**REVISÃO DE BANCO DE DADOS #01**

show databases;  
create database clinica;  
use clinica;  
create table ambulatorio (  
    nroa int primary key auto\_increment,  
    andar int not null,  
    capacidade smallint  
);  
/\* escreve o texto \*/  
insert into ambulatorio (andar,capacidade)   
values                  ( 1,30);  
insert into ambulatorio (andar,capacidade)   
values                  ( 1,50);  
select \* from ambulatorio;  
insert into ambulatorio (andar,capacidade)   
values                  (2,40),  
                        (2,25),  
                        (2,55);  
Create table paciente (  
        codp int primary key auto\_increment,  
        nome varchar(40) not null,  
        idade smallint not null,  
        cidade char(30),  
        cpf varchar(11) unique,  
        doenca varchar(40)  
  );                        
  select \* from paciente;  
insert into paciente (nome,idade,cidade,cpf,doenca) values   
                ('Ana',20,'Florianopolis','20000200000','gripe');  
insert into paciente (nome,idade,cidade,cpf,doenca) values   
                ('Paulo',24,'Palhoca','20000220000','Fratura');  
insert into paciente (nome,idade,cidade,cpf,doenca) values   
                ('Luiza',30,'Biguacu','22000200000','Tendinite');  
insert into paciente (nome,idade,cidade,cpf,doenca) values   
                ('Carlos',28,'Joinville','11000220000','Sarampo');  
create table medico (  
    codm int primary key auto\_increment,  
    nome varchar(40) not null,  
    idade smallint not null,  
    especialidade char(20),  
    cpf varchar(11) unique,  
    cidade varchar(30),  
    fknroa int,  
    foreign key(fknroa) references ambulatorio(nroa)  
);  
insert into medico (nome,idade,especialidade,cpf,cidade,fknroa) values   
                  ('Joao',40,'ortopedia','10000100000','Florianopolis',1);  
select \* from medico;  
insert into medico (nome,idade,especialidade,cpf,cidade,fknroa) values   
                  ('Maria',42,'Traumatogia','10000110000','Blumenau',2);  
insert into medico (nome,idade,especialidade,cpf,cidade,fknroa) values   
                  ('Pedro',51,'Pediatria','11000100000','São José',2);  
insert into medico (nome,idade,especialidade,cpf,cidade,fknroa) values   
                  ('Carlos',28,'ortopedia','11000110000','Joinville',4);  
insert into medico (nome,idade,especialidade,cpf,cidade,fknroa) values   
                  ('Marcia',33,'Neurologia','11000111000','Biguacu',3);  
Create table Funcionario (  
        codf int primary key auto\_increment,  
        nome varchar(40) not null,  
        idade smallint not null,  
        cidade char(30),  
        cpf varchar(11) unique,  
        salario decimal(10,2),  
        cargo varchar(20)  
  );                      
insert into funcionario (nome,idade,cidade,cpf,salario) values   
                  ('Rita',32,'Sao Jose','11000111000',1200.45);  
select \* from funcionario;

O código MySQL abaixo cria um banco de dados de clínica com as seguintes tabelas:

* ambulatorio: Armazena informações sobre os ambulatórios da clínica, como o andar e a capacidade.
* paciente: Armazena informações sobre os pacientes da clínica, como o nome, a idade, a cidade, o CPF e a doença.
* medico: Armazena informações sobre os médicos da clínica, como o nome, a idade, a especialidade, o CPF, a cidade e o ambulatório em que trabalham.
* funcionario: Armazena informações sobre os funcionários da clínica, como o nome, a idade, a cidade, o CPF, o salário e o cargo.

O código é dividido em quatro partes:

1. Declaração de variáveis e inicialização: A primeira parte declara as variáveis necessárias e inicializa as conexões com o banco de dados.
2. Criação do banco de dados: A segunda parte cria o banco de dados clinica.
3. Criação das tabelas: A terceira parte cria as tabelas ambulatorio, paciente, medico e funcionario.
4. Inserção de dados: A quarta parte insere dados nas tabelas.

Explicação detalhada:

Declaração de variáveis e inicialização:

SQL

show databases;

create database clinica;

use clinica;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

A primeira linha exibe uma lista de todos os bancos de dados existentes. A segunda linha cria o banco de dados clinica. A terceira linha muda o contexto para o banco de dados clinica.

Criação do banco de dados:

SQL

create table ambulatorio (

nroa int primary key auto\_increment,

andar int not null,

capacidade smallint

);

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução cria a tabela ambulatorio com três campos:

* nroa: O número do ambulatório (chave primária auto-incrementável).
* andar: O andar do ambulatório.
* capacidade: A capacidade do ambulatório.

Criação das tabelas:

SQL

create table paciente (

codp int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

cidade char(30),

cpf varchar(11) unique,

doenca varchar(40)

);

create table medico (

codm int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

especialidade char(20),

cpf varchar(11) unique,

cidade varchar(30),

fknroa int,

foreign key(fknroa) references ambulatorio(nroa)

);

create table Funcionario (

codf int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

cidade char(30),

cpf varchar(11) unique,

salario decimal(10,2),

cargo varchar(20)

);

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Estas instruções criam as tabelas paciente, medico e funcionario com os seguintes campos:

* paciente:
  + codp: O número do paciente (chave primária auto-incrementável).
  + nome: O nome do paciente.
  + idade: A idade do paciente.
  + cidade: A cidade do paciente.
  + cpf: O CPF do paciente.
  + doenca: A doença do paciente.
* medico:
  + codm: O número do médico (chave primária auto-incrementável).
  + nome: O nome do médico.
  + idade: A idade do médico.
  + especialidade: A especialidade do médico.
  + cpf: O CPF do médico.
  + cidade: A cidade do médico.
  + fknroa: O número do ambulatório em que o médico trabalha (chave estrangeira para ambulatorio.nroa).
* funcionario:
  + codf: O número do funcionário (chave primária auto-incrementável).
  + nome: O nome do funcionário.
  + idade: A idade do funcionário.
  + cidade: A cidade do funcionário.
  + cpf: O CPF do
* **REVISÃO DE BANCO DE DADOS #02**

O código MySQL abaixo cria um banco de dados de clínica com as seguintes tabelas:

* ambulatorio: Armazena informações sobre os ambulatórios da clínica, como o andar e a capacidade.
* medico: Armazena informações sobre os médicos da clínica, como o nome, a idade, a especialidade, o CPF, a cidade e o ambulatório em que trabalham.
* paciente: Armazena informações sobre os pacientes da clínica, como o nome, a idade, a cidade, o CPF e a doença.
* funcionario: Armazena informações sobre os funcionários da clínica, como o nome, a idade, a cidade, o CPF, o salário e o cargo.
* consulta: Armazena informações sobre as consultas médicas, como a data, a hora, o médico e o paciente.

O código é dividido em 14 partes:

1. Declaração de variáveis e inicialização: A primeira parte declara as variáveis necessárias e inicializa as conexões com o banco de dados.
2. Criação do banco de dados: A segunda parte cria o banco de dados clinica.
3. Criação das tabelas: A terceira parte cria as tabelas ambulatorio, medico, paciente e funcionario.
4. Inserção de dados: A quarta parte insere dados nas tabelas.
5. Criação da tabela consulta: A quinta parte cria a tabela consulta.
6. Inserção de dados na tabela consulta: A sexta parte insere dados na tabela consulta.
7. Criação da tabela copia\_paciente: A sétima parte cria a tabela copia\_paciente como uma cópia da tabela paciente.
8. Descrição da tabela funcionario: A oitava parte exibe uma descrição da tabela funcionario.
9. Adição de uma coluna à tabela funcionario: A nona parte adiciona uma coluna fk\_nroa à tabela funcionario.
10. Adição de uma chave estrangeira à tabela funcionario: A décima parte adiciona uma chave estrangeira à tabela funcionario.
11. Atualização de um valor na tabela funcionario: A décima primeira parte atualiza um valor na tabela funcionario.
12. Remoção de uma coluna da tabela funcionario: A décima segunda parte remove uma coluna da tabela funcionario.
13. Atualização de um valor na tabela paciente: A décima terceira parte atualiza um valor na tabela paciente.
14. Consultas às tabelas: A décima quarta parte executa consultas às tabelas.

Explicação detalhada:

Declaração de variáveis e inicialização:

SQL

use clinica;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta linha muda o contexto para o banco de dados clinica.

Criação do banco de dados:

SQL

create database clinica;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução cria o banco de dados clinica.

Criação das tabelas:

SQL

create table ambulatorio (

nroa int primary key auto\_increment,

andar int not null,

capacidade smallint

);

create table medico (

codm int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

especialidade char(20),

cpf varchar(11) unique,

cidade varchar(30),

fknroa int,

foreign key(fknroa) references ambulatorio(nroa)

);

create table paciente (

codp int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

cidade char(30),

cpf varchar(11) unique,

doenca varchar(40)

);

create table funcionario (

codf int primary key auto\_increment,

nome varchar(40) not null,

idade smallint not null,

cidade char(30),

cpf varchar(11) unique,

salario decimal(10,2),

cargo varchar(20)

);

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Estas instruções criam as tabelas ambulatorio, medico, paciente e funcionario com os seguintes campos:

* ambulatorio:
  + nroa: O número do ambulatório (chave primária auto-

O código começa com a instrução use clinica;, que muda o contexto para o banco de dados clinica.

Em seguida, o código executa uma série de comandos select para exibir os dados das tabelas ambulatorio, medico, paciente e funcionario.

O código então insere dados nas tabelas funcionario e consulta.

A instrução create table consulta cria a tabela consulta, que armazena informações sobre as consultas médicas, como a data, a hora, o médico e o paciente.

A instrução insert into consulta insere dados na tabela consulta. Os dados inseridos representam 14 consultas, cada uma com um médico e um paciente específicos.

Aqui está uma explicação detalhada dos comandos SQL usados no código:

* use clinica;: Muda o contexto para o banco de dados clinica.
* select \* from ambulatorio;: Exibe todos os dados da tabela ambulatorio.
* select \* from medico;: Exibe todos os dados da tabela medico.
* select \* from paciente;: Exibe todos os dados da tabela paciente.
* select \* from funcionario;: Exibe todos os dados da tabela funcionario.
* insert into funcionario (nome,idade,cidade,cpf,salario) values ('Maria',55,'Palhoca','30000111000',1220.45);: Insere um novo registro na tabela funcionario.
* insert into funcionario (nome,idade,cidade,cpf,salario) values ('Caio',45,'Florianopolis','40000111000',1100.45);: Insere um novo registro na tabela funcionario.
* insert into funcionario (nome,idade,cidade,cpf,salario) values ('carlos',44,'Florianopolis','50000111000',1200.45);: Insere um novo registro na tabela funcionario.
* insert into funcionario (nome,idade,cidade,cpf,salario) values ('Paula',33,'Florianopolis','60000111000',2500.45);: Insere um novo registro na tabela funcionario.
* create table consulta (num\_consulta int primary key auto\_increment, data\_consulta date, hora time, fk\_codm int, foreign key(fk\_codm) references medico(codm), fk\_codp int, foreign key(fk\_codp) references paciente(codp));: Cria a tabela consulta.
* insert into consulta (fk\_codm,fk\_codp,data\_consulta,hora) values (1,1,"2006-06-12","14:00"), (1,4,"2006-06-13","10:00"), (2,1,"2006-06-13","09:00"), (2,2,"2006-06-13","11:00"), (2,3,"2006-06-14","14:00"), (2,4,"2006-06-14","17:00"), (3,1,"2006-06-19","18:00"), (3,3,"2006-06-12","10:00"), (3,4,"2006-06-19","13:00"), (4,4,"2006-06-20","13:00"), (4,4,"2006-06-22","19:30"), (4,1,"2006-06-19","14:00"), (3,1,"2006-06-22","14:00"), (2,1,"2006-06-25","14:00");: Insere dados na tabela consulta.
* use clinica;  
  select \* from ambulatorio;  
  select \* from medico;  
  select \* from paciente;  
  select \* from funcionario;  
  insert into funcionario (nome,idade,cidade,cpf,salario) values   
                    ('Maria',55,'Palhoca','30000111000',1220.45);  
  insert into funcionario (nome,idade,cidade,cpf,salario) values   
                    ('Caio',45,'Florianopolis','40000111000',1100.45);  
  insert into funcionario (nome,idade,cidade,cpf,salario) values   
                    ('carlos',44,'Florianopolis','50000111000',1200.45);  
  insert into funcionario (nome,idade,cidade,cpf,salario) values   
                    ('Paula',33,'Florianopolis','60000111000',2500.45);  
  create table consulta (  
      num\_consulta int primary key auto\_increment,  
      data\_consulta  date,  
      hora           time,  
      fk\_codm int,  
      foreign key(fk\_codm) references medico(codm),  
      fk\_codp int,  
      foreign key(fk\_codp) references paciente(codp)  
  );  
  insert into consulta (fk\_codm,fk\_codp,data\_consulta,hora) values   
                       (1,1,"2006-06-12","14:00"),  
                       (1,4,"2006-06-13","10:00"),  
                       (2,1,"2006-06-13","09:00"),  
                       (2,2,"2006-06-13","11:00"),  
                       (2,3,"2006-06-14","14:00"),  
                       (2,4,"2006-06-14","17:00"),  
                       (3,1,"2006-06-19","18:00"),  
                       (3,3,"2006-06-12","10:00"),  
                       (3,4,"2006-06-19","13:00"),  
                       (4,4,"2006-06-20","13:00"),  
                       (4,4,"2006-06-22","19:30"),  
                       (4,1,"2006-06-19","14:00"),  
                       (3,1,"2006-06-22","14:00"),  
                       (2,1,"2006-06-25","14:00");  
  select \* from consulta;  
  select \* from paciente;  
  create table copia\_paciente as select \* from paciente;  
  select \* from copia\_paciente;
* describe funcionario;  
  alter table funcionario add column fk\_nroa int;  
  alter table funcionario add foreign key(fk\_nroa) references ambulatorio(nroa);  
  select \* from funcionario;  
  update funcionario set fk\_nroa = 1 where codf = 1;
* alter table funcionario drop column cargo;
* select \* from paciente;  
  update paciente set cidade = "Ilhota" where codp = 2;
* select \* from paciente where codp = 2;  
  select \* from paciente where nome = "Paulo";  
  select \* from paciente where nome like "Paulo";  
  select \* from paciente where nome = "P%";  
  select \* from paciente where nome like "P%";
* select \* from consulta;  
  select \* from consulta where fk\_codm = 1 or fk\_codp = 4;  
  select \* from consulta where fk\_codm <> 1;  
  select \* from consulta where fk\_codm = 1 and fk\_codp = 4;  
  update consulta set data\_consulta = '2006-07-04', hora = '12:00' where fk\_codm = 1 and fk\_codp = 4;  
  update consulta set data\_consulta = '2006-07-04', hora = '12:00' where num\_consulta = 2;
* select \* from paciente;  
  update paciente set doenca = 'Hepatite', idade = idade +1 where codp = 1;
* select \* from consulta where fk\_codm = 3 and fk\_codp = 4;  
  update consulta set hora = addtime(hora,"01:30") where num\_consulta = 9;
* select \* from funcionario;  
  delete from funcionario where codf = 4;
* select \* from consulta where hora > '19:00';  
  delete from consulta where hora > '19:00';
* delete from paciente where doenca like 'Cancer' or idade < 10;
* delete from medico where cidade = 'Biguacu' or cidade = 'Palhoca';  
  delete from medico where cidade in('Biguacu','Palhoca');  
  select \* from medico where cidade in('Biguacu','Florianopolis');  
  select \* from medico;  
  select \* from medico where   
       especialidade in( select especialidade from medico where cidade <> 'Biguacu');
* O trecho de código que você forneceu é uma sequência de comandos SQL que executam consultas e atualizações nas tabelas consulta, paciente, funcionario e medico.
* O código começa com uma série de comandos select para exibir os dados das tabelas consulta e paciente.
* Em seguida, o código cria uma nova tabela chamada copia\_paciente com os mesmos dados da tabela paciente.
* O código então executa um comando describe para exibir a descrição da tabela funcionario.
* Em seguida, o código adiciona uma nova coluna chamada fk\_nroa à tabela funcionario. Esta coluna armazenará o número do ambulatório em que o funcionário trabalha.
* O código então adiciona uma chave estrangeira à coluna fk\_nroa da tabela funcionario. Esta chave estrangeira referencia a coluna nroa da tabela ambulatorio.
* O código então exibe os dados da tabela funcionario novamente.
* Em seguida, o código atualiza a coluna fk\_nroa da tabela funcionario para o valor 1 para o funcionário com o código 1.
* O código então remove a coluna cargo da tabela funcionario.
* Em seguida, o código exibe os dados da tabela paciente novamente.
* O código então atualiza a coluna cidade da tabela paciente para o valor "Ilhota" para o paciente com o código 2.
* O código então exibe os dados do paciente com o código 2.
* O código então exibe os dados de todos os pacientes com o nome "Paulo".
* O código então exibe os dados de todos os pacientes com o nome que começa com a letra "P".
* O código então exibe os dados de todos os pacientes com o nome que é igual a "P%".
* O código então exibe os dados de todos os pacientes com o nome que começa com a letra "P".
* O código então exibe os dados de todas as consultas.
* O código então exibe os dados de todas as consultas com o médico de código 1 ou com o paciente de código 4.
* O código então exibe os dados de todas as consultas com o médico de código diferente de 1.
* O código então exibe os dados de todas as consultas com o médico de código 1 e com o paciente de código 4.
* O código então atualiza a data e a hora da consulta de código 4 para o dia 04/07/2006 e a hora 12:00.
* O código então atualiza a data e a hora da consulta de código 2 para o dia 04/07/2006 e a hora 12:00.
* O código então exibe os dados da tabela paciente novamente.
* O código então atualiza a doença do paciente de código 1 para "Hepatite" e a idade do paciente de código 1 para a idade atual + 1.
* O código então exibe os dados de todas as consultas com o médico de código 3 e com o paciente de código 4.
* O código então atualiza a hora da consulta de código 9 para a hora atual + 1 hora e 30 minutos.
* O código então exibe os dados da tabela funcionario novamente.
* O código então deleta o funcionário de código 4.
* O código então exibe os dados de todas as consultas com a hora maior que 19:00.
* O código então deleta todas as consultas com a hora maior que 19:00.
* O código então deleta todos os pacientes com a doença "Cancer" ou com a idade menor que 10 anos.
* O código então deleta todos os médicos da cidade de Biguacu ou da cidade de Palhoca.
* O código então deleta todos os médicos das cidades de Biguacu e Palhoca, usando a cláusula in.
* O código então exibe os dados de todos os médicos da cidade de Biguacu ou da cidade de Florianópolis.
* O código então exibe os dados de todos os médicos.
* O código então exibe os dados de todos os médicos com uma especialidade que está na lista de especialidades dos médicos que não moram na cidade de Biguacu.

**REVISÃO BANCO DE DADOS #03**

use clinica;  
show tables;  
select \*   
from paciente;  
select codp as 'Numero do Prontuério',nome as 'Nome do Paciente',doenca  
from paciente;  
select \* from funcionario;  
/\* Gere um relatorio dos funcionario com nome, salario e a proposta do novo salario com reajuste  
    de 12% \*/  
      
select nome,salario from funcionario;  
select nome,  
       format(salario,2) as Salário,  
       format((salario \* 1.12),2) as 'Novo Salário'  
from funcionario;

select count(\*) as Total from funcionario;  
select \* from medico;  
select count(\*) from medico;  
select \* from medico;  
select \* from medico order by nome;  
select \* from medico order by nome desc;  
select \* from medico order by especialidade,nome;  
select especialidade,nome,idade from medico order by especialidade,nome;  
select especialidade,count(\*) from medico group by especialidade;

select \* from consulta;  
/\*  Qual o total de consultas agendadas pelo ANA \*/  
select \* from consulta;  
select \* from consulta;  
select fkcodp, data\_consulta, count(\*) as qtdConsultas   
from consulta where fkcodp=2 group by num\_consulta;  
select f as Total from consulta where fk\_codp = 1;

select consulta.fk\_codp as 'Codigo Paciente',  
        paciente.nome,  
        count(\*) as Total  
from consulta,paciente   
where consulta.fk\_codp = paciente.codp and consulta.fk\_codp = 2;

select consulta.fk\_codp as 'Codigo Paciente',  
        paciente.nome,  
        count(\*) as Total  
from consulta,paciente   
where consulta.fk\_codp = paciente.codp  
group by consulta.fk\_codp;

select \* from funcionario;  
select sum(salario) as 'Total dos Salarios' from funcionario;  
select format(avg(salario),2) as 'Total dos Salarios' from funcionario;

select \* from medico;  
select especialidade from medico group by especialidade;  
select especialidade, count(\*) from medico group by especialidade;  
select distinct especialidade from medico;  
select \* from paciente;  
select \* from funcionario;  
select distinct cidade from funcionario;

select \* from paciente where nome = 'Ana';  
select \* from paciente where nome like 'Ana';  
select \* from paciente where nome = 'A%';  
select \* from paciente where nome like 'A%';  
select \* from paciente;  
select \* from paciente where nome like '%A%';  
select \* from paciente where nome like '\_A%';  
alter table paciente add column senha varchar(50);  
select \* from paciente;  
update paciente set senha = 'Mxx123' where codp=1;  
update paciente set senha = sha('Mxx123') where codp=2;

select \* from paciente where nome not like 'A%';  
select \* from paciente;  
select \* from  paciente where senha is not null;  
select \* from paciente;

/\* gere uma listagem dos paciente com idade entre 24 e 35 anos \*/  
select \* from paciente where idade between 24 and 35; /\* errado \*/  
select \* from paciente where idade > 24 and idade < 35;

select \* from medico;  
/\* liste os medicos que moram nas cidades Blumenau e Joinville \*/  
select \* from medico where cidade = 'Blumenau' or  cidade = 'Joinville';  
select \* from medico where cidade in('Blumenau','Joinville');  
select \* from medico where cidade in(select nome from cidade);

Linha 1: Conecta à base de dados "clinica".

SQL

use clinica;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução conecta o MySQL à base de dados "clinica".

Linha 2: Lista todas as tabelas da base de dados.

SQL

show tables;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução lista todas as tabelas da base de dados.

Linha 3: Seleciona todas as colunas da tabela "paciente".

SQL

select \*

from paciente;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as colunas da tabela "paciente".

Linha 4: Seleciona as colunas "codp", "nome" e "doenca" da tabela "paciente" e define novos nomes para elas.

SQL

select codp as 'Numero do Prontuério',nome as 'Nome do Paciente',doenca

from paciente;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona as colunas "codp", "nome" e "doenca" da tabela "paciente" e define novos nomes para elas.

Linha 5: Seleciona todas as colunas da tabela "funcionario".

SQL

select \*

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as colunas da tabela "funcionario".

Linha 6-9: Gera um relatório dos funcionários com nome, salário e proposta do novo salário com reajuste de 12%.

SQL

select nome,salario

from funcionario;

select nome,

format(salario,2) as Salário,

format((salario \* 1.12),2) as 'Novo Salário'

from funcionario;

select count(\*) as Total

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Estas instruções geram um relatório dos funcionários com nome, salário e proposta do novo salário com reajuste de 12%.

Linha 10: Conta o número total de registros na tabela "funcionario".

SQL

select count(\*) as Total

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução conta o número total de registros na tabela "funcionario".

Linha 11-12: Seleciona todas as colunas da tabela "medico".

SQL

select \*

from medico;

select count(\*)

from medico;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Estas instruções selecionam todas as colunas da tabela "medico" e contam o número total de registros na tabela.

Linha 13-14: Ordena a tabela "medico" por nome.

SQL

select \*

from medico

order by nome;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução ordena a tabela "medico" por nome.

Linha 15-16: Ordena a tabela "medico" por nome decrescente.

SQL

select \*

from medico

order by nome desc;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução ordena a tabela "medico" por nome decrescente.

Linha 17-18: Ordena a tabela "medico" por especialidade e nome.

SQL

select \*

from medico

order by especialidade,nome;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução ordena a tabela "medico" por especialidade e nome.

Linha 19-20: Seleciona as colunas "especialidade", "nome" e "idade" da tabela "medico" e ordena a tabela por especialidade e nome.

SQL

select especialidade,nome,idade

from medico

order by especialidade,nome;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona as colunas "especialidade", "nome" e "idade" da tabela "medico" e ordena a tabela por especialidade e nome.

Linha 21-22: Conta o número de registros na tabela "medico" agrupados por especialidade.

SQL

select especialidade,count(\*)

from medico

group by especialidade;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução conta o número de registros na tabela "medico" agrupados por especialidade.

Linha 23-24: Seleciona todas as colunas da tabela "consulta".

SQL

select \*

from consulta;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as colunas da tabela "consulta".

Linha 25-26: Seleciona o número de consultas agendadas pelo paciente com código 2.

SQL

select count(\*) as qtdConsultas

from consulta

where fk\_codp=2;

Linha 26-27: Seleciona o número de consultas agendadas pelo paciente com código 2.

SQL

select count(\*) as qtdConsultas

from consulta

where fk\_codp=2;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona o número de consultas agendadas pelo paciente com código 2.

Linha 28: Seleciona o número de consultas agendadas pelo paciente com nome "Ana".

SQL

select count(\*) as qtdConsultas

from consulta

where nome='Ana';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona o número de consultas agendadas pelo paciente com nome "Ana".

Linha 29-30: Seleciona o número de consultas agendadas por cada paciente.

SQL

select consulta.fk\_codp as 'Codigo Paciente',

paciente.nome,

count(\*) as Total

from consulta,paciente

where consulta.fk\_codp = paciente.codp

group by consulta.fk\_codp;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona o número de consultas agendadas por cada paciente.

Linha 31-32: Seleciona a soma dos salários dos funcionários.

SQL

select sum(salario) as 'Total dos Salarios'

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona a soma dos salários dos funcionários.

Linha 33-34: Seleciona a média dos salários dos funcionários.

SQL

select format(avg(salario),2) as 'Total dos Salarios'

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona a média dos salários dos funcionários.

Linha 35-36: Seleciona todas as especialidades dos médicos.

SQL

select especialidade

from medico;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as especialidades dos médicos.

Linha 37-38: Seleciona todas as especialidades dos médicos, juntamente com o número de médicos por especialidade.

SQL

select especialidade,count(\*)

from medico

group by especialidade;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as especialidades dos médicos, juntamente com o número de médicos por especialidade.

Linha 39-40: Seleciona todas as especialidades distintas dos médicos.

SQL

select distinct especialidade

from medico;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as especialidades distintas dos médicos.

Linha 41-42: Seleciona todas as cidades dos funcionários.

SQL

select cidade

from funcionario;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as cidades dos funcionários.

Linha 43-44: Seleciona todos os pacientes com nome "Ana".

SQL

select \*

from paciente

where nome='Ana';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todos os pacientes com nome "Ana".

Linha 45-46: Seleciona todos os pacientes com nome que começa com a letra "A".

SQL

select \*

from paciente

where nome like 'A%';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todos os pacientes com nome que começa com a letra "A".

Linha 47-48: Seleciona todos os pacientes com nome que termina com a letra "a".

SQL

select \*

from paciente

where nome like '%a';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todos os pacientes com nome que termina com a letra "a".

Linha 49-50: Seleciona todos os pacientes com nome que contém a letra "A".

SQL

select \*

from paciente

where nome like '%A%';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todos os pacientes com nome que contém a letra "A".

Linha 51-52: Seleciona todos os pacientes com nome que contém a letra "A" no início.

SQL

select \*

from paciente

where nome like '\_A%';

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todos os pacientes com nome que contém a letra "A" no início.

Linha 53-54: Adiciona a coluna "senha" à tabela "paciente".

SQL

alter table paciente add column senha varchar(50);

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução adiciona a coluna "senha" à tabela "paciente".

Linha 55-56: Seleciona todas as colunas da tabela "paciente".

SQL

select \*

from paciente;

Use o código com cuidado. [Saiba mais](https://bard.google.com/faq#coding)

content\_copy

Esta instrução seleciona todas as coluna