

Normalização
1 Forma normal, não podem ter recursos com mais de um item (vetores)
dados que devem ser detalhados ...
#Revisão

#não normalizado
#Normalização (Nada é redundante)

Cliente

idCliente	nome	email	telefone	endereço
1	edson	edson@gmail	981190108, 22684219	rua a, numero 60, bairro Grajau, cidade Rio de Janeiro

#Erros telefone Vetor
#Erros Endereço Possui muitos valores

1 _Forma Normal

Cliente

idCliente	nome	email
1	edson	edson@gmail

Telefone

idTelefone	Numero	id_Cliente
10	98199-0108	1
11	22684219	1

Endereço

idEndereço	rua	bairro	cidade	id_Cliente
100	rua a	grajau	Rio de Janeiro	1

2 Forma Normal

Observar os dados sempre verificando se tudo está preso a chave primária

A normalização de dados é uma série de passos que se seguem no projeto de um banco de dados, que permitem um armazenamento consistente e um eficiente acesso aos dados em bancos de dados relacionais. Esses passos reduzem a redundância de dados e as chances dos dados se tornarem inconsistentes.

No entanto, muitos SGBDs relacionais não têm separação suficiente entre o projeto lógico da base de dados e a implementação física do banco de dados, e isso tem como consequência que as consultas feitas a um banco de dados totalmente normalizado tenham mau desempenho. Nestes casos, usa-se por vezes a desnormalização para melhorar o desempenho, com o custo de menores garantias de consistência.

- Segunda Forma Normal (ou 2FN) requer que não haja dependência funcional não-trivial de um atributo que não seja a chave, em parte da chave candidata;

Segunda Forma Normal:

Uma relação está na 2FN se, e somente se, estiver na 1FN e cada atributo não-chave for dependente da chave primária inteira, isto é, cada atributo não-chave não poderá ser dependente de apenas parte da chave.

No caso de tabelas com chave primária composta, se um atributo depende apenas de uma parte da chave primária, então esse atributo deve ser colocado em outra tabela.

```
idPedido  dataPedido  codProduto  nomeProduto  qtde  valorUnitario
valorTotal
10        20/03/2015  1           celular      2     300
600
```

#Erro na Segunda Forma Normal

O nome Do Produto está preso ao código, os dados sempre presos

@ a chave primária ...

```
create database bnc5;
use bnc5;
```

```
create table produto(
    idProduto int primary key,
    nomeProduto varchar (35),
    valorUnitario double
);
```

```
insert into produto values (100, 'celular Asus', 600);
insert into produto values (101, 'celular iphone', 3000);
insert into produto values (102, 'celular samsung', 1500);
```

```
create table pedido (
    idPedido int primary key,
    dataPedido date,
    codigoProduto int,
    qtde int,
    valorTotal double ,
    foreign key(codigoProduto) references produto(idProduto));
```

```
insert into pedido values (500, '2015/03/10', 100, 1, 600);
insert into pedido values (501, '2015/03/11', 102, 3, 4500);
insert into pedido values (502, '2015/03/11', 101, 2, 3000);
```

desc produto;

Field	Type	Null	Key	Default	Extra
idProduto	int(11)	NO	PRI	NULL	
nomeProduto	varchar(35)	YES		NULL	
valorUnitario	double	YES		NULL	

desc pedido;

Field	Type	Null	Key	Default	Extra
idPedido	int(11)	NO	PRI	NULL	
dataPedido	date	YES		NULL	
codigoProduto	int(11)	YES	MUL	NULL	
qtde	int(11)	YES		NULL	
valorTotal	double	YES		NULL	

```
select*from produto;
```

idProduto	nomeProduto	valorUnitario
100	celular Asus	600
101	celular iphone	3000
102	celular samsung	1500

```
select*from pedido;
```

idPedido	dataPedido	codigoProduto	qtde	valorTotal
500	2015-03-10	100	1	600
501	2015-03-11	102	3	4500
502	2015-03-11	101	2	3000

#Observar que Geramos 2 Tabelas, porque o nome do produto estava preso

a tabela pedido, com o codigoProduto -->NomeDoProduto

NomeProduto (Fora por não estar preso ao Pedido)

#3 Forma Normal

Terceira Forma Normal:

Uma relação R está na 3FN se ela estiver na 2FN e cada atributo não-chave de R não possuir dependência transitiva, para cada chave candidata de R .

Valor Total Depende da multiplicação da qtde * o valorUnitario
Nenhuma dependencia pode ser transitiva

```
idPedido dataPedido codProduto nomeProduto qtde
valorUnitario valorTotal
10      20/03/2015  1      celular      2      300
600
```

#Cortar VALORTOTAL

#ValorTotal NA TERCEIRA FORMA NORMAL NÃO ENTROU ...

```
drop table pedido3fn;
drop table produto3fn;

create table produto3fn(
    idProduto int primary key,
    nomeProduto varchar (35),
    valorUnitario double
);

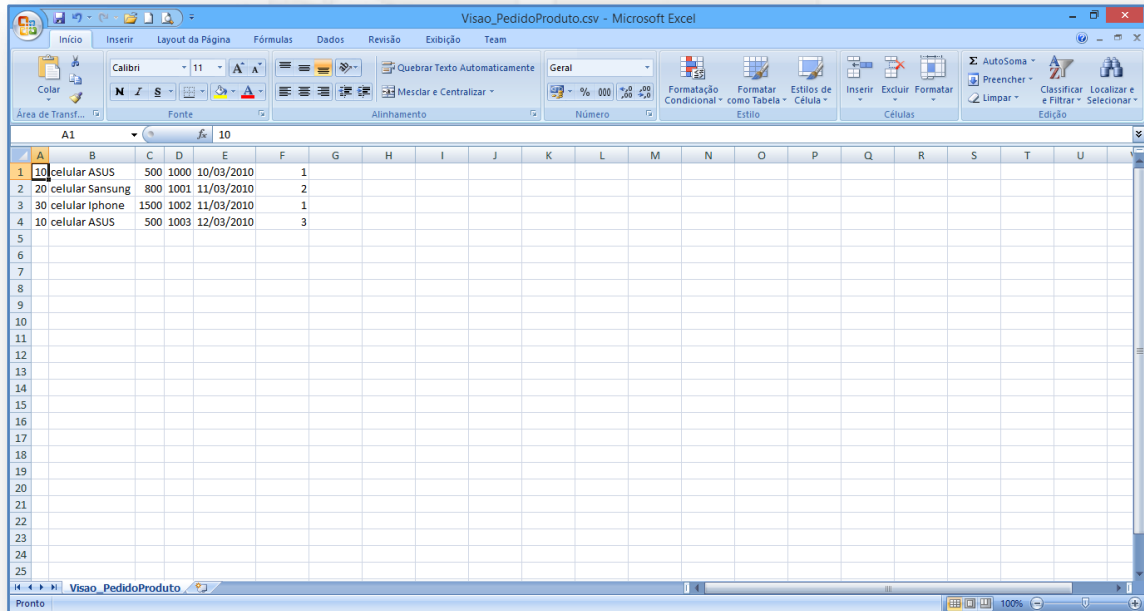
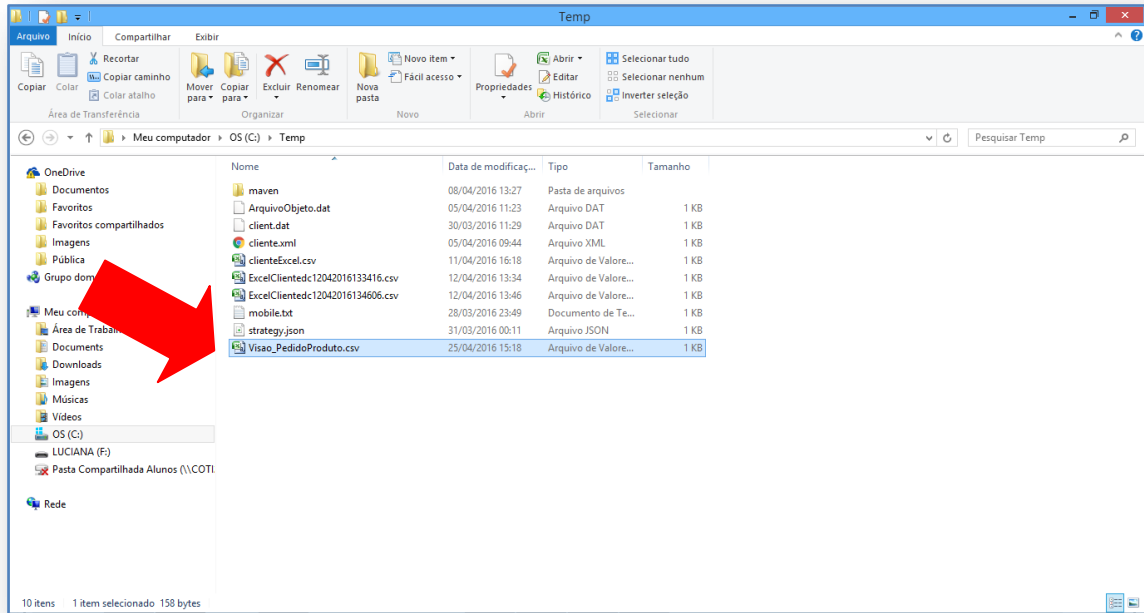
create table pedido3fn(
    idPedido int primary key,
    dataPedido date,
    codigoProduto int,
    qtde int,
    foreign key(codigoProduto) references produto3fn(idProduto)
);

insert into produto3fn values (10, 'celular ASUS', 500);
insert into produto3fn values (20, 'celular Samsung', 800);
insert into produto3fn values (30, 'celular Iphone', 1500);

insert into pedido3fn values (1000, '2010/03/10', 10, 1);
insert into pedido3fn values (1001, '2010/03/11', 20, 2);
insert into pedido3fn values (1002, '2010/03/11', 30, 1);
insert into pedido3fn values (1003, '2010/03/12', 10, 3);
commit;

create view V$Pedido_Produto as
select p.idProduto, p.nomeProduto, p.valorUnitario,
       pe.idPedido, pe.dataPedido, pe.qtde
       from produto3fn p
       inner join pedido3fn pe
       on p.idProduto = pe.codigoProduto;

select idProduto, nomeProduto, valorUnitario, idPedido,
       dataPedido, qtde
       into outfile 'c:/temp/Visao_PedidoProduto.csv'
       fields terminated by ';'
       lines terminated by '\n'
       from V$Pedido_Produto;
```



#Não ter total em tabela geralmente se aplica na 3 Forma Normal...