

Comandos DDL (Data Definition Language) - 2

MySQL

Prof. Rosana Traversa

Comandos de Definição das Estruturas de Dados de um BD:

- Como vimos os comandos básicos são:
 - **CREATE** : usado para criar bancos de dados, tabelas, visões(views), procedimentos (stored procedures), etc..
 - **ALTER** : usado para alterar alguma característica de uma estrutura já pronta
 - **DROP** : usado para excluir alguma estrutura criada anteriormente

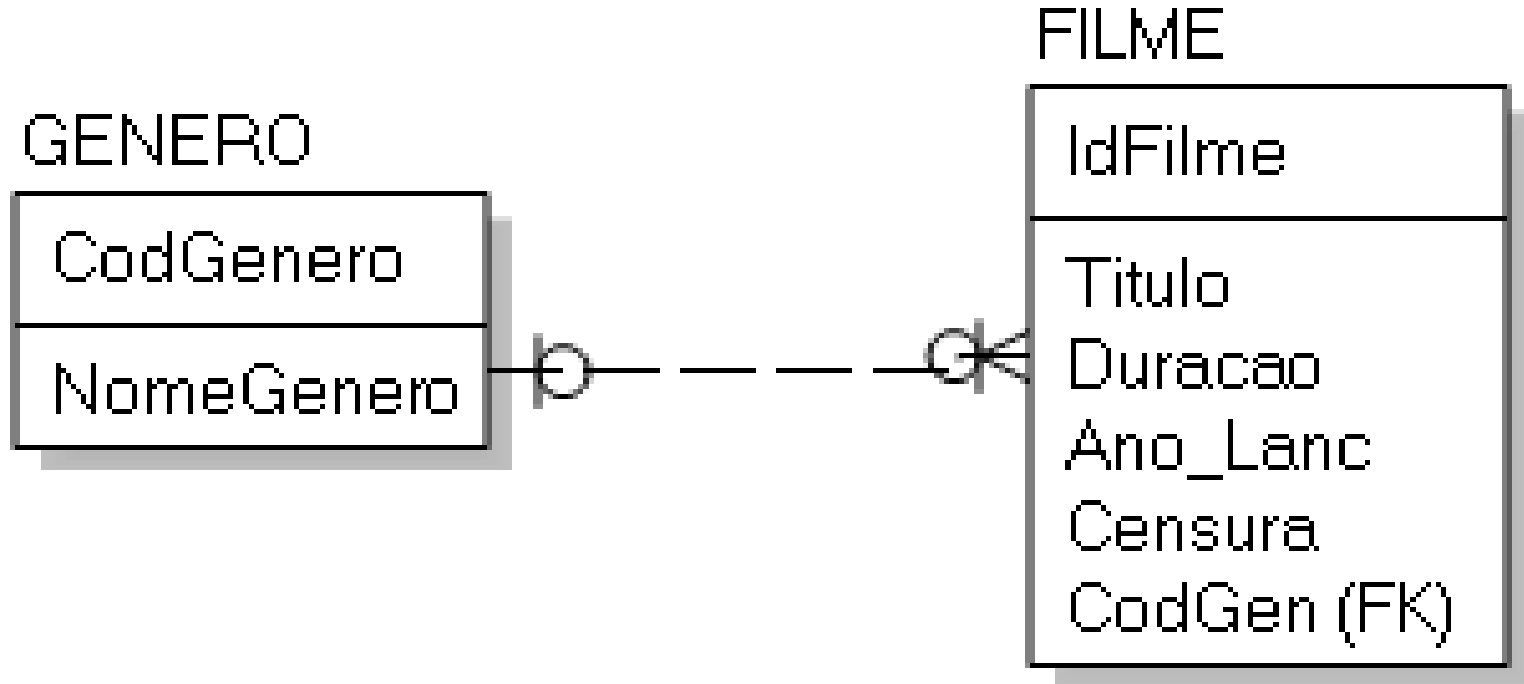
Comandos DDL

- Como definir um relacionamento entre tabelas?



Relacionamento 1:N

CINEMA



```
CREATE DATABASE CINEMA;
```

```
USE CINEMA;
```

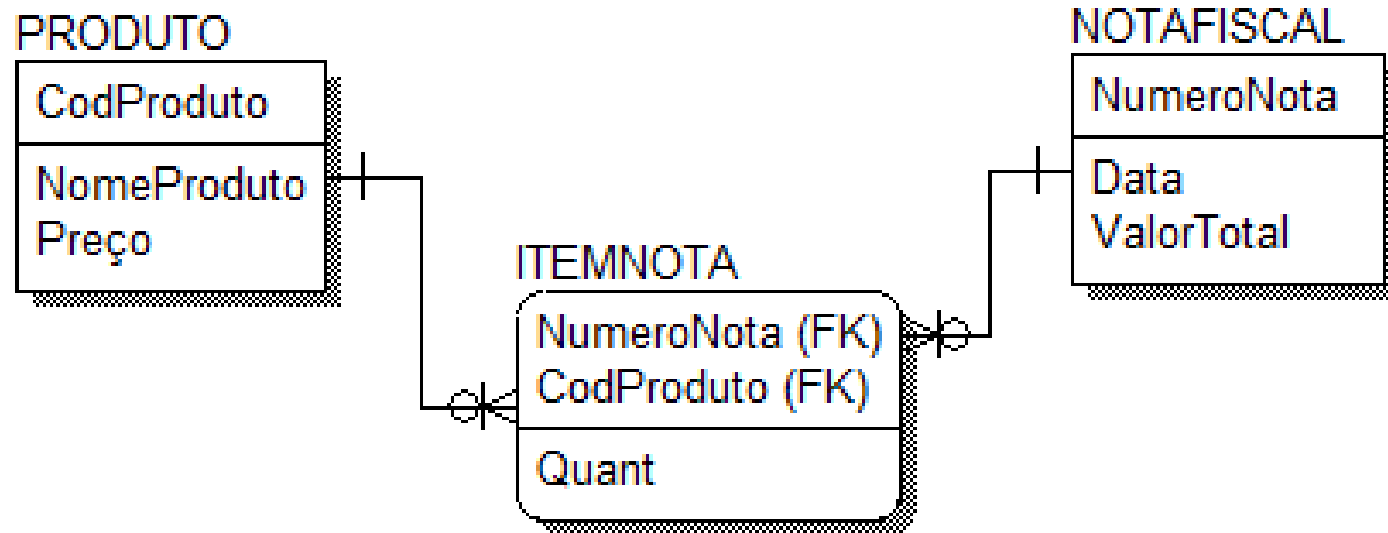
```
CREATE TABLE GENERO (  
  CodGenero int,  
  NomeGenero varchar (20) not null,  
  Primary key(CodGenero));
```

```
CREATE TABLE FILME(  
  idFilme int primary key auto_increment,  
  Titulo varchar(30) not null,  
  Duracao time, /* 'hh:mm' */  
  Ano_lanc year,  
  Censura varchar(15) default 'livre',  
  CodGen int,  
  Foreign key (CodGen) references Genero(CodGenero));
```

Relacionamento
1:N

Relacionamento N:N

VENDAS



Create – Exemplos (cont)

```
CREATE DATABASE VENDAS;
```

```
USE VENDAS;
```

```
CREATE TABLE PRODUTO (
```

```
CodProduto int primary key,
```

```
NomeProduto varchar (20) not null,
```

```
Preco numeric(10,2) not null ) ;
```

Relacionamento N:N

```
CREATE TABLE NOTAFISCAL(
```

```
NumeroNota numeric(10),
```

```
Data date,
```

```
ValorTotal numeric(10,2) default 0,
```

```
Primary key(NumeroNota) ) ;
```

continua...

Ou simplesmente...

```
CREATE TABLE ITEMNOTA (
```

```
    NumeroNota numeric(10),
```

```
    CodProduto int,
```

```
    Quant int NOT NULL CHECK ( Quant >0),
```

```
    PRIMARY KEY ( NumeroNota, CodProduto ) ,
```

(designação de chave primária composta)

```
    FOREIGN KEY ( NumeroNota )
```

```
        REFERENCES NotaFiscal( NumeroNota ) ,
```

```
    FOREIGN KEY ( CodProduto )
```

```
        REFERENCES Produto( CodProduto )
```

```
);
```

Create – Exemplos (cont)

```
CREATE TABLE ITEMNOTA (  
    NumeroNota numeric(10),  
    CodProduto int,  
    Quant int NOT NULL CHECK ( Quant >0),  
    CONSTRAINT pk_itemnota PRIMARY KEY ( NumeroNota,  
    CodProduto ) ,  
    CONSTRAINT fk_notafiscal FOREIGN KEY ( NumeroNota )  
    REFERENCES NotaFiscal( numeronota ) ,  
    CONSTRAINT fk_codProduto FOREIGN KEY ( CodProduto )  
    REFERENCES Produto( CodProduto )  
);
```

Constraint – termo opcional – refere-se a uma restrição nomeada

Constraints

- O termo **Constraint** refere-se a uma restrição **nomeada** que pode se relacionar à chave primária, à chave estrangeira , ou ainda a alguma condição check, unique, null ou default .
- Uma **constraint** pode ser adicionada por um comando ALTER

- **Exemplo:**

```
ALTER TABLE ITEMNOTA add constraint fk_codproduto  
foreign key(Codproduto) references PRODUTO(Codproduto);  
(adiciona a restrição de chave estrangeira denominada  
fk_codproduto à tabela ITEMNOTA )
```

Alterando uma tabela para incluir/excluir a especificação de chave estrangeira

Exemplos:

```
ALTER TABLE ALUNO ADD CONSTRAINT fk_curso FOREIGN  
KEY (curso) REFERENCES CURSO (codCurso)
```

-transforma a coluna curso em chave estrangeira por meio de uma constraint explícita

ou simplesmente....

```
ALTER TABLE ALUNO ADD FOREIGN KEY (curso)  
REFERENCES CURSO (codCurso)
```

```
ALTER TABLE ALUNO DROP FOREIGN KEY (curso)
```

-remove a designação de chave estrangeira e não a coluna curso

Alterando uma tabela para incluir/excluir a especificação de chave estrangeira

Exemplo:

```
Create table UF(  
    sigla char(2) primary key,  
    estado varchar(30) not null);
```

```
Create table Empresa(  
    idEmpresa int primary key,  
    CNPJ char(18) unique,  
    endereco varchar(60),  
    cidade varchar(40),  
    sigla char(2)  
);
```

```
Alter table Empresa add foreign key(sigla) references UF(sigla);
```

Relacionamento entre Tabelas –

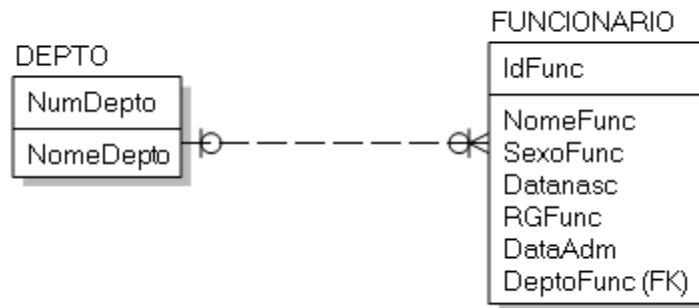
parte 1

SQL

Relacionamentos entre tabelas

- Estes relacionamentos podem ser:
 - **Relacionamentos binários do tipo 1:N ou N:1.** Nestes casos, uma instância de uma tabela se relaciona a uma ou várias outras instância de outra tabela. Ex: FILME x GENERO (um filme possui 1 gênero e 1 gênero está associado a diversos filmes)
 - **Relacionamentos binários do tipo N:N ou n-ários.** Pela modelagem lógica, vimos que estes relacionamentos são desdobrados em 2 relacionamentos binários caindo no caso anterior, passando a **existir uma tabela de relacionamento.**
 - **Auto-relacionamentos** (quando uma instância de uma tabela se relaciona a outra instância da **mesma** tabela. Ex : Funcionários e supervisores (ambos são funcionários))

Exemplo de Relacionamento binário (1:N)



Um funcionário trabalha em um Depto e em 1 Depto trabalham vários funcionários

```
CREATE DATABASE EMPRESA_ABC;  
USE EMPRESA_ABC;  
CREATE TABLE DEPTO (  
  NumDeptO INT PRIMARY KEY,  
  NomeDeptO VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE FUNCIONARIO (  
  IdFunc INT PRIMARY KEY,  
  NomeFunc VARCHAR(20) NOT NULL,  
  SexoFunc CHAR(1) CHECK (SexoFunc='F' OR SexoFunc='M'),  
  DataNasc DATE,  
  RGFunc NUMERIC(12),  
  DataAdm DATE,  
  DeptoFunc INT, FOREIGN KEY(DeptoFunc) REFERENCES DEPTO(NumDeptO)  
);
```

Preenchimento da tabela DEPTO

NumDeppto	NomeDeppto
1	Marketing
2	Compras
3	Recursos Humanos

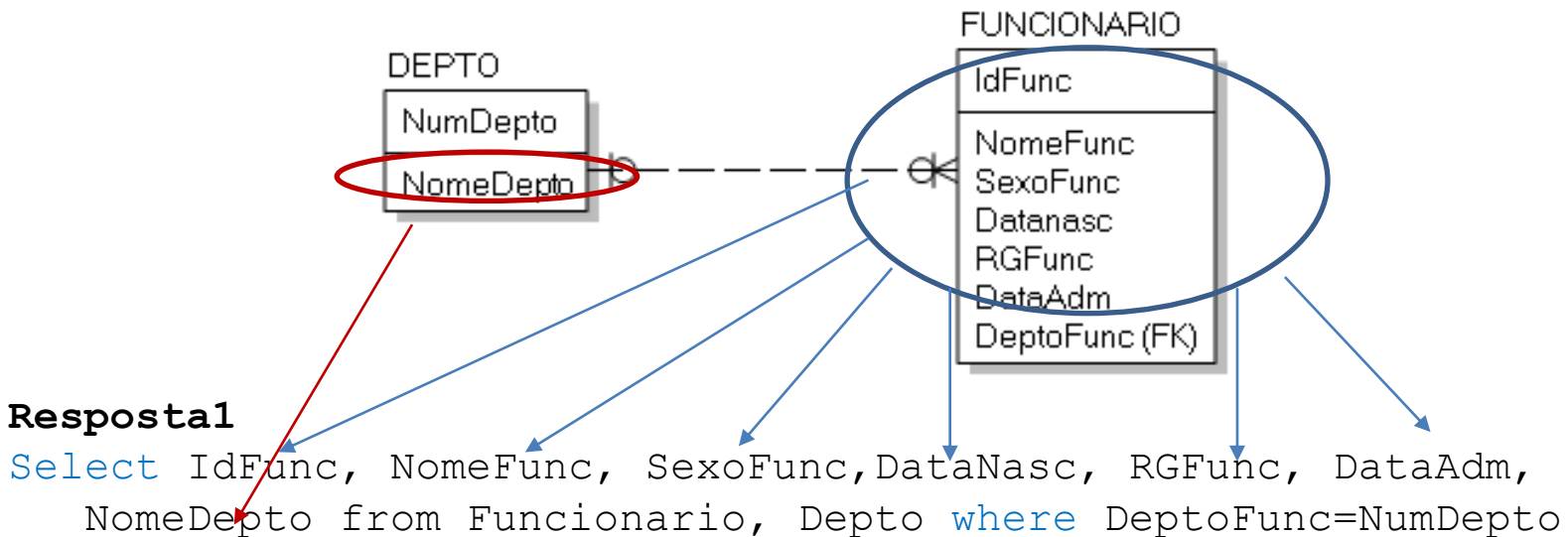
Preenchimento da tabela FUNCIONARIO

IdFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	DepptoFunc
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	1
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	1
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	3
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	1
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	NULL
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	3
700	Fabdio Moraes	m	1989-06-15	5555555	2011-01-01	1

Como saber os nomes dos departamentos(e não os números) onde os funcionários trabalham?

```
Select * from funcionario
```

```
-- vai mostrar os dados dos funcionários incluindo o numero do seu depto
```



**** Note: aqui não houve necessidade de qualificação pois todos os nomes de colunas são únicos no Banco de dados**

Resposta1:

```
Select IdFunc, NomeFunc, SexoFunc, DataNasc,  
      RGFunc, DataAdm, NomeDepto from  
Funcionario, Depto where DeptoFunc=NumDepto
```

idFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	NomeDepto
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	Marketing
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Marketing
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Recursos Humanos
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Marketing
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	Recursos Humanos
700	Fabdio Moraes	m	1989-06-15	5555555	2011-01-01	Marketing

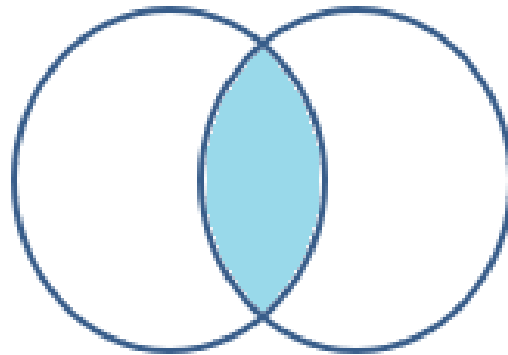
Tipos de join (**inner join** ou **join**) - junção de tabelas

Resposta2:

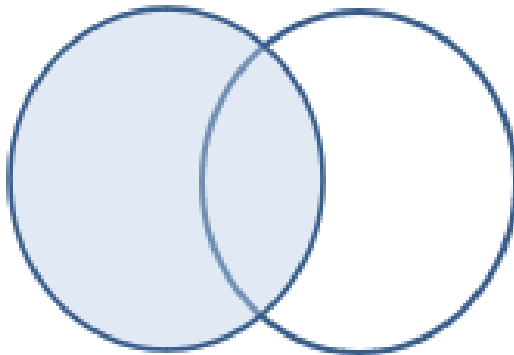
```
Select IdFunc, NomeFunc, SexoFunc, DataNasc,  
        RGFunc, DataAdm, NomeDepto from Funcionario  
inner join Depto on DeptoFunc=NumDepto
```

idFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	NomeDepto
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	Marketing
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Marketing
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Recursos Humanos
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Marketing
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	Recursos Humanos
700	Fabdio Moraes	m	1989-06-15	5555555	2011-01-01	Marketing

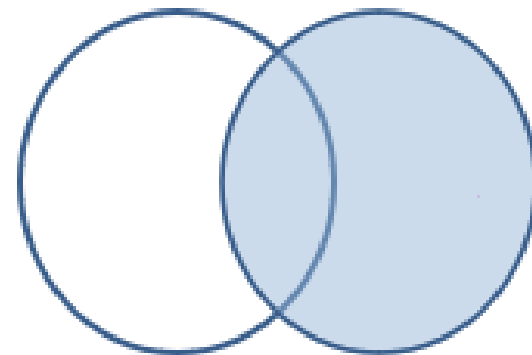
Tipos de Relacionamento 1:N



join ou inner join



left join



right join

Outro tipo de join (left)

Resposta3:

```
Select IdFunc, NomeFunc, SexoFunc, DataNasc, RGFunc,  
DataAdm, NomeDepto from Funcionario left join Depto  
on DeptoFunc=NumDepto
```

idFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	NomeDepto
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	Marketing
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Marketing
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Recursos Humanos
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Marketing
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	NULL
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	Recursos Humanos
700	Fabdio Moraes	m	1989-06-15	5555555	2011-01-01	Marketing

Neste caso, todas as instâncias da primeira entidade são relatadas mesmo que não haja correspondência na segunda entidade

Outro tipo de join (right)

Resposta4:

```
Select IdFunc, NomeFunc, SexoFunc, DataNasc, RGFunc,  
DataAdm, NomeDepto from Funcionario right join  
Depto on DeptoFunc=NumDepto
```

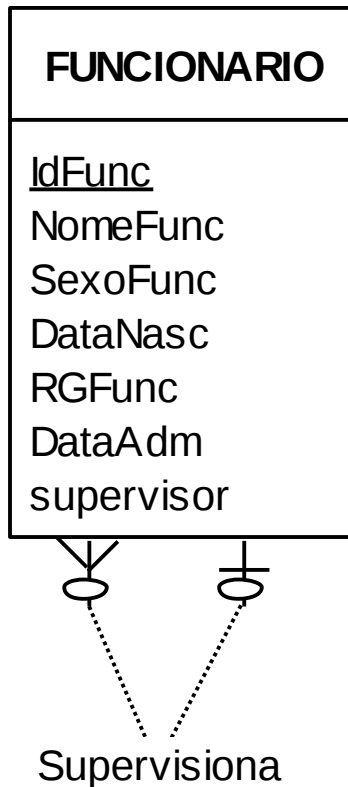
idFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	NomeDepto
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	Marketing
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Marketing
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Marketing
700	Fabdio Moraes	m	1989-06-15	5555555	2011-01-01	Marketing
NULL	NULL	NULL	NULL	NULL	NULL	Compras
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Recursos Humanos
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	Recursos Humanos

*Neste caso, todas as instâncias da **segunda** entidade são relatadas mesmo que não haja correspondência na **primeira** entidade, ou seja, os departamentos que não possuem funcionários também aparecem como se **eventualmente** houvesse algum.*

Auto-relacionamentos

1:N

Exemplo de auto-relacionamento (1:N)



```
CREATE TABLE FUNCIONARIO
(
  IdFunc  INT PRIMARY KEY,
  NomeFunc VARCHAR(20) NOT NULL,
  SexoFunc CHAR(1)
          CHECK (SexoFunc='F' OR SexoFunc='M'),
  DataNasc DATE,
  RGFunc  NUMERIC(12),
  DataAdm DATE,
  supervisor INT,
  FOREIGN KEY (supervisor) REFERENCES
  FUNCIONARIO (IdFunc)
);
```

Preenchimento da tabela FUNCIONARIO

IdFunc	NomeFunc	SexoFunc	DataNasc	RGFunc	DataAdm	Supervisor
100	Paulo Reis	M	1982-12-12	1234567	2002-01-01	NULL
200	Sandra Souza	F	1978-10-21	2233445	2004-01-10	100
300	Rui Siqueira	M	1980-05-18	7657890	2000-05-06	100
400	Miriam Dias	F	1990-11-11	3546728	2008-01-01	200
500	Pedro Ramos	M	1986-07-17	2334999	2007-10-10	100
600	Daniela Kaz	F	1981-08-21	1112223	2000-02-02	NULL
700	Fabio Moraes	M	1989-06-25	5555555	2011-01-01	200

Como saber os nomes dos supervisores imediatos?

```
select * from funcionario  
-- vai mostrar apenas o numero de matricula do  
funcionário supervisor
```

Resposta1:

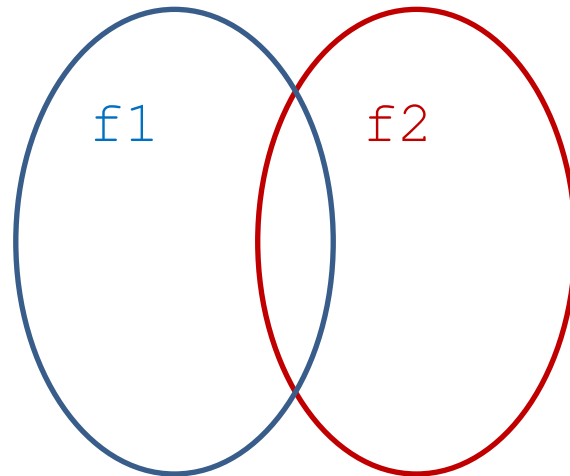
```
Select f1.IdFunc, f1.NomeFunc,  
f1.SexoFunc, f1.DataNasc, f1.RGFunc,  
f1.DataAdm, f2.NomeFunc from Funcionario f1,  
Funcionario f2 where  
f1.supervisor=f2.IdFunc
```

Por que a qualificação f1, f2?

```
Select f1.IdFunc, f1.NomeFunc, f1.SexoFunc,  
f1.DataNasc, f1.RGFunc, f1.DataAdm, f2.NomeFunc  
from Funcionario f1, Funcionario f2 where  
f1.supervisor=f2.IdFunc;
```

Funcionário
(comuns)

Funcionário
(supervisores)



Como saber os nomes dos supervisores imediatos?

Resposta1:

```
Select f1.IdFunc, f1.NomeFunc, f1.SexoFunc,  
f1.DataNasc, f1.RGFunc, f1.DataAdm, f2.NomeFunc as  
supervisor from Funcionario f1, Funcionario f2  
where f1.supervisor=f2.IdFunc;
```

idfunc	nomefunc	sexofunc	datanasc	rgfunc	dataadm	supervisor
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Paulo Reis
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Paulo Reis
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Sandra Souza
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	Paulo Reis
700	Fabio Moraes	m	1989-06-15	5555555	2011-01-01	Sandra Souza

Outra solução idêntica (inner join)

Resposta2:

```
Select f1.IdFunc, f1.NomeFunc, f1.SexoFunc,  
f1.DataNasc, f1.RGFunc, f1.DataAdm, f2.NomeFunc as  
supervisor from Funcionario f1 inner join  
Funcionario f2 on f1.supervisor=f2.IdFunc;
```

idfunc	nomefunc	sexofunc	datanasc	rgfunc	dataadm	supervisor
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Paulo Reis
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Paulo Reis
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Sandra Souza
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	Paulo Reis
700	Fabio Moraes	m	1989-06-15	5555555	2011-01-01	Sandra Souza

Neste caso, apenas as instâncias da primeira entidade que têm correspondência na segunda entidade são reportadas.

Outro tipo de join (left)

Resposta3:

```
Select f1.IdFunc, f1.NomeFunc, f1.SexoFunc,  
f1.DataNasc, f1.RGFunc, f1.DataAdm, f2.NomeFunc as  
supervisor from Funcionario f1 left join  
Funcionario f2 on f1.supervisor=f2.IdFunc;
```

idfunc	nomefunc	sexofunc	datanasc	rgfunc	dataadm	supervisor
100	Paulo Reis	m	1982-12-12	1234567	2002-01-01	NULL
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Paulo Reis
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Paulo Reis
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Sandra Souza
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	Paulo Reis
600	Daniela Kaz	f	1981-08-21	1112222	2000-02-02	NULL
700	Fabio Moraes	m	1989-06-15	5555555	2011-01-01	Sandra Souza

Neste caso, todas as instâncias da primeira entidade são relatadas mesmo que não haja correspondência na segunda entidade

Outro tipo de join (**right**)

Resposta4:

```
Select f1.IdFunc, f1.NomeFunc, f1.SexoFunc,  
f1.DataNasc, f1.RGFunc, f1.DataAdm, f2.NomeFunc as  
supervisor from Funcionario f1 right join  
Funcionario f2 on f1.supervisor=f2.IdFunc;
```

idfunc	nomefunc	sexofunc	datanasc	rgfunc	dataadm	supervisor
200	Sandra Souza	f	1978-12-12	2233445	2004-10-01	Paulo Reis
300	Rui Siqueira	m	1980-05-18	7657890	2000-05-05	Paulo Reis
400	Miriam Dias	f	1990-11-11	3567892	2008-01-01	Sandra Souza
500	Pedro Ramos	m	1986-07-17	2334999	2007-10-10	Paulo Reis
700	Fabio Moraes	m	1989-06-15	5555555	2011-01-01	Sandra Souza
NULL	NULL	NULL	NULL	NULL	NULL	Rui Siqueira
NULL	NULL	NULL	NULL	NULL	NULL	Miriam Dias
NULL	NULL	NULL	NULL	NULL	NULL	Pedro Ramos
NULL	NULL	NULL	NULL	NULL	NULL	Daniela Kaz
NULL	NULL	NULL	NULL	NULL	NULL	Fabio Moraes

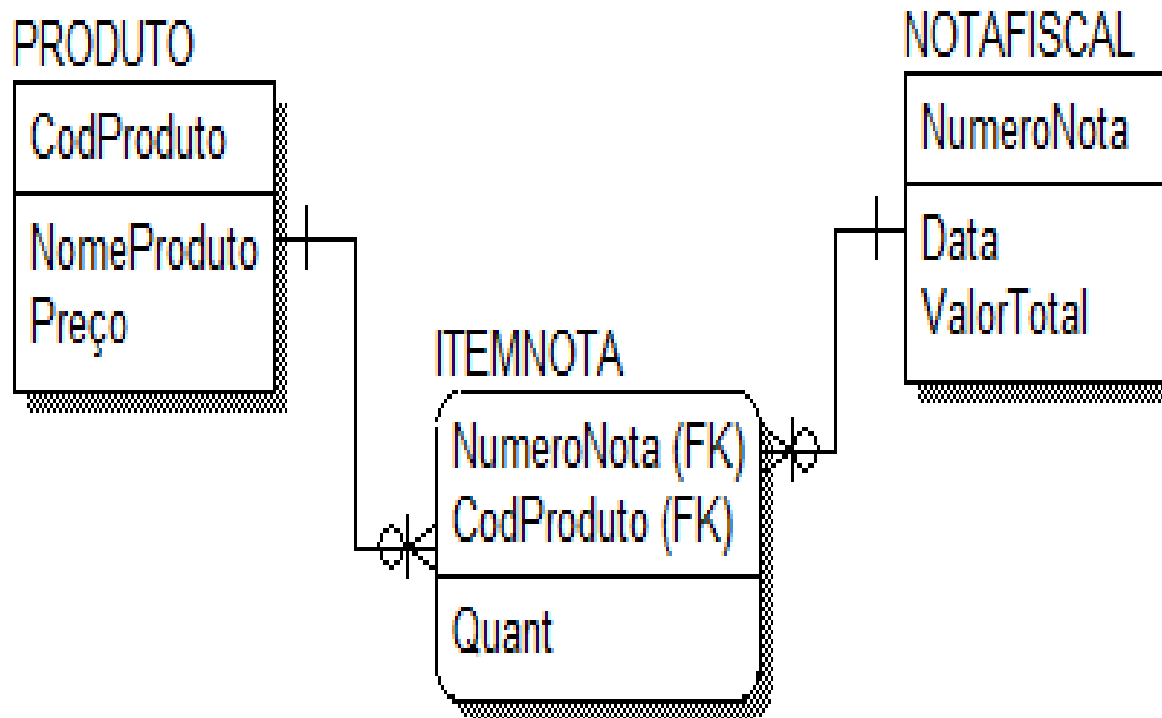
*Neste caso, todas as instâncias da **segunda** entidade são relatadas mesmo que não haja correspondência na **primeira** entidade, ou seja, os funcionários que não supervisionam ninguém também aparecem como se **eventualmente** supervisionassem alguém.*

Relacionamento entre Tabelas – parte 2

Relacionamentos N:N

Relacionamentos N:N

Database Vendas



Relacionamentos N:N

```
CREATE DATABASE VENDAS;
```

```
USE VENDAS;
```

```
CREATE TABLE Produto (  
    CodProduto int not null primary key,  
    NomeProduto varchar (20) not null,  
    Preco numeric(10,2) not null ) ;
```

```
CREATE TABLE NotaFiscal(  
    NumNota numeric(10) primary key,  
    Data Date,  
    ValorTotal numeric(10,2) default 0 ) ;
```

```
CREATE TABLE ItemNota (  
    NumNota numeric(10),  
    CodProduto int,  
    Quant int not null check (Quant > 0),  
    primary key (NumeroNota, CodProduto) ,  
    foreign key (NumeroNota) references NotaFiscal (NumeroNota),  
    foreign key (CodProduto) references Produto (CodProduto) );
```



tabela de relacionamento

Preenchimento das tabelas

Produto

CodProduto	NomeProduto	Preco
100	Produto100	100.00
200	Produto200	200.00
300	Produto300	300.00

NotaFiscal

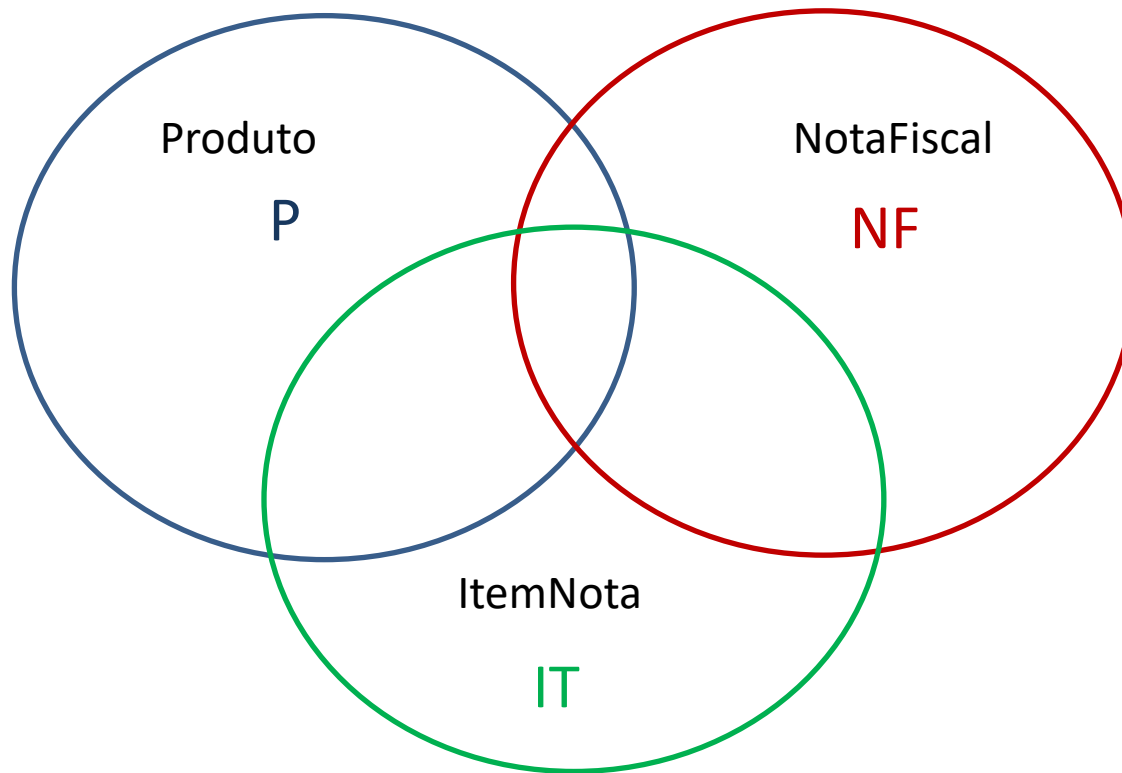
NumNota	Data	ValorTotal
1	2020-06-05	0
2	null	0
3	null	0

ItemNota

NumeroNota	CodProduto	Quant
1	100	10
1	200	10
1	300	10
2	200	100
3	100	1
3	200	1
3	300	1

Relacionamentos N:N

Database Vendas



Como relatar o conteúdo das notas fiscais?

Resposta1:

```
select it.numeronota,nf.data,it.codproduto,  
p.nomeproduto, it.quant, p.preco, it.quant*p.preco as  
subtotal from itemnota it, notafiscal nf, produto p  
where it.numeronota=nf.numeronota and  
it.codproduto=p.codproduto
```

numnota	data	codproduto	nomeproduto	preco	quant	subtotal
1	2019-08-03	100	produto100	100.00	10	1000.00
1	2019-08-03	300	produto300	300.00	10	3000.00
1	2019-08-03	200	produto200	200.00	10	2000.00
2	2019-08-03	200	produto200	200.00	100	20000.00
3	2019-08-03	200	produto200	200.00	1	200.00
3	2019-08-03	100	produto100	100.00	1	100.00
3	2019-08-03	300	produto300	300.00	1	300.00

7 rows in set (0.00 sec)

Outra resposta... usando inner join

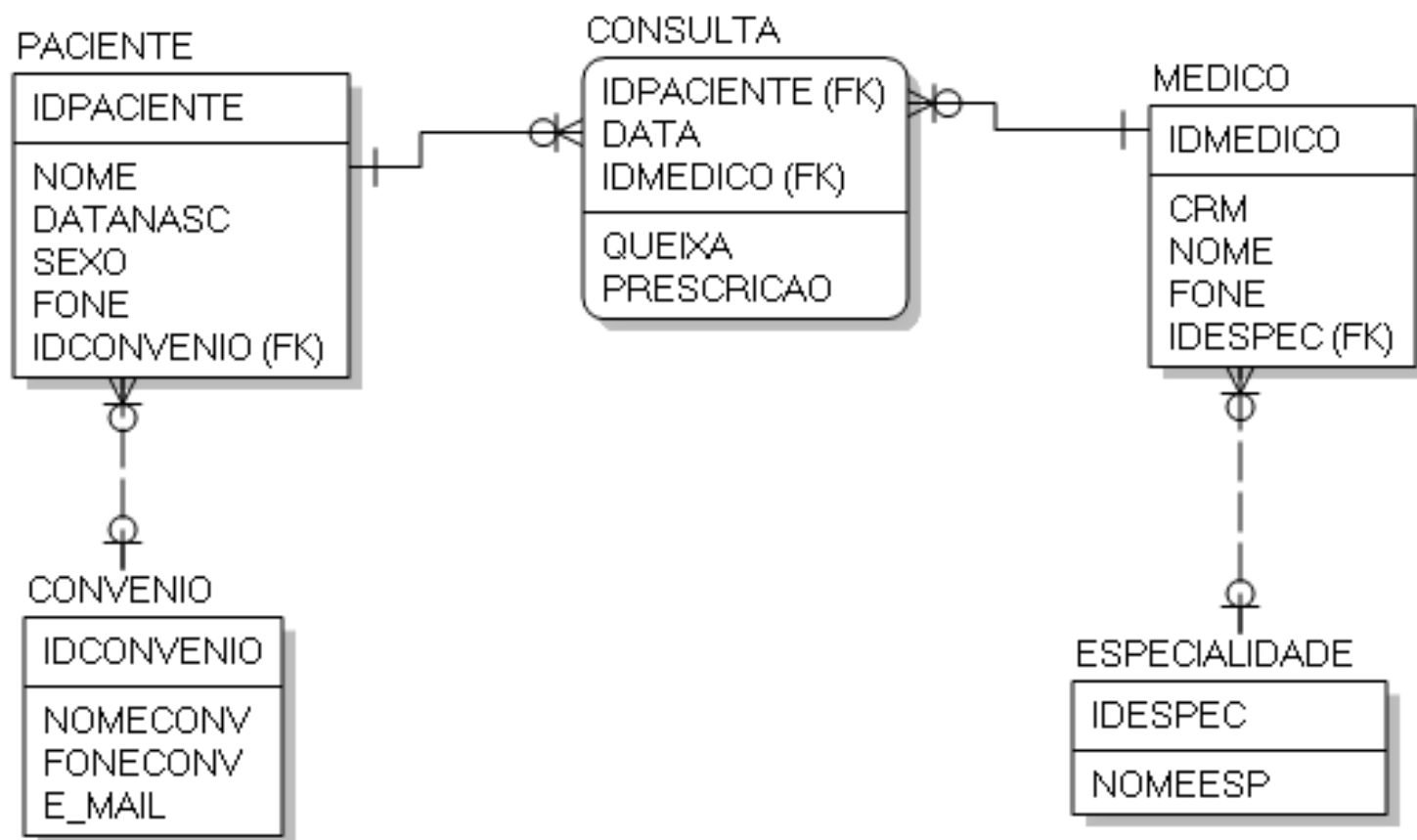
Resposta2:

```
select it.numeronota, nf.data, it.codproduto,
p.nomeproduto, it.quant, p.preco, it.quant*p.preco as
subtotal from itemnota it inner join notafiscal nf on
it.numeronota=nf.numeronota inner join produto p on
it.codproduto=p.codproduto
```

numnota	data	codproduto	nomeproduto	preco	quant	subtotal
1	2019-08-03	100	produto100	100.00	10	1000.00
1	2019-08-03	300	produto300	300.00	10	3000.00
1	2019-08-03	200	produto200	200.00	10	2000.00
2	2019-08-03	200	produto200	200.00	100	20000.00
3	2019-08-03	200	produto200	200.00	1	200.00
3	2019-08-03	100	produto100	100.00	1	100.00
3	2019-08-03	300	produto300	300.00	1	300.00

7 rows in set (0.00 sec)

Consultório



```
/*criacao do banco Consultorio*/  
CREATE DATABASE Consultorio;  
/*conecta ao banco consultorio*/  
USE Consultorio;  
/*criacao da tabela Convenio */  
CREATE TABLE CONVENIO(  
    IDCONVENIO INT AUTO_INCREMENT  
        PRIMARY KEY,  
    NOMECONV VARCHAR(30) NOT NULL,  
    FONECON CHAR(12),  
    E_MAIL VARCHAR(80)  
);  
/*criacao da tabela paciente */  
CREATE TABLE PACIENTE(  
    IDPACIENTE INT PRIMARY KEY  
        AUTO_INCREMENT,  
    NOME VARCHAR(40) NOT NULL,  
    SEXO CHAR(1) DEFAULT 'F',  
    DATANASC DATE,  
    FONE CHAR(12),  
    IDCONVENIO INT, FOREIGN KEY(IDCONVENIO) REFERENCES  
        CONVENIO(IDCONVENIO)  
);
```

```
/*criacao da tabela de especialidade*/  
CREATE TABLE ESPECIALIDADE(  
    IDESPEC INT AUTO_INCREMENT PRIMARY KEY,  
    NOMEESPEC VARCHAR(20) NOT NULL  
);
```

```
/*criacao da tabela de medico*/  
CREATE TABLE MEDICO(  
    IDMEDICO INT PRIMARY KEY AUTO_INCREMENT,  
    CRM NUMERIC(8) NOT NULL,  
    NOME VARCHAR(40) NOT NULL,  
    FONE CHAR(12),  
    IDESPEC INT, FOREIGN KEY (IDESPEC)  
REFERENCES ESPECIALIDADE(IDESPEC)  
);
```

/*criacao da tabela consulta */

```
CREATE TABLE CONSULTA(  
    IDPACIENTE INT,  
    IDMEDICO INT,  
    DATA DATE,  
    QUEIXA VARCHAR(100),  
    PRESCRICAO VARCHAR(100),  
    PRIMARY KEY (IDPACIENTE,IDMEDICO,DATA),  
    FOREIGN KEY (IDPACIENTE) REFERENCES PACIENTE(IDPACIENTE),  
    FOREIGN KEY (IDMEDICO) REFERENCES MEDICO(IDMEDICO)  
);
```

/*Dados dos Convênios*/

```
INSERT INTO CONVENIO VALUES  
(NULL, 'SUL AMERICA',  
6012378,'sulamerica.saude@gruposul.com.br'),  
(NULL,'SAUDE TOTAL',999098,'saude.total@segurosaude.com'),  
(NULL,'GOOD HEALTH',776544,'good.health@safetylives@com');
```

/*Dados dos Pacientes*/

```
INSERT INTO PACIENTE VALUES  
(default, 'JOAO DA SILVA','M', '1998-10-10',5566667,1),  
(default, 'VERA RODRIGUES','F', '1992-08-10',5543211,2),  
(default, 'PAULO RANGEL','M','1990-04-12', 8874662, 3);
```

*/*Dados das Especialidades*/*

```
INSERT INTO ESPECIALIDADE (NOMEESPEC)  
VALUES('CLINICA GERAL'), ('ORTOPEDIA');
```

*/*Dados dos Médicos*/*

```
INSERT INTO MEDICO VALUES  
(NULL, 50345, 'PAULO','77778888', 1),  
(NULL, 12345,' MARINA', '55554444',2);
```

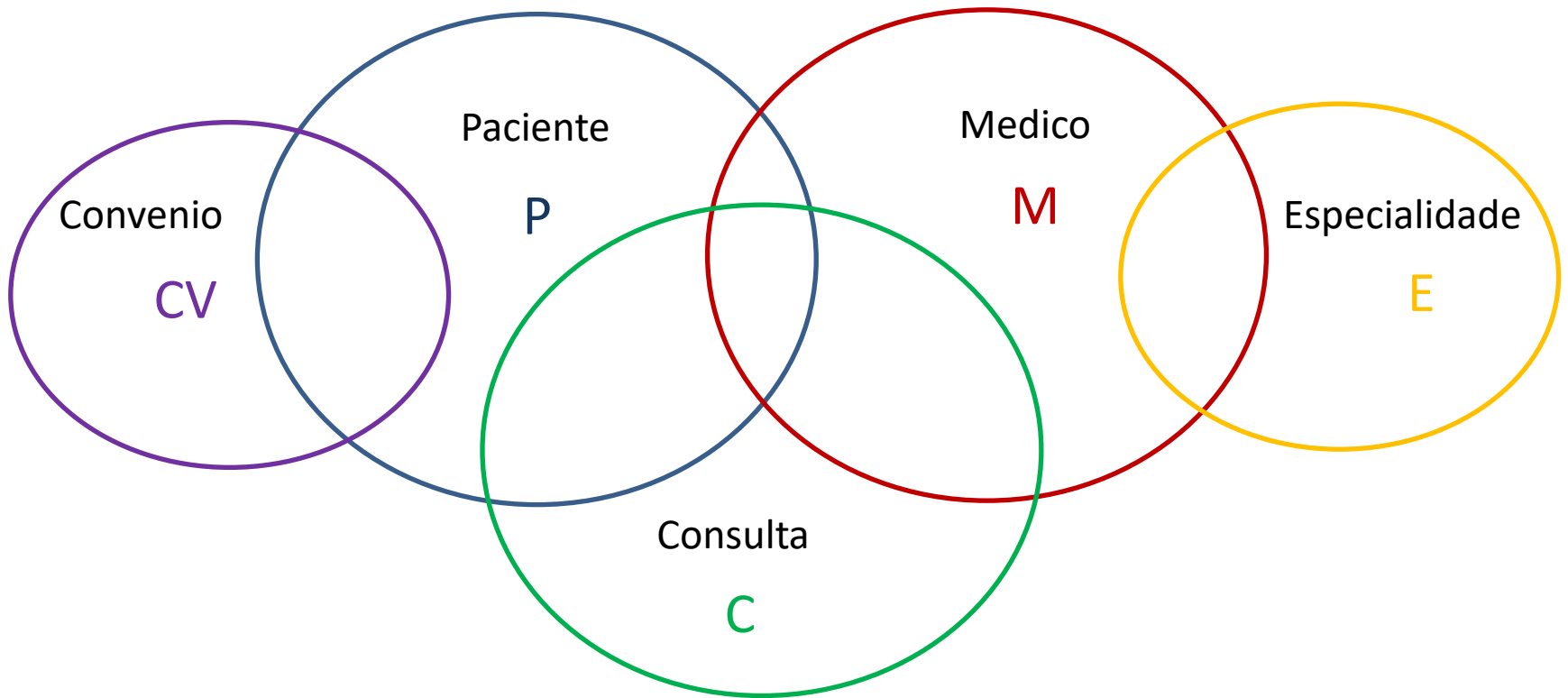
*/*Dados das Consultas*/*

```
INSERT INTO CONSULTA VALUES  
(1,1,CURDATE(),'DOR DE CABECA','ANALGESICO'),  
(1,2,CURDATE(),'DOR NAS COSTAS','RAIO X'),  
(2,1,CURDATE(),'AZIA','OMEPRAZOL');
```

(*) CURDATE() → *data corrente do Sistema Operacional,*

Relacionamentos N:N

Database Consultório



MariaDB [Consultorio]> select * from convenio;

IDCONVENIO	NOMECONV	FONECON	E_MAIL
1	SUL AMERICA	6012378	sulamerica.saude@gruposul.com.br
2	SAUDE TOTAL	999098	saude.total@segurosaude.com
3	GOOD HEALTH	776544	good.health@safetylives@com

3 rows in set (0.000 sec)

MariaDB [Consultorio]> select * from paciente;

IDPACIENTE	NOME	SEXO	DATANASC	FONE	IDCONVENIO
1	JOAO DA SILVA	M	1998-10-10	5566667	1
2	VERA RODRIGUES	F	1992-08-10	5543211	2
3	PAULO RANGEL	M	1990-04-12	8874662	3

3 rows in set (0.000 sec)

Como ver os dados dos Pacientes com seus respectivos Convênios e telefones ??

```
MariaDB [consultorio]> select * from paciente, convenio where  
-> paciente.idconvenio=convenio.idconvenio;
```

IDPACIENTE	NOME	SEXO	DATANASC	FONE	IDCONVENIO	IDCONVENIO	NOMECONV	FONECON
1	JOAO DA SILVA	M	1998-10-10	5566667	1	1	SUL AMERICA	6012378
2	VERA RODRIGUES	F	1992-08-10	5543211	2	2	SAUDE TOTAL	999098
3	PAULO RANGEL	M	1990-04-12	8874662	3	3	GOOD HEALTH	776544

3 rows in set (0.001 sec)

```
MariaDB [consultorio]> select idpaciente,nome,sexo,datanasc,fone,nomeconv,  
-> fonecon from paciente p, convenio c where p.idconvenio=c.idconvenio;
```

idpaciente	nome	sexo	datanasc	fone	nomeconv	fonecon
1	JOAO DA SILVA	M	1998-10-10	5566667	SUL AMERICA	6012378
2	VERA RODRIGUES	F	1992-08-10	5543211	SAUDE TOTAL	999098
3	PAULO RANGEL	M	1990-04-12	8874662	GOOD HEALTH	776544

MariaDB [Consultorio]> select * from especialidade;

IDESPEC	NOMEESPEC
1	CLINICA GERAL
2	ORTOPEDIA

2 rows in set (0.001 sec)

MariaDB [Consultorio]> select * from medico;

IDMEDICO	CRM	NOME	FONE	IDESPEC
1	50345	PAULO	77778888	1
2	12345	MARINA	55554444	2

2 rows in set (0.001 sec)

MariaDB [Consultorio]> select * from consulta;

IDPACIENTE	IDMEDICO	DATA	QUEIXA	PRESCRICAO
1	1	2020-06-15	DOR DE CABECA	ANALGESICO
1	2	2020-06-15	DOR NAS COSTAS	RAIO X
2	1	2020-06-15	AZIA	OMEPRAZOL

3 rows in set (0.000 sec)

Como ver os dados dos Médicos e suas respectivas Especialidades ??

```
MariaDB [Consultorio]> select * from medico, especialidade where  
-> medico.idespec=especialidade.idespec;
```

IDMEDICO	CRM	NOME	FONE	IDESPEC	IDESPEC	NOMEESPEC
1	50345	PAULO	77778888	1	1	CLINICA GERAL
2	12345	MARINA	55554444	2	2	ORTOPEDIA

2 rows in set (0.002 sec)

```
MariaDB [Consultorio]> select idmedico,crm,nome,fone,nomeespec from medico m,  
-> especialidade e where m.idespec=e.idespec;
```

idmedico	crm	nome	fone	nomeespec
1	50345	PAULO	77778888	CLINICA GERAL
2	12345	MARINA	55554444	ORTOPEDIA

2 rows in set (0.001 sec)

Como ver quais pacientes foram consultados por quais médicos ??

```
MariaDB [consultorio]> select paciente.nome, medico.nome, data, queixa, prescricao from  
-> consulta, medico, paciente where consulta.idpaciente=paciente.idpaciente and  
-> consulta.idmedico=medico.idmedico;
```

nome	nome	data	queixa	prescricao
JOAO DA SILVA	PAULO	2020-06-06	DOR DE CABECA	ANALGESICO
JOAO DA SILVA	MARINA	2020-06-06	DOR NAS COSTAS	RAIO X
VERA RODRIGUES	PAULO	2020-06-06	AZIA	OMEPRAZOL

3 rows in set (0.001 sec)

```
MariaDB [consultorio]>  
MariaDB [consultorio]>  
MariaDB [consultorio]> select p.nome, m.nome, data, queixa, prescricao from paciente p,  
-> medico m, consulta c where c.idpaciente=p.idpaciente and c.idmedico=m.idmedico;
```

nome	nome	data	queixa	prescricao
JOAO DA SILVA	PAULO	2020-06-06	DOR DE CABECA	ANALGESICO
JOAO DA SILVA	MARINA	2020-06-06	DOR NAS COSTAS	RAIO X
VERA RODRIGUES	PAULO	2020-06-06	AZIA	OMEPRAZOL

3 rows in set (0.002 sec)

Como ver paciente/convenio consultado por médico/especialidade???

```
MariaDB [consultorio]> select paciente.nome, convenio.nomeconv, medico.nome, especialidade.nomeespec,  
-> consulta.data, consulta.queixa, consulta.prescricao from paciente, convenio, medico,  
-> especialidade, consulta where paciente.idpaciente=consulta.idpaciente and  
-> paciente.idconvenio=convenio.idconvenio and medico.idmedico=consulta.idmedico and  
-> medico.idespec=especialidade.idespec;
```

nome	nomeconv	nome	nomeespec	data	queixa	prescricao
JOAO DA SILVA	SUL AMERICA	PAULO	CLINICA GERAL	2020-06-06	DOR DE CABECA	ANALGESICO
VERA RODRIGUES	SAUDE TOTAL	PAULO	CLINICA GERAL	2020-06-06	AZIA	OMEPRAZOL
JOAO DA SILVA	SUL AMERICA	MARINA	ORTOPEDIA	2020-06-06	DOR NAS COSTAS	RAIO X

3 rows in set (0.002 sec)

```
MariaDB [consultorio]>  
MariaDB [consultorio]>  
MariaDB [consultorio]> select p.nome, cv.nomeconv, m.nome, e.nomeespec, c.data, c.queixa, c.prescricao  
-> from paciente p, medico m, convenio cv, especialidade e, consulta c where  
-> p.idpaciente=c.idpaciente and p.idconvenio=cv.idconvenio and m.idmedico=c.idmedico and  
-> m.idespec=e.idespec;
```

nome	nomeconv	nome	nomeespec	data	queixa	prescricao
JOAO DA SILVA	SUL AMERICA	PAULO	CLINICA GERAL	2020-06-06	DOR DE CABECA	ANALGESICO
VERA RODRIGUES	SAUDE TOTAL	PAULO	CLINICA GERAL	2020-06-06	AZIA	OMEPRAZOL
JOAO DA SILVA	SUL AMERICA	MARINA	ORTOPEDIA	2020-06-06	DOR NAS COSTAS	RAIO X

3 rows in set (0.003 sec)

Como ver paciente/convenio consultado por médico/especialidade???

outra forma...

```
MariaDB [consultorio]> select p.nome, cv.nomeconv, m.nome, e.nomeespec, c.data, c.queixa, c.prescricao  
-> from paciente p inner join convenio cv on p.idconvenio=cv.idconvenio inner join consulta c on  
-> p.idpaciente=c.idpaciente inner join medico m on c.idmedico=m.idmedico inner join especialidade e  
-> on m.idespec=e.idespec;
```

nome	nomeconv	nome	nomeespec	data	queixa	prescricao
JOAO DA SILVA	SUL AMERICA	PAULO	CLINICA GERAL	2020-06-06	DOR DE CABECA	ANALGESICO
VERA RODRIGUES	SAUDE TOTAL	PAULO	CLINICA GERAL	2020-06-06	AZIA	OMEPRAZOL
JOAO DA SILVA	SUL AMERICA	MARINA	ORTOPEDIA	2020-06-06	DOR NAS COSTAS	RAIO X

3 rows in set (0.003 sec)

Views

MySql

Prof. Rosana

Visões [views]

- Uma visão [view] é uma forma alternativa de olhar os dados contidos em uma ou mais tabelas.
- Uma view é também chamada de tabela lógica.
- Para definir uma visão [view] usa-se o comando SELECT para fazer uma consulta sobre a(s) tabela(s).
- A view aparece depois como se fosse uma tabela (mas não há alocação física de dados)

Views

- VIEWS TÊM AS SEGUINTE VANTAGENS:
- Uma view pode restringir quais colunas da tabela podem ser acessadas (para leitura ou para modificação), o que é útil no caso de controle de acesso.
- Uma consulta SELECT usada com muita frequência pode ser criada como uma view.
- Views podem conter valores calculados ou valores de resumo, o que simplifica a operação.
- Uma view pode ser usada para exportar dados para outras aplicações.

Database escola

```
MariaDB [escola]> select * from aluno;
```

matricula	nome	sobrenome	sexo	datanasc	curso	nota1	nota2
1	Carlos	Santos	m	2000-10-10	informatica	5.00	0.00
2	Pedro	Siqueira	m	2002-08-10	mecanica	5.00	0.00
3	Antonio	Cardoso	m	2002-10-08	informatica	5.00	0.00
4	Debora	NULL	f	2000-12-25	informatica	5.00	8.00
5	Tania	Clemente	f	2002-11-15	artes	5.00	7.50
6	Carmen	Clemente	f	2001-04-01	mecanica	5.00	6.00
7	Paulo	Ribeiro	m	2000-02-15	mecanica	7.00	8.50
8	Vera	Sampaio	f	2000-05-20	informatica	10.00	9.00
9	Leonardo	Santos	m	2001-09-07	mecanica	5.00	3.00
10	Elen	Menezes	f	2001-12-25	artes	8.00	8.00

```
10 rows in set (0.000 sec)
```

Criação de Exibições – usando DDL

```
Create view <nome_view> as  
    [ comando sql....]
```

Exemplo:

```
create view visaol as  
    select * from aluno order by nome;
```

Database escola

```
MariaDB [escola]> create view visao1 as  
-> select * from aluno order by nome;  
Query OK, 0 rows affected (0.058 sec)
```

```
MariaDB [escola]> describe visao1;
```

Field	Type	Null	Key	Default	Extra
matricula	int(11)	NO		0	
nome	varchar(20)	NO		NULL	
sobrenome	varchar(20)	YES		NULL	
sexo	char(1)	YES		m	
datanasc	date	YES		NULL	
curso	varchar(20)	YES		NULL	
nota1	decimal(5,2)	YES		5.00	
nota2	decimal(5,2)	YES		0.00	

```
8 rows in set (0.008 sec)
```

```
MariaDB [escola]> select * from visao1;
```

matricula	nome	sobrenome	sexo	datanasc	curso	nota1	nota2
3	Antonio	Cardoso	m	2002-10-08	informatica	5.00	0.00
1	Carlos	Santos	m	2000-10-10	informatica	5.00	0.00
6	Carmen	Clemente	f	2001-04-01	mecanica	5.00	6.00
4	Debora	NULL	f	2000-12-25	informatica	5.00	8.00
10	Elen	Menezes	f	2001-12-25	artes	8.00	8.00
9	Leonardo	Santos	m	2001-09-07	mecanica	5.00	3.00
7	Paulo	Ribeiro	m	2000-02-15	mecanica	7.00	8.50
2	Pedro	Siqueira	m	2002-08-10	mecanica	5.00	0.00
5	Tania	Clemente	f	2002-11-15	artes	5.00	7.50
8	Vera	Sampaio	f	2000-05-20	informatica	10.00	9.00

Criação de Views – usando DDL

Outros exemplos:

```
create view visao2 as
```

```
select matricula, nome, (nota1+nota2)/2 as media
```

```
from alunos where curso = 'informática' ;
```

```
MariaDB [escola]> create view visao2 as  
-> select matricula, nome, (nota1+nota2)/2 as media from aluno where curso='informatica';  
Query OK, 0 rows affected (0.056 sec)
```

```
MariaDB [escola]> describe visao2;
```

Field	Type	Null	Key	Default	Extra
matricula	int(11)	NO		0	
nome	varchar(20)	NO		NULL	
media	decimal(10,6)	YES		NULL	

```
3 rows in set (0.010 sec)
```

```
MariaDB [escola]> select * from visao2;
```

matricula	nome	media
1	Carlos	2.500000
3	Antonio	2.500000
4	Debora	6.500000
8	Vera	9.500000

```
4 rows in set (0.002 sec)
```

Criação de Views – usando DDL

Outros exemplos:

```
create view mediaporcurso as
```

```
select curso, round(avg((nota1+nota2)/2),2) as
```

```
'media por curso' from alunos group by curso;
```

```
MariaDB [escola]> create view mediaporcurso as
```

```
-> select curso, round(avg((nota1+nota2)/2),2) as 'media por curso' from aluno
```

```
-> group by curso;
```

```
Query OK, 0 rows affected (0.057 sec)
```

```
MariaDB [escola]> desc mediaporcurso;
```

Field	Type	Null	Key	Default	Extra
curso	varchar(20)	YES		NULL	
media por curso	decimal(7,2)	YES		NULL	

```
2 rows in set (0.012 sec)
```

```
MariaDB [escola]> select * from mediaporcurso;
```

curso	media por curso
artes	7.13
informatica	5.25
mecanica	4.94

```
3 rows in set (0.001 sec)
```

Show Tables

As views aparecem no comando Show Tables como se fossem tabelas físicas, mas são tabelas lógicas!

```
MariaDB [escola]> show tables;  
+-----+  
| Tables_in_escola |  
+-----+  
| aluno  
| mediaporcurso  
| visao1  
| visao2  
+-----+  
4 rows in set (0.002 sec)
```

Criação de Views – usando DDL

As views são muito úteis quando reúnem dados de diferentes tabelas:

```
create view pacienteconvenio as
```

```
  Select p.idpaciente, p.nome, p.sexo, p.datanasc,  
  p.fone, p.idconvenio, c.nomeconv from paciente p,  
  convenio c where p.idconvenio=c.idconvenio;
```

```
MariaDB [consultorio]> create view pacienteconvenio as  
  -> select p.idpaciente,p.nome,p.sexo, p.datanasc,p.idconvenio,c.nomeconv from  
  -> paciente p, convenio c where p.idconvenio=c.idconvenio;  
Query OK, 0 rows affected (0.065 sec)
```

```
MariaDB [consultorio]> desc pacienteconvenio;
```

Field	Type	Null	Key	Default	Extra
idpaciente	int(11)	NO		0	
nome	varchar(40)	NO		NULL	
sexo	char(1)	YES		F	
datanasc	date	YES		NULL	
idconvenio	int(11)	YES		NULL	
nomeconv	varchar(30)	NO		NULL	

```
6 rows in set (0.012 sec)
```

```
MariaDB [consultorio]> select * from pacienteconvenio;
```

idpaciente	nome	sexo	datanasc	idconvenio	nomeconv
1	JOAO DA SILVA	M	1998-10-10	1	SUL AMERICA
2	VERA RODRIGUES	F	1992-08-10	2	SAUDE TOTAL
3	PAULO RANGEL	M	1990-04-12	3	GOOD HEALTH

Execução de Views

```
Select */<lista de colunas>  
      from <nome_view> [.....];
```

```
Select * from visao2 order by nome;
```

```
Select * from mediaporcurso where  
      curso in ( 'informática' , 'artes' );
```

```
Select nome,sexo,datanasc,nomeconv  
      as convenio from pacienteconvenio;
```