

Faça como eu fiz

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Para prosseguirmos precisamos ter o MySQL e o MySQL Workbench instalados. Veja em cursos anteriores a esta carreira como instalar estes produtos, caso seja necessário.

2) Se conecte no MySQL através do MySQL Workbench.

3) Crie um anova base de dados chamada **empresanosql**.

4) Baixe o arquivo **ExportEmpresa.sql**.

5) Carregue este arquivo no MySQL Workbench e execute o script. O banco de dados **empresanosql** será carregado.

6) se você executar os comandos abaixo verá o conteúdo das tabelas que serão usadas a seguir no nosso curso:

```
SELECT * FROM empresanosql.tb_funcionario;
SELECT * FROM empresanosql.tb_dependente;
SELECT * FROM empresanosql.tb_object_funcionario;
```

7) No caso do último comando (Tabela **tb_object_funcionario**) você pode copiar e colar o conteúdo do campo chamado **JSON** em um **JSON Editor** que pode ser buscado no browser. Veja que os dados estarão salvos conforme mostrado abaixo.

```
{
  "Cpf": "12345678966",
  "Sexo": "M",
  "Salario": 30000,
  "Endereco": "Rua das Flores 751 São Paulo SP",
  "Nome_Meio": "B",
  "Dependentes": [
    {
      "Sexo": "F",
      "Parentesco": "Filha",
      "Data_Nascimento": "1988-01-30T00:12:00",
      "Nome_Dependente": "Alicia"
    },
    {
      "Sexo": "F",
      "Parentesco": "Esposa",
      "Data_Nascimento": "1967-01-05T00:05:00",
      "Nome_Dependente": "Elizabeth"
    },
    {
      "Sexo": "M",
      "Parentesco": "Filho",
      "Data_Nascimento": "1988-01-01T00:04:00",
      "Nome_Dependente": "Michael"
    }
  ]
}
```

```
],  
  "Ultimo_Nome": "Silva",  
  "Primeiro_Nome": "João",  
  "Cpf_Supervisor": "33344555587",  
  "Data_Nascimento": "1965-01-01T00:09:00",  
  "Numero_Departamento": "5"  
}
```

8) Na tabela chamada tradicional podemos filtrar os dados conforme um critério (Exemplo: Todos os funcionários do sexo feminino)

```
SELECT * FROM empresanosql.tb_funcionario where sexo = 'F';
```

9) Mas o mesmo não pode ser feito se usarmos a tabela com os dados salvos no formato JSON.

```
SELECT * FROM empresanosql.tb_object_funcionario where sexo = 'F';
```

10) O MySQL fornece funções para trabalhar com conteúdos no formato JSON. Mas, para isso, o formato do campo onde o JSON é armazenado deve possuir um tipo especial chamado **JSON**.

11) Quando exibimos o conteúdo da tabela no formato JSON, o resultado vem todo em um alinhamento só. Mas podemos apresentar o resultado da consulta, na forma mais amigável, com o JSON formatado, através do comando abaixo:

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

12) Podemos extrair um conteúdo de uma propriedade do campo JSON como mostrado abaixo. (Aqui estamos exibindo o campo SEXO).

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

13) O resultado é apresentado delimitado com aspas duplas. Podemos retirá-las executando o comando abaixo:

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

14) A extração do campo de um JSON pode usar uma função específica chamada **JSON_EXTRACT**.

```
SELECT REPLACE(JSON_EXTRACT(`JSON`, "$.Sexo"), "\"", "") AS SEXO FROM tb_object_funcionario;
```

15) Para extrair todos os dados referentes a tabela de funcionários, contidos no JSON, você deve executar:

```
SELECT  
  REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf"), "\"", "") AS CPF,  
  REPLACE(JSON_EXTRACT(`JSON`, "$.Data_Nascimento"), "\"", "") AS Data_Nascimento,  
  REPLACE(JSON_EXTRACT(`JSON`, "$.Primeiro_Nome"), "\"", "") AS Primeiro_Nome,  
  REPLACE(JSON_EXTRACT(`JSON`, "$.Nome_Meio"), "\"", "") AS Nome_Meio,  
  REPLACE(JSON_EXTRACT(`JSON`, "$.Ultimo_Nome"), "\"", "") AS Ultimo_Nome,
```

```

REPLACE(JSON_EXTRACT(`JSON`, "$.Endereco"), "", "") AS Endereco,
REPLACE(JSON_EXTRACT(`JSON`, "$.Salario"), "", "") AS Salario,
REPLACE(JSON_EXTRACT(`JSON`, "$.Sexo"), "", "") AS Sexo,
REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf_Supervisor"), "", "") AS Cpf_Supervisor,
REPLACE(JSON_EXTRACT(`JSON`, "$.Numero_Departamento"), "", "") AS Numero_Departamento,
REPLACE(JSON_EXTRACT(`JSON`, "$.Dependentes"), "", "") AS Dependentes
FROM tb_object_funcionario;

```

16) Como podemos extrair o conteúdo de uma propriedade do JSON através de funções específicas do MySQL, podemos aplicar isso para filtrar o resultado. O comando abaixo, finalmente, resolve o problema que nos deparamos no início desta aula. Listar todos os funcionários do sexo feminino.

```

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

```

17) Podemos estender a consulta acima para mostrar todos os campos do JSON:

```

SELECT REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf"), "", "") AS CPF,
REPLACE(JSON_EXTRACT(`JSON`, "$.Data_Nascimento"), "", "") AS Data_Nascimento,
REPLACE(JSON_EXTRACT(`JSON`, "$.Primeiro_Nome"), "", "") AS Primeiro_Nome,
REPLACE(JSON_EXTRACT(`JSON`, "$.Nome_Meio"), "", "") AS Nome_Meio,
REPLACE(JSON_EXTRACT(`JSON`, "$.Ultimo_Nome"), "", "") AS Ultimo_Nome,
REPLACE(JSON_EXTRACT(`JSON`, "$.Endereco"), "", "") AS Endereco,
REPLACE(JSON_EXTRACT(`JSON`, "$.Salario"), "", "") AS Salario,
REPLACE(JSON_EXTRACT(`JSON`, "$.Sexo"), "", "") AS Sexo,
REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf_Supervisor"), "", "") AS Cpf_Supervisor,
REPLACE(JSON_EXTRACT(`JSON`, "$.Numero_Departamento"), "", "") AS Numero_Departamento,
REPLACE(JSON_EXTRACT(`JSON`, "$.Dependentes"), "", "") AS Dependentes
FROM tb_object_funcionario WHERE JSON_EXTRACT(`JSON`, "$.Sexo") = 'F';

```

18) Todos os filtros que são permitidos usando SQL podem ser aplicados no conteúdo JSON. Execute o comando abaixo para filtrar os dados por data.

```

SELECT * FROM tb_object_funcionario WHERE JSON_EXTRACT(`JSON`, "$.Data_Nascimento") >= '1980-01-01';

```

19) As funções de manipulação de datas pode ser aplicadas a propriedade extraída do JSON.

```

SELECT * FROM tb_object_funcionario WHERE YEAR(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) >= 1980;

```

20) Podemos mesclar critérios de filtros usando os operadores lógicos do MySQL. Execute o comando abaixo para buscar todos os funcionários que nasceram a partir de 1980 e que são do sexo masculino.

```

SELECT REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf"), "", "") AS CPF,
JSON_EXTRACT(`JSON`, "$.Data_Nascimento") as Data_Nascimento,
YEAR(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) AS Ano_Nascimento,
MONTH(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) AS Mes_Nascimento,
REPLACE(JSON_EXTRACT(`JSON`, "$.Primeiro_Nome"), "", "") AS Primeiro_Nome,
REPLACE(JSON_EXTRACT(`JSON`, "$.Sexo"), "", "") AS Sexo,
REPLACE(JSON_EXTRACT(`JSON`, "$.Salario"), "", "") AS Salario

```

```
FROM tb_object_funcionario
WHERE YEAR(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) >= 1980
AND JSON_EXTRACT(`JSON`, "$.Sexo") = 'M';
```

21) Podemos complementar a consulta acima acrescentando o critério de funcionários que ganham mais de 19000.

```
SELECT REPLACE(JSON_EXTRACT(`JSON`, "$.Cpf"), "", "") AS CPF,
JSON_EXTRACT(`JSON`, "$.Data_Nascimento") as Data_Nascimento,
YEAR(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) AS Ano_Nascimento,
MONTH(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) AS Mes_Nascimento,
REPLACE(JSON_EXTRACT(`JSON`, "$.Primeiro_Nome"), "", "") AS Primeiro_Nome,
REPLACE(JSON_EXTRACT(`JSON`, "$.Sexo"), "", "") AS Sexo,
REPLACE(JSON_EXTRACT(`JSON`, "$.Salario"), "", "") AS Salario
FROM tb_object_funcionario
WHERE YEAR(JSON_EXTRACT(`JSON`, "$.Data_Nascimento")) >= 1980
AND JSON_EXTRACT(`JSON`, "$.Sexo") = 'M'
AND JSON_EXTRACT(`JSON`, "$.Salario") >= 19000;
```

22) Podemos aplicar o agrupamento de linhas da tabelas aplicando , sobre os campos numéricos do JSON, funções de agregação. Execute o comando abaixo para obter o total de salários pagos para todos os funcionários.

```
SELECT SUM(JSON_EXTRACT(`JSON`, "$.Salario")) AS SALARIO_TOTAL FROM tb_object_funcionario;
```

23) Podemos agrupar este resultado por departamento. Execute:

```
SELECT Numero_Departamento, SUM(salario) as SALARIO_TOTAL from tb_funcionario
GROUP BY Numero_Departamento;
```