

Um atributo pertence a uma chave
 \Rightarrow Atributo primo

- Dependência Funcional Elementar

Seja $R(X, Y, Z, \dots)$ com X, Y, Z , conjuntos de atributos

a DF $X \rightarrow Y$ é elementar
se $\forall X' \subset X : X' \not\rightarrow Y$

- Dependência Funcional Parcial

Se $X \rightarrow Y$ e $X' \rightarrow Y'$ ($X' \subset X$)
dig-se que $X \rightarrow Y$ é uma
dependência parcial

Primeira forma normal (1FN)

- Cada atributo contém apenas valores atômicos
- Não há conjuntos de atributos repetidos descrevendo a mesma característica

PessoaCursos1

Nome	Endereço	NIF	Cursos
Artur	Covilhã	123456789	Programador
Ana	Fundão	222222222	Operador, Programador
Carlos	Covilhã	222333444	Analista, Programador, Operador
Paulo	Guarda	555666777	Operador, Analista

- O atributo Cursos contém valores não atômicos !!!

PessoaCursos2

Nome	Endereço	NIF	Curso1	Curso2	Curso3
Artur	Covilhã	123456789	Programador		
Ana	Fundão	222222222	Operador	Programador	
Carlos	Covilhã	222333444	Analista	Programador	Operador
Paulo	Guarda	555666777	Operador	Analista	

- São repetidos atributos do mesmo tipo, curso1, curso2, curso3.
(Diz-se que a relação tem um grupo repetitivo)
- Os tuplos correspondentes a alunos com apenas 1 ou dois cursos vão ter valores nulos para alguns atributos.

Segunda forma normal (2FN)

Seja $R(A_1, A_2, \dots, A_n)$

R está na 2FN

se $\forall A_i \notin \text{chave}(R)$,

$\forall X \text{ chave}$ se verifica $X \rightarrow A_i$
é elementar

De outra forma:

- Uma relação está na 2ª forma normal se está na 1ª FN e os atributos que não são chave dependem da totalidade da chave.

Nota:

Um atributo pertencente a uma chave diz-se um atributo primo

EXEMPLO

Linha-mota-enc (N-mota-enc,
Desc-produto,
Preço-prod,
Quantidade)

Dependências Funcionais

$\{ N-mota-enc, Cod-produto \} \rightarrow Desc-produto$

$\{ N-mota-enc, Cod-produto \} \rightarrow Preço-produto$

Mão são elementares, pois:

$Cod-produto \rightarrow Desc-produto$ (1)

$Cod-produto \rightarrow Preço-produto$ (2)

$Desc-produto \notin \{ N-mota-enc, Cod-prod \}$

$\forall X$ chave ———— \uparrow

($X \rightarrow Desc-prod$ é elemental)

$\Rightarrow \forall X' \subset X$ (p.en. $Cod-produto$)

$X' \not\rightarrow Desc-prod$

$X \rightarrow Y$ (primeira forma normal)
 \rightarrow FALSO!

Casos especiais de relações em 2FN:

- Se todos os atributos de uma relação são primos.
ou
- A chave da relação consiste num único atributo.

\Rightarrow Está na 2FN

Terceira Forma Normal (3FN)

$N_emp \rightarrow Dep$

$Dep \not\rightarrow N_emp$

$Dep \rightarrow Cidade$

$Cidade \not\rightarrow Dep$

E

N_emp	Dep	Cidade
a	A	X
b	A	X
c	A	X
d	B	Y
e	B	Y
f	C	X
g	C	X

Anomalias

Inserção: não podemos inserir um departamento se não tem empregados

Eliminação: Se eliminamos o último empregado de um departamento perdemos a informação do departamento

Modificação: O atributo cidade é repetido para cada empregado de um dado departamento; Uma alteração da localização de um departamento implica alterar a localização de todos os empregados desse departamento.

Dependência Funcional DIRETA

Dependência Funcional DIRETA

$R(X, Y, Z, \dots)$ $X \rightarrow Z$ é direta
se $\nexists Y \in R : Y \nrightarrow X$ e $Y \rightarrow Z$

$$\underbrace{X \xleftrightarrow{\quad} (\exists Y) \rightarrow Z}_{X \rightarrow Z \text{ é direta}}$$

• Definição 3FN

$R(A_1, A_2, \dots, A_n)$

$\forall A_i \notin \text{chave}(R),$

$\forall X \text{ chave}, \text{ se verifica que:}$

$X \rightarrow A_i$ é direta

EXEMPLO

Nota-de-enc (
N-nota-enc, Cod-Cliente, Nome-Cliente,
Morada-cliente
)

Cod-Cliente \rightarrow (Nome-cliente,
Morada-cliente)

N-nota-enc \rightarrow Nome-cliente,
Morada-cliente

Merada - cliente

$$\Rightarrow \exists Y \in R : Y \nrightarrow X \wedge Y \rightarrow Z$$

\Rightarrow não é direta

Exercícios do teste modelo 2023

30 de maio de 2023

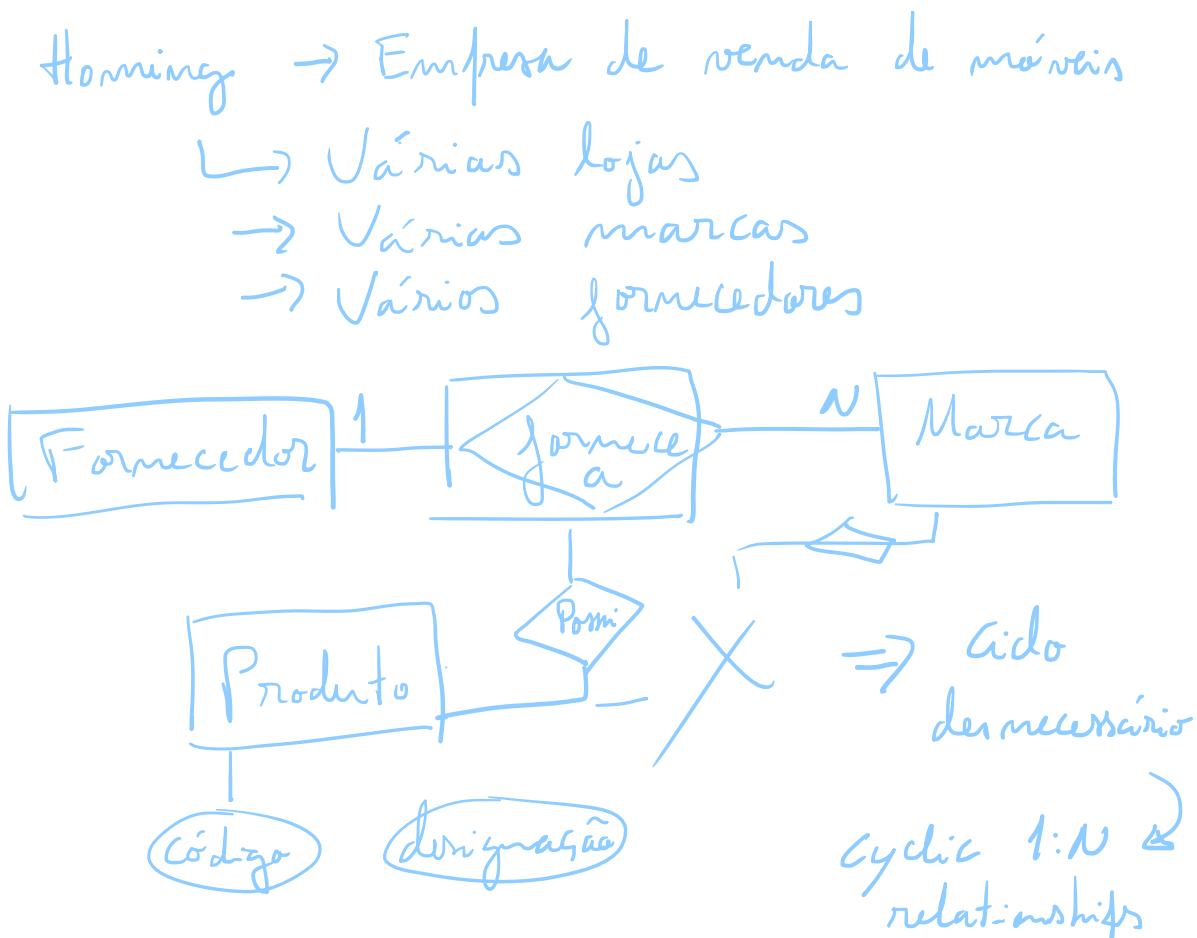
19:41

7. Considere o seguinte caso para estudo:

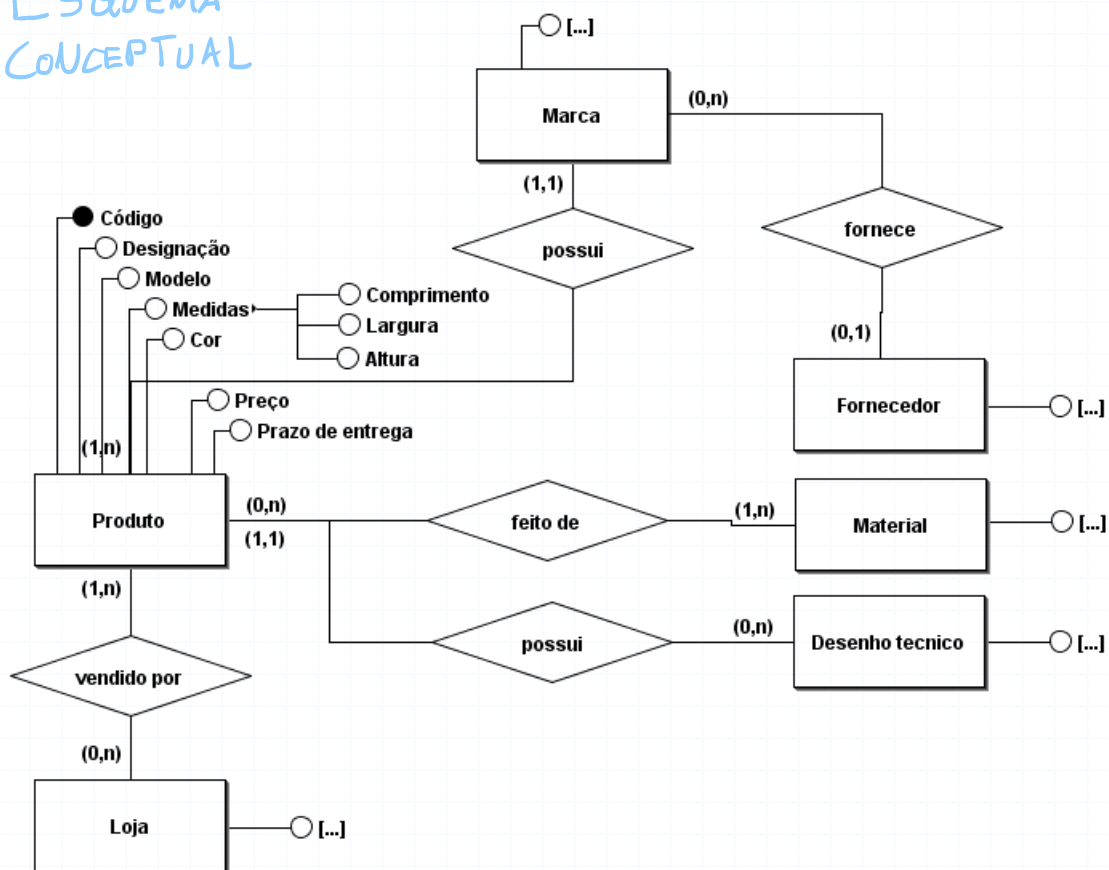
A "Homing" é uma empresa de vendas de móveis para ambientes domésticos. Hoje, as suas atividades comerciais cobrem já todo o território nacional e são suportadas por um conjunto muito diversificado de lojas, localizadas nos principais centros comerciais do país. A empresa representa várias marcas de móveis conceituadas, cujo fornecimento dos produtos é assegurado por um conjunto de fornecedores selecionados. A empresa permite que um seu fornecedor possa fornecer mais do que uma marca, mas não aceita que uma marca possa ser fornecida por mais do que um fornecedor. Para cada uma dessas marcas possui um catálogo para os seus produtos que integra, em termos gerais, vários itens de dados comuns acerca dos produtos que nele figuram, nomeadamente: código, designação, modelo, medidas (comprimento, largura, altura e profundidade), cor, materiais que incorpora, preço de venda, prazo de entrega, e um conjunto de desenhos técnicos. Os catálogos, logo após a sua emissão – data de emissão –, são avaliados pelo diretor comercial da empresa que determina, com base em historiais de vendas que tem disponível, quais serão as lojas que irão comercializar, e consequentemente ter em loja, os produtos desse catálogo. (...)

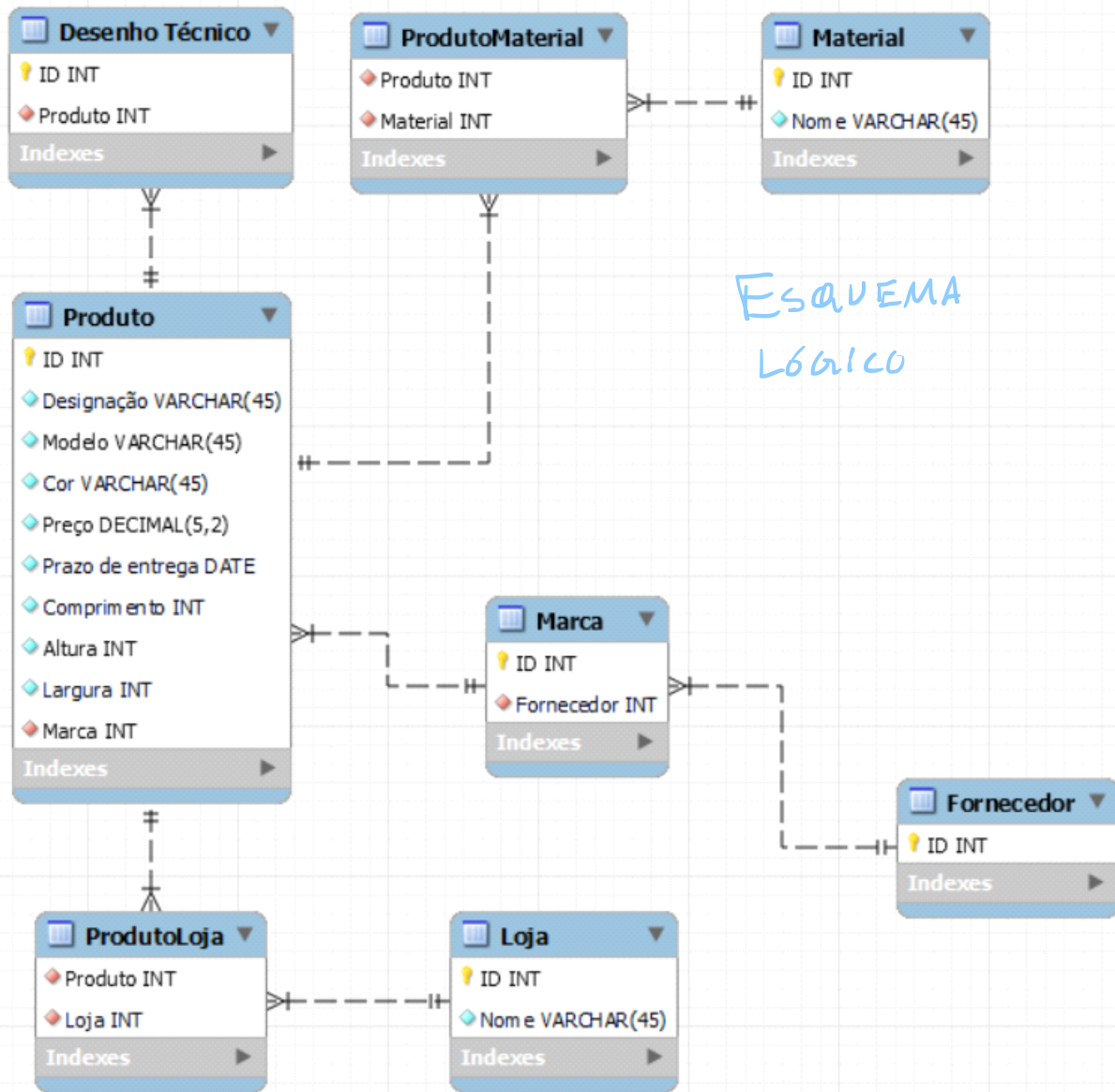
Com base no caso de estudo e na lista de requisitos apresentada, pretende-se que desenvolva:

a) um esquema CONCEPTUAL para uma base de dados capaz de acolher a informação que o caso revela;



ESQUEMA CONCEPTUAL





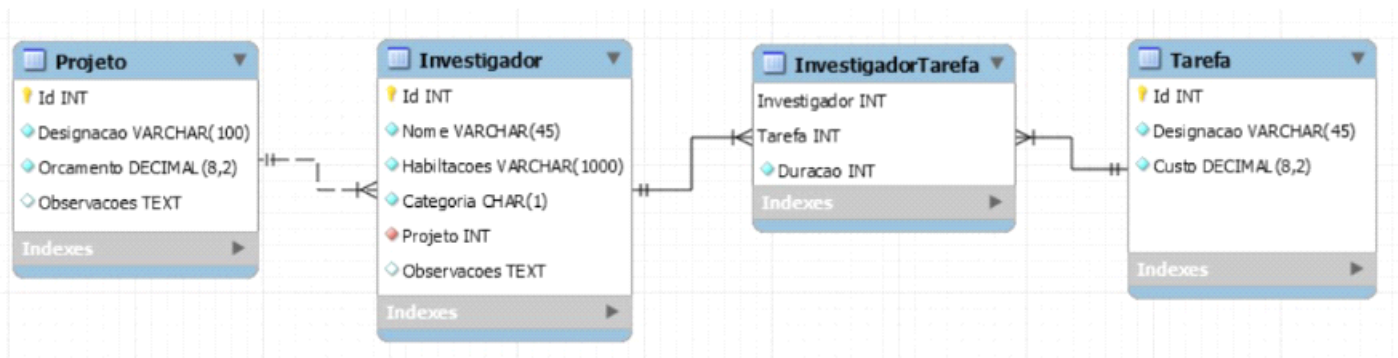


Figura 1 – Esquema lógico de uma base de dados de “Projetos”.

- b. Tendo em consideração o esquema lógico da base de dados apresentado na Figura 1, pretende-se que apresente em SQL as instruções necessárias para realizar as seguintes operações:
- acrescentar na tabela “Investigador” dois novos atributos, “DataNascimento” e “eMail”, caracterizando-os de forma adequada;
 - obter uma lista com os projetos que tenham um orçamento superior a ‘100.000,00’€, cujos investigadores responsáveis sejam da categoria ‘A’;
 - criar uma vista que forneça uma lista com os nomes dos investigadores dos projetos, ordenada decrescentemente por orçamento dos projetos;
 - remover da base de dados toda a informação relativa à tarefa ‘Limpeza de Microscópio’;
 - desenvolver uma função que permita obter o tempo total relativo à realização das tarefas associadas com um dado investigador.

a)

```
ALTER TABLE Investigador
    ADD DataNascimento DATE() NOT NULL,
    ADD eMail VARCHAR(40) NOT NULL;
```

b)

```
select P.*
from Projeto as P
    inner join Investigador as I on P.ID = I.Projeto
where I.Categoria = 'A' and P.Orcamento > 100000.0;
```

c)

```

-- sem safe mode
delete T1
    from Tarefa
    where Designação = 'Limpeza de Microscópio';

-- com safe mode
delete T1
    from Tarefa as T1
    inner join (
        select ID
            from Tarefa
            where Designação = 'Limpeza de Microscópio'
        ) as subQuery
    on T1.ID = subQuery.ID;

```

d)

```

DELIMITER $$
create procedure TempoTotalTarefasInvestigador
    (in ID INT)
begin
    select sum(IT.Duracao)
        from Investigador as I
        inner join InvestigadorTarefa as IT on I.ID = IT.Investigador
        group by I.ID
        having I.ID = ID;
end $$

call TempoTotalTarefasInvestigador(1);

```

Criação das tabelas:

```

1  create table Projeto (
2      ID int not null,
3      Designacao varchar(100) not null,
4      Orcamento decimal(8,2) not null,
5      Observacoes text,
6      primary key (ID)
7  )
8  engine = InnoDB;
9
10 insert into Projeto values
11     (1, 'random', 100000, 'random'),
12     (2, 'random', 200000, 'random'),
13     (3, 'random', 300000, 'random'),
14     (4, 'random', 400000, 'random');
15
16 create table Investigador (
17     ID int not null,
18     Nome varchar(45) not null,
19     Habilitacoes varchar(1000) not null.

```

```

20     Categoria char(1) not null,
21     Projeto int,
22     Observacoes text,
23     index Projeto_idx (Projeto asc) visible,
24     constraint InvestigadorProjeto
25         foreign key (Projeto)
26         references bookstore.Projeto (ID)
27         on delete set null
28         on update no action,
29     primary key (ID)
30 )
31 engine = InnoDB;
32
33 insert into Investigador values
34     (1, 'random', 'random', 'a', 4, 'random'),
35     (2, 'random', 'random', 'b', 3, 'random'),
36     (3, 'random', 'random', 'c', 2, 'random'),
37     (4, 'random', 'random', 'd', 1, 'random');
38
39 create table Tarefa (
40     ID int not null,
41     Designacao varchar(45),
42     Custo decimal(8,2),
43     primary key (ID)
44 )
45 engine = InnoDB;
46
47 insert into Tarefa values
48     (1, 'random', 1000),
49     (2, 'random', 2000),
50     (3, 'random', 3000),
51     (4, 'random', 4000);
52
53 create table InvestigadorTarefa (
54     Investigador int not null,
55     Tarefa int not null,
56     Duracao int not null,
57     index Investigador_idx (Investigador asc) visible,
58     index Tarefa_idx (Tarefa asc) visible,
59     constraint ITInvestigador
60         foreign key (Investigador)
61         references bookstore.Investigador (ID)
62         on delete cascade
63         on update no action,
64     constraint ITTarefa
65         foreign key (Tarefa)
66         references bookstore.Tarefa (ID)
67         on delete cascade
68         on update no action
69 )
70 engine = InnoDB;
71
72 insert into InvestigadorTarefa values
73     (1, 2, 3),
74     (1, 3, 4),
75     (2, 3, 5),
76     (3, 1, 2),
77     (4, 4, 6);

```

Anotações aleatórias

31 de maio de 2023 23:42

- Adicionar Fornecimento ✓
- Calcular Preço líquido ✓

Alterar tabela:

```
-- Alteração do nome da tabela "".  
ALTER TABLE Receita  
| RENAME TO Receitas;  
  
-- Adição de um novo atributo numa tabela.  
--  
ALTER TABLE TiposReceita  
| ADD Observações TEXT NULL;  
  
-- Adição simultânea de vários atributos numa tabela.  
--  
ALTER TABLE TiposReceita  
| ADD ValorBase DECIMAL(10,2) NOT NULL DEFAULT 0  
| | AFTER Observações;  
  
-- Modificação da definição de vários atributos de uma tabela.  
--  
ALTER TABLE TiposReceita  
| MODIFY Observações VARCHAR(25) NOT NULL,  
| MODIFY ValorBase DECIMAL(8,2) AFTER Descrição;  
  
-- Modificação da definição de vários atributos de uma tabela.  
--  
ALTER TABLE Receita  
| ADD Facebook VARCHAR(200);  
ALTER TABLE Receita  
| CHANGE COLUMN Facebook Instagram VARCHAR(150) NULL;  
  
-- Remoção de um atributo numa tabela.  
--  
ALTER TABLE Receita  
| DROP COLUMN Facebook;
```

```
SELECT *  
| FROM staff  
| WHERE staff_id =9;  
-- Desfazer a operação realizada.  
ROLLBACK;
```

1 DELIMITER \$\$

```

2  create procedure calcularPreçoEncomenda (in EncomendaID int)
3  begin
4      declare result decimal (6,2);
5      declare eID int;
6      declare codigoValor int;
7      declare correçãoDePreço decimal(6, 2);
8
9      select E.ID, sum(L.Preço), coalesce(C.Valor, 0) into eID, result,
10     codigoValor
11     from Encomenda as E
12         left join `Código promocional` as C on E.Código = C.Código
13         inner join EncomendaLivro as EL on E.ID =
14     EL.Encomenda
15         inner join Livro as L on EL.Livro = L.ID
16         group by E.ID
17     having E.ID = EncomendaID;
18
19     set correçãoDePreço = result * (1 - (codigoValor / 100.0));
20     select eID as Encomenda,
21         result as `Preço total`,
22         concat(codigoValor, '%') as 'Promoção',
23         correçãoDePreço as `Correção de preço`;
24
25     update Encomenda
26         set Valor = correçãoDePreço
27     where ID = EncomendaID;
28
29 end $$
30 DELIMITER ;

```

Nice.

1 2 3 4 5 6 7 8 9 10
16-17-18-19-20-21-22-23-00-1-2

• Exercícios

• Testes

A

a1	a2	a3
...

B

b1	b2	b3	b4	a1	a2
		...			

C

c1	c2	c3	b1	c4	c5
			...		

a1 → a2, a3

$b_1 \rightarrow b_2, b_3, b_4, a_1$

$c_1 \rightarrow c_2, c_3, b_1$

$c_3 \rightarrow c_4, c_5$

?

A			B				
a1	a2	a3	b1	b2	b3	b4	a1

C				C2		
c1	c2	c3	b1	c3	c4	c5

$c_1 \rightarrow c_3 \rightarrow c_4$

\hookrightarrow não é direta! \Rightarrow Nova tabela

$b_1 \rightarrow a_1$

$a_1 \rightarrow a_2$

Importância da identificação e caracterização de entidades

Uma entidade é um objeto ou um conceito que está claramente identificado numa organização como tendo uma existência independente. Um dos métodos para identificar entidades é através da análise da especificação de requisitos.

Uma forma alternativa de identificação pode ser realizada através de observação de objetos que têm uma existência por direito próprio - ex.: mobiliário, equipamentos, etc.

Caracterização de uma entidade: Nome, descrição, sinónimos, ocorrência

Em relação a relacionamentos, no pun intended, é necessário determinar a cardinalidade e restrições de participação dos tipos de relacionamentos.

TESTE 2018/2019

2. Uma das “regras” de verificação que a teoria da normalização considera é reconhecida como terceira forma normal – 3FN. De forma sucinta explique essa forma normal, ilustrando a sua explicação com um pequeno exemplo prático da sua aplicação.

De < https://elearning.uminho.pt/bbcswebdav/pid-1312065-dt-content-rid-6919799_1/courses/2223.J304N3_2/201819-UC-BD-Teste.pdf >

$$\left[\begin{array}{l} R(X, Y, Z, \dots) \quad X \rightarrow Z \text{ é direta} \\ \text{me } \nexists Y \in R : Y \nrightarrow X \wedge Y \rightarrow Z \end{array} \right]$$

Definição 3FN:

$R(A_1, A_2, A_3, \dots)$

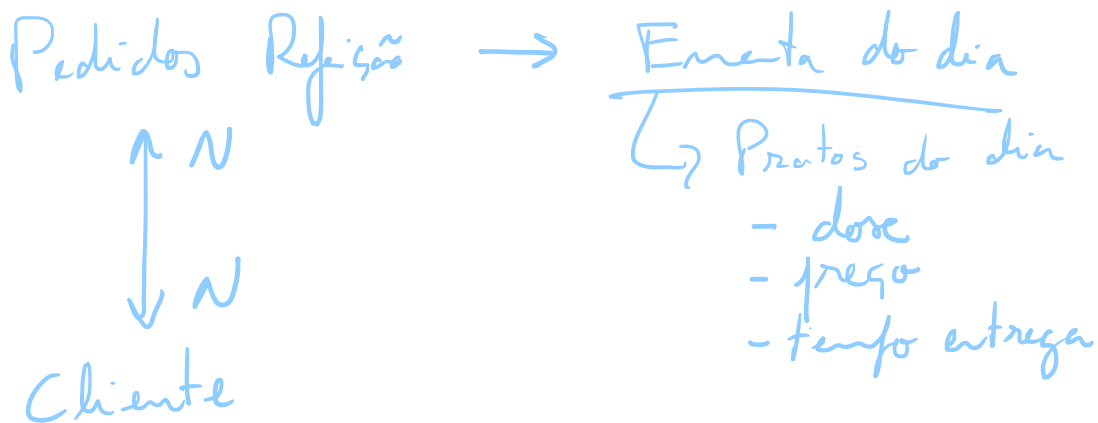
$\forall A_i \notin \text{super chave}$

$\forall X \text{ chave} : X \rightarrow A_i \text{ é direta}$

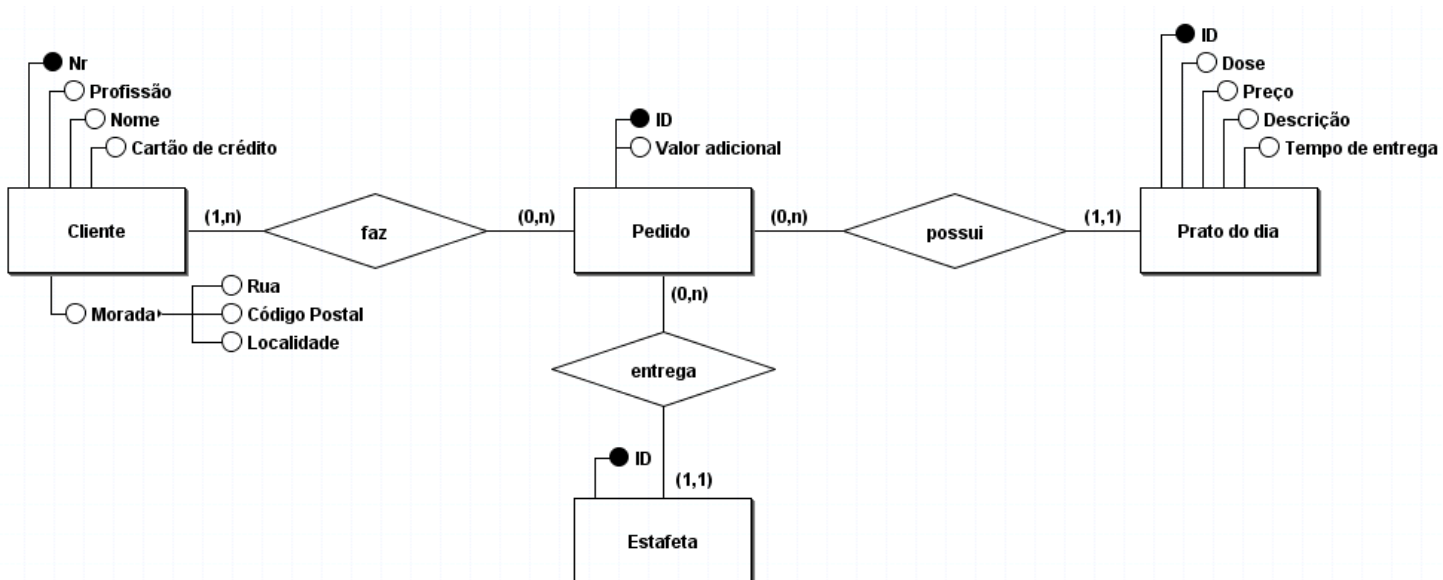
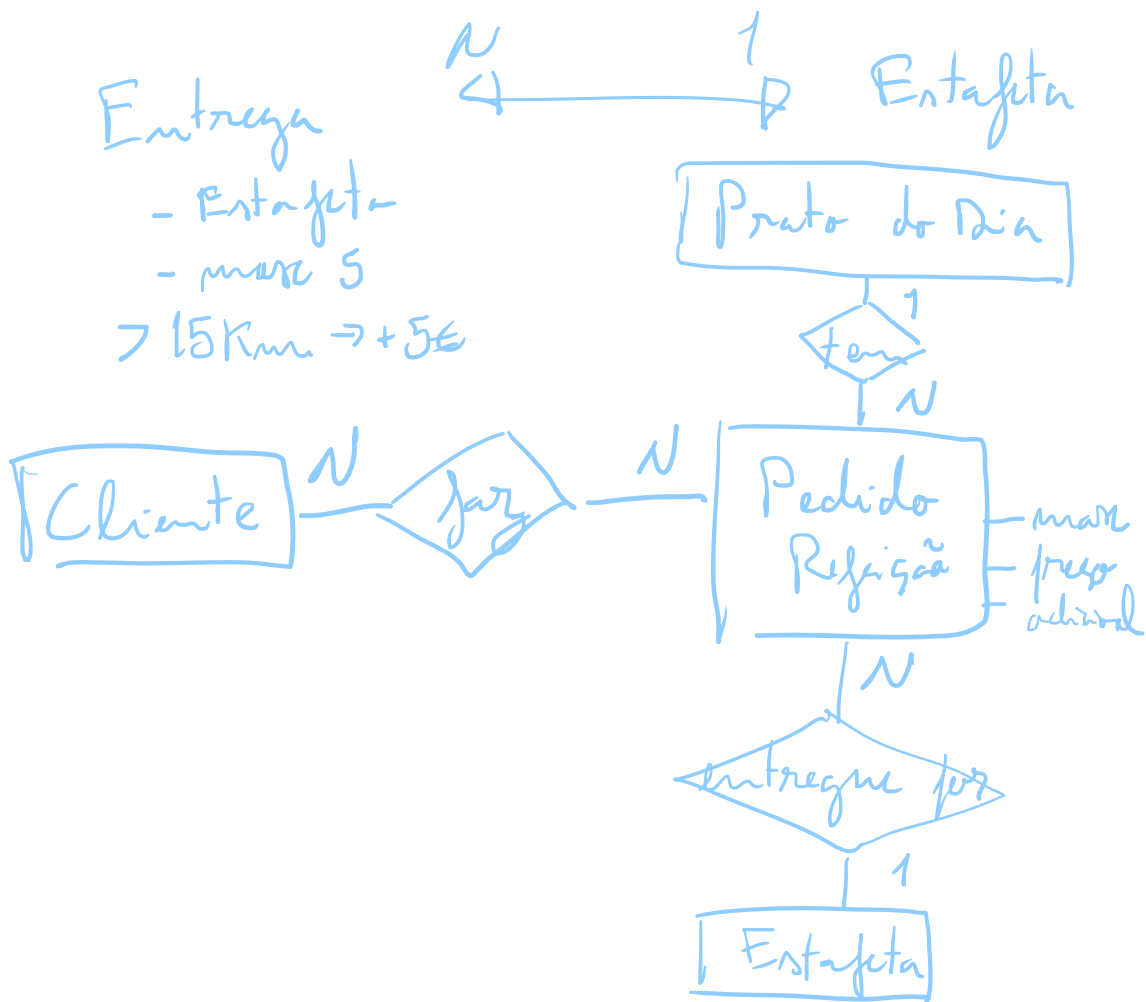
ou seja

$$\nexists A_j : A_j \nrightarrow X \wedge A_j \rightarrow A_i, i \neq j$$

PARTE 2



- nome
 - nome
 - profissão
 - morada
 - cartão de crédito
- { sua
 localidade
 código Postal



PARTE III

```

1 create table if not exists Explicações (
2     Nr int not null,
3     Data date not null,
4     HoraInício time not null,

```

```

5      HoraFim time not null,
6      Disciplina int not null,
7      Monitor int not null,
8      Estudante int not null,
9      Sumario varchar(150) not null,
10     Observações text null,
11     index Disciplina_idx (Disciplina asc) visible,
12     index Monitor_idx (Monitor asc) visible,
13     index Estudante_idx (Estudante asc) visible,
14     constraint EdisciplinaNr
15         foreign key (Disciplina)
16             references Disciplina (Nr)
17             on delete cascade
18             on update cascade,
19     constraint EmonitorNr
20         foreign key (Monitor)
21             references Monitor (Nr)
22             on delete cascade
23             on update cascade,
24     constraint EestudanteNr
25         foreign key (Estudante)
26             references Estudante (Nr)
27             on delete cascade
28             on update cascade,
29     primary key (Nr))
30 engine = InnoDB;

```