



Banco de Dados

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

1

2. Modelagem de dados



Nas aulas anteriores...

2

2. Modelagem de dados

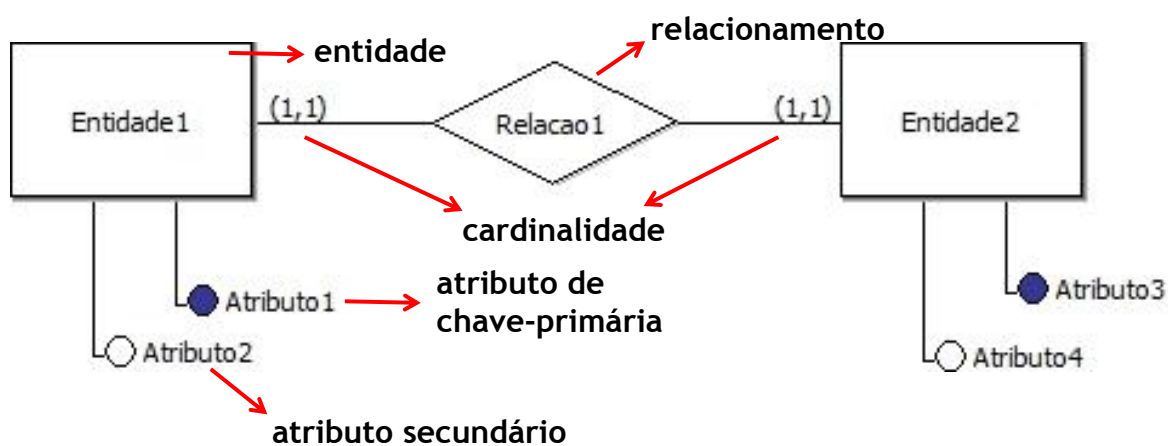
Três tipos:

- a. Modelagem Conceitual
- b. Modelagem Lógica
- c. Modelagem Física

3

2. Modelagem de dados

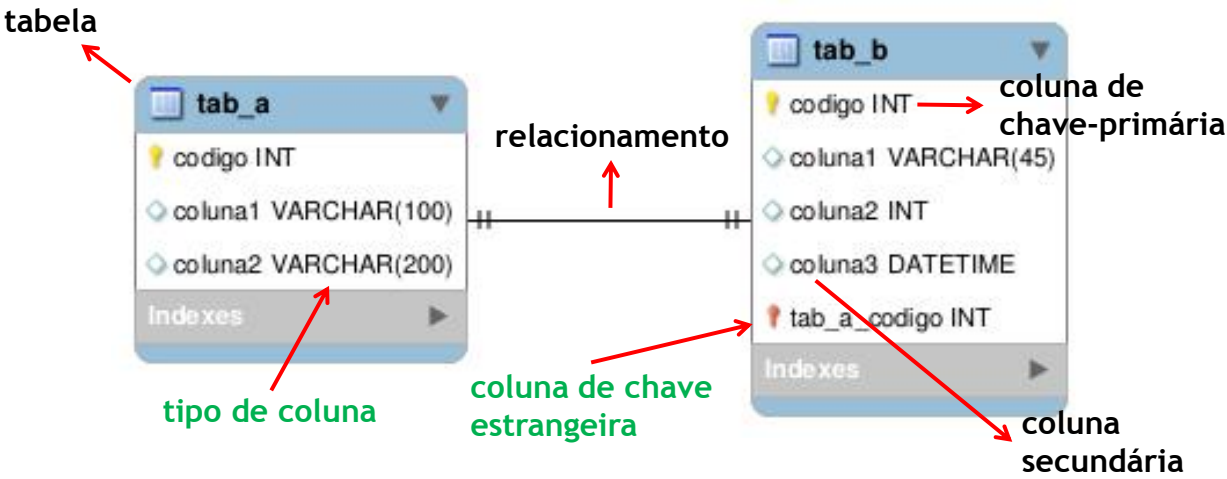
a. Modelagem Conceitual: Modelo Entidade Relacionamento (MER)



4

2. Modelagem de dados

b. Modelagem Lógica: Diagrama Entidade Relacionamento (DER)



5

2. Modelagem de dados

Resumo dos relacionamentos no modelo lógico:

cardinalidade	diagrama
1:1	
1:m	
m:n	

Ex. de relacionamento opcional (fraco):



Gera uma
terceira
tabela

6

2. Modelagem de dados



c. Modelagem Física:

- A SQL está dividida em:
 - DDL (*Data Definition Language*)
 - CREATE, ALTER, DROP.
 - DML (*Data Manipulation Language*)
 - SELECT, INSERT, UPDATE, DELETE.
 - DCL (*Data Control Language*)
 - GRANT, REVOKE (*vamos ver ainda*).

7

2. Modelagem de dados



*Fique agora com o conteúdo
da aula de hoje...*

8



Banco de Dados

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

9

3. SQL - DCL



- DCL: *Data Control Language*;
- Comandos que servem para controlar o acesso dos usuários que usam diretamente o banco de dados;
- Comandos:
 - **GRANT**: utilizado para conceder privilégios específicos a determinados usuários;
 - **REVOKE**: utilizado para retirar privilégios específicos de determinado usuário.

10

3. SQL - DCL



- Exemplos:

- Tarefa 1: Conceder permissão de **SELECT** na tabela “**tab_cliente**” para o grupo de usuários do BD “**estagiarios**”:
 - **GRANT SELECT ON tab_cliente TO estagiarios;**
- Tarefa 2: Retirar permissões de **INSERT** e de **DELETE** na tabela “**tab_endereco**” do usuário “**alansouza**”:
 - **REVOKE INSERT, DELETE ON tab_endereco FROM alansouza;**

11



Banco de Dados

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

12

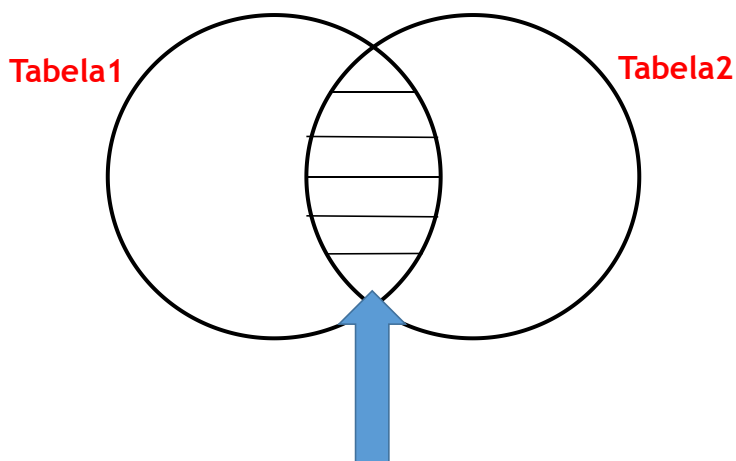
4. SQL Join

- Join: junção;
- Permite extrair dados de várias tabelas distintas a partir de um único select;
- Principais tipos:
 - inner join
 - left join
 - right join

13

4. SQL Join

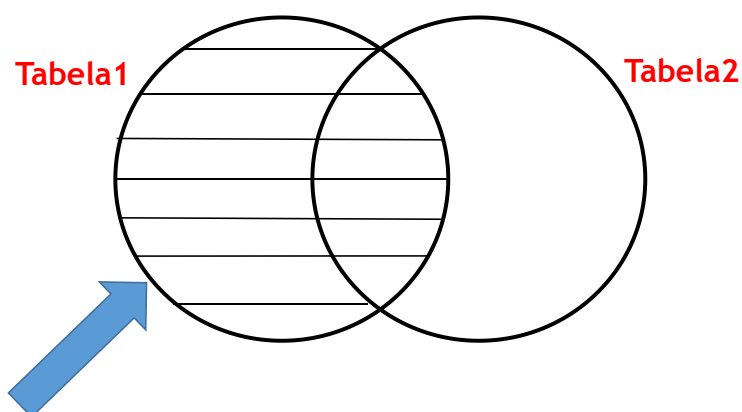
- **inner join**: pega a interseção, ou seja, o que há de comum entre tabelas:



14

4. SQL Join

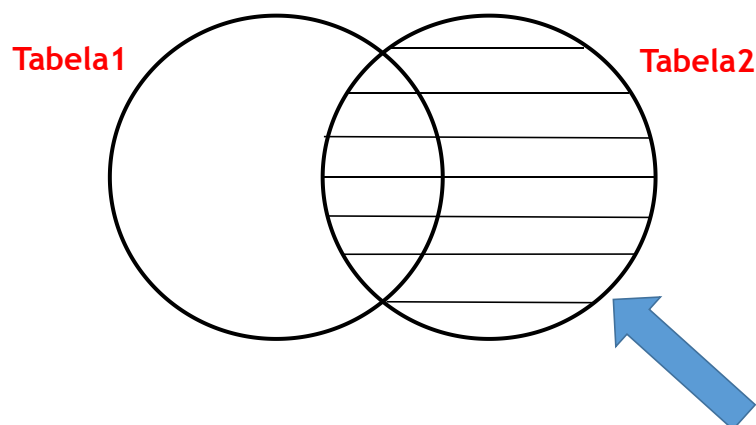
- **left join**: pega a interseção e os dados da tabela da esquerda:



15

4. SQL Join

- **right join**: pega a interseção e os dados da tabela da direita:



16

4. SQL Join

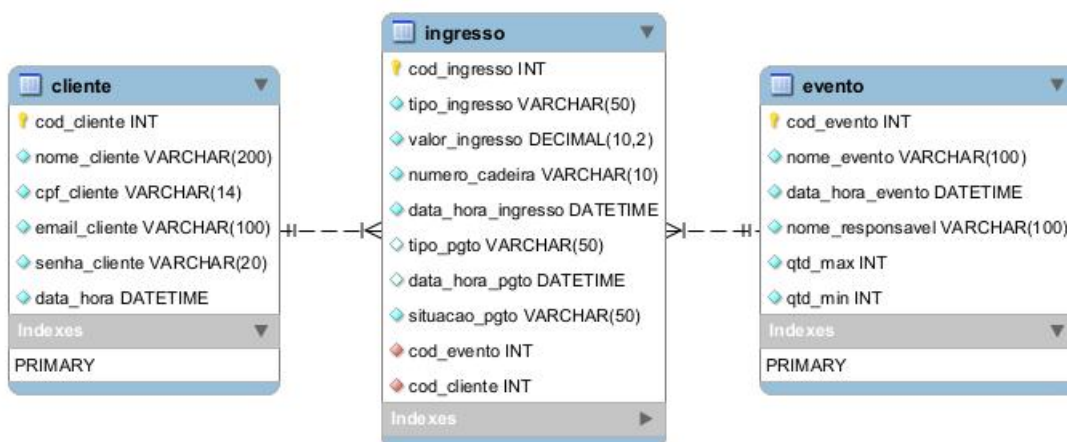
Afinal, quem é a tabela da esquerda e quem é a tabela da direita?

Vamos ver um exemplo para entender melhor...

17

4. SQL Join

Exemplo: considere o modelo lógico do banco de dados do teatro:



18

4. SQL Join



Exemplo: considere os dados em cada tabela (RESUMIDO):

cliente		
cod_cli	nome	email
1	Maria	maria@email.com
2	João	joao@email.com
3	Paulo	paulo@email.com

evento		
cod_eve	nome	data_hora_eve
1	Orquestra	2020-01-03 8:00
2	Standup	2020-01-07 18:00
3	Musical	2020-02-11 16:00

ingresso				
cod_ing	tipo	valor	cod_cli	cod_eve
1	Inteira	80.00	1	1
2	Meia	40.00	2	1
3	Inteira	80.00	NULL	1
4	Meia	50.00	NULL	2

19

4. SQL Join



Problema 1: Como podemos pegar as informações de clientes e dos respectivos ingressos adquiridos por eles através de um único select?

Consulta:

Tabela da esquerda

Tabela da direita

```
select * from cliente inner join ingresso on cliente.cod_cliente = ingresso.cod_cliente
```

Resultado:

cod_cli	nome	email	cod_ing	tipo	valor	cod_cli	cod_eve
1	Maria	maria@email.com	1	Inteira	80.00	1	1
2	João	joao@email.com	2	Meia	40.00	2	1

20

4. SQL Join



Problema 2: Como podemos pegar as informações de clientes e de ingressos independente se o ingresso foi adquirido ou não pelo cliente?

Solução:



```
select * from cliente left join ingresso on cliente.cod_cliente = ingresso.cod_cliente
```

Resultado:

cod_cli	nome	email	cod_ing	tipo	valor	cod_cli	cod_eve
1	Maria	maria@email.com	1	Inteira	80.00	1	1
2	João	joao@email.com	2	Meia	40.00	2	1
3	Paulo	paulo@email.com	NULL	NULL	NULL	NULL	NULL

21

4. SQL Join



Problema 3: Como podemos pegar as informações de ingressos independente se eles estão vinculados a um cliente ou não?

Solução:



```
select * from cliente right join ingresso on cliente.cod_cliente = ingresso.cod_cliente
```

Resultado:

cod_cli	nome	email	cod_ing	tipo	valor	cod_cli	cod_eve
1	Maria	maria@email.com	1	Inteira	80.00	1	1
2	João	joao@email.com	2	Meia	40.00	2	1
NULL	NULL	NULL	3	Inteira	80.00	NULL	1
NULL	NULL	NULL	4	Meia	50.00	NULL	2

22

4. SQL Join



Problema 4: Como podemos extrair apenas as seguintes informações de todo mundo que adquiriu ingresso: nome e email do cliente; tipo e valor do ingresso?

Solução:

```
select cliente.nome_cliente, cliente.email_cliente,
ingresso.tipo_ingresso, ingresso.valor_ingresso
from cliente
inner join ingresso on cliente.cod_cliente =
ingresso.cod_cliente;
```

Especifica-se
cada coluna de
cada tabela
(operador ponto)

23

4. SQL Join



Problema 4: Como podemos extrair apenas as seguintes informações de todo mundo que adquiriu ingresso: nome e email do cliente; tipo e valor do ingresso?

Resultado:

nome	email	tipo	valor
Maria	maria@email.com	Inteira	80.00
João	joao@email.com	Meia	40.00

24

4. SQL Join



Problema 5: Como podemos selecionar o nome do evento que aconteceu no dia 03/01/2020 às 8:00 e o código de todos os ingressos vinculados ao respectivo evento?

Solução:

```
select evento.nome, ingresso.cod_ing
from evento
inner join ingresso on evento.cod_evento = ingresso.cod_evento
where evento.data_hora_evento = '2020-01-03 8:00'
```

4. SQL Join



Problema 5: Como podemos selecionar o nome do evento que vai acontecer dia 11/02/2020 às 16:00 e o código de todos os ingressos vinculados ao respectivo evento?

Resultado:

nome	cod_ing
Orquestra	1
Orquestra	2
Orquestra	3

4. SQL Join



Exercícios:

- 1) Crie um select para pegar todos os dados de ingressos e de eventos independente se o ingresso está vinculado a um evento.
- 2) Faça um select para extrair o email de todos os clientes que possuem ingresso pago através de boleto.
- 3) Desenvolva um select para pegar o nome dos clientes, o valor do respectivo ingresso e o nome do respectivo evento que ele irá participar.