

# Relatório do Trabalho Final

## Disciplina de Banco de Dados I

Arthur Scarpatto Rodrigues, Leonardo Luiz Gambalonga

### Contents

---

<b>1</b>	<b>Sistema de Gerenciamento de RH</b>	<b>1</b>
1.1	Descrição do Objetivo Geral do Sistema . . . . .	1
1.2	Descrição Detalhada do Sistema . . . . .	1
1.2.1	Tabela <i>employees</i> . . . . .	1
1.2.2	Tabela <i>departments</i> . . . . .	2
1.2.3	Tabela <i>roles</i> . . . . .	2
1.2.4	Tabela <i>employee_roles</i> . . . . .	2
1.2.5	Tabela <i>schedules</i> . . . . .	3
1.2.6	Tabela <i>projects</i> . . . . .	3
1.2.7	Tabela <i>project_assignments</i> . . . . .	3

## 1. Sistema de Gerenciamento de RH

---

Nosso trabalho segue o esquema de um software de gerenciamento de RH simples, que oferece recursos à empresa para considerar seus custos, tempo, e alocar recursos efetivamente.

### 1.1. Descrição do Objetivo Geral do Sistema

O objetivo do sistema é providenciar um sistema de gerenciamento de RH de uma empresa fictícia, onde constam dados sobre os funcionários, sobre a hierarquia de comando, compensação, calendários, projetos e alocações de fundos, horários de projetos e de funcionários, alocação de tempo e funcionários para projetos específicos, assim como algumas outras funcionalidades, seguindo o mesmo padrão.

### 1.2. Descrição Detalhada do Sistema

O sistema contém sete tabelas, que giram em torno da tabela principal, *employees*, responsável por armazenar dados de funcionários específicos. As outras tem como objetivo tratar da alocação de tempo e recursos para projetos, quais funções os funcionários desempenham, e seus horários individuais. Seguem abaixo as especificações técnicas de cada tabela.

#### 1.2.1 Tabela *employees*

A tabela *employees* tem como colunas os seguintes valores:

A tabela tem uma chave estrangeira para a tabela *departments*, a ser discutida a diante. Essa relação é uma **many-to-one**, já que vários funcionários podem estar associados a um único departamento cada.

A maioria das colunas é auto-explicatória, de acordo com seus nomes, em relação à função que desempenham. Vale mencionar que *gender* e *contact-number* não são necessários administrativamente (no caso de *contact-number*, porque já temos o *email*), e os funcionários podem preferir não mencioná-las, então são consideradas anuláveis.

Nome da Coluna	Tipo	Modificadores	Referencia
employee_id	serial	PRIMARY KEY	departments (FK)
department_id	integer	NOT NULL	
name	varchar	NOT NULL	
salary	integer	NOT NULL	
date_of_birth	date	NOT NULL	
gender	varchar		
contact_number	varchar		
email	varchar	NOT NULL	

Table 1: Informações técnicas sobre a tabela *employees*.

### 1.2.2 Tabela *departments*

Essa é uma tabela simples que contém informações básicas sobre o departamento, para fins de organização. Segue abaixo sua especificação:

Nome da Coluna	Tipo	Modificadores	Referencia
department_id	serial	PRIMARY KEY	
department_name	varchar	NOT NULL	

Table 2: Informações técnicas sobre a tabela *departments*.

Essa tabela não contém referências a nenhuma outra. Também não contém nenhum **CASCADE** associado, já que podemos fechar um departamento sem necessariamente demitir os funcionários.

### 1.2.3 Tabela *roles*

Outra simples tabela que contém informações sobre as funções que podem estar associadas a um certo funcionário, de acordo com a especificação abaixo:

Nome da Coluna	Tipo	Modificadores	Referencia
role_id	serial	PRIMARY KEY	
role_name	varchar	NOT NULL	

Table 3: Informações técnicas sobre a tabela *roles*.

Também não referencia nenhuma outra tabela. No entanto, possui efeito de **CASCADE** ao ser deletada uma entrada, efetivamente deletando a(s) entrada(s) correspondente(s) na tabela *employee\_roles*, que representa uma associação **many-to-many** entre *employees* e *roles*.

Nesse sentido, *employee\_roles* representa uma função de **many-to-one** para com *roles*, já que cada entrada na primeira tabela contém uma referência para alguma entrada da segunda.

### 1.2.4 Tabela *employee\_roles*

Como dito anteriormente, essa tabela representa uma relação **many-to-many** entre *employees* e *roles*. Dessa maneira, vários funcionários podem desempenhar vários papéis na empresa, e vários papéis podem ser desempenhados por vários funcionários.

Não possui efeito de **CASCADE** próprio, mas é deletada caso uma entrada da tabela *roles* ou uma da tabela *employees* seja deletada, pois não faz sentido existir uma associação de um funcionário com uma função se algum dos dois não existe.

Nome da Coluna	Tipo	Modificadores	Referencia
employee_role_id	serial	PRIMARY KEY	employees (FK) roles (FK)
employee_id	integer	NOT NULL	
role_id	integer	NOT NULL	
start_date	date	NOT NULL	
end_date	date	NOT NULL	

Table 4: Informações técnicas sobre a tabela *employee\_roles*.

### 1.2.5 Tabela *schedules*

Representa os horários dos funcionários e os dias em que começaram a trabalhar/dias em que vão acabar, ou sair da empresa. Essa última informação, claro, pode ser anulável, pois na maioria dos casos, não se sabe ao certo previamente quanto tempo um funcionário vai ficar na empresa.

Nome da Coluna	Tipo	Modificadores	Referencia
schedule_id	serial	PRIMARY KEY	employees (FK)
employee_id	integer	NOT NULL	
start_date	date	NOT NULL	
end_date	date		
start_time	timestamp	NOT NULL	
end_time	timestamp		

Table 5: Informações técnicas sobre a tabela *schedules*.

Aqui, o *start\_time* e *end\_time* se referem aos horários de um funcionário, em um determinado dia.

A tabela *schedules* é deletada por efeito de **CASCADE** caso alguma entrada de *employees* seja deletada. Nessa tabela, temos uma relação de **many-to-one** para com *employees*.

### 1.2.6 Tabela *projects*

Essa tabela representa projetos desenvolvidos pela empresa, assim como suas respectivas verbas, e metadados como a data de início e data de fim prevista (opcional).

Nome da Coluna	Tipo	Modificadores	Referencia
project_id	serial	PRIMARY KEY	
project_name	varchar	NOT NULL	
budget	integer	NOT NULL	
start_date	date	NOT NULL	
end_date	date		

Table 6: Informações técnicas sobre a tabela *projects*.

A tabela *projects* possui efeito de **CASCADE** em relação à tabela *project\_assignments*, que representa uma relação **many-to-many** de *employees* para *projects*. Dessa forma, *project\_assignments* também relaciona um papel de **many-to-one** para com *projects*.

### 1.2.7 Tabela *project\_assignments*

Como dito anteriormente, essa tabela representa um **many-to-many** de *employees* e *projects*, com a função lógica de que um funcionário pode estar em vários projetos, e um projeto pode ser executado por vários funcionários.

Aqui, *assigned\_at* significa a data em que um funcionário foi alocado para um dado projeto.

Nome da Coluna	Tipo	Modificadores	Referencia
assignment_id	serial	PRIMARY KEY	
project_id	integer	NOT NULL	projects (FK)
employee_id	integer	NOT NULL	employees (FK)
assigned_at	date	DEFAULT CURRENT_DATE	

Table 7: Informações técnicas sobre a tabela *project\_assignments*.

Essa tabela sofre efeitos de **CASCADE** tanto de *employees* quanto de *projects*: se alguma entrada em uma dessas tabelas for deletada, e tiver entradas relacionadas em *project\_assignments*, então essas entradas também serão deletadas.