SECTI

Introdução à Banco de Dados

Relacionamento de Tabelas

Prof. Ulpio Netto.

OXETECH

Relacionamento

- Em um SGBD relacional, o relacionamento de tabelas é uma parte fundamental do design, permitindo associação de dados entre diferentes entidades.
- Em geral, existem três tipos principais de de relacionamentos:
  - (1:1) – Onde cada registro em uma tabela está associado a apenas um registro em outra tabela.
  - (1:N) – Onde cada registro de uma tabela pode estar relacionado com vários registros de outra tabela.
  - (N:N) – Onde cada registro em uma tabela pode estar associado a vários registros em outra tabela e vice-versa. Geralmente usando uma tabela de junção que conecta as tabelas principais.

OXETECH

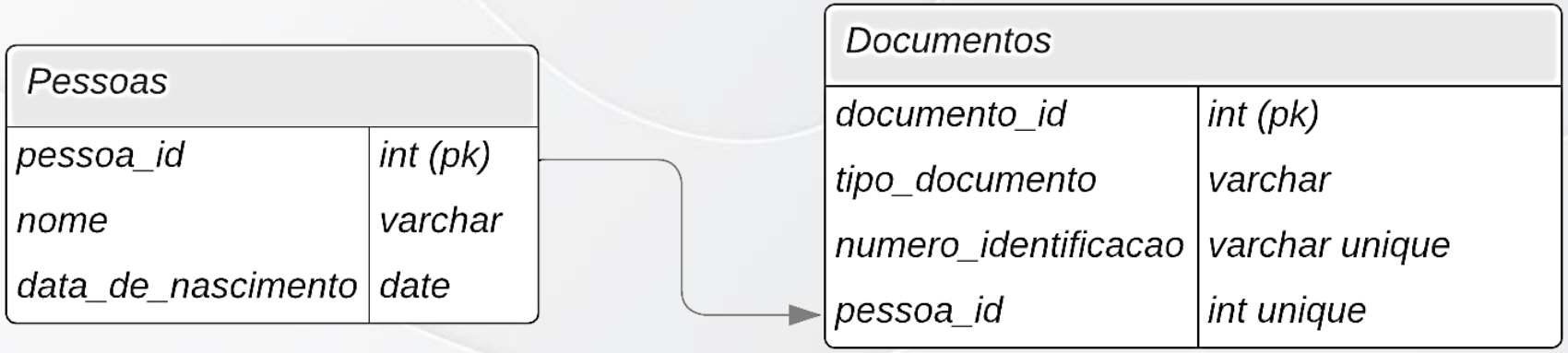
Relacionamento (1:1)

- Cada registro em uma tabela está associado a, no máximo, um único registro em outra tabela.
- Esse tipo de relacionamento é usado quando duas entidades estão tão fortemente relacionadas, que faz sentido agrupá-las em uma única entidade.
- Benefícios do relacionamento (1:1):
  - Normalização dos dados, reduzindo redundância de informações;
  - Organização Lógica, torna mais lógico e intuitivo;
  - Integridade Referencial, usa de chave estrangeiras para garantir integridade dos dados, evitando inconsistências;

OXETECH

Exemplo : Relacionamento (1:1)

- Exemplo: Considere que você tem uma tabela de **pessoas** e outra tabela de **documentos**. Cada pessoa pode ter no máximo um documento associado, como um número de identidade. A tabela de **documentos** contém informações específicas do documento.



Antes de relacionar...

- Rode o comando: **pragma foreign_key;**
- Caso responda: 1, não é necessário fazer nada.
- Agora caso responda: 0, Insira o comando pragma foreign_key = on;
- Confira usando novamente o comando da primeira linha.
- Deve retornar 1.

OXETECH

Exemplo: Relacionamento (1:1)

```
sqlite> create table pessoas(
(x1...> pessoa_id int primary key,
(x1...> nome varchar,
(x1...> data_de_nascimento date
(x1...> );
```

```
sqlite> create table documentos(
(x1...> documento_id int primary key,
(x1...> tipo_documento varchar(50),
(x1...> numero_identificacao varchar unique,
(x1...> pessoa_id int unique,
(x1...> foreign key (pessoa_id) references pessoas(pessoa_id)
(x1...> );
```

OXETECH

Estrutura de uma Foreign Key

- Para usar uma Foreign Key, deve-se seguir a estrutura:

```
sqlite> foreign key (coluna_dessa_tabela)
   ...> references nome_da_outra_tabela(pk_da_outra_tabela);
```

OXETECH

Inserindo dados nas tabelas.

- Primeiro vamos inserir na tabela de **pessoas**.

```
sqlite> insert into pessoas(pessoa_id,nome,data_de_nascimento)
   ...> values(1,'Ulpio','2002-01-22');
sqlite> select * from pessoas
   ...> ;
1|Ulpio|2002-01-22
```

- Para depois inserirmos na tabela de documentos, já que precisa de um pessoa_id para existir.

```
sqlite> insert into documentos(documento_id,tipo_documento,numero_identificacao,pessoa_id)
   ...> values (1,'CPF','033.123.123',1);
sqlite> select * from documentos;
1|CPF|033.123.123|1
```

OXETECH

Consultando...

- Agora, vamos consultar informações de pessoas e seus documentos.

```
sqlite> select pessoas.nome,documentos.numero_identificacao
   ...> from pessoas
   ...> join documentos on pessoas.pessoa_id = documentos.pessoa_id;
Ulpio|033.123.123
```

- Essa consulta retorna o nome da pessoa e o número de identificação do documento dela.
- Se tivermos mais pessoas, teremos mais linhas como resultado...

OXETECH

Exercicio 1

- Ainda usando o conceito de relacionamentos (1:1), faça inserções de mais 3 pessoas.
- Depois de realizara inserção dos dados na tabela pessoas, faça o mesmo na tabela de documentos, criando documentos para as pessoas que inserimos.

- Ao final, use a seleção do slide anterior para mostrar o nome de cada pessoa e o documento da mesma.

OXETECH

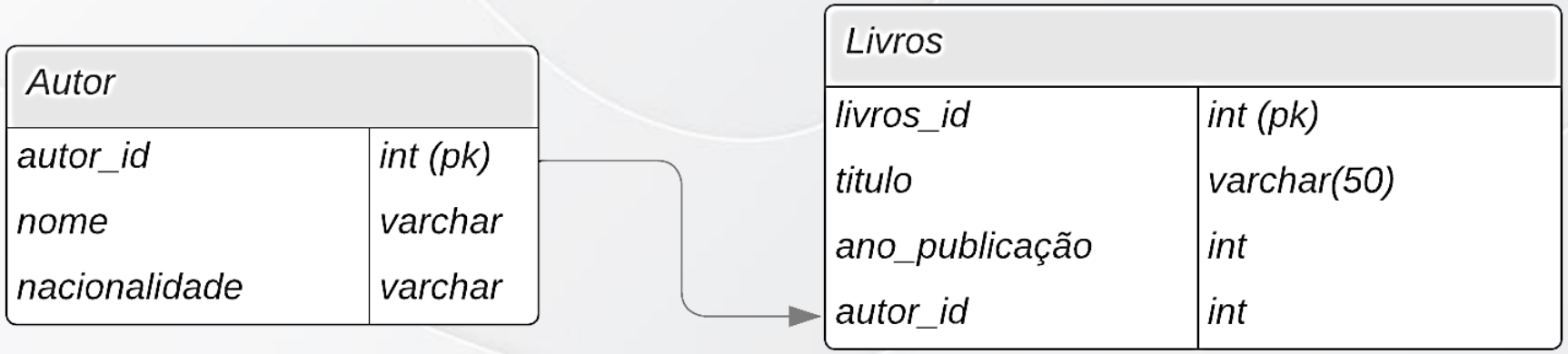
Relacionamento (1:N)

- Conhecido também como um pra muitos, indica que cada registro em uma tabela pode estar associado a vários registros de outra tabela.
- Esse tipo de relacionamento é comum e é utilizado quando um registro em uma tabela tem interação com vários outros registros de uma tabela.

OXETECH

Exemplo: Relacionamento (1:N)

- Considere um exemplo de um sistema de biblioteca, onde temos uma tabela **autores** e uma tabela **livro.** Cada autor pode ter escrito vários livros, porém, cada livro só pode ter sido escrito por um autor.



Exemplo: Relacionamento (1:N)

```
sqlite> create table autores(
(x1...> autor_id int primary key,
(x1...> nome varchar(50),
(x1...> nacionalidade varchar);
```

```
sqlite> create table livros(
(x1...> livro_id int primary key,
(x1...> titulo varchar(50),
(x1...> ano_publicacao int,
(x1...> autor_id int,
(x1...> foreign key (autor_id) references autores(autor_id)
(x1...> );
```

OXETECH

Inserindo dados nas tabelas.

- Primeiro vamos inserir na tabela de **autores**.

```
sqlite> insert into autores(autor_id,nome,nacionalidade)
   ...> values (1,'J.R.R Tolkien','África do Sul');
```

- Para depois inserirmos na tabela de Livros, já precisamos de autores para escrever esses livros.

```
sqlite> insert into livros(livro_id,titulo,ano_publicacao,autor_id)
   ...> values (1,'O Silmarilion',1977,1);
sqlite> insert into livros(livro_id,titulo,ano_publicacao,autor_id)
   ...> values (2,'O Senhor dos Anéis',1945,1);
```

OXETECH

Consultando...

- Consultando...

```
sqlite> select * from autores;
1|J.R.R Tolkien|Africa do Sul
```

```
sqlite> select * from livros;
1|O Silmarilion|1977|1
2|O Senhor dos Anéis|1945|1
```

- Listando os livros e seus respectivos autores

```
sqlite> select livros.titulo,autores.nome
   ...> from livros
   ...> join autores on livros.autor_id = autores.autor_id;
O Silmarilion|J.R.R Tolkien
O Senhor dos Anéis|J.R.R Tolkien
```

OXETECH

Exercicio 2

- Ainda usando o conceito de relacionamentos (1:N), faça a inserção de mais 2 autores
- Depois de inserir os autores, crie dois livros para cada um. Pesquise as informações
- Depois de realizar as inserções, faça a consulta como o slide anterior. Mostrando o nome dos livros e seus respectivos autores.

OXETECH

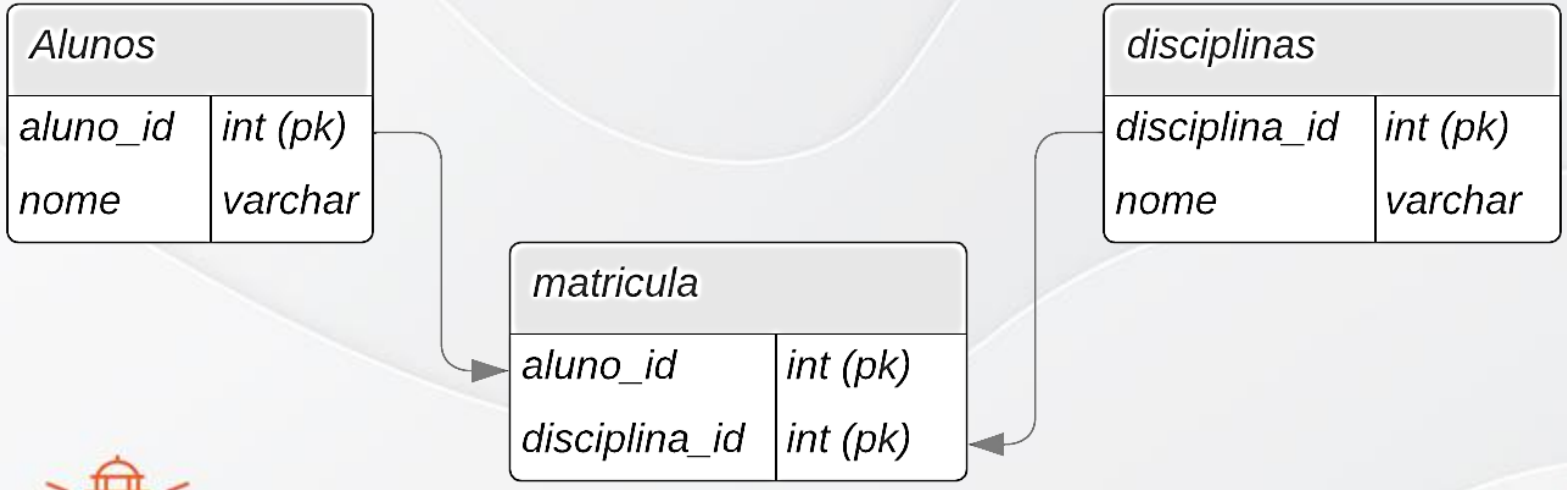
Relacionamento (N :N)

- Conhecido também como muito pra muitos. Indica que cada registro da tabela pode estar associado a vários registros e vice-versa.
- É comum quando há uma relação complexa entre duas entidades, e é geralmente resolvido através de uma tabela de junção(Tabela de relacionamento).

OXETECH

Exemplo: Relacionamento (N :N)

- Considere um exemplo de um sistema de universidade, onde temos uma tabela **alunos** e uma tabela **disciplinas**. Um aluno pode estar matriculado em várias disciplinas, e uma disciplina pode ter vários alunos matriculados.



Exemplo: Relacionamento (N:N)

```
sqlite> create table alunos(
(x1...> aluno_id int primary key,
(x1...> nome varchar
(x1...> );
```

```
sqlite> create table disciplinas(
(x1...> disciplina_id int primary key,
(x1...> nome varchar
(x1...> );
```

```
sqlite> create table matriclas (
(x1...> aluno_id int,
(x1...> disciplina_id int,
(x1...> primary key(aluno_id, disciplina_id),
(x1...> foreign key (aluno_id) references alunos(aluno_id),
(x1...> foreign key (disciplina_id) references disciplinas(disciplina_id)
(x1...> );
```

OXETECH

Inserindo dados nas tabelas.

- Inserindo alguns alunos:

```
sqlite> insert into alunos(aluno_id,nome)
   ...> values (302,'Ulpio'),
   ...> (305,'Gustavo'),
   ...> (309,'Juan')
   ...> ;
```

- Fazendo o mesmo com disciplinas

```
sqlite> insert into disciplinas(disciplina_id,nome)
   ...> values(3,'Banco de Dados'),
   ...> (9,'Java'),
   ...> (1,'Lógica de Programação');
```

- Relacionando as tabelas

```
sqlite> insert into matriclas(aluno_id,disciplina_id)
   ...> values(302,9),(302,1),(302,3);
sqlite> insert into matriclas(aluno_id,disciplina_id)
   ...> values (305,1),(305,3);
sqlite> insert into matriclas(aluno_id,disciplina_id)
   ...> values(309,9);
```

Consultando...

- Consultando as tabelas:

```
sqlite> select * from alunos;
302|Ulpio
305|Gustavo
309|Juan
```

```
sqlite> select * from disciplinas;
3|Banco de Dados
9|Java
1|Lógica de Programaçao
```

```
sqlite> select * from matriclas;
302|9
302|1
302|3
305|1
305|3
309|9
```

OXETECH

Consultando com Join

- Usando o Join para consultas os dados das tabelas originais, através da tabela de relação, para resgatar o nome do aluno e a disciplina que ele está matriculado

```
sqlite> select alunos.nome,disciplinas.nome
   ...> from alunos
   ...> join matriclas on alunos.aluno_id = matriclas.aluno_id
   ...> join disciplinas on matriclas.disciplina_id = disciplinas.disciplina_id;
Ulpio|Java
Ulpio|Lógica de Programaçao
Ulpio|Banco de Dados
Gustavo|Lógica de Programaçao
Gustavo|Banco de Dados
Juan|Java
```

OXETECH

Exercicio 3

- Adicione ainda mais alunos.
- Adiciona mais uma disciplinas
- Gere mais matriculas
- Consulte usando o Select do ultimo slide.

OXETECH

SECTI

OXETECH