

SQL – Inserção

✓ **INSERÇÃO**

- Utiliza-se o comando **INSERT INTO** para incluir dados nas relações.

// não definimos a ordem, por padrão seguir ordem dos atributos no banco.

```
INSERT INTO nome_tabela  
VALUES (ordem_atributo1, ordem_atributo2,...)
```

// se definirmos a ordem dos atributos devemos inserir seus dados respectivamente

```
INSERT INTO nome_tabela (atributo1, atributo2, atributo 3)  
VALUES (valor_atributo1, valor_atributo2, valor_atributo3)
```

```
INSERT INTO nome_tabela (atributo2, atributo3, atributo 1)  
VALUES (valor_atributo2, valor_atributo3, valor_atributo1)
```


SQL – Inserção

✓ **INSERÇÃO**

➤ É possível, na inserção de tuplas, fornecer valores somente para alguns atributos do esquema

```
INSERT INTO conta  
VALUES (NULL, 'A-401',1200)
```

SQL – Inserção



Tables (4)
alunos
Columns (3)
alu_iden
alu_nome
alu_cpf
Constraints

```
1  /*SEM DEFINIR ORDEM, PEGAR ORDEM PADRÃO DA TABELA*/
2  INSERT INTO alunos VALUES (1,'MARIA JOAQUINA','874.963.111-87');
3
4  /*DEFININDO A ORDEM DOS ATRIBUTOS, TENHO QUE INSERIR NA MESMA SEQUENCIA*/
5  INSERT INTO alunos (alu_nome, alu_cpf) VALUES ('MARCELA','887.698.321-87');
6  INSERT INTO alunos (alu_cpf, alu_nome) VALUES ('222.632.541-87','RAMBO');
7  INSERT INTO alunos (alu_nome, alu_cpf) VALUES ('TARANTINO','414.587.321-99');
8
9  /*DEFININDO INSERÇÃO DE UM CAMPO NULO*/
10 INSERT INTO alunos (alu_nome, alu_cpf) VALUES ('TARANTINO',null);
11
```

➤Obs. Caso execute e a sequence afirmar que já exista o valor 1, rode novamente. Ela vai incrementar em um e tudo volta ao normal.

SQL – DML

✓ Cláusula Select

- O resultado de uma consulta de SQL é, naturalmente, uma relação/tabela.

"mostre todos os dados da tabela alunos"

1	SELECT	*
2	FROM	alunos

Data Output	Explain	Messages	Notifications
	alu_iden [PK] integer	alu_nome character varying	alu_cpf character varying
1	1	MARIA JOAQUINA	874.963.111-87
2	2	MARCELA	887.698.321-87
3	3	RAMBO	222.632.541-87
4	4	TARANTINO	414.587.321-99
5	5	TARANTINO	[null]

✓ Cláusula Select

- Nos casos em que desejamos forçar a eliminação de duplicidade, podemos inserir a palavra chave **DISTINCT** depois de **SELECT**

```
SELECT DISTINCT atributos  
FROM nome_tabela
```

```
SELECT ALL atributos  
FROM nome_tabela
```



Com ALL as duplicidades não serão eliminadas

SQL – DML



✓ Cláusula Select

- Nos casos em que desejamos forçar a eliminação de duplicidade, podemos inserir a palavra chave **DISTINCT** depois de **SELECT**

1	SELECT alu_nome
2	FROM alunos

Data Output	Explain	Message
 alu_nome character varying		
1	MARIA JOAQUINA	
2	MARCELA	
3	RAMBO	
4	TARANTINO	
5	TARANTINO	

1	SELECT DISTINCT alu_nome
2	FROM alunos

Data Output	Explain	Messages	No
 alu_nome character varying			
1	TARANTINO		
2	MARCELA		
3	RAMBO		
4	MARIA JOAQUINA		

SQL – DML

✓ Cláusula Select

- Também pode conter expressões aritméticas envolvendo os operadores **+**, **-**, ***** e **/**

```
SELECT atributo1, atributo2 * 100  
FROM nome_tabela
```

1	SELECT	adi_carga_horaria, adi_carga_horaria * 10
2	FROM	alunos_disciplinas
3		

	Data Output	Explain	Messages	Notifications
	adi_carga_horaria integer		?column? integer	
1	60		600	
2	100		1000	
3	150		1500	
4	65		650	
5	61		610	
6	40		400	

✓ Cláusula Where

- A SQL usa conectores lógicos AND, OR e NOT ao invés de símbolos matemáticos.
- Operadores dos conectivos lógicos podem ser expressões envolvendo operações de comparação: <, <=, >, >=, = e <>

```
SELECT atributos  
FROM nome_tabela  
WHERE atributo1 <= 100000  
        AND atributo2 >= 90000
```


SQL – Valores nulos

- Podemos utilizar a palavra chave **NULL** como predicado para testar a existência de valores nulos.

```
SELECT numero_emprestimo  
FROM emprestimo  
WHERE total IS NULL
```

```
SELECT numero_emprestimo  
FROM emprestimo  
WHERE total IS NOT NULL
```


SQL – DML

✓ Cláusula Where

- Obter cargas horárias (maior que 65 e menor igual a 150)

```
1  SELECT adi_carga_horaria
2  FROM alunos_disciplinas
3  WHERE adi_carga_horaria > 65 AND adi_carga_horaria <=150
```

Data Output Explain Messages Notifications

	adi_carga_horaria	
	integer	
1	100	
2	150	

SQL – DML

✓ Cláusula Where

- A SQL possui um operador de comparação **between** para simplificar a cláusula where.

```
SELECT atributos  
FROM nome_Tabela  
WHERE atributo1 BETWEEN 900 AND 1000
```

```
1 SELECT adi_carga_horaria  
2 FROM alunos_disciplinas  
3 WHERE adi_carga_horaria BETWEEN 65 AND 150
```

Data Output

Explain

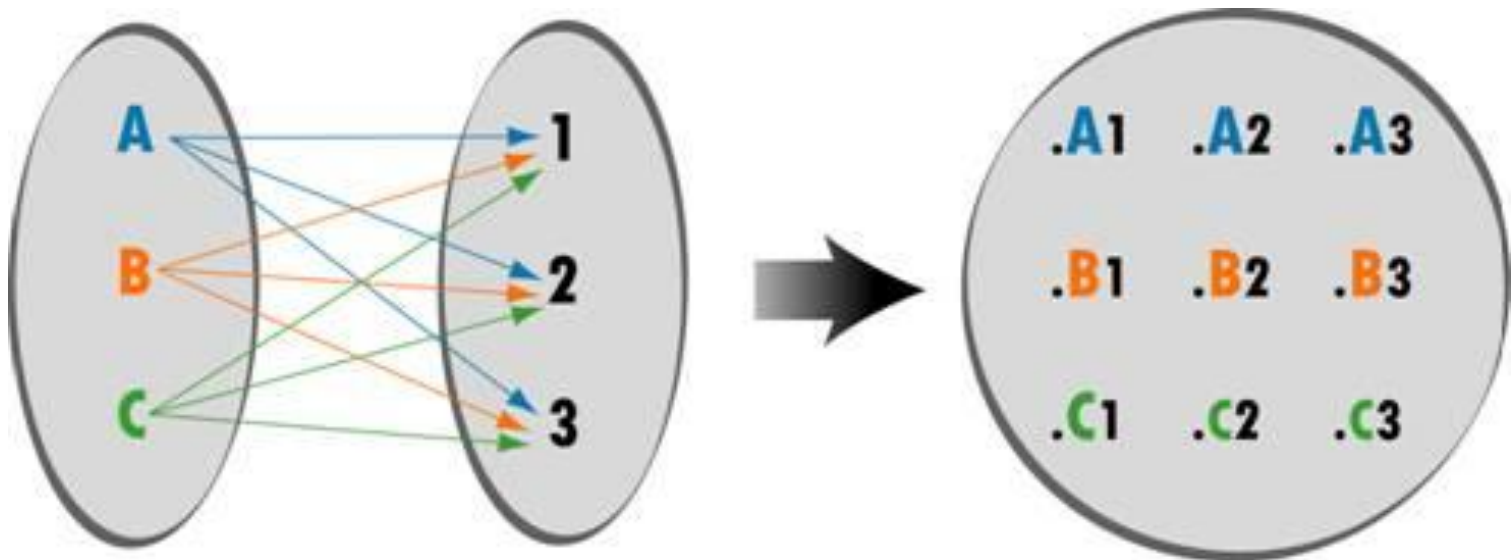
Messages

Notifications

	adi_carga_horaria	
	integer	
1	100	
2	150	
3	65	

✓ Cláusula FROM

- Deve se informar qual(is) tabela(s) são necessárias para se realizar a consulta.
- **É um produto cartesiano**

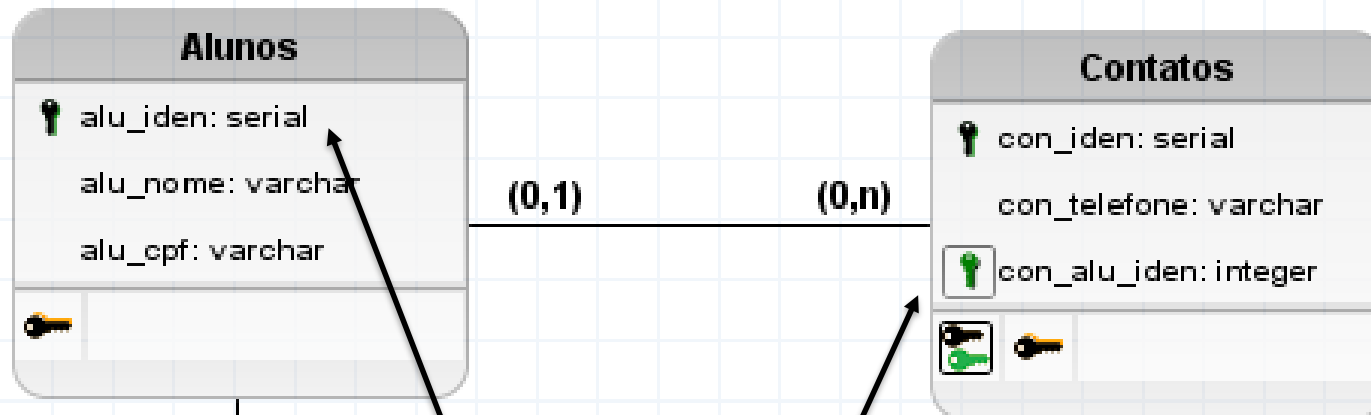


✓ Cláusula FROM

- Note que SQL usa a notação ***nome_relacionamento.nome_atributo*** para evitar ambigüidades.
- Usou no from mais de uma tabela, deve-se realizar essa comparação de chave estrangeira com chave primária da outra tabela.

```
SELECT atributos  
FROM nome_tabela1, nome_tabela2  
WHERE chave_primaria_tab1 = chave_estrangeira_tab2
```

SQL – DML

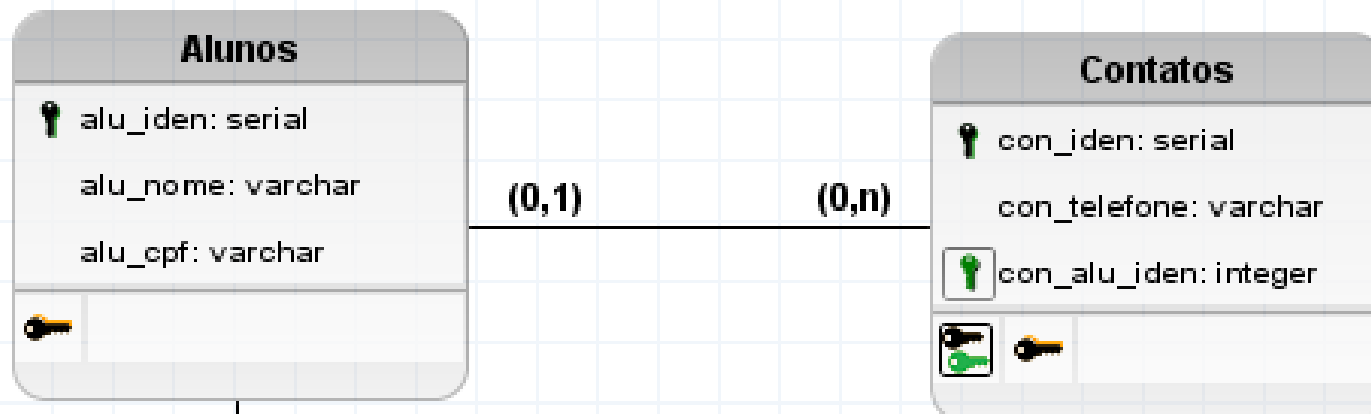


```
1  SELECT alu_nome, contatos.*
2  FROM alunos, contatos
3  WHERE alunos.alu_iden = contatos.con_alu_iden
```

Data Output Explain Messages Notifications

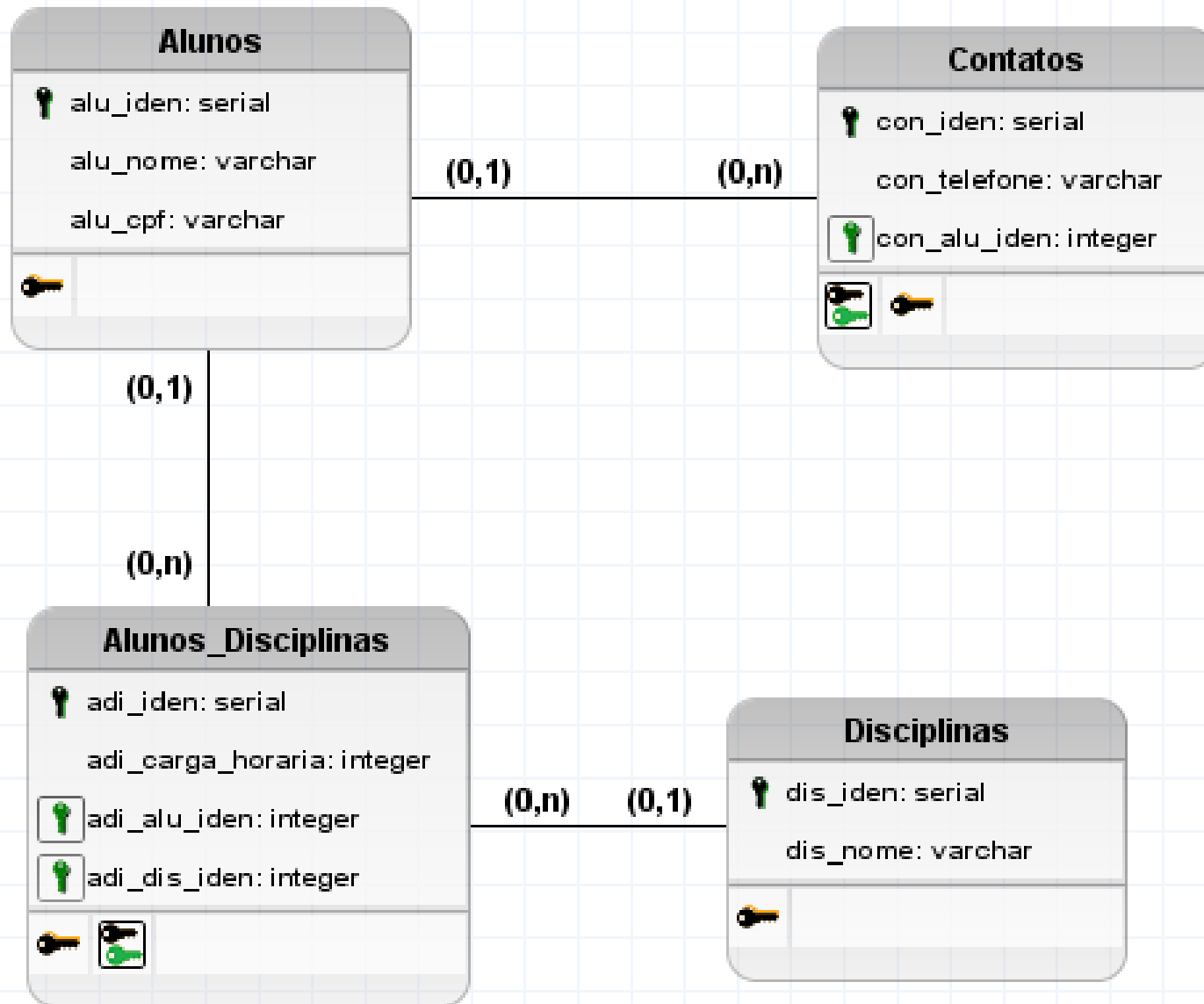
	alu_nome character varying	con_iden integer	con_telefone character varying	con_alu_iden integer
1	MARIA JOAQUINA	7	5555-5555	1
2	MARIA JOAQUINA	8	5444-5555	1
3	RAMBO	9	5333-5555	3

SQL – DML



```
1  /**JUNÇÃO SEM COMANDO JOIN*/
2  SELECT *
3  FROM alunos a, contatos c
4  WHERE a.alu_iden = c.con_alu_iden
5
6  /**USANDO JOIN*/
7  SELECT *
8  FROM alunos a JOIN contatos c
9       ON a.alu_iden = c.con_alu_iden
10
```

SQL – DML



SQL – DML

```
1  /*JUNÇÃO NA UNHA*/
2  SELECT *
3  FROM alunos a, contatos c, alunos_disciplinas ad, disciplinas d
4  WHERE a.alu_iden = c.com_alu_iden AND
5         a.alu_iden = ad.adi_alu_iden AND
6         d.dis_iden = ad.adi_dis_iden
7
8  /*USO DO JOIN*/
9  SELECT *
10 FROM alunos a JOIN contatos c ON c.com_alu_iden = a.alu_iden
11      JOIN alunos_disciplinas ad ON a.alu_iden = ad.adi_alu_iden
12      JOIN disciplinas d ON d.dis_iden = ad.adi_dis_iden
13
```