

```
#nao fazer  
#Só pode fazer para testar ...
```

```
drop database if exists modelo4;
```

```
create database modelo4;
```

```
use modelo4;
```

```
#Auto_relacionamento  
#Pessoa = Conjuge = filho  
#Observar para Criticar  
#O que achou ?  
# Deixando Como Estávamos Observando ...
```

```
create table pessoa(  
    idPessoa int primary key,  
    nome varchar (50),  
    datanascimento date  
);
```

```
create table dependente(  
    idDependente int primary key,  
    nome varchar (50),  
    datanascimento date,  
    id_pessoa int,  
    foreign key(id_pessoa) references pessoa(idPessoa)  
);
```

```
#Utilizando técnicas de Auto_relacionamento  
# Auto_Relacionamento (Uma tabela que se relaciona com ela  
mesma)  
#Auto_relacionamento (1 p 1) Cardinalidade
```

```
create table segurado(  
    idSegurado int primary key ,  
    nome varchar (35),  
    datanascimento date,  
    conjuge int,  
    foreign key(conjuge) references segurado(idsegurado));
```

```
#Por que null no conjuge?  
#porque ela ainda não existe  
#Ele começa null
```

```
insert into segurado values (100,'jose','1990-03-10',null);
insert into segurado values (101,'maria','1992-12-15',100);
```

```
select m.idsegurado, m.nome "marido", m.datanascimento,
       e.idsegurado, e.nome "esposa" , e.datanascimento
from segurado m, segurado e
where m.idsegurado = e.conjuge;
```

```
+-----+-----+-----+-----+-----+
| idsegurado | marido | datanascimento | idsegurado | esposa |
| datanascimento |
+-----+-----+-----+-----+-----+
| 100 | jose | 1990-03-10 | 101 | maria | 1992-12-15 |
+-----+-----+-----+-----+-----+
```

#Quem é o titulo

```
select nome,datanascimento from segurado where conjuge is null;
```

```
+-----+-----+
| nome | datanascimento |
+-----+-----+
| jose | 1990-03-10 |
+-----+-----+
```

#Completa

#Auto-Relacionamento 1 para 1, 1 para Muitos ...

```
create table seguroFuncionario(
  idSegurado int primary key ,
  nome varchar (35),
  datanascimento date,
  premio double,
  tipo enum ('titular','dependente') not null,
  conjuge int unique,
  dependentes int,
  foreign key(conjuge) references seguroFuncionario(idsegurado),
  foreign key(dependentes) references
seguroFuncionario(idsegurado)
);
```

```
insert into seguroFuncionario
  values(1,'manoel','1980-05-10',1000,'titular',null,null );
insert into seguroFuncionario
  values(2,'sandra','1990-01-10',null,'dependente',1,null );
insert into seguroFuncionario
  values(3,'rodrigo','2010-06-12',null,'dependente',null,1 );
insert into seguroFuncionario
  values(4,'marcela','2013-06-20',null,'dependente',null,1 );
```

#Mostrar os titular

```
select * from seguroFuncionario where tipo='titular';
```

#Mostrar os dependente

```
-----+-----+-----+-----+-----+-----+-----+
|idSegurado|nome|datanascimento|premio|tipo|conjuge|dependentes|
-----+-----+-----+-----+-----+-----+-----+
| 1 | manoel | 1980-05-10 | 1000 | titular | NULL | NULL |
-----+-----+-----+-----+-----+-----+-----+
```

```
select * from seguroFuncionario where tipo='dependente';
```

```
+-----+-----+-----+-----+-----+-----+-----+
|idSegurado|nome|datanascimento|premio|tipo|conjuge|dependentes|
+-----+-----+-----+-----+-----+-----+-----+
| 2 | sandra | 1990-01-10 | NULL | dependente | 1 | NULL |
| 3 | rodrigo | 2010-06-12 | NULL | dependente | NULL | 1 |
| 4 | marcela | 2013-06-20 | NULL | dependente | NULL | 1 |
+-----+-----+-----+-----+-----+-----+-----+
```

###Data

#mostrar A data Formatada e a idade

#truncate sem casa **decimal**

#datadiff diferença entre data

# %y (final ano (2 digitos))

# %Y ano Completo ...

```
select nome,
       date_format(datanascimento, '%d-%m-%Y') as nascimento,
       truncate(datediff(now(), datanascimento)/365.25, 0) as
idade, tipo from segurofuncionario;
```

```
# truncate, 0) sem casa decimal
# Confirma o calculo da data
```

```
select truncate(datediff(now(), '1973-04-18')/365.25, 0) as idade;
```

```
+-----+
| idade |
+-----+
|    43 |
+-----+
```

###Mostrar os dependentes, quem é o seu titular, conjuge

```
select t.idsegurado, t.nome, t.tipo,
       c.idsegurado "idConjuge", c.nome "nome Conjuge", c.tipo
"tipoConjuge",
       d.idsegurado "iddependente", d.nome "nome dependente", c.tipo
"tipoDependente"
from segurofuncionario t
right join segurofuncionario c
on t.idsegurado = c.conjuge
right join segurofuncionario d
on t.idsegurado = d.dependentes;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| idsegurado | nome   | tipo   | idConjuge | nome Conjuge |
tipoConjuge | iddependente | nome dependente | tipoDependente |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| NULL | NULL | NULL | NULL | NULL | NULL | 1 | Manoel | NULL
| NULL | NULL | NULL | NULL | NULL | NULL | 2 | Sandra | NULL
| 1 | manael | titular | 2 | Sandra | dependente | 3 | rodrigo | dependente |
| 1 | manael | titular | 2 | Sandra | dependente | 4 | marcela | dependente |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Modelar (poucos dados muitas tabelas) evitar redundancia

#Apaga da Relacional para Não Relacional

#Apagar do ultimo para o primeiro

```
drop table dependente;
```

```
drop table funcionario;
```

```
create table funcionario(  
    idFuncionario int primary key,  
    nome varchar (35),  
    datanascimento date,  
    premio double,  
    salario Double  
);
```

```
drop table dependente;
```

```
create table dependente(  
    idDependente int primary key,  
    nome varchar (35),  
    tipo enum ('conjuge', 'filho'),  
    datanascimento date,  
    idade int,  
    premio double,  
    id_funcionario int,  
    foreign key(id_funcionario) references  
funcionario(idFuncionario)  
);
```

#Trigger (gatilho é um programa preso a tabela), **insert**, **update**,  
**delete**

#EU ATIVO GATILHO QUANDO ELE INSERE **NO** FUNCIONARIO

```
drop trigger if exists gat_funcionario;
```

```
delimiter $$  
create trigger gat_funcionario  
before insert on funcionario  
for each row  
begin
```

```
declare idade int default 0;
set @idade :=0;
set @idade := truncate(datediff(now(),
new.datanascimento)/365.25,0);
set new.idade := @idade;
if @idade<=20 then
set new.premio = 500;
elseif @idade>20 and @idade<=25 then
set new.premio = 600;
elseif @idade>25 and @idade<=30 then
set new.premio = 700;
elseif @idade>30 and @idade<=35 then
set new.premio = 800;
elseif @idade>35 and @idade<=40 then
set new.premio = 900;
elseif @idade>40 and @idade<=45 then
set new.premio = 1000;
end if;

end;
$$
delimiter ;
```

```
drop trigger if exists gat_dependente;
```

```
delimiter $$
```

```
create trigger gat_dependente
before insert on dependente
for each row
begin
declare idade int default 0;
set @idade :=0;
set @idade := truncate(datediff(now(),
new.datanascimento)/365.25,0);
set new.idade := @idade;
if (new.tipo='conjugue') then
if @idade<=20 then
set new.premio = 500;
elseif @idade>20 and @idade<=25 then
set new.premio = 600;
elseif @idade>25 and @idade<=30 then
set new.premio = 700;
elseif @idade>30 and @idade<=35 then
```

```

        set new.premio = 800;
    elseif @idade>35 and @idade<=40 then
        set new.premio = 900;
    elseif @idade>40 and @idade<=45 then
        set new.premio = 1000;
    end if;

    elseif (new.tipo='filho') then
        if @idade<=20 then
            set new.premio = 500;
        else
            set new.premio = 0;
        end if;
    end if;
end;
$$
delimiter ;

```

```

insert into funcionario values (2005,'jose','1980-03-10',0,
3000);
insert into dependente values (900,'sergio','filho','2000-10-
10', 0,0,2005);
insert into dependente values (901,'maria','conjuge','1985-10-
10',0,0,2005);

```

```
select * from dependente;
```

idDependente	nome	tipo	datanascimento	idade	premio	id_funcionario
900	sergio	filho	2000-10-10	15	500	2005
901	maria	conjuge	1985-10-10	30	700	2005

```
#left join
```

```

drop table pagamento;
drop table fornecedor;
create table fornecedor(
    id int primary key,
    nome varchar (35)
);

```

```
create table pagamento(  
    idPagamento int primary key,  
    valor double,  
    datapagto date,  
    id_fornecedor int,  
    foreign key(id_fornecedor) references fornecedor(id)  
);
```

```
insert into fornecedor values (100, 'tulio Pecas');  
insert into fornecedor values (101, 'marcos Industria');  
insert into fornecedor values (102, 'carlos auto');
```

```
insert into pagamento values (1,1000, '2016-04-15',100);  
insert into pagamento values (2,800, '2016-03-15',100);  
insert into pagamento values (3,500, now(),102);  
insert into pagamento values (4,500, '2016-03-15',102);
```

```
#furos de pagamentos (foram 2 meses)  
#inner join _ Exatidão  
# inner join
```

```
select id, nome , idpagamento, valor, datapagto  
from fornecedor f inner join pagamento p  
on f.id = p.id_fornecedor;
```

id	nome	idpagamento	valor	datapagto
100	tulio Pecas	1	1000	2016-04-15
100	tulio Pecas	2	800	2016-03-15
102	carlos auto	3	500	2016-04-18
102	carlos auto	4	500	2016-03-15

```
# left join
```

```
#MOstra TODOS OS FORNECEDORES .... MESMO OS QUE NÃO FORAM PAGOS
```

```
select id, nome , idpagamento, valor, datapagto  
from fornecedor f left join pagamento p  
on f.id = p.id_fornecedor;
```

```
#Olha parça tabela antes do JOIN (left join)  
# olha para tabela fornecedor, mostra os que não receberam
```



id	nome	idpagamento	valor	datapagto
100	tulio Pecas	1	1000	2016-04-15
100	tulio Pecas	2	800	2016-03-15
101	marcos Industria	NULL	NULL	NULL
102	carlos auto	3	500	2016-04-18
102	carlos auto	4	500	2016-03-15

```

create table pai(
  id int primary key,
  nome varchar (35)
);

create table filho(
  idFilho int primary key,
  nomefilho varchar (35),
  id_pai int,
  foreign key(id_pai) references pai(id)
);

insert into pai values (100,'carlos' );
insert into pai values (115,'manoel' );

insert into filho values (1,'ricardo',100);
insert into filho values (2,'maria',null);
insert into filho values (3,'carina',null);
insert into filho values (4,'estevan',100);

-- pai Left join filho

select id, nome, nomefilho
  from pai p left join filho f on p.id = f.id_pai;

```

id	nome	nomefilho
100	carlos	ricardo
100	carlos	estevan
115	manoel	NULL

+-----+-----+-----+

```
-- pai right join filho
select id, nome, nomefilho
  from pai p right join filho f on p.id = f.id_pai;
```

```
+-----+-----+-----+
| id   | nome  | nomefilho |
+-----+-----+-----+
| 100  | carlos | ricardo   |
| NULL | NULL  | maria     |
| NULL | NULL  | carina    |
| 100  | carlos | estevan   |
+-----+-----+-----+
```



Escola de Nerds

[www.cotiinformatica.com.br](http://www.cotiinformatica.com.br)