

## Questão I

### SQL:

```
SELECT codProj, nomeProj FROM Projeto WHERE  
    (anoInicio BETWEEN 2001 AND 2010) AND (anoFim > 2011);
```

### Expressão algébrica:

$\pi$  codProj, nomeProj ( $\sigma$ (anoInicio  $\geq$  2001 AND anoInicio  $\leq$  2010 AND anoFim  $>$  2011)(Projeto))

## Questão II

### (a) Usando Join

#### SQL:

```
SELECT p.numeroCartao, p.nomePess, c.nomeCurso FROM Pessoa p  
    INNER JOIN Curso c ON c.codCurso = p.codCurso  
    WHERE (p.sexo = 'F') AND (c.codCurso IN (1, 2, 3));
```

### Expressão algébrica:

$\pi$ (p.numeroCartao, p.nomePess, c.nomeCurso)( $\sigma$ (p.sexo = 'F' AND (c.codCurso = 1 OR c.codCurso = 2 OR c.codCurso = 3))( $\rho$  p (Pessoa)  $\bowtie$  (c.codCurso = p.codCurso)  $\rho$  c(Curso)))

### (b) Usando produto cartesiano

#### SQL:

```
SELECT p.numeroCartao, p.nomePess, c.nomeCurso FROM Pessoa p , Curso c  
    WHERE (c.codCurso = p.codCurso) AND  
        (p.sexo = 'F') AND  
        (c.codCurso IN (1, 2, 3));
```

### Expressão algébrica:

$\pi$ (p.numeroCartao, p.nomePess, c.nomeCurso)( $\sigma$ (p.sexo = 'F' AND (c.codCurso = 1 OR c.codCurso = 2 OR c.codCurso = 3) AND (c.codCurso = p.codCurso))( $\rho$  p (Pessoa)  $\times$   $\rho$  c(Curso)))

A partir da questão III é necessário o uso álgebra relacional estendida, que é uma extensão para a álgebra relacional padrão que permite expressar funções de agregação da SQL estendida. Como o assunto não foi abordado em sala de aula, decidimos pesquisar o básico sobre o assunto e mostramos na questão III, IV, V um exemplo de como é possível criar uma expressão algébrica de uma consulta SQL nessa versão estendida da álgebra relacional, utilizando funções de agregação. Entretanto, não estão completas as expressões algébricas das questões IV e V, pois é necessário o uso de GROUP BY, ORDER BY e HAVING, assunto que não foi tratado em aula.

## Questão III

### SQL:

```
SELECT tit.nomeProj, COUNT(tit.papelPessProj)  
    FROM  
    (  
        SELECT p.nomePess, pp.PapelPessProj, proj.nomeProj  
        FROM Pessoa p JOIN projetoPessoa pp ON p.numeroCartao = pp.numeroCartao  
        JOIN Projeto proj ON pp.codProj = proj.codProj  
        WHERE p.sexo = 'M' AND pp.papelPessProj = 'Membro' AND proj.anoInicio > 2004
```

```

    ) as tit
GROUP BY tit.nomeProj
ORDER BY tit.nomeProj;

```

#### Expressão algébrica:

```

π tit.nomeProj, G(count(tit.papelPessProj))
  (π p.nomePess, pp.papelPessProj, proj.nomeProj
    σ (p.sexo = 'M' AND pp.papelPessProj = 'Membro' AND proj.anoInicio > 2004)
    ((ρ p (Pessoa) ⋈ (p.numeroCartao = pp.numeroCartao) ρ pp (projetoPessoa))
      ⋈ (pp.codProj = proj.codProj) (ρ proj (Projeto))))
  )

```

### Questão IV

#### SQL:

```

SELECT proj.codProj, proj.nomeProj, COUNT(pj.codProj) FROM Pessoa p
  INNER JOIN Curso c ON c.codCurso = p.codCurso
  INNER JOIN ProjetoPessoa pj ON pj.NumeroCartao = p.numeroCartao
  INNER JOIN Projeto proj ON proj.codProj = pj.codProj
  WHERE (c.nomeCurso = 'Ciencias da Computacao') AND (p.sexo = 'M')
  GROUP BY proj.codProj, proj.nomeProj
  ORDER BY proj.nomeProj ASC;

```

#### Expressão algébrica:

```

π proj.codProj, proj.nomeProj, G(count(pj.codProj))
( σ c.nomeCurso = 'Ciencias da Computacao' ^ p.sexo = 'M'
  (
    (
      ( ρ p (Pessoa) ⋈ (c.codCurso = p.codCurso) (ρ c (Curso)) )
      ⋈ (pj.numeroCartao = p.numeroCartao) (ρ pj ProjetoPessoa)
    )
    ⋈ (proj.codProj = pj.codProj) (ρ proj (Projeto))
  )
)

```

### Questão V

#### SQL:

```

SELECT proj.nomeProj, COUNT(pj.codProj) FROM Pessoa p
  INNER JOIN ProjetoPessoa pj ON pj.NumeroCartao = p.numeroCartao
  INNER JOIN Projeto proj ON proj.codProj = pj.codProj
  WHERE (p.sexo = 'F') AND
    (pj.papelPessProj = 'Membro')
  GROUP BY proj.codProj
  HAVING COUNT(pj.codProj) > 1;

```

**Expressão algébrica:**
$$\pi \text{ proj.nomeProj, G(count(pj.codProj))}$$
$$(\sigma \text{ pj.papelPessProj} = \text{'Membro'} \wedge \text{p.sexo} = \text{'F'}$$
$$(\quad$$
$$(\rho \text{ p (Pessoa)} \bowtie (\text{pj.numeroCartao} = \text{p.numeroCartao}) (\rho \text{ pj (ProjetoPessoa)}))$$
$$\bowtie (\text{proj.codProj} = \text{pj.codProj}) (\rho \text{ proj (Projeto)}))$$
$$)$$
$$)$$