Aonde você quer chegar? Vai com a





Disciplina: Banco de Dados Prof. Maurício P. de Freitas MSc.

Aula 05 – 15/08/2024 MER (Modelo Entidade Relacionamento) e Normalização





Livros Minha Biblioteca



Heuser, C. A. (2011). Projeto de banco de dados - UFRGS. V.4, 6th Edition



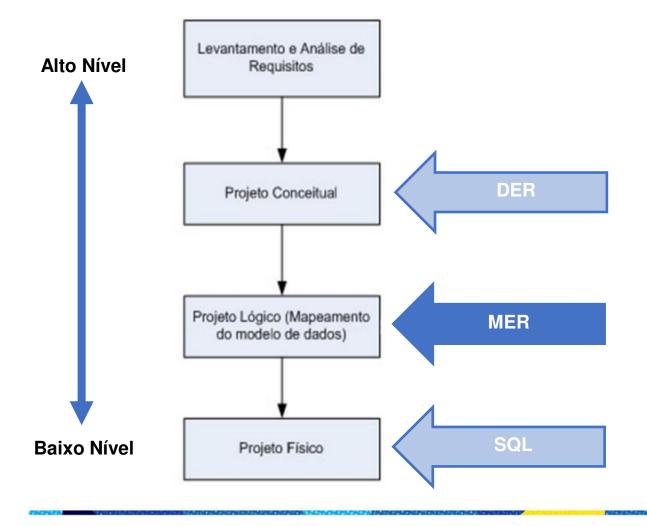
Banco de Dados I

Aula 05 Conteúdo:

- Modelo Entidade Relacionamento
- Normalização







Projeto de Banco de Dados

Esquema Lógico 01

```
Pessoa(<u>id</u>, nome, sobrenome, dtnasc, email)
```

Pessoa_endereço(<u>id</u>, id_pessoa, cep, rua, numero, complemento, cidade, estado) id_pessoa referencia Pessoa(id)

Pessoa_telefone(<u>id</u>, id_pessoa, numero, tipo) **id_pessoa referencia Pessoa(id)**

Tutor(<u>id_pessoa</u>, cpf, profissão, dtpgto)

id_pessoa referencia Pessoa(id)

Veterinario(<u>id_pessoa</u>, crmv, especialidade, banco, agencia, conta, tipo) id_pessoa referencia Pessoa(id)

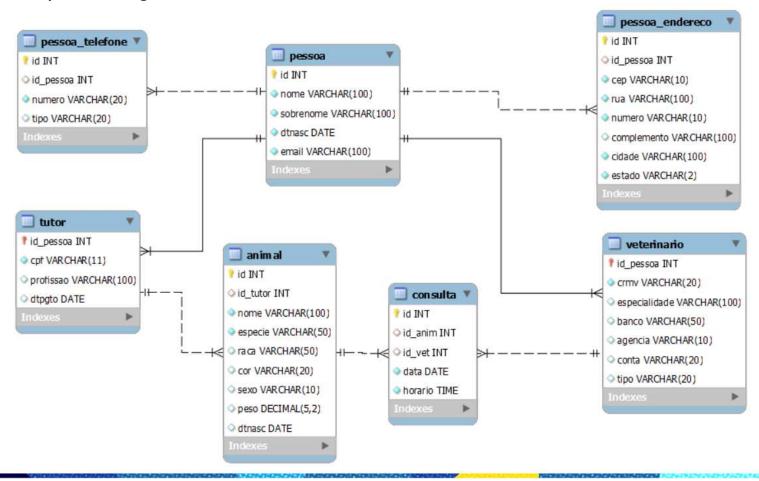
Animal(<u>id</u>, id_tutor, nome, espécie, raca, cor, sexo, peso, dtnasc) id_tutor referencia Tutor(id_pessoa)

Consulta(<u>id</u>, id_anim, id_vet, data, horario)

id_anim referencia Animal(id)

id_vet referencia Veterionario(id_pessoa)

MER → Esquema lógico 01



Esquema lógico 02

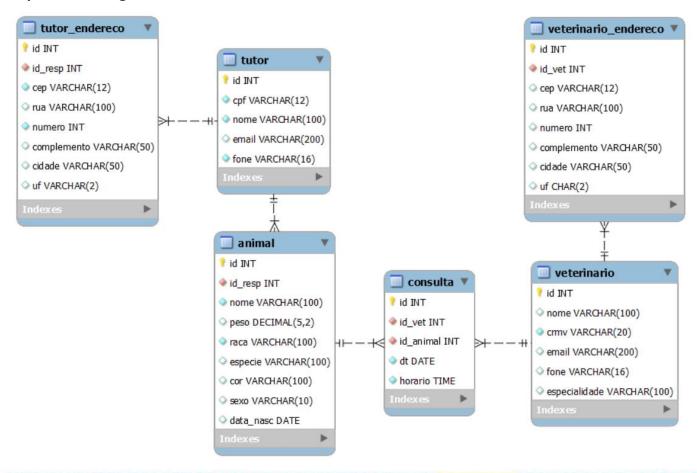
Tutor(id, cpf, nome, email, fone)

Tutor_endereço(id, id_resp, cep, rua, numero, complemento, cidade, uf) id_resp referencia Responsavel(id)

Animal(id, id_resp, peso, raca, especie, cor, sexo, data_nasc) id_resp referencia Responsável(id)

Veterinário(id, nome, crmv, email, fone, especialidade)

MER → Esquema lógico 02



Conceito

 É uma atividade na qual o objetivo não é apenas obter uma estrutura de dados apropriada para armazenamento, mas desenvolve-lo de maneira que garanta um bom desempenho.



Elementos

- Dicionário de Dados.
- Modelo Entidade-Relacionamento.
- A partir do projeto lógico normalizado, damos início ao dicionário de dados. Em seguida, criamos o MER. Por fim, criamos o Banco de Dados utilizando a linguagem SQL.



Tipos de Dados - Numéricos

Tipo		Range			
	Armazenamento	Signed	Unsigned		
TINYINT	1 byte	-128 a 127	0 a 255		
SMALLINT	2 bytes	-32768 to 32767	0 to 65535		
MEDIUMINT	3 bytes	-8388608 to 8388607	0 to 16777215		
INT	4 bytes	-2147483648 to 2147483647	0 to 4294967295		
BIGINT	8 bytes	-9223372036854775808 to 9223372036854775807	0 to 18446744073709551615		
FLOAT (p)	4 bytes se 0 <= p <= 24, 8 bytes se 25 <= p <= 53	-			
FLOAT	4 bytes	-3.402823466E+38 a 3.402823466E+38	0 a 3.402823466E+38		
DOUBLE	8 bytes	-1.7976931348623157E+308 a 1.7976931348623157E+308	0 a 1.7976931348623157E+308		
DECIMAL (M, D)	varia de acordo com o valor de M	varia de acordo com o valor de M			
BIT (M)	aproximadamente (M+7)/8 bytes	1 a 64			



Tipos de Dados – Texto

- CHAR[(M)]: uma string de tamanho fixo, representado pelo M que pode assumir um valor entre 0 e 255. M pode ser informado para definir o tamanho da coluna e caso omitido, recebe o valor 1. Caso todas as posições da string não sejam utilizadas, ela será completada com espaços até atingir o tamanho total.
- VARCHAR(M): uma string de tamanho variável. O valor de M pode variar de 0 a 65535.



Tipos de Dados – Texto

- TEXT [(M)]: suas características são como as do BLOB. O valor máximo de M é 65535. Seus subtipos são:
 - □ **TINYTEXT:** suporta até 255 caracteres;
 - MEDIUMTEXT: suporta até 16.777.326 caracteres;
 - □ **LONGTEXT:** suporta até 4.294.967.295 caracteres.



Tipos de Dados – Data/Hora

- DATE: se trata de uma data simples, no formato 'AAAA-MM-DD'.
 O tipo DATE permite valores informados como string ou números.
 Aceita dados entre 1000–01–01 e 9999–12–31.
- DATETIME: uma combinação de data e hora, no formato 'AAAA-MM-DD HH:MM:SS[.fração]' e permite que os valores sejam informados como string ou números. Suporta valores entre 1000–01–01 00:00:00.000000 e 9999–12–31 23:59:59.999999.



Tipos de Dados – Data/Hora

- □ **TIMESTAMP:** salva uma data e hora utilizando definindo o fuso horário (1970–01–01 00:00:00 UTC).
- TIME: armazena uma faixa de tempo, no formato 'HH:MM:SS[.fração]' e seu valor pode estar entre 838:59:59.000000 e 838:59:59.000000.
- YEAR: ano representado por 4 dígitos, no formato 'AAAA' e pode ter valores entre 1901 e 2155, além do valor 0000.



Tipos de Dados

- BLOB [(M)]: seu nome significa objeto binário grande. Caso um valor seja informado para M, o MySQL vai criar o menor subtipo de BLOB possível para armazenar aquela quantidade de dado, que são:
 - □ **TINYBLOB:** tamanho máximo de 255 bytes;
 - □ **MEDIUMBLOB:** tamanho máximo de 16.777.326 bytes (16MB);
 - □ LONGBLOB: tamanho máximo de 4.294.967.295 bytes (4GB).



Dicionário de Dados

- Representa os tipos de dados de cada coluna da tabela.
 - Cada tabela do projeto lógico, possuirá um dicionário de dados.

USUÁRIO	TIPO	TAMANHO	DESC	OBS
id	Int	6	Código do Funcionário	Chave Primária
nome	Varchar	50	Nome do Funcionário	
cpf	Int	12	CPF do Funcionário	
rg	Int	12	RG do Funcionário	
salário	Float	20	Salário Bruto do Funcionário	
departamento_id(FK)	Int	30	Departamento do Funcionário	Chave



MER

- Modelo Entidade-Relacionamento (MER).
- Representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação.
- Utilizado na Engenharia de Software para representar:
 - Tabelas;
 - Colunas com os tipos de dados;
 - Relacionamentos.



MER – Relacionamentos

one-to-one (1-para-1) non-identifying one-to-many (1-para-muitos) non-identifying one-to-one (1-para-1) identifying one-to-many (1-para-muitos) identifying many-to-many (muitos-para-muitos) identifying

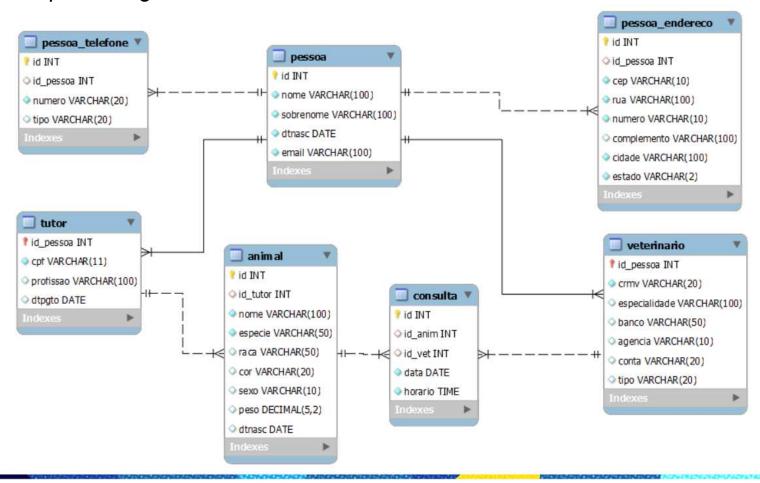


MER – Relacionamentos

- identifying: linha sólida entre as tabelas.
 - □ É aquele em que a tabela filho não pode ser identificada exclusivamente sem seu pai.
 - Geralmente ocorre quando uma tabela associativa é criada para resolver um relacionamento muitos-para-muitos.
 - □ A chave primária geralmente é uma chave composta pelas chaves primárias das duas tabelas originais.
- non-identifying: linha tracejada entre as tabelas.
 - Cria apenas uma coluna que representa a chave estrangeira.



MER → Esquema lógico 01



Normalização





Esquema Lógico 01

Pessoa(<u>id</u>, nome, sobrenome, dtnasc, email)

Pessoa_endereço(<u>id</u>, id_pessoa, cep, rua, numero, complemento, cidade, estado) id_pessoa referencia Pessoa(id)

Pessoa_telefone(<u>id</u>, id_pessoa, numero, tipo)

id_pessoa referencia Pessoa(id)

Tutor(<u>id_pessoa</u>, cpf, profissão, dtpgto)

id_pessoa referencia Pessoa(id)

Veterinario(<u>id_pessoa</u>, crmv, especialidade, banco, agencia, conta, tipo) id_pessoa referencia Pessoa(id)

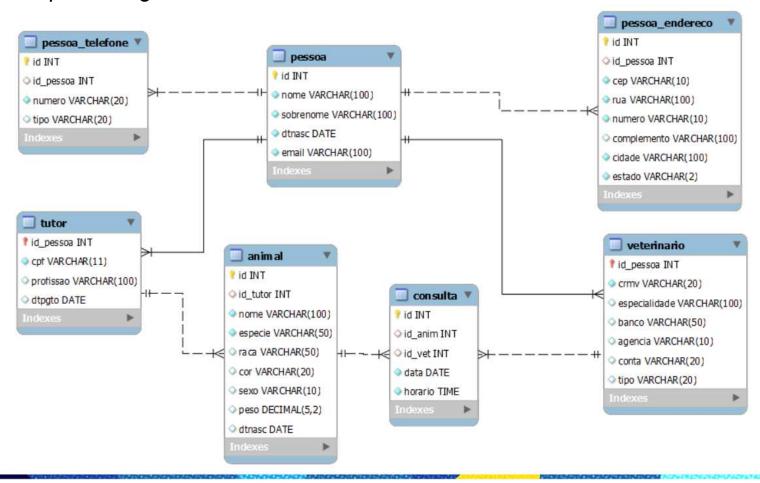
Animal(<u>id</u>, id_tutor, nome, espécie, raca, cor, sexo, peso, dtnasc) id_tutor referencia Tutor(id_pessoa)

Consulta(<u>id</u>, id_anim, id_vet, data, horario)

id_anim referencia Animal(id)

id_vet referencia Veterionario(id_pessoa)

MER → Esquema lógico 01



Normalização

- Conjunto de regras aplicadas numa tabela com o objetivo de corrigir possíveis erros de projeto.
- Existem diversas regras de normalização.
- São aplicadas no Projeto Conceitual ou no Projeto Lógico!



Objetivos

- Elimina redundância de dados;
- Facilita manutenção;
- Reduz o espaço de armazenamento.
- Faz com que a informação esteja sempre consistente.
 - Evita o risco de, ao gerar uma consulta, o dados virem desatualizados, pois a informação está armazenada em dois locais (um desatualizado e um atualizado).



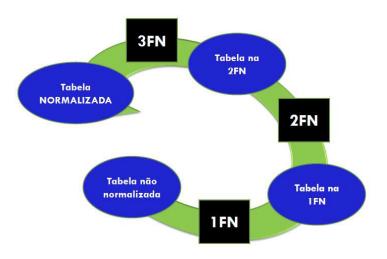
Formas Normais

- Uma série de regras aplicadas em um BD para garantir que as tabelas possuam uma boa estruturada.
- As regras são chamadas Formas Normais:
 - □ 1ª Forma Normal (1 FN);
 - □ 2ª Forma Normal (2 FN);
 - □ 3ª Forma Normal (3 FN).



Formas Normais







1ª Forma Normal (1 FN)

- Dizemos que uma tabela está na primeira forma normal, quando ela não possui:
 - Atributos multivalorados;
 - Atributos compostos.
- Cada atributo composto ou multivalorado deve ser transformado numa nova tabela.



1ª Forma Normal (1 FN)

Código	Nome	Endereço	Telefone			
1001	Diego Machado	Rua Tal 321 Porto	5312345678 5398765432			
1002	Fulano de Tal	Avenida Tal 71 Centro	1 5187654321 5143215678 Nome		Endereço	Bairro
		Código				
	1001	Diego Machado		Rua Tal 321	Porto	
		1002	Fulano	ılano de Tal	Avenida Tal 71	Centro
		Código	Telefone 53123456		+stat Nove	
		1001				Į.
		1001	53987654		132	
		1002	51123456		578	
		1002		31123430	,,,	

Dependência Funcional (DF)

- Para avançar para a 2 FN e para a 3 FN é necessário conhecer o conceito de dependência funcional.
- É a principal ferramenta de avaliação para identificar se o agrupamento de atributos de uma tabela é apropriado:
 - Evitando redundância de dados;
 - Inconsistências;
 - Perda de dados em operações de remoções ou alterações.



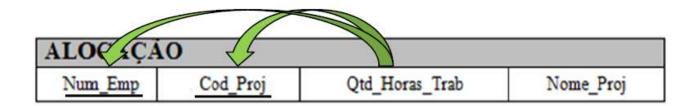
Dependência Funcional (DF)

- Existem 3 tipos:
 - Dependência Funcional Total (1 FN);
 - Dependência Funcional Parcial (2 FN);
 - Dependência Funcional Transitiva (3 FN).



DF Total

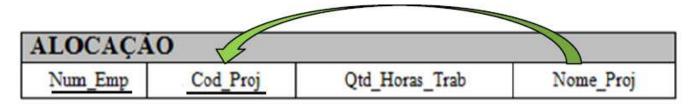
- Os atributos não chave de uma tabela têm que depender totalmente da chave primária e somente dela.
 - Exemplo: uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional total se e somente se C depender funcionalmente de A e B.





DF Parcial

- Os atributos não chave de uma tabela dependem de parte da chave primária.
 - Exemplo: uma determinada tabela possui sua chave primária composta pelos atributos A e B. Logo, C será dependente funcional parcial se e somente se C depender funcionalmente de A ou B.





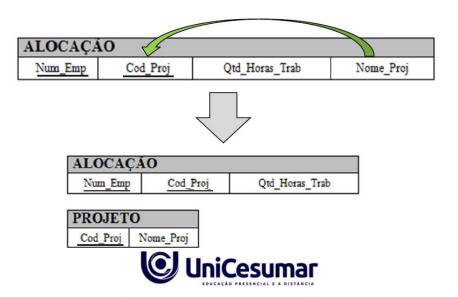
2ª Forma Normal (2 FN)

- Uma tabela está na 2 FN se ela já estiver na 1 FN.
- Todo atributo que n\(\tilde{a}\)o for chave prim\(\tilde{a}\)ria deve ser dependente funcional total:
 - Não deve existir dependência funcional parcial.
 - O campo não chave deverá depender de todos os campos que compõem a chave primária.



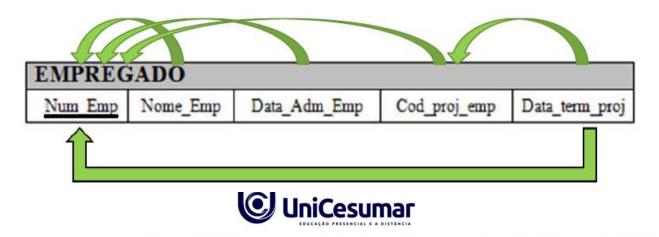
2ª Forma Normal (2 FN)

Solução para dependência funcional parcial:



DF Transitiva

 O atributo C é dependente funcional transitivo de A se C é funcionalmente dependente de B e B funcionalmente dependente de A, na mesma tabela.



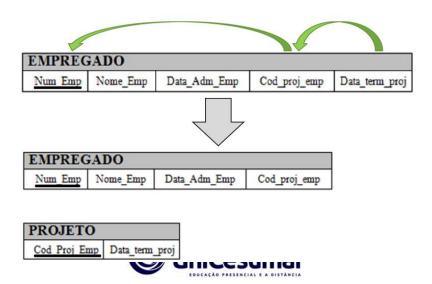
3ª Forma Normal (3 FN)

- Uma tabela está na 3 FN se ela já estiver na 2 FN.
 - Não deve existir dependência funcional transitiva entre atributos não chave.



3ª Forma Normal (3 FN)

Solução para dependência funcional transitiva:



3ª Forma Normal (3 FN)

- A terceira forma normal se preocupa em transformar o esquema de tabela não normalizado em um esquema relacional sem tabelas aninhadas.
- Projeto_Empregado (<u>ID</u>, Tipo, Descricao, ID_Empregado, Nome, Setor, Salario, Data_Admissao)
 - |
- Projeto (<u>ID</u>, Tipo, Descricao)
- Empregado (<u>ID</u>, ID_Projeto, Nome, Setor, Salario, Data_Admissao)



Vamos praticar!?





Questão 1

O projeto lógico abaixo está normalizado?

```
aluno (<u>id</u>, nome, idade, status)
disciplina (<u>id</u>, nome)
escola (<u>id</u>, nome)
curso (<u>id</u>, id_escola, nome)
grade_curricular (<u>id</u>, id_curso, nome, status)
aluno_disciplina(<u>id</u> aluno, <u>id</u> disciplina)
grade_disciplina(<u>id</u> grade, <u>id</u> disciplina)
```



Questão 1 – Resolução

O projeto lógico abaixo está normalizado? Sim.

```
aluno (<u>id</u>, nome, idade, status)
disciplina (<u>id</u>, nome)
escola (<u>id</u>, nome)
curso (<u>id</u>, id_escola, nome)
grade_curricular (<u>id</u>, id_curso, nome, status)
aluno_disciplina(<u>id aluno</u>, <u>id disciplina</u>)
grade_disciplina(<u>id grade</u>, <u>id disciplina</u>)
```



Questão 2

O projeto lógico abaixo está normalizado?

```
veiculo (<u>id</u>, nome, modelo, marca, valor)
cliente (<u>id</u>, nome, cpf, data_nascimento, endereco, telefone*, status)
vendedor (<u>id</u>, nome, comissao, data_admissao, endereco,
telefone*, status)
venda (<u>id</u>, id_cliente, id_vendedor, id_veiculo, data, valor_total,
valor_desconto, forma_pagamento, status)
```



Questão 2 – Resolução

```
marca (id, nome)
estado (id, nome, uf)
cor (id, nome)
modelo (id, id marca, nome)
veiculo (id, id modelo, nome, valor)
cidade (id, id estado, nome)
endereco (id, id cidade, logradouro, numero, bairro, cep)
cliente (id, id endereco, nome, cpf, data nascimento, status)
telefone cliente (id, id cliente, numero, tipo)
vendedor (id, id endereco, nome, comissao, data admissao, status)
telefone vendedor (id, id vendedor, numero, tipo)
venda (id, id cliente, id vendedor, id veiculo, data, valor total, valor desconto,
forma pagamento, status)
```



Questão 3

O projeto lógico abaixo está normalizado?

```
livro (id. codigo interno, titulo, resumo, volume, ano, edicao, editora, autores, genero,
qtd pagina, qtd exemplar, isbn, idioma, disponibilidade, tipo capa)
usuario (id, nome, email, codigo pessoa, senha)
aluno (id, id usuario, curso, matricula, status)
comunidade (id, id usuario, cpf, endereco, telefone)
funcionario (id, id usuario, cargo, setor, ramal, status)
emprestimo(id, id usuario, data registro, data devolução, qtd renovação, status)
livro emprestimo (id emprestimo, id livro)
reserva(id, id_usuario, data_registro, prioridade, status)
livro reserva (id reserva, id livro)
multa(<u>id</u>, id emprestimo, valor, data vencimento, tipo pagamento, status)
```



Questão 3 - Resolução

```
autor (id, nome)
editora (id. nome)
genero (id. nome)
idioma (id, nome)
disponibilidade (id, nome)
tipo capa (id, nome)
setor (id, nome)
curso (id. nome)
prioridade (id, nome)
estado (id. nome, uf)
tipo pagamento (id, nome)
cidade (id. id estado, nome)
livro (id, id editora, id idioma, id disponibilidade, id tipo capa, codigo interno, titulo, resumo, volume, ano, edicao, qtd pagina, qtd exemplar, isbn)
autor livro (id livro, id autor)
genero livro (id livro, id genero)
usuario (id, codigo pessoa, nome, email, senha)
aluno (id. id usuario, id curso, matricula, status)
comunidade (id. id usuario, id cidade, cpf, logradouro, bairro, numero, cep)
comunidade telefone (id. id comunidade, telefone, tipo)
funcionario (id. id usuario, id setor, ramal, status)
emprestimo(id, id_usuario, data_registro, data_devolucao, qtd_renovacao, status)
livro emprestimo (id emprestimo, id livro)
reserva(id, id usuario, id prioridade, data registro, status)
livro reserva (id reserva, id livro)
multa(id, id emprestimo, id tipo pagamento, data vencimento, valor, status)
```



REFERÊNCIAS

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de gerenciamento de banco de dados. São Paulo, SP: McGraw-Hill Interamericana do Brasil, 2008. xxvii, 884 p.

NAVATHE, Shamkant B.; ELMASRI, Ramez. Sistemas de banco de dados. Sham, Addison. Ribeirão Preto SP, 2005.



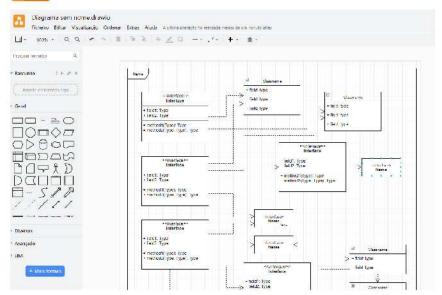
Ferramentas CASE





Ferramentas CASE

draw.io





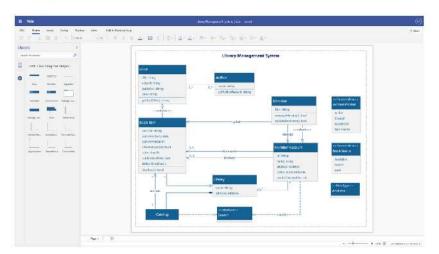




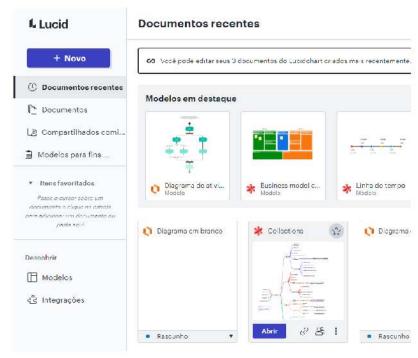


Ferramentas CASE











"Sucesso é o acúmulo de pequenos esforços, repetidos dia e noite."

Robert Collier



OBRIGADO E BONS ESTUDOS!



