

ARTEFATO A - REGRA DE NEGÓCIO, REQUISITO DO SISTEMA E ATRIBUTOS NO MODELO

Todas as características solicitadas foram atendidas. Algumas alterações foram realizadas no modelo. O código com as mudanças e as características também estarão disponíveis no arquivo de texto “pgsql_create_table.txt” com comentários; O arquivo de texto “pgsql_popula_banco.txt” corresponde ao código utilizado para popular o banco. As inserções de tais características no modelo estarão indicadas separadamente, quando solicitadas, neste documento.

1 CATEGORIA – OBJETOS COMPLEXOS

Regra de Negócio:

Optamos por utilizar a característica “atributo multivalorado” para atender a este artefato. Para isto criamos o atributo complexo do tipo “jsonb” chamado “Telephone” na tabela Employee. Este campo terá como responsabilidade guardar os meios de contato remoto do funcionário como Celular (“mobile”), Fixo (“landline”) e até mesmo Comercial (“comercial”), por exemplo, para que seja possível ter registrado formas de contato com o funcionário mesmo quando este se tornar um ex-colaborador.

Atributo no Modelo:

Employee	
PK	idt_employee SERIAL
FK	idt_department INTEGER
FK	idt_dependent INTEGER
	name_employee VARCHAR
	date_admission DATE
	salary DECIMAL
	telephone JSONB

2 CATEGORIA – TIPO REFERÊNCIA

Requisito do Sistema:

Como “tipo referência” entendemos que a criação de um tipo específico no modelo atenderia a este artefato. Para isto criamos o tipo “Address_Type” e o utilizamos na tabela “Employee” ao criar o atributo “address”. O objetivo é fazer com que um atributo seja formado pelo conjunto de outros atributos (remetendo ao conceito de “atributo composto”) sendo obrigatória a necessidade, no momento da inserção, de utilizar (mesmo que vazio) todos os campos pertencentes ao novo tipo. Este fato que não ocorre na categoria anterior, quando criamos um atributo multivalorado, já que este pode possuir vários valores diferentes.

Atributo no Modelo: (mesmo não sendo necessário, optamos demonstrar por “parecer” um atributo composto)

Employee	
PK	idt_employee SERIAL
FK	idt_department INTEGER
FK	idt_dependent INTEGER
	name_employee VARCHAR
	date_admission DATE
	salary DECIMAL
	telephone JSONB
	address Address_Type

3 CATEGORIA – HERANÇA

Regra de Negócio:

Nessa categoria levamos em consideração o feedback que recebemos referente a “Parte 1” do trabalho e, pensando sempre em aprender e melhorar, resolvemos “ajustar” o modelo para adequar não somente a apenas uma herança mas sim todas as (quatro) heranças que tínhamos planejado desde o início. Sendo assim, implementamos o conceito de herança na tabela “Employee”. Mantendo a mesma ideia, porém alterando a regra de negócio e a reescrevendo aqui.

As tabelas “Director”, “Manager”, “Specialist” e “Analyst” agora herdam os atributos da tabela “Employee” além de conter mais características e novas definições:

A tabela “Director” diz respeito à um funcionário com cargo de diretor. E o que o diferencia são os atributos “rate_plr”, no qual contém a porcentagem de participação de lucro, e o atributo “luxury_car”, que virá preenchido com o nome do carro de luxo seguido da placa. Tais atributos são referentes a benefícios que somente um diretor possui na empresa.

A tabela “Manager” diz respeito à um funcionário com cargo de gerente. E o que o diferencia são os atributos “bonus”, no qual contém o valor em reais do bônus, e o atributo “sport_car”, que virá preenchido com o nome do carro “econômico” seguido da placa. Tais atributos são referentes a benefícios que somente um gerente possui na empresa.

Ambos os carros (de luxo para diretor, e sport para gerente) são benefícios nos quais a empresa disponibiliza para funcionários desses cargos (não seriam os carros próprios dos funcionários, e sim um carro disponibilizado pela empresa para o funcionário).

A tabela “Specialist” diz respeito à um funcionário com cargo de especialista. E o que o diferencia são os atributos “company”, no qual contém o nome da empresa que esse especialista é funcionário¹, e o atributo “cnpj”, que corresponde ao CNPJ de tal empresa. Decidimos definir que para esse cargo (especialista) poderão existir funcionários de empresas externas, logo, fazendo com que esses atributos sejam característicos apenas a este tipo de funcionário.

E a tabela “Analyst” diz respeito à um funcionário com cargo de analista. E o que o diferencia são os atributos “university”, no qual contém o nome da faculdade que o analista está estudando (ou, a última faculdade em que estudou), e o atributo “class”, que virá preenchido com o nome do curso que está matriculado (ou, o último curso que cursou), em tal faculdade. Tais atributos são referentes a características que somente um analista possui na empresa ao passo que decidimos definir que analistas deverão obrigatoriamente possuir um diploma (ou, estar em processo de obter um).

Essas últimas duas tabelas possuem uma chave estrangeira “idt_level_knowledge”. Tal chave optamos por retirar da tabela “Employee” e passar a utilizá-la nas tabelas “Specialist” e “Analyst” visto que a tabela “Level_Knowledge” possui somente os níveis de conhecimento de um analista ou um especialista (levando em consideração que um “Estagiário” é um analista).

Observação: A tabela “Dependent” era uma entidade fraca que dependia de “Employee”. Essa entidade, após a inclusão da herança, teve de se tornar forte (adicionamos uma chave estrangeira da tabela “Dependent” na tabela “Employee”) pois é algo que o Postgres não permite. A tabela “Team” cumpre com o requisito, de ser uma entidade fraca (dependendo de “Project”), solicitado na “Parte 1” deste trabalho.

Atributo no Modelo:

