

Visões

Banco de Dados
Carlos Arruda Baltazar
UNIP – Cidade Universitária

- Visão (do inglês: *View*) é um resultado originado de uma consulta pré-definida. Essencialmente é um metadado que mapeia uma *query* para outra, por isto pode ser considerado como uma tabela virtual. Como o próprio nome diz, ela representa uma **visão** de dados e não contém dados.

- É basicamente um texto de um SELECT. Pense que ela é uma query que você armazena em uma variável e ao invés de escrever toda a query de novo sempre que for necessária usa a variável.

- Essas tabelas virtuais são dinâmicas, elas só existem no momento em que você precisa delas. Somente a *query* para sua geração é pré-definida. Se um dado for atualizado logo em seguida uma *view* for acessada, essa atualização não será considerada neste momento, somente na próxima vez que a *view* for usada.

- Uma *View* funciona de forma semelhante a uma tabela. É utilizada em comandos ***SELECT***, ***INSERT***, ***UPDATE*** e ***DELETE***, para recuperação e manipulação de dados (com restrições), porém, não armazena esses dados.

- Este objeto tem suas linhas e colunas calculadas dinamicamente através de um ***SELECT*** pré-estabelecido, cada vez que solicitamos. Apenas a sua definição é armazenada no dicionário de dados.

- Você pode usar este resultado em outras consultas diminuindo a complexidade, afinal você fará referência a uma tabela virtual montada fora desta consulta.
- Estas **consultas** pré-definidas ficam armazenadas e você não precisa lembrar de como criá-las.(Não confundir com resultados)

- Como o próprio nome ajuda identificar elas permitem criar uma visão mais lógica para um humano entender a modelagem.
- Facilita a troca do modelo físico sem o perigo de quebrar *queries* existentes.
- Eventualmente o banco de dados **pode** fazer algumas otimizações por já ter conhecimento das *queries* utilizadas nas *views*.

- Você pode colocar permissões na *view*, ou seja, você pode proibir acesso à tabelas em seu estado bruto, mas em uma certa condição o usuário pode ter acesso à informação da forma como você definiu na *view*. Você controla melhor o que e como o usuário pode acessar a informação. Funciona como um *firewall*.

- Regras de negócio podem ser adotadas nas *views*. A maneira como você vai acessar os dados está pré-definida. Isto é útil para formatar dados, ajudar ferramentas externas e facilitar o acesso via APIs.

- Podem facilitar o acesso em base legada, tornando migrações e transições indolores.
- Se a *view* for materializada pode ter ganho de performance para o acesso aos dados já “consolidados”.

- Esconde uma complexidade da *query* podendo enganar o desenvolvedor quanto à performance necessária para acessar determinada informação.
- E pode ser pior quando *views* usam outras *views*. Em alguns casos você pode estar fazendo consultas desnecessárias sem saber de forma muito intensiva.

- Cria uma camada extra. Mais objetos para administrar. Algumas pessoas consideram isto um aumento de complexidade. Uma outra forma de ver isto é que uma *view* pode ser mal usada.

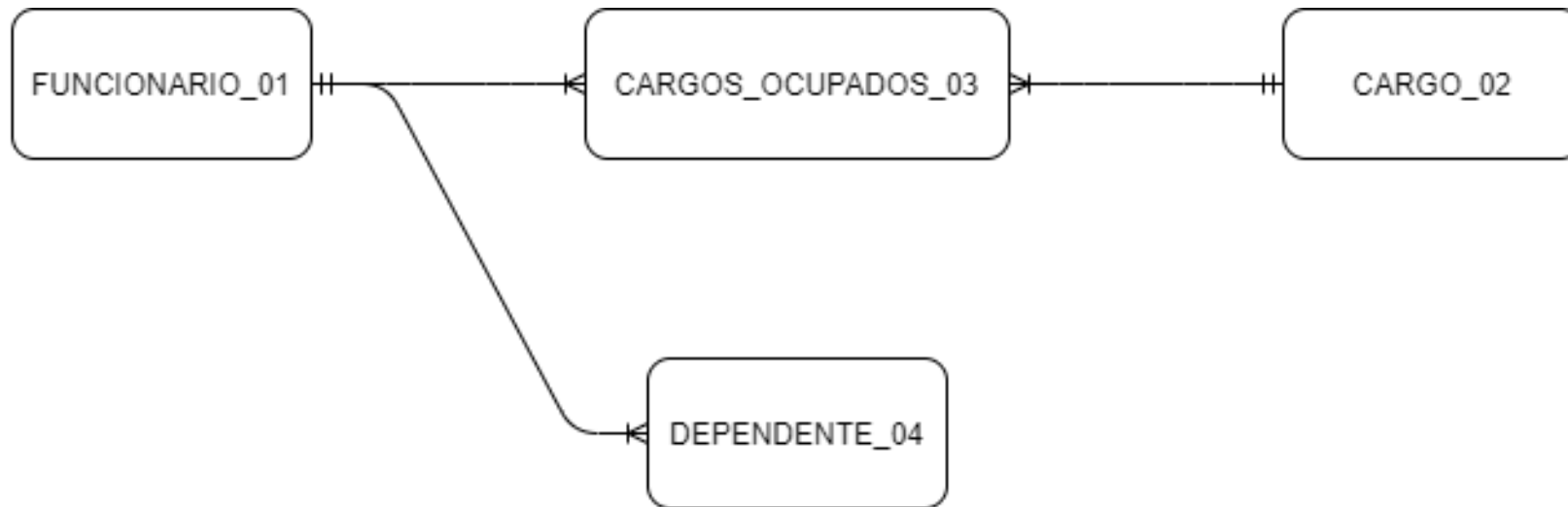
- Pode limitar exageradamente o que o usuário pode acessar impedindo certas tarefas.
- Se a *view* for materializada fará com que alterações nas tabelas reais envolvidas sejam mais lentas, afinal são mais tabelas para atualizar. Este tipo de *view* funciona de forma semelhante a um gatilho.

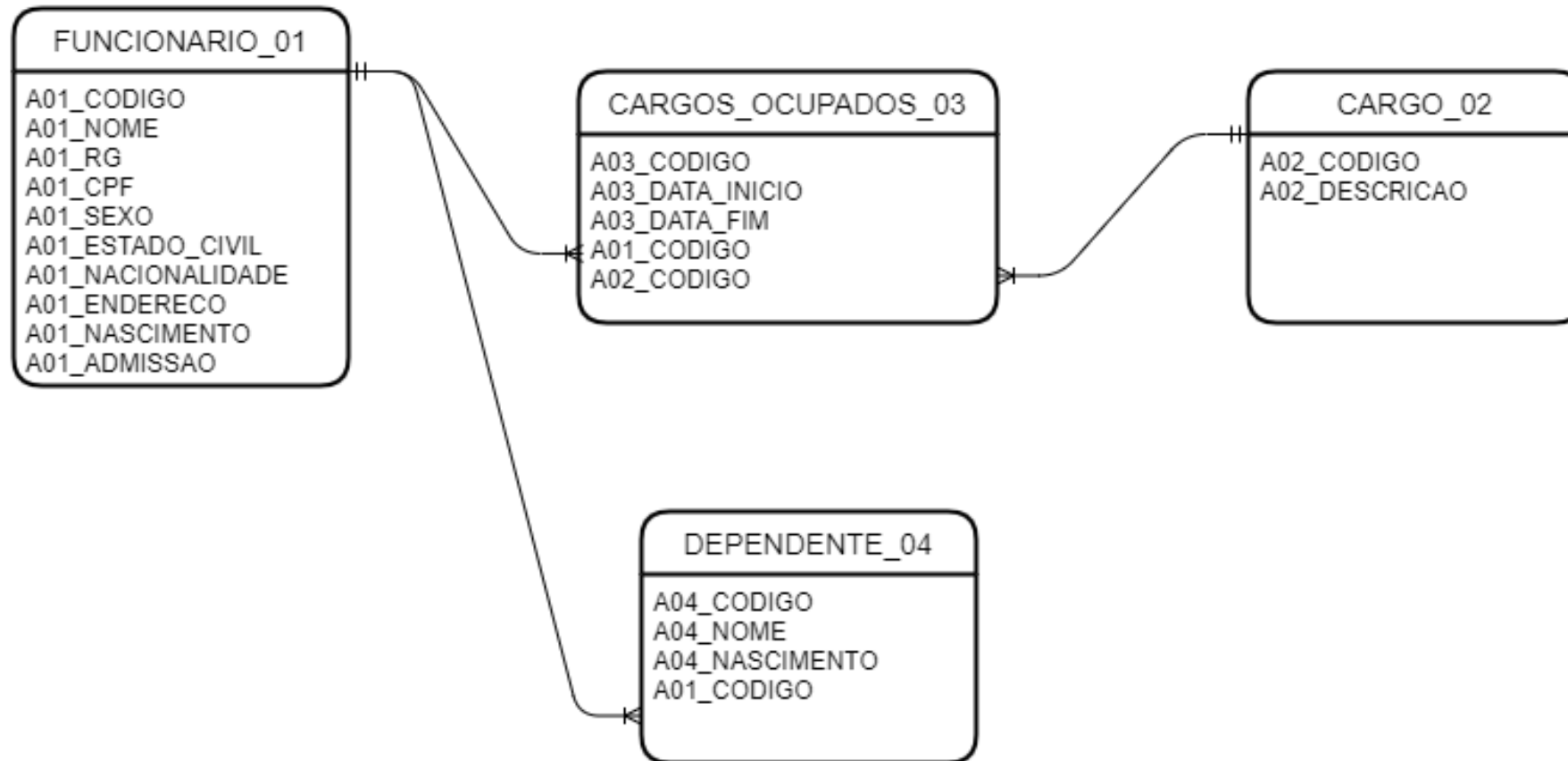
- A *view* é uma consulta simples armazenada no banco de dados que cria uma ilusão de ser uma tabela, e pode ser usada em diversas operações para:
 - simplificar as *queries* e facilitar o acesso à determinadas informações
 - conformar melhor com o modelo lógico
 - permitir controlar melhor o acesso aos dados para determinados usuários. Pode-se criar uma *view* com determinadas colunas e dar permissão de acesso a um usuário ou grupo para esta *view*, não para a tabela física, aí ele só terá acesso a estes dados.

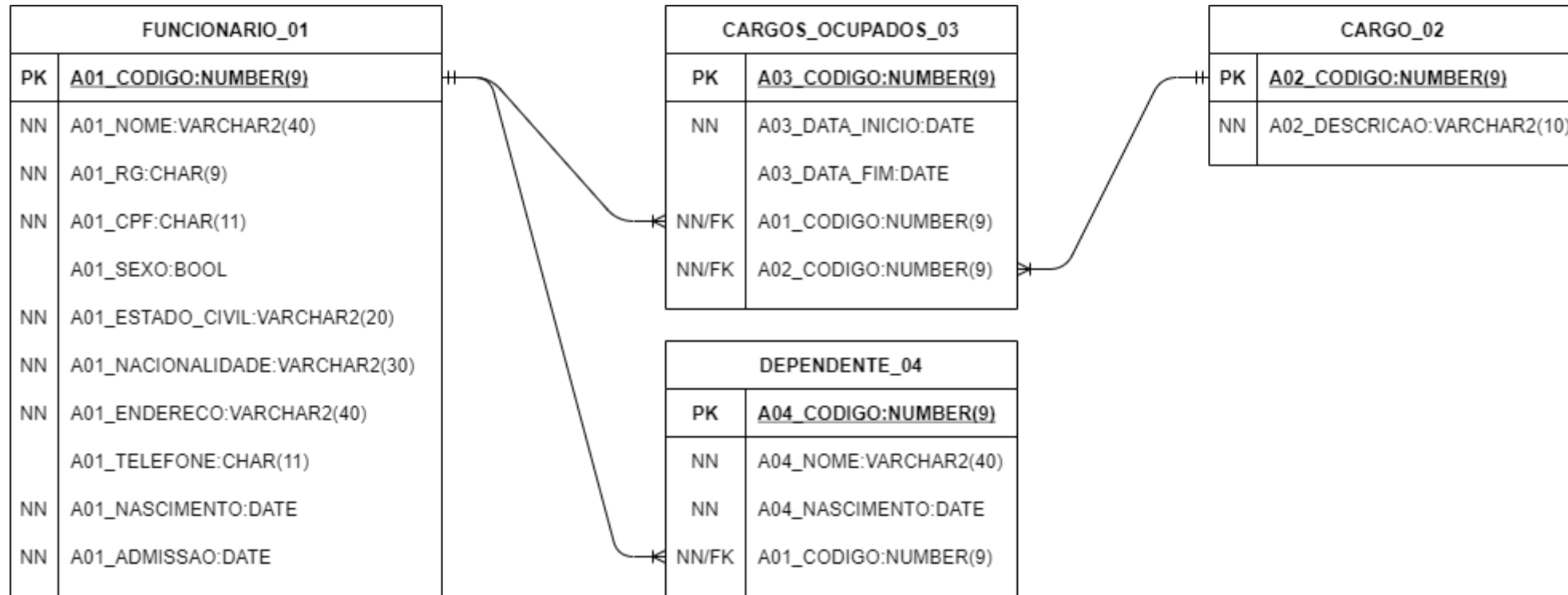
- Todo conjunto de dados da *view* é gerado no momento que é solicitado (se nenhuma otimização for feito pelo banco de dados). E este é o único custo, que nem pode ser considerado exatamente de extra porque se está usando ela, provavelmente precisaria fazer isto de qualquer jeito.

Normalize a tabela abaixo com o objetivo de armazenar os dados cadastrais dos funcionários de uma empresa. Leve em consideração que o funcionário pode ter vários cargos e dependentes.

Dados Cadastrais do Funcionário		
Matrícula:	Nome:	
Data de Nasc.:	Nacionalidade:	Sexo:
Est. Civil:	RG:	CIC:
Endereço:	Telefone:	Data de Admissão:
Cargos Ocupados		
Cargo:	Dt. Início:	Dt. Fim:
Cargo:	Dt. Início:	Dt. Fim:
Dependentes		
Nome:	Dt. Nascimento:	
Nome:	Dt. Nascimento:	







Nome da tabela:	FUNCIONARIO_01										
Nome do campo:	A01_CODIGO	A01_NOME	A01_RG	A01_CPF	A01_SEXO	A01_ESTADO_CIVIL	A01_NACIONALIDADE	A01_ENDERECO	A01_TELEFONE	A01_NASCIMENTO	A01_ADMISSAO
Tipo de chave:	PK	-	-	-	-	-	-	-	-	-	-
Único/nulo:	UK/NN	NN	NN	UK/NN	-	NN	NN	NN	-	NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(40)	CHAR(9)	CHAR(11)	BOOL	VARCHAR2(20)	VARCHAR2(30)	VARCHAR2(40)	CHAR(11)	DATE	DATE
Dados simples:	10	João	111111111	11111111111	True	Casado	Brasileiro	Rua 1	11912345678	03/04/1976	02/06/2000
	20	Marieta	222222222	22222222222	False	Solteira	Portuguesa	Rua 2	-	07/09/1988	09/01/2003
	30	Renan Souza	123456789	34567891212	-	Divorciado	Brasileiro	Rua José Marcelo, 109	1132940073	16/11/1994	15/09/2014
	40	Jose dos Santos	234567890	45678912313	True	Solteiro	Brasileiro	Rua Julio Grotta, 23	-	27/02/1990	19/03/2017

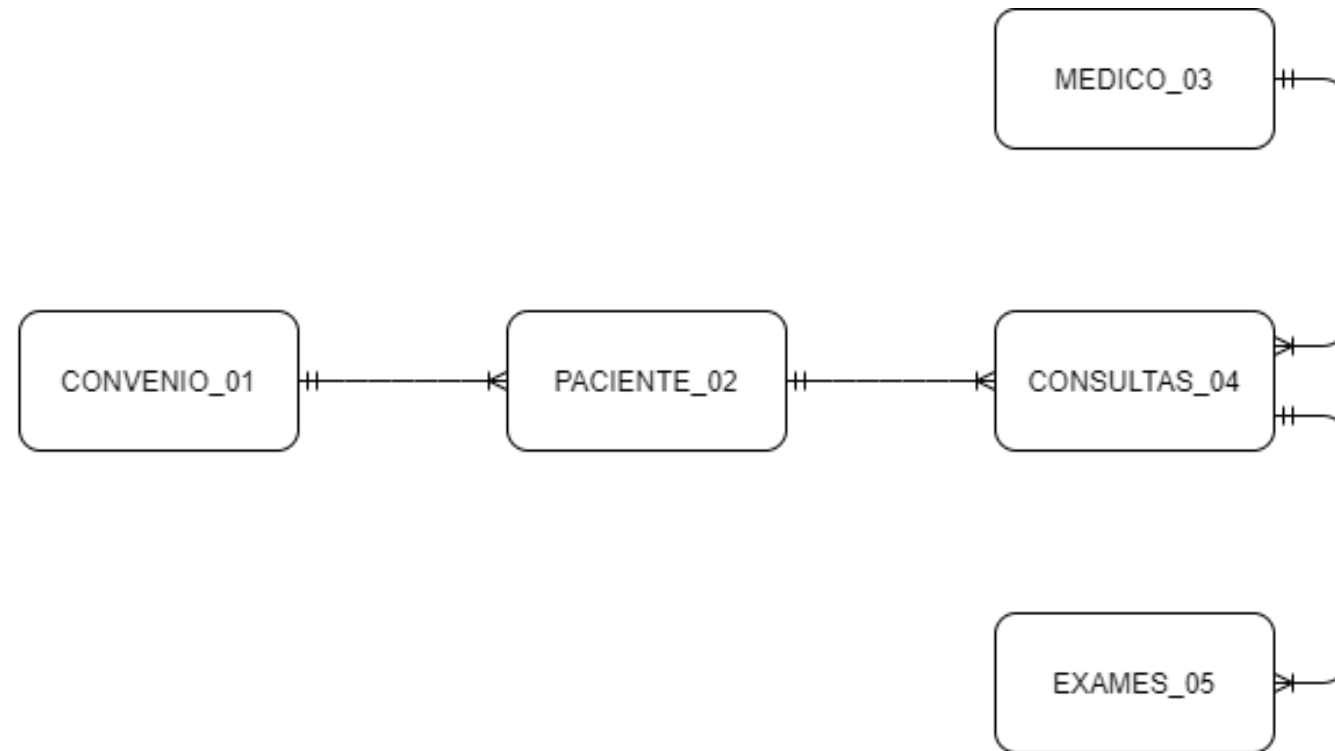
Nome da tabela:	CARGO_02	
Nome do campo:	A02_CODIGO	A02_DESCRICAO
Tipo de chave:	PK	-
Único/nulo:	UK/NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(10)
Dados simples:	1	Presidente
	2	Diretor
	3	Gerente
	4	Supervisor

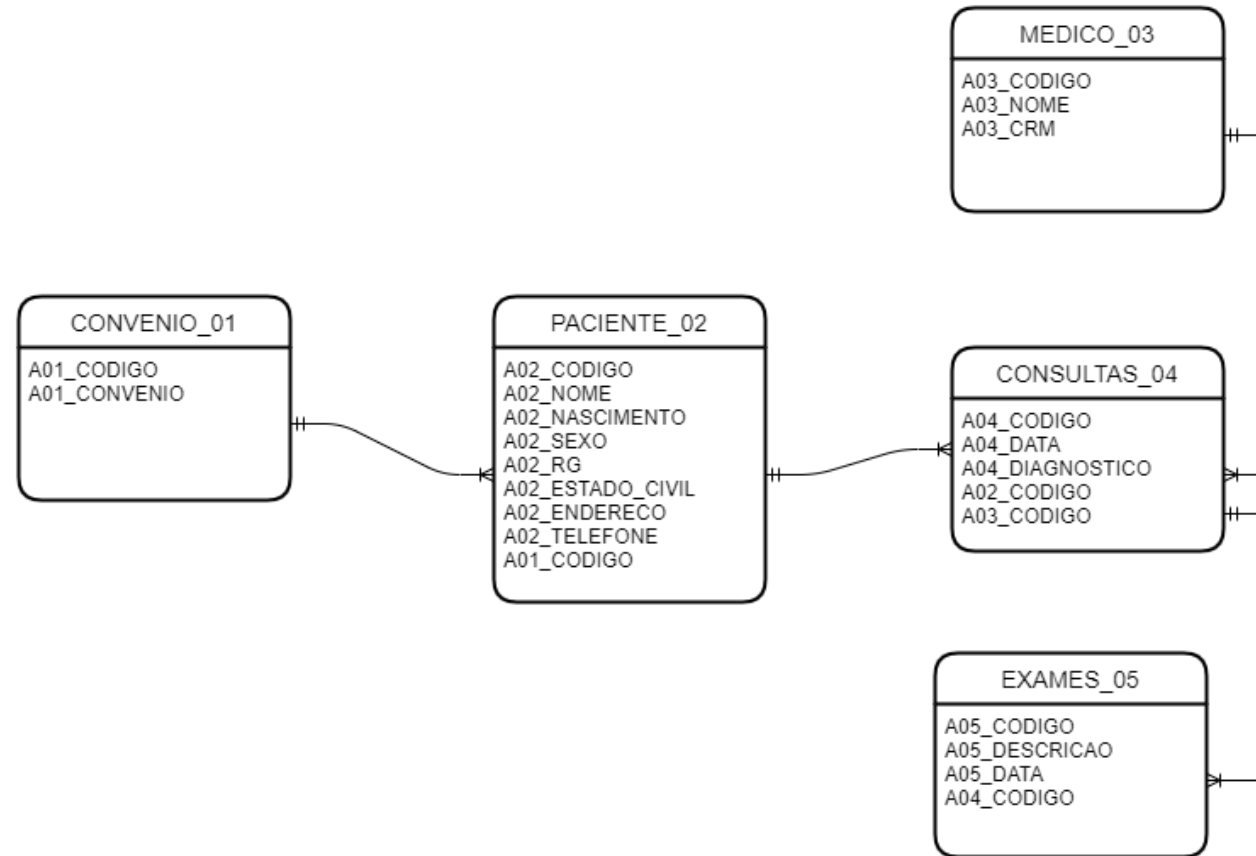
Nome da tabela:	CARGOS_OCUPADOS_03				
Nome do campo:	A03_CODIGO	A03_DATA_INICIO	A03_DATA_FIM	A01_CODIGO	A02_CODIGO
Tipo de chave:	PK	-	-	FK	FK
Único/nulo:	UK/NN	NN	-	NN	NN
Tipo de dado:	NUMBER(9)	DATE	DATE	NUMBER(9)	NUMBER(9)
Dados simples:	1	02/06/2000	-	1	4
	2	09/01/2003	19/08/2019	2	1
	3	15/09/2014	-	3	3
	4	19/03/2017	-	4	2

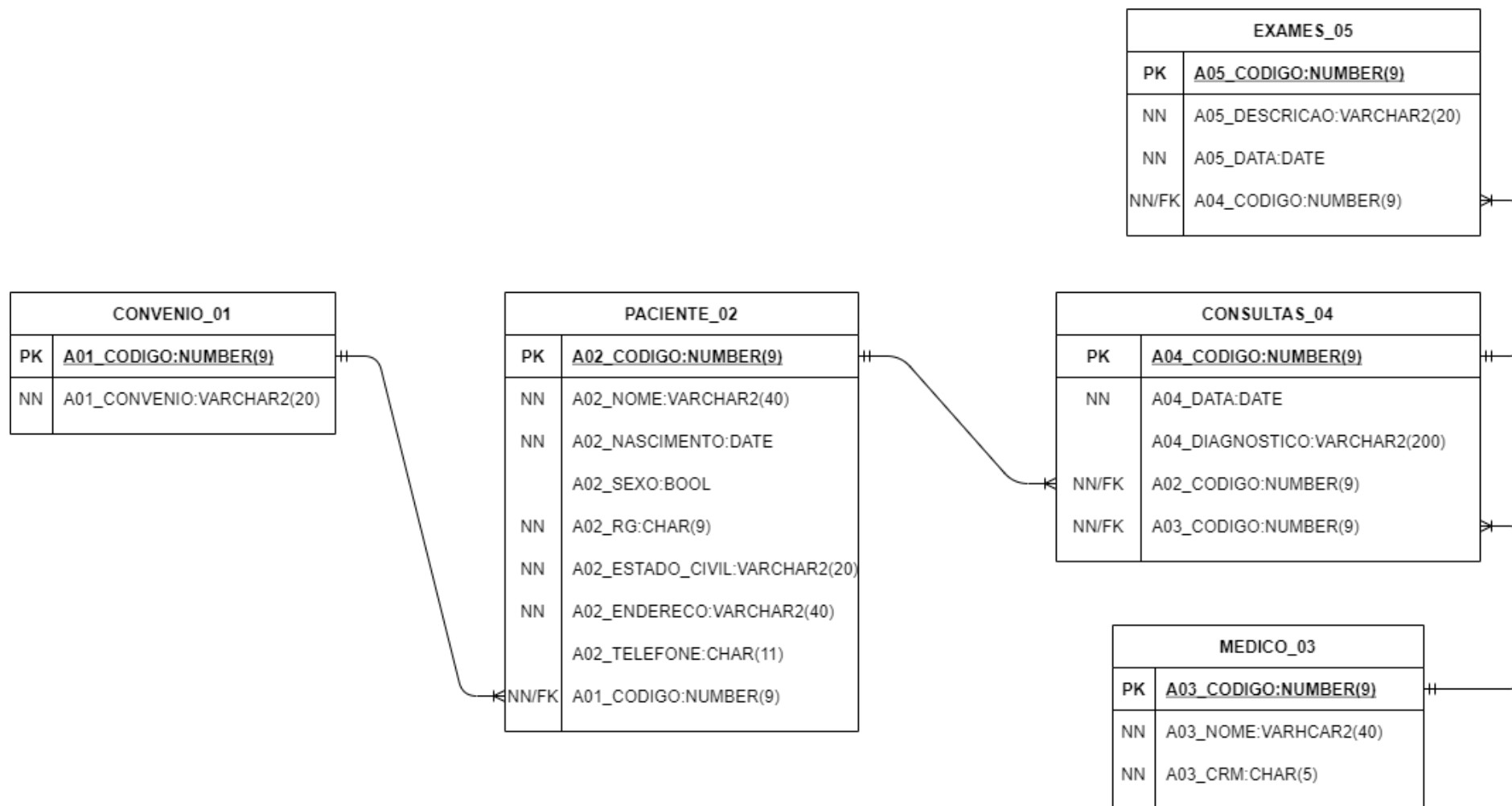
Nome da tabela:	DEPENDENTE_04			
Nome do campo:	A04_CODIGO	A04_NOME	A04_NASCIMENTO	A01_CODIGO
Tipo de chave:	PK	-	-	FK
Único/nulo:	UK/NN	NN	NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(40)	DATE	NUMBER(9)
Dados simples:	1	Priscila	12/10/1992	1
	2	Roberto	22/03/1998	1
	3	Juliana Souza	30/11/2001	3
	4	Enzo dos Santos	16/09/2006	4

Normalize a tabela abaixo com o objetivo de armazenar os dados das fichas médicas de todos os pacientes de uma clínica. Leve em consideração que o CRM do médico serve para identificá-lo unicamente.

Ficha Médica				
Número paciente:		Nome:		
Data de Nasc.:		Sexo:	Convênio:	
Est. Civil:		RG:	Telefone:	
Endereço:				
Consultas				
Número Consulta	Data	Médico (CRM)	Diagnóstico	
Exames				
Número Consulta	Exame	Data		







Nome da tabela:	CONVENIO_01	
Nome do campo:	A01_CODIGO	A01_CONVENIO
Tipo de chave:	PK	-
Único/nulo:	UK/NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(20)
Dados simples:	1	Sul América
	2	UNIMED
	3	GEAP
	4	Prevent Senior

Nome da tabela:	PACIENTE_02								
Nome do campo:	A02_CODIGO	A02_NOME	A02_RG	A02_SEXO	A02_ESTADO_CIVIL	A02_ENDERECO	A02_TELEFONE	A02_NASCIMENTO	A01_CODIGO
Tipo de chave:	PK	-	-	-	-	-	-	-	FK
Único/nulo:	UK/NN	NN	NN	-	NN	NN	-	NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(40)	CHAR(9)	BOOL	VARHCAR2(20)	VARCHAR2(40)	CHAR(11)	DATE	NUMBER(9)
Dados simples:	1	João	111111111	True	Casado	Rua 1	11912345678	03/04/1976	1
	2	Marieta	222222222	False	Solteira	Rua 2	-	07/09/1988	2
	3	Renan Souza	123456789	-	Divorciado	Rua José Marcelo, 109	1132940073	16/11/1994	3
	4	Jose dos Santos	234567890	True	Solteiro	Rua Julio Grota, 23	-	27/02/1990	4

Nome da tabela:	MEDICO_03		
Nome do campo:	A03_CODIGO	A03_NOME	A03_CRM
Tipo de chave:	PK	-	-
Único/nulo:	UK/NN	NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(10)	CHAR(5)
Dados simples:	1	José Moraes	11111
	2	Danilo Aguiar	12345
	3	Maria da Silva	23456
	4	João Bonifácio	34567

Nome da tabela:	CONSULTAS_04				
Nome do campo:	A04_CODIGO	A04_DATA	A04_DIAGNOSTICO	A02_CODIGO	A03_CODIGO
Tipo de chave:	PK	-	-	FK	FK
Único/nulo:	UK/NN	NN	-	NN	NN
Tipo de dado:	NUMBER(9)	DATE	VARCHAR2(200)	NUMBER(9)	NUMBER(9)
Dados simples:	1	02/06/2000	Outras doenças pulmonares obstrutivas crônicas	1	4
	2	09/01/2003	Outras formas especificadas de doença pulmonar obstrutiva crônica	2	1
	3	15/09/2014	DPOC	3	3
	4	19/03/2017	congestao	4	2

Nome da tabela:	EXAMES_05			
Nome do campo:	A05_CODIGO	A05_EXAME	A05_DATA	A04_CODIGO
Tipo de chave:	PK	-	-	FK
Único/nulo:	UK/NN	NN	NN	NN
Tipo de dado:	NUMBER(9)	VARCHAR2(20)	DATE	NUMBER(9)
Dados simples:	1	Tomografia	12/10/2010	1
	2	Radiografia	12/10/2020	1
	3	Hemograma	30/11/2001	3
	4	RM	16/09/2006	4

OBRIGADO