



UNIVERSIDADE
LUSÓFONA

Base de Dados

SQL: Integridade Referencial

2022/2023

Sumário

Correção Trabalho de Casa

- VIEW
- INDEX

Hoje

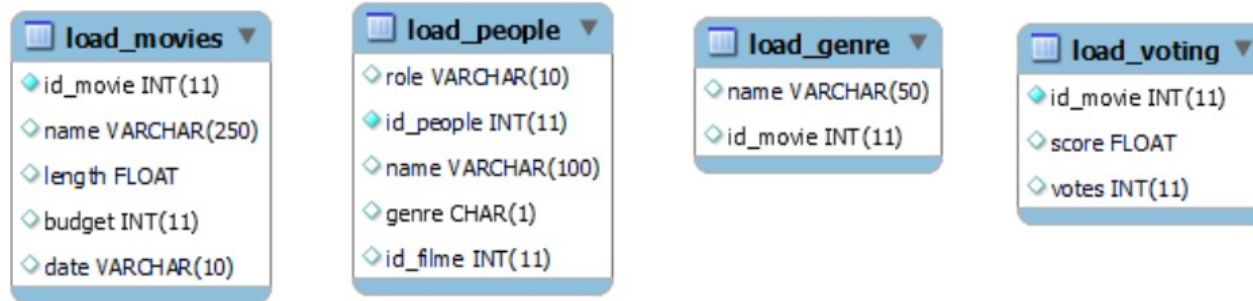
- Restrições integridade referencial
- ForeignKeys

A blue ribbon graphic with a 3D effect, featuring a dark blue shadow on the left side. The ribbon is horizontal and contains the text "Trabalho de Casa Aula 8" in white.

Trabalho de Casa Aula 8

Trabalho de Casa

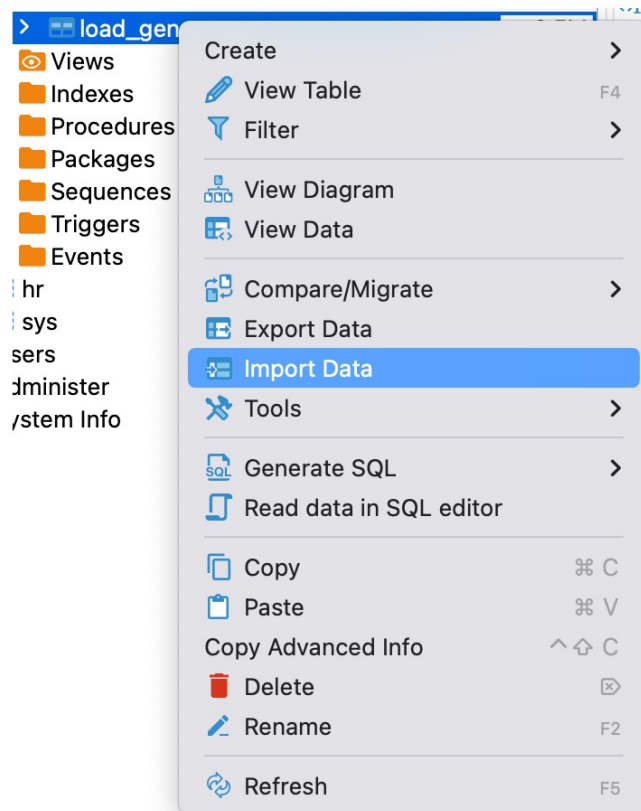
1. Comece por criar a base de dados deisiflix e as tabelas load_movies, load_people, load_genres e load_voting correspondentes conforme o esquema apresentado no enunciado.



```
CREATE TABLE load_genre (  
    name VARCHAR(50),  
    id_movie INT  
)
```

Trabalho de Casa (2)

2. Carregue os dados utilizando a função import data no dbeaver carregando cada ficheiro de dados na tabela respetiva e mapeando as respetivas colunas. Confirme o correto carregamento dos dados.



Trabalho de Casa (3)

3. Os sistemas de Gestão de Base de Dados oferecem ferramentas de profiling que permitem medir a performance de execucao das queries. Neste trabalho iremos usar o profiler do MySQL para medir o tempo de execução de uma query antes e depois da criação de um índice de forma a verificar o ganho de performance. Active o profiler no MySQL usando a instrução para alterar a flag e iniciar a medição de performance nas queries seguintes:

```
SET profiling=1;  
flush status;
```

4. Para testar a performance vamos usar a query

```
SELECT sql_no_cache *  
FROM load_genres lg  
WHERE id_movie >400000  
ORDER BY id_movie DESC LIMIT 5;
```

nota: sql_no_cache permite garantir que não há otimização pelo MariaDB através da utilização de resultados desta query em cache

5. Crie um índice sobre o atributo movie_id
6. Execute novamente a mesma query
7. Utilize o comando:

```
SHOW profiles;
```

Verifique o tempo de execução da query antes e depois do indice de forma a observar a melhoria que o índice introduziu.

Trabalho de Casa (4)

```
CREATE DATABASE deisiflix;
```

```
USE deisiflix;
```

```
⊖ CREATE TABLE load_genres (  
    name varchar(50),  
    id_movie int  
);
```

```
⊖ #profiling using SHOW PROFILES  
    set profiling=1;  
    drop index load_genres_ix1 ON load_genres;
```

```
⊖ #execute queries  
    flush status;  
    select sql_no_cache * from load_genres lg where id_movie >400000 order by id_movie desc limit 5;
```





```
⊖ #add indexes  
    create index load_genres_ix1 on load_genres(id_movie);
```

```
⊖ #execute queries  
    select sql_no_cache * from load_genres lg where id_movie >400000 order by id_movie desc limit 5;
```

```
⊖ #show execution times  
    show profiles;
```



Trabalho de Casa (5)

	123 Query_ID 	123 Duration 	ABC Query 
1	94	0.00021886	set @@SQL_SELECT_LIMIT=DEFAULT
2	95	0.00040986	SELECT DATABASE()
3	96	0.00024475	set @@SQL_SELECT_LIMIT=200
4	97	0.06162414	select sql_no_cache * from load_genres lg where id_r
5	98	0.00028015	set @@SQL_SELECT_LIMIT=DEFAULT
6	99	0.00023901	SELECT DATABASE()
7	100	0.00023264	set @@SQL_SELECT_LIMIT=200
8	101	0.19504124	#add indexes¶create index load_genres_ix1 on load_g
9	102	0.00019662	set @@SQL_SELECT_LIMIT=DEFAULT
10	103	0.00021885	SELECT DATABASE()
11	104	0.0004981	set @@SQL_SELECT_LIMIT=200
12	105	0.00099024	#execute queries¶select sql_no_cache * from load_ge
13	106	0.00019418	set @@SQL_SELECT_LIMIT=DEFAULT
14	107	0.00016589	SELECT DATABASE()
15	108	0.0001863	set @@SQL_SELECT_LIMIT=200



SQL

Processo de Criação de uma BD Relacional

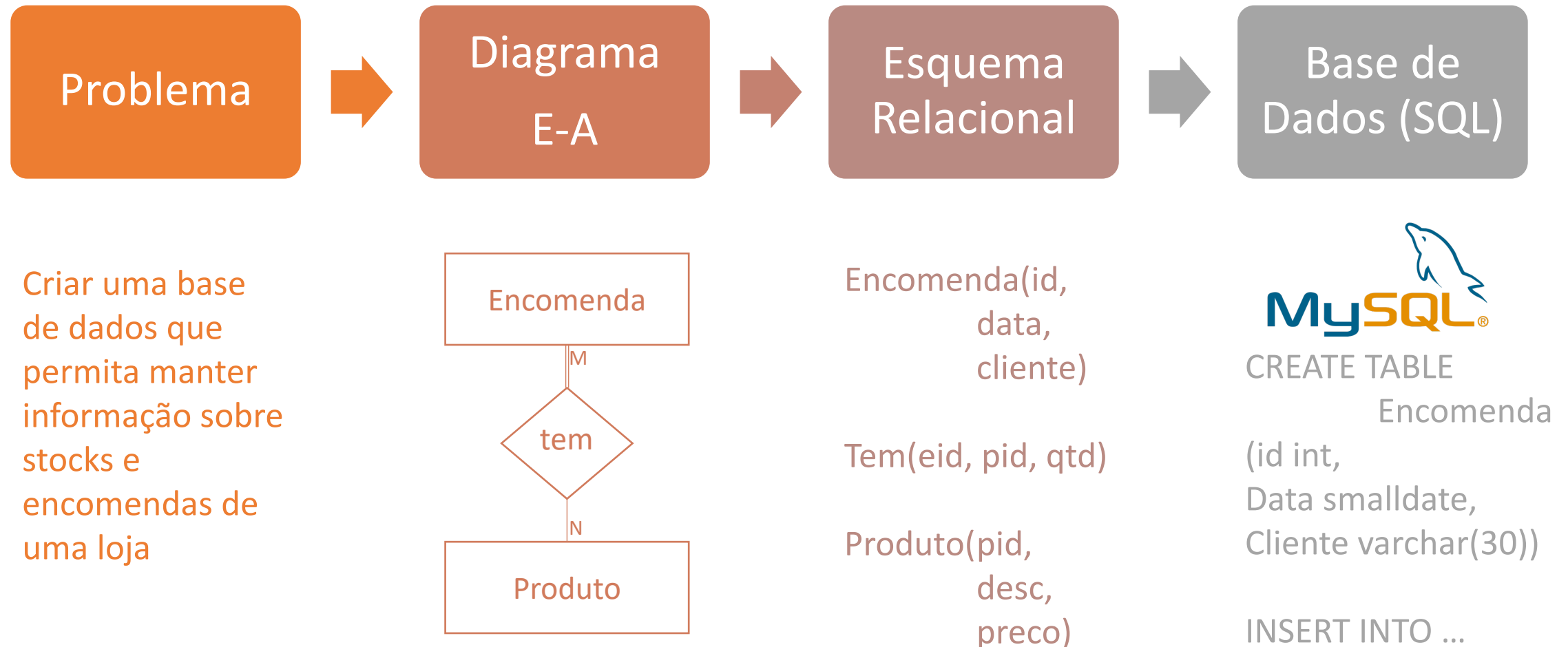
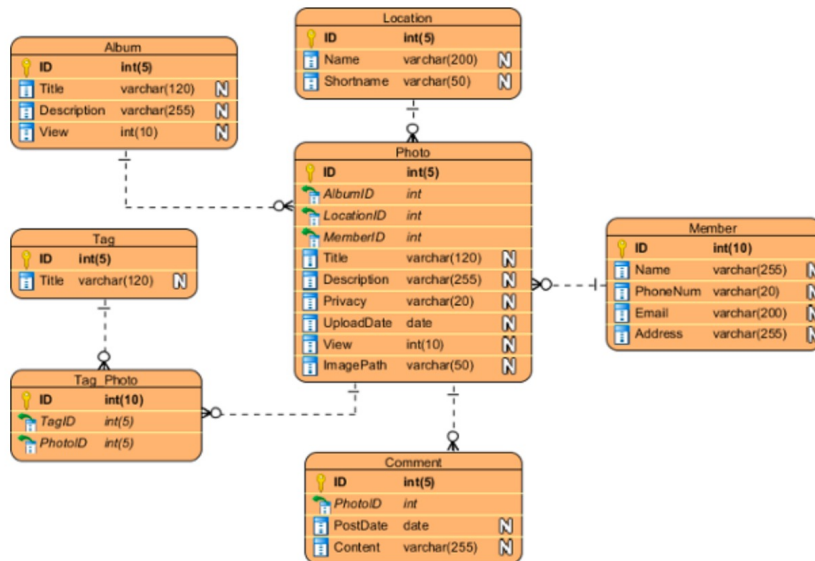
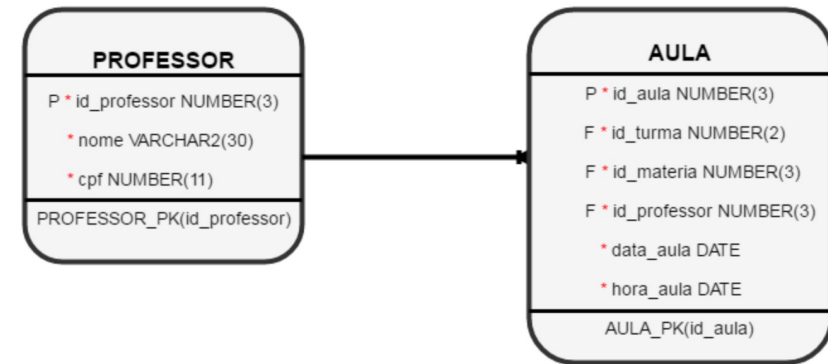


Diagrama conceptual e físico



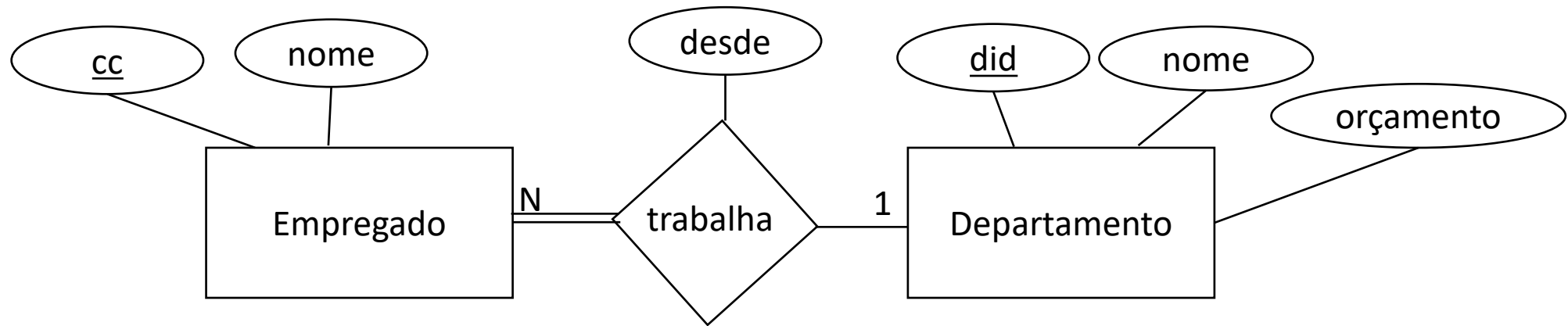
```

CREATE TABLE turma (
  idturma INTEGER(4) NOT NULL AUTO_INCREMENT,
  capacidade INTEGER(2) NOT NULL,
  idProfessor INTEGER(4) NOT NULL,
  PRIMARY KEY (idturma),
  FOREIGN KEY (idProfessor) REFERENCES professor(idProfessor),
  UNIQUE KEY idturma (idturma)
)
  
```

```

CREATE TABLE professor (
  idProfessor INTEGER(4) NOT NULL AUTO_INCREMENT,
  telefone INTEGER(10) NOT NULL,
  nome CHAR(80) COLLATE NOT NULL DEFAULT '',
  PRIMARY KEY (idProfessor),
  FOREIGN KEY(idTurma) REFERENCES turma(idturma),
  UNIQUE KEY idProfessor (idProfessor)
)
  
```

Representação de entidades

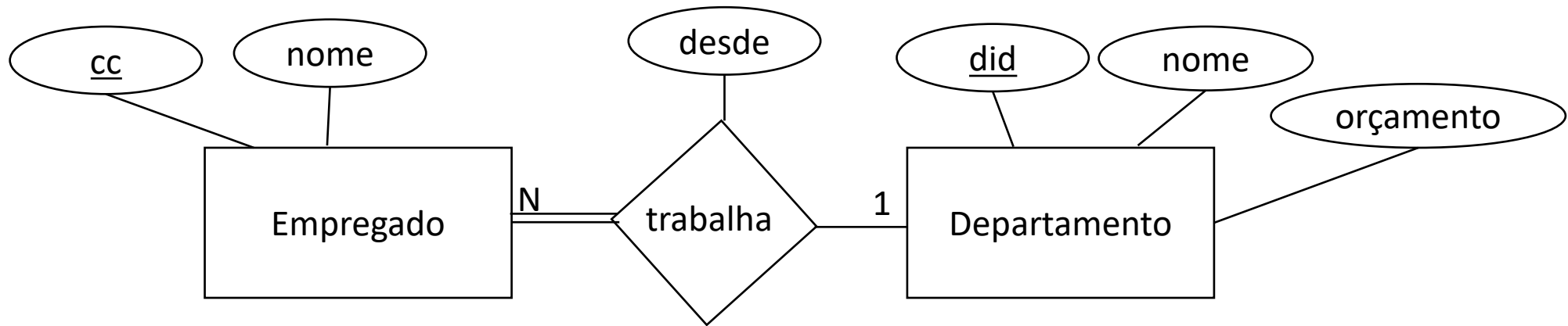


- Uma entidade é representada por um esquema com os mesmos atributos

empregado(cc, nome)

departamento(did, nome, orçamento)

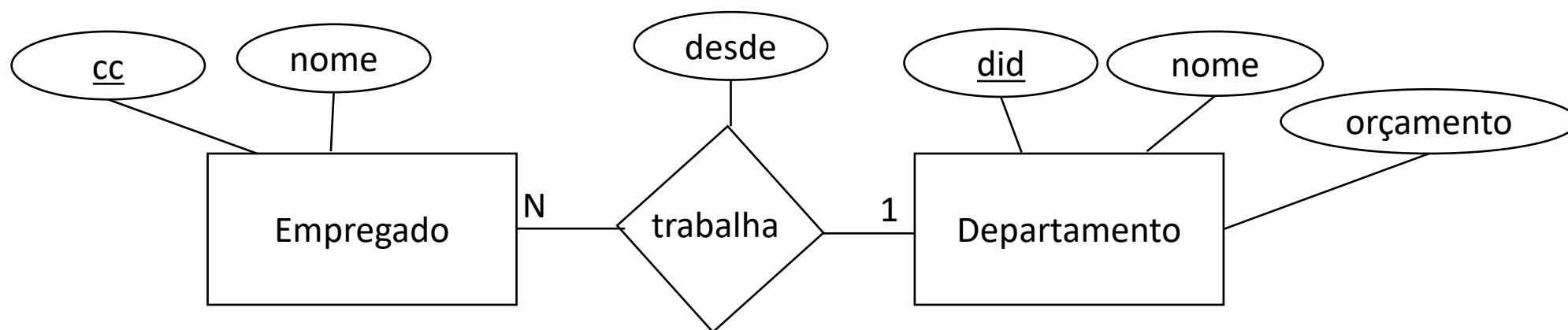
Associações 1:N com obrigatoriedade



- Associações 1:N/N:1 com obrigatoriedade de participação do lado N representa-se adicionando um atributo extra ao lado N com a chave primária do lado 1

empregado(cc, nome, did, desde)
did: FK(departamento)

Representação associações 1:N/N:1/1:1

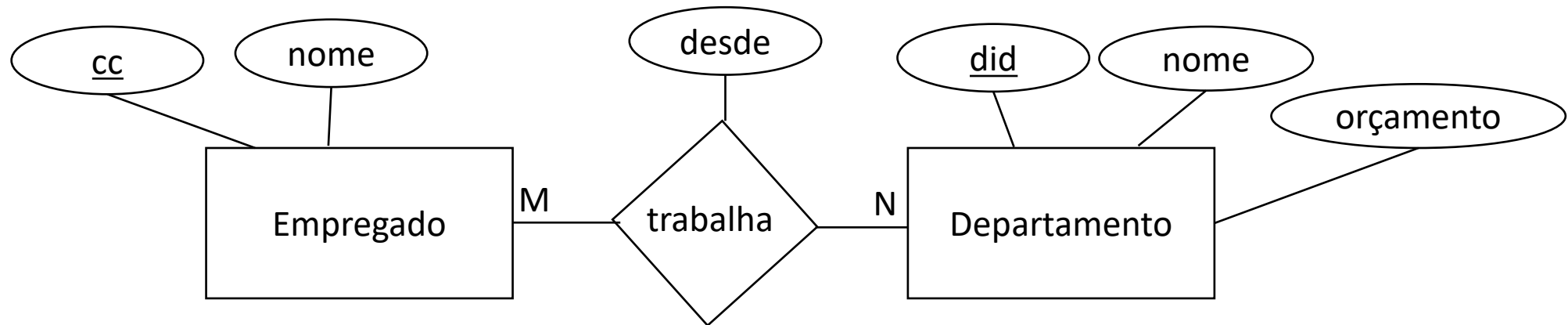


- Representa-se utilizando como chave primária o atributo chave primária do lado “muitos” (N)

trabalha(cc, did, desde)
cc: FK(empregado)
did: FK(departamento)

- Em associações um para um (1:1) pode-se escolher a chave de qualquer um dos lados

Representação de associações M:N



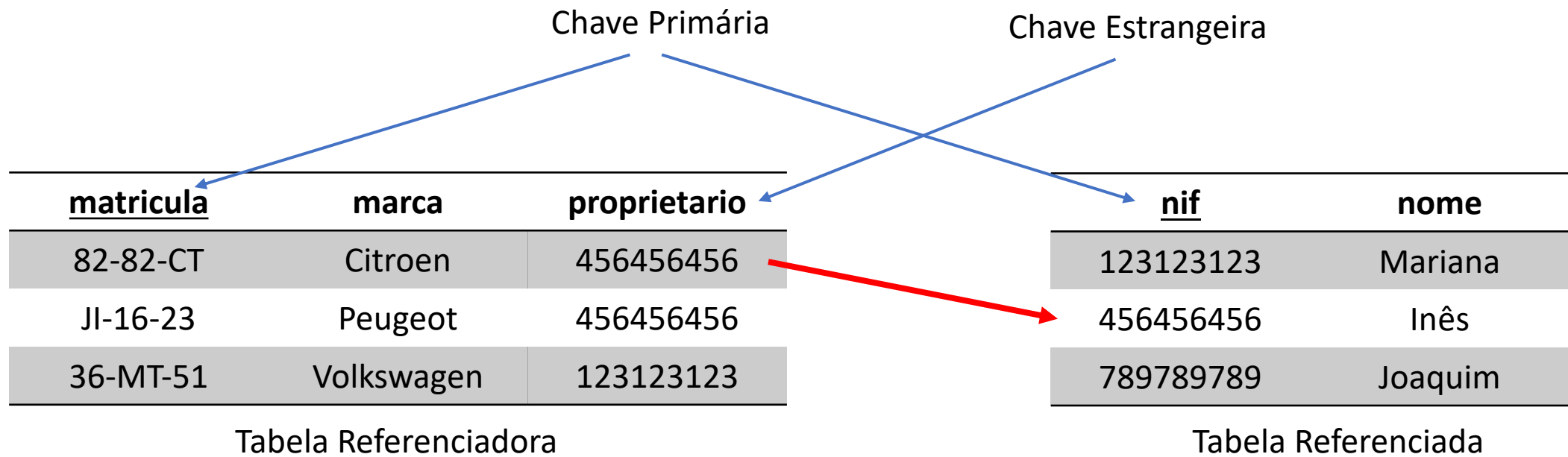
- Representado por um esquema constituído por atributos para as chaves primárias das duas entidades participantes, acrescido dos restantes atributos descritivos

trabalha(cc, did, desde)
cc: FK(empregado)
did: FK(departamento)

Integridade Referencial

- Uma chave estrangeira numa tabela referencia uma chave noutra tabela
- Não podem existir ocorrências de chave estrangeiras para a qual não exista ocorrência idêntica da chave candidata da tabela que a referencia

Chave estrangeira



Chave estrangeira

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

Integridade Referencial (on delete restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

Integridade Referencial (on delete restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

O que acontece quando apagamos ('123123123', Mariana)?

Integridade Referencial (on delete restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

O que acontece quando apagamos
(‘123123123’, Mariana)?



**SGBD emite erro e impede eliminação
uma vez que se encontra referenciada
na tabela viaturas**

ON DELETE RESTRICT
(comportamento por defeito)

Integridade Referencial

- Sintaxe

```
FOREIGN KEY (col_name)  
REFERENCES tbl_name (col_name)  
[ON DELETE reference_option]  
[ON UPDATE reference_option]
```

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

Integridade Referencial (on delete cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE CASCADE  
)
```

Integridade Referencial (on delete cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE CASCADE  
)
```

O que acontece quando apagamos ('123123123', Mariana)?

Integridade Referencial (on delete cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE CASCADE  
)
```

O que acontece quando apagamos ('123123123', Mariana)?



Registos na tabela viatura com referencia a '123123123' são também eliminadas

ON DELETE CASCADE
(eliminação em cascata)

Integridade Referencial (on delete cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456

<u>nif</u>	nome
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE CASCADE  
)
```

O que acontece quando apagamos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são também
eliminadas

ON DELETE CASCADE
(eliminação em cascata)

Integridade Referencial (on delete set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE SET NULL  
)
```

Integridade Referencial (on delete set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE SET NULL  
)
```

O que acontece quando apagamos ('123123123', Mariana)?

Integridade Referencial (on delete set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE SET NULL  
)
```

O que acontece quando apagamos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são
atualizados com valor NULL

ON DELETE SET NULL
(atribuição de NULL)

Integridade Referencial (on delete set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	NULL

<u>nif</u>	nome
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON DELETE SET NULL  
)
```

O que acontece quando apagamos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são
atualizados com valor NULL

ON DELETE SET NULL
(atribuição de NULL)

Integridade Referencial (on update restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

Integridade Referencial (on update restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

O que acontece quando alteramos ('123123123', Mariana)?

Integridade Referencial (on update restrict)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
)
```

O que acontece quando alteramos
(‘123123123’, Mariana)?



**SGBD emite erro e impede atualização
uma vez que se encontra referenciada
na tabela viaturas**

ON UPDATE RESTRICT
(comportamento por defeito)

Integridade Referencial (on update cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE CASCADE  
)
```

Integridade Referencial (on update cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE CASCADE  
)
```

O que acontece quando alteramos
(‘123123123’, Mariana)?

Integridade Referencial (on update cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
321321321	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE CASCADE  
)
```

O que acontece quando alteramos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são
atualizados com mesmo valor

ON UPDATE CASCADE
(atualização em cascata)

Integridade Referencial (on update cascade)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	321321321

<u>nif</u>	nome
321321321	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE CASCADE  
)
```

O que acontece quando alteramos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são
atualizados com mesmo valor

ON UPDATE CASCADE
(atualização em cascata)

Integridade Referencial (on update set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	123123123

<u>nif</u>	nome
123123123	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE SET NULL  
)
```

Integridade Referencial (on update set null)

<u>matricula</u>	marca	proprietario
82-82-CT	Citroen	456456456
JI-16-23	Peugeot	456456456
36-MT-51	Volkswagen	NULL

<u>nif</u>	nome
321321321	Mariana
456456456	Inês
789789789	Joaquim

```
CREATE TABLE viatura (  
    matricula CHAR(8),  
    marca VARCHAR(15),  
    proprietario CHAR(9),  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
    ON UPDATE SET NULL  
)
```

O que acontece quando alteramos
(‘123123123’, Mariana)?



Registos na tabela viatura com
referencia a ‘123123123’ são
atualizados com valor NULL

ON UPDATE SET NULL
(atribuição NULL)

Integridade referencial (ALTER TABLE)

- Podemos adicionar ou remover restrições de integridade referencial em relações existentes utilizando a cláusula ALTER TABLE

```
ALTER TABLE viatura
```

```
ADD CONSTRAINT fk_proprietario_pessoa FOREIGN KEY (proprietario) REFERENCES pessoa(nif)  
ON UPDATE SET NULL;
```

```
ALTER TABLE viatura
```

```
DROP CONSTRAINT fk_proprietario_pessoa;
```


Exercícios – GitHub aula09

- Exercícios publicados no repositório git:
<https://github.com/ULHT-BD/aula09>

aula07

Nas aulas anteriores utilizámos a linguagem SQL para efetuar consultas de dados, recuperando dados e efetuando operações sobre estes de forma a obter a informação desejada. Nesta aula olhamos para a cláusula CREATE e respetica sintaxe, um comando que nos permite criar novas relações de acordo com o esquema definido. Bom trabalho!

0. Requisitos

1. CREATE TABLE

2. Tipos de Dados

3. Restrições de Integridade

A blue ribbon graphic with a 3D effect, featuring a dark blue shadow on the left side. The word "Obrigado." is written in white text on the main blue surface.

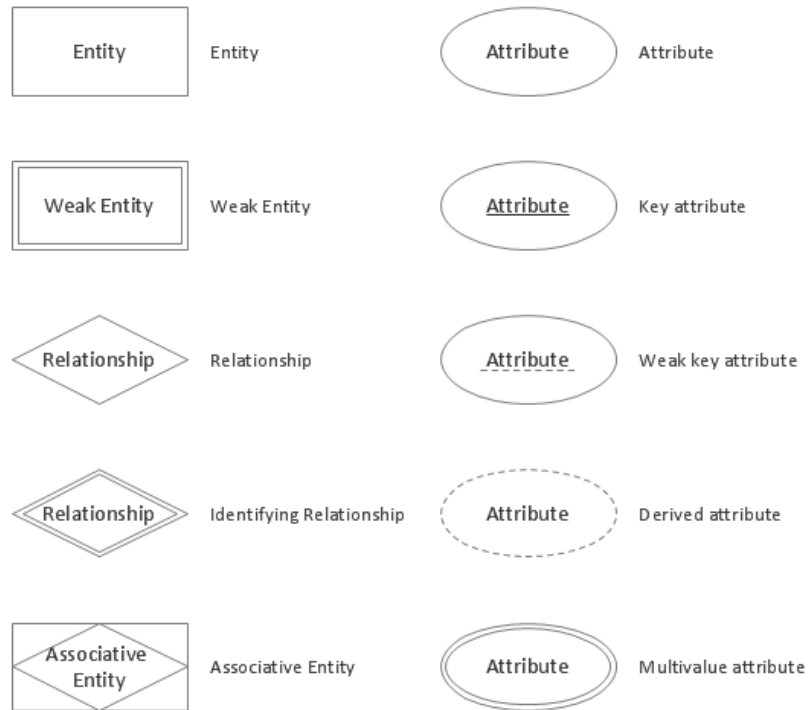
Obrigado.

Material Extra

Apoio: Resumo sobre diferentes notações

Chen Notation

Chen ERD



Participations

Cardinality can be shown or hidden

Mandatory

<u>1</u>	<u>1</u>	(0:1)
<u>1</u>	<u>1</u>	(1:1)
<u>1</u>	<u>N</u>	(0:N)
<u>1</u>	<u>N</u>	(1:N)
<u>1</u>	<u>M</u>	(0:M)
<u>1</u>	<u>M</u>	(1:M)

Optional

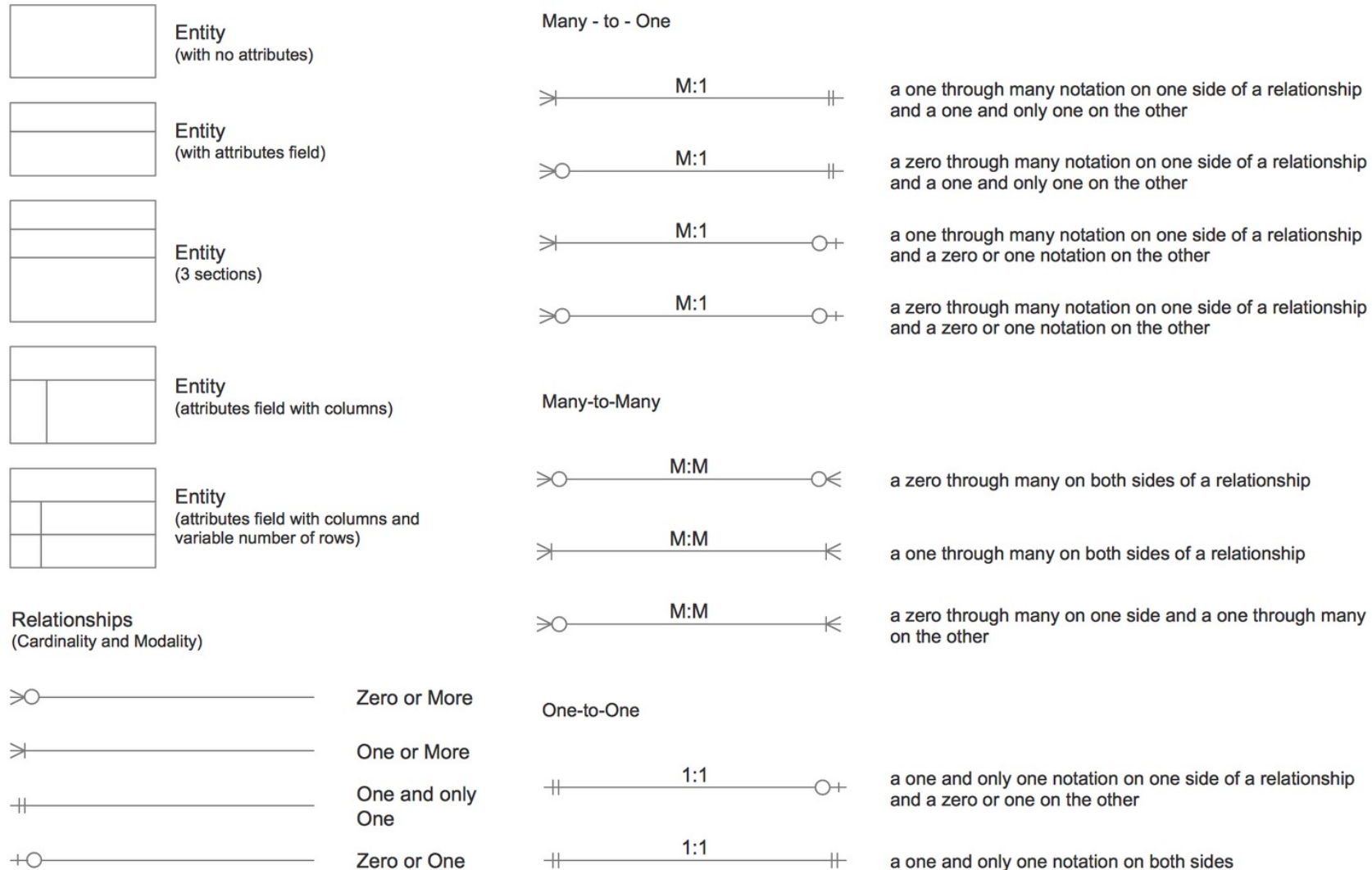
.....11	(0:1)
.....11	(1:1)
.....1N	(0:N)
.....1N	(1:N)
.....1M	(0:M)
.....1M	(1:M)

Recursive Relationship

Cardinality can be shown or hidden

<u>1</u>	<u>1</u>	(0:1)
<u>1</u>	<u>1</u>	(1:1)
<u>1</u>	<u>N</u>	(0:N)
<u>1</u>	<u>N</u>	(1:N)
<u>1</u>	<u>M</u>	(0:M)
<u>1</u>	<u>M</u>	(1:M)

Crow's Foot Notation



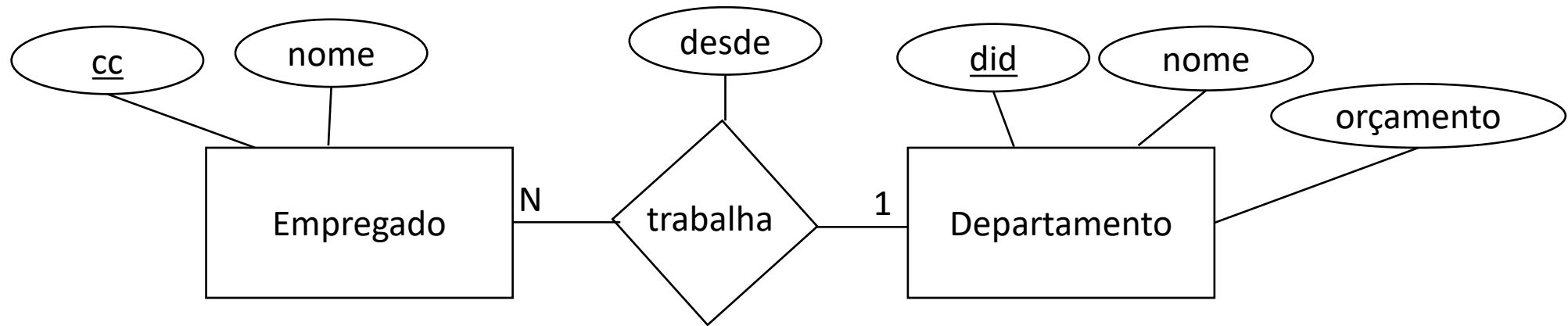
Material Extra

Apoio: Conversão E-A no Esquema Relacional

Conversão de E-A para Esquema Relacional

- Uma BD definida num diagrama EA pode ser representada por uma coleção de esquemas relacionais
- Para cada entidade e associação existe um esquema ao qual se atribui o nome da entidade ou associação correspondente
- Cada esquema tem um número de colunas (geralmente correspondente aos atributos) com nomes únicos

Representação de entidades

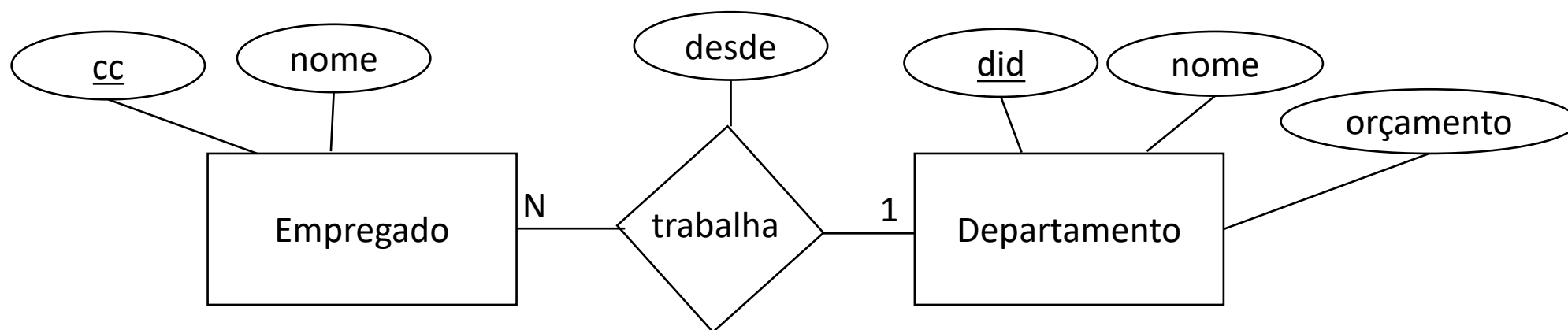


- Uma entidade é representada por um esquema com os mesmos atributos

empregado(cc, nome)

departamento(did, nome, orçamento)

Representação associações 1:N/N:1/1:1

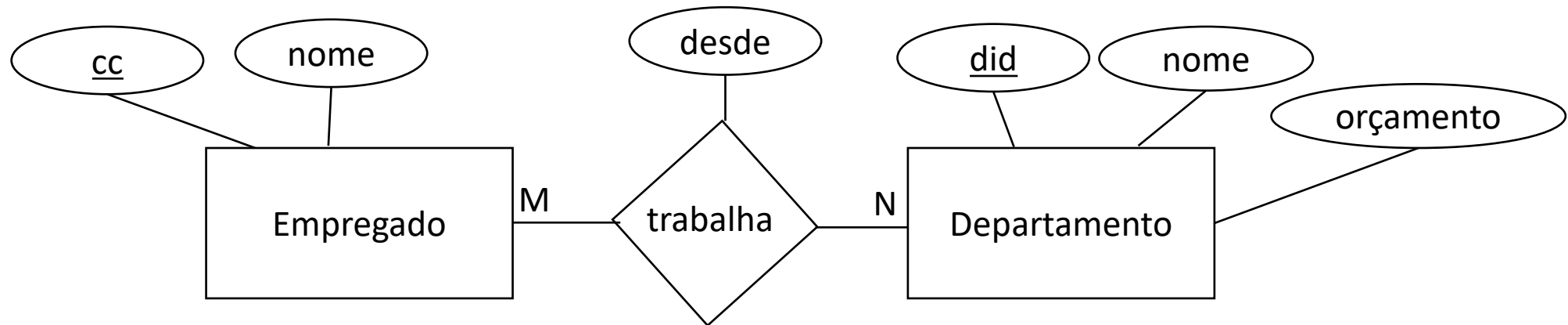


- Representa-se utilizando como chave primária o atributo chave primária do lado “muitos” (N)

trabalha(cc, did, desde)
cc: FK(empregado)
did: FK(departamento)

- Em associações um para um (1:1) pode-se escolher a chave de qualquer um dos lados

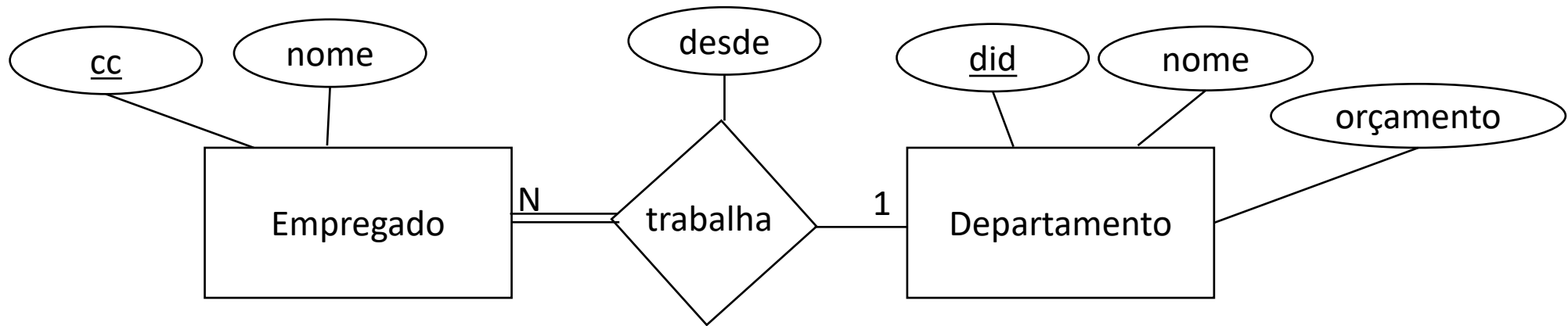
Representação de associações M:N



- Representado por um esquema constituído por atributos para as chaves primárias das duas entidades participantes, acrescido dos restantes atributos descritivos

trabalha(cc, did, desde)
cc: FK(empregado)
did: FK(departamento)

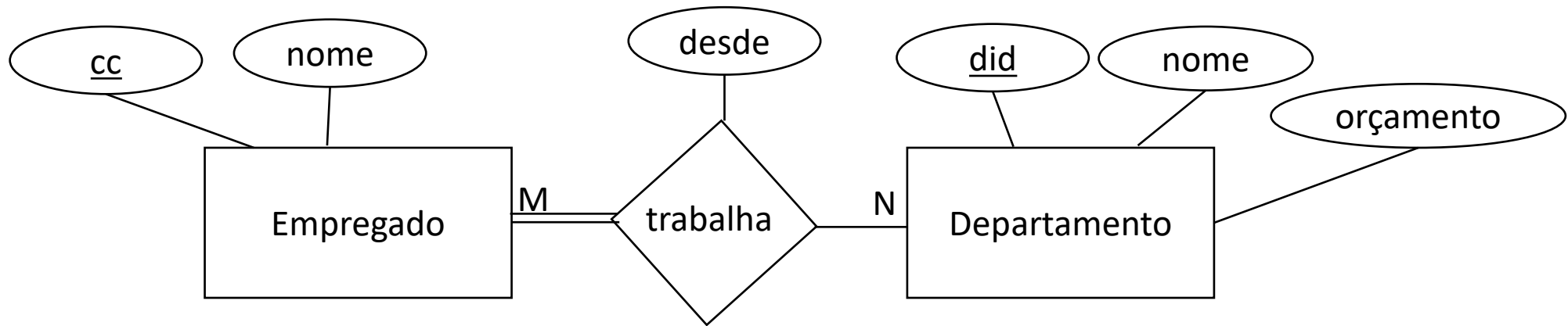
Associações 1:N com obrigatoriedade



- Associações 1:N/N:1 com obrigatoriedade de participação do lado N representa-se adicionando um atributo extra ao lado N com a chave primária do lado 1

empregado(cc, nome, did, desde)
did: FK(departamento)

Associações M:N com obrigatoriedade



- Não é possível representar: necessário usar uma restrição de integridade genérica

empregado(cc, nome)

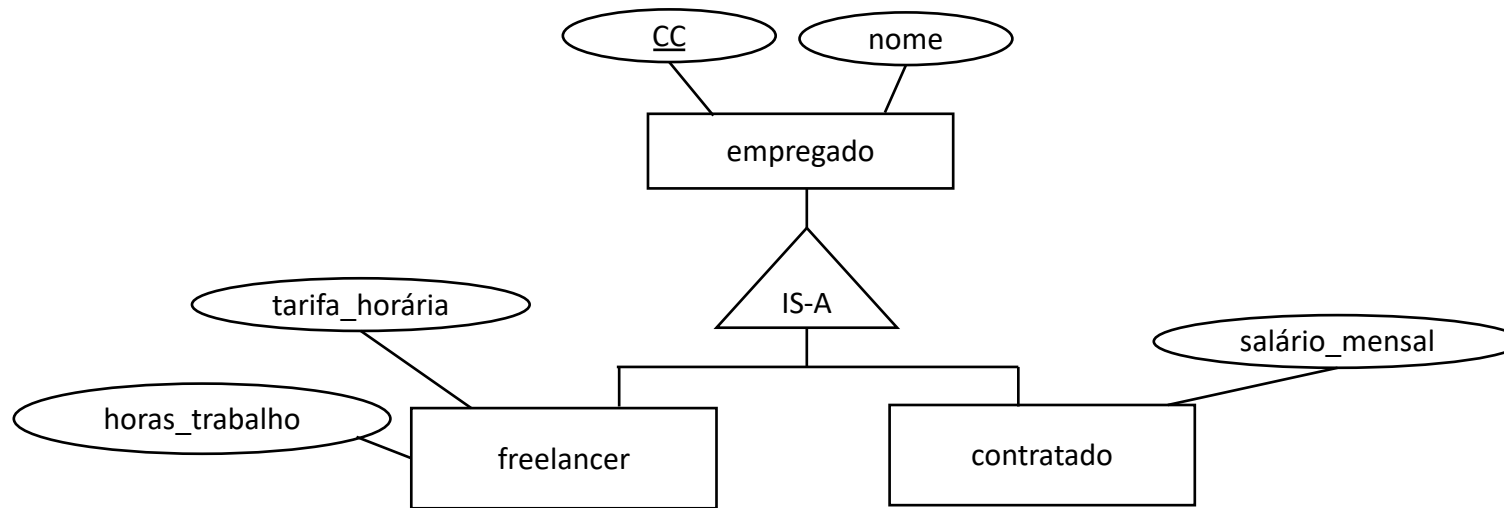
IC-1: todo o cc em empregado tem de estar presente em trabalha

trabalha(cc, did, desde)

cc: FK(empregado)

did: FK(departamento)

Especialização



Empregado(cc, nome)

Freelancer(cc, horas_trabalho, tarifa_horario)

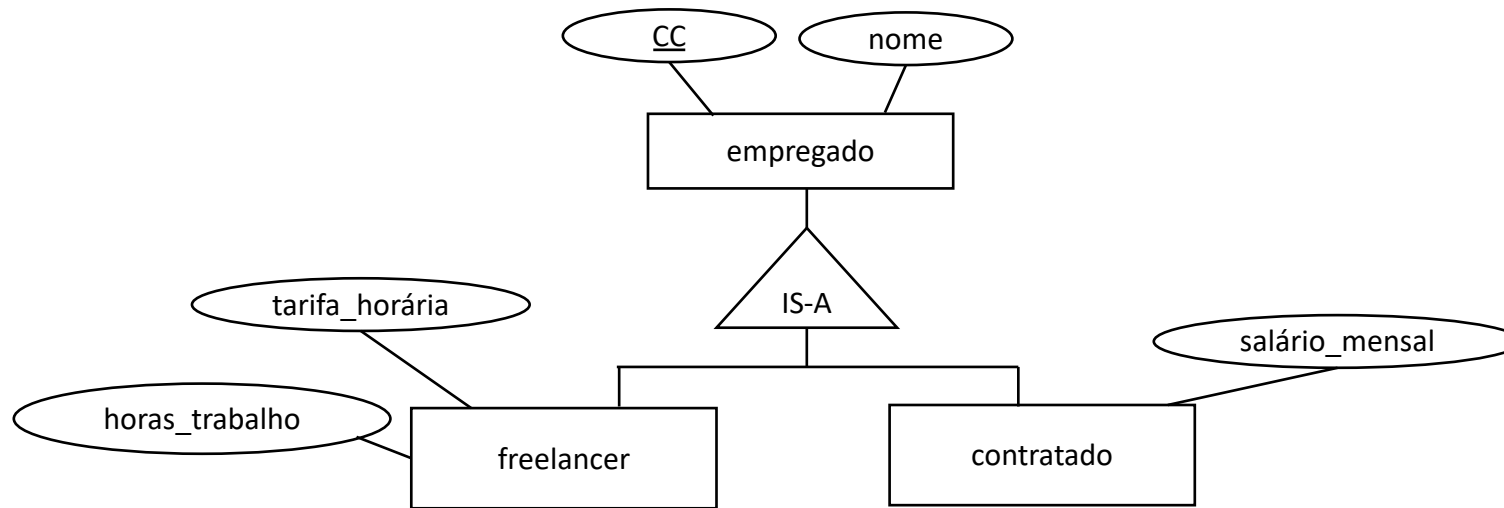
cc: FK(empregado)

Contratado(cc, salario_mensal)

cc: FK(empregado)

Inconveniente: obter informação sobre um empregado implica aceder a duas relações

Especialização (alternativa)



Empregado(cc, nome)

Freelancer(cc, nome, horas_trabalho, tarifa_horario)

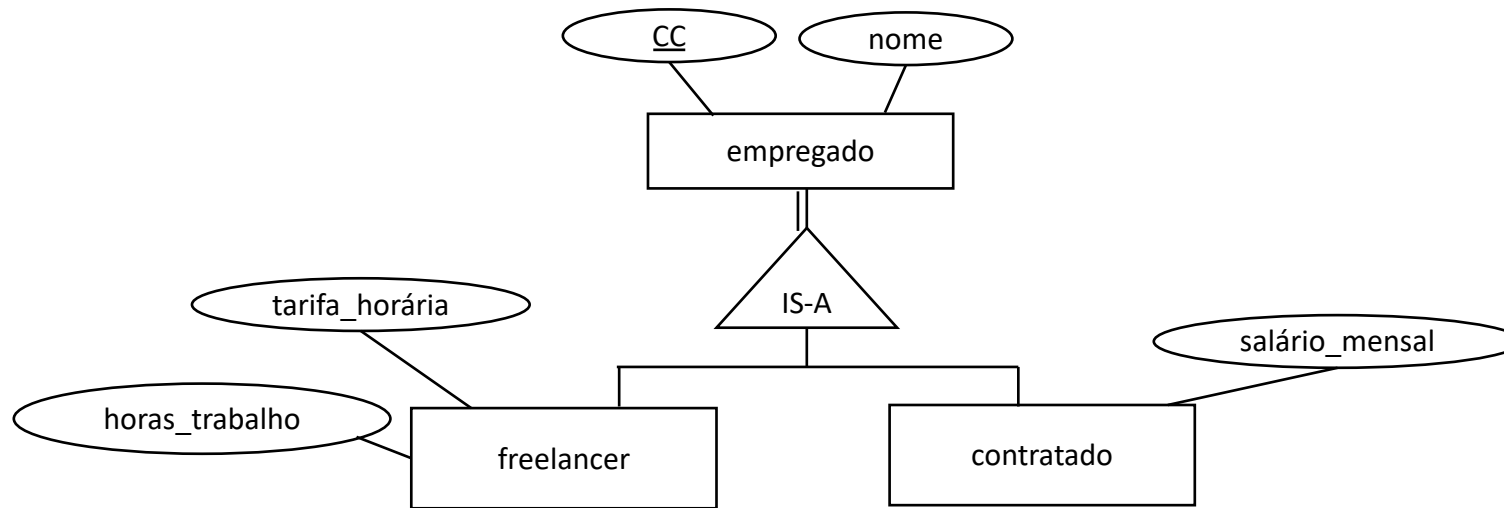
cc: FK(empregado)

Contratado(cc, nome, salario_mensal)

cc: FK(empregado)

Inconveniente: redundancia de dados
(problemas discutidos mais tarde)

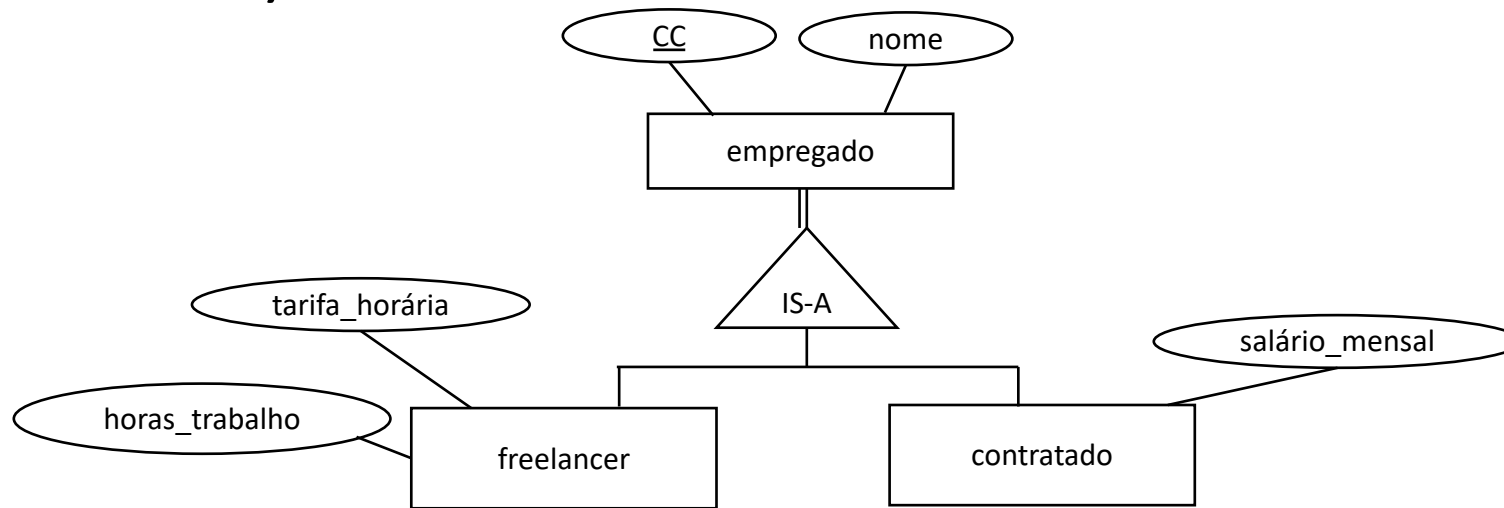
Especialização com obrigatoriedade



empregado(cc, nome)
freelancer(cc, nome, horas_trabalho, tarifa_horario)
cc: FK(empregado)
contratado(cc, nome, salario_mensal)
cc: FK(empregado)

IC-1: todos os cc devem aparecer em pelo menos num de freelancer e contratado

Especialização com obrigatoriedade (alternativa)

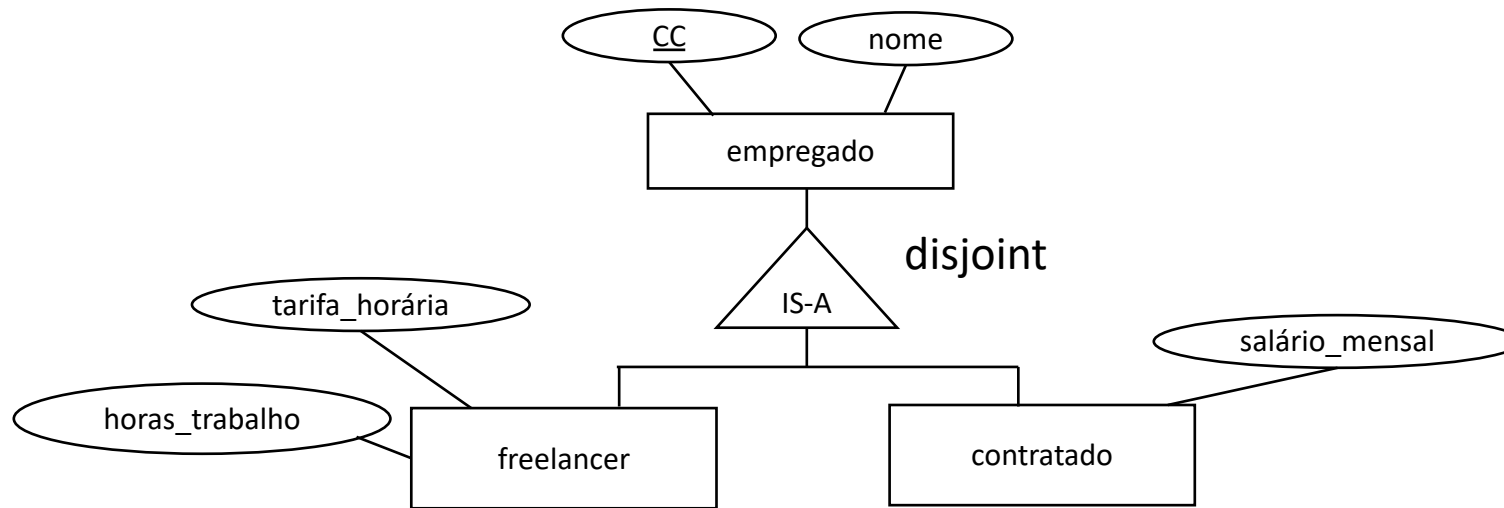


Freelancer(<u>cc</u>, nome, horas_trabalho, tarifa_horario)

Contratado(<u>cc</u>, nome, salario_mensal)

- Possível porque todas as entidades em empregado estão em freelancer ou contratado

Especialização disjunta



empregado(cc, nome)
freelancer(cc, horas_trabalho, tarifa_horario)
cc: FK(empregado)
contratado(cc, salario_mensal)
cc: FK(empregado)

IC-1: cc pode aparecer em freelancer ou contratado mas não em ambas