

Use o seguinte esquema de Banco de Dados para responder às questões abaixo.

<b>Tabelas do banco</b>	<b>Descrição de cada tabela</b>
LOCALIDADE (CODIGO_LOCAL, NOME_LOCAL)	/* tabela de localidades */
PESSOA (COD_PESSOA, SEXO, NOME, SOBRENOME, DATA_DE_NASCIMENTO, CODIGO_LOCAL_NASC, DATA_DE_FALECIMENTO, CODIGO_LOCAL_FALEC, CODIGO_UNIAO_PAIS); (CODIGO_UNIAO_PAIS) referencia UNIAO (CODIGO_UNIAO) (CODIGO_LOCAL_NASC) referencia LOCALIDADE (CODIGO_LOCAL) (CODIGO_LOCAL_FALEC) referencia LOCALIDADE (CODIGO_LOCAL)	/* tabela com dados das pessoas – CODIGO_UNIAO_PAIS identifica a união que gerou a pessoa */
UNIAO (CODIGO_UNIAO, CODIGO_LOCAL, COD_PESSOA_ESPOSA, COD_PESSOA_MARIDO, DATA_UNIAO) (COD_PESSOA_ESPOSA) referencia PESSOA (COD_PESSOA) (CODIGO_LOCAL) referencia LOCALIDADE (CODIGO_LOCAL) (COD_PESSOA_MARIDO) referencia PESSOA (COD_PESSOA)	/* tabela com as uniões (casamentos) entre pessoas */

**Sobre a base de dados acima, resolver as consultas que seguem usando SQL:**

- Obter os nomes das localidades, onde nasceram pessoas em 12/12/1900 geradas por união ocorrida em 01/01/1900.
  - Resolver usando produto cartesiano.  

```
SELECT l.nome_local
FROM localidade l, pessoa p, uniao u
WHERE l.codigo_local = p.codigo_local_nasc AND p.codigo_uniao_pais = u.codigo_uniao AND p.data_de_nascimento = '1900-12-12' AND u.data_uniao = '1900-01-01'
```
  - Resolver usando junções. Se possível, usar junção natural.  

```
SELECT l.nome_local
FROM localidade l JOIN pessoa p ON l.codigo_local = p.codigo_local_nasc JOIN uniao u ON p.codigo_uniao_pais = u.codigo_uniao
WHERE p.data_de_nascimento = '1900-12-12' AND u.data_uniao = '1900-01-01'
```
- Obter os códigos e nomes das localidades nas quais não houve uniões.  

```
SELECT *
FROM localidade
MINUS
SELECT l.codigo_local, l.nome_local
FROM localidade l JOIN uniao u ON l.codigo_local = u.codigo_local
```
- Obter **uma** única tabela contendo as seguintes colunas:
  - Código e nome de cada localidade;
  - Código e nome de cada mulher (PESSOA.SEXO = 'F') nascida na localidade (se na localidade não houver nascimentos de mulheres, estas colunas devem aparecer em branco);
  - Código de cada marido desta mulher (se a pessoa não tiver maridos, esta coluna deve aparecer vazia).

```
SELECT l.codigo_local, l.nome_local, e.cod_pessoa, e.nome, m.cod_pessoa
FROM localidade l LEFT JOIN pessoa e ON l.codigo_local = e.codigo_local_nasc AND e.sexo = 'F'
LEFT JOIN uniao u ON e.cod_pessoa = u.cod_pessoa_esposa
LEFT JOIN pessoa m ON m.cod_pessoa = u.cod_pessoa_marido
```
- Obter o nome de cada pessoa que nasceu e faleceu em duas localidades diferentes, mas cujos nomes destas localidades são iguais.  

```
SELECT p.nome
FROM pessoa p JOIN localidade ln ON p.cod_local_nasc = ln.codigo_local
JOIN localidade lf ON p.cod_local_falec = lf.codigo_local
WHERE ln.nome_local = lf.nome_local AND ln.codigo_local <> lf.codigo_local
```

5. Por um erro de software, um grande de número de uniões incorretas foi cadastrado na base de dados. Estas uniões têm a característica de ligar uma pessoa como marido de todas pessoas da localidade denominada "Guanxuma". Obter os códigos das pessoas que são marido de todas pessoas da localidade "Guanxuma".

6. Considere a seguinte expressão algébrica:

$\pi$  NOME\_LOCALIDADE

$\sigma$  PESSOA.COD\_UNIAO\_PAIS = UNIAO.COD\_UNIAO  $\wedge$  LOCALIDADE.COD\_LOCAL = UNIAO.COD\_LOCAL\_UNIAO  $\wedge$  PESSOA.SEXO = 'M'  $\wedge$  UNIAO.DATA\_UNIAO = '01/07/1750'

(PESSOA X (UNIAO X LOCALIDADE) )

- a. Exprese a consulta em SQL.

```
SELECT l.nome_local
FROM uniao u NATURAL JOIN localidade l
      JOIN pessoa p ON p.sexo = 'M' AND p.cod_uniao_pais = u.codigo_uniao
WHERE u.data_uniao = '1750-07-01'
```