

Link do vídeo: <https://www.youtube.com/watch?v=6NYkIRCr2cU>

Esquema do banco de dados:



Na primeira tabela foi verificado que a tabela consulta estava guardando dados que não faziam parte dela, como dados de pacientes e médicos, deixando-a muito confusa e sobrecarregada. Após uma verificação de como dividir e melhorar ela, substitui por 4 tabelas, as quais são consulta, paciente, médico e especialidades. Na tabela da consulta em si, ficou apenas os dados que lhe convim e 2 chaves estrangeiras apontado para paciente e médico. Na tabela médico, apenas seu nome ficou nela, visto que como um médico pode ter mais de 1 especialidade, seria o ideal criar uma nova tabela e gerar uma relação entre elas, guardando-as de forma a não duplicar duplas na tabela original. Na tabela paciente, os dados pessoais dele serão guardados, ocorreu apenas uma dúvida em relação ao telefone e endereço, mas como não indicava a possibilidade de ter mais de um telefone ou de ter que separar o endereço, optei por deixar dessa forma.

Na tabela receita, foi necessário apenas dividir em 2 partes, uma que cuidaria da receita em si e outra que cuidaria dos remédios, visto que, uma receita poderia ter mais de um medicamento. Dessa forma, uma única receita pode ter agregado ao seu ID inúmeros medicamentos, no qual tem o nome dele, dosagem e frequência para se tomar, além do tipo de exame, data e resultado.

Na tabela pagamento_exame, assim como na receita, dividi em duas partes, a primeira que consta apenas os exames que serão realizados, com uma chave estrangeira para nossa consulta, o qual juntaria ambas. Na outra, somente a parte de pagamento foi indexada nela, datas, valores e tipos de pagamento, além de uma ligação com a tabela exame.

Quanto a normalização todas as tabelas são aprovadas no 1FN, visto que para que não possa passar dessa fase, seria necessário que em alguma tabela eu não tivesse atomicidade nos meus atributos, por exemplo, caso o telefone fosse um multivalorado e eu não criasse uma tabela a parte para ele.

No caso da 2FN, para que ela não possa estar normalizada nesse padrão, seria necessário que eu tivesse pelo menos mais de 1 chave dentro da tabela, aí verificaríamos se ocorreria dependência funcional parcial em alguma delas, no entanto em todas as tabelas, há apenas 1 chave primária, sendo assim, não há motivos que levam a acreditar que não estão nessa norma.

A normalização 3FN, indica que não podemos ter atributos não-primos dependendo de não-primos, ao analisar todas as tabelas, verificamos que isso não ocorre em momento algum, sendo assim, se encontra normalizado até a 3FN.

Abaixo estarei deixando mais detalhadamente, tabela por tabela, as normalizações.

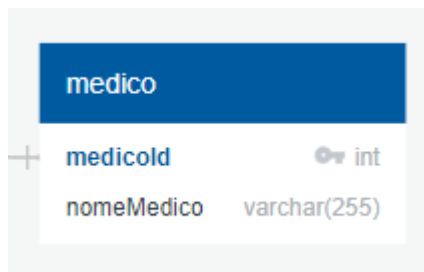
Tabela paciente



Paciente	
pacienteld	int
nomePaciente	varchar(255)
dataNascimentoPaciente	date
enderecoPaciente	varchar(255)
telefonePaciente	varchar(15)
emailPacinete	varchar(255)

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos (indivisíveis) e cada registro é único;
- **2NF:** A tabela está em 2NF, pois não possui dependências parciais (a tabela possui uma chave primária simples);
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária (pacienteld) e não há dependências transitivas.

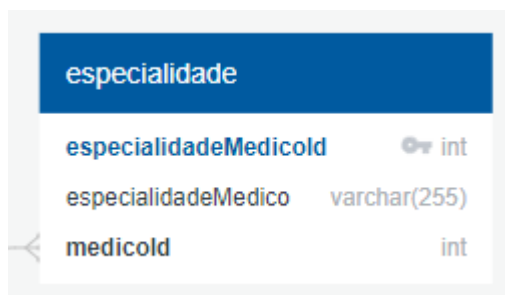
Tabela medico



medico	
medicoid	int
nomeMedico	varchar(255)

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois não possui dependências parciais (a tabela possui uma chave primária simples);
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária (medicoid).

Tabela especialidade



especialidade		
especialidadeMedicoid	int	
especialidadeMedico	varchar(255)	
medicoid		int

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é especialidadeMedicoid, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.

Tabela consulta

consulta	
consultald	int
dataConsulta	date
diagnostico	varchar(255)
tratamento	varchar(255)
medicoid	int
pacienteld	int

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é consultald, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.

Tabela receita

receita	
receitald	int
consultald	int
dataReceita	date

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é receitald, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.

Tabela medicamentos

medicamentos	
medicamentosId	int
receitaId	int
nome	varchar(255)
dosagem	varchar(255)
frequencia	varchar(255)

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é medicamentosId, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.

Tabela exame

exame	
exameld	int
tipoExame	varchar(255)
dataExame	varchar(255)
resultadoExame	varchar(255)
consultaId	int

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é exameld, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.

Tabela pagamentoExame

pagamentoExame	
pagamentold	int,
dataPagamento	date,
valorPagamento	decimal(10,2),
metodoPagamento	varchar(50),
exameld	int

- **1NF:** A tabela está em 1NF, pois todos os atributos contêm valores atômicos e cada registro é único;
- **2NF:** A tabela está em 2NF, pois a chave primária composta é pagamentold, e todos os atributos não chave dependem totalmente dela;
- **3NF:** A tabela está em 3NF, pois todos os atributos não chave são totalmente dependentes da chave primária e não há dependências transitivas.