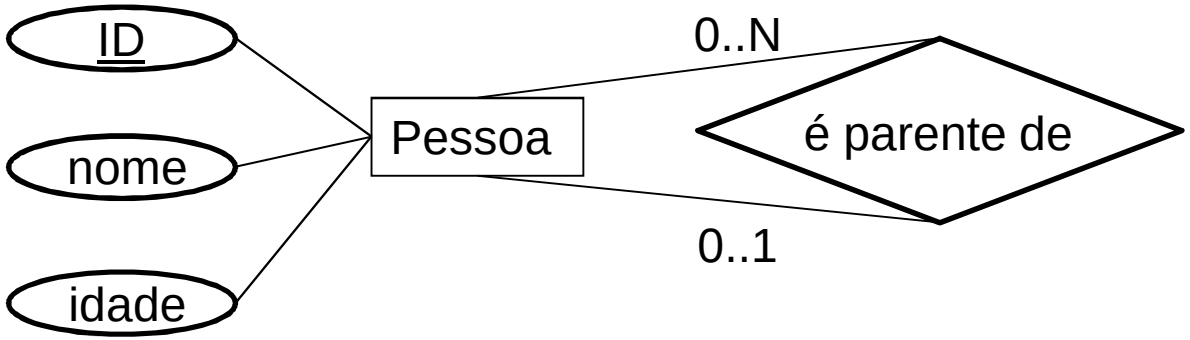
Curso Superior de Tecnologia em Banco de Dados

Linguagem SQL

RESOLUÇÃO
SIMULADO P1

SIMULADO

Observem o seguinte modelo:

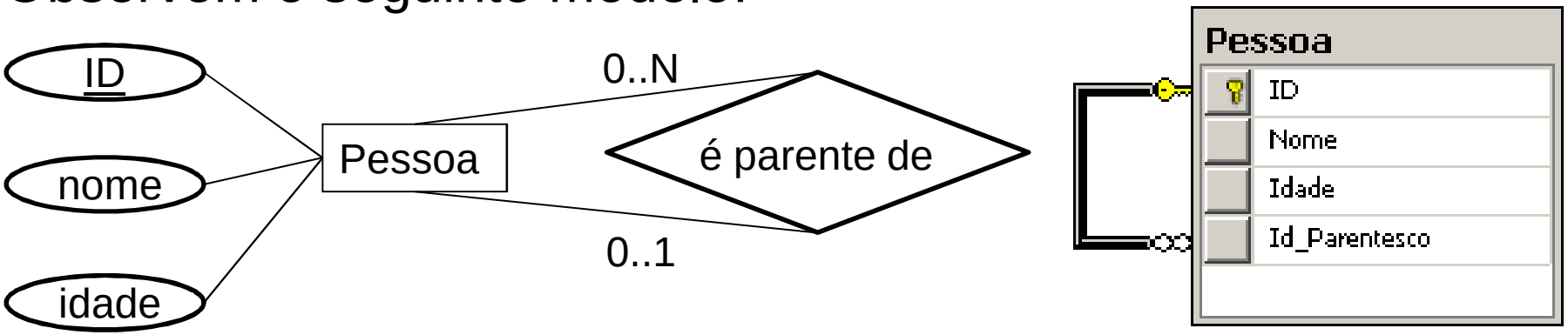


Ou seja, uma pessoa com id, nome e idade pode ser parente se outras N pessoas (auto-relacionamento com integridade referencial).

1) Escreva o modelo físico de banco de dados referente a este modelo.(modelo)

SIMULADO

Observem o seguinte modelo:



2) Escreva, e execute, os comandos para criar este modelo em SQL. (create table)

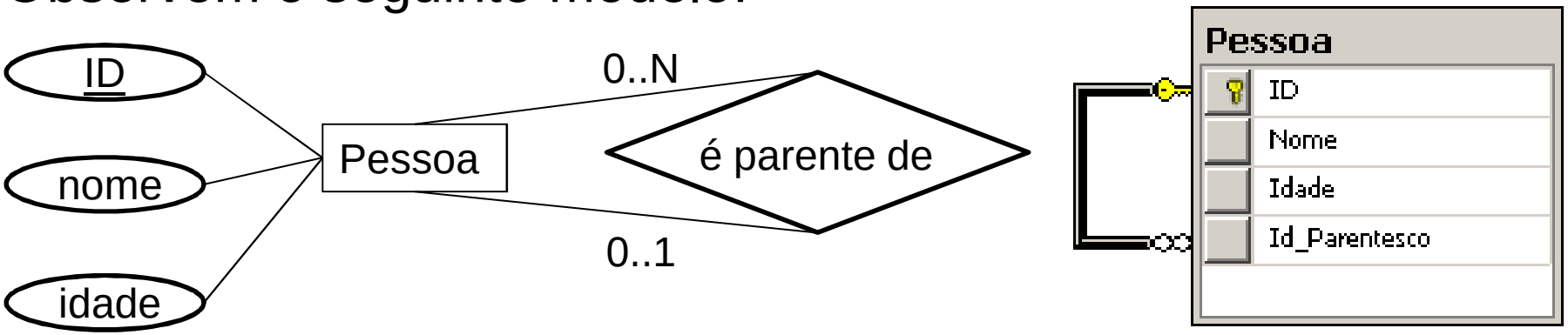
Os nomes não aceitam valores nulos e a idade só aceita valores positivos (idade > 0)

```
CREATE TABLE Pessoa
(   Id INT NOT NULL PRIMARY KEY IDENTITY(1,1)
    , Nome NVARCHAR(50) NOT NULL
    , Idade INT NOT NULL CHECK ( Idade >= 0 )
    , id_parentesco INT NULL
        CONSTRAINT FK_Dependencia REFERENCES Pessoa(id)
)
```

SIMULADO

	Id	Nome	Idade	id_parentesco
1	1	Pedro	77	NULL
2	2	Maria	41	1
3	3	Felipe	16	2

Observem o seguinte modelo:



3. Escreva, e execute, os comandos para inserir neste modelo 3 linhas, de forma que Pedro, de 77 anos, será parente(pai) de Maria, de 41, que será parente(mãe) de Felipe, de 16. (insert)

```
INSERT INTO Pessoa(Nome, Idade, id_parentesco) VALUES ( 'Pedro', 77, NULL)
-- select SCOPE_IDENTITY()
INSERT INTO Pessoa(Nome, Idade, id_parentesco) VALUES ( 'Maria', 41, 1)
-- select SCOPE_IDENTITY()
INSERT INTO Pessoa(Nome, Idade, id_parentesco) VALUES ( 'Felipe', 16, 2)
```

SIMULADO

Um dos alunos de Linguagem SQL notou um erro neste modelo, perguntou ele:

“Mas e se um filho tiver pai e mãe, ou mesmo irmãos, primos, sobrinhos, etc. O relacionamento entre Pessoa e Pessoa não deveria ser N para N ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

4) Escreva, e execute, os comandos para remover da tabela Pessoa as pessoas com nome igual ‘Maria’ ou ‘ Felipe’. (apenas Pedro permanece) (delete)

```
DELETE FROM Pessoa WHERE Nome = 'Felipe'
go
DELETE FROM Pessoa WHERE Nome = 'Maria'
go
```

	Id	Nome	Idade	id_parentesco
1	1	Pedro	77	NULL

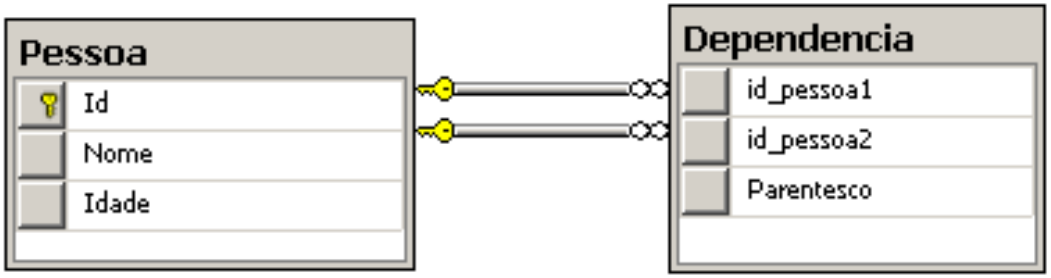
SIMULADO

Um dos alunos de Linguagem SQL notou um erro neste modelo, perguntou ele:

“Mas e se um filho tiver pai e mãe, ou mesmo irmãos, primos, sobrinhos, etc. O relacionamento entre Pessoa e Pessoa não deveria ser N para N ?”

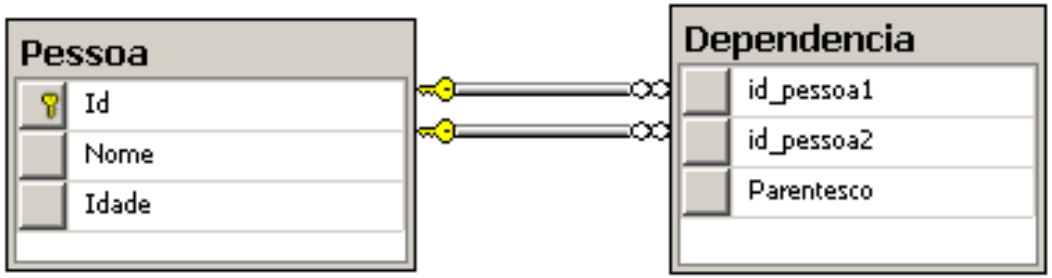
O professor concordou e sugeriu ao aluno executar as seguintes operações:

5. Escreva o modelo físico que atenda às seguintes especificações: (modelo)
   - Uma pessoa pode ter varios dependentes e ao mesmo tempo ser referenciada por varias pessoas
   - O relacionamento entre pessoas deve indicar agora o grau de parentesco (pai, primo) entre elas.



	Id	Nome	Idade	id_parentesco
1	1	Pedro	77	NULL

SIMULADO



	Id	Nome	Idade	id_parentesco
1	1	Pedro	77	NULL

6. Escreva, e execute, os comandos para criar alterar o modelo existente, transformando-o na nova especificação. (create + alter table)

```
ALTER TABLE Pessoa DROP CONSTRAINT FK_Dependencia
go
ALTER TABLE Pessoa DROP COLUMN id_parentesco
go
CREATE TABLE Dependencia (
    Id_pessoa1 INT NOT NULL REFERENCES Pessoa(id)
    , Id_pessoa2 INT NOT NULL REFERENCES Pessoa(id)
    , Parentesco NVARCHAR(50) NOT NULL
)
go
```

SIMULADO

	Id	Nome	Idade
1	1	Pedro	77
2	2	Joana	75
3	3	Maria	41

	Id_pessoa1	Id_pessoa2	Parentesco
1	3	1	FILHA
2	3	2	FILHA
3	1	2	CONJUGE

7. Escreva, e execute, os comandos para inserir neste modelo 4 linhas, de forma que Joana, 75 anos, seja casada com Pedro; e Maria, 41 anos, como filha de Joana (parentesco = mãe) e Pedro (parentesco = pai). (insert)

```
--INSERT INTO Pessoa(Nome, Idade) VALUES ( 'Pedro', 77)
INSERT INTO Pessoa(Nome, Idade) VALUES ( 'Joana', 75 )
-- select * from Pessoa
INSERT INTO Dependencia ( Id_pessoa1, Id_pessoa2, Parentesco ) VALUES (1,2,'CONJUGE')
INSERT INTO Pessoa(Nome, Idade) VALUES ( 'Maria', 41 )
INSERT INTO Dependencia ( Id_pessoa1, Id_pessoa2, Parentesco ) VALUES (3,1,'FILHA')
INSERT INTO Dependencia ( Id_pessoa1, Id_pessoa2, Parentesco ) VALUES (3,2,'FILHA')
```

SIMULADO

Outro aluno de Linguagem SQL notou outro erro neste modelo, disse ele:

“Mas ao salvar a idade não corremos o risco de ter que atualizar a idade a todo ano que se passa ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

8) Escreva, e execute, os comandos para atualizar a idade de todos as pessoas em +1, simulando o que aconteceria na virada de um ano. (update)

UPDATE Pessoa SET Idade += 1

	Id	Nome	Idade
1	1	Pedro	78
2	2	Joana	76
3	3	Maria	42

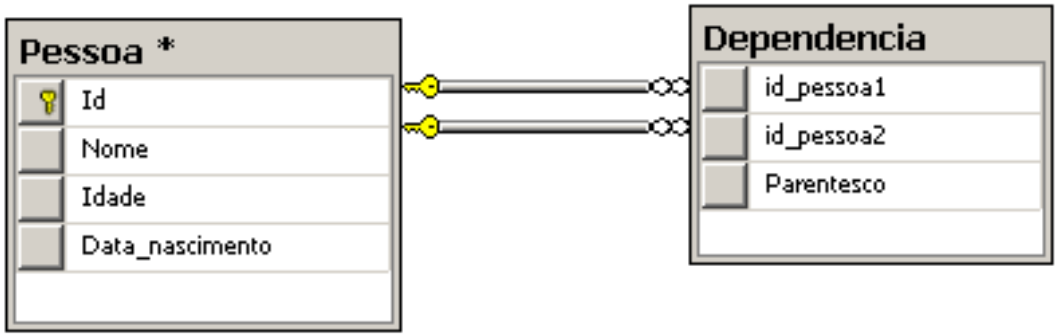
SIMULADO

Outro aluno de Linguagem SQL notou outro erro neste modelo, disse ele:

“Mas ao salvar a idade não corremos o risco de ter que atualizar a idade a todo ano que se passa ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

9) Escreva o modelo físico que atenda às seguintes especificações:
   - Além da coluna Idade, incluir a coluna data de nascimento
   - A coluna idade passará a ser uma coluna calculada a partir da data de nascimento.



SIMULADO

	Id	Nome	Data_nascimento	Idade
1	1	Pedro	1932-03-31	78
2	2	Joana	1934-03-31	76
3	3	Maria	1968-03-31	42

Outro aluno de Linguagem SQL notou outro erro neste modelo, disse ele:

“Mas ao salvar a idade não corremos o risco de ter que atualizar a idade a todo ano que se passa ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

10) Escreva, e execute, os comandos para criar alterar o modelo existente, transformando-o na nova especificação. (alter table)

```
ALTER TABLE Pessoa ADD Data_nascimento DATE NULL
-- update Pessoa set Data_nascimento = dateadd(year, IDADE*-1, getdate() )
-- sp_help Pessoa
ALTER TABLE Pessoa DROP CONSTRAINT CK__Pessoa__Idade__403A8C7D
ALTER TABLE Pessoa drop column Idade
ALTER TABLE Pessoa ADD Idade as datediff(year,Data_nascimento,getdate() )
```

SIMULADO

Outro aluno de Linguagem SQL notou outro erro neste modelo, disse ele:

“Mas ao salvar a idade não corremos o risco de ter que atualizar a idade a todo ano que se passa ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

11) Escreva, e execute, os comandos para inserir neste modelo 1 linha, de forma que Felipe, nascido em 21/02/1995, como filho de Maria. (insert)

```
INSERT INTO Pessoa(Nome, Data_nascimento) VALUES ( 'Felipe', '21/02/1995')
-- select * from Pessoa
INSERT INTO Dependencia ( Id_pessoa1, Id_pessoa2, Parentesco ) VALUES (4,3,'FILHO')
```

	Id	Nome	Data_nascimento	Idade
1	1	Pedro	1932-03-31	78
2	2	Joana	1934-03-31	76
3	3	Maria	1968-03-31	42
4	4	Felipe	1995-02-21	15

	Id_pessoa1	Id_pessoa2	Parentesco
1	3	1	FILHA
2	3	2	FILHA
3	1	2	CONJUGE
4	4	3	FILHO

SIMULADO

12) Escreva, e execute, os comandos para atualizar o parentesco entre Felipe e Maria para ‘ filho adotivo com guarda provisória segundo lei #63459/3’. (update)

```
UPDATE Dependencia
SET Parentesco = 'filho adotivo com guarda provisória segundo lei #63459/3'
WHERE id_pessoa1 = 4 AND id_pessoa2 = 3
```

-- ALTER TABLE Dependencia ALTER COLUMN Parentesco NVARCHAR(100) NOT NULL

Mensagem 8152, Nível 16, Estado 4, Linha 2

Dados de cadeia ou binários seriam truncados.

A instrução foi finalizada.

	Id_pessoa1	Id_pessoa2	Parentesco
1	3	1	FILHA
2	3	2	FILHA
3	1	2	CONJUGE
4	4	3	filho adotivo com guarda provisória segundo lei #63459/3

SIMULADO

Outro aluno de Linguagem SQL teve outro comentário sobre o modelo, disse ele:

“Não gostei deste modelo, vamos jogá-lo fora ?”

O professor concordou e sugeriu ao aluno executar as seguintes operações:

13) Escreva, e execute, os comandos para remover a tabela de Pessoas e suas dependentes do banco de dados (drop table)

```
DROP TABLE Dependencia
go
DROP TABLE Pessoa
go
```

Curso Superior de Tecnologia em Banco de Dados

Obrigado!

Prof. Gustavo Bianchi Maia
gbmaia@gmail.com