



DISCIPLINA: Banco de Dados 1

Prof. **GIOVANI** Volnei Meinerz

Aula 14 – SQL (cont.)

Objetivos da Aula

→ Entender como **inserir, modificar e excluir** informações com SQL

Inserir com SQL

→ O comando **INSERT**

→ Permite realizar a inserção de dados em uma tabela do BD

→ Sintaxe

```
INSERT INTO nome_da_tabela (A1, A2, ... , An)  
VALUES (V1, V2, ... , Vn);
```

→ Onde

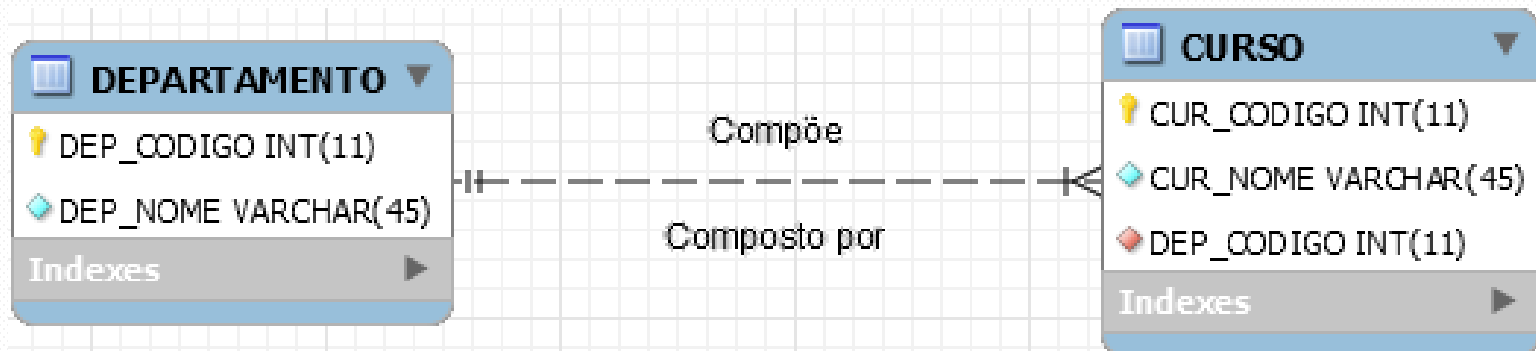
→ **A** é o atributo a receber determinado valor

→ **V** é o valor a ser atribuído a um atributo

Inserir com SQL (cont.)

➤ Cenário

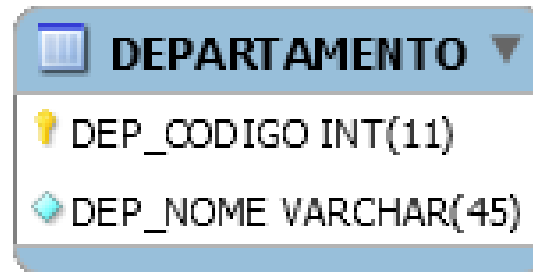
- Após criado o banco de dados e as tabelas referente ao modelo a seguir, devem ser inseridos os dados



Inserir com SQL (cont.)

→ Cenário (cont.)

→ Inserir os seguintes valores na tabela **DEPARTAMENTO**



Atributo	Valor
DEP_CODIGO	1
DEP_NOME	Departamento Acadêmico de Mecânica

Inserir com SQL (cont.)

→ Cenário (cont.)

DEPARTAMENTO ▼	
🔑	DEP_CODIGO INT(11)
💠	DEP_NOME VARCHAR(45)

→ Quando atributos e valores são especificados

```
-- Lista os atributos e os valores a serem atribuídos
INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME)
VALUES (1, 'Departamento Acadêmico de Mecânica');
```

Inserir com SQL (cont.)

→ Cenário (cont.)

DEPARTAMENTO ▼	
🔑	DEP_CODIGO INT(11)
💠	DEP_NOME VARCHAR(45)

- Quando atributos não são especificados (apenas valores)
 - Quando não listamos os atributos, os valores correspondentes devem ser especificados na ordem em que os atributos são listados na tabela

```
-- Atributos não são listados. Valores devem estar na ordem
INSERT INTO DEPARTAMENTO
    VALUES (1, 'Departamento Acadêmico de Mecânica');
```

Inserir com SQL (cont.)

→ Cenário (cont.)

DEPARTAMENTO ▼	
🔑	DEP_CODIGO INT(11)
💎	DEP_NOME VARCHAR(45)

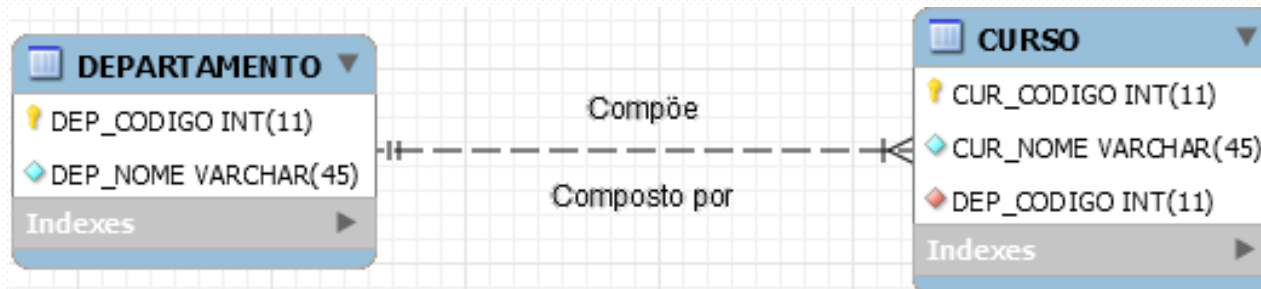
- Quando valores são especificados fora da ordem
 - Valores especificados fora da ordem requerem que os atributos correspondentes também sejam listados na mesma ordem destes

```
-- Valores listados fora da ordem
INSERT INTO DEPARTAMENTO (DEP_NOME, DEP_CODIGO)
    VALUES ('Departamento Acadêmico de Mecânica', 1);
```


Inserir com SQL (cont.)

→ Cenário (cont.)




→ Pretende-se inserir os seguintes valores na tabela **CURSO**



Atributo	Valor
CUR_CODIGO	1
CUR_NOME	Engenharia de Computação
DEP_CODIGO	2

Inserir com SQL (cont.)

→ Cenário (cont.)

CURSO	
	CUR_CODIGO INT(11)
	CUR_NOME VARCHAR(45)
	DEP_CODIGO INT(11)
Indexes	

```
select * from departamento;
```

DEP_CODIGO	DEP_NOME
1	Departamento Acadêmico de Mecânica

→ Expressão SQL está correta?

```
-- Atributos não são listados. Valores devem estar na ordem  
INSERT INTO CURSO  
VALUES (1, 'Engenharia de Computação', 2);
```

→ Expressão SQL será executada com sucesso?

Inserir com SQL (cont.)

- O comando **INSERT** poderá falhar em algumas situações, entre as quais, destacam-se:
 - Ao tentar inserir mais de uma vez o mesmo valor em um atributo definido como **chave primária**

DEPARTAMENTO ▼
DEP_CODIGO INT(11)
DEP_NOME VARCHAR(45)

```
select * from departamento;
```

DEP_CODIGO	DEP_NOME
1	Departamento Acadêmico de Mecânica

```
INSERT INTO DEPARTAMENTO  
VALUES (1, 'Departamento Acadêmico de Computação');
```

✖ 4 19:32:29 INSERT INTO DEPARTAMENTO VALUES (1, 'Departamento Acadêmico de Computação') Error Code: 1062. Duplicate entry '1' for key 'PRIMARY'

Inserir com SQL (cont.)

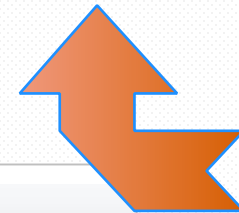
- O comando **INSERT** poderá **falhar** em algumas situações, entre as quais, destacam-se (cont.):
- Ao tentar inserir mais de uma vez o mesmo valor em um atributo definido com restrição de **unicidade**

DEPARTAMENTO
DEP_CODIGO INT(11)
DEP_NOME VARCHAR(45)
Indexes
PRIMARY
DEP_NOME_UNIQUE

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
DEP_CODIGO	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DEP_NOME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
select * from departamento;
```

DEP_CODIGO	DEP_NOME
1	Departamento Acadêmico de Mecânica



```
INSERT INTO DEPARTAMENTO  
VALUES (2, 'Departamento Acadêmico de Mecânica');
```

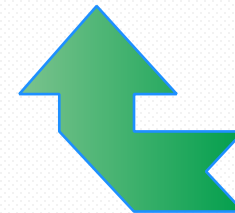
❌ 11 19:50:11 INSERT INTO DEPARTAMENTO VALUES (2, 'Departam... Error Code: 1062. Duplicate entry 'Departamento Acadêmico de Mecânica' for key 'DEP_NOME_UNIQUE'

Inserir com SQL (cont.)

- O comando **INSERT** poderá **falhar** em algumas situações, entre as quais, destacam-se (cont.):
- Ao tentar inserir um valor **NULL** em um atributo definido como **NOT NULL**

DEPARTAMENTO
DEP_CODIGO INT(11)
DEP_NOME VARCHAR(45)
Indexes
PRIMARY
DEP_NOME_UNIQUE

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
DEP_CODIGO	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DEP_NOME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	



```
INSERT INTO DEPARTAMENTO  
VALUES (2, NULL);
```

❌ 19 20:18:33 INSERT INTO DEPARTAMENTO VALUES (2, NULL) Error Code: 1048. Column 'DEP_NOME' cannot be null

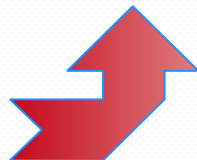
Inserir com SQL (cont.)

- O comando **INSERT** poderá **falhar** em algumas situações, entre as quais, destacam-se (cont.):
- Se um valor de um atributo (especificado na cláusula **VALUES**) não for compatível com o tipo de dado definido para o atributo

DEPARTAMENTO
DEP_CODIGO INT(11)
DEP_NOME VARCHAR(45)
Indexes
PRIMARY
DEP_NOME_UNIQUE

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
DEP_CODIGO	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DEP_NOME	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

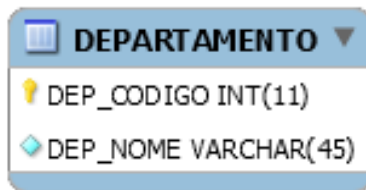
```
INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME)  
VALUES ('Dois', 'Departamento Acadêmico de Computação');
```



✖ 25 21:41:41 INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME) VALUES ('Dois'... Error Code: 1366. Incorrect integer value: 'Dois' for column 'DEP_CODIGO' at row 1

Inserir com SQL (cont.)

- O comando **INSERT** poderá falhar em algumas situações, entre as quais, destacam-se (cont.):
 - Se a quantidade de atributos for diferente da quantidade de valores associados



```
INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME)
VALUES (2, 'Departamento Acadêmico de Computação', 'Bloco K');
```

✖ 28 21:57:06 INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME) VALUES (2, 'Departam... Error Code: 1136. Column count doesn't match value count at row 1

```
INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME)
VALUES ('Departamento Acadêmico de Computação');
```

✖ 29 22:08:53 INSERT INTO DEPARTAMENTO (DEP_CODIGO, DEP_NOME) VALUES ('Departament... Error Code: 1136. Column count doesn't match value count at row 1

Modificar com SQL

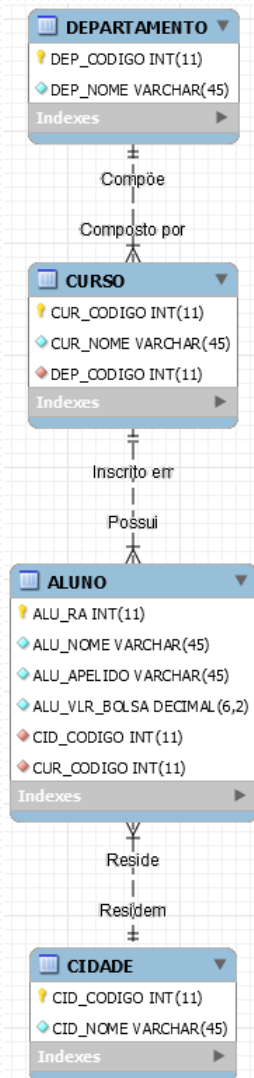
- O comando **UPDATE** permite modificar valores já existentes em atributos de uma tabela
- Sintaxe SQL

```
UPDATE nome_da_tabela  
SET Atributo1 = Expressão1 [, Consulta1 ]  
      Atributon = Expressãon [, Consultan ]  
[WHERE Condição];
```

- É possível modificar um ou mais atributos
- A cláusula **WHERE** é opcional
 - Se especificada, modificará a tupla que atender a condição
 - Se omitida, todas as tuplas serão modificadas

Modificar com SQL (cont.)

→ Considere as seguintes tabelas do BD



DEP_CODIGO	DEP_NOME
1	Departamento Acadêmico de Computação
2	Departamento Acadêmico de Elétrica
3	Departamento Acadêmico de Matemática
4	Departamento Acadêmico de Eletrônica
5	Departamento Acadêmico de Eletrotécnica
6	Departamento Acadêmico de Mecânica
NULL	NULL

CUR_CODIGO	CUR_NOME	DEP_CODIGO
1	Engenharia de Computação	1
2	Engenharia Elétrica	2
3	Engenharia Mecânica	6
4	Engenharia de Software	1
5	TADS	1
NULL	NULL	NULL

ALU_RA	ALU_NOME	ALU_APELIDO	ALU_VLR_BOLSA	CID_CODIGO	CUR_CODIGO
1	Johannes	Come Muito	550.00	1	1
2	Sebastian	Dorminhoco	700.00	2	3
3	Lisa	Nerd	400.00	3	1
4	Michaela	Come Pouco	480.00	1	4
5	Hanka	Dorme Pouco	850.00	2	2
NULL	NULL	NULL	NULL	NULL	NULL

CID_CODIGO	CID_NOME
1	Cornélio Procópio
2	Londrina
3	Marinóia
NULL	NULL

Modificar com SQL (cont.)

→ Cenário

→ Aumentar o valor das bolsas de estudo dos alunos em 5%

Table: **aluno**

Columns:

ALU_RA	int(11) PK
ALU_NOME	varchar(45)
ALU_APELIDO	varchar(45)
ALU_VLR_BOLSA	decimal(6,2)
CID_CODIGO	int(11)
CUR_CODIGO	int(11)

Antes do
aumento

Nome	Apelido	Bolsa
Johannes	Come Muito	550.00
Sebastian	Dorminhoco	700.00
Lisa	Nerd	400.00
Michaela	Come Pouco	480.00
Hanka	Dorme Pouco	850.00

```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa
5 FROM
6     aluno;
```

```
UPDATE ALUNO
SET ALU_VLR_BOLSA = ALU_VLR_BOLSA * 1.05
```

Após o
aumento

Nome	Apelido	Bolsa
Johannes	Come Muito	577.50
Sebastian	Dorminhoco	735.00
Lisa	Nerd	420.00
Michaela	Come Pouco	504.00
Hanka	Dorme Pouco	892.50

Modificar com SQL (cont.)

→ Cenário

- Após um aumento geral, aplicar mais 2% sobre as bolsas dos alunos cuja naturalidade não seja Cornélio Procópio

Table: **aluno**

Columns:

<u>ALU_RA</u>	int(11) PK
ALU_NOME	varchar(45)
ALU_APELIDO	varchar(45)
ALU_VLR_BOLSA	decimal(6,2)
CID_CODIGO	int(11)
CUR_CODIGO	int(11)

Antes do aumento

Nome	Apelido	Bolsa	Código da Cidade	Cidade
Johannes	Come Muito	577.50	1	Cornélio Procópio
Sebastian	Dorminhoco	735.00	2	Londrina
Lisa	Nerd	420.00	3	Marinóá
Michaela	Come Pouco	504.00	1	Cornélio Procópio
Hanka	Dorme Pouco	892.50	2	Londrina

```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa,
5     a.cid_codigo AS 'Código da Cidade',
6     cid_nome AS Cidade
7 FROM
8     aluno AS a NATURAL JOIN cidade AS c;
```

```
10 • UPDATE ALUNO
11 SET ALU_VLR_BOLSA = ALU_VLR_BOLSA * 1.02
12 WHERE CID_CODIGO <> 1;
```

Table: **cidade**

Columns:

<u>CID_CODIGO</u>	int(11) PK
CID_NOME	varchar(45)

Após o aumento

Nome	Apelido	Bolsa	Código da Cidade	Cidade
Johannes	Come Muito	577.50	1	Cornélio Procópio
Sebastian	Dorminhoco	749.70	2	Londrina
Lisa	Nerd	428.40	3	Marinóá
Michaela	Come Pouco	504.00	1	Cornélio Procópio
Hanka	Dorme Pouco	910.35	2	Londrina

Modificar com SQL (cont.)

→ Cenário

- Modificar o apelido do(s) aluno(s) do curso de “Engenharia Mecânica”, cujo valor de bolsa é superior a R\$500,00, para “Ganha Demais”

Table: **aluno**

Columns:

<u>ALU_RA</u>	int(11) PK
ALU_NOME	varchar(45)
ALU_APELIDO	varchar(45)
ALU_VLR_BOLSA	decimal(6,2)
<u>CID_CODIGO</u>	int(11)
<u>CUR_CODIGO</u>	int(11)

```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa,
5     a.cur_codigo AS 'Código do Curso',
6     cur_nome AS Curso
7 FROM
8     aluno AS a NATURAL JOIN curso AS c;
```

Antes da
modificação

Nome	Apelido	Bolsa	Código do Curso	Curso
Johannes	Come Muito	577.50	1	Engenharia de Computação
Sebastian	Dorminhoco	749.70	3	Engenharia Mecânica
Lisa	Nerd	428.40	1	Engenharia de Computação
Michaela	Come Pouco	504.00	4	Engenharia de Software
Hanka	Dorme Pouco	910.35	2	Engenharia Elétrica

```
10 • UPDATE aluno
11 SET alu_apelido = 'Ganha Demais'
12 WHERE alu_vlr_bolsa > 500 AND cur_codigo = 3;
```

Após a
modificação

Nome	Apelido	Bolsa	Código do Curso	Curso
Johannes	Come Muito	577.50	1	Engenharia de Computação
Sebastian	Ganha Demais	749.70	3	Engenharia Mecânica
Lisa	Nerd	428.40	1	Engenharia de Computação
Michaela	Come Pouco	504.00	4	Engenharia de Software
Hanka	Dorme Pouco	910.35	2	Engenharia Elétrica

Table: **curso**

Columns:

<u>CUR_CODIGO</u>	int(11) PK
CUR_NOME	varchar(45)
DEP_CODIGO	int(11)

Excluir com SQL

- O comando **DELETE** permite excluir registros de uma tabela
 - Exclui registros inteiros
 - Não permite excluir apenas valores de atributos particulares

→ Sintaxe SQL

```
DELETE FROM nome_da_tabela  
[WHERE Condição];
```

- A cláusula **WHERE** pode ser omitida quando todos os registros da tabela precisarem ser excluídos

Excluir com SQL (cont.)

→ Cenário

- Excluir os alunos cujo valor de bolsa seja inferior a R\$500,00

Table: **aluno**

Columns:

ALU_RA int(11) PK
ALU_NOME varchar(45)
ALU_APELIDO varchar(45)
ALU_VLR_BOLSA decimal(6,2)
CID_CODIGO int(11)
CUR_CODIGO int(11)

Antes da
exclusão

Nome	Apelido	Bolsa
Johannes	Come Muito	577.50
Sebastian	Dorminhoco	749.70
Lisa	Nerd	428.40
Michaela	Come Pouco	504.00
Hanka	Dorme Pouco	910.35

```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa
5 FROM
6     aluno;
```

```
9 • DELETE FROM aluno
10 WHERE alu_vlr_bolsa < 500;
```

Após a
exclusão

Nome	Apelido	Bolsa
Johannes	Come Muito	577.50
Sebastian	Dorminhoco	749.70
Michaela	Come Pouco	504.00
Hanka	Dorme Pouco	910.35

Excluir com SQL (cont.)

→ Cenário

- Excluir os alunos cuja naturalidade não seja Cornélio Procópio

Table: **aluno**

Columns:

<u>ALU_RA</u>	int(11) PK
ALU_NOME	varchar(45)
ALU_APELIDO	varchar(45)
ALU_VLR_BOLSA	decimal(6,2)
CID_CODIGO	int(11)
CUR_CODIGO	int(11)

Antes da
exclusão

Nome	Apelido	Bolsa	Código da Cidade	Cidade
Johannes	Come Muito	577.50	1	Cornélio Procópio
Sebastian	Dorminhoco	749.70	2	Londrina
Michaela	Come Pouco	504.00	1	Cornélio Procópio
Hanka	Dorme Pouco	910.35	2	Londrina

```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa,
5     a.cid_codigo AS 'Código da Cidade',
6     cid_nome AS Cidade
7 FROM
8     aluno AS a NATURAL JOIN cidade AS c;
```

```
11 • DELETE FROM aluno
12 WHERE cid_codigo != 1;
```

Table: **cidade**

Columns:

<u>CID_CODIGO</u>	int(11) PK
CID_NOME	varchar(45)

Após a
exclusão

Nome	Apelido	Bolsa	Código da Cidade	Cidade
Johannes	Come Muito	577.50	1	Cornélio Procópio
Michaela	Come Pouco	504.00	1	Cornélio Procópio

Excluir com SQL (cont.)

→ Cenário

→ Excluir todos os registros de alunos

Antes da exclusão

Nome	Apelido	Bolsa	Código da Cidade	Cidade	Código do Curso	Curso
Johannes	Come Muito	577.50	1	Cornélio Procópio	1	Engenharia de Computação
Michaela	Come Pouco	504.00	1	Cornélio Procópio	4	Engenharia de Software

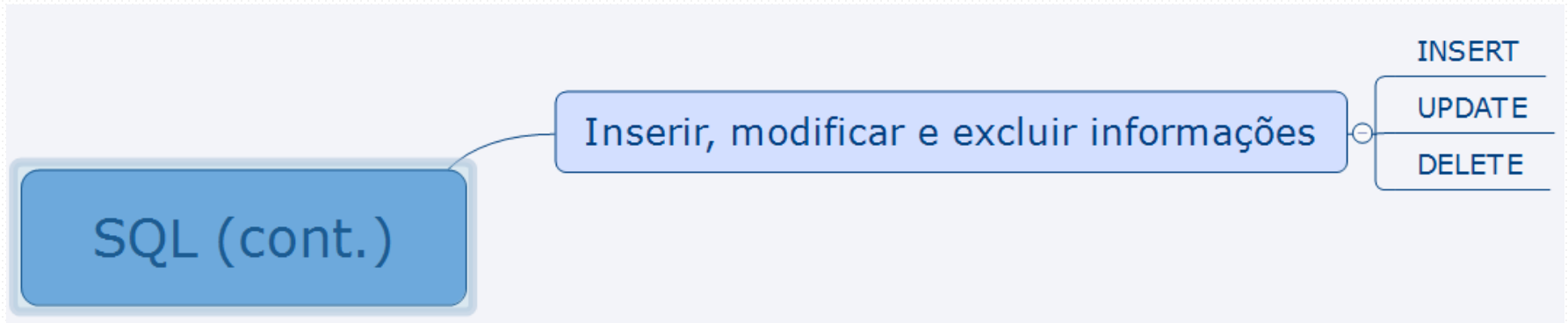
```
1 • SELECT
2     alu_nome AS Nome,
3     alu_apelido AS Apelido,
4     alu_vlr_bolsa AS Bolsa,
5     aluno.cid_codigo AS 'Código da Cidade',
6     cid_nome AS Cidade,
7     aluno.cur_codigo AS 'Código do Curso',
8     cur_nome AS Curso
9 FROM
10  aluno NATURAL JOIN cidade NATURAL JOIN curso;
```

```
13 • DELETE FROM aluno;
```

Após a exclusão

Nome	Apelido	Bolsa	Código da Cidade	Cidade	Código do Curso	Curso

Resumo da Aula





DISCIPLINA: Banco de Dados 1

Prof. **GIOVANI** Volnei Meinerz

Aula 14 – SQL (cont.)