- ☐ Primeira forma normal (1FN).
- ☐ Segunda forma normal (2FN).
- ☐ Terceira forma normal (FNBC).
- ☐ Demais formas normais.

- Normalização é o processo que avalia os atributos de uma entidade, com o objetivo de evitar problemas na inclusão, alteração, exclusão e consulta de tuplas ➔ **eliminar redundâncias e inconsistências.**
- Introduzida por Codd, em 1970 ➔ Primeira Forma Normal (1FN);
- Evoluído até a Quinta Forma Normal (5FN);
- Ajustar o modelo lógico ao percorrer sequencialmente pelas formas normais disponíveis;
- Para cada FN ➔ há um objetivo e um roteiro para conseguir alcançá-lo.

Primeira forma normal (1FN)

- Tabela está na 1FN quando não possuir grupos de registros repetitivos.


Tabela de Empregado					Grupo	
NumMat	NomEmp	NomCar	DatCar	VlrSal	CodOrg	NomOrg
1000	João	Analista	11052000	R\$1.500,00	10	CPD
1000	João	Gerente	30122002	R\$4.000,00	10	CPD
1001	Marcos	Diretor	15072003	R\$6.000,00	10	CPD



EMPREGADO

 NUM_MATRICULA
NOM_EMPREGADO
NOM_CARGO
VLR_SALARIO
DAT_INICIO_CARGO
COD_ORGAO (FK)

ORGAO

 COD_ORGAO
NOM_ORGAO



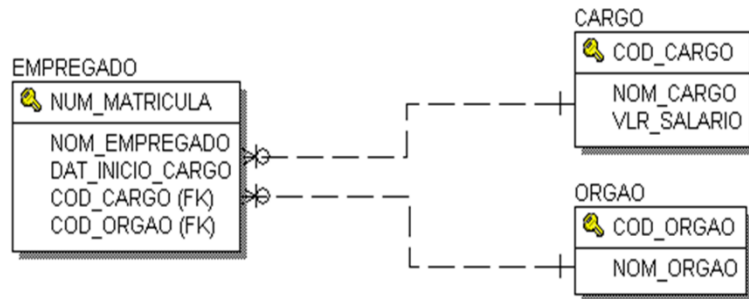
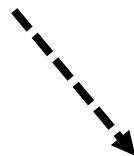
NumMat	NomEmp	NomCar	DatCar	VlrSal	CodOrg
1000	João	Analista	11052000	R\$1.500,00	10
1000	João	Gerente	30122002	R\$4.000,00	10
1001	Marcos	Diretor	15072003	R\$6.000,00	10

Segunda forma normal (2FN)

- Tabela está na 2FN se e somente se estiver na 1FN e quando todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela)



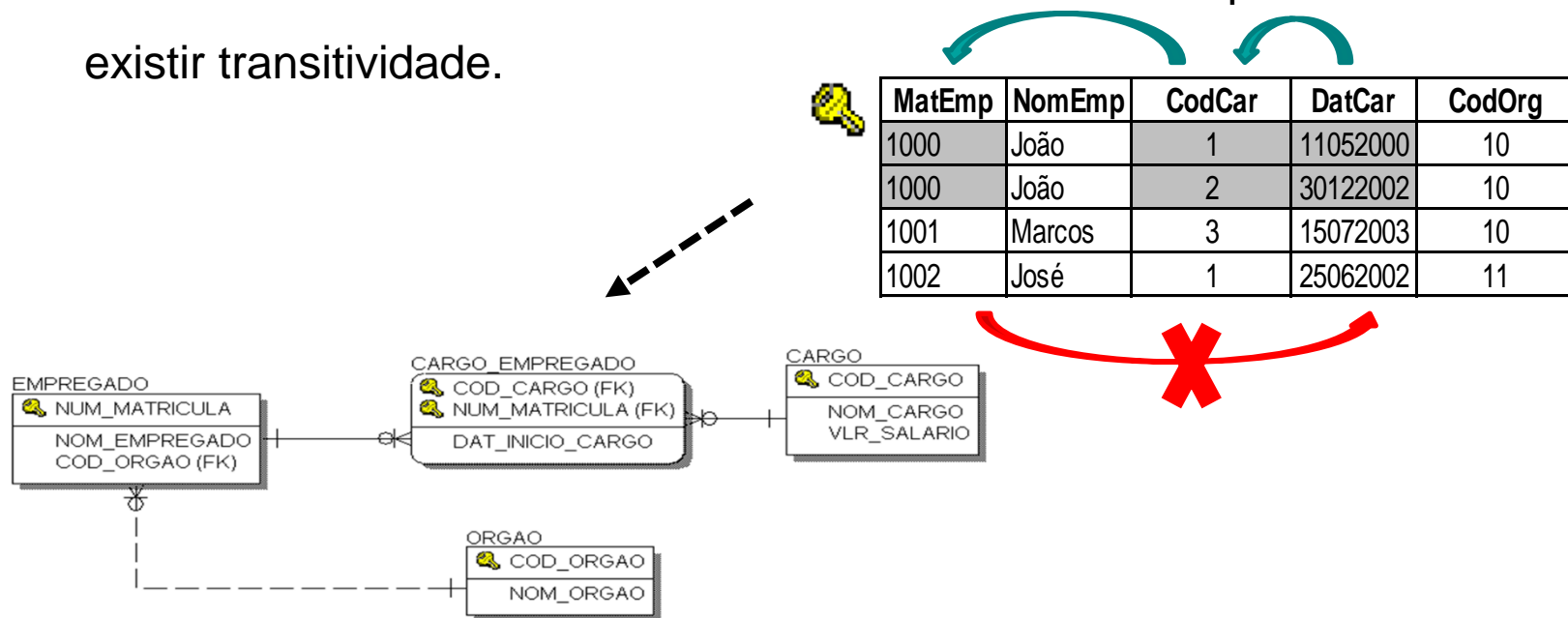
NumMat	NomEmp	NomCar	DatCar	VlrSal	CodOrg
1000	João	Analista	11052000	R\$1.500,00	10
1000	João	Gerente	30122002	R\$4.000,00	10
1001	Marcos	Diretor	15072003	R\$6.000,00	10
1002	José	Analista	25062002	R\$ 1.500,00	11



MatEmp	NomEmp	CodCar	DatCar	CodOrg
1000	João	1	11052000	10
1000	João	2	30122002	10
1001	Marcos	3	15072003	10
1002	José	1	25062002	11

Terceira forma normal (FNBC)

- **Forma Normal de Boyce e Codd** → corrigiu problemas da 3FN;
- Tabela está na 3FN se e somente se estiver na 2FN e quando não existir transitividade.



- Na grande maioria dos projetos, encerra-se a normalização na 3FN;
 - Baixíssimo nível de redundância e nenhuma inconsistência.
- **Quarta Forma Normal (4FN):** uma tabela está na 4FN se e somente se estiver na 3FN e não existirem dependências multivaloradas.

3FN: *TAB_LIVRO (Num_Livro, Titulo, Cod_Editora, Ano_publicacao)*

TAB_EDITORA (Cod_Editora, Nom_Editora)

TAB_LIVROAUTORASSUNTO(Num_Livro, Cod_Autor, Cod_Assunto)

4FN: *TAB_LIVRO (Num_Livro, Titulo, Cod_Editora, Ano_publicacao)*

TAB_EDITORA (Cod_Editora, Nom_Editora)

TAB_LIVROAUTOR (Num_Livro, Cod_Autor)

TAB_LIVROASSUNTO (Num_Livro, Cod_Assunto)

- **Quinta Forma Normal (5FN):** relacionamentos múltiplos (\geq ternários).

- ☑ Benefícios da Normalização: reduzir / eliminar redundâncias e inconsistências.
- ☑ Primeira e segunda formas normais são essenciais.
- ☑ Terceira forma normal (FNBC) recomendada na grande maioria de projetos de bancos de dados.

Próxima aula

☐ Modelo físico.



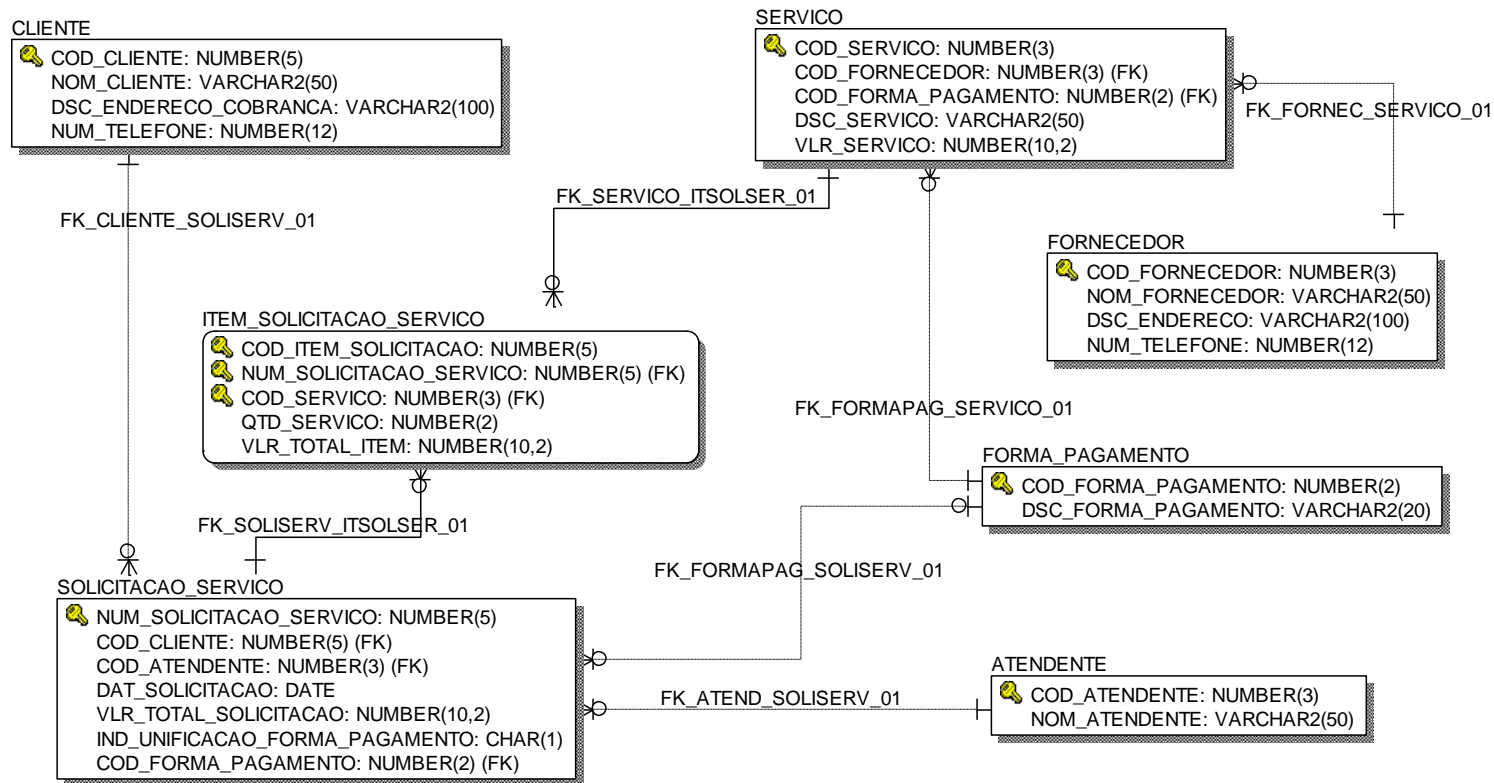
Aula 1.5. Modelo Físico

Nesta aula

- ☐ O modelo físico.
- ☐ Geração do modelo físico.
- ☐ Linguagem SQL.

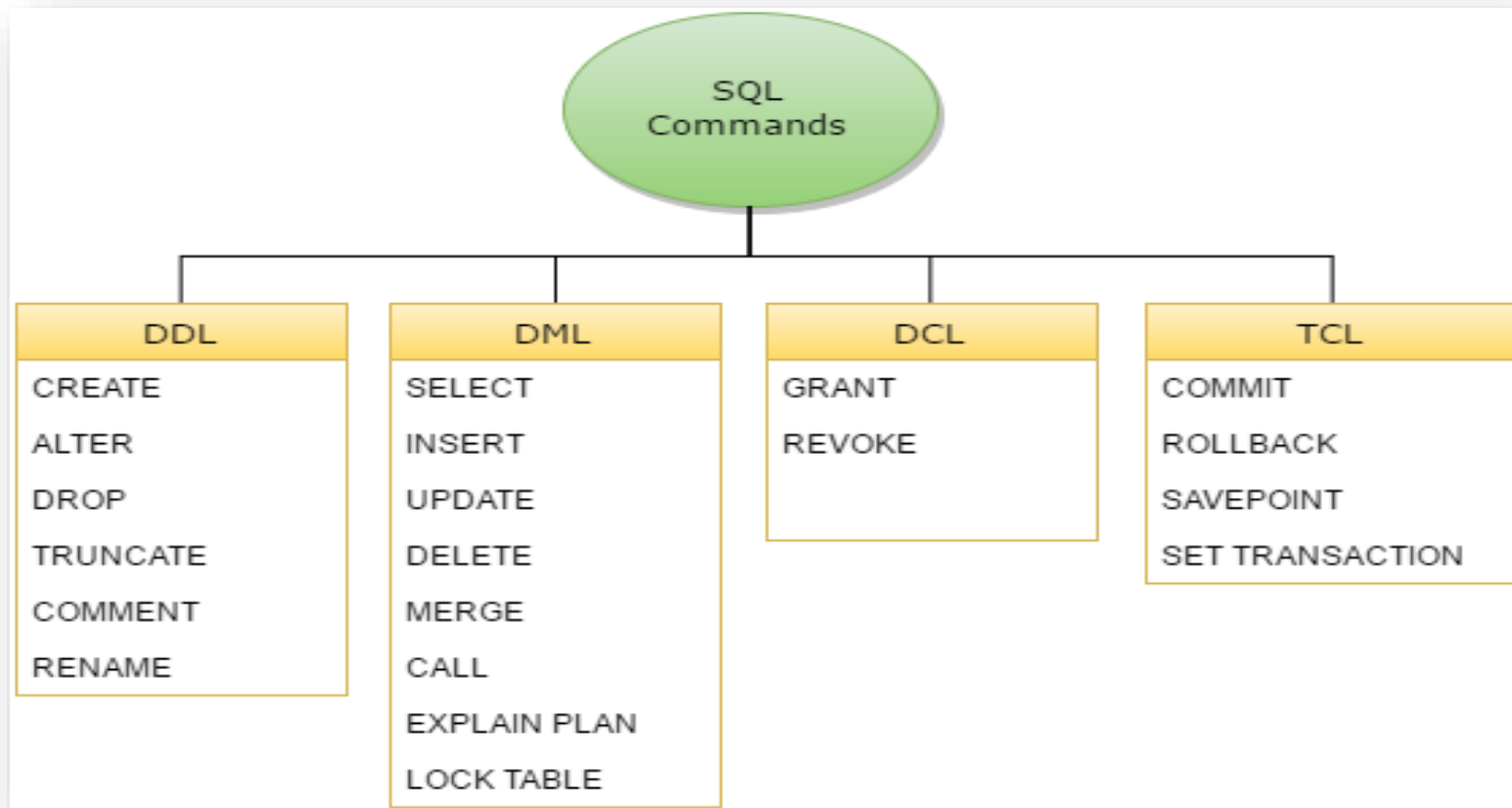
- **MFD** → última etapa no processo de modelagem de dados;
- Sem abstração de visão → coerente com a abordagem (relacional, hierárquica, de rede ou NoSQL) escolhida na derivação do MLD;
- Sem independência física de dados → totalmente dependente do SGBD selecionado para armazenar as estruturas de dados;
- Criação das estruturas físicas (tabelas, colunas, visões, etc);
- Regras de segurança e acesso aos dados (usuários e perfis);
- Limitador → recursos oferecidos pelo SGBD selecionado.

O Modelo Físico

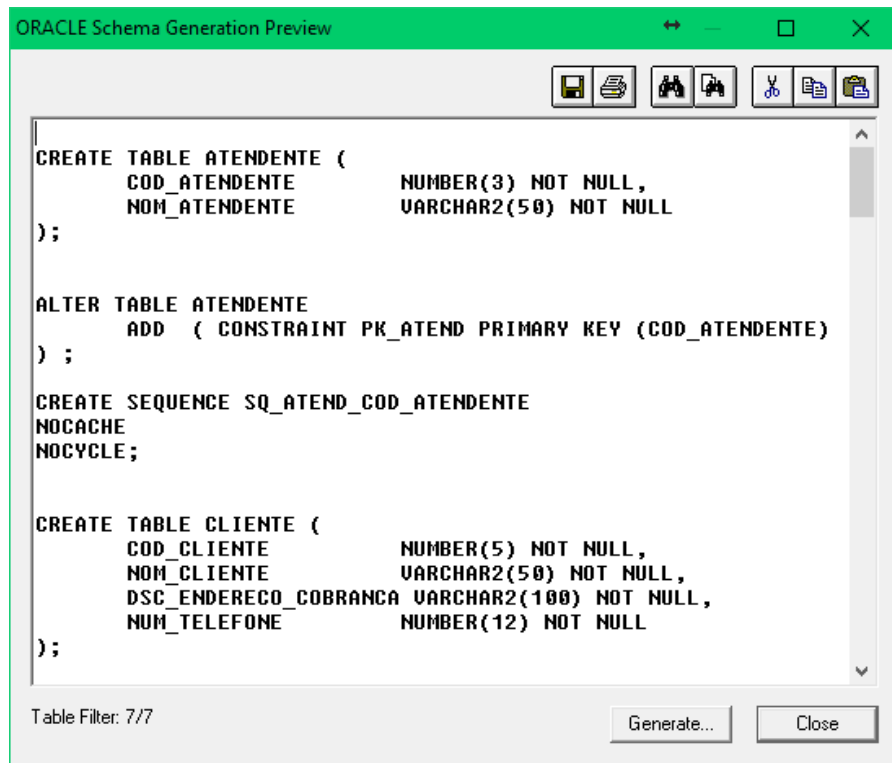


- MLD + Recursos Disponíveis do SGBD → MFD.
- Preocupação com a performance:
 - Tipos de dados adequados;
 - Índices;
 - Código compilado armazenado;
 - Distribuição / Particionamento de dados.
- Produto final dessa etapa:
 - Script DDL (Data Definition Language) → criação das estruturas.
 - Script DCL (Data Control Language) → usuários e privilégios necessários.

- SQL (*Structured Query Language*)
 - Criada em 1974 com nome inicial de “*SEQUEL*”;
 - Teve como base os fundamentos do modelo relacional de Codd;
 - Revisada entre 1976 e 1977 → nome alterado para “*SQL*” por razões jurídicas.
- Para criação e utilização das estruturas de dados;
- Em 1982 → definida pelo *American National Standard Institute (ANSI)* como a linguagem padrão oficial para bancos de dados relacionais;
 - SGBDs implementam pequenas variações e comandos específicos;
 - Continua um padrão, com pequenas diferenças na sintaxe.



- A maioria dos SGBDs fornece interface gráfica para implementação do modelo físico de dados;
- Utilização de scripts SQL:
 - Versionamento;
 - Validações.



```
CREATE TABLE ATENDENTE (
  COD_ATENDENTE      NUMBER(3) NOT NULL,
  NOM_ATENDENTE      VARCHAR2(50) NOT NULL
);

ALTER TABLE ATENDENTE
  ADD ( CONSTRAINT PK_ATEND PRIMARY KEY (COD_ATENDENTE)
) ;

CREATE SEQUENCE SQ_ATEND_COD_ATENDENTE
NOCACHE
NOCYCLE;

CREATE TABLE CLIENTE (
  COD_CLIENTE        NUMBER(5) NOT NULL,
  NOM_CLIENTE        VARCHAR2(50) NOT NULL,
  DSC_ENDERECO_COBRANCA VARCHAR2(100) NOT NULL,
  NUM_TELEFONE       NUMBER(12) NOT NULL
);
```

Table Filter: 7/7

Generate... Close

- ☑ O Modelo Físico é a etapa final da modelagem em um projeto de banco de dados.
- ☑ Geração do Modelo Físico a partir do MLD, sem abstração de visão e sem independência física de dados (dependente do SGBD).
- ☑ Linguagem SQL para criação e utilização das estruturas de dados em um SGBD relacional.

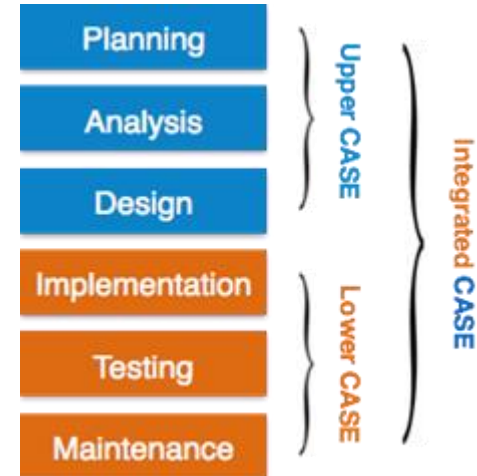
- ❑ Ferramentas Case de Mercado para Modelagem Relacional.



Aula 1.6. Ferramentas Case de Mercado para Modelagem Relacional

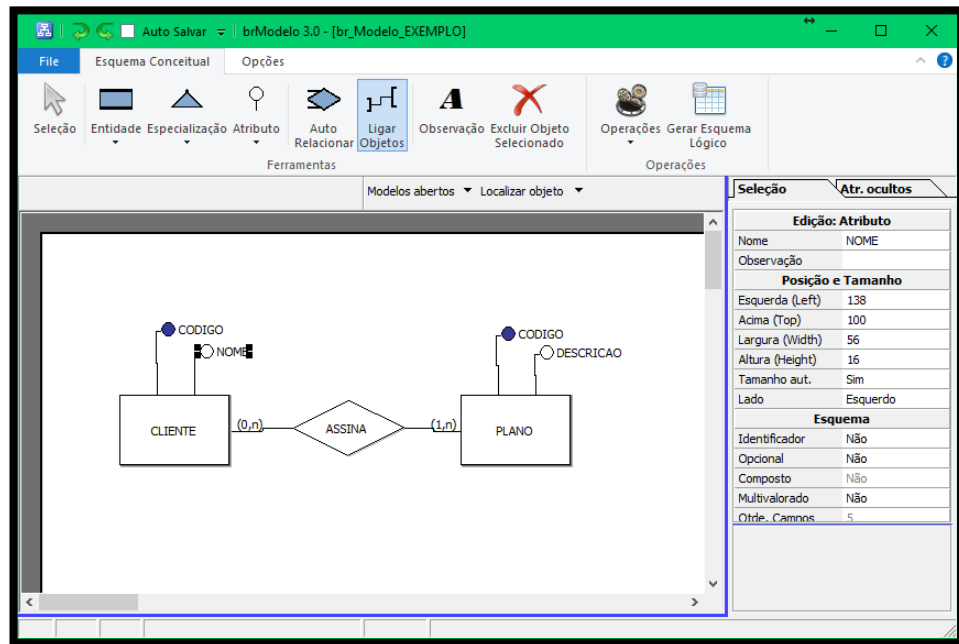
- ☐ Ferramenta CASE.
- ☐ Ferramentas CASE para Modelagem Conceitual.
- ☐ Ferramentas CASE para Modelo Lógico / Físico Relacional.

- **CASE:** *Computer-Aided Software Engineering;*
- Softwares que auxiliam no processo de desenvolvimento de sistemas;
 - Análise de requisitos;
 - Modelagem de dados e de classes;
 - Modelagem de processos, etc.;
 - Programação e testes.
- Viés de design e documentação:
 - Facilita também a conversão do produto final de uma etapa para o início da próxima.



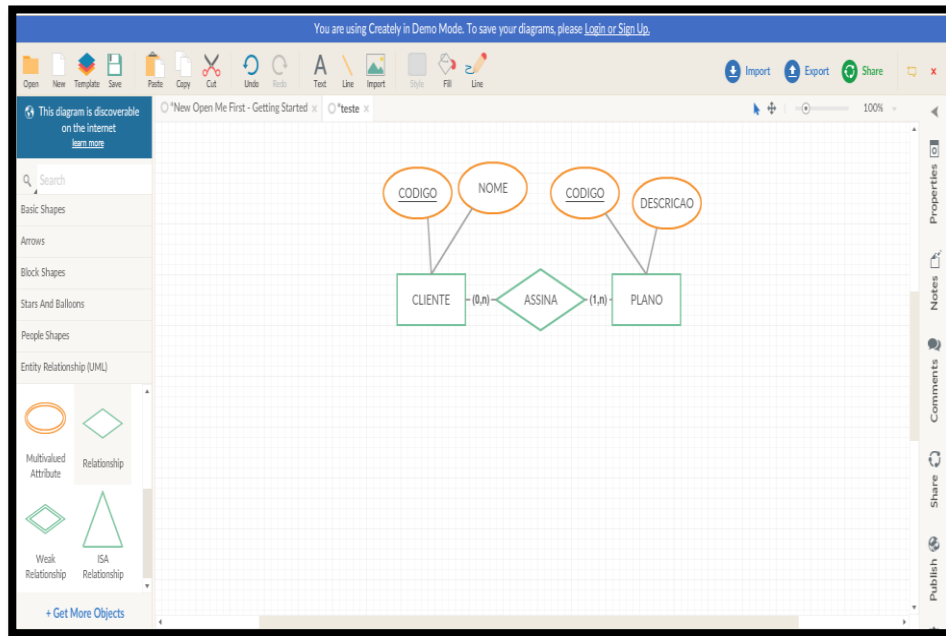
▪ *brModelo*

- Gratuita;
- Modelo conceitual
- Modelo lógico
- Modelo físico;
- Não faz engenharia reversa;
- Scripts do schema físico
 - Gerados como ANSI
 - Sem opções específicas do SGBD;
- Link para download: <http://www.sis4.com/brModelo/index.html>.



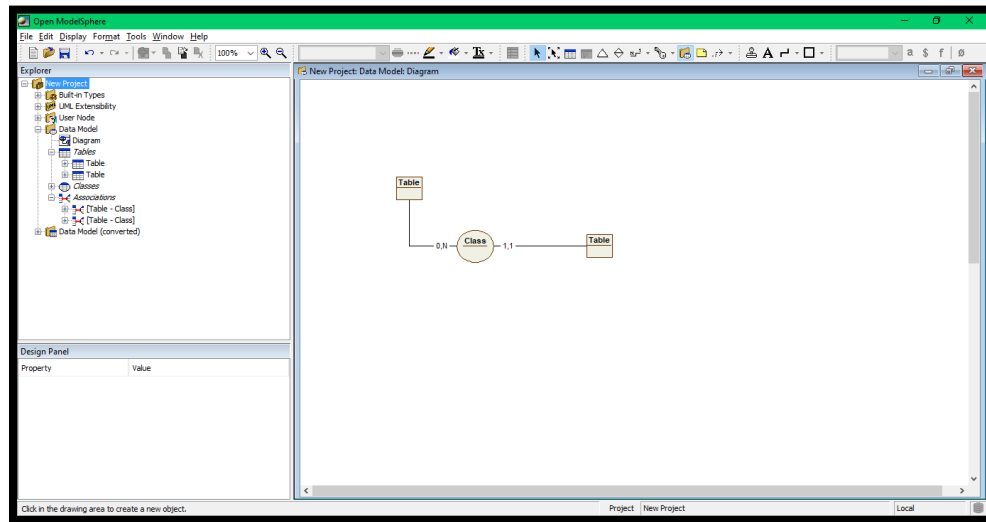
▪ Creately

- On-line e paga;
- Modelo conceitual
- Modelo lógico
- Não faz engenharia reversa;
- Não gera scripts de criação do schema físico;
- <https://creately.com>



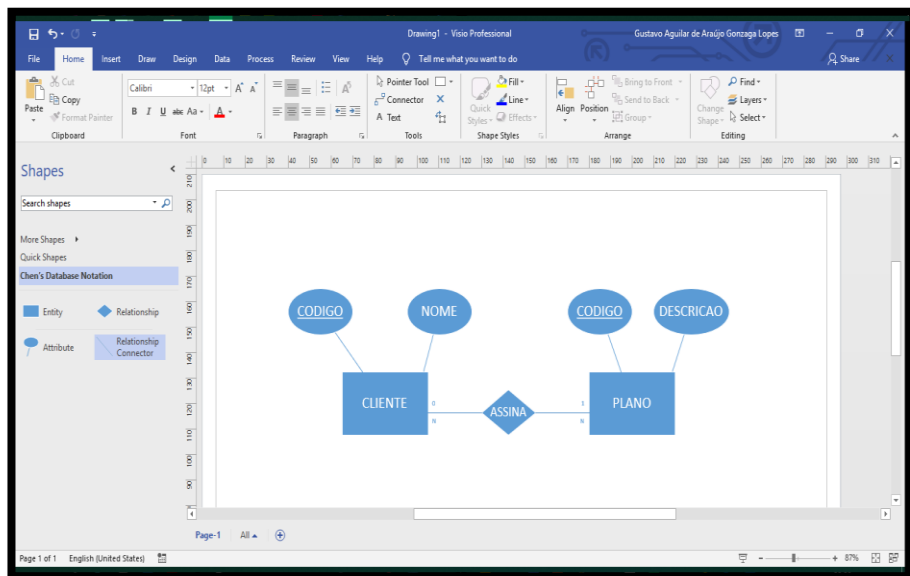
■ Open ModelSphere

- Gratuito;
- MCD + MLD + MFD
- Outros modelos (UML)
- Faz engenharia reversa;
- Scripts do schema físico;
- Faz comparação entre os modelos e o banco de dados;
- Representação no modelo conceitual um pouco diferente da usada no DER;
- <http://www.modelsphere.com/org>.



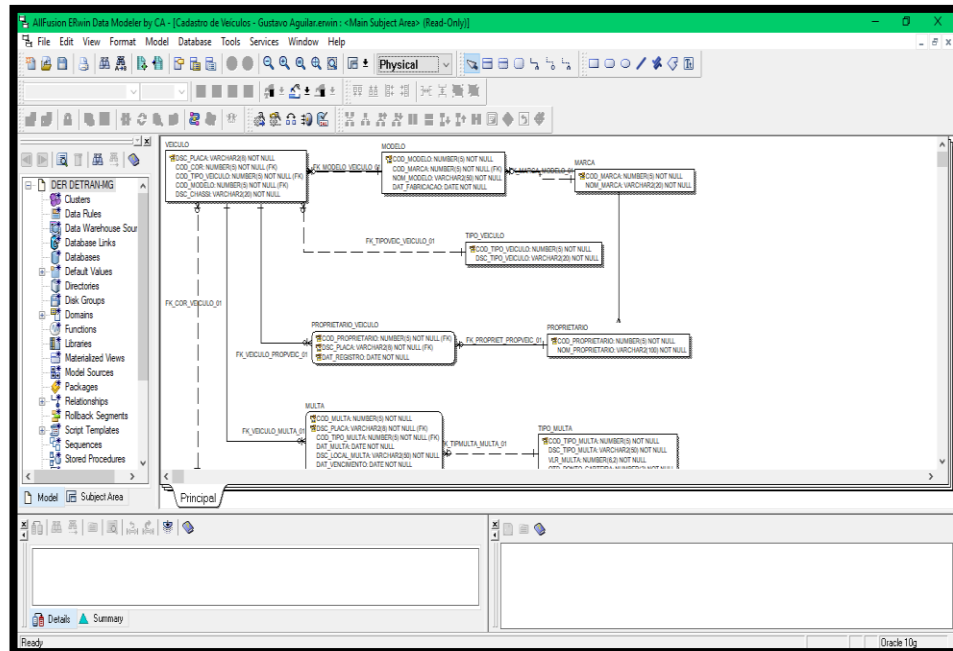
■ Microsoft Viso

- Pago;
- Modelo conceitual
- Modelo lógico
- Outros modelos (BMP, UML);
- Não faz engenharia reversa;
- Não gera scripts;
- Não faz comparação entre os modelos e o banco de dados;
- Link: <https://office.live.com/start/visio.aspx>.



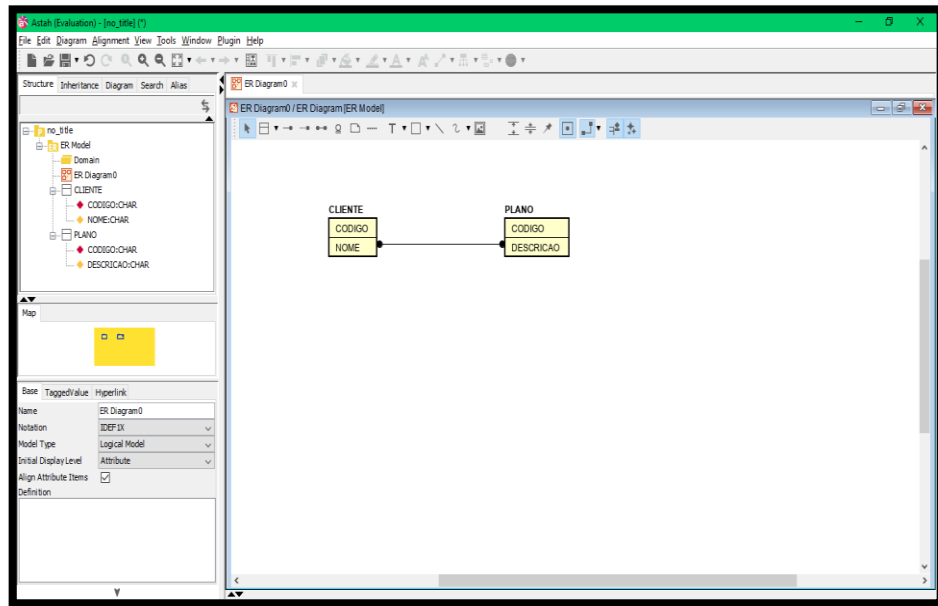
■ Erwin Data Modeler

- Pago, mas muito amigável;
- Modelo lógico e físico;
- Modelagem dimensional;
- Diversos SGBDs;
- Notações IE e IDEF1X;
- Engenharia reversa e *forward*;
- Gera scripts do schema físico;
- Comparação e sincronização entre modelos e o banco de dados.



■ Astah Professional

- Pago
 - Licença gratuita para estudantes;
- Modelo lógico e físico
- Notações IE e IDEF1X;
- Diversos diagramas (Usecase);
- Gera scripts do schema físico;
- Sem engenharia reversa;
- Sem comparação e sincronização com o banco de dados;
- <http://astah.net/editions/professional>.



- Lucidchart
- Draw.io
- DbDesigner.net
- ERBuilder Data Modeler
- MySQL Workbench
- Microsoft SQL Server
Management Studio
- Oracle SQL Developer Data
Modeler
- SQL Maestro

- ☑ Ferramenta CASE para auxiliar e documentar.
- ☑ Ferramentas CASE para Modelagem Conceitual, algumas com suporte aos três tipos de modelos de dados.
- ☑ Ferramentas CASE para Modelo Lógico / Físico Relacional, com algumas até de administração do banco de dados também.

□ Capítulo 2: Mapeamento Objeto Relacional.