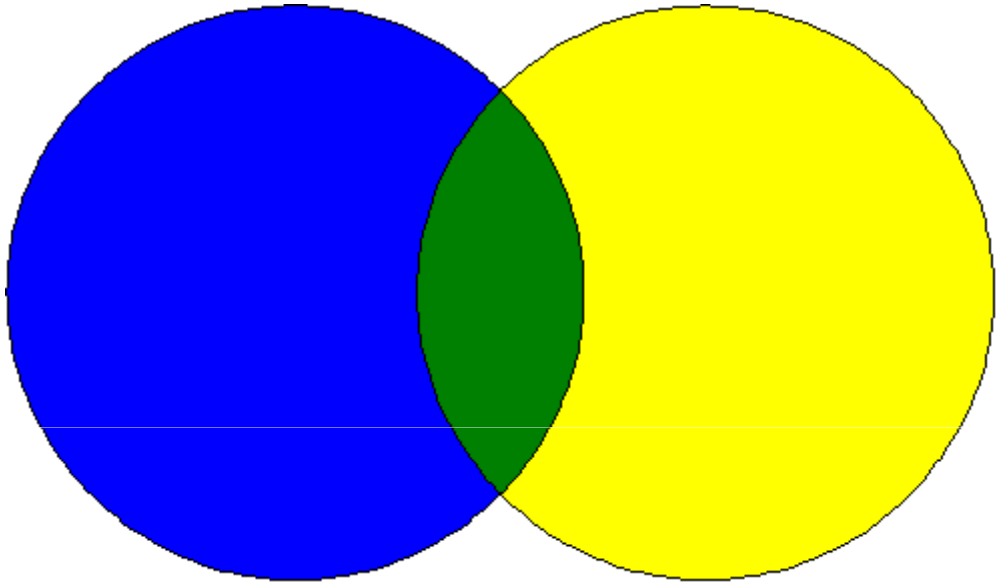
Linguagem SQL – DML
Operações binárias

Profa Dra Jeroniza Nunes Marchaukoski

Linguagem SQL DML - Álgebra Binárias

- **Operações de conjunto:**
  - **SQL possui as operações de união (union), interseção (intersect) e exceto (except).**
  - **Elas correspondem às operações ∪ (union), ∩ (intersect) e – (minus) da álgebra relacional.**

Linguagem SQL DML - Álgebra Binárias Union



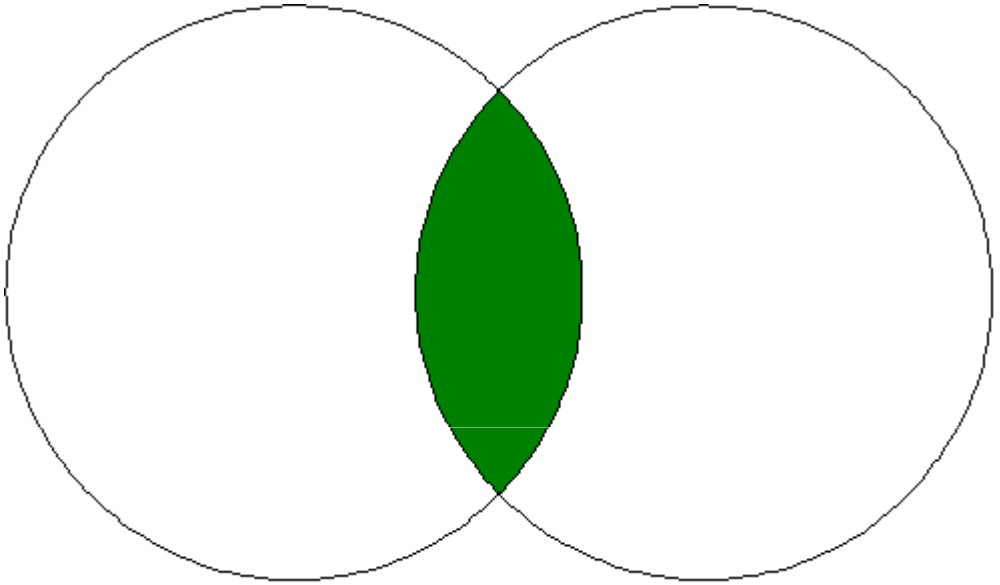
Conjunto A U Conjunto B

Linguagem SQL DML - Álgebra Binárias Union

- **Mostrar os clientes e os fornecedores**
  - **Select nome from cliente**
  - **union**
  - **Select nome from fornecedor**

  - π **nome** **(cliente)** **U** π **nome** **(fornecedor)**

Linguagem SQL DML - Álgebra Binárias Intersect



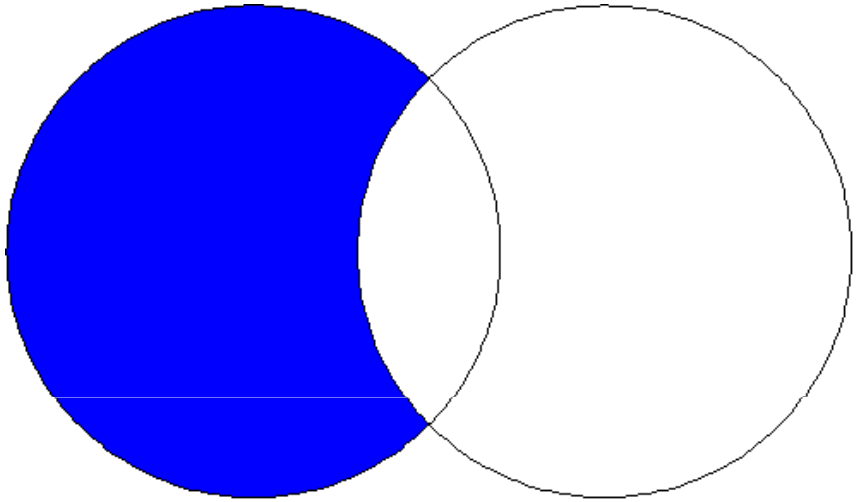
Conjunto A ∩ Conjunto B

Linguagem SQL DML - Álgebra Binárias Intersect

- **Mostrar apenas os clientes que sejam também fornecedores**
  - **Select nome from cliente**
  - **intersect**
  - **Select nome from fornecedor**

  - π **nome** **(cliente)** $\cap$ π **nome** **(fornecedor)**

Linguagem SQL DML - Álgebra Binárias Minus



Conjunto A - Conjunto B

Linguagem SQL DML - Álgebra Binárias Minus

- **Mostrar apenas os clientes que os clientes que não sejam fornecedores**
  - **Select nome from cliente**
  - **minus**
  - **Select nome from fornecedor**

  - π **nome** **(cliente)** **-** π **nome** **(fornecedor)**

Linguagem SQL DML - Álgebra

- **Produto cartesiano**

```
SELECT CURSO. Nome, ALUNO. Nome,
   ALUNO.dtNasc
FROM CURSO, ALUNO
WHERE CURSO.cod =  ALUNO.codCurso
                   PK                FK
```

Linguagem SQL DML - Álgebra

- **Produto cartesiano**
  - πaluno.nome,curso.nome (σ(aluno.curso=curso.cod_curso)**(alunoXcurso)**
  - **Usando apelido no SQL**
    - **Select a.nome as aluno, c.nome as curso**
      from aluno a, curso c
      where a.curso = c.cod_curso;

Linguagem SQL DML - Álgebra

- **Join**
  - **Inner join: junção natural**
  - **Left outer join: junção à esquerda**
  - **Right outer join: junção à direita**
  - **Full outer join: junção total**
  - **Cross join: produto cartesiano**

Linguagem SQL DML - Álgebra JOIN

- Inner join
  - Exemplo: Mostre o nome do professor e o nome da sua especialização. (usando junção natural).
    - $\pi_{professor.nome,especialização.nome}(professor \bowtie especialização)$
    - SELECT professor.nome, especialização.nome FROM professor natural inner join especialização;
    - SELECT professor.nome, especialização.nome FROM professor inner join especialização on (professor.especialização = especialização.cod);

Linguagem SQL DML - Álgebra JOIN

- Left outer join
  - Exemplo: Mostre os professores com e sem especialização (left join).
    - $\pi_{professor.nome,especialização.nome}(professor \triangleright\triangleleft especialização)$
    - SELECT professor.nome, especialização.nome FROM professor left outer join especialização on professor.especialização = especialização.cod;

Linguagem SQL DML - Álgebra JOIN

- Right outer join
  - Exemplo: Mostre as aulas com e sem professores (right join).
    - $\pi_{professor.nome,especialização.nome}$(professor ⟖ aulas)
    - SELECT professor.nome, aulas.nome FROM professor right join aulas;

Linguagem SQL DML - Álgebra JOIN

- Regras:
  - O uso de condições é obrigatório na junções externas e opcional na interna (se for omitido é gerado um produto cartesiano)
  - As palavras inner e outer são opcionais

Linguagem SQL DML - Álgebra JOIN

- Regras:
  - Condições natural: mesmo nome em ambas tabelas.
    - **SELECT * FROM COUNTRIES NATURAL JOIN CITIES**
  - On: explícita, colocando os campos e valores
    - **SELECT * FROM COUNTRIES JOIN CITIES ON (COUNTRIES.COUNTRY = CITIES.COUNTRY)**
  - Using: colocando os nomes das colunas que devem ser do mesmo tipo.
    - **SELECT * FROM COUNTRIES JOIN CITIES USING (COUNTRY) /*campo em ambas TBs*/**