

Aluno: Luiz Henrique Martendal

```
-- criando tabela pessoa
CREATE TABLE pessoa (
    id BIGINT(8) AUTO_INCREMENT,
    nome VARCHAR(100),
    escolaridade VARCHAR(100),
    atividade VARCHAR(100),
    email VARCHAR(100),
    logradouro VARCHAR(100),
    cep VARCHAR(100),
    cidade VARCHAR(100),
    regiao VARCHAR(100),
    pais VARCHAR(100),
    telefone VARCHAR(100),
    alimentos VARCHAR(250),
    PRIMARY KEY (id)
) ENGINE=MYISAM;
```

```
-- importando dados do .csv
```

The screenshot shows a database management tool interface with a sidebar on the left listing databases like 'base_ceps', 'bd_luiz_martendal', 'bdpessoas', 'db_teste', 'furbot_daex', 'furbot_v2', 'information_schema', 'mysql', 'performance_schema', 'rest_with_spring', 'spring_boot_exp', and 'sys'. The main window displays a SQL query editor with the 'CREATE TABLE pessoa' statement. Overlaid on this is the 'Importar arquivo de texto' dialog box. The 'Arquivo de entrada' (Input file) section shows the file path 'Repositories\Graduações\FURB\5° Semestre\Banco de Dados Avançado\Unidade 1\dados_pessoas.csv'. The 'Opções' (Options) section includes checkboxes for 'Baixa prioridade, evita sobrecarregar no servidor' (checked), 'O arquivo de entrada possui números em formatos locais' (unchecked), and 'Limpar a tabela destino antes de importar' (unchecked). The 'Controle de registros duplicados' (Duplicate record control) section has 'INSERT (erros poderão acontecer)' selected. The 'Método' (Method) section has 'O cliente vai ler o conteúdo do arquivo' selected. The 'Destino' (Destination) section shows the database 'bd_luiz_martendal' and table 'pessoa'. The 'Colunas' (Columns) section shows a list of columns: 'id', 'nome', 'escolaridade', 'atividade', 'email', 'logradouro', 'cep', and 'cidade'. The 'id' column is selected as the primary key. The 'Importar!' button is highlighted.

```
-- identificar as diferentes descrições de escolaridade
SELECT DISTINCT escolaridade FROM pessoa;
```

escolaridade
Ensino Fundamental
Ensino Médio
Ensino Superior
Pós-Grad.
Ensino Superior incompleto
Graduação
Não informada
Curso Técnico Profissionalizante
Nível Superior
Pós-Graduação

```
-- associando campos parecidos
UPDATE pessoa p
SET p.escolaridade = 'Ensino Superior'
WHERE p.escolaridade = 'Graduação'
      OR p.escolaridade = 'Nível Superior'
      OR p.escolaridade = 'Pós-Grad.'
      OR p.escolaridade = 'Pós-Graduação';
```

escolaridade
Ensino Fundamental
Ensino Médio
Ensino Superior
Ensino Superior incompleto
Não informada
Curso Técnico Profissionalizante

```
-- criando tabela escolaridade
CREATE TABLE escolaridade (
    id BIGINT(8) AUTO_INCREMENT,
    escolaridade VARCHAR(100),
    PRIMARY KEY (id)
) ENGINE=MYISAM;
```

```
-- inserir os dados a partir da tabela pessoa
INSERT INTO escolaridade (descricao)
SELECT DISTINCT escolaridade FROM pessoa;
```

```
-- criando uma tabela temporaria de pessoa
CREATE TABLE pessoa_temp
AS SELECT * FROM pessoa WHERE id <= 1000;

ALTER TABLE pessoa_temp ENGINE=MYISAM
```

```

-- teste para trocar valor do campo escolaridade por id da tabela criada
UPDATE pessoa_temp pt
SET pt.escolaridade = (SELECT e.id FROM escolaridade e WHERE
e.escolaridade = pt.escolaridade);

SELECT DISTINCT pt.escolaridade FROM pessoa_temp pt;

-- script realizado na tabela pessoa
UPDATE pessoa p
SET p.escolaridade = (SELECT e.id FROM escolaridade e WHERE
e.escolaridade = p.escolaridade);

-- alterando tipo da coluna escolaridade para bigint
ALTER TABLE pessoa
MODIFY COLUMN escolaridade BIGINT(8);

-- adicionando constraint fk na coluna escolaridade da tabela pessoa
ALTER TABLE pessoa
ADD CONSTRAINT pessoa_escolaridade_fk FOREIGN KEY (escolaridade)
REFERENCES escolaridade(id);

-- alterando as tabelas para innodb
ALTER TABLE escolaridade ENGINE=INNODB;
ALTER TABLE pessoa ENGINE=INNODB;
ALTER TABLE pessoa_temp ENGINE=INNODB;

-- criando tabela atividade
CREATE TABLE atividade (
    id BIGINT(8) AUTO_INCREMENT,
    atividade VARCHAR(100),
    PRIMARY KEY (id)
);

-- adicionando atividades na tabela atividade
INSERT INTO atividade (atividade)
SELECT DISTINCT p.atividade FROM pessoa p;

-- alterando os valores da atividade na tabela pessoa para id
UPDATE pessoa p
SET p.atividade = (SELECT a.id FROM atividade a WHERE a.atividade =
p.atividade);

-- alterando tipo da coluna atividade para bigint
ALTER TABLE pessoa
MODIFY COLUMN atividade BIGINT(8);

-- adicionando a referência fk à coluna atividade
ALTER TABLE pessoa
ADD CONSTRAINT pessoa_atividade_fk
FOREIGN KEY (atividade) REFERENCES atividade(id);

-- criando tabela pessoa_alimento
CREATE TABLE pessoa_alimento
AS

```

```

SELECT id,
       trim(SUBSTRING_INDEX(SUBSTRING_INDEX(pt.alimentos, ',',
numeros.qt), ',', -1)) AS alimento
FROM pessoa pt
JOIN (
    SELECT 1 AS qt UNION
    SELECT 2 UNION
    SELECT 3 UNION
    SELECT 4 UNION
    SELECT 5 UNION
    SELECT 6 UNION
    SELECT 7 UNION
    SELECT 8 UNION
    SELECT 9
) AS numeros
ON LENGTH(pt.alimentos) - LENGTH(REPLACE(pt.alimentos, ',', '')) >=
numeros.qt - 1;

-- criando tabela alimento
CREATE TABLE alimento (
    id BIGINT(8) AUTO_INCREMENT,
    alimento VARCHAR(250),
    PRIMARY KEY (id)
);

-- inserindo valores na tabela alimento
INSERT INTO alimento (alimento)
SELECT pa.alimento FROM pessoa_alimento pa GROUP BY pa.alimento;

-- alterando nome do campo id para pessoa_id
ALTER TABLE pessoa_alimento
CHANGE COLUMN id pessoa_id BIGINT(8);

-- atualizando dados de alimento para id da tabela
UPDATE pessoa_alimento pa
SET pa.alimento = (SELECT a.id FROM alimento a WHERE a.alimento =
pa.alimento);

-- alterando nome da coluna e seu tipo
ALTER TABLE pessoa_alimento
CHANGE COLUMN alimento alimento_id BIGINT(8);

-- adicionando constraints na tabela pessoa_alimento
ALTER TABLE pessoa_alimento
ADD CONSTRAINT pessoa_alimento_alimento_fk
    FOREIGN KEY (alimento_id) REFERENCES alimento(id),
ADD CONSTRAINT pessoa_alimento_pessoa_fk
    FOREIGN KEY (pessoa_id) REFERENCES pessoa(id);

-- adicionando constraint pk para elas
ALTER TABLE pessoa_alimento
ADD CONSTRAINT pessoa_alimento_pk
    PRIMARY KEY (pessoa_id, alimento_id);
-- removendo coluna alimentos de pessoa
ALTER TABLE pessoa
DROP COLUMN alimentos;

```

Também acho que seria interessante criar uma tabela de endereços, porém, preferi o desafio da tabela alimentos e acabei ficando sem tempo... Foi um desafio muito interessante e de grande aprendizado