



ENTREGA III

Tema 5

Bases de Dados

Grupo 707

Mestrado Integrado em Engenharia Informática e Computação

André Daniel Gomes
António Cadilha da Cunha Bezerra
Luís Filipe Recharte

up201806224@fe.up.pt
up201806854@fe.up.pt
up201806743@fe.up.pt

Índice:

1. Contexto	2
2. Diagrama UML	3
2.1 UML Original	3
2.2 UML Revisto	4
Lista de mudanças	4
3. Esquema Relacional	5
4. Análise de Dependências Funcionais e Formas Normais	6
5. Implementação das Restrições	9
6. Lista de Interrogações	16
7. Lista de Gatilhos	17

1. Contexto

A nossa base de dados consiste num sistema de gestão de projetos de uma empresa que oferece serviços a clientes empresariais.

A empresa recebe pedidos de projetos (Request) por parte de empresas-cliente (Company), das quais se armazena o nome, morada e NIF. Estes pedidos ficam associados a um documento (Document) que pode ser de vários tipos (Type) que a empresa disponibiliza.

Sobre os tipos guarda-se uma descrição do mesmo e o estado de ativo/inativo (bem como a data da última alteração), já que os tipos de documentos elaborados pela empresa podem ser alterados ao longo do tempo face às necessidades e condições dos recursos humanos.

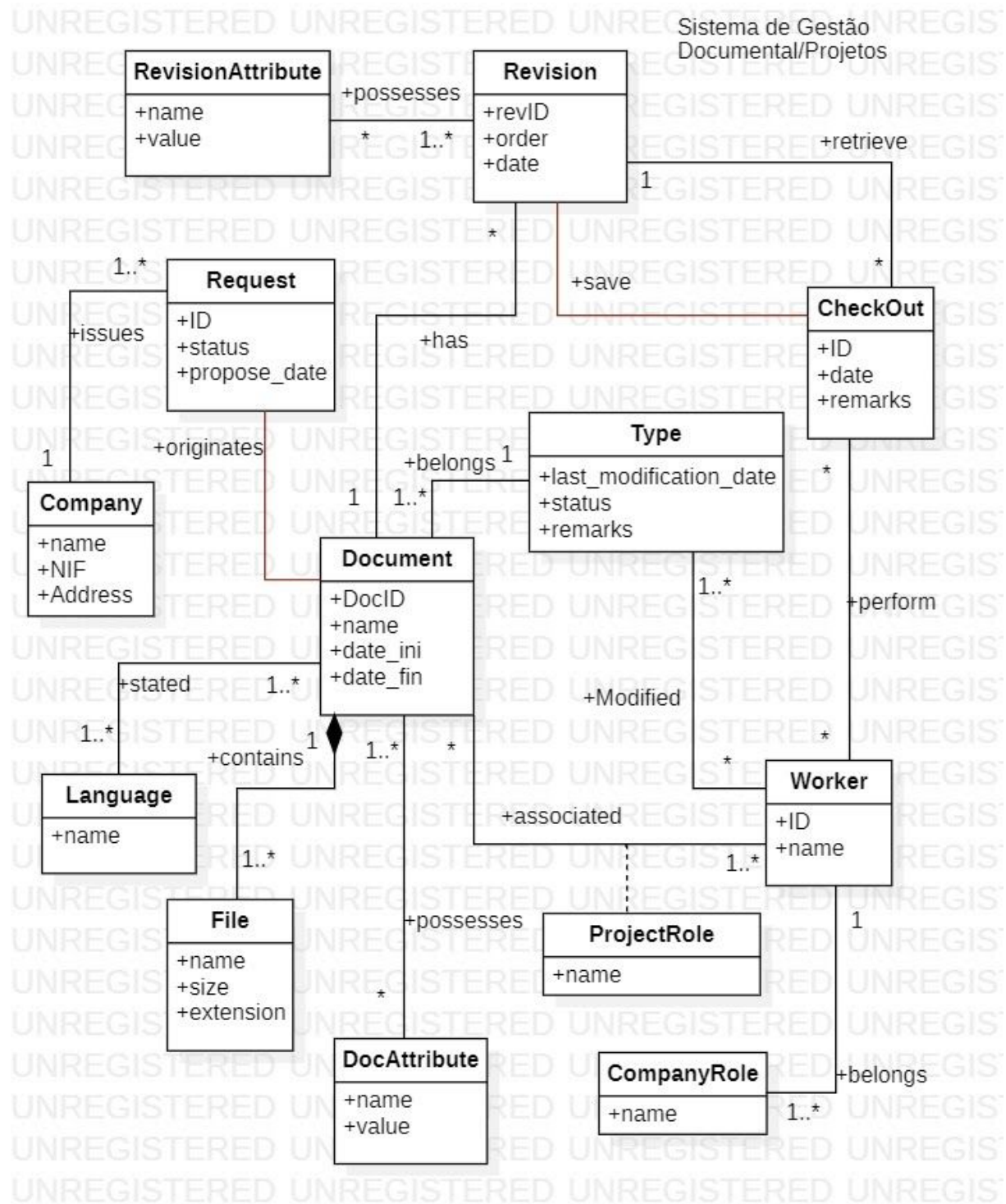
Cada documento pode conter vários ficheiros (File) de tipos e tamanho diferentes. Cada documento está associado ao(s) idioma(s) (Language) em que é desenvolvido. Sobre os documentos guardam-se ainda o nome do projeto e as datas de início e fim do projeto. Um documento pode ainda ter atributos complementares (DocAttribute).

Cada documento é propriedade de um trabalhador (Worker), sendo que pode ser este o único a trabalhar no documento ou pode fazê-lo em equipa. O papel (ProjectRole) de um trabalhador num dado projeto é determinado pela classe de associação, condicionando o tipo de acesso ao documento que cada trabalhador tem. Ao trabalhador está também associado o seu papel na empresa (CompanyRole).

O acesso ao documento (CheckOut) é efetuado através de uma revisão do documento (Revision) podendo-se aceder à última revisão do documento ou a uma revisão anterior específica. Quando se pretende guardar o trabalho efectuado num CheckOut é criada uma nova revisão. Uma revisão pode ainda ter atributos complementares (RevisionAttribute).

2. Diagrama UML

2.1 UML Original



```

classDiagram
    class Status {
        -SID
        -sDescription
    }
    class Request {
        -requestID
        -propose_date
    }
    class Company {
        -Cname
        -NIF
        -Address
    }
    class Language {
        -Lname
    }
    class File {
        -Fname
        -size
        -extension
    }
    class Revision {
        -revID
        -version
        -date
    }
    class Document {
        -docID
        -dNAME
        -date_ini
        -date_fin
    }
    class DocAttribute {
        -DName
        -DValue
    }
    class Type {
        -Tname
        -last_mod_date
        -status
        -remarks
    }
    class CheckOut {
        -CDate
    }
    class Worker {
        -WID
        -Wname
    }
    class ProjectRole {
        -PRname
    }
    class CompanyRole {
        -CRname
    }

    Status "1" -- "*" Request : sDescription
    Request "*" -- "1" Document : requestID, propose_date
    Company "1" -- "1..*" Document : Cname, NIF
    Language "1..*" -- "1..*" Document : Lname
    File "1..*" -- "1" Document : Fname, size, extension
    Revision "*" -- "1" Document : has
    Document "1..*" -- "*" Revision : revID, version, date
    Document "1..*" -- "1" Type : belongs
    Document "1..*" -- "1" Worker : owner
    Document "1..*" -- "1..*" ProjectRole : possesses
    Worker "1" -- "1" CheckOut : modified
    Worker "1..*" -- "1..*" CompanyRole : belongs
    CheckOut "1" -- "*" Revision : CheckIn
    CheckOut "1" -- "1" Document : CheckOut
    CheckOut "1" -- "1" Document : CheckOut
  
```

- Request e COInfo tornaram-se classes de associação;
- Adicionaram-se Restrições às classes Document, Request e COInfo;
- Adicionaram-se multiplicidades omitidas;
- Adicionou-se uma associação ternária entre Document, Role e Worker;
- Adicionou-se uma associação owner entre Document e Worker;

3. Esquema Relacional

Os atributos a sublinhado são **chaves primárias**.

Os atributos identificados com “->” são **chaves estrangeiras**.

- Company (CID, Cname, NIF, address)
- Document (docID, dNAME, date_ini, date_fin, type->Type, ownerID->Worker)
- Language (LID, Lname)
- File (FID, Fname, size, extension, docID -> Document)
- DocAttribute (DAID, DAname, DAvalue)
- Type (TID, Tname, last_mod_date, status, remarks)
- Worker (WID, Wname)
- CompanyRole (CRID, CRname)
- Revision (revID, version, date, docID->Document, authorID->Worker)
- RevisionAttribute (RAID, RAname, RAvalue)
- Stated (LID -> Language, docID -> Document)
- DocPossesses (DAID -> DocAttribute, docID -> Document)
- RevPossesses (RAID -> RevisionAttribute, revID -> Revision)
- Belongs (WID -> Worker, CRID -> CompanyRole)
- Modified (TID -> Type, WID -> Worker)
- Request (requestID, docID -> Document, CID -> Company, SID->Status, propose_date)
- ProjectRole (PRID, PRname)
- DocumentRoleWorker(docID->Document, WID->Worker, PRID->ProjectRole)
- CheckOut (revID -> Revision, WID -> Worker, COdate)
- Status(SID, sDescription)

4. Análise de Dependências Funcionais e Formas Normais

Company (CID, Cname, NIF, address)

CID -> Cname, NIF, address

NIF -> CID, Cname, address

CName, Address -> CID, NIF

BCNF: Não é violada

3NF: Não é violada

Document (docID, Dname, date_ini, date_fin, type -> Type, ownerID -> Worker)

docID -> Dname, date_ini, date_fin, type, ownerID

BCNF: Não é violada

3NF: Não é violada

Language (LID, Lname)

LID -> Lname

Lname -> LID

BCNF: Não é violada

3NF: Não é violada

File (FID, Fname, size, extension, docID -> Document)

FID -> Fname, size, extension, docID

Fname -> FID, size, extension, docID

BCNF: Não é violada

3NF: Não é violada

DocAttribute (DAID, DAname, DAvalue)

DAID -> DAname, DAvalue

BCNF: Não é violada

3NF: Não é violada

Type (TID, Tname, last_mod_date, status, remarks)

TID -> Tname, last_mod_date, status, remarks

Tname -> TID, last_mod_date, status, remarks

remarks -> TID, Tname, last_mod_date, status

BCNF: Não é violada

3NF: Não é violada

Worker (WID, Wname)

WID -> Wname

BCNF: Não é violada

3NF: Não é violada

CompanyRole (CRID, CRname)

CRID -> CRname

CRname -> CRID

BCNF: Não é violada

3NF: Não é violada

Revision (revID, version, date, docID -> Document, authorID -> Worker)

revID -> version, date, docID, authorID

BCNF: Não é violada

3NF: Não é violada

RevisionAttribute (RAID, RAname, RValue)

RAID -> RAname, RValue

BCNF: Não é violada

3NF: Não é violada

Stated (LID -> Language, docID -> Document)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

DocPossesses (DAID -> DocAttribute, docID -> Document)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

RevPossesses (RAID -> RevisionAttribute, revID -> Revision)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

Belongs (WID -> Worker, CRID -> CompanyRole)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

Modified (TID -> Type, WID -> Worker)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

Request (requestID, docID -> Document, CID -> Company, SID->Status, propose_date)

requestID -> docID, CID, SID, propose_date

docID -> CID, SID, propose_date

BCNF: Não é violada

3NF: Não é violada

ProjectRole (PRID, PRname)

PRID -> PRname

PRname -> PRID

BCNF: Não é violada

3NF: Não é violada

DocumentRoleWorker(docID -> Document, WID -> Worker, PRID -> ProjectRole)

Sem dependências funcionais não triviais.

BCNF: Não é violada

3NF: Não é violada

CheckOut (revID -> Revision, WID -> Worker, COdate)

revID -> WID, COdate

BCNF: Não é violada

3NF: Não é violada

Status(SID, sDescription)

SID -> sDescription

sDescription -> SID

BCNF: Não é violada

3NF: Não é violada

O critério para que uma relação, R, esteja conforme a Boyce-Codd Normal Form (BCNF) é de que para todas as suas dependências funcionais não triviais $A \rightarrow B$, A seja uma superkey, ou seja, que A^+ contenha todos os atributos de R.

Como a partir de todos os lados esquerdos das dependências identificadas foi possível determinar todos os atributos das tabelas, todas as tabelas estão conforme a BCNF. Como a BCNF é um subset da 3ª forma normal (3NF), conclui-se também que todas as tabelas respeitam a 3NF.

5. Implementação das Restrições

Relação	Atributos	Restrição	Implementação
Company	CID	Cada empresa tem um identificador único	PRIMARY KEY
	Cname, address	Todas as empresas têm nome e morada.	NOT NULL
	NIF	Não existem empresas com o mesmo NIF.	UNIQUE
Document	docID	Cada documento tem um identificador único.	PRIMARY KEY
	dNAME, date_ini	Todos os documentos têm um nome e data de início.	NOT NULL
	date_ini, date_fin	Quando definida, a data de finalização do documento tem que ser posterior à data de início.	CHECK (date_ini < date_fin OR date_fin = NULL)
	type, ownerID	Cada documento está associado a um tipo e tem um proprietário.	FOREIGN KEY
	type	Não se pode apagar um tipo ao qual estejam associados documentos.	ON DELETE RESTRICT
	ownerID	Se o proprietário de um documento for eliminado, o documento persiste com o campo a NULL.	ON DELETE SET NULL

Language	LID	Cada language tem um identificador único.	PRIMARY KEY
	Lname	Todas as linguagens têm um nome único.	UNIQUE NOT NULL
File	FID	Cada ficheiro tem um identificador único.	PRIMARY KEY
	Fname	Cada ficheiro tem um caminho único e não pode ser nulo.	UNIQUE NOT NULL
	docID	Todos os ficheiros estão associados a um documento.	NOT NULL FOREIGN KEY
	docID	Caso um documento seja removido, os ficheiros associados são removidos.	ON DELETE CASCADE
	size	O tamanho de um ficheiro é não negativo.	NOT NULL CHECK (size >=0)
DocAttribute	DAID	Cada DocAttribute tem um identificador único.	PRIMARY KEY
	DAname, DValue	Cada DocAttribute tem nome e valor únicos.	UNIQUE NOT NULL
Type	TID	Cada type tem um identificador único.	PRIMARY KEY
	Tname, status	Cada type tem nome e estado	NOT NULL
	remarks	Cada type tem uma descrição única e não nula.	UNIQUE NOT NULL

	status	O estado de um type é 1 para ativo ou 0 para inativo.	CHECK (status=0 OR status=1)
Worker	WID	Cada trabalhador tem um identificador único.	PRIMARY KEY
	Wname	Todos os trabalhadores têm nome.	NOT NULL
CompanyRole	CRID	Cada cargo tem um identificador único.	PRIMARY KEY
	CRname	Todos os cargos têm nome único e não nulo.	UNIQUE NOT NULL
Revision	revID	Cada revisão tem um identificador único.	PRIMARY KEY
	date	Todas as revisões têm uma data.	NOT NULL
	docID, authorID	Todas as revisões estão associadas a um documento e a um autor.	NOT NULL FOREIGN KEY
	docID	Se for eliminado um documento, as revisões associadas também o são.	ON DELETE CASCADE
	authorID	Se for eliminado o autor de uma revisão, esta deixa de estar associada a um autor.	ON DELETE SET NULL
RevisionAttribute	RAID	Cada RevisionAttribute tem um identificador único.	PRIMARY KEY

	RAname, RAvalue	Cada RevisionAttribute tem nome e valor únicos.	UNIQUE NOT NULL
Stated	LID, docID	Associação entre Language(LID) e Document(docID).	PRIMARY KEY FOREIGN KEY
	docID	Se o documento for apagado, a associação também o é.	ON DELETE CASCADE
	LID	Não se podem remover linguagens referenciadas.	ON DELETE RESTRICT
DocPossesses	DAID, docID	Associação entre DocAttribute e Document.	PRIMARY KEY FOREIGN KEY
	docID	Se o documento for apagado, a associação também o é.	ON DELETE CASCADE
	DAID	Não se podem remover DocAttribute referenciados.	ON DELETE RESTRICT
RevPossesses	RAID, revID	Associação entre RevisionAttribute e Revision.	PRIMARY KEY FOREIGN KEY
	revID	Se a revisão for apagada, a associação também o é.	ON DELETE CASCADE
	RAID	Não se podem remover RevisionAttribute referenciados.	ON DELETE RESTRICT
Belongs	WID, CRID	Associação entre Worker e Company Role.	PRIMARY KEY FOREIGN KEY

	WID	Se o trabalhador for apagado, a associação também o é.	ON DELETE CASCADE
	CRID	Não se podem remover cargos referenciados.	ON DELETE RESTRICT
Modified	TID, WID	Associação entre Type e Worker.	PRIMARY KEY FOREIGN KEY
	WID	Se o trabalhador for apagado, a associação também o é.	ON DELETE CASCADE
	TID	Não se podem remover tipos referenciados.	ON DELETE RESTRICT
Request	requestID	Cada Request tem um identificador único.	PRIMARY KEY
	docID	Cada pedido está associado a um documento e é identificado por este.	UNIQUE FOREIGN KEY
	CID, SID	Cada pedido está associado a uma empresa e tem um estado.	NOT NULL FOREIGN KEY
	docID	Se for eliminado um documento, o pedido associado também o é.	ON DELETE CASCADE
	CID, SID	Não se pode apagar uma empresa ou um estado a que um pedido esteja associado.	ON DELETE RESTRICT

	propose_date	Todos os pedidos têm uma data de proposta.	NOT NULL
ProjectRole	PRID	Cada cargo tem um identificador único.	PRIMARY KEY
	PRname	Cada cargo tem um nome único.	UNIQUE NOT NULL
DocumentRoleWorker	docID, WID, PRID	Associação entre Document, Worker e ProjectRole	PRIMARY KEY FOREIGN KEY
	docID, WID	Se for eliminado um documento ou trabalhador, a associação também o é.	ON DELETE CASCADE
	PRID	Não se podem remover cargos referenciados.	ON DELETE RESTRICT
CheckOut	revID	Um check-out só pode ser feito uma vez por revisão e é identificado pela revisão associada. Se a revisão for apagada, o check-out também.	PRIMARY KEY FOREIGN KEY ON DELETE CASCADE
	COdate	Cada check-out tem que ter uma data.	NOT NULL
	WID	Cada check-out tem que ser feito por um trabalhador. Se este for removido, o check-out persiste com o campo a NULL.	FOREIGN KEY ON DELETE SET NULL

Status	SID	Cada status tem um identificador próprio.	PRIMARY KEY
	sDescription	Cada status tem uma descrição única.	UNIQUE NOT NULL

Adicionalmente, a todas as constraints do tipo **FOREIGN KEY** foi adicionada a constraint **ON UPDATE CASCADE**, pois em caso de alteração da chave, esta deve-se refletir nos sítios onde é referida.

6. Lista de Interrogações

1. Quantas revisões foram feitas aos documentos de cada empresa ?
2. Que trabalhador foi mais vezes reviewer de documentos?
3. Qual o número de CheckOuts efetuados por cada trabalhador em documentos da empresa 'Super Bock Group', por ordem descendente?
4. Com quantos ficheiros do tipo pdf é que a empresa já trabalhou?
5. Quais os trabalhadores que não estiveram envolvidos no documento 'Efacec Rebranding'?
6. Quais os trabalhadores que estiveram envolvidos em todos os documentos?
7. Qual é o trabalhador com mais CheckOuts registados? (sem usar MAX())
8. Quais são os documentos que estiveram a ser desenvolvidos em simultâneo com outro documento (ou seja, entre a data de início e de fim de outro)?
9. Que Company Roles estão mais vezes associados a cada tipo de documento?
10. Quais os pares de trabalhadores que já trabalharam no mesmo documento?

7. Lista de Gatilhos

Foram criados os seguintes gatilhos:

1. Ao inserir um novo tuplo em Request, criar e associar um documento vazio correspondente.
 - Após uma inserção em Request, é inserido um novo tuplo em Document com valores a NULL e é atualizada a chave estrangeira docID do novo Request para apontar para o novo Document.
 - Para complementar este gatilho, poderiam ser criados outros para a atualização de valores nestas relações como, por exemplo, um gatilho que removesse um Document se o Status correspondente ao Request que o originou fosse alterado para um estado de rejeição da proposta.
2. Ao alterar Status ou Remarks de Type -> atualizar last_mod_date
 - Caso haja uma atualização na tabela Type, nos atributos Status ou Remarks sem ser especificada a data da modificação, essa data passa a ser a do momento da atualização.
3. Só deixar um Worker criar uma nova revisão se tiver feito CheckOut da última revisão anteriormente.
 - Caso se insira uma revisão a um Document ao qual o Worker associado a essa revisão não tenha efetuado checkout da última revisão do documento, essa inserção é ignorada.