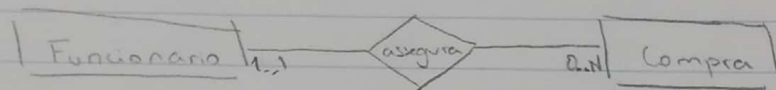


Teste - 2015/2016

1. O administrador é responsável por:

- realização física da base de dados, ou seja, pelo design e implementação da mesma
- segurança e controlo de integridade
- manutenção do sistema operativo
- garantir um desempenho satisfatório das aplicações para os utilizadores

2. Tomemos o seguinte ER como exemplo:



Este diagrama representa uma relação entre as entidades "Funcionario" e "Compra", na qual um funcionário poderá assegurar 0 ou mais compras e uma compra apenas é assegurada por um funcionário.

3. A normalização de bases de dados é um processo de remoção de dados redundantes das tabelas que a constituem, com o intuito de aumentar a eficiência de armazenamento, a integridade dos dados e as suas características de estabilidade.

1ª Forma Normal - se todos os valores de todos os atributos forem atômicos, isto é, se não for possível decompor-los

- cada entrada apenas tem um valor
- o domínio de cada atributo de uma relação apenas pode conter valores atômicos, e o valor de cada atributo é constituído apenas por um simples valor do seu domínio.

prim - pertence à chave candidata

2ª Forma Normal - os atributos não-primos (não pertencem à chave candidata) não pode ser funcionalmente dependentes de um subconjunto próprio de qualquer chave candidata

- se todos os seus atributos não-primos forem totalmente dependentes da sua chave primária
- se estiver em 1ª forma normal.

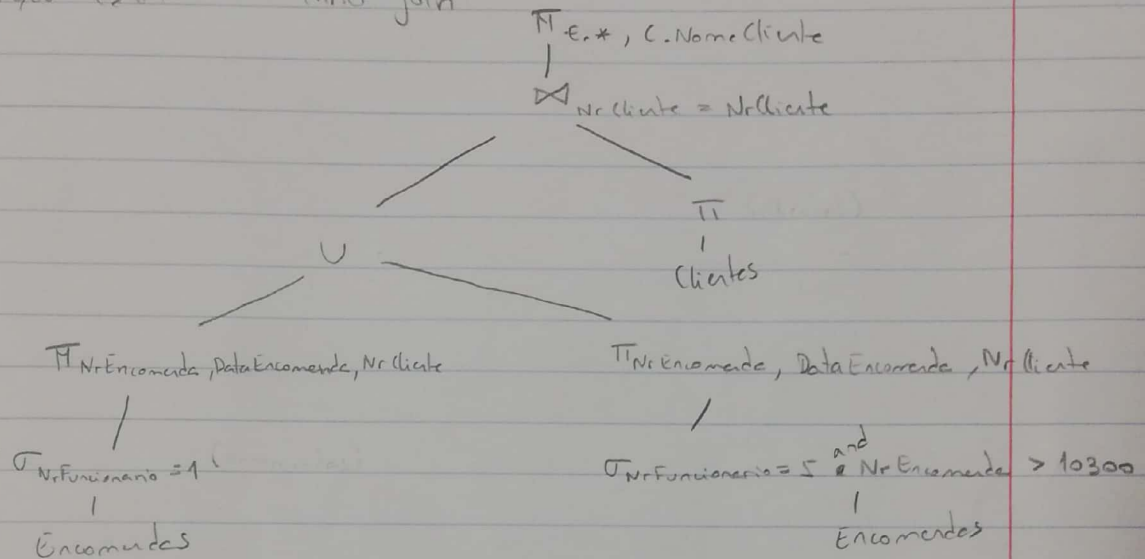
3ª Forma Normal - se escrever em 2ª Forma Normal

- Nenhum atributo não primo depende de atributos não primos.
(completar melhor)

4. Seleção (σ) - (Select * from tabela) where...

Projeção (π) - Select abcd from tabela

Junção (\Join) - inner join



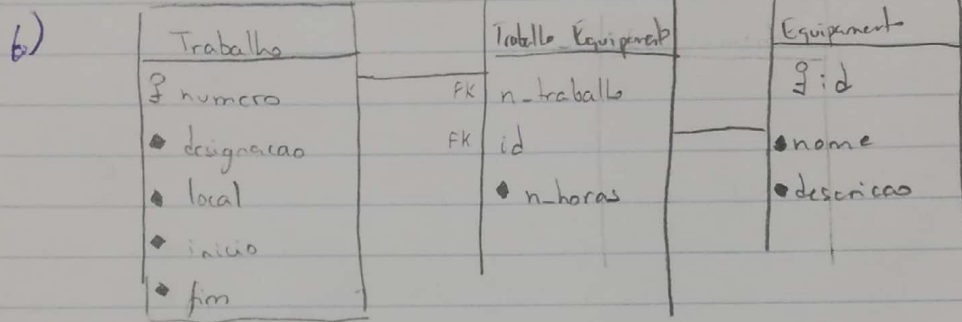
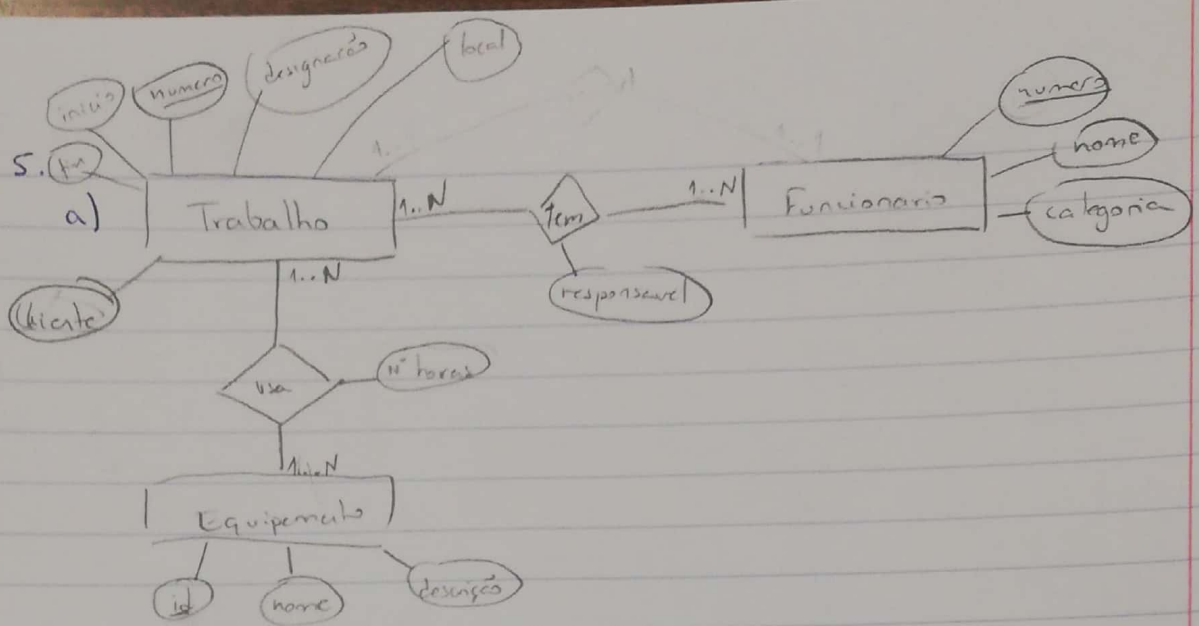
$\pi_{E.*, C.Nome_cliente} (E \Join_{E.Nr_cliente = C.Nr_cliente} C)$

$C \leftarrow \pi (Clientes)$

$E \leftarrow A \cup B$

$A \leftarrow \pi_{Nr_Encomenda, Data_Encomenda, Nr_cliente} (\sigma_{Nr_funcionario = 1} (Encomendas))$

$B \leftarrow \pi_{Nr_Encomenda, Data_Encomenda, Nr_cliente} (\sigma_{Nr_funcionario = 5 \wedge Nr_Encomenda > 10300} (Encomendas))$

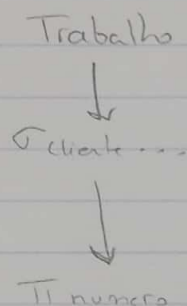


(...)

c)

a) Select numero from Trabalho
 where cliente = 'Pedras Brutas, SA' and inicio >= '2015-01-01'
 and fim <= '2015-12-31';

$\Pi_{\text{numero}} (\sigma_{\text{cliente} = \text{'Pedras...'} \text{ and } \text{inicio} \geq \text{'2015-01-01'} \text{ and } \text{fim} \leq \text{'2015-12-31'}} (\text{Trabalho}))$



b) Select n-trabalho, id, n-horas from Trabalho-Equipamento
 where n-trabalho = 1 or
 n-trabalho = 3 or
 n-trabalho = 5

6.

a) Select distinct F.Descricao from Funcionario as F
 inner join Despesa as D
 on F.id = D.Funcionario
 where D.FormaPagamento = (select id from FormaPagamento
 where Descricao = 'Cartão de crédito')
 and
 D.Cliente in ('66', '77', '88', '99') and
 D.DataHora >= '2015-01-01' and D.DataHora < '2016-01-01'
 order by F.Descricao;

b) Create view montante-despesas as
Select year(DataHora) as Ano, sum(Valor) as Montante
From Despesa
group by Ano
order by Ano desc;

c) Create Table ClienteVisitas (
NrCliente INT NOT NULL,
NrVisita INT AUTO-INCREMENT,
Data Date,
primary key (NrCliente, NrVisita),
Constraint fk_cliente
Foreign key (NrCliente)
references Cliente (NrCliente));

d) Delimiter \$\$
Create Procedure p() (in Numero int)
Begin
Insert into ClienteVisitas
Select Cliente, Nr, DataHora From Despesa
where Cliente = Numero
End \$\$