



BandTec

DIGITAL SCHOOL

BD – Banco de Dados

Modelagem Lógica – MySQL Workbench
Entidade Forte e Entidade Fraca (Revisão)
Funções: sum, avg, max, min, count

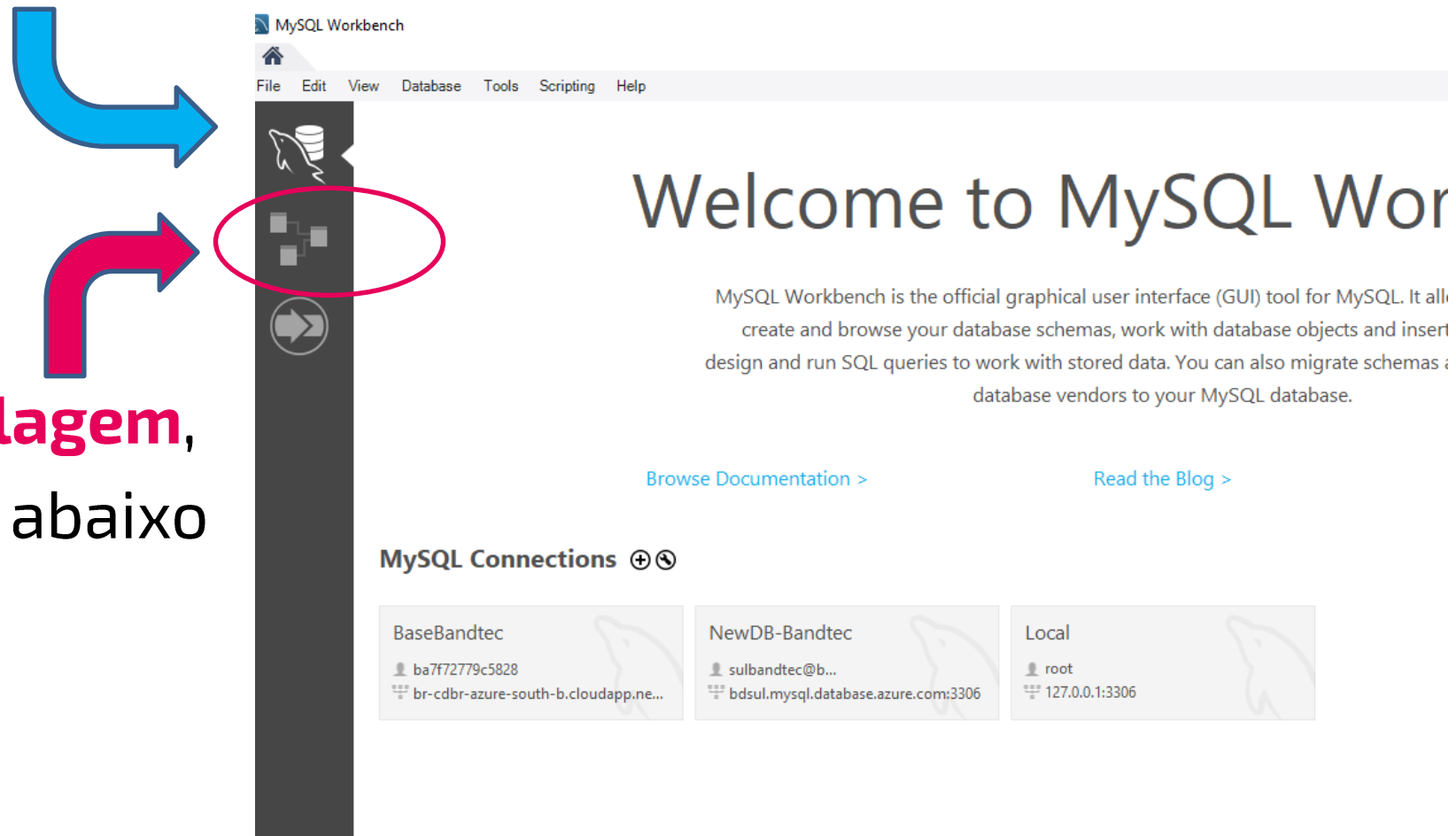
Modelagem Lógica vs Modelagem Conceitual

- Modelagem Conceitual (usamos o brModelo)
 - 1º nível da modelagem do sistema de banco de dados
 - Retrata uma visão macro do banco de dados
 - NÃO deve apresentar as chaves estrangeiras, nem as tabelas associativas
- Modelagem Lógica (utilizaremos o MySQL Workbench)
 - 2º nível da modelagem do sistema de banco de dados
 - Já se sabe qual SGBD será utilizado
 - Apresenta as chaves estrangeiras e as tabelas associativas

Modelagem Lógica – MySQL Workbench

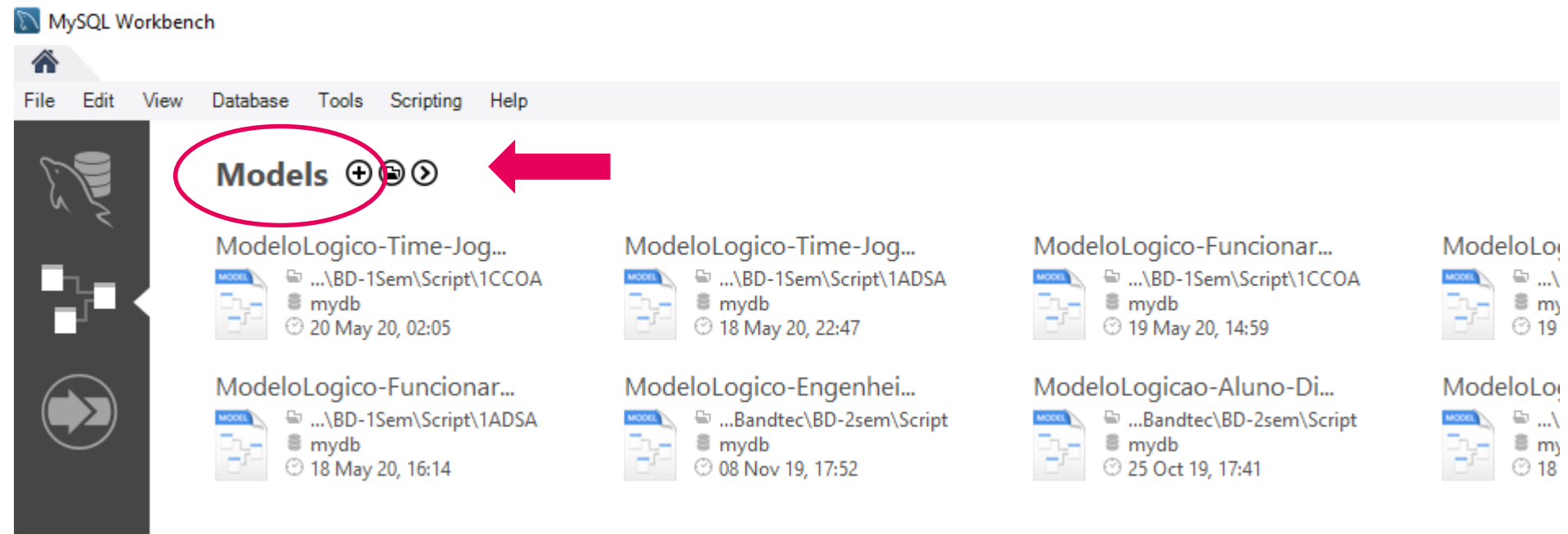
- Inicie o MySQL Workbench
- Normalmente, na página inicial, está selecionado o ícone do Golfinho (para conexão com o Server e execução de comandos)

- Para a **modelagem**, clique no ícone abaixo ao do golfinho



Modelagem Lógica – MySQL Workbench

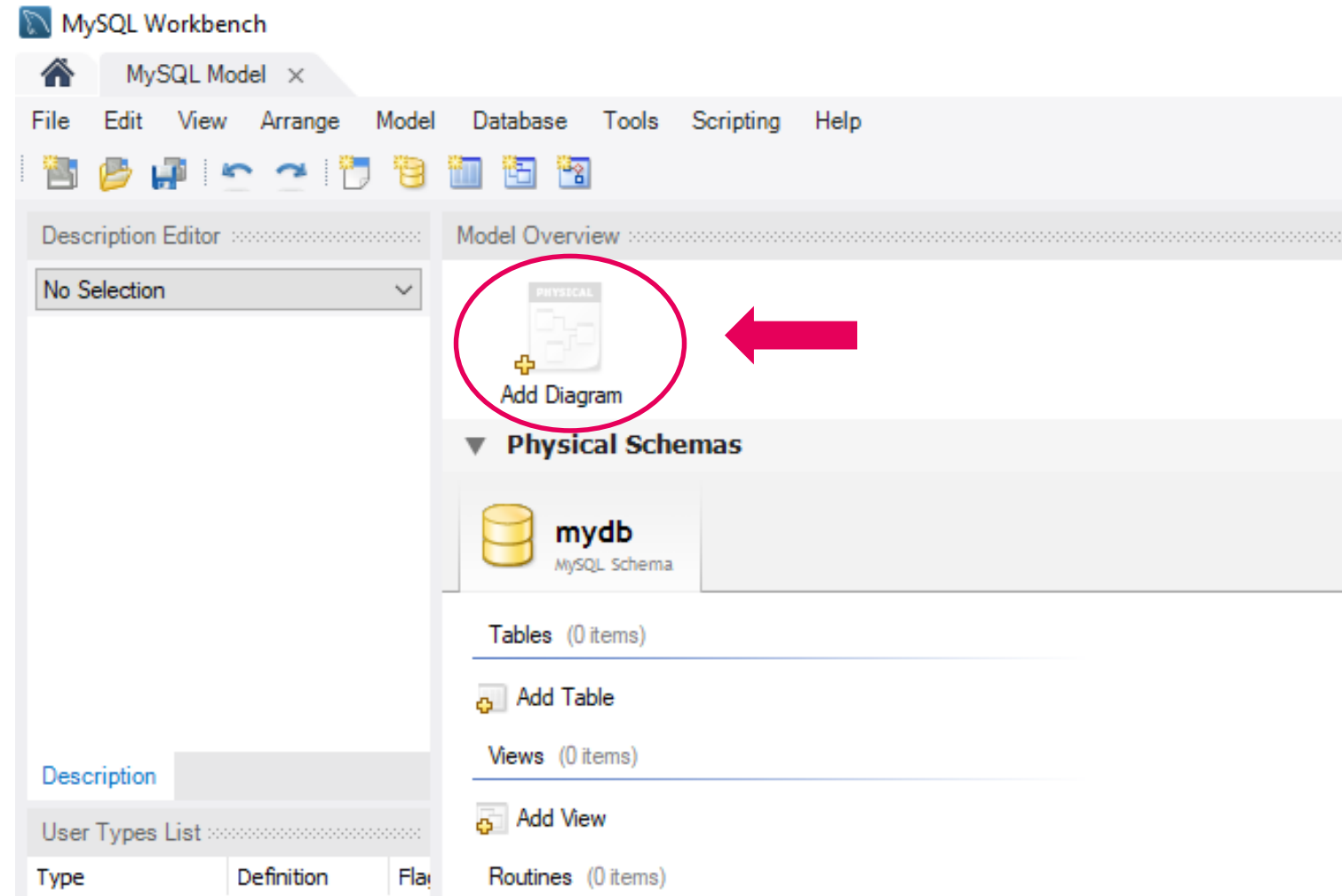
- Aparecerá esta página:



- Clique no sinal de \oplus ao lado de Models

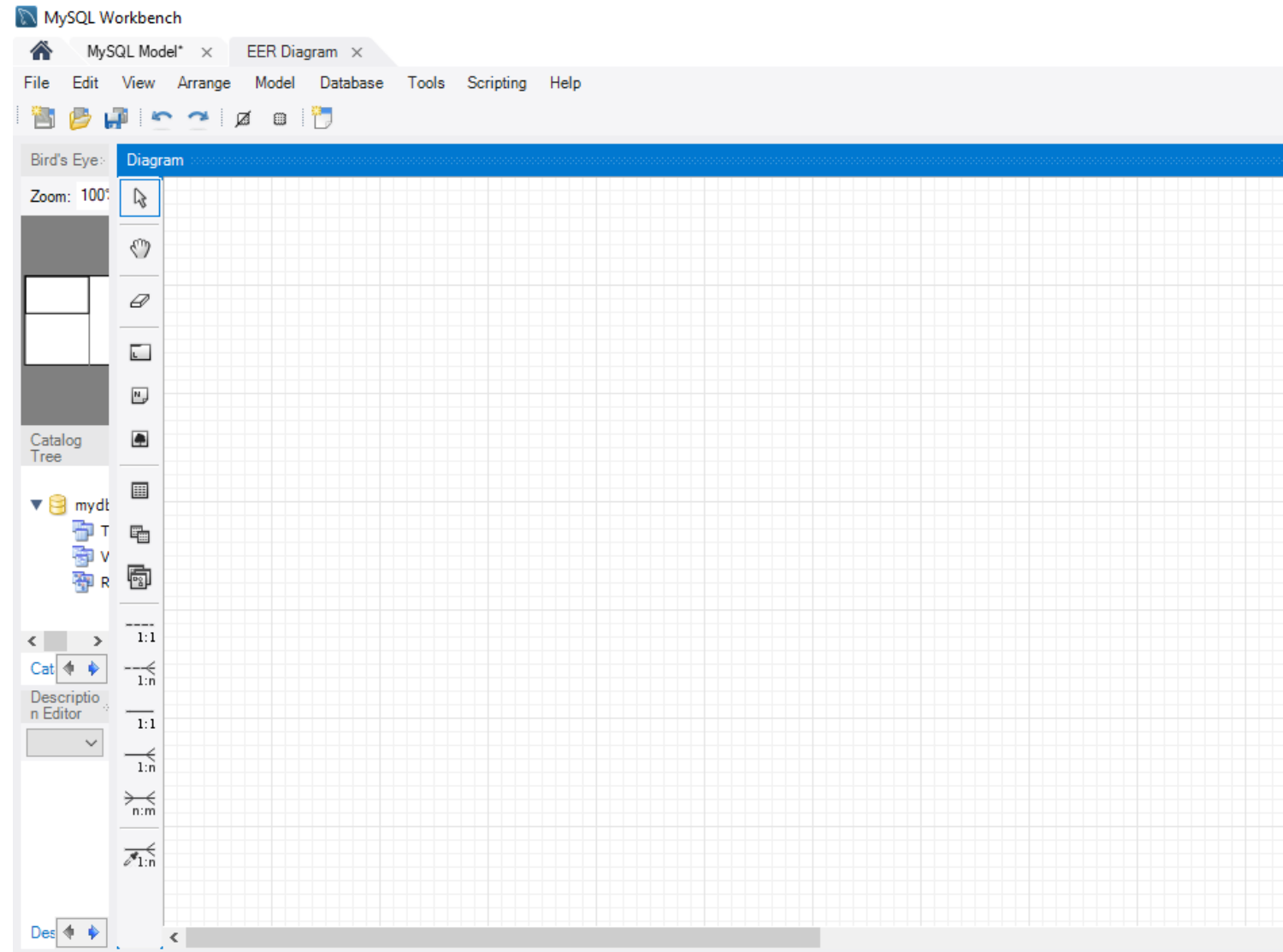
Modelagem Lógica – MySQL Workbench

- Esta aba será aberta:
- Dê um duplo clique em Add Diagram



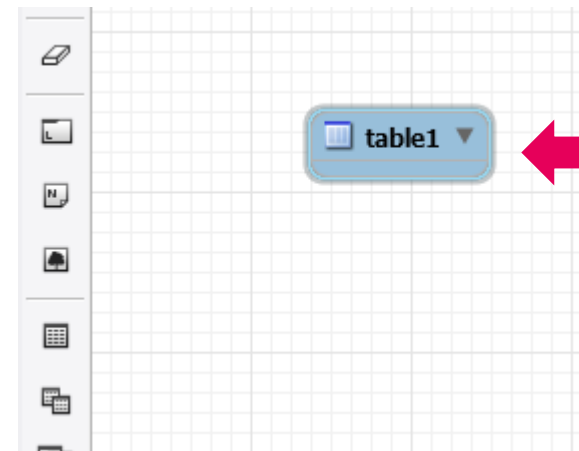
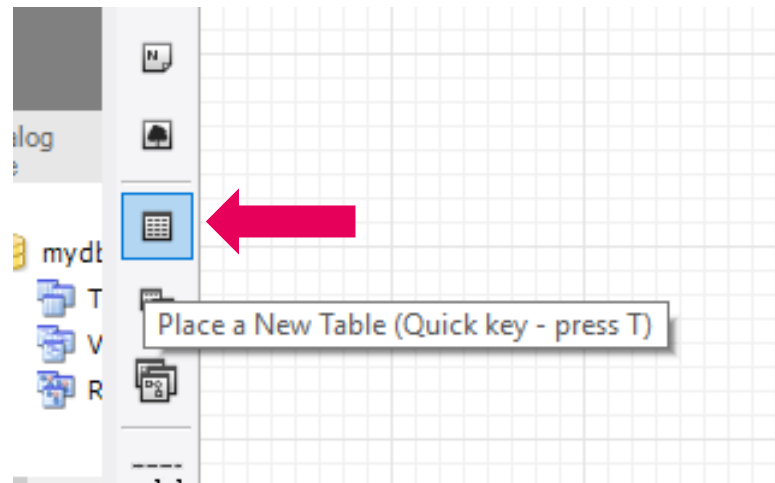
Modelagem Lógica – MySQL Workbench

- Esta outra aba será aberta:



Modelagem Lógica – MySQL Workbench

- Posicionando uma tabela no diagrama: clique no ícone ilustrado à esquerda e depois no diagrama



dê um duplo clique

- Aparecerá uma janela na parte inferior da tela, para editar as informações da tabela:

table1 - Table

Table Name: Schema: **mydb**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:

Charset/Collation:

Comments:

Data Type:

Default:

Storage: ☐ Virtual ☐ Stored

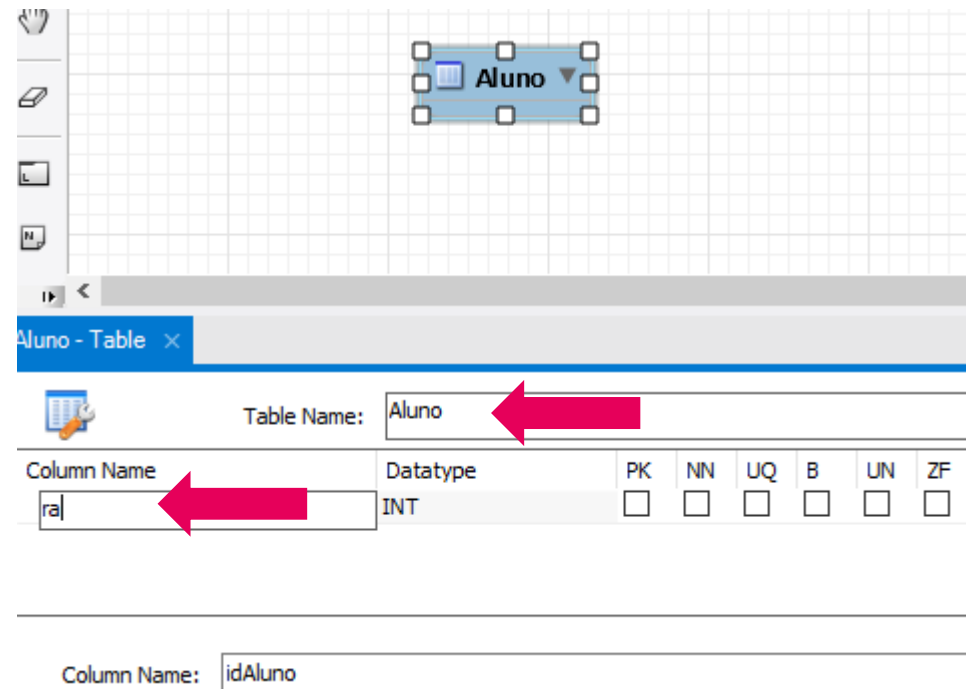
☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Modelagem Lógica – MySQL Workbench

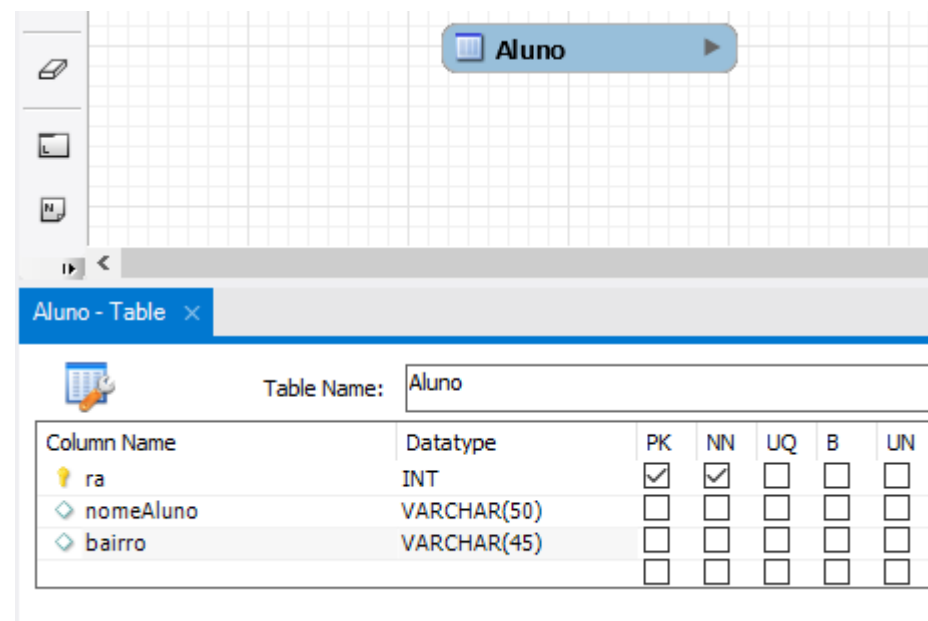
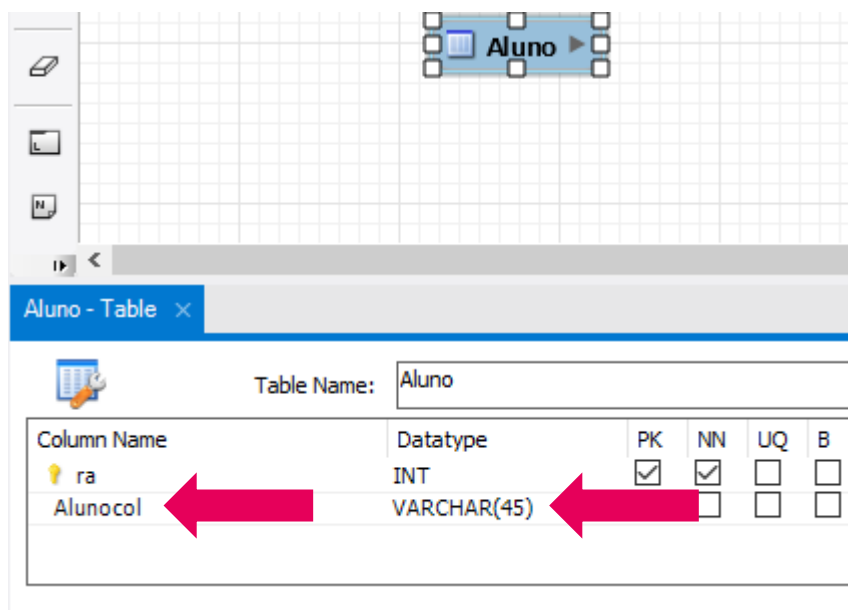
- Nessa janela que aparece na parte inferior da tela, configure o nome da tabela em Table Name e tecle Enter.



- Posicione o cursor do mouse abaixo de Column Name e dê um duplo clique
- O Workbench criará um campo chamado idAluno, altere o nome para ra e tecle Enter

Modelagem Lógica – MySQL Workbench

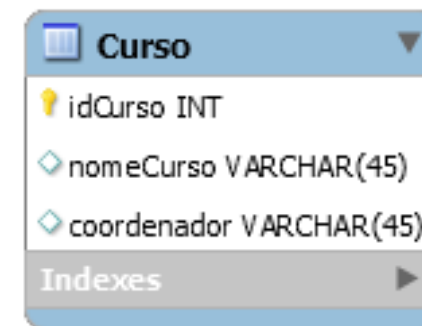
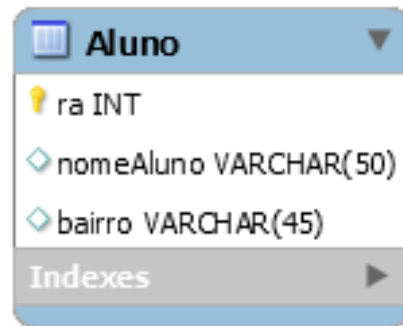
- Dê um duplo clique abaixo da coluna ra e o Workbench criará um campo chamado Alunocol, altere para nomeAluno e altere o tipo para VARCHAR(50)



- Crie outra coluna chamada bairro
- Depois feche essa janela inferior

Modelagem Lógica – MySQL Workbench

- Crie também uma tabela chamada Curso, com os campos idCurso (PK), nomeCurso e coordenador, como ilustrado abaixo:

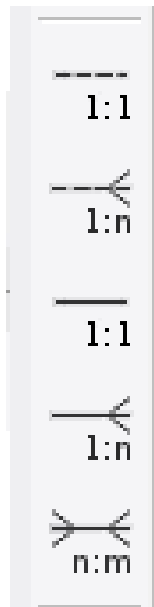


- Agora, falta colocar o relacionamento entre as 2 tabelas:
 - Verificamos a cardinalidade do relacionamento
 - Qual é a cardinalidade máxima de cada lado do relacionamento?
 - UM Aluno está matriculado em no máximo **UM Curso**
 - Um Curso tem no máximo **MUITOS Alunos** matriculados
 - Então é um relacionamento 1:N
 - As 2 entidades são fortes?
 - SIM

Modelagem Lógica – MySQL Workbench

- Para configurar o relacionamento, você deve escolher um dos itens abaixo:

Sendo que:

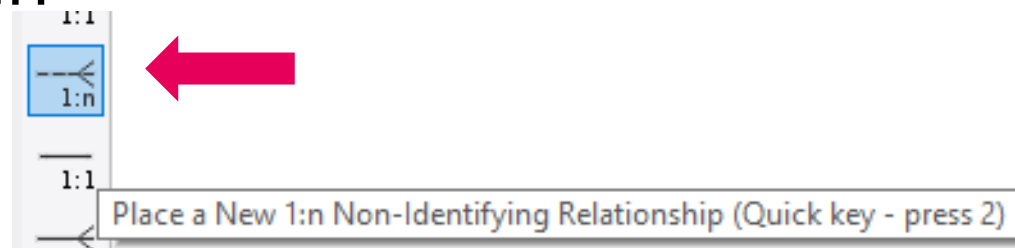


- a **linha tracejada** é quando as 2 entidades são fortes
- a **linha cheia** é quando uma entidade é forte e a outra é fraca
- **1:1** é quando a cardinalidade máxima é 1 dos dois lados
- **1:n** é quando a cardinalidade máxima é 1 de um lado e N do outro
- **n:m** é quando a cardinalidade máxima é N dos dois lados

Modelagem Lógica – MySQL Workbench

- No caso das entidades Aluno e Curso, as 2 são entidades fortes e a cardinalidade máxima é 1:n

- Então clicamos no ícone



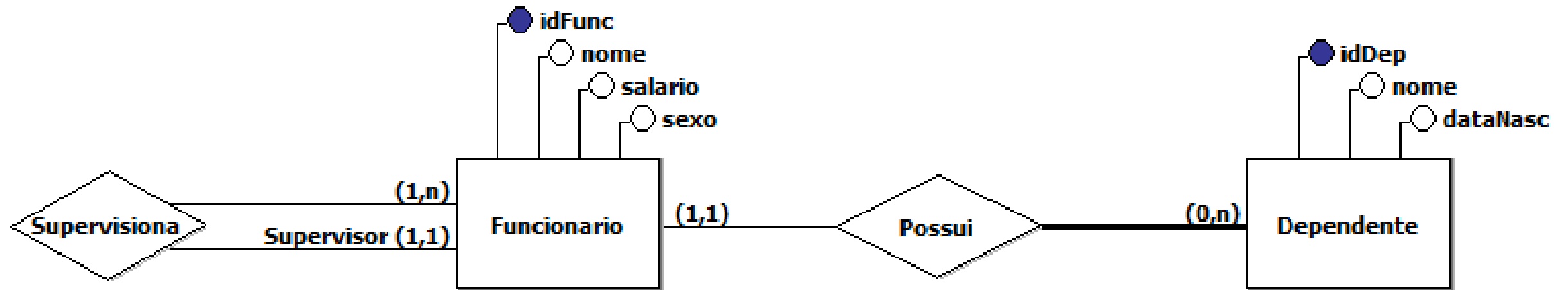
- Depois clicamos na entidade que é N (muitos) : Aluno
- E, por último, clicamos na entidade que é 1 : Curso



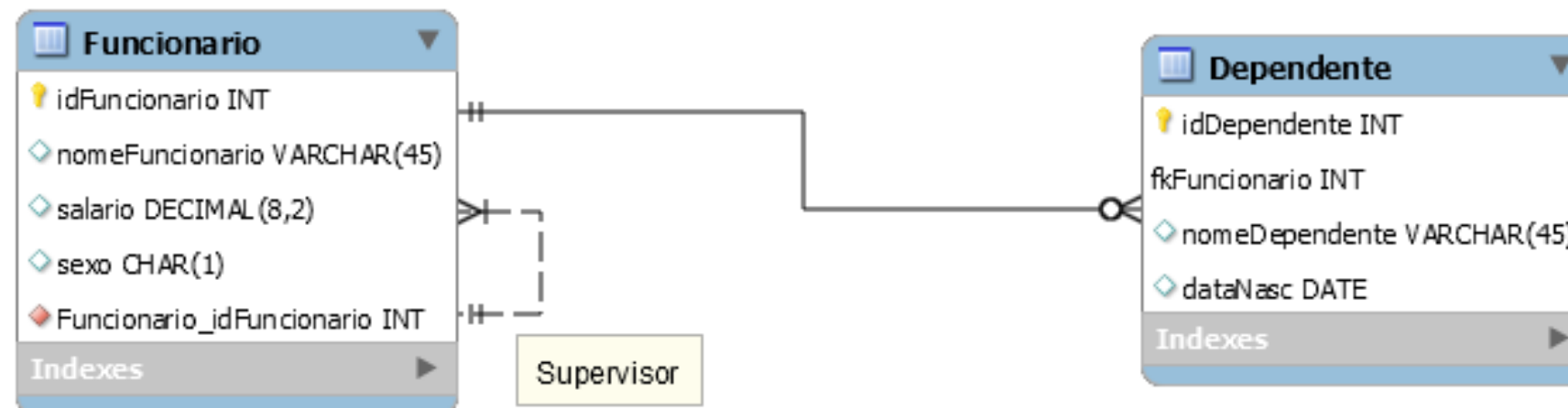
- Curso_idCurso é a chave estrangeira desse relacionamento
- Se preferir, pode mudar o nome desse campo para fkCurso

Modelagem Lógica – MySQL Workbench

- Da mesma forma, podemos elaborar o modelo lógico correspondente a este DER:

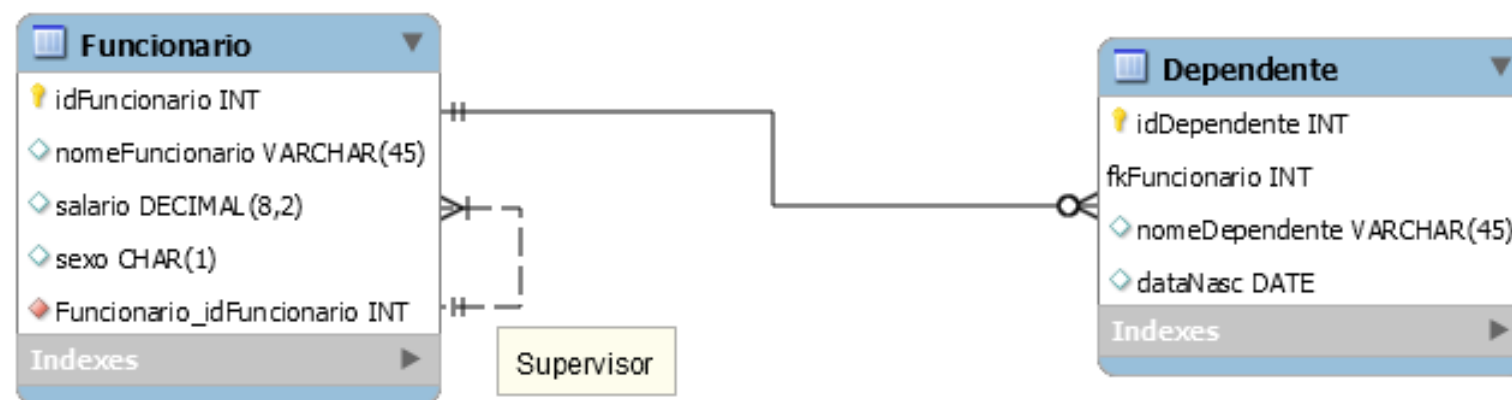


- Ficaria assim:



Modelagem Lógica – MySQL Workbench

- Repare que agora o relacionamento entre Funcionario e Dependente é um relacionamento entre uma entidade forte e uma fraca
- Por isso, a linha é cheia



- A tabela Dependente tem uma chave primária composta pelo idDependente e pelo fkFuncionario (informação que vem da tabela Funcionario)

Dependente - Table

Table Name: Dependente

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
idDependente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
fkFuncionario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nomeDependente	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dataNasc	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Implementação da Entidade Fraca no SQL

- Na Aula08, do Moodle, temos o Script-Funcionario, que implementa o modelo do Funcionario e do Dependente
- Create table da tabela Funcionario (valor de idFunc começará de 1000):

```

) create table Funcionario (
    idFunc int primary key auto_increment,
    nome varchar(40),
    sexo char(1),
    check (sexo = 'm' or sexo = 'f' or sexo = 'n'),
    -- no MySQL, versões anteriores a 8.0.16, devem usar enum no lugar do check:
    -- sexo enum('m','f','n'),
    salario decimal(7,2), -- 7 significa que o número tem 7 dígitos, considerando os dígitos antes da
                        -- vírgula e os depois
                        -- 2 significa que tem 2 dígitos após a vírgula
                        -- decimal(7,2) significa um número do tipo 99999,99
    check (salario > 0), -- isso não funciona em versões anteriores a 8.0.16 do MySQL
                        -- se fosse nota: check (nota >= 0 and nota <= 10)
    fkSupervisor int, -- implementação de um relacionamento recursivo um para muitos
    foreign key(fkSupervisor) references Funcionario(idFunc)
) auto_increment = 1000;

```


Implementação da Entidade Fraca no SQL

- Inserção de registros na tabela Funcionario

```
-- inserção de um funcionário, que não tem supervisor
insert into Funcionario values
    (null, 'Maria Aparecida','f',15000,null);
-- inserção de mais um funcionário (supervisor deste funcionário será a Maria Aparecida - id 1000)
insert into Funcionario values
    (null, 'José da Silva','m',5000,1000);
-- inserção de mais um funcionário (supervisor deste funcionário será o José da Silva - id 1001)
insert into Funcionario values
    (null, 'João Oliveira','m',2000,1001);
```

- Exibição da tabela Funcionario

19 • `select * from Funcionario;`

< Result Grid Filter Rows: Edit:

	idFunc	nome	sexo	salario	fkSupervisor
▶	1000	Maria Aparecida	f	15000.00	NULL
	1001	José da Silva	m	5000.00	1000
	1002	João Oliveira	m	2000.00	1001
*	NULL	NULL	NULL	NULL	NULL

Implementação da Entidade Fraca no SQL

- Criação da tabela Dependente (entidade fraca)

```

1      -- Criação da tabela Dependente, que é uma entidade fraca em relação a Funcionario
2
3  • ○ create table Dependente (
4      fkFunc int,
5      foreign key (fkFunc) references Funcionario(idFunc),
6      idDep int,
7      primary key (fkFunc, idDep), -- chave primária composta
8      nome varchar(40),
9      dataNasc date
10  );

```

- Inserção de registros na tabela Dependente:

```

-- Inserção de dependentes
insert into Dependente values
    (1000, 1, 'Ana', '2012-10-01'),
    (1000, 2, 'Paulo', '2014-03-05'),
    (1001, 1, 'Antônio', '2013-09-06'),
    (1002, 1, 'Clara', '2014-06-01');

```

- Observe que o idDep começa com o valor 1 para cada fkFunc diferente

Função SUM – Retorna a soma

- Quando uma coluna tem valores numéricos, podemos obter a soma dos valores da coluna
- Ex: podemos obter a soma da coluna salario da tabela Funcionario

```
72 • select sum(salario) from Funcionario;
```

```
73
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	sum(salario)			
	22000.00			

- Costumamos usar o **as** para nomear a coluna resultante do comando:

```
73 • select sum(salario) as Total_Salario from Funcionario;
```

```
74
```


Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Total_Salario			
	22000.00			

Função AVG – Retorna a média (average)

- Da mesma forma, é possível obter a média dos valores da coluna

```
75 • select avg(salario) from Funcionario;
```

```
76
```

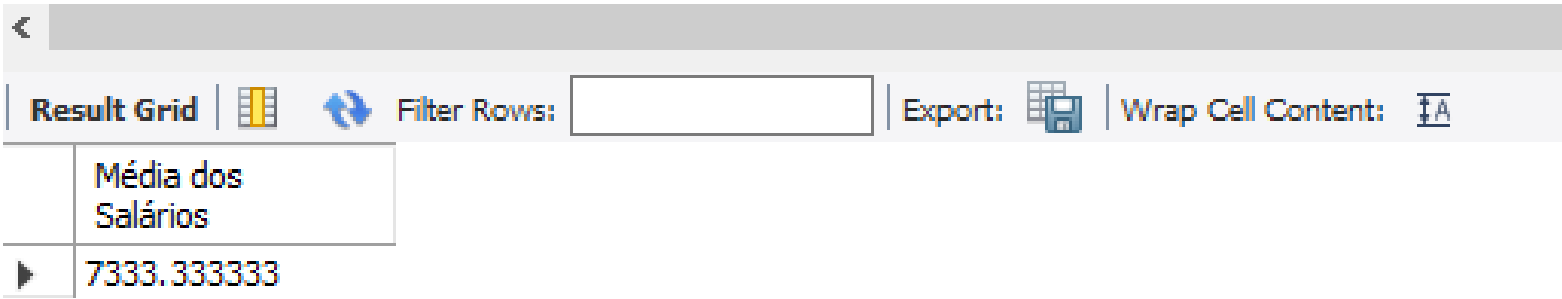


avg(salario)
7333.333333

- Repetindo o comando, mas nomeando a coluna resultante com o **as**:

```
76 • select avg(salario) as 'Média dos Salários' from Funcionario;
```

```
77
```






Média dos Salários
7333.333333

- Quando o nome após o **as** contém espaços, use aspas simples, como **'Média dos Salários'**

Função ROUND – Arredonda um valor

- Quando o valor retornado tem várias casas decimais, como a média do slide anterior (7333.333333), podemos usar a função **round** para arredondar o valor para a quantidade de casas decimais desejadas:
- Para arredondar para 2 casas decimais:

```
77 • select round(avg(salario),2) as 'Média dos Salários' from Funcionario;
78
```

<	
Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 	
	Média dos Salários
▶	7333.33

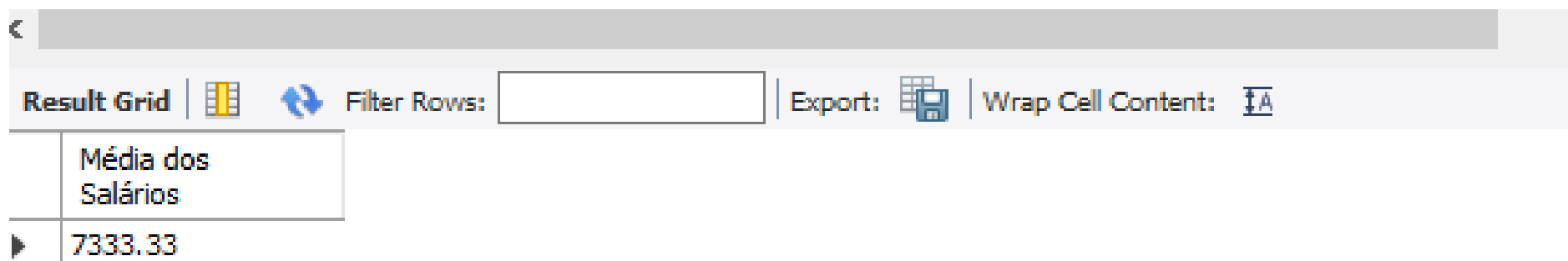
Função TRUNCATE – Trunca um valor

- Quando o valor retornado tem várias casas decimais, como a média 7333.333333, podemos usar também a função **truncate** para truncar o valor para a quantidade de casas decimais desejadas:

- Para truncar para 2 casas decimais:

```
78 • select truncate(avg(salario),2) as 'Média dos Salários' from Funcionario;
```

79



The screenshot shows a database query result in a 'Result Grid' view. The query is: `select truncate(avg(salario),2) as 'Média dos Salários' from Funcionario;`. The result is a single row with the column name 'Média dos Salários' and the value '7333.33'. The interface includes a toolbar with options like 'Filter Rows', 'Export', and 'Wrap Cell Content'.

Média dos Salários
7333.33




- Nesse caso, o round e o truncate produziram o mesmo resultado
- Se o número fosse 7333.338888:
 - Round para 2 casas decimais produziria 7333.34
 - Truncate para 2 casas decimais produziria 7333.33

Função MAX e MIN – Retornam maior e menor valor

- Da mesma forma, é possível obter o maior e o menor valor de uma coluna:

```
79 • select max(salario) as 'Maior Salário', min(salario) as 'Menor Salário' from Funcionario;
80
```

<

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

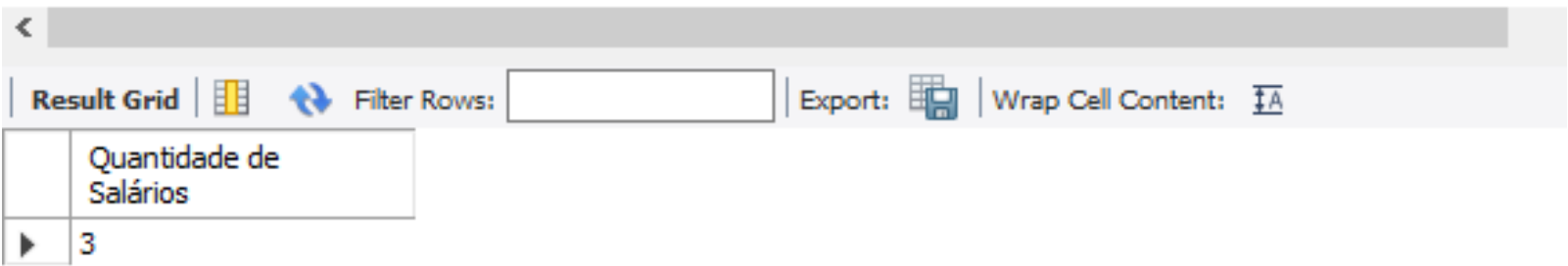
	Maior Salário	Menor Salário
▶	15000.00	2000.00

- Obs.: No exemplo acima, solicitei os 2 valores num único comando, mas é possível também fazer separadamente.

Função COUNT – Retorna a quantidade

- A função **count** é utilizada para saber a quantidade de valores:

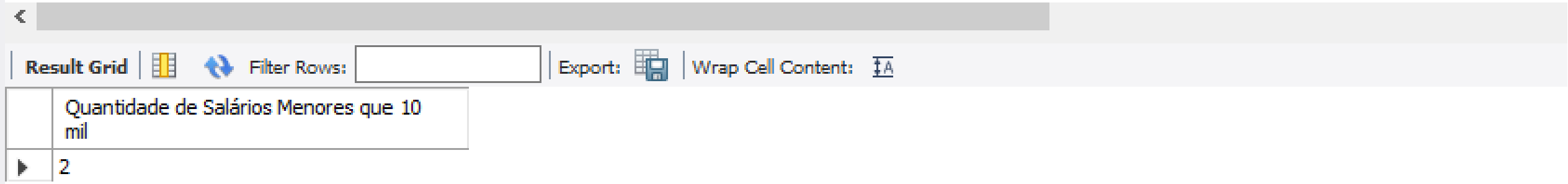
```
80 • select count(salario) as 'Quantidade de Salários' from Funcionario;
81
```



	Quantidade de Salários
▶	3

- No exemplo a seguir, usamos **count** para saber quantos salários são menores do que 10 mil:

```
81 • select count(salario) as 'Quantidade de Salários Menores que 10 mil' from Funcionario where salario < 10000;
```



	Quantidade de Salários Menores que 10 mil
▶	2

Soma e média somente de alguns salários

- Também é possível somar e obter a média de apenas alguns salários da coluna, como por exemplo, os salários menores que 10000.

```
82 • select sum(salario) as 'Total Salários Abaixo 10k',  
83      avg(salario) as 'Média Salários Abaixo 10k' from Funcionario where salario < 10000;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Total Salários Abaixo 10k	Média Salários Abaixo 10k
▶	7000.00	3500.000000

Obrigada!

BandTec
DIGITAL SCHOOL

Em caso de dúvidas, entre em contato com:
celia.taniwaki@bandtec.com.br