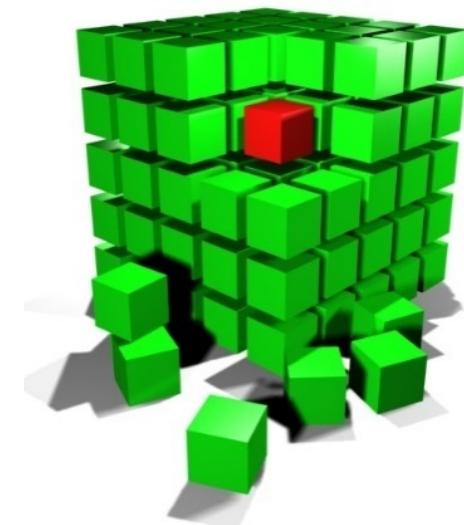


■ Aula 03

- A importância da modelagem de dados
- Modelos de dados
- Regras de negócio
- Evolução dos modelos de dados
- Grau de abstração do modelo



■ Introdução

- O projeto de banco de dados deve focar em como a estrutura do banco de dados será utilizada para armazenar e gerenciar os dados do usuário final.
- Nesse sentido, uma de suas primeiras etapas recebe o nome de **modelagem de dados**.
- Uma característica fundamental para esse tipo de abordagem é que ela oferece um nível de **abstração de dados**.



■ Introdução

- O conceito de abstração está relacionado aos detalhes de como as estruturas utilizadas para o armazenamento dos dados deverão ser criadas.
- A compreensão desse conceito é importante, pois, ele permitirá que situações complexas existentes em um cenário do mundo real possam ser adaptadas para um sistema computacional.
- Um **modelo de dados** oferece os meios necessários para que essa abstração possa ser realizada de forma adequada.

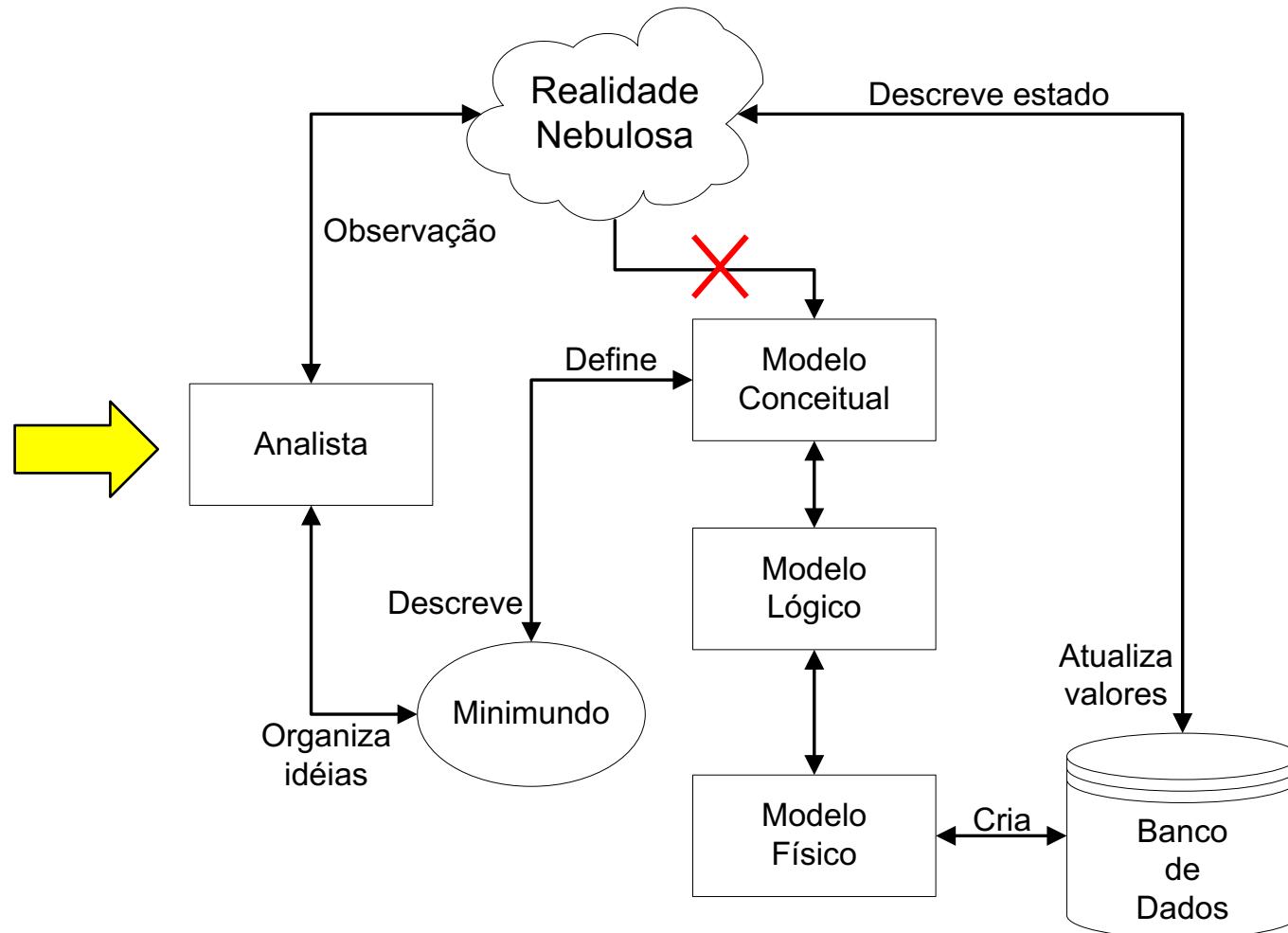


■ Modelo de dados

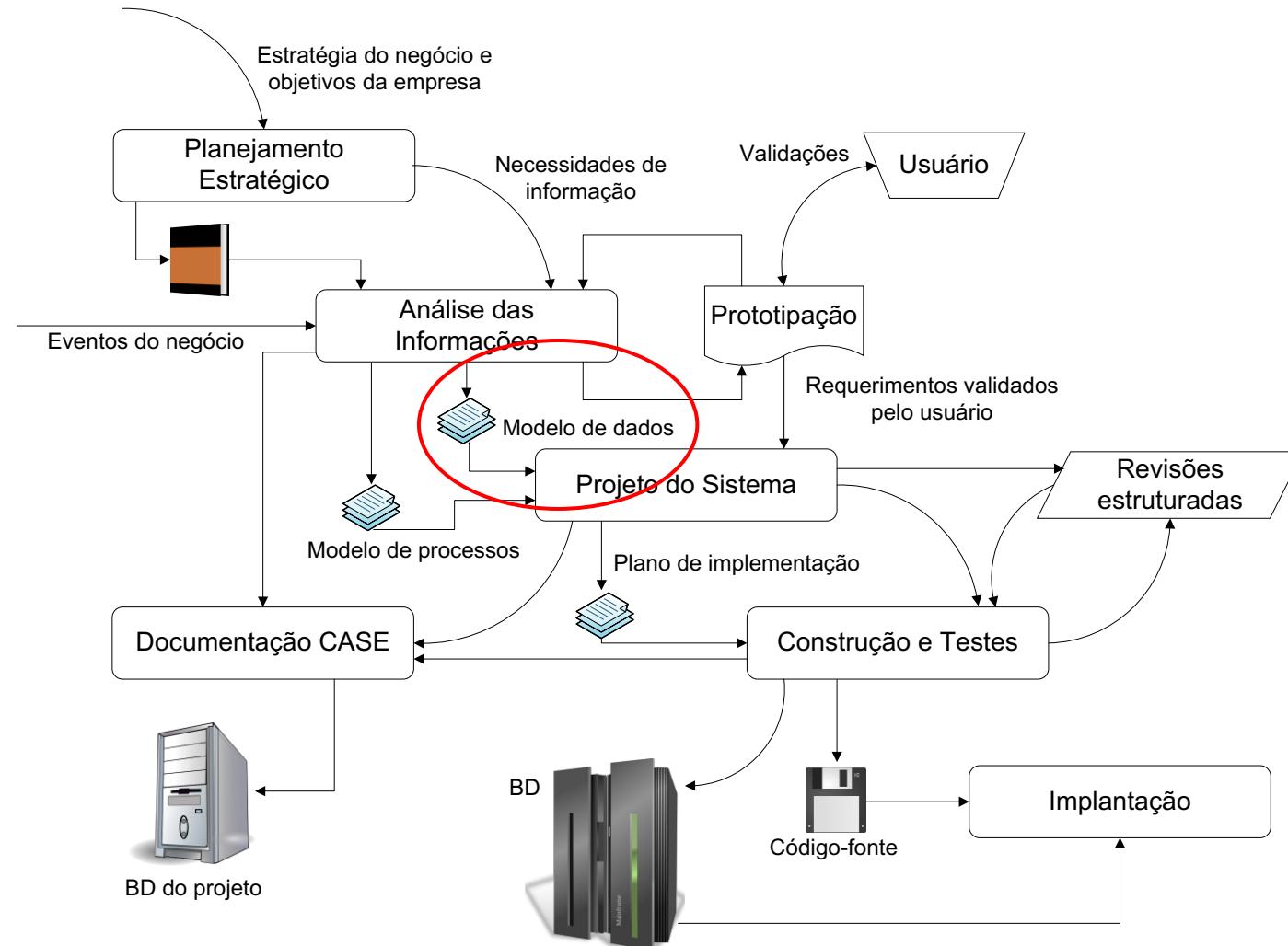
- O **modelo de dados** é uma representação simples e geralmente gráfica, de estruturas de dados mais complexas existentes no mundo real.
- Ele representa uma abstração de um objeto real, sendo que sua principal função é auxiliar no entendimento das complexidades que existem em um determinado cenário.
- O modelo de dados contém as estruturas utilizadas para armazenar os dados, suas características e restrições, além das relações existentes entre elas. Ele também pode conter outros elementos, que tenham a finalidade de dar suporte ao problema específico de um domínio.



■ Modelo de dados



■ Ciclo de vida da Engenharia da Informação

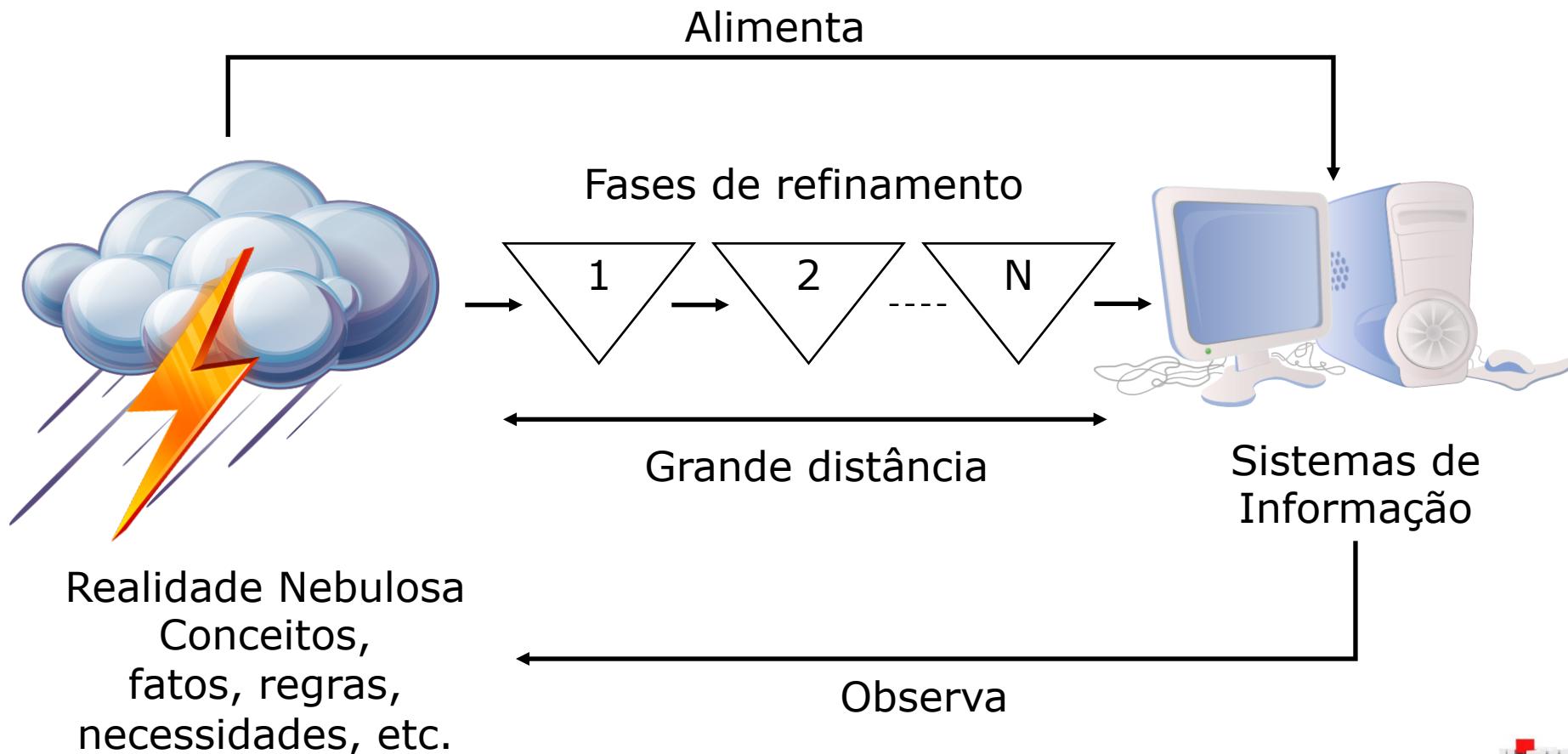


- **Modelo de dados**

- A modelagem de dados é um processo iterativo e progressivo.
- Inicialmente, temos uma **compreensão mais simples** do domínio do problema e, conforme esse nível de compreensão se desenvolve, o nível de detalhes do modelo também aumenta.



■ Modelo de dados



■ **Modelo de dados**

- Um modelo de dados adequado e pronto para ser utilizado para a implementação deve conter pelo menos os seguintes componentes:
 1. Descrição das estruturas que serão utilizadas para armazenar os dados do usuário final.
 2. Conjunto de regras aplicáveis para garantir a integridade dos dados.



Modelos de Dados

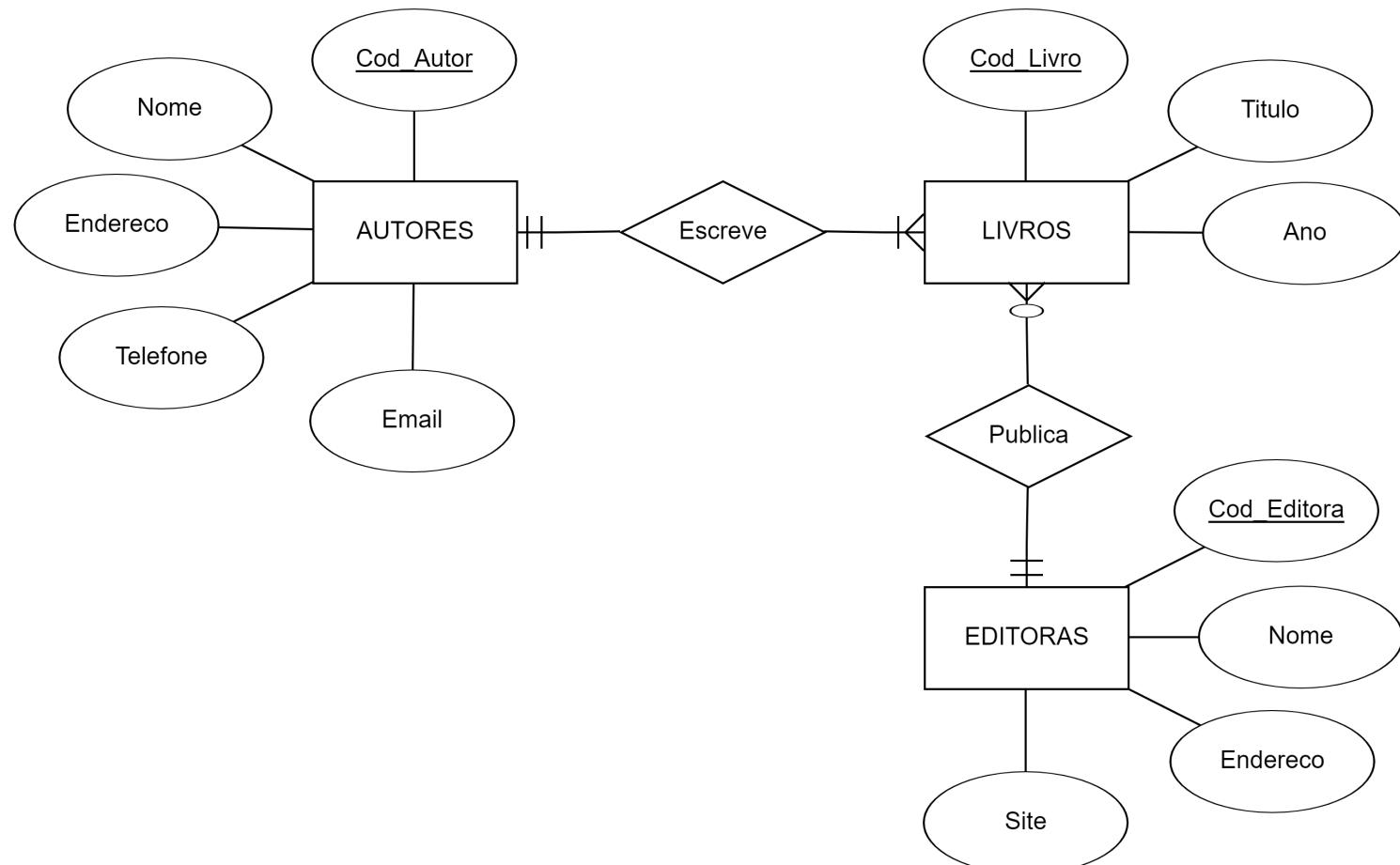
- **Blocos básicos para a construção de modelos**
 - Os blocos básicos utilizados para construir um modelo de dados são: **entidades, atributos, relacionamentos e restrições.**
 - Independente da notação utilizada, esses blocos estarão presentes em todos os modelos de dados que estiverem sendo desenvolvidos.



- Blocos básicos para a construção de modelos**

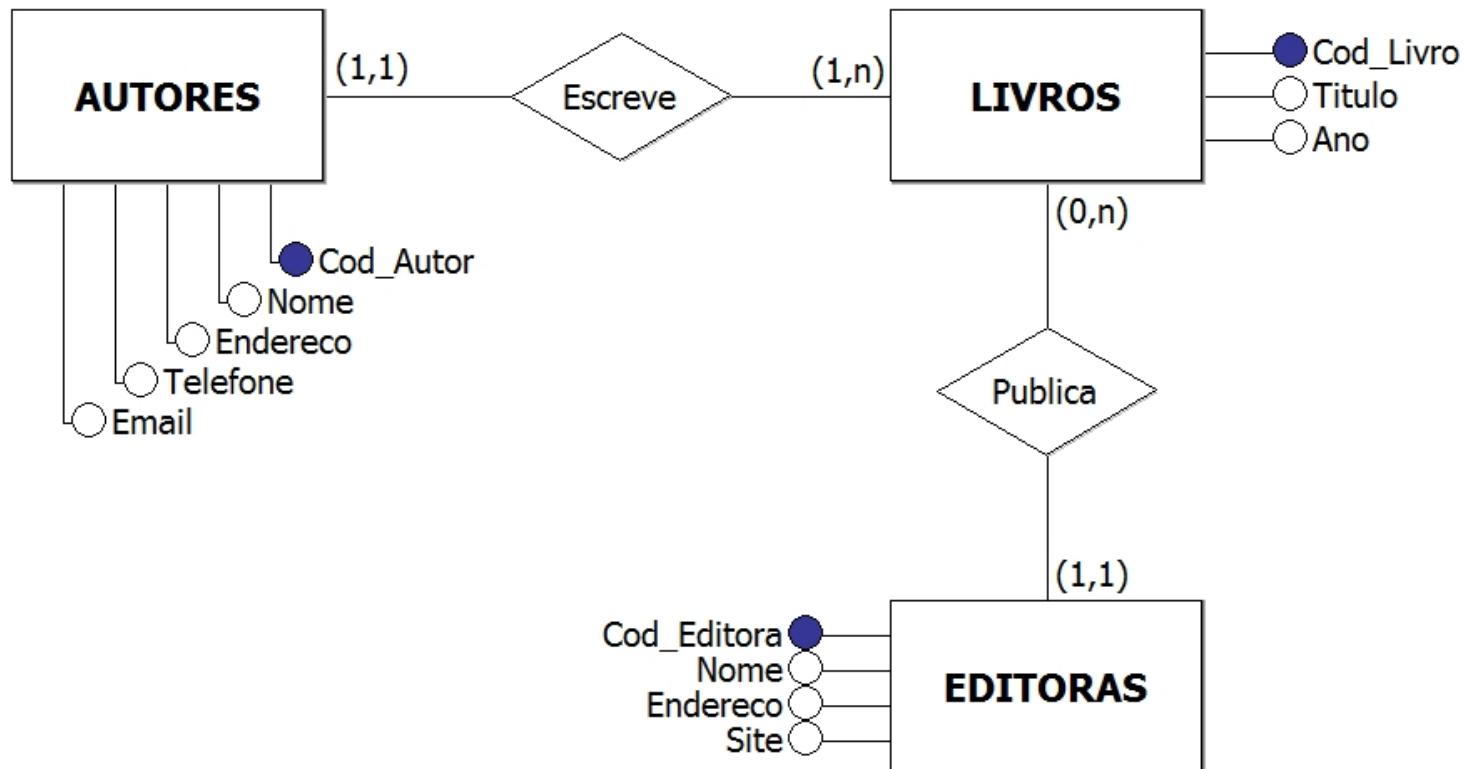


- Exemplo desenvolvido no ERDPlus



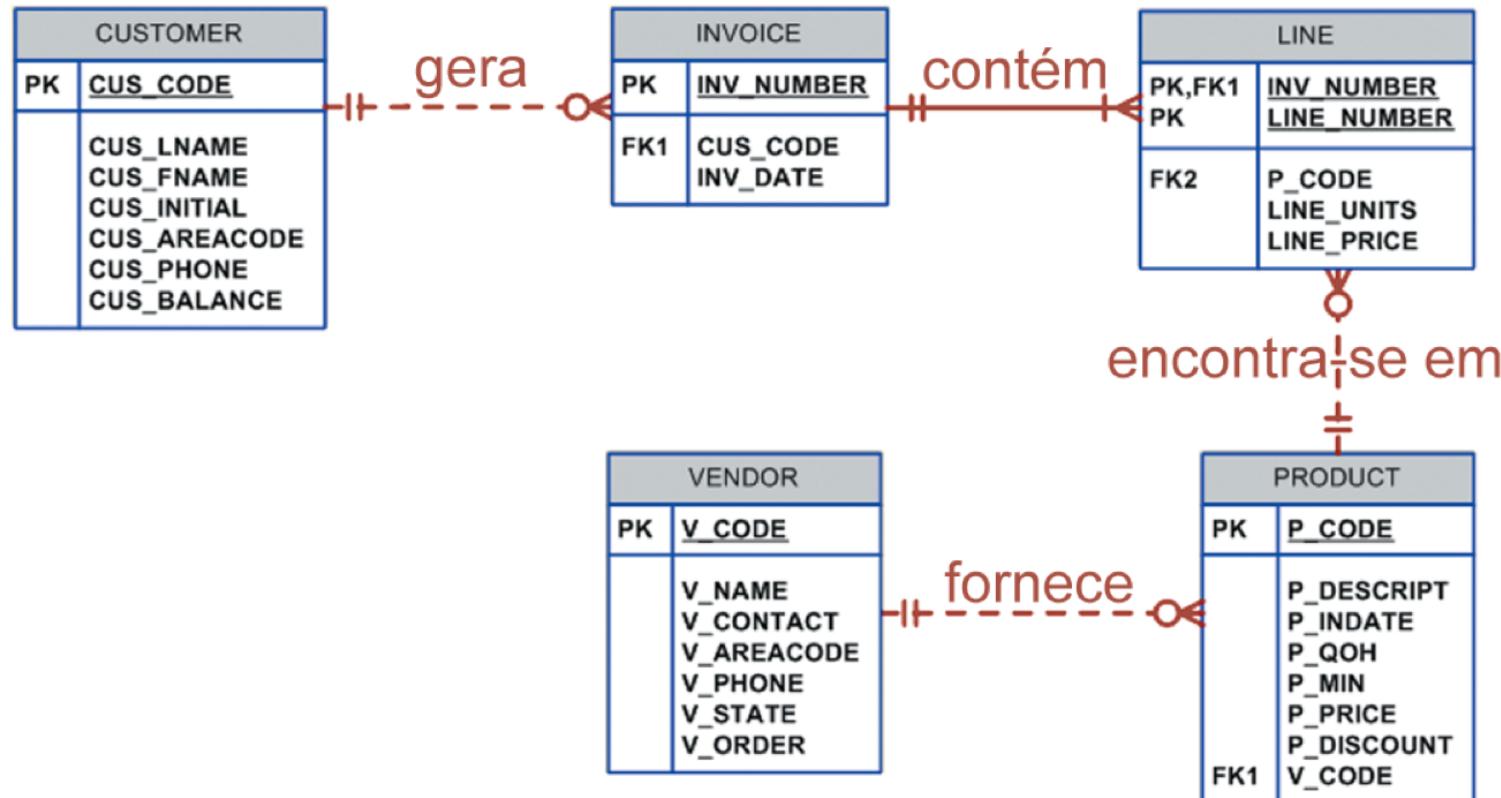
Modelos de Dados

- Exemplo desenvolvido no brModelo



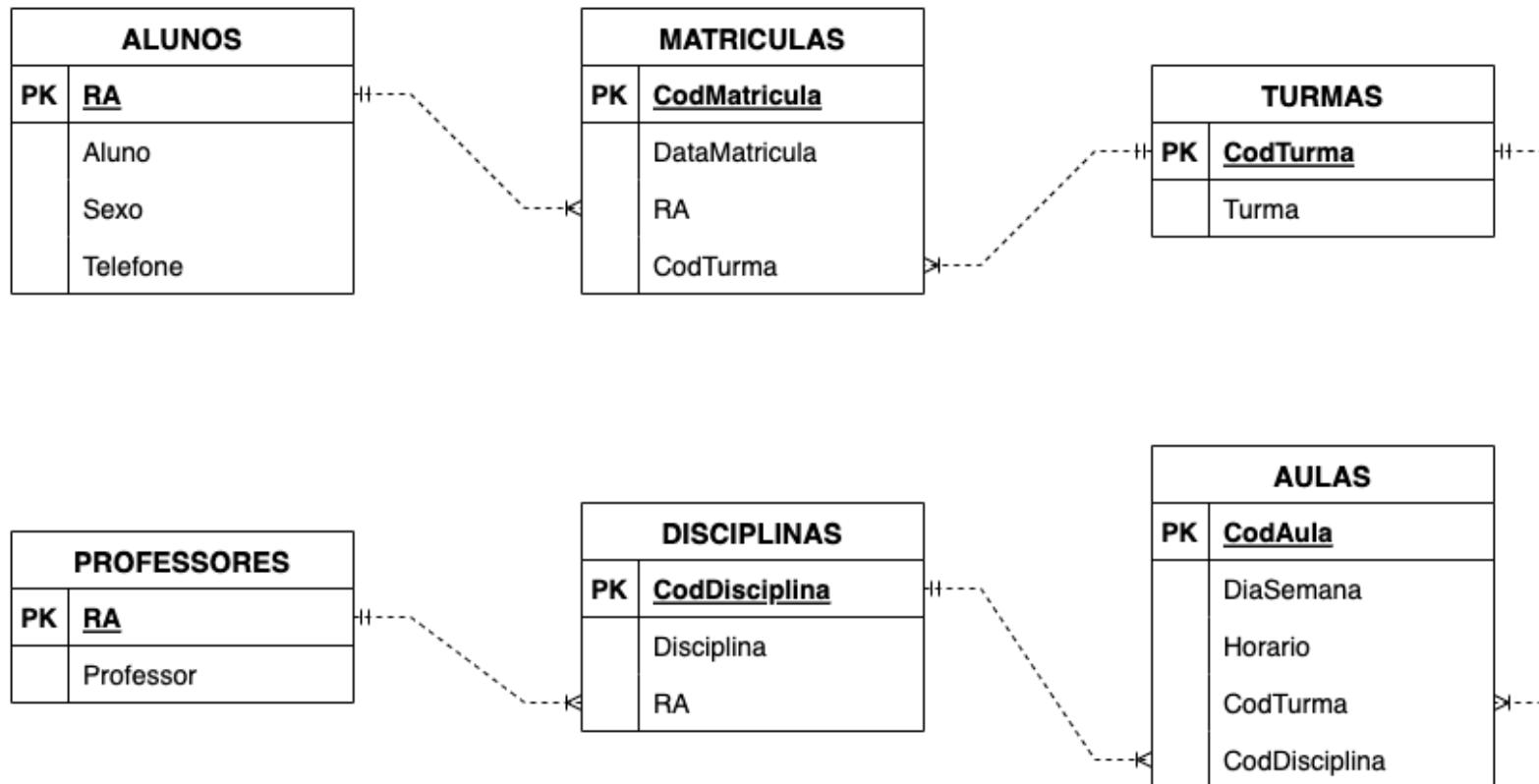
Modelos de Dados

- Exemplo desenvolvido no Microsoft Visio 2007



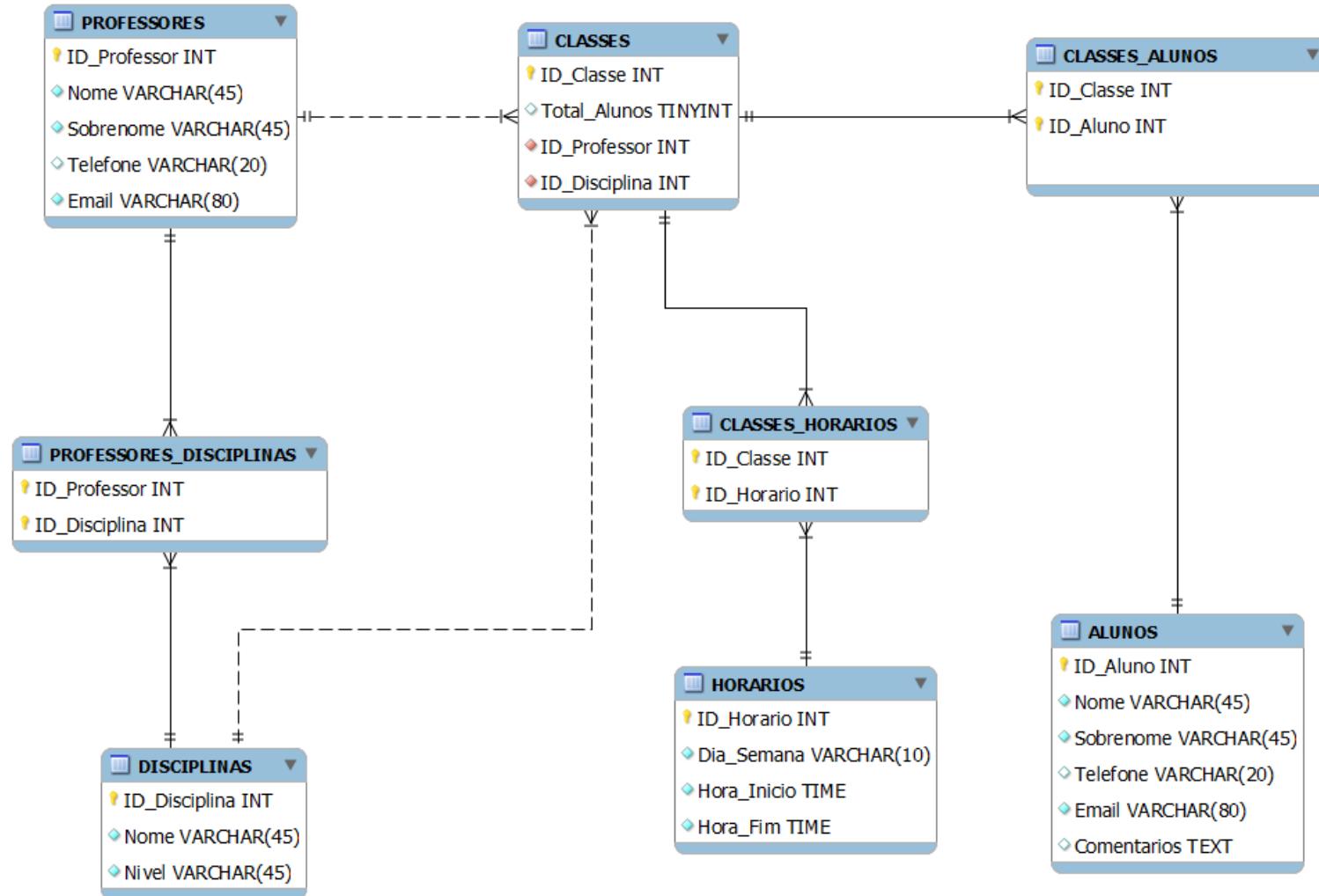
Modelos de Dados

- Exemplo desenvolvido no Drawio**



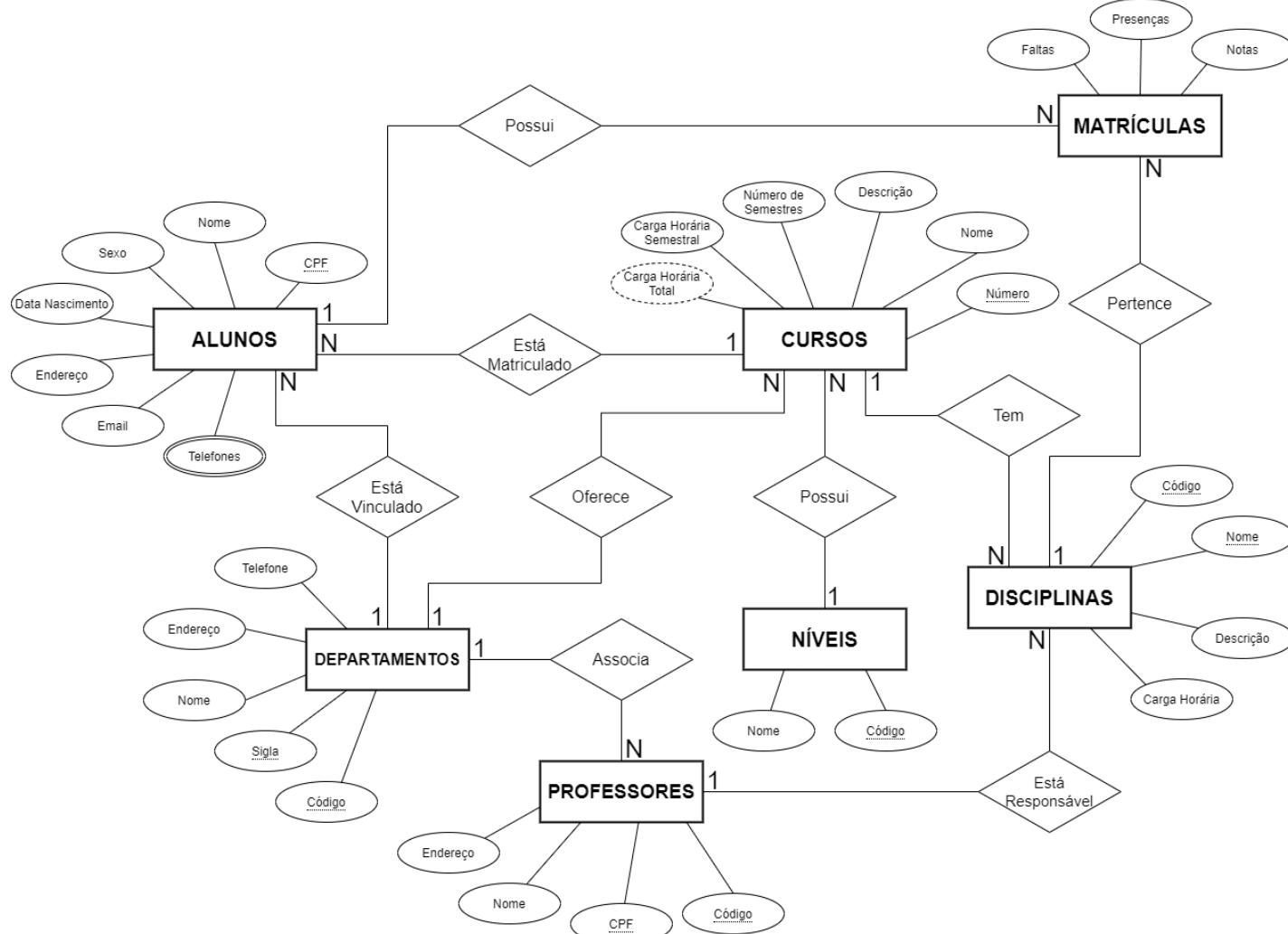
Modelos de Dados

- Exemplo desenvolvido no MySQL Workbench



Modelos de Dados

- Exemplo de modelo na notação de Peter Chen



- **Regras de negócio**

- O modelo de dados somente adquire significado quando representa de maneira adequada o que são chamadas de **regras de negócio**.
- **Regra de negócio:** descrição breve, precisa e sem ambiguidades, de uma política, um procedimento ou um princípio, utilizado em uma determinada organização.



■ Regras de negócio

- As regras de negócio devem ser fornecidas por escrito e atualizadas, de acordo com o ambiente onde o banco de dados será utilizado. Geralmente, esse ambiente é conhecido por **cenário**.
- Elas precisam ser de fácil compreensão, além de descrever as principais características dos dados, conforme vistos pela empresa.



- Exemplos de regras de negócio
 - Um **cliente** pode gerar muitas **faturas** e cada fatura é gerada por apenas um cliente.
 - Um **médico** pode atender vários **pacientes**. Cada paciente pode realizar vários exames.
 - Um **sócio** pode ter vários **dependentes** e, para cada um deles, devemos armazenar seu nome, data de nascimento, endereço, telefone e e-mail.



- **Exemplo de trecho de um cenário**

Uma floricultura deseja informatizar suas operações. Inicialmente, ela deseja manter um cadastro de todos os seus clientes, armazenando informações tais como código do cliente, RG, nome, telefone e endereço. A empresa também deseja manter um cadastro contendo informações sobre os produtos que vende, tais como código do produto, nome do produto, tipo (flor, vaso, planta, etc.), preço e quantidade em estoque. Um cliente pode realizar vários pedidos e cada pedido pode conter diversos itens de produtos. Cada item corresponde a um produto. Quando um cliente faz um pedido, o mesmo é armazenado, mantendo informação sobre o cliente que fez o pedido, a data do pedido, o valor total e os produtos comprados.



■ Importância das regras de negócio

- Ajudam a padronizar a visualização dos dados.
- Podem ser utilizadas como uma ferramenta de comunicação pela equipe de desenvolvimento.
- Permitem que o projetista compreenda a natureza, o papel e o escopo dos dados.
- Permitem que o projetista compreenda os processos comerciais realizados na empresa.
- Permitem que o projetista desenvolva regras e restrições adequadas para os relacionamentos, criando um modelo de dados preciso e coerente.



Modelos de Dados

- Principais fontes de regras de negócio



■ Evolução dos modelos de dados

Primeira Geração
1960 - 1970



Sistemas de Arquivos

- Dados armazenados em arquivos;
- Gerenciamento de registros, sem relacionamentos;
- Duplicidade no armazenamento;
- Segurança precária.

Segunda Geração
1970



Hierárquico e em Rede

- Baseados em árvores binárias;
- Primeiros sistemas de banco de dados;
- Busca eficiente;
- Complexidade de implementação.

Terceira Geração
1970 - atual



Modelo Relacional

- Proposto por Edgar Frank Codd;
- Simplicidade conceitual;
- Peter Chen cria o Modelo Entidade-Relacionamento;
- Suporte a modelagem relacional de dados;
- Linguagem SQL.

Quarta Geração
1980 - atual



Modelo Orientado a Objetos

- Suporte a dados complexos;
- Produtos relacionais estendidos, com suporte a objetos;
- Banco de dados na web tornam-se comuns.

Quinta Geração
1990 - atual



Novas Tecnologias

- Gerenciamento de dados não estruturados;
- Suporte a documentos em formato XML;
- Bancos de dados NoSQL.



■ Modelo hierárquico

- Considerado o primeiro tipo de banco de dados, foi desenvolvido na década de 1960, para gerenciar grandes quantidades de dados em projetos considerados mais complexos.
- Sua organização lógica é representada por uma estrutura de árvore invertida, composta por **níveis** ou **segmentos**.
- Nesse tipo de modelo, encontramos dois conceitos fundamentais: **registros** e **relacionamentos pai-filho**.

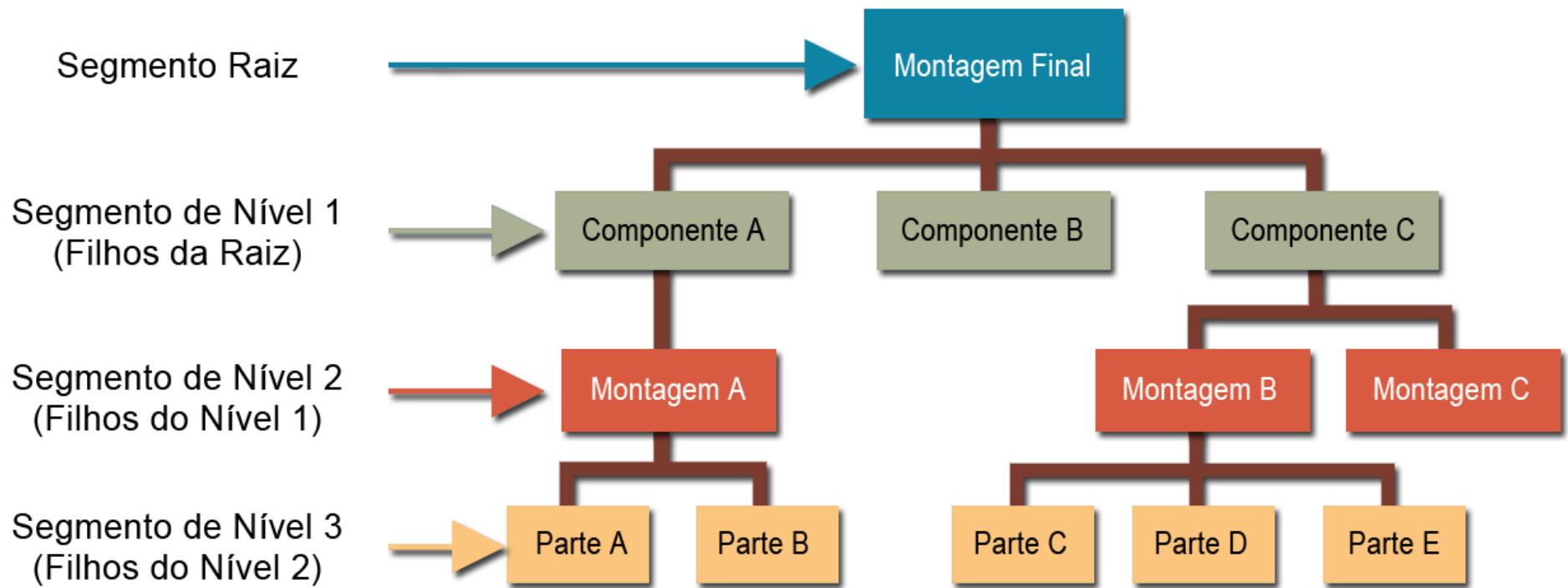


■ Modelo hierárquico

- O **registro** é uma coleção de valores que representam informações sobre uma dada entidade de um relacionamento.
- Os registros do mesmo tipo são denominados de **tipos de registros**. Eles são similares às **tabelas** que existem no modelo relacional.
- Nessa hierarquia, os registros que antecedem outros são denominados de **Pai**, sendo que seus sucessores são denominados de **Filhos**.



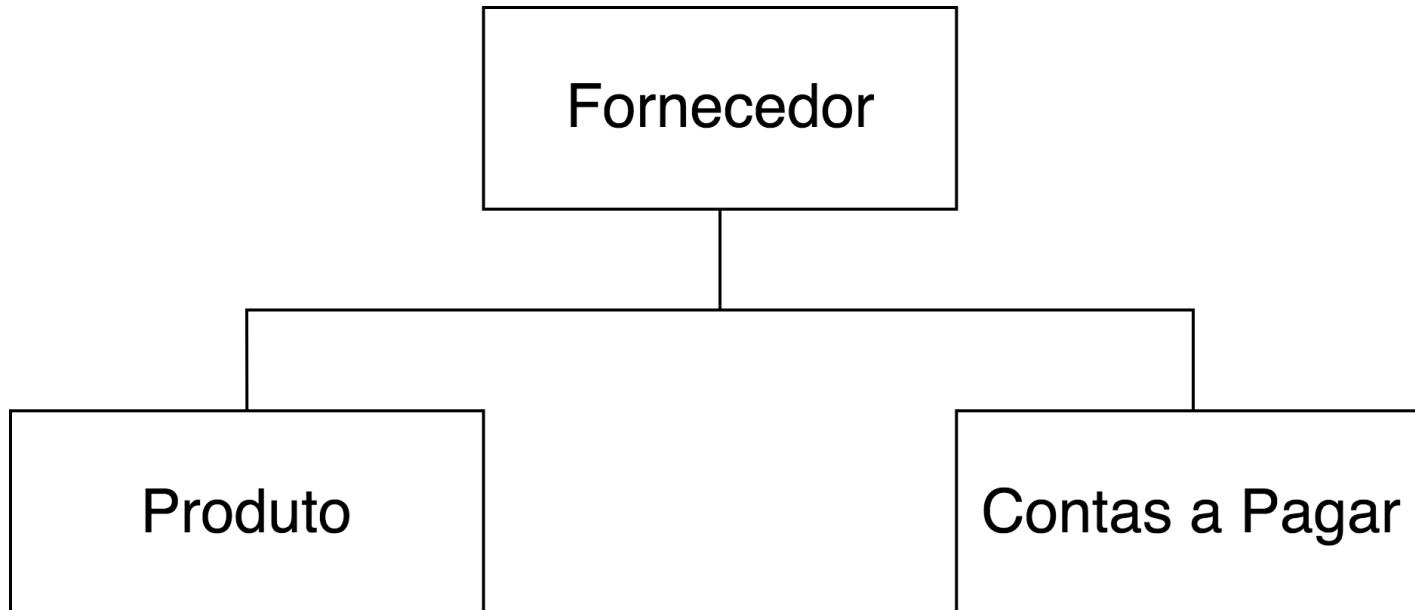
■ Modelo hierárquico



Exemplo da estrutura do modelo hierárquico, ilustrando a relação Pai-Filho.



- **Modelo hierárquico**

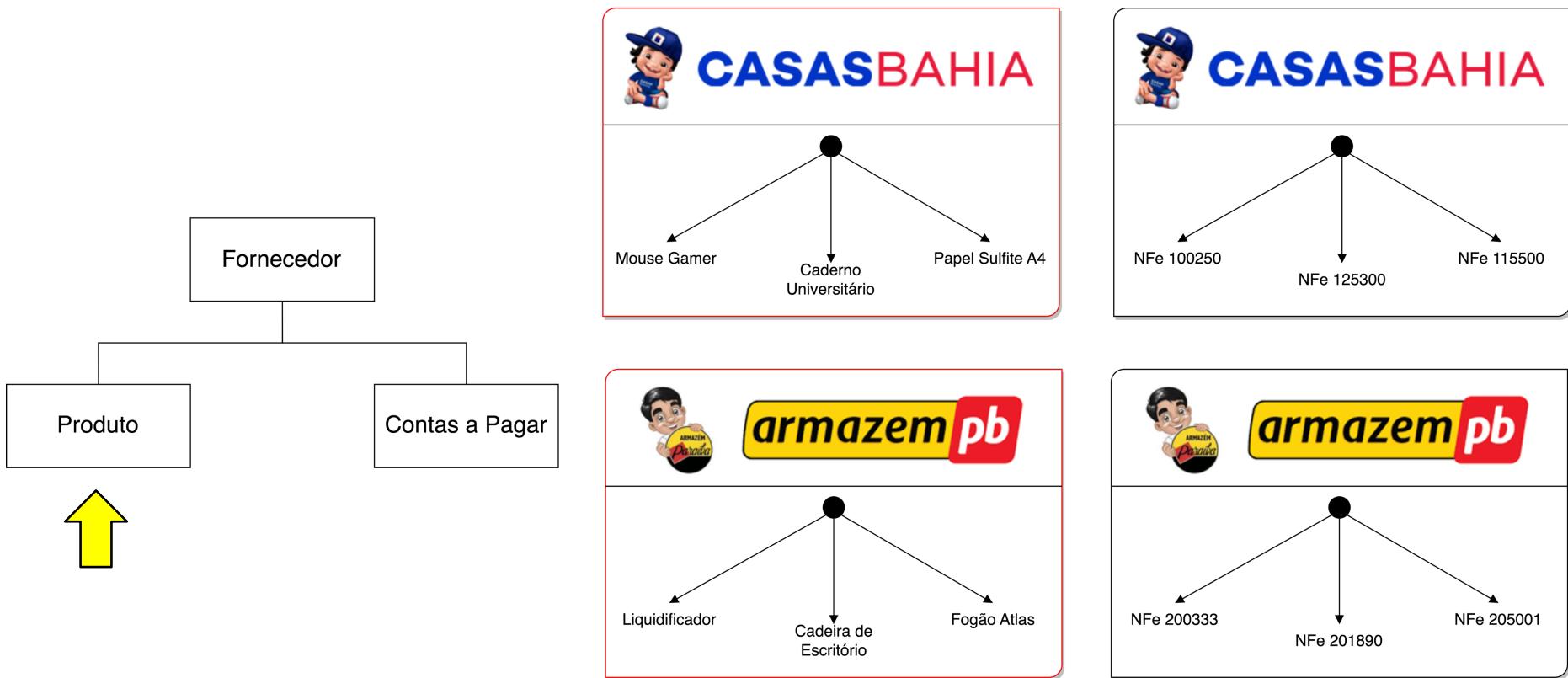


Exemplo da organização de registros em um banco de dados hierárquico.



Modelos de Dados

■ Modelo hierárquico

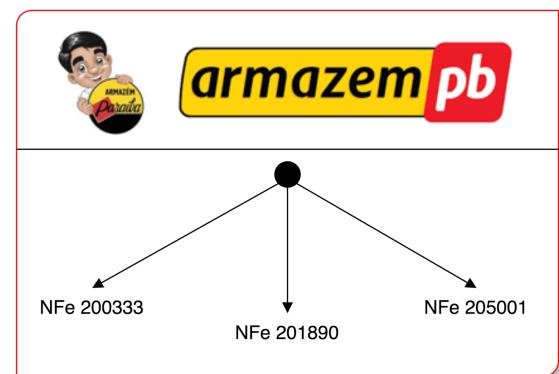
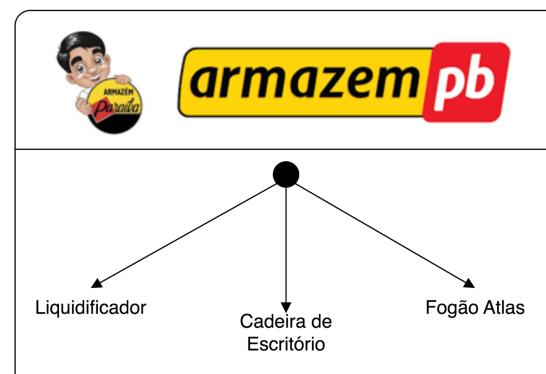
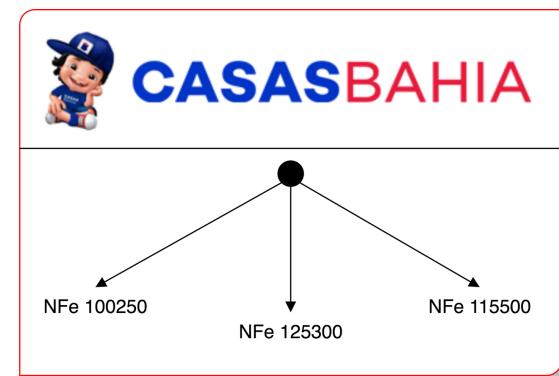
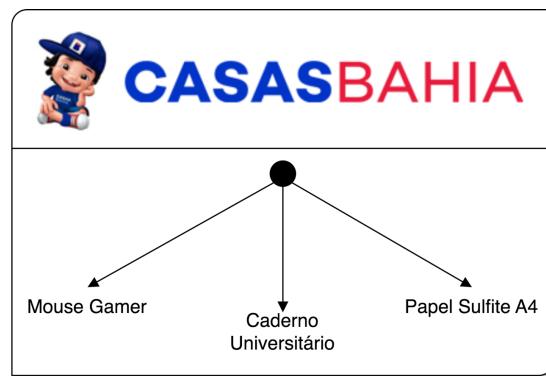
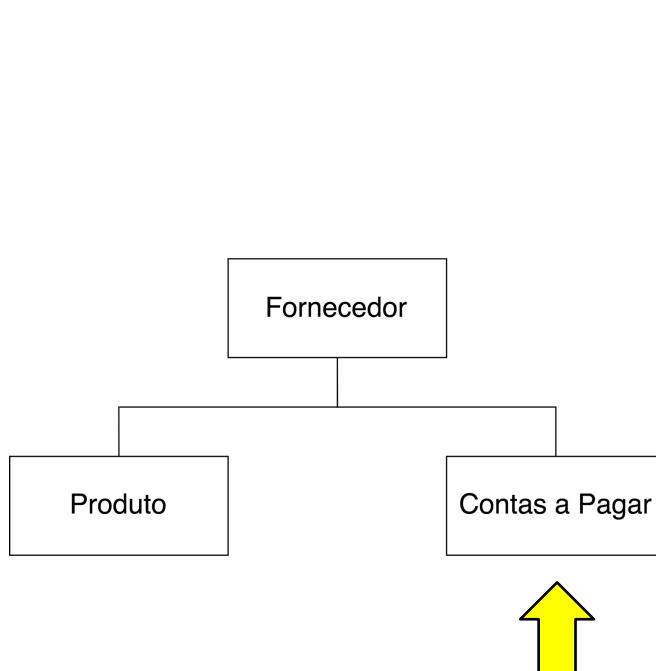


Exemplo da organização de registros em um banco de dados hierárquico.



Modelos de Dados

■ Modelo hierárquico



Exemplo da organização de registros em um banco de dados hierárquico.



■ Modelo em rede

- Esse tipo de modelo foi criado para representar relacionamentos de dados complexos com mais eficiência do que o modelo hierárquico.
- Além de melhorar o desempenho dos bancos de dados existentes na época, ele também definiu um **padrão de desenvolvimento**, facilitando a **portabilidade de aplicações**.
- Para ajudar a estabelecer esses padrões, o comitê da **Conferência sobre Linguagens de Sistemas de Dados** (CODASYL) criou o **Grupo de Tarefas sobre Bancos de Dados** (DBTG).



■ Modelo em rede

- O relatório final do DBTG apresentou as especificações para três componentes fundamentais de bancos de dados:
 1. **Esquema**, utilizado para representar a organização conceitual do banco de dados;
 2. **Subesquema**, utilizado para que os aplicativos pudessem acessar os dados armazenados em um banco de dados;
 3. **Linguagem de gerenciamento de dados (DML)**, utilizada para definir o ambiente em que os dados poderiam ser gerenciados.



■ Modelo em rede

- O modelo em rede é muito utilizado em computadores de grande porte. Convém lembrar que o CODASYL se dedica ao desenvolvimento de sistemas e linguagens de gerenciamento de bancos de dados, dentre as quais se destaca o **COBOL**.
- Diferente do modelo hierárquico, seus registros podem participar de vários relacionamentos. Além disso, o modelo também permite o acesso direto a um determinado registro, sem a obrigatoriedade de se passar pelo segmento raiz.



■ Exemplo de código em COBOL

```
1 IDENTIFICATION DIVISION.          22      05 WS-ID      PIC 9(4).
2 PROGRAM-ID. SQLEXEMPLO.          23      05 WS-NOME    PIC X(25).
3                                         24      05 WS-ENDERECO X(50).
4 DATA DIVISION.                  25      EXEC SQL END DECLARE SECTION
5                                         26      END-EXEC.
6   WORKING-STORAGE SECTION.        27
7                                         28 PROCEDURE DIVISION.
8   * Declara o SQLCA              29      * Consulta SQL
9   EXEC SQL                      30      EXEC SQL
10  INCLUDE SQLCA                 31      SELECT ID, NOME, ENDERECO
11  END-EXEC.                     32      INTO :WS-ID, :WS-NOME, WS-ENDERECO FROM CLIENTES
12                                         33      WHERE ID = 10
13   * Inclui a tabela            34      END-EXEC.
14   EXEC SQL                      35
15   INCLUDE CLIENTES             36      * Exibe o resultado
16  END-EXEC.                     37      IF SQLCODE=0
17                                         38      DISPLAY WS-CLIENTES-RECORD
18   * Estrutura para armazenar os dados 39      ELSE DISPLAY 'Erro na execucao do programa!'
19   EXEC SQL BEGIN DECLARE SECTION 40      END-IF.
20  END-EXEC.                     41
21   01 WS-CLIENTES-REC.          42      STOP RUN.
```

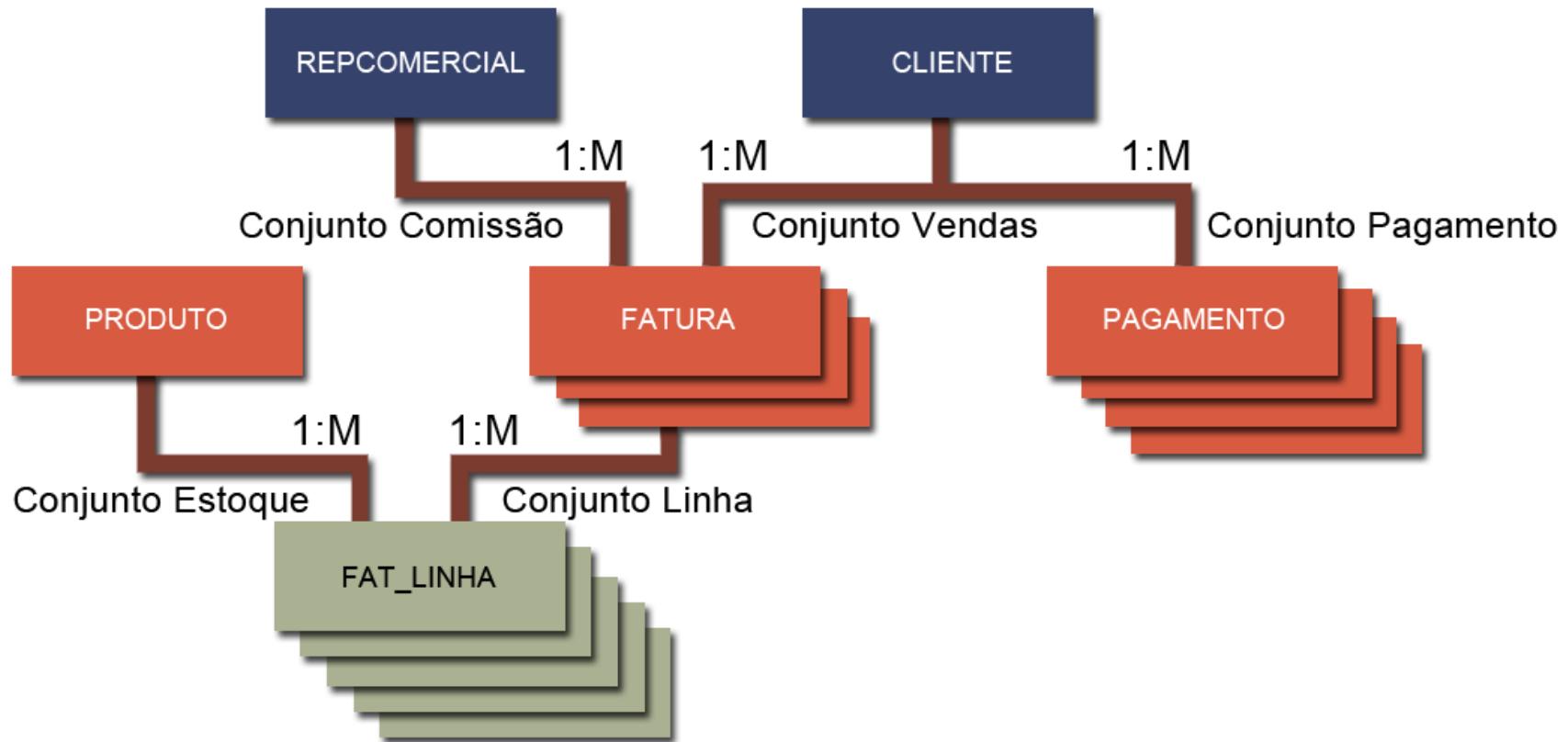


■ Modelo em rede

- As duas estruturas fundamentais de um modelo em rede são os **registros** e os **conjuntos**.
- Os **registros** contêm dados relacionados e são agrupados em tipos de registros que armazenam os mesmos tipos de informações.
- Os **conjuntos** são a forma de representação dos relacionamentos entre os diversos tipos de registros, na forma de um-para-muitos, ou 1:N.
- Um **tipo de conjunto** possui em sua definição três componentes: nome do tipo de conjunto, tipo de registro proprietário e tipo de registro membro.



■ Modelo em rede



Exemplo da estrutura do modelo em rede, ilustrando registros e conjuntos.



■ Modelo relacional

- A maioria dos sistemas de gerenciamento de bancos de dados se enquadra no tipo relacional.
- O **modelo relacional** se caracteriza por organizar os dados em relações, também conhecidas como **tabelas**, que são formadas por linhas e colunas.
- Essas tabelas são similares aos conjuntos de elementos, uma vez que relacionam as informações referentes a um mesmo assunto, de maneira organizada.



■ Modelo relacional

- O modelo relacional foi apresentado em 1970, pelo matemático **Edgar Frank Codd**, pesquisador da IBM.
- Baseado no artigo *A Relational Model of Data for Large Shared Databanks*, Raymond Boyce e Donald Chamberlin criaram a primeira versão da linguagem SQL.
- O fundamento do modelo é um conceito matemático conhecido como **relação**. Ted Codd formulou os princípios do modelo relacional baseando-se na teoria dos conjuntos e na **Álgebra Relacional**.



Modelos de Dados

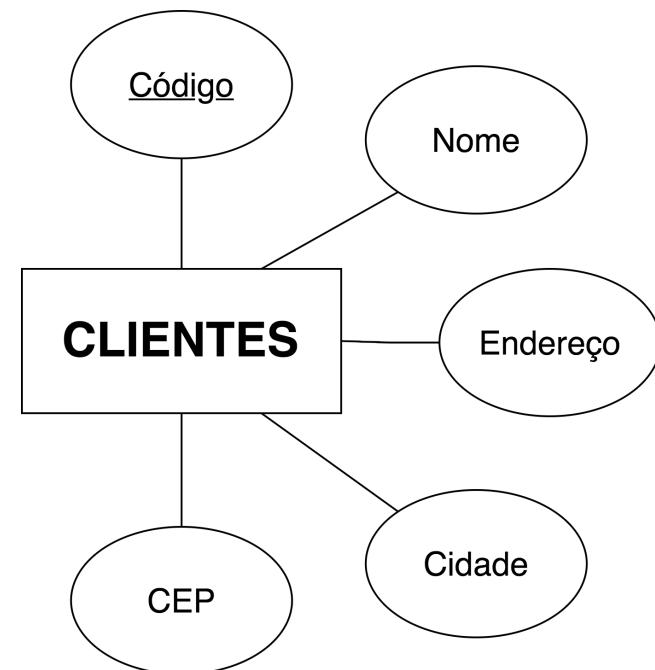
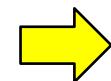
- Exemplo de tabela no modelo relacional

Tabela



Column Name	Data Type	Allow Nulls
CODIGO	int	<input type="checkbox"/>
NOME	varchar(30)	<input type="checkbox"/>
ENDERECO	varchar(30)	<input type="checkbox"/>
CIDADE	varchar(20)	<input type="checkbox"/>
CEP	char(9)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Entidade



Modelos de Dados

- Exemplo de tabela no modelo relacional

Table - dbo.Cliente		Table - dbo.Cliente		Summary	
	CODIGO	NOME	ENDERECO	CIDADE	CEP
▶	1	Antônio Carlos	Rua das Madeiras	Campos	11000-253
	2	Pedro Renato	Av. das Nações	São Paulo	10560-666
	3	Maria Pereira	Rua João XXIII	Sertãozinho	23562-000
	4	Caroline Gomes	Rua Teresa Muniz	Campinas	16566-666
	5	Jorge Luís	Av. Maria Bonita	Batatais	14145-541
	6	Eddie	Rua Sessenta e Seis	Birigui	12345-666
*	NULL	NULL	NULL	NULL	NULL

Instâncias
ou Registros

Atributos ou Propriedades

Exemplo de clientes cadastrados em uma tabela relacional.



Modelos de Dados

■ Modelo relacional

Nome da tabela: AGENT (Corretor)

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Ligaçāo por meio de AGENT_CODE

Nome da tabela: CUSTOMER (Cliente)

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2008	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2008	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2009	502
10013	Olowksi	Paul	F	615	894-2180	S1	300.00	14-Oct-2008	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2008	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2008	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2009	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2008	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2008	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2009	503

Exemplo de tabelas relacionais.



Modelos de Dados

■ Modelo relacional

Nome da tabela: AGENT (Corretor)

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Ligaçāo por meio de AGENT_CODE

Nome da tabela: CUSTOMER (Cliente)

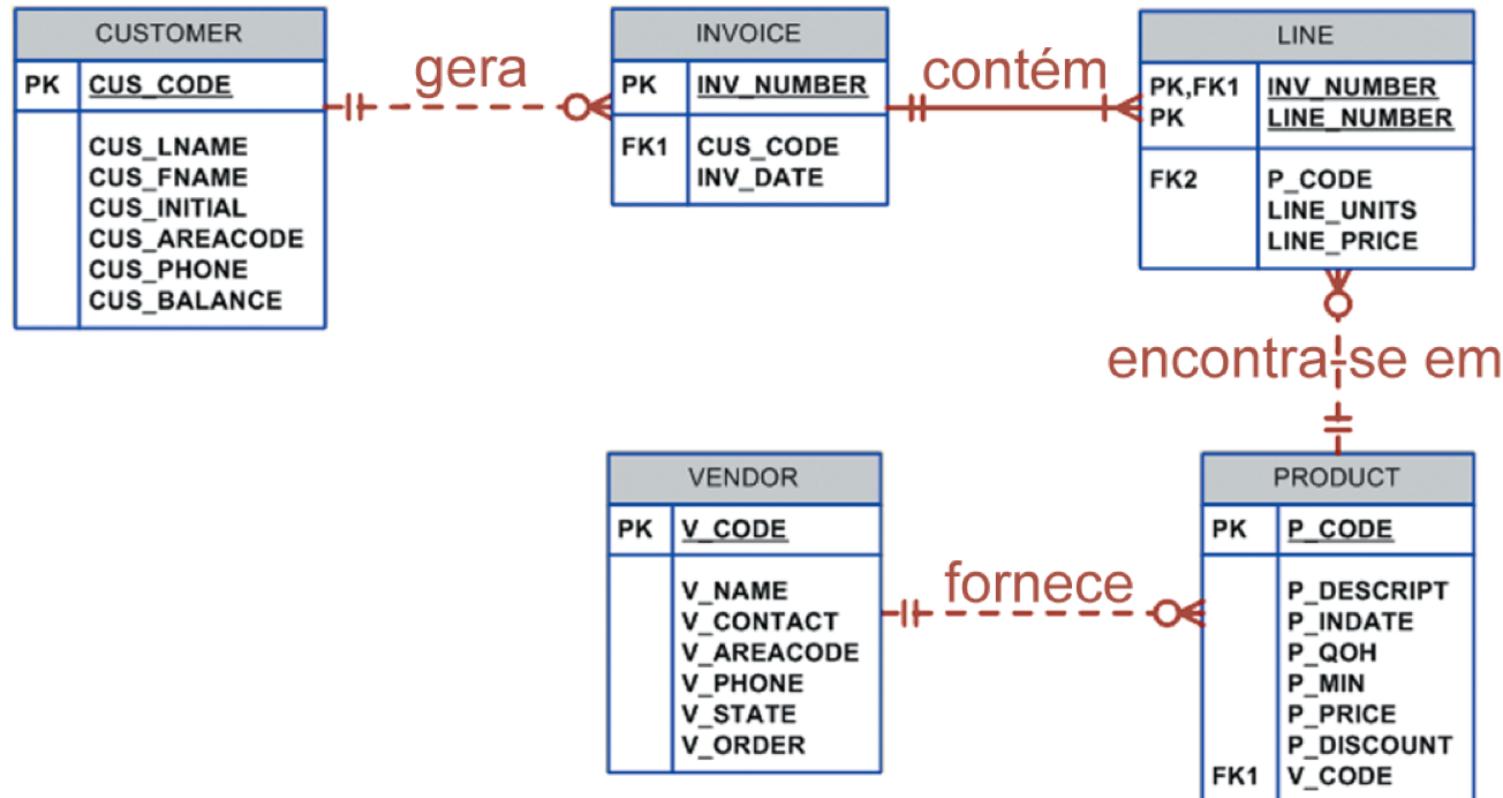
CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2008	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2008	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2009	502
10013	Olowksi	Paul	F	615	894-2180	S1	300.00	14-Oct-2008	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2008	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2008	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2009	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2008	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2008	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2009	503

Exemplo de tabelas relacionais, com a indicação do corretor e seus clientes.



Modelos de Dados

- Exemplo de modelo relacional



■ Modelo orientado a objetos

- O **modelo de dados orientado a objetos** surgiu em meados da década de 1980, em virtude da necessidade de representação e armazenamento de dados mais complexos, como sistemas de geoprocessamento, por exemplo.
- Nesse modelo, tanto os dados como os seus relacionamentos estão contidos em uma estrutura conhecida como **objeto**.
- Isso significa que um registro é mais parecido com uma **classe**, definida em linguagens de programação tais como C++, C#, Java, entre outras.



- **Modelo orientado a objetos**
 - O modelo de dados orientado a objetos é baseado nas seguintes características:
 - Cada **objeto** é uma abstração de uma **entidade** real.
 - Os **atributos** são utilizados para descrever as **propriedades** desse objeto.
 - Os objetos que compartilham características semelhantes são agrupados em **classes**.

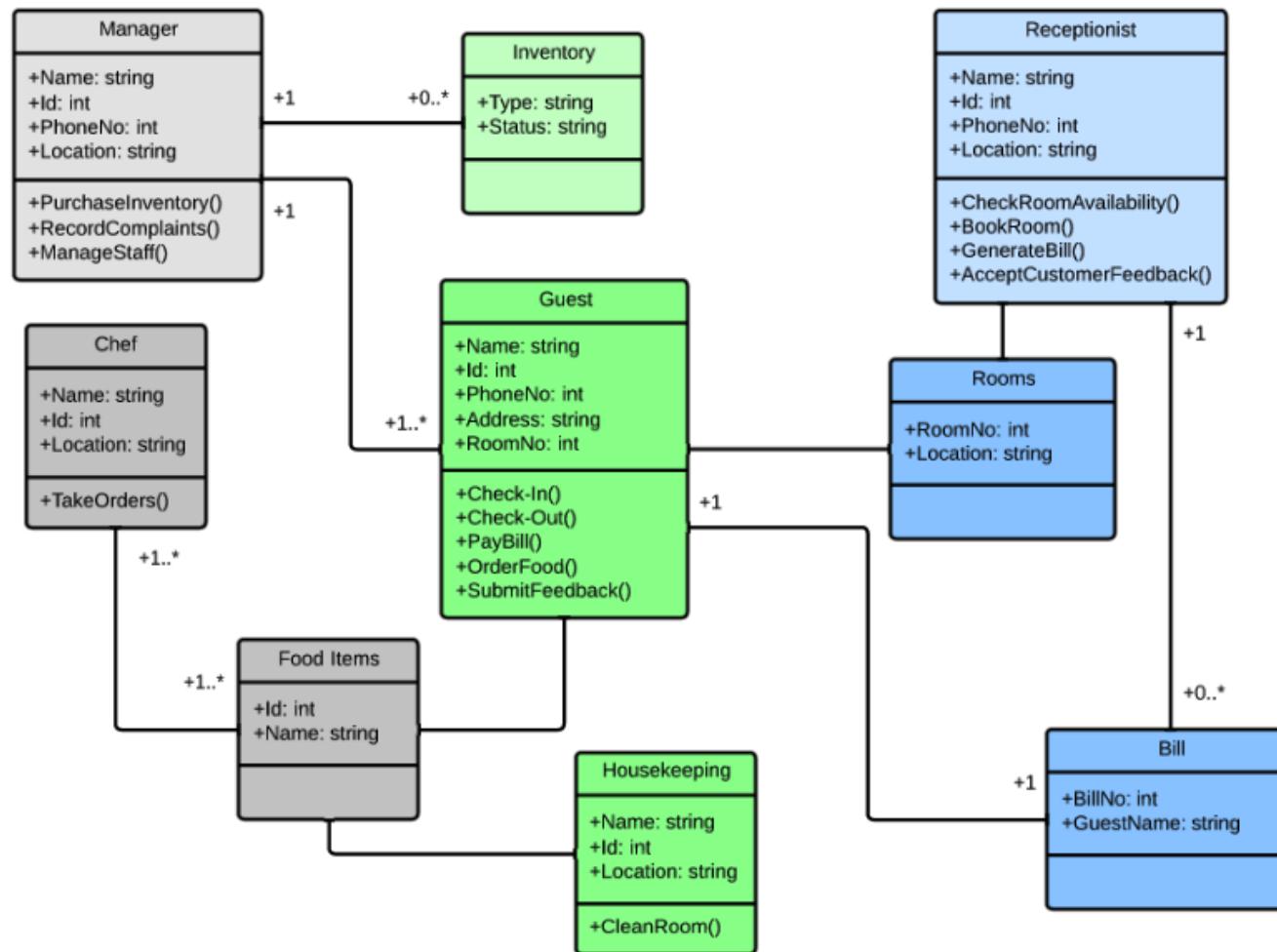


- **Modelo orientado a objetos**
 - Os modelos de dados orientados a objetos normalmente são representados por meio dos diagramas de classe da **UML**, ou **Linguagem de Modelagem Unificada** (*Unified Modeling Language*).
 - Esses diagramas são utilizados para poder realizar a representação dos objetos, suas propriedades e relacionamentos.



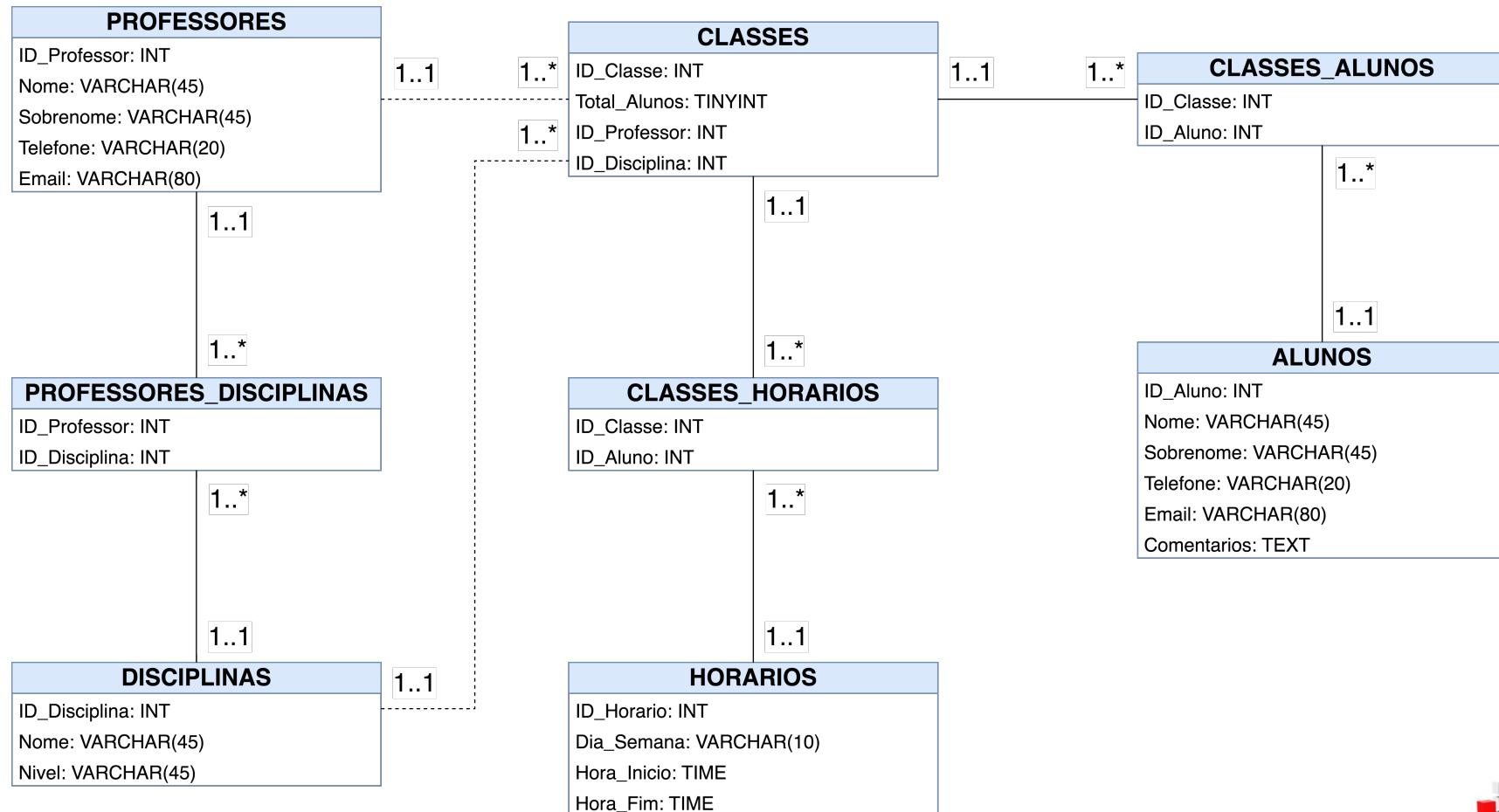
Modelos de Dados

- Exemplo de diagrama de classes em UML



Modelos de Dados

- Exemplo de modelo de dados representado em UML

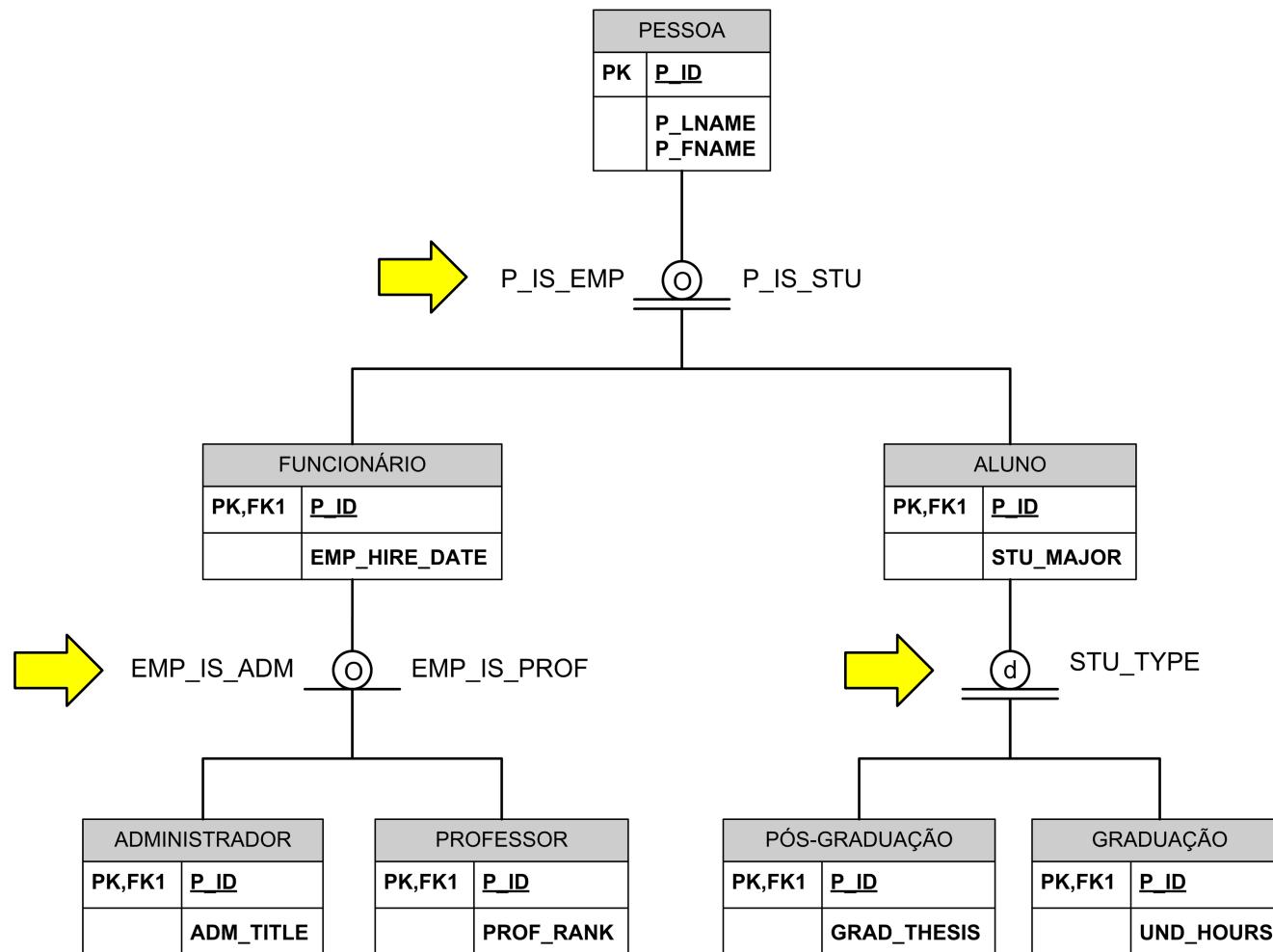


■ Modelo relacional estendido

- Esse tipo de modelo surgiu devido à complexidade crescente das aplicações, com o objetivo de fornecer maior conteúdo semântico para a modelagem de dados.
- Ele expande os recursos presentes no modelo relacional, permitindo que se utilize também parte da abordagem presente no modelo orientado a objetos.
- Além dos conceitos contidos no modelo relacional, o **modelo relacional estendido** também permite a inclusão de **subtipos e supertipos, especialização e generalização, representação do tipo de categoria, herança de atributos e herança de relacionamentos.**



- Exemplo de modelo relacional estendido



■ **Modelo de banco de dados e a Internet**

- A utilização da Internet como principal ferramenta comercial alterou drasticamente o papel e o escopo do mercado de bancos de dados.
- Isso ocorreu por causa da necessidade cada vez maior de se realizar o gerenciamento de dados não estruturados.
- Dessa maneira, os banco de dados atuais oferecem suporte às tecnologias da Internet, permitindo-se trabalhar com dados semiestruturados (XML, JSON, etc.)



Modelos de Dados

- Exemplo de modelo orientado a documentos**

Documento: Usuários

```
{  
  _id: user001,  
  nome: "Paulo Giovani"  
}
```

Documento: Contatos

```
{  
  _id: contato001,  
  user_id: user001,  
  telefone: "12-123456",  
  email: "paulogiovani@ifsp.edu.br"  
}
```

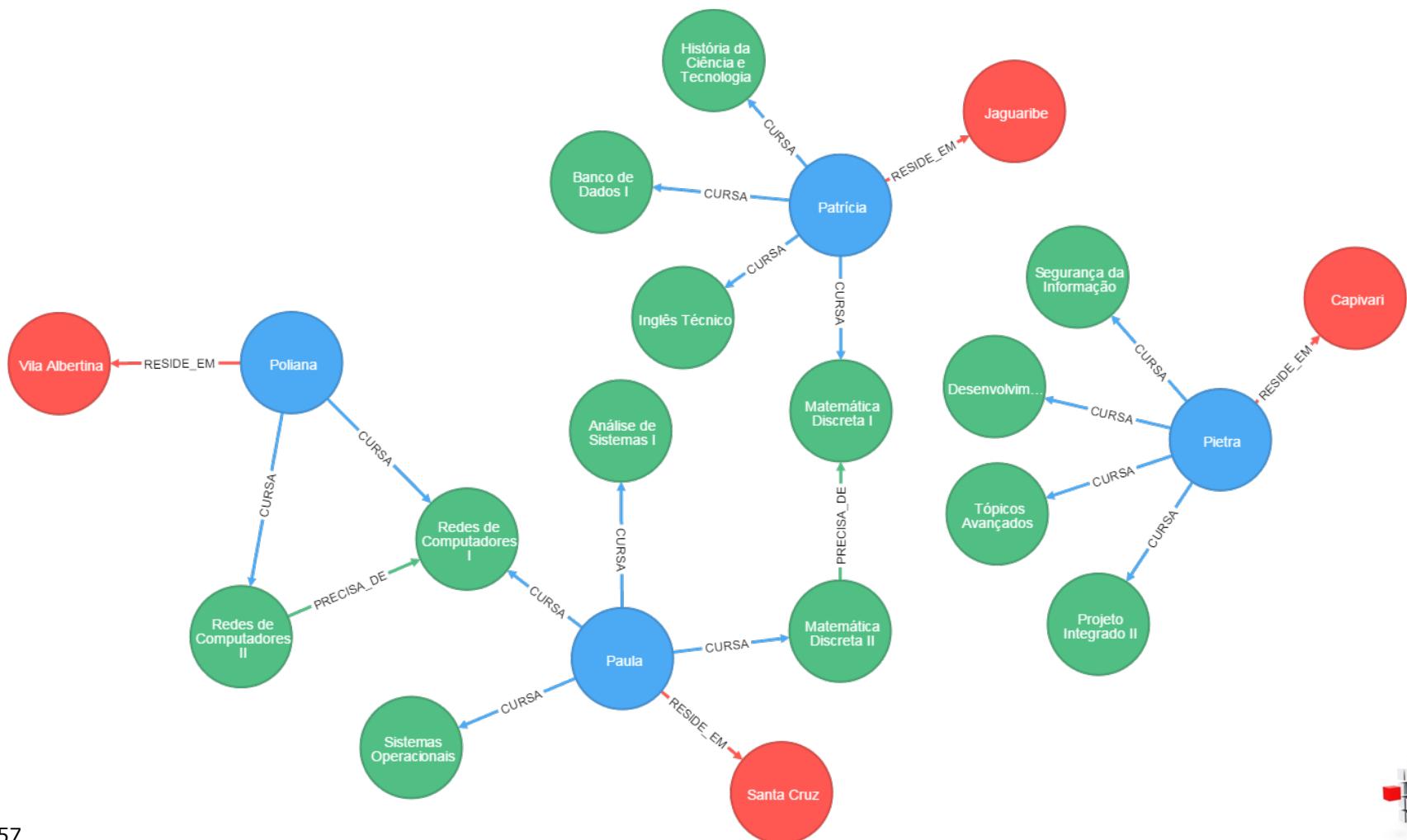
Documento: Equipes

```
{  
  _id: equipe001,  
  user_id: user001,  
  nivel: 5,  
  grupo: "Programadores"  
}
```



Modelos de Dados

- Exemplo de modelo orientado a grafos

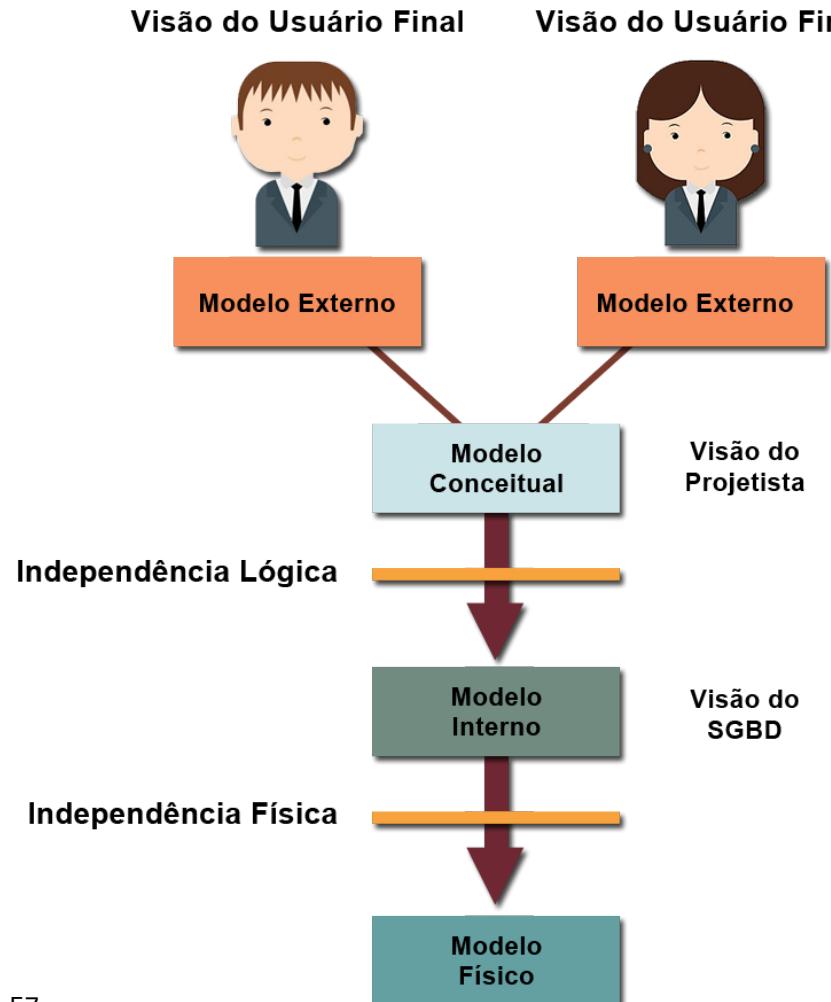


■ Grau de abstração do modelo de dados

- **Modelo externo:** representa a perspectiva dos usuários finais do ambiente de banco de dados.
- **Modelo conceitual:** representa uma visão global do banco de dados, conforme visto pela empresa.
- **Modelo lógico:** mapeia o modelo conceitual para as estruturas do modelo relacional, definindo os tipos de dados que serão necessários para cada atributo.
- **Modelo físico:** descreve o modo como o banco de dados deverá ser criado em um SGBD específico.



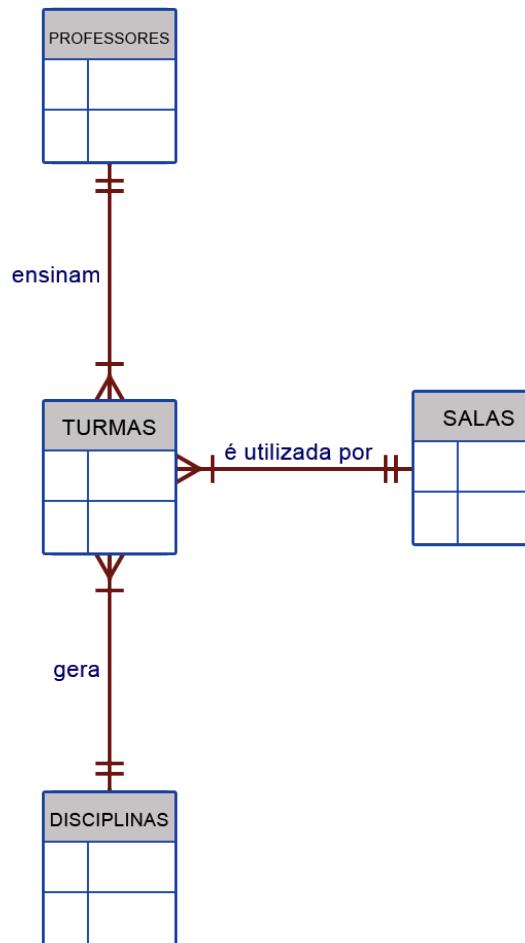
- Grau de abstração do modelo de dados**



Grau de Abstração	Características
Alto	ER Orientado a Objetos
Médio	Relacional
Baixo	Em Rede Hierárquico



■ Exemplo de modelo físico em SQL



```

CREATE TABLE PROFESSORES
(
    PROF_ID          NUMBER      PRIMARY KEY,
    PROF_SOBRENOME  CHAR(15),
    PROF_INICIAL    CHAR(1),
    PROF_NOME       CHAR(15),
    ...
);

CREATE TABLE TURMAS
(
    TURMA_ID        NUMBER      PRIMARY KEY,
    DIS_ID          CHAR(8)    FOREIGN KEY REFERENCES DISCIPLINAS,
    PROF_ID         NUMBER    FOREIGN KEY REFERENCES PROFESSORES,
    SALA_ID         CHAR(8)    REFERENCES SALAS,
    ...
);

CREATE TABLE SALAS
(
    SALA_ID         CHAR(8)      PRIMARY KEY,
    SALA_TIPO       CHAR(3),
    ...
);

CREATE TABLE DISCIPLINAS
(
    DIS_ID          CHAR(8)      PRIMARY KEY,
    DIS_NOME        CHAR(25),
    DIS_CREDITO     NUMBER,
    ...
);
  
```



Modelos de Dados

- Na próxima aula veremos...
 - Modelo Relacional.

