

BASE DE DADOS



NORMALIZAÇÃO

Teórico-Práticas
Ano Lectivo 2016/2017
Rosa Reis



Objetivos

1. Propósito da normalização.
2. Os potenciais problemas associados aos dados redundantes das relações.
3. O conceito de dependência funcional
4. As características das dependências funcionais utilizadas na normalização
5. Como identificar dependências funcionais para uma determinada relação.
6. Como as dependências funcionais identificam a chave primária para uma relação.
7. Como realizar o processo de normalização.



Propósito da Normalização

- ★ Na sequência do aparecimento do modelo relacional, e uma vez que é necessário organizar os dados de forma a que possam ser tratados relacionalmente, surge o processo designado de “**Normalização**”.
- ★ É um processo que consiste em **estruturar a informação em tabelas** na forma mais adequada afim de evitar:
 - ★ redundâncias desnecessárias
 - ★ certos problemas associados à inserção, eliminação e atualização de dados
- ★ Benefícios:
 - ★ Mais fácil para o utilizador aceder e manter os dados;
 - ★ ocupação de espaço mínimo de armazenamento



Redundância de Dados

- ✳ Objetivo principal do design de uma base de dados relacional é agrupar os atributos em relações de forma a minimizar a redundância de dados.
- ✳ A redundância está na origem de vários problemas associados a esquemas relacionais
 - ✳ mau uso do espaço de armazenamento
 - ✳ inconsistência provocada por anomalias na manipulação dos dados



Problemas da Redundância dos dados

★ **Armazenamento redundante**

- ★ Mesmos dados gravados em vários locais
- ★ Menos espaço disponível para gravar outros dados

★ **Anomalias** - Incoerências que podem existir aquando da escrita de dados

★ **Inserção**

- Pode não ser possível inserir dados, sem serem fornecidos outros, não relacionados
- Uma alternativa seria usar NULL nos outros dados, mas nem sempre é possível

★ **Atualização**

- Pode existir uma incoerência nos dados se apenas uma das cópias for atualizada

★ **Eliminação**

- Pode não ser possível apagar dados, sem apagar outros, não relacionados



Problemas da Redundância dos dados

Exemplo de redundância

Número Aluno	Nome Aluno	Sigla Curso	Número Disciplina	Nome Disciplina	Número Professor	Nome Professor	Grau Professor	Nota
21934	Antunes	INF	04	Álgebra	21	Gil	PA	15
			14	Análise	87	Ana	PC	12
			23	Estatística	43	Plínio	AS	16
42346	Bernardo	MAT	08	Topologia	32	Topo	AE	10
			04	Álgebra	21	Gil	PA	12
			12	Geometria	21	Gil	PA	18
			16	Lógica	32	Topo	AE	13
54323	Correia	EIO	04	Álgebra	21	Gil	PA	11
			08	Topologia	32	Topo	AE	10

Anomalias

- **Inserção** de um novo professor requer indicação de outros dados
- **Atualização** do grau de um professor tem de afetar várias linhas
- **Remoção** do professor Gil elimina os dados de Álgebra e Geometria



Dependências Funcionais

- ★ Do processo de normalização emergem três tipos de **dependências** entre os dados: **funcionais**, **multivalor** e de **junção**.
- ★ **As dependências funcionais** referem-se à semântica dos dados e não ao seu conteúdo.
- ★ Descrevem as relações entre os atributos



Dependências Funcionais

★ Definição:

Numa relação R , diz-se que o atributo y é funcionalmente dependente de x ($x, y \in R$), se e apenas se, em qualquer instante, cada valor de x em R tem associado apenas um valor de y em R

- ★ Uma dependência funcional para R é uma expressão da forma $R: X \rightarrow Y$, onde X e Y são conjuntos de atributos de R

Exemplo:

número de aluno \rightarrow nome de aluno

Lê-se : nome de aluno depende funcionalmente do número de aluno,
ou, número de aluno determina o nome do aluno)



Dependências Funcionais

★ Dependência funcional total

- ★ Numa relação R , o atributo y é funcionalmente dependente total de x ($x, y \in R$), no caso de x ser um atributo composto, se e apenas se, é funcionalmente dependente de x e não é funcionalmente dependente de qualquer subconjunto dos atributos de x
- ★ Dizemos que X é um atributo chave se pertence ao identificador da Entidade ou à chave primária da Relação

Exemplo:

NoFactura, Produto \rightarrow quantidade

★ Dependência funcional transitiva

- ★ Uma dependência funcional $R: x \rightarrow y$ é transitiva, se existe um atributo z que não é um subconjunto de x , tal que $x \rightarrow z$ e $z \rightarrow y$



Dependências Funcionais

★ Dependência funcional multivalor

★ Numa relação R , o atributo y tem uma dependência funcional multivalor relativamente a x ($x, y \in R$), se para cada par de tuplos de R contendo os mesmos valores de x , também existe um par de tuplos de R correspondentes à troca dos valores de y no par original

★ Uma dependência multivalor só se verifica nos casos em que a relação tem pelo menos 3 atributos

★ Os tuplos que não têm valores repetidos, satisfazem por redução esta regra

★ Consideremos a relação $R = \{a, b, c\}$

★ Existem 2 dependências multivalor

➤ $R: a \twoheadrightarrow b$

➤ $R: a \twoheadrightarrow c$

a	b	c
x1	y1	z1
x1	y1	z2
x1	y2	z1
x1	y2	z2
x3	y1	z1
x4	y3	z2



Dependências Funcionais

★ Dependência funcional de junção

★ Uma dependência de junção numa relação só existe quando, dadas algumas projecções sobre a relação, apenas é possível reconstruir a relação inicial através de algumas junções bem específicas, mas não de todas

★ Consideremos a relação $R = \{a, b, c\}$ e três projecções:

★ $P1 = \{a, b\}$, $P2 = \{a, c\}$, $P3 = \{b, c\}$

★ Se não é possível reconstruir a relação com:

➤ $P1$ e $P2$

➤ $P2$ e $P3$

➤ $P1$ e $P3$

E se for, por exemplo, apenas com $P1$, $P2$ e $P3$...

Diz-se que R possui uma dependência de junção



Propriedades das Dependências Funcionais

Considerando X , Y , e Z como conjuntos de colunas de uma relação

- ★ União

- Se $X \rightarrow Y$ e $X \rightarrow Z$, então $X \rightarrow YZ$

- ★ Decomposição

- Se $X \rightarrow YZ$, então $X \rightarrow Y$ e $X \rightarrow Z$

- ★ Transitividade

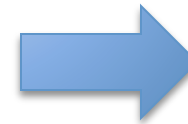
- Se $X \rightarrow Y$ e $Y \rightarrow Z$, então $X \rightarrow Z$

- ★ Reflexividade

- Se $X \supseteq Y$, então $X \rightarrow Y$

- ★ Extensão

- Se $X \rightarrow Y$, então $XZ \rightarrow YZ$



AXIOMAS
DE
ARMSTRONG

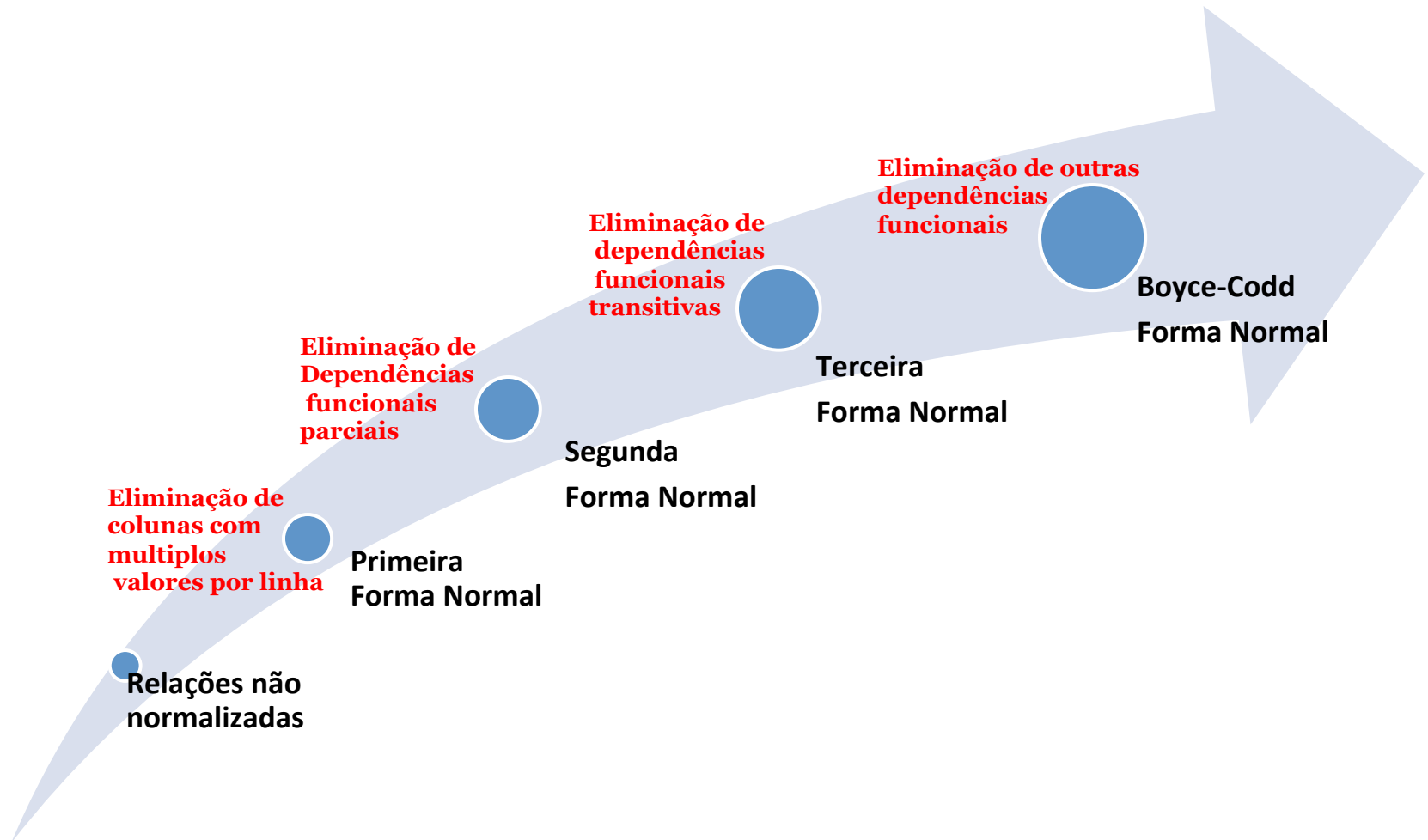


Processo de Normalização

- ★ Técnica formal para analisar uma relação com base na sua chave primária e nas dependências funcionais entre os atributos dessa relação.
- ★ Identificar um conjunto de dependências funcionais para uma relação tem como objetivo especificar o conjunto de restrições de integridade que devem ser titulares de uma relação.
- ★ Uma restrição de integridade importante a considerar em primeiro lugar é a identificação das chaves candidatas, uma das quais será selecionada para ser a chave primária para a relação.
- ★ O processo de normalização é executado a partir de uma série de passos. Cada passo corresponde a uma **forma normal**, com propriedades específicas.



Processo da Normalização





Primeira Forma Normal

- ★ Uma relação está na 1FN se:
 - Os atributos chave estão definidos
 - Não existem grupos repetitivos
 - Todos os atributos estão definidos em domínios que contêm apenas valores atômicos, isto é, cada atributo só pode admitir valores elementares e não conjunto de valores
 - Todos os atributos dependem funcionalmente da chave primária

- ★ **Visa eliminar a existência de grupos de valores repetidos**
 - A uma ocorrência da chave só pode corresponder uma ocorrência dos outros atributos não chave



Primeira Forma Normal

Suponhamos

Aluno (idAluno, nome, morada, (idDisciplina, nomeDisciplina))

Esta estrutura não se encontra na 1FN, uma vez que as colunas **idDisciplina** e **nomeDisciplina** admitem um conjunto de valores

<u>idAluno</u>	nome	morada	idDisciplina	nomeDisciplina
----------------	------	--------	--------------	----------------

Contudo existe uma forma de contornar o problema.

A3	Pedro	Rua C	D1, D2	Matemática, Economia
A4	Filipa	Rua D	D1	Matemática

Cada linha das colunas **idDisciplina** e **nomeDisciplina** pode conter um conjunto de valores! ➡ a tabela não está na 1FN



Primeira Forma Normal

<u>idAluno</u>	nome	morada	idDisciplina	nomeDisciplina
A1	João	Rua A	D1	Matemática
A1	João	Rua A	D2	Economia
A1	João	Rua A	D3	Direito
A2	Ana	Rua B	D1	Matemática
A2	Ana	Rua B	D4	Física
A3	Pedro	Rua C	D1	Matemática
A3	Pedro	Rua C	D2	Economia
A4	Filipa	Rua D	D1	Matemática

Mas esta tabela também não se encontra na primeira forma normal e apresenta problemas!

Problemas de redundância e ainda outros problemas, quais?



Primeira Forma Normal

★ Problemas de **Atualização**

- Se um aluno muda de morada, esta tem de ser alterada em várias linhas sob a pena de se gerar um estado de inconsistência

★ Problemas de **Inserção**

- Suponha que um aluno está matriculado, mas não se encontra inscrito a nenhuma cadeira, só podemos guardar as suas informações colocando um nulo nas colunas ***idDisciplina*** e ***nomeDisciplina***

★ Problemas de **Remoção**

- Se por qualquer motivo a aluna A4 anula a inscrição à disciplina de Matemática, a remoção dessa linha leva a perda de todos os restantes dados da aluna



Primeira Forma Normal

1. Identificar quais os atributos que poderão ser repetidos

idDisciplina e ***nomeDisciplina***

2. Identificar os atributos que poderão ser chave: um que identifique os atributos não repetidos e outro para identificar os atributos repetidos

chave dos atributos não repetidos: ***idAluno***

chave dos atributos repetidos: ***idDisciplina***

3. Separar os atributos repetidos dos atributos não repetidos (aplicar o critério da 1FN):
 - Converter os atributos não atômicos em atributos atômicos, para que não seja possível incluir mais do que um valor em cada atributo de uma relação
 - Separar os atributos repetidos dos não repetidos, passando a considerá-los como elementos de uma nova relação. A chave primária da relação dos atributos não repetidos deve ser acrescentada, como chave estrangeira, à chave primária dos atributos repetidos



Primeira Forma Normal

Aluno (idAluno, nome, morada)

<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D

AlunoInscrito (idAluno, idDisciplina, nomeDisciplina)

<u>Id aluno</u>	<u>idDisciplina</u>	nomeDisciplina
A1	D1	Matematica
A1	D2	Economia
A1	D3	Direito
A2	D1	Matematica
A2	D4	Fisica
A3	D1	Matematica
A3	D2	Economia
A4	D1	Matematica



Segunda Forma Normal

★ Uma relação está na 2FN se:

➤ Estiver na 1FN

➤ Cada atributo não chave depende funcionalmente da totalidade da chave

✧ Não existem dependências parciais

✓ Todos os atributos que não pertencem à chave dependem funcionalmente da chave no seu conjunto e

✓ Não dependem de nenhum dos seus elementos ou subconjuntos tomados isoladamente



Conversão da estrutura para a 2 FN

1. Se a relação só tem um atributo como chave primária e se essa relação já estiver na 1FN, então a relação também se encontra na 2FN
2. **Se a chave primária é composta e** se algum atributo não-chave depende apenas de uma parte da chave primária, então a relação deverá ser decomposta, para que cada atributo dependa da totalidade da chave primária

Exemplo 1

- A tabela **Aluno** já está na 1ª FN e a chave primária contém apenas um atributo ⇒ também está na 2ª FN!


<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D



Conversão da estrutura para a 2 FN

- ★ A tabela ***AlunoInscrito*** encontra-se na 1ª FN mas a sua chave primária é composta
- ★ Necessário decompor a tabela ***AlunoInscrito*** pois existe uma dependência funcional entre o atributo não-chave ***nomeDisciplina*** e apenas parte da chave primária, com o atributo ***idDisciplina***

$\text{idDisciplina} \rightarrow \text{nomeDisciplina}$



<u>Id aluno</u>	<u>idDisciplina</u>	nomeDisciplina
A1	D1	Matemática
A1	D2	Economia
.....



Conversão da estrutura para a 2 FN

Aluno(idAluno, nome, morada)

<u>Id aluno</u>	nome	morada
A1	João	Rua A
A2	Ana	Rua B
A3	Pedro	Rua C
A4	Filipa	Rua D

Disciplina(idDisciplina, nomeDisciplina)

<u>idDisciplina</u>	nomeDisciplina
D1	Matemática
D2	Economia
D3	Direito
D4	Física

AlunoInscrito(idAluno(FK), idDisciplina(FK))

<u>Id aluno</u>	<u>idDisciplina</u>
A1	D1
A1	D2
A1	D3
A2	D1
A2	D4
A3	D1
A3	D2
A4	D1



Terceira Forma Normal

★ Uma relação está na 3FN se:

➤ Estiver na 2FN

➤ Nenhum dos seus atributos depender funcionalmente de atributos não chave

✓ Nenhum dos atributos que não fazem parte da chave pode ser funcionalmente dependente de qualquer combinação dos restantes

✓ Cada atributo depende apenas da chave e não de qualquer outro atributo ou conjunto de atributos

EXEMPLO

★ Esta tabela não se encontra na 3FN porque o atributo não-chave **nomeCurso** depende funcionalmente do atributo **codCurso**

Aluno(idAluno, nome, codCurso, nomecurso)

<u>Id aluno</u>	nome	codCurso	nomeCurso
A1	João	01	Informática
A2	Ana	02	Civil
A3	Pedro	01	Informática
A4	Filipa	03	Química



Conversão da estrutura para a 3FN

1. Procurar dependências funcionais entre os atributos não-chave da relação
2. Se a relação que já está na 2FN e **tiver apenas um atributo não-chave**, então a relação também já se encontra na 3FN
3. Se existir algum conjunto de atributos não-chave na relação que tenha dependência funcional em relação a um outro conjunto de atributos não-chave da mesma relação, então a relação deve ser decomposta de modo a que qualquer atributo não-chave da relação só dependa da chave primária da relação



Conversão da estrutura para a 3FN

- ★ A tabela está na 2FN mas não está na 3FN
- ★ Necessário decompor a tabela **Aluno** pois existe uma dependência funcional (transitiva) entre o atributo não-chave **codCurso** e o atributo **nomeCurso**

Aluno(idAluno, nome, codCurso(FK))

<u>Id aluno</u>	nome	codCurso
A1	João	01
A2	Ana	02
A3	Pedro	01
A4	Filipa	03

Curso(codCurso, nomeCurso)

<u>codCurso</u>	nomeCurso
01	Informática
02	Civil
01	Informática
03	Química



Boyce – Codd Forma Normal

- ★ Uma relação está na forma normal de Boyce-Codd, se e apenas se, todos os seus atributos são funcionalmente dependentes da chave, de toda a chave e nada mais do que a chave.

Ou

- ★ Uma relação encontra-se na forma normal de *Boyce-Codd* se qualquer determinante é uma chave candidata.

Consideremos a relação:

$R = \{\underline{a}, b, c\}$ e as dependências funcionais em R:

$DF's = \{ (a, b) \rightarrow c \text{ e } c \rightarrow b \}$

- R está na 3ª FN, mas tem uma dependência que invalida a forma normal de Boyce-Codd

Como Resolver?



Boyce – Codd Forma Normal

★ Criando duas relações:

- $R1 = \{\underline{c}, b\}$ correspondente à dependência funcional $R: c \rightarrow b$
- $R2 = \{\underline{a}, c\}$ correspondente à dependência funcional $R: (a, b) \rightarrow c$
 - ... mas na verdade perde-se a dependência funcional $R: (a, b) \rightarrow c$, que não se encontrando explicitamente incorporada no modelo relacional teria de ser implementada no nível aplicativo !

O ideal será então

- obter uma solução que, embora mais redundante, mantém todas as dependências funcionais, ou seja, não normalizar até Boyce-Codd...

$$R = \{\underline{a}, \underline{b}, c\} \text{ e } R1 = \{\underline{c}, b\}$$



Conversão da estrutura para Boyce- Codd FN

Considere a relação A que se encontra na 3FN

$A = (\underline{\text{Id_aluno}}, \underline{\text{codDisciplina}}, \text{professor})$

e os seguintes pressupostos:

- um aluno frequenta várias disciplinas
- cada professor lecciona apenas uma disciplina
- uma disciplina pode ser leccionada por vários professores
- cada aluno em cada disciplina tem apenas um professor

<u>Id aluno</u>	<u>codDisciplina</u>	professor
A1	D1	P1
A1	D2	P2
A2	D1	P1
A3	D3	P3
A4	D3	P4



Conversão da estrutura para Boyce- Codd FN

- ✱ Existem duas chaves compostas possíveis:

id_aluno, codDisciplina e id_aluno, professor
que têm em comum o atributo id_aluno

- ✱ o atributo professor não é chave candidata mas é um determinante, pois cada professor só lecciona uma disciplina e, portanto, o professor determina a disciplina.

<u>Id_aluno</u>	<u>codDisciplina</u>	<u>professor</u>
A1	D1	P1
A1	D2	P2
A2	D1	P1
A3	D3	P3
A4	D3	P4



Conversão da estrutura para Boyce- Codd FN

- ✳ Ao eliminar o aluno A1 na disciplina D2, elimina-se o professor P2 a leccionar a disciplina D2.

Solução

Decompor a relação, separando os atributos que dependem daqueles que não são chave candidata, isto é, decompor a relação em projecções por forma a eliminar todas as dependências funcionais em que o determinante não seja chave.

<u>Id aluno</u>	<u>professor</u>
A1	P1
A1	P2
A2	P1
A3	P3
A4	P4

<u>professor</u>	<u>codDisciplina</u>
P1	D1
P2	D2
P3	D3
P4	D3



Conclusão

- ★ O nível de normalização deve ser pensado contra outros critérios
 - Por exemplo, um nível de normalização exagerado pode originar problemas de performance
- ★ A redundância entre os dados não pode ser completamente eliminada
 - de facto, as chaves estrangeiras são também uma forma de redundância
 - Problemas que a redundância pode trazer
 - ✧ Custo de espaço de armazenamento - a redundância implica ocupar espaço adicional com algo que não acrescenta nada ao que já existe armazenado
 - ✧ Manutenção - uma simples alteração ou remoção pode implicar o acesso a várias tabelas, tornando-se difícil manter a coerência dos dados armazenados
 - ✧ Desempenho - Se a redundância for significativa, isso implicará mais acessos a disco para trazer os mesmos dados



Exercício



Normalize a estrutura apresentada

Nr Factura	Data	codCliente	NomeCliente	Produto	Valor	Quantidade	Desconto
000257	01-07-2016	1234567	João Gomes	Lápis Bic	100	250	5%
000257	01-07-2016	1234567	João Gomes	Bloco de notas	1000	200	5%
000257	01-07-2016	1234567	João Gomes	Caneta	70	50	0%
000258	01-07-2016	1234568	Ana Marques	Lápis Bic	100	400	6%
000258	01-07-2016	1234568	Ana Marques	Caderno	500	350	6%
000258	01-07-2016	1234568	Ana Marques	Régua	100	20	0%