

**ESCOLA SUPERIOR DE ADMINISTRAÇÃO MARKETING E COMUNICAÇÃO
(ESAMC)**

BRUNO VICTOR DA SILVA VICENTE

BANCO DE DADOS - TEORIA

SANTOS 2020

BRUNO VICTOR DA SILVA VICENTE

BANCO DE DADOS-TEORIA

Trabalho realizado individualmente e apresentado a instituição ESAMC para a disciplina de Banco de dados - teoria do curso de engenharia ministrado pelo Prof. Joseffe

Orientador: Joseffe

SANTOS 2020

RESUMO

Este trabalho visa aprender a linguagem SQL, no decorrer do semestre aprendemos na teoria e na pratica a lidar com um banco relacional. Foi possível compreender a importância do banco de dados, pois em qualquer área que lide com tecnologia e tenha seus próprios bancos de dados, poderá exigir conhecimento em SQL como complemento profissional.

Palavra - chaves: SQL, Programação, Banco de dados relacional

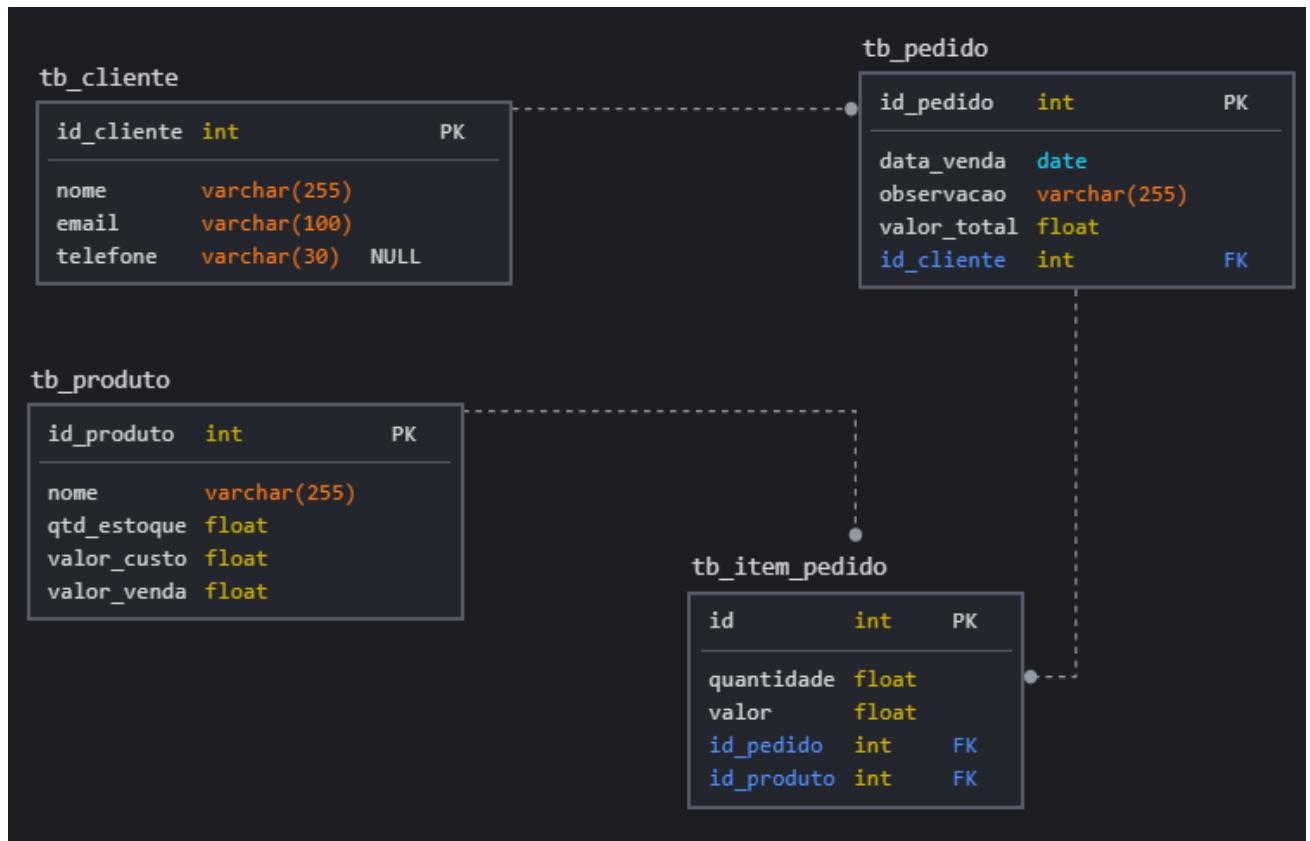
ABSTRACT

This work aims to learn the SQL language, during the semester we learn in theory and in practice to deal with a relational bank. It was possible to understand the importance of the database, because in any area that deals with technology and has its own databases, it may require knowledge in SQL as a professional complement.

Keyword: SQL, Programming, Relational database

1. PROJETO SISTEMA DE VENDAS

- Modelagem de dados



- Script SQL de criação das tabelas

```
create database db_faculdade;
```

```
drop database db_faculdade;
```

```
use db_faculdade;
```

```
create table tb_cliente(  
  id_cliente int not null auto_increment,  
  nome varchar(255) not null,  
  email varchar(100) not null,  
  telefone varchar(30) null,  
  primary key(id_cliente)  
);
```

```
create table tb_produto(  
id_produto    int          not null auto_increment,  
nome          varchar(255) not null,  
qtd_estoque   float        not null,  
valor_custo   float        not null,  
valor_venda   float        not null,  
primary key(id_produto)  
);
```

```
create table tb_pedido(  
id_pedido     int          not null auto_increment,  
id_cliente    int          not null,  
data_venda    date        not null,  
observacao    varchar(255) not null,  
valor_total   float        not null,  
primary key(id_pedido),  
foreign key(id_cliente) references tb_cliente(id_cliente)  
);
```

```
create table tb_item_pedido(  
id            int    not null auto_increment,  
id_pedido     int    not null,  
id_produto    int    not null,  
quantidade    float not null,  
valor         float not null,  
primary key(id),  
foreign key(id_pedido) references tb_pedido(id_pedido),  
foreign key(id_produto) references tb_produto(id_produto)  
);
```

```
show tables;
```

- **Ingestão/Inclusão de dados nas tabelas**

-- para inserir na tabela tb_cliente --

```
INSERT INTO tb_cliente (nome,email,telefone)
values ('Bruno Victor','bruno.victor32@gmail.com.br','13988123456'),
       ('Matheus Henrique','matheus32@gmail.com.br','13988222222'),
       ('Gracineide','gracineide32@gmail.com.br','13988333333'),
       ('Adrianooooo','adriano32@gmail.com.br','13988444444'),
       ('Thiago','thiago32@gmail.com.br',null);
```

```
select * from tb_cliente;
```

-- para inserir na tabela tb_produto --

```
INSERT INTO tb_produto (nome,qtd_estoque,valor_custo,valor_venda)
VALUES   ('Assolan'    ,10, 06.00, 10.00),
        ('Detergente' ,25,10.00 ,15.00),
        ('Amaciante'  ,11, 15.00, 20.00),
        ('Detergente' ,25,20.00 ,25.00),
        ('Omo'        ,30, 30.00,35.00),
        ('Desinfetante',4, 21.00,25.00);
```

```
select * from tb_produto;
```

-- para inserir na tabela tb_pedido --

```
INSERT INTO tb_pedido (id_cliente,data_venda,observacao,valor_total)
VALUES      ('4','2012-08-21','Por favor entregar logo',50.00),
            ('2','2013-09-22','Por favor entregar ',60.00),
            ('3','2014-10-27','Por favor ',81.00),
            ('5','2016-08-23','Por favor entregar logo',99.99);
```

```
select * from tb_pedido;
```

-- para inserir na tabela tb_item_pedido --

```
INSERT INTO tb_item_pedido (id_pedido,id_produto,quantidade,valor)
VALUES ('4','1',3, 40.00),
      ('1','3',8,40.00),
      ('2','2',9, 65.00),
      ('3','5',10, 90.00);
```

- **Exibição dos dados das tabelas**

-- Exibir os Clientes --

```
select * from tb_cliente;
```

-- Exibir todos os Clientes que não tem telefone --

```
SELECT id_cliente,nome FROM tb_cliente where (telefone is null);
```

-- Exibir todos os Clientes que tem telefone --

```
SELECT id_cliente,nome FROM tb_cliente where (telefone is not null);
```

-- Exibir os Produtos --

```
select * from tb_produto;
```

```
select nome from tb_produto;
```

-- Exibir todos os Produtos que tem o valor de custo superior a 10 reais e o valor de venda inferior a 20 reais --

```
select nome from tb_produto where (valor_venda > 10.00) and (valor_venda < 20.00);
```

-- Exibir todos os Produtos que tem o valor de custo superior a 20 reais e quantidade de estoque menor que 5 peças --

```
select nome from tb_produto where (valor_custo > 20.00) and (qtd_estoque < 5);
```

-- Exibir as informações de Pedidos, o Nome do Cliente e o Email --

```
select a.observacao as Informação,
```

```
       b.nome as Nome,
```


b.email as Email

from tb_pedido a inner join tb_cliente b on (a.id_pedido = b.id_cliente);

-- Exibir as informações do Id do Pedido, Nome do Produto, Nome do Cliente e Telefone --

select a.id_pedido as IdPedido,

c.nome as NomeProduto,

d.nome as NomeCliente,

d.telefone as NumeroTelefone

from tb_pedido a inner join tb_item_pedido b on (a.id_pedido = b.id_pedido)

inner join tb_produto c on (b.id_produto = c.id_produto)

inner join tb_cliente d on (a.id_cliente = d.id_cliente);

-- Exibir as informações do Id do Pedido, Nome do Produto,
Quantidade comprada, Valor de Venda e Nome do Cliente --

select a.id_pedido as IdPedido,

c.nome as NomeProduto,

b.quantidade as QuantidadeComprada,

c.valor_venda as ValorVenda,

d.nome as NomeCliente,

d.telefone as NumeroTelefone

from tb_pedido a inner join tb_item_pedido b on (a.id_pedido =

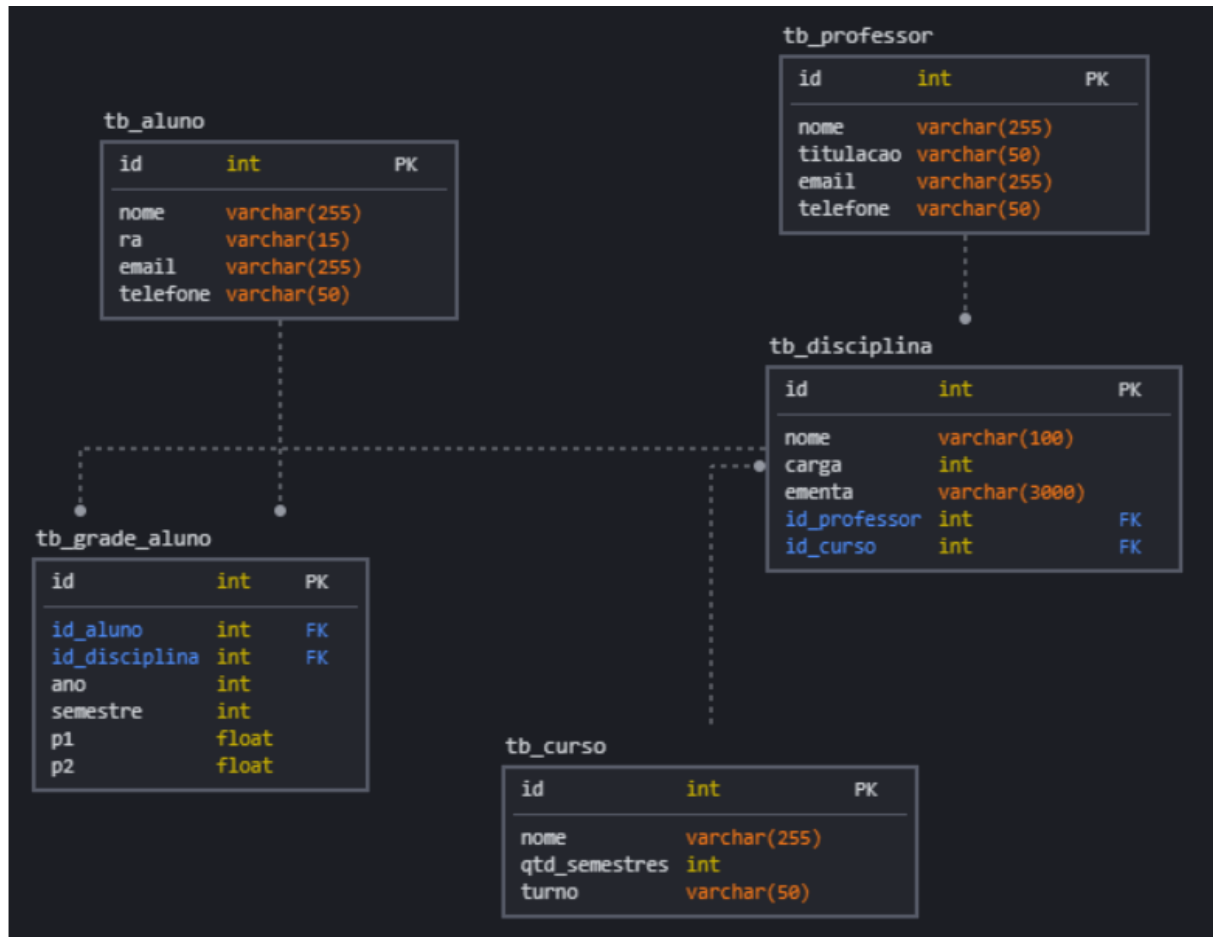
b.id_pedido)

inner join tb_produto c on (b.id_produto = c.id_produto)

inner join tb_cliente d on (a.id_cliente = d.id_cliente);

2. PROJETO SISTEMA DE NOTAS E DISCIPLINAS DA FACULDADE

- Modelagem de Dados



- Script SQL de criação de tabelas

```
create database sist_notas_disc;
```

```
use sist_notas_disc;
```

```
create table tb_aluno
```

```
(
```

```
    id      int auto_increment not null,
```

```
    nome    varchar(255)      not null,
```

```
    ra      int               not null,
```

```
    email   varchar(255)      not null,
```

```
    telefone varchar(50)      not null,
```

```
primary key (id)
);
```

```
create table tb_professor
(
    id          int auto_increment not null,
    nome        varchar(255)        not null,
    titulacao   varchar(255)        not null,
    email       varchar(255)        not null,
    telefone    varchar(50)         null,
    primary key (id)
);
```

```
create table tb_curso
(
    id          int auto_increment not null,
    nome        varchar(255)        not null,
    qtd_semestres int              not null,
    turno       varchar(50)         not null,
    primary key (id)
);
```

```
create table tb_disciplina
(
    id          int auto_increment not null,
    nome        varchar(100)        not null,
    carga       int                 not null,
    ementa      varchar(3000)       not null,
    id_professor int                not null,
    foreign key (id_professor) references tb_professor (id),
    id_curso     int                not null,
    foreign key (id_curso) references tb_curso (id),
    primary key (id)
);
```

```

create table tb_grade_aluno
(
    id            int auto_increment not null,
    ano           int                not null,
    semestre     int                not null,
    p1           float              null,
    p2           float              null,
    id_aluno     int                not null,
    foreign key (id_aluno) references tb_aluno (id),
    id_disciplina int                not null,
    foreign key (id_disciplina) references tb_disciplina (id),
    primary key (id)
);

```

- **Ingestão/Inclusão de dados nas tabelas**

```

insert into tb_professor (nome, titulacao, email, telefone)
values ('Renato', 'Mestre', 'renato@gmail.com', '1234566789'),
      ('Joseffe', 'Mestre', 'joseffe@gmail.com', '1234566789'),
      ('Anderson Brasil', 'Mestre', 'brasil@gmail.com', '1234566789'),
      ('Claudio', 'Doutor', 'claudio@gmail.com', '1234566789'),
      ('Amália', 'Doutor', 'amalia@gmail.com', '1234566789');

```

```

insert into tb_aluno (nome, ra, email, telefone)
values ('João', '12345', 'joao@gmail.com', '1122334455'),
      ('Maria', '22345', 'maria@gmail.com', '1122334455'),
      ('José', '32345', 'jose@gmail.com', '1122334455'),
      ('Luizinho', '42345', 'luizinho@gmail.com', '1122334455'),
      ('Iguinho', '52345', 'iguinho@gmail.com', '1122334455');

```

```

insert into tb_curso (nome, qtd_semestres, turno)
values('Engenharia da Computação', 10, 'Manhã'),
      ('Engenharia da Computação', 10, 'Noite'),

```

```

('Engenharia Elétrica', 10, 'Manhã'),
('Engenharia Elétrica', 10, 'Noite'),
('Engenharia Mecânica', 10, 'Manhã'),
('Engenharia Mecânica', 10, 'Noite'),
('Engenharia Química', 10, 'Manhã'),
('Engenharia Química', 10, 'Noite');

```

```

insert into tb_disciplina (nome, carga, ementa, id_professor, id_curso)
values ('Banco de Dados', 80, 'Ementa...', 2, 2),
      ('Programação I', 40, 'Ementa...', 2, 2),
      ('Programação II', 80, 'Ementa...', 1, 2),
      ('Arquitetura de Computadores', 80, 'Ementa...', 2, 4),
      ('Interface Humano Computador', 40, 'Ementa...', 2, 5),
      ('Compiladores', 40, 'Ementa...', 2, 1),
      ('Tecnologia Web', 40, 'Ementa...', 1, 2),
      ('Programação Orientada à Objetos', 40, 'Ementa...', 2, 3);

```

```

insert into tb_grade_aluno (ano, semestre, p1, p2, id_aluno,
id_disciplina)
values (2020, 1, null, null, 4, 5),
      (2020, 1, null, null, 4, 2),
      (2020, 1, null, null, 4, 4),
      (2020, 1, null, null, 2, 1),
      (2020, 1, null, null, 2, 3),
      (2020, 1, null, null, 2, 4),
      (2020, 1, null, null, 5, 5),
      (2020, 1, null, null, 5, 4),
      (2020, 1, null, null, 5, 2);

```

- **Exibição dos dados das tabelas**

-- - Exibir os professores e sua titulação --

```

select nome,
       titulacao

```

```
from tb_professor
```

```
-- -----
```

```
-- - Exibir as disciplinas de um determinado curso --
```

```
select id,  
       nome  
from tb_disciplina  
where id_curso = 1
```

```
/*Aqui vai exibir as disciplinas por exemplo do curso "engenharia da  
computação" ou seja o id_curso que vai ser o filtro*/
```

```
-- -----
```

```
-- - Exibir as disciplinas com carga superior a 40h --
```

```
select id,  
       nome  
from tb_disciplina  
where carga > 40
```

```
-- -----
```

```
-- - Exibir os professores de um determinado curso --
```

```
select b.id,  
       b.nome  
from tb_disciplina a inner join tb_professor b on (a.id_professor = b.id)  
where a.id_curso = 2  
group by b.id,  
       b.nome
```

```
/*Group by vai retirar repetições ou seja não vai mais aparecer professores  
repetidos igual abaixo*/
```

/*Aqui eu estou exibindo os professores da engenharia da computação*/

/*OU*/

```
select b.id,  
       b.nome  
from tb_disciplina a inner join tb_professor b on (a.id_professor = b.id)  
           inner join tb_curso c on (a.id_curso = b.id)  
where a.id_curso = 2
```

/*Aqui eu estou exibindo os professores da engenharia da computação,aqui eu pego os professores da tabela tb_disciplina si mais de um professor dá aula para mais de uma disciplina ele vai aparecer 2 vezes*/

/*A tabela tb_disciplina e a primeira do select por que a partir dela eu chego nas outras tabelas, si eu começar da tabela tb_professor eu não consigo chegar na tabela tb_disciplina

devido a tabela tb_professor não ter uma chave estrangeira.Quem tem as chaves estrangeiras e a tabela tb_disciplina relacionadas com a tb_professor e tb_curso*/

/*Nota: na tabela tb_disciplina existe 2 chaves estrangeiras "id_professor e id_curso" si esses campos forem not null eu vou utilizar inner join,si for null vou left join*/

-- -----

-- - Exibir todos os alunos matriculados em uma determinada disciplina ordenado por nome --

```
select b.nome  
from tb_grade_aluno a inner join tb_aluno b on (a.id_aluno = b.id)  
where a.id_disciplina = 4
```

order by b.nome

/*order by = ordene por ...*/

/*OU*/

```
select c.nome
from tb_grade_aluno a inner join tb_disciplina b on (a.id_disciplina = b.id)
              inner join tb_aluno c on (a.id_aluno = c.id)
where a.id_disciplina = 4
order by c.nome
```

-- -----

-- - Exibir todos os alunos que tiveram média superior a 8,5 no primeiro semestre de 2020 --

```
select b.nome
from tb_grade_aluno a inner join tb_aluno b on (a.id_aluno = b.id)
where ((a.p1+a.p2)/2) > 8.5
      and a.ano = 2020
      and a.semestre = 1
```

/*Para colocar mais filtros coloco um "and"*/

/*Não aparece notas devido eu ter deixado nulo, si eu tivesse inserido as notas tinha aparecido*/

/*Esse "a" que colocamos ai e chamado em inglês de alian que significa apelido*/

-- -----

-- - Exibir o nome do professor e disciplina dos alunos que tiveram média igual a 10 no primeiro semestre de 2020 --

```
select c.nome,
       b.nome
from tb_grade_aluno a inner join tb_disciplina b on (a.id_disciplina = b.id)
```



```
inner join tb_professor c on (b.id_professor = c.id)
where a.p1 = 10
and a.p2 = 10
and a.ano = 2020
and a.semestre = 1
```

-- - Exibir a quantidade de disciplinas alocadas para cada professor --

```
select b.nome,
       count(a.nome)
from tb_disciplina a inner join tb_professor b on (a.id_professor = b.id)
group by b.nome
```

/*O group by,ele retira as repetições,mas quando ele retira as repetições,ele pode alem de tirar as repetições,ele guarda a quantidade de vezes que repetiu para cada professor,ele guarda isso.Esse guardar vou exibir em razão do count*/

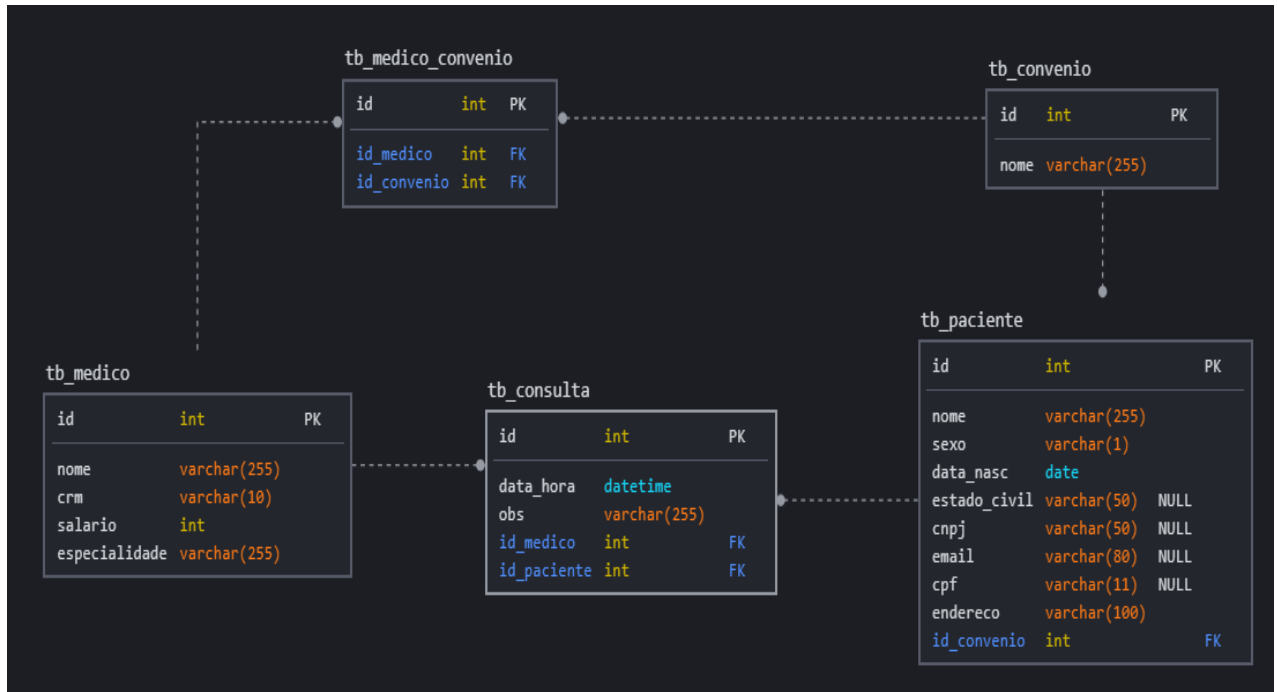
-- - Exibir a quantidade de disciplinas que cada aluno está cursando no primeiro semestre de 2020 --

```
select b.nome,
       count(c.nome)
from tb_grade_aluno a inner join tb_aluno b on (a.id_aluno = b.id)
                    inner join tb_disciplina c on (a.id_disciplina = c.id)
where a.ano = 2020
and a.semestre = 1
group by b.nome
```

```
-- Exibir todos os alunos matriculados em uma determinada disciplina --  
select c.nome  
from tb_grade_aluno a inner join tb_disciplina b on (a.id_disciplina = b.id)  
                    inner join tb_aluno    c on (a.id_aluno = c.id)  
where a.id_disciplina = 4
```

3. PROJETO DE SISTEMA PARA CONSULTÓRIO MÉDICO

- Modelagem de dados



- Script SQL de criação de tabelas

```
create database db_consultorio;
```

```
drop database db_consultorio;
```

```
use db_consultorio;
```

```
create table tb_convenio(
```

```
id      int          not null auto_increment,
```

```
nome varchar(255) not null,
```

```
primary key(id)
```

```
);
```

```
create table tb_medico(
```

```
id      int          not null auto_increment,
```

```
nome          varchar(255) not null,  
crm           varchar(10) not null,  
salario       int          not null,  
especialidade varchar(255) not null,  
primary key(id)
```

```
);
```

```
create table tb_paciente(
```

```
id            int          not null auto_increment,  
nome          varchar(255) not null,  
sexo          varchar(1)   not null,  
data_nasc     date         not null,  
estado_civil  varchar(50)  null,  
cnpj          varchar(50)  null,  
email         varchar(80)  null,  
cpf           varchar(11)  null,  
endereco      varchar(100) not null,  
id_convenio   int          null,
```

```
primary key(id),
```

```
foreign key(id_convenio) references tb_convenio(id)
```

```
);
```

```
create table tb_medico_convenio(
```

```
id            int not null auto_increment,  
id_medico     int not null,  
id_convenio   int not null,
```

```

primary key(id),

foreign key (id_medico) references tb_medico(id),

foreign key (id_convenio) references tb_convenio(id)

);

create table tb_consulta(

id            int            not null auto_increment,

data_hora     datetime      not null,

obs           varchar(255) null,

id_paciente   int            not null,

id_medico     int            not null,

primary key(id),

foreign key(id_paciente) references tb_paciente(id),

foreign key(id_medico) references tb_medico(id)

);

```

- **Ingestão/Inclusão de dados nas tabelas**

```
insert into tb_convenio(nome)
```

```
values ('Bradesco'),
```

```
      ('SulAmerica),
```

```
      (Unimed),
```

```
      ('Amil'),
```

```
      ('BioVida),
```

```
      ('Plena);
```

```
insert into tb_medico (nome,crm,salario,especialidade)
```

```
values ('Francisco','11111',1000.00,'Dentista'),
```

('Guilherme','21111',2000.00,'Ginecologista'),

('Pablo','31111',3000.00,'Dermatologista'),

('Flash','41111',4000.00,'Clinico Geral'),

('Chico','51111',5000.00,'Urologista');

insert into tb_paciente (nome, sexo, data_nasc, estado_civil, cnpj, email, cpf, endereco,id_convenio)

values ('Bruno Victor','M','1996-11-

24',null,'11462','bruno.victor32@yahoo.com.br','12345678910','Rua:Monteiro da cruz','1'),

('Claudia','F','2000-10-

22','casada','12463','claudia@yahoo.com.br','11111111111','Rua:Monteiro do cuscuz','3'),

('Fernanda','F','2001-09-

19','solteira','13464','fernanda@yahoo.com.br','2222222222','Rua:Monteiro da Picanha','2'),

('Wesley','M','1959-08-

18','divorciado','15468','wesley@yahoo.com.br','3333333333','Rua:Monteiro da alcatra','5'),

('Cleber','M','2003-07-

17','amigado','16469','cleber@yahoo.com.br','44444444444','Rua:Monteiro do abacate','5'),

('Maiki','M','1997-11-24',null,'13462',null,'12345678910','Rua:Monteiro da cruz','5')

('Cleide','F','1960-11-

24',null,'11462','bruno.victor32@yahoo.com.br','12345678910','Rua:Monteiro da cruz','1');

```
insert into tb_medico_convenio (id_medico,id_convenio)
```

```
values ('1','2'),  
       ('2','1'),  
       ('3','2'),  
       ('1','3'),  
       ('3','3'),  
       ('4','2'),  
       ('2','3');
```

```
insert into tb_consulta (data_hora,obs,id_paciente,id_medico)
```

```
values ('2002-08-18 14:29:36','Dentista','1','1'),  
       ('2002-08-18 14:29:36','Dentista','3','1'),  
       ('2002-08-18 14:29:36','Dentista','2','1'),  
       ('2002-08-18 14:29:36','Dentista','4','1'),  
       ('2002-08-18 14:29:36','Dentista','5','1'),  
       ('2002-08-18 14:29:36','Dentista','3','1'),  
       ('2011-09-18 16:29:36','Ginecologista','2','2'),  
       ('2010-10-18 15:29:36','Raio-x','3','2');
```

- **Exibição dos dados das tabelas**

```
select*from tb_convenio;
```

```
select*from tb_medico_convenio;
```

```
select*from tb_consulta;
```

```
select*from tb_paciente;
```

```
select*from tb_medico;
```

-- - Exibir os Pacientes e seu respectivo convênio --

```
select a.nome,  
       b.id,  
       b.nome  
from   tb_paciente a inner join tb_convenio b on (a.id_convenio = b.id);
```

```
select a.nome,  
       b.nome  
from   tb_paciente a left join tb_convenio b on (a.id_convenio = b.id);
```

-- - Exibir todos os Pacientes que não tem email --

```
select id,  
       nome  
from   tb_paciente  
where  email is null;
```

-- - Exibir os Convênios --

```
select nome  
from   tb_convenio;
```

-- - Exibir todos os Médicos e os convênios que cada médico atende --

```
select
b.nome
,c.nome
from tb_medico_convenio a inner join tb_medico b on (a.id_medico = b.id)
                                inner join tb_convenio c on (a.id_convenio =
c.id);
```

-- - Exibir a quantidade de médicos para cada convênio --

```
select b.nome as "Convenio",
       count(*) as "Quantidade de medico para cada convenio"
from tb_medico_convenio a inner join tb_convenio b on (a.id_convenio = b.id)
group by b.nome;
```

-- - Exibir a quantidade de consultas de cada médico --

```
select b.nome as "medico",
       count(a.id_paciente) as "consultas"
```

```
from tb_consulta a inner join tb_medico b on (a.id_medico = b.id)

group by b.nome;
```

```
-- Exibir a quantidade de consultas de cada paciente --
```

```
select b.nome as "Paciente",
       count(a.id_paciente) as "consultas"
from tb_consulta a inner join tb_paciente b on (a.id_paciente = b.id)
group by b.nome;
```

```
-- Exibir todos os pacientes de um determinado médico --
```

```
select
a.nome
,b.nome
from tb_medico a inner join tb_paciente b inner join tb_consulta c on (a.id =
c.id_medico and b.id = c.id_paciente ) ;
```

```
-- - Exibir todas as consultas realizadas no mês de outubro para um determinado
médico --
```

```
SELECT * FROM tb_consulta WHERE MONTH(data_hora) = 10
and id_medico = 1;
```

-- - Exibir todas as consultas realizadas na terceira semana do mês de outubro, contendo o paciente, convênio e médico --

```
select a.id_paciente,
       a.id_medico,
       b.nome,
       c.nome
from tb_consulta a inner join tb_paciente b on (a.id_paciente = b.id)
              inner join tb_medico c on (a.id_medico = c.id)
where (FLOOR((DAYOFMONTH(a.data_hora) - 1) / 7) + 1) = 3 ;
```

-- - Exibir todos os pacientes acima de 60 anos e seus médicos --

```
select b.id
       , b.nome as paciente
       , c.nome as medico
from tb_consulta a inner join tb_paciente b on (a.id_paciente = b.id)
              inner join tb_medico c on (a.id_medico = c.id)
where timestampdiff(year, data_nasc, now()) >60;
```

-- Exibir qual é o convênio que possui mais médicos vinculados --

```
select id_convenio, count(id_convenio) as quantidade from tb_medico_convenio
group by id_convenio;
```

