Bases de Dados

Linguagem SQL – DML (Parte I)

Profa. Elaine Parros Machado de Sousa

ICMC USP SÃO CARLOS SCC

Grupo de Bases de Dados e Imagens
USP - São Carlos

DML - Introdução

- **Comandos da DML:**
  - **`INSERT`**
  - **`UPDATE`**
  - **`DELETE`**
  - **`SELECT`**

Comandos DML

- **INSERT** – insere uma ou mais tuplas em uma tabela

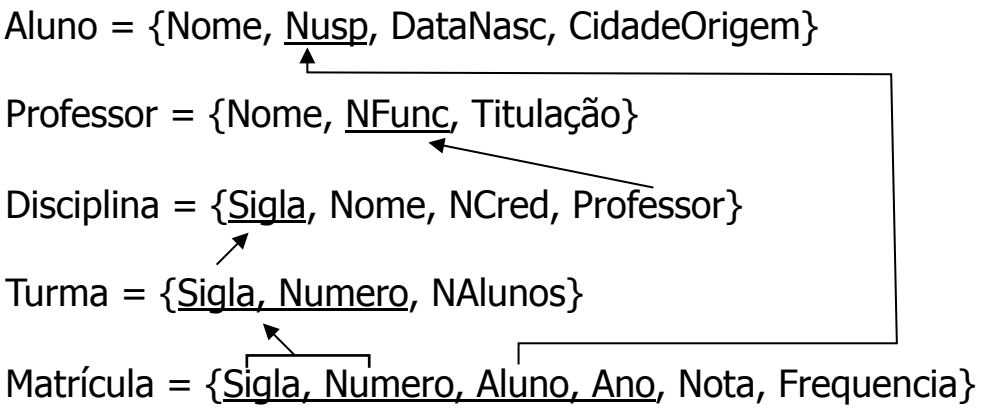
- Inserção de 1 tupla:

```
INSERT INTO tabela [(atrib1,atrib2,...)]
       VALUES (valor1, valor2,...);
```

- Inserção de múltiplas tuplas:

```
INSERT INTO tabela [(atrib1,atrib2,...)]
       <comando SELECT>;
```

Exercício



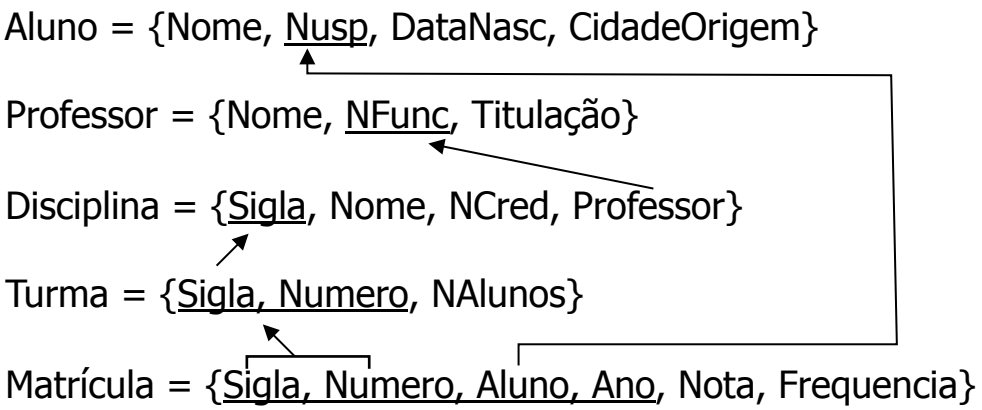
- Inserir os seguintes dados:
  - aluna de nome Juliana, nro usp 222, nascida em 10 de abril de 1989

```
-- inserção com valores para todos os atributos
-- atribuição por posição - não precisa nomear atributos
INSERT INTO Aluno
       VALUES ('JULIANA', 222,
                  TO_DATE('10/04/1989','dd/mm/yyyy'), DEFAULT);
```

  - disciplina SCC240, Bases de Dados, com 4 créditos.

```
-- inserção com apenas alguns valores da tupla
-- necessário indicar os atributos
INSERT INTO Disciplina(Sigla, Nome, NCred)
       VALUES ('SCC240', 'BASES DE DADOS', 4);
```

Exercício



- Criar uma tabela para os alunos menores de idade e alimentar com os alunos da tabela **Aluno** que têm menos de 18 anos.

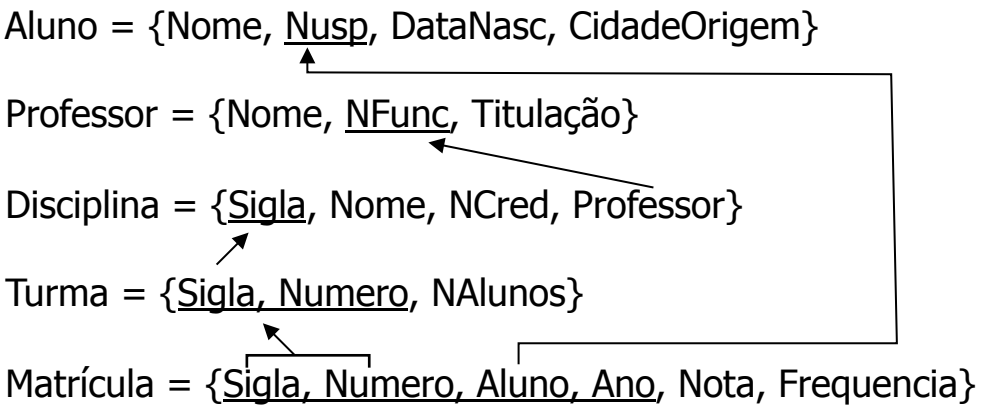
```
CREATE TABLE AlunosMenores (…);

INSERT INTO AlunosMenores
       (SELECT * FROM aluno
        WHERE ((SYSDATE - datanasc)/365) < 18);
```

Comandos DML

- **UPDATE** – modifica o valor de um atributo em uma ou mais tuplas da tabela

```
UPDATE tabela SET
      atributo1 = <valor ou expressão>,
      atributo2 = <valor ou expressão>,
      ...
WHERE <condição de localização>;
```

Exercício



- Atualizar os seguintes dados:
  - alterar para 70% a frequência de todos os alunos com nota acima de 5.0 e frequência abaixo de 70%

```
UPDATE  Matricula SET Frequencia = 70
        WHERE Nota >= 5 AND Frequencia < 70;
```

  - acrescentar um crédito para as disciplinas do departamento de Ciências de Computação (SCC)

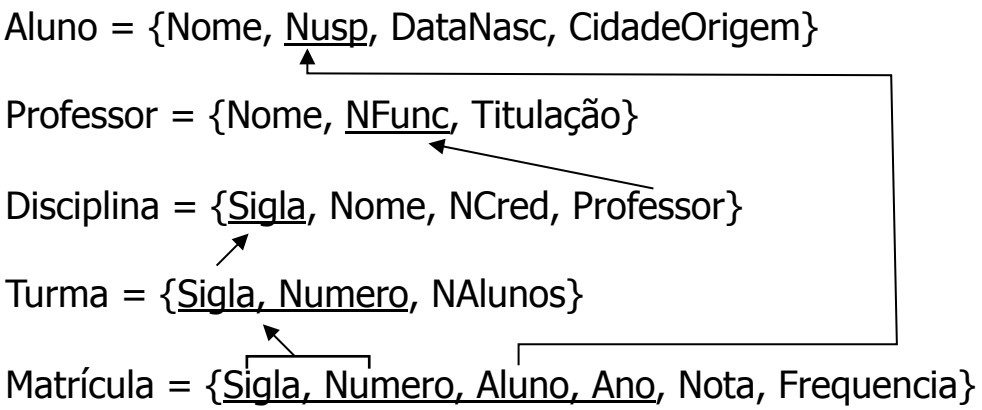
```
UPDATE  Disciplina SET Ncred = Ncred + 1
                   WHERE Sigla LIKE 'SCC%';
```

Comandos DML

- **DELETE** – remove uma ou mais tuplas da tabela

```
DELETE FROM tabela
WHERE <condição de localização>;
```

Exercícios



- Remover os seguintes dados
  - matrícula dos alunos da turma 2 de SCC240

```
DELETE FROM Matricula
       WHERE Numero = 2 AND Sigla = 'SCC240';
```

  - matrículas de alunos com 16 anos ou menos

```
DELETE FROM Matricula WHERE
  Aluno IN (SELECT Nusp FROM Aluno
            WHERE (SYSDATE - DataNasc)/365 <= 16);
```

DML - Introdução

- **Comandos da DML:**
  - `INSERT`
  - `UPDATE`
  - `DELETE`
  - **`SELECT`**

SELECT

```
SELECT [DISTINCT|ALL] <lista de
                       atributos>
FROM <lista de tabelas>
[WHERE <condições>]
[GROUP BY atributo]
     [HAVING <condições>]
[ORDER BY atributo [ASC|DESC]
```

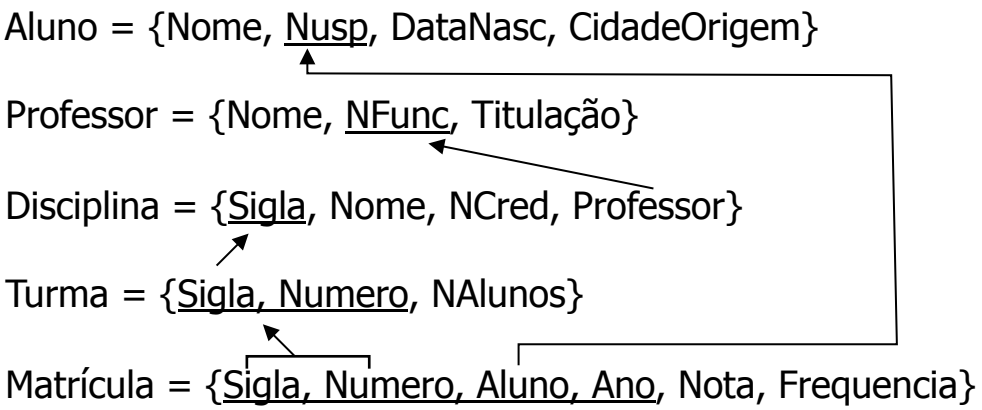
SELECT

```
SELECT [DISTINCT|ALL] <lista de
                       atributos>
FROM <lista de tabelas>
[WHERE <condições>]
[GROUP BY atributo]
     [HAVING <condições>]
[ORDER BY atributo [ASC|DESC]
```

- **SELECT** → **O QUE** se deseja na tabela resultado
  - ***<lista de atributos>*** ou
  - ***** (para todos os atributos)
  - **ALL** – resultado pode conter tuplas duplicadas (*default*)
  - **DISTINCT** – resultado contém somente tuplas distintas

- **SELECT** → **O QUE** se deseja na tabela resultado
  - ***<lista de atributos>*** ou
  - ***** (para todos os atributos)
  - **ALL** – resultado pode conter tuplas duplicadas (*default*)
  - **DISTINCT** – resultado contém somente tuplas distintas
- **FROM** → **DE ONDE** retirar os dados necessários
- **WHERE** → **CONDIÇÕES** de consulta
  - expressão condicional booleana
  - condições de seleção
  - condições de junção
  - ➢ **condições são avaliadas linha a linha (se não houver atributo indexado)**

Exercícios



1) Selecionar nome, nusp e data de nascimento de todos os alunos que vieram de São Paulo.

```
select Nome, Nusp, DataNasc
from Aluno
where UPPER(CidadeOrigem) = 'SAO PAULO';
```

SELECT

```
SELECT [DISTINCT|ALL] <lista de atributos>
FROM <lista de tabelas>
[WHERE <condições>]
[GROUP BY atributo]
        [HAVING <condições>]
[ORDER BY atributo [ASC|DESC]
```

- Cláusula **FROM** com mais de uma tabela
  - **Junção Interna** (*Inner Join*)
  - **WHERE** ⇒ condição de junção

```
SELECT [DISTINCT|ALL] <atributos>
FROM tabela1, tabela2
WHERE tabela1.atributo1 =
      tabela2.atributo2;
```

Junção Interna (*Inner Join*)

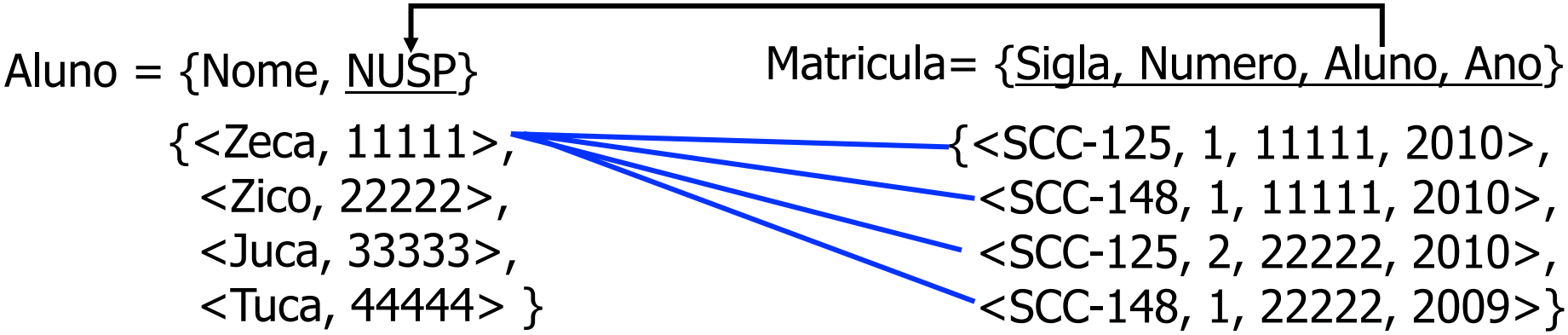
- Resultado: combinação de tuplas da ***tabela1*** com tuplas da ***tabela2***, de modo que a condição de junção seja satisfeita (`tabela1.atributo1= tabela2.atributo2`)

- Tuplas com valores nulos para atributos de junção **não estão** no resultado

Junção Interna (*Inner Join*)

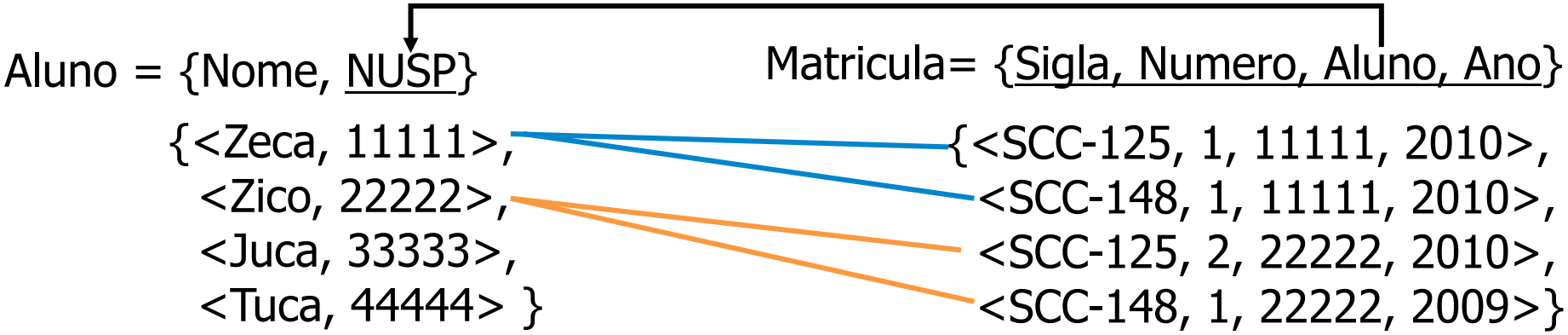
- Operação importante em bases de dados relacionais
  - usada para combinar tuplas (relacionadas) de diferentes relações em uma única tupla
  - permite **processamento de relacionamentos** entre relações
  - muito usada com **relações vinculadas por chave estrangeira**

Exemplo: Junção




```
select A.nome, A.nusp, M.sigla
from Aluno A, Matricula M
where A.nusp = M.aluno;
```

Exemplo: Junção




```
select A.nome, A.nusp, M.sigla
from Aluno A, Matricula M
where A.nusp = M.aluno;
```

{Nome, NUSP, Sigla}
{<Zeca, 11111, SCC-125>,
<Zeca, 11111, SCC-148>,
<Zico, 22222, SCC-125>,
<Zico, 22222, SCC-148 >}

Exemplo: Junção

Aluno = {Nome, NUSP}

{<Zeca, 11111>,
<Zico, 22222>,
<Juca, 33333>,
<Tuca, 44444> }

Matricula= {Sigla, Numero, Aluno, Ano}

{<SCC-125, 1, 11111, 2010>,
<SCC-148, 1, 11111, 2010>,
<SCC-125, 2, 22222, 2010>,
<SCC-148, 1, 22222, 2009>}

```
select A.nome, A.nusp, M.Sigla
from Aluno A, Matricula M;
```

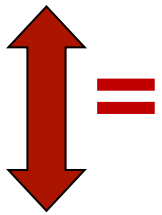
E se não colocarmos a condição de junção???

Junção Interna – operador JOIN

```
SELECT [DISTINCT|ALL] <atributos>
    FROM tabela1 T1
    [INNER] JOIN tabela2 T2
    ON T1.atributo1 =
       T2.atributo2;
```

Junção Interna

```
SELECT <atributos>
    FROM tabela1 T1 , tabela2 T2
    WHERE T1.atributo1 = T2.atributo2;
```

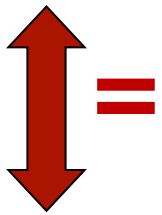


=

```
SELECT <atributos>
    FROM tabela1 T1 JOIN tabela2 T2
    ON T1.atributo1 = T2.atributo2;
```

Exemplo – Junção Interna

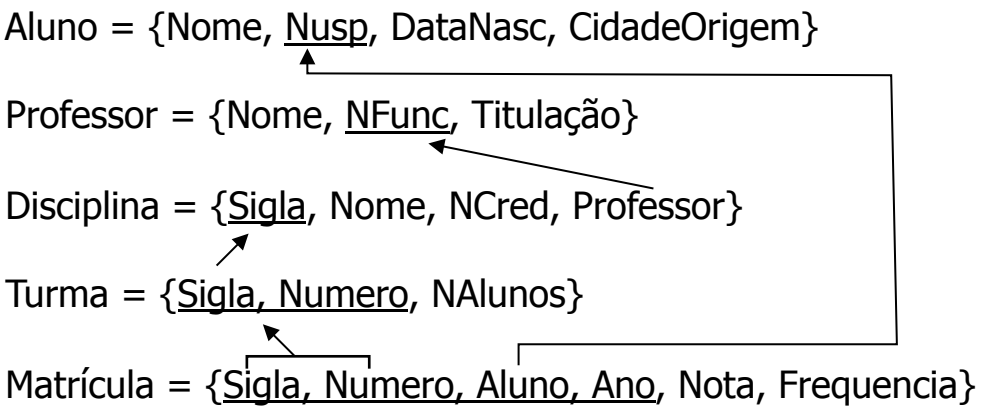
```
select A.nome, A.nusp, M.Sigla
  from Aluno A, Matricula M
  where A.nusp = M.aluno;
```



=

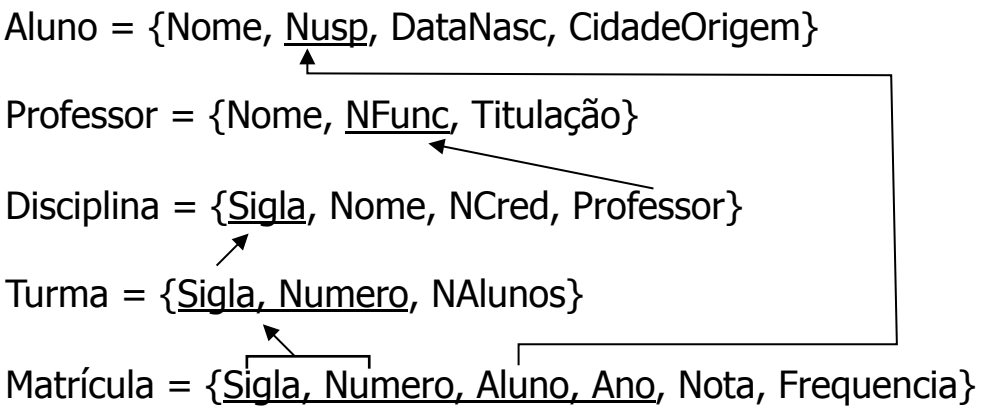
```
select A.nome, A.nusp, M.Sigla
  from Aluno A join Matricula M
  on A.nusp = M.aluno;
```

Exercícios



1) ...
2) Selecionar nome e NUSP de todos os alunos matriculados em disciplinas do SCC

Exercícios



1) ...
2) ...
3) Selecionar, para todos os alunos matriculados em disciplinas do SCC:
   a) nome e nusp, nome e sigla da disciplina, e número da turma.

Faça 2 versões equivalentes em resultado porém diferentes em eficiência...

Junção Externa (*Outer Join*)

- Extensão da operação de junção interna para consultas em que tuplas de uma tabela devem ser combinadas a tuplas de outra tabela, **mas sem desconsiderar tuplas que não têm correspondência.**
- Três tipos de junção externa
  1. Junção externa à esquerda (*Left Outer Join*)
  2. Junção externa à direita (*Right Outer Join*)
  3. Junção externa completa (*Full Outer Join)*

Junções Externas

```
SELECT [DISTINCT|ALL] <atributos>
    FROM tabela1 T1
    [LEFT | RIGHT |FULL][OUTER]JOIN
    tabela2 T2
    ON T1.atributo1 = T2.atributo2;
```

Junções Externas

- **`Left Outer Join`**
  - resultado:
    - tuplas que atendem à condição de junção

      +

    - tuplas da tabela à esquerda que não têm correspondentes na tabela à direita

Exemplo:

Aluno = {Nome, NUSP}

Matricula= {Sigla, Numero, Aluno, Ano}

{<Zeca, 11111>,
<Zico, 22222>,
<Juca, 33333>,
<Tuca, 44444> }

{<SCC-125, 1, 11111, 2010>,
<SCC-148, 1, 11111, 2010>,
<SCC-125, 2, 22222, 2010>,
<SCC-148, 1, 22222, 2009>}

Selecionar nome e nusp de **todos os alunos e, para os que estão matriculados em alguma disciplina**, retornar também sigla da disciplina

```
select A.nome, A.nusp, M.Sigla
from Aluno A left join Matricula M
on A.nusp = M.aluno;
```

{Nome, NUSP, Sigla}
{<Zeca, 11111, SCC-125>,
<Zeca, 11111, SCC-148>,
<Zico, 22222, SCC-125>,
<Zico, 22222, SCC-148>,
<Juca, 33333, NULL >,
<Tuca, 44444, NULL>}

E se invertermos a ordem das tabelas no `left join`?

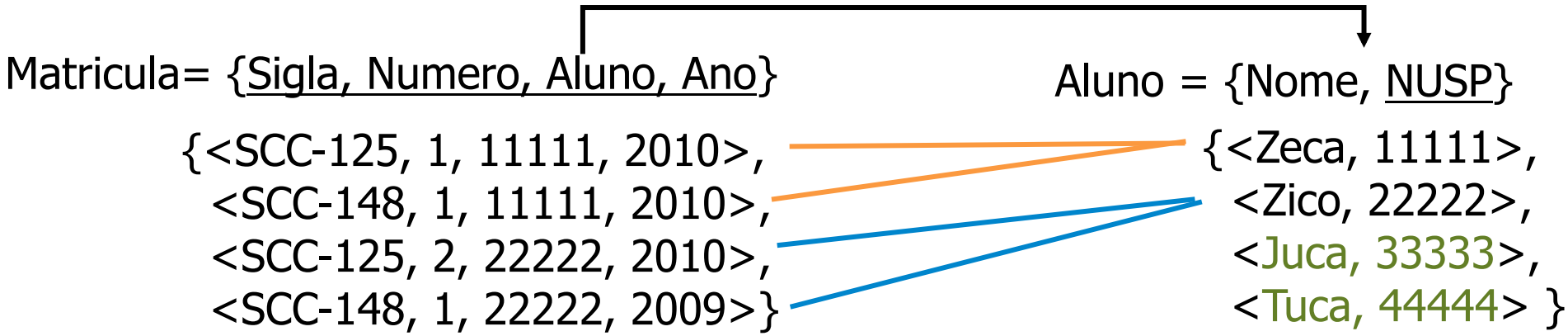
Junções Externas

- **`Right Outer Join`**
  - resultado:
    - tuplas que atendem à condição de junção

      +

    - tuplas da tabela à direita que não têm correspondentes na tabela à esquerda

Exemplo:




```
select A.nome, A.nusp, M.Sigla
from Matricula M right join Aluno A
on M.aluno = A.NUSP;
```

{Nome, NUSP, Sigla}
{<Zeca, 11111, SCC-125>,
<Zeca, 11111, SCC-148>,
<Zico, 22222, SCC-125>,
<Zico, 22222, SCC-148>,
<Juca, 33333, NULL >,
<Tuca, 44444, NULL>}

Junções Externas

- **Full Outer Join**
  - resultado:
    - tuplas que atendem à condição de junção

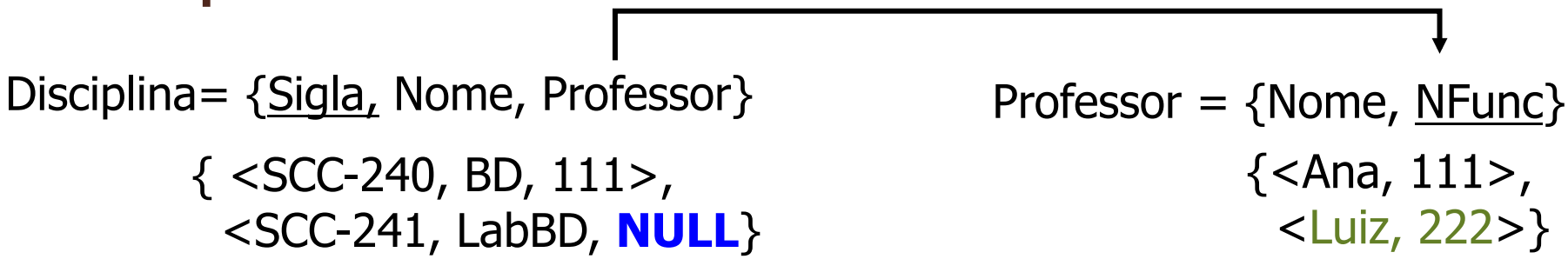
      +

    - tuplas da tabela à esquerda que não têm correspondentes na tabela à direta

      +

    - tuplas da tabela à direita que não têm correspondentes na tabela à esquerda

Exemplo:




```
select D.Sigla, P.NFunc, P.Nome
from Disciplina D full join Professor P
on D.Professor = P.NFunc;
```

{Sigla, NFunc, Nome}

{ <SCC-240, 111, Ana>,
<SCC241, NULL, NULL >,
<NULL, 222, Luiz> }

Junções Externas

(+) indica de qual tabela serão aceitos, **no resultado**, valores nulos para os atributos de junção

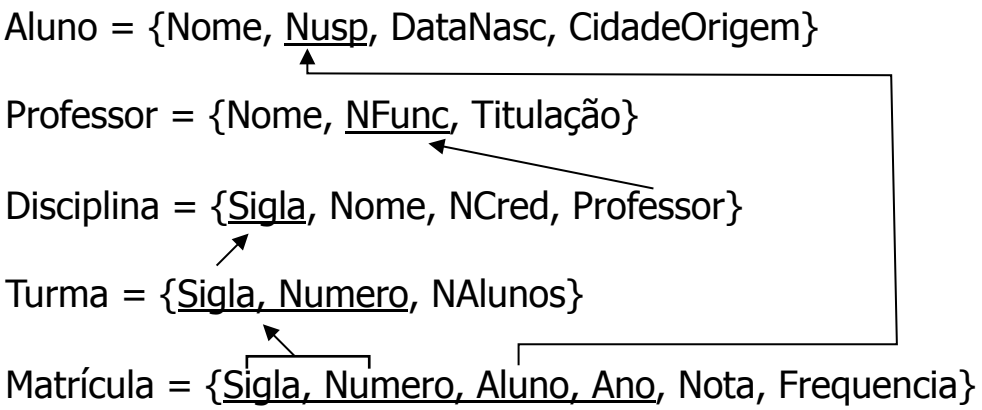
- ***LEFT JOIN COM (+)***

```
SELECT [DISTINCT|ALL] <atributos>
    FROM tabela1 T1, tabela2 T2
    WHERE T1.atributo1 = T2.atributo2 (+);
```

- ***RIGHT JOIN COM (+)***

```
SELECT [DISTINCT|ALL] <atributos>
    FROM tabela1 T1, tabela2 T2
    WHERE T1.atributo1 (+) = T2.atributo2;
```

Exercícios



1) ...
2) ...
3) ...
4) Selecionar, para **todos os alunos, nome, nusp e, para os que estiverem matriculados em alguma disciplina,** nome e sigla da disciplina.
   - Faça duas versões desta consulta e avalie eficiência

Planos de Consulta – Otimizador Oracle

Como visualizar o plano de consulta – otimizador ORACLE

TelefoneCandidato = {candidato, telefone}

```
delete from plan_table;

explain plan set statement_id = 'teste1' for
select candidato from telefonecandidato;


-- visualização do plano
SELECT lpad(' ',level-1)||operation||' '||
       options||' '||object_name "Plan"
  FROM PLAN_TABLE

CONNECT BY prior id = parent_id
       AND prior statement_id = statement_id
  START WITH id = 0
       AND statement_id = 'teste1'
  ORDER BY id;
```

Como visualizar o plano de consulta – otimizador ORACLE

- Alternativa usando o pacote DBMS

```
explain plan for
select candidato from telefonecandidato;

SELECT plan_table_output
  FROM TABLE(dbms_xplan.display());
```

Exemplo: planos de consulta – otimizador ORACLE

TelefoneCandidato = {candidato, telefone}

```
select candidato from TelefoneCandidato;
```

```
-- Plano de consulta:
SELECT STATEMENT
 INDEX FULL SCAN PK_TELEFONECANDIDATO
```

Leitura recomendada

- R. Elmasri, S. Navathe: *Sistemas de Banco de Dados*
  - 4ª Edição
    - Capítulos 8 e 9
  - 6ª Edição
    - Capítulos 4 e 5
- A. Silberschatz, H. F. Korth, s. Sudarshan: *Sistema de Banco de Dados*
    - Capítulo 4
- Oracle - Manuais em *list of books* no site
  - ***SQL Language Reference***
  - ***Performance Tuning Guide***