



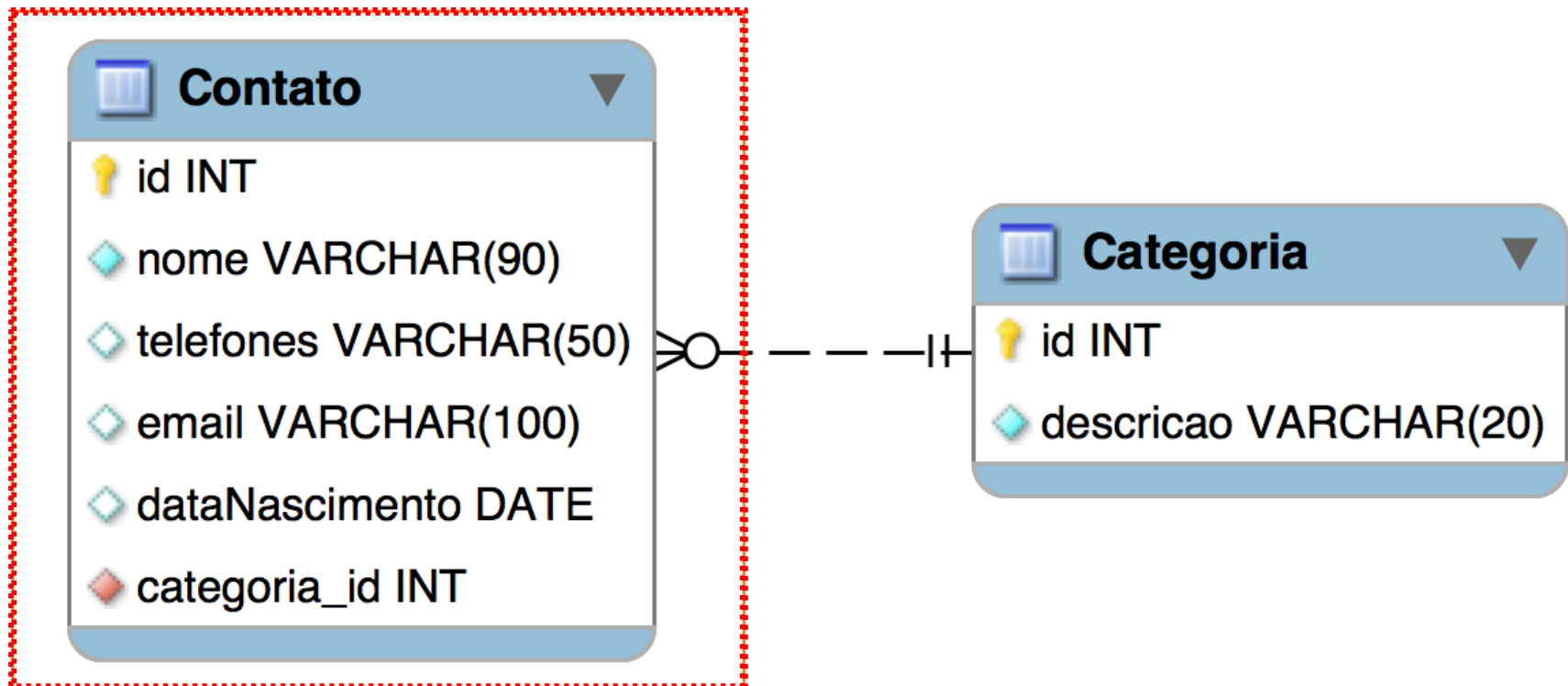
# **BANCO DE DADOS LINGUAGEM SQL**

## **FOREIGN KEY E JOINS**

**Prof. Fabiano Papaiz  
IFRN**

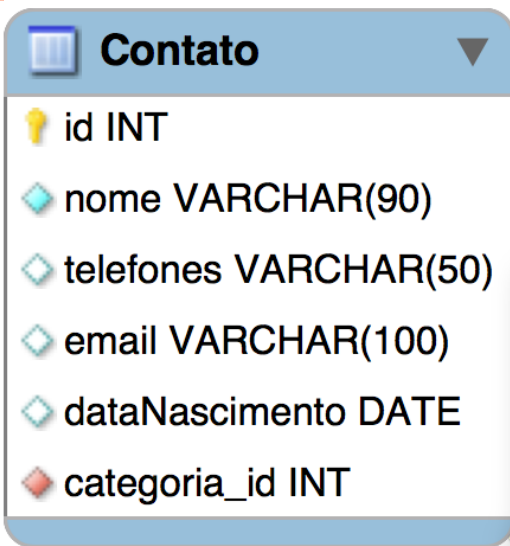
# BANCO DE DADOS

- Agora vamos criar uma tabela no BD que possui uma **chave-estrangeira**



# BANCO DE DADOS

- Criando a tabela **Contato** no BD através do comando SQL **CREATE TABLE**



A screenshot of a database table structure for 'Contato'. The table has the following columns:

id	nome	telefones	email	dataNascimento	categoria_id
INT	VARCHAR(90)	VARCHAR(50)	VARCHAR(100)	DATE	INT

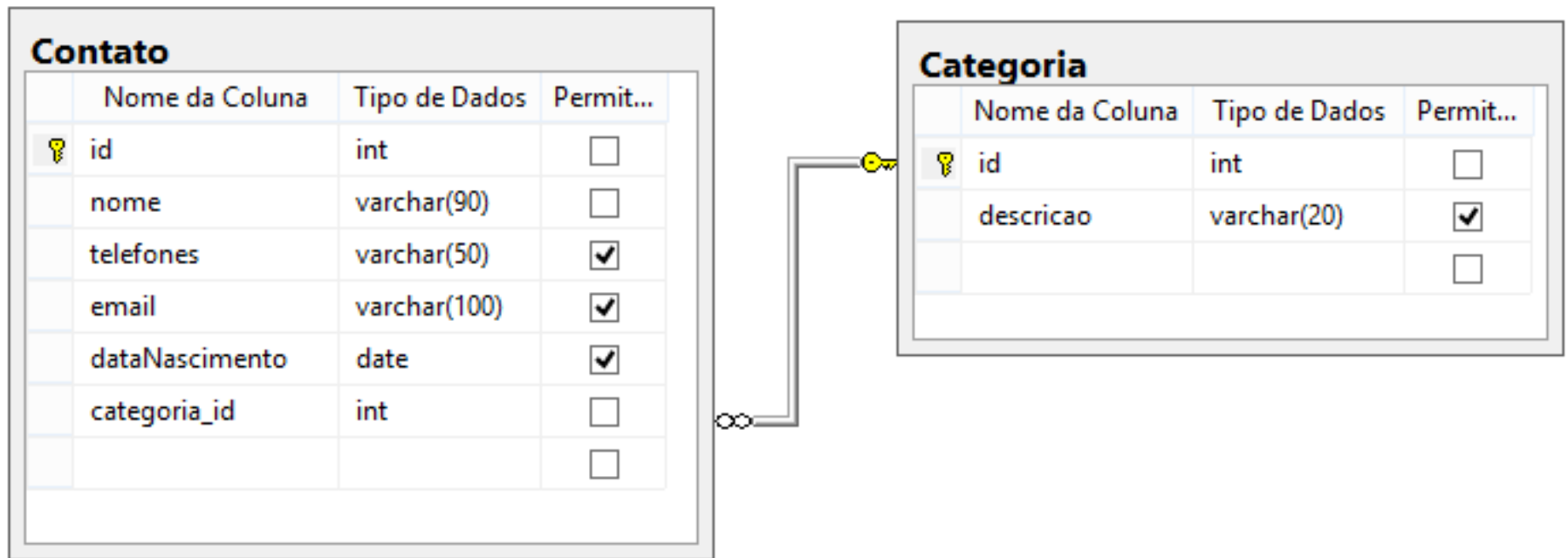


```
CREATE TABLE Contato(  
    id INT IDENTITY(1,1) NOT NULL,  
    nome VARCHAR(90) NOT NULL,  
    telefones VARCHAR(50),  
    email VARCHAR(100),  
    dataNascimento DATE,  
    categoria_id INT NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY (categoria_id) REFERENCES Categoria(id)  
)
```

- **FOREIGN KEY**: comando SQL para definir a chave estrangeira, fazendo a ligação entre o campo **categoria\_id**, da tabela **Contato**, com o campo **id** da tabela **Categoria**.

# BANCO DE DADOS

- Estrutura física das tabelas **Contato** e **Categoria** no SQL Server



# BANCO DE DADOS

- Inserindo registros na tabela **Contato** através do comando SQL **INSERT**.

```
INSERT INTO Contato(nome, telefones, email, dataNascimento, categoria_id)
VALUES('José da Silva', '3211-1233', 'jose@hotmail.com', '19950131', 1)
```

```
INSERT INTO Contato(nome, telefones, email, dataNascimento, categoria_id)
VALUES('Maria Mariana', NULL, NULL, NULL, 2)
```

```
INSERT INTO Contato(nome, categoria_id)
VALUES('João Pereira', 3)
```

- Os campos **NOT NULL** devem obrigatoriamente ser informados no comando **INSERT**.

# BANCO DE DADOS

- Alterando registros na tabela Contato

```
UPDATE Contato
  SET nome = 'João da Silva Sauro',
      telefones = '3211-1233 / 3222-4455',
      email = 'josesilva@hotmail.com',
      dataNascimento = '1993-03-25',
      categoria_id = 2
  WHERE id = 1
```

# BANCO DE DADOS

- Alterando registros na tabela Contato

```
UPDATE Contato
  SET nome = 'Maria Mariana de Mar Marinho',
      telefones = NULL
WHERE id = 2
```

```
UPDATE Contato
  SET categoria_id = 2
```

# BANCO DE DADOS

- Selecionando os registros da tabela de Contato **juntando** (*join*) com a tabela Categoria.

```
SELECT con.nome,  
       con.telefones,  
       cat.descricao as 'categoria_descricao'  
  
FROM Contato con  
INNER JOIN Categoria cat ON cat.id = con.categoria_id  
  
ORDER BY con.nome
```

- **Conceitos novos:**

- “**INNER JOIN**”: usado para juntar 2 tabelas (geralmente juntamos uma **PK** de uma tabela com uma **FK** de outra tabela)
- “**AS**”: permite definir um nome para a coluna para ser exibido no resultado
- “**Alias**”: permite definir um apelido para a tabela (**con** e **cat**) para facilitar a definição dos campos no *SELECT*, *WHERE*, *ORDER BY* etc, como também nos casos onde as tabelas possuem campos com o mesmo nome.



# BANCO DE DADOS

- Selecionando os registros da tabela de Contato **juntando** (*join*) com a tabela Categoria.

```
SELECT con.nome,  
       con.telefones,  
       cat.descricao as 'categoria_descricao'  
  
FROM Contato con  
INNER JOIN Categoria cat ON cat.id = con.categoria_id  
  
ORDER BY con.nome
```

- **Resultado do *SELECT*:**

nome	telefones	categoria_descricao
João da Silva Sauro	3211-1233 / 3222-4455	Família
João Pereira	NULL	Trabalho
Maria Mariana de Mar Marinho	NULL	Família

# BANCO DE DADOS

- **Exercícios:** criar comandos *SQL SELECT* para:
  1. Retornar os campos **id**, **nome** e **email** da tabela **Contato**, ordenados pelo nome do contato.
  2. Retornar os campos **nome** e **email** da tabela **Contato** juntamente com o campo **descricao** da tabela **Categoria**.
  3. Retornar os campos **nome** e **telefones** da tabela **Contato** juntamente com o campo **descricao** da tabela **Categoria**, ordenados inversamente pela descrição da categoria.



**FIM**