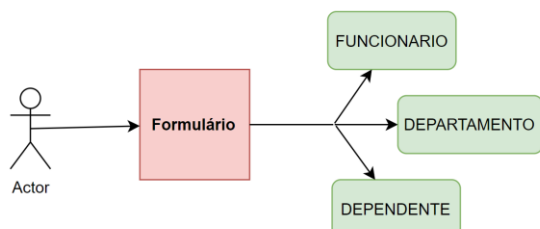




JSON Em banco de dados

As vezes **um formulário** precisa manipular informações de **mais de uma tabela**. Por exemplo, uma tela de **formulário de cadastro** de **UM FUNCIONARIO** pode manipular outras **N tabelas** como **DEPENDENTE**, **PROJETO** e etc..... Dessa maneira a aplicação pode ficar complexa:



Uma ideia de resolução é **adaptar esse formulário inteiro como um JSON no banco de dados** com o **identificador sendo o CPF do funcionário**. Dessa maneira seria possível converter o Json em objetos e vice versa:

CPF	JSON
12345678966	{"Cpf": "12345678966", "Sexo": "M", "Salario": 30000.0, "Endereco": "Rua das Flores 751 São Paulo SP", "Nome_Meio": "B", "Dependentes": [{"Sexo": "F", "Parentesco": "..."}]}
33344555587	{"Cpf": "33344555587", "Sexo": "M", "Salario": 40000.0, "Endereco": "Rua da Lapa 34 São Paulo SP", "Nome_Meio": "T", "Dependentes": [{"Sexo": "F", "Parentesco": "Fil..."}]}
44111555587	{"Cpf": "44111555587", "Sexo": "M", "Salario": 31000.0, "Endereco": "Rua Rodrigues Alves 771 Rio de Janeiro RJ", "Nome_Meio": "V", "Dependentes": [{"Sexo": "F", "Parentesco": "..."}]}
44222555587	{"Cpf": "44222555587", "Sexo": "F", "Salario": 36000.0, "Endereco": "Rua 9 de Julho 800 São Paulo SP", "Nome_Meio": "G", "Dependentes": [{"Sexo": "M", "Parentesco": "..."}]}
44333555587	{"Cpf": "44333555587", "Sexo": "M", "Salario": 34000.0, "Endereco": "Rua Abelardo Barbosa 412 São Paulo SP", "Nome_Meio": "K", "Dependentes": [{"Sexo": "F", "Parentesco": "..."}]}

Mas como posso **fazer consultas filtradas no banco de dados se todas as informações estão em um JSON** numa única coluna?

Alguns **bancos de dados conseguem lidar com esse tipo de formato** (o campo precisa ser do tipo *json*) usando **SELECTS** com **sintaxes** para manipular **JSON**, o **MySQL** por exemplo tem **algumas maneiras de extrair propriedades de JSON**, sendo o **"\$"** a **raiz do JSON**:

```
SELECT JSON_PRETTY(`JSON`) AS JSON_DOCUMENTO FROM tb_object_funcionario;

SELECT JSON` -> `$.Sexo` AS SEXO FROM tb_object_funcionario;

SELECT REPLACE(JSON` -> `$.Sexo`, "M", "F") AS SEXO FROM tb_object_funcionario;

SELECT JSON_EXTRACT(`JSON`, `$.Cpf`) AS CPF FROM tb_object_funcionario;
```

A **chave** para as **consultas** com filtro é justamente **extrair do JSON as propriedades** necessárias tanto para **RESULTADO**, como para o **SELECT**:

```
SELECT * FROM tb_object_funcionario WHERE JSON_EXTRACT(`JSON`, `$.Sexo`) = "M" ;
```

```
SELECT
JSON` -> `$.Sexo` AS SEXO,
JSON` -> `$.Primeiro_Nome` AS NOME,
JSON` -> `$.Salario` AS SALARIO
FROM tb_object_funcionario
WHERE JSON_EXTRACT(`JSON`, `$.Sexo`) = "M" ;
```

Obviamente **funções de soma, agrupamento e ordenação funcionam da mesma forma**:

```
SELECT SUM(JSON_EXTRACT(`JSON`, "$.Salario")) from tb_object_funcionario;

SELECT JSON_EXTRACT(`JSON`, "$.Numero_Departamento") AS Numero_Departamento,
SUM(JSON_EXTRACT(`JSON`, "$.Salario")) AS Salario_Total
FROM tb_object_funcionario
GROUP BY `JSON` -> "$.Numero_Departamento";
```



Array de JSON em banco de dados

Um jeito de **navegar nos itens de um array de JSON** é usar a **posição do elemento** **"\$.Dependentes[0]":**

```
SELECT JSON_EXTRACT(`JSON`, "$.Dependentes[0]") AS DEPENDENTES
FROM tb_object_funcionario WHERE `JSON` -> "$.Cpf" = "12345678966";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

DEPENDENTES
{"Sexo": "F", "Parentesco": "Filha", "Data_Nascimento": "1988-01-30T00:12:00", "Nome_Dependente": "Alicia"}

A navegação é bem parecida com orientação a objetos na real, então dá pra **entrar em "atributos" de um elemento do array** (e também ir se aprofundando na estrutura desse JSON em cada quebra de lista e etc):

```
SELECT JSON_EXTRACT(`JSON`, "$.Dependentes[0].Nome_Dependente") AS DEPENDENTES
FROM tb_object_funcionario WHERE `JSON` -> "$.Cpf" = "12345678966";
```

Result Grid | Filter Rows: | Export: |

DEPENDENTES
"Alicia"



Transformar array de JSON em registros de tabela única

Podemos transformar um Array de JSON em tabelas unicas utilizando a estrutura **JSON_TABLE**, ela basicamente é uma **sintaxe de mapeamento de um Json para uma tabela**.

Onde temos que dizer **de qual Array essa tabela será feita**, **quais elementos do array vamos usar**, **como serão as constraints das colunas** e de qual caminho vem o valor delas (qual elemento do array vai ser o valor da coluna) e obviamente **referenciar essa nova tabela**:

```
SELECT JSON_EXTRACT(tb_object_funcionario.`JSON`, "$.Cpf") AS Cpf_Funcionario,
TB_OBJECT_DEPENDENTE.Nome_Dependente,
TB_OBJECT_DEPENDENTE.Sexo,
TB_OBJECT_DEPENDENTE.Parentesco
FROM tb_object_funcionario
CROSS JOIN
JSON_TABLE (JSON_EXTRACT(`JSON`, "$.Dependentes"), "$[*]")
COLUMNS (Nome_Dependente VARCHAR(20) PATH "$.Nome_Dependente",
Sexo VARCHAR(20) PATH "$.Sexo", DT_NASC DATETIME PATH "$.Data_Nascimento",
Parentesco VARCHAR(20) PATH "$.Parentesco")) AS TB_OBJECT_DEPENDENTE;
```

Cpf_Funcionario	Nome_Dependente	Sexo	Parentesco
"12345678966"	Alicia	F	Filha
"12345678966"	Elizabeth	F	Esposa
"12345678966"	Michael	M	Filho
"33344555587"	Alicia	F	Filha
"33344555587"	Janaina	F	Esposa
"33344555587"	Tiago	M	Filho
"44111555587"	Joaquina	F	Esposa
"44222555587"	Claudio	M	Filho
"44222555587"	Juvenal	M	Marido
"44333555587"	Katia	F	Esposa