Banco de Dados – C07

Felipe Tagawa Reis

Introdução a Banco de Dados Relacionais

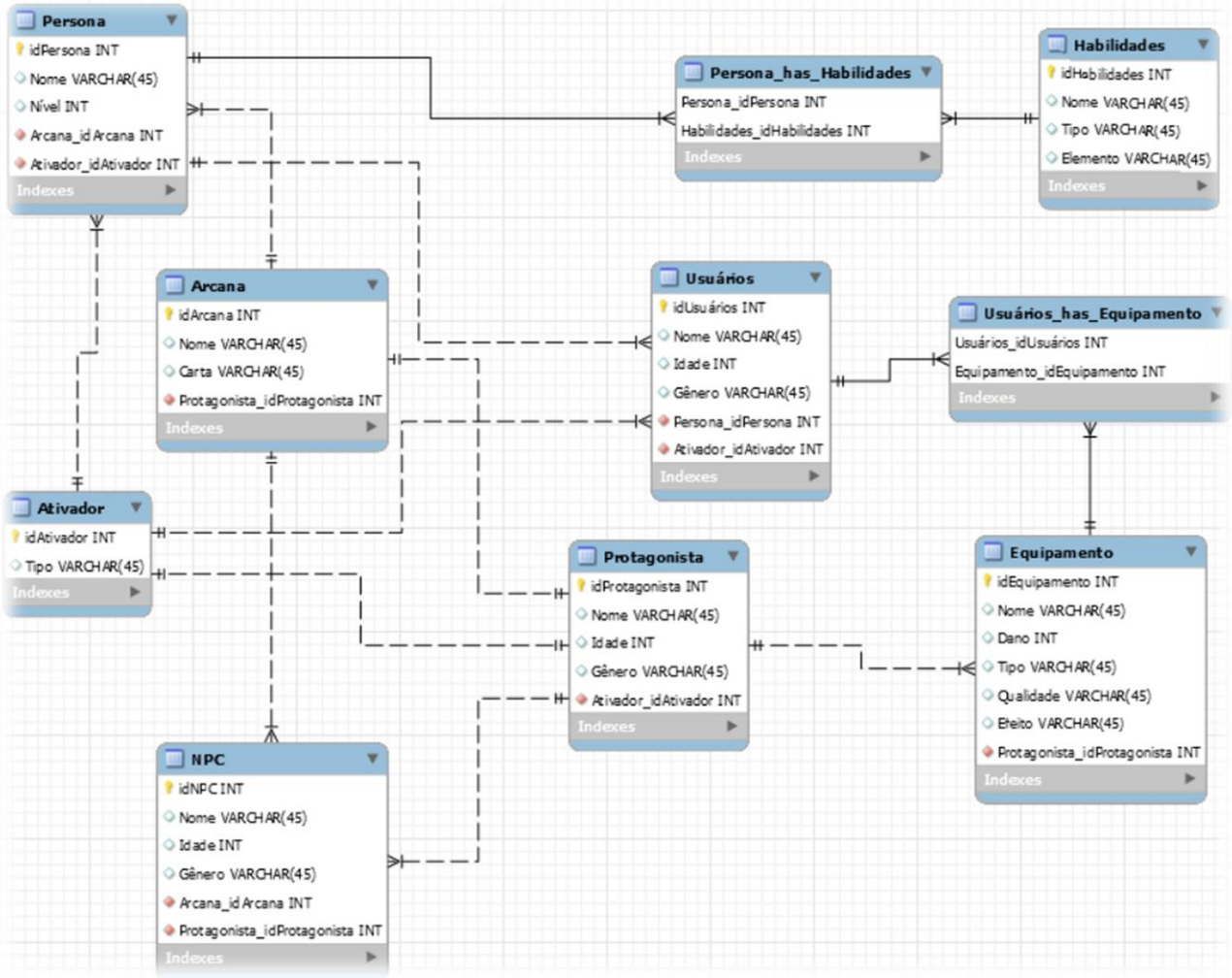
Inatel

Instalação do MySQL Workbench

- Vídeo com os detalhes da instalação no canal do Teams.
- Arquivo com todos os detalhes no TEAMS ou no meu repositório da disciplina no Github.
- IMPORTANTE: para facilitar entradas posteriores (principalmente no projeto), recomendo colocar o **nome de usuário e senha como “root”**.
- Vídeo e material de consulta para instalação do programa já em ‘Aula 1’ no Teams.

Bancos de Dados Relacionais

São sistemas de gerenciamento e armazenamento de dados em tabelas estruturadas com linhas e colunas, onde as informações são distribuídas e organizadas de forma lógica e inter-relacionada.



Inatel

Prós e Contras

Prós

- Garantia **ACID** – Segurança e Consistência;
- Sistema Robusto de consultas;
- Padronização – Estabilidade.

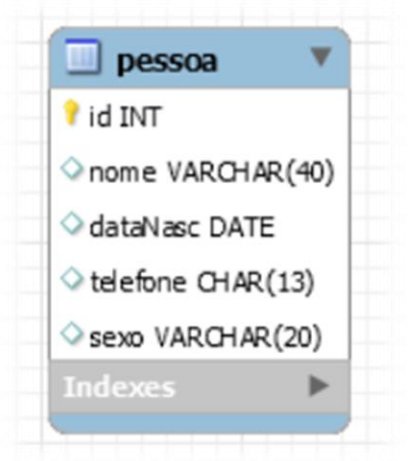
Contras

- Escala Vertical (Baixa Escalabilidade);
- Esquema Rígido (Pouca Flexibilidade);
- Menor Desempenho em cenários Big Data – Exemplo do Post do Instagram.

Transações ACID

- A (Atomicidade)
  - Atomicidade trata das transações serem indivisíveis;
- C (Consistência)
  - Integridade geral dos dados;
- I (Isolamento)
  - Impossibilidade de interferência entre transações;
- D (Durabilidade)
  - Disponibilidade dos Dados, mesmo em casos adversos.

Tabelas



Modelo do Workbench

id(PK)	nome	dataNasc	telefone	sexo

Tabela Simples

Inatel

Tipos de Dados mais Utilizados

Numéricos

- INTEGER(INT);
- DECIMAL(i,j);
- FLOAT(j);
- DOUBLE.

i: dígitos totais;
j: dígitos decimais.

Data e Hora

- DATE;
- TIME;

DATE (YYYY-MM-DD);
TIME (HH:MM:SS).

Lógico

- BOOLEAN.

String

- CHAR(n);
- VARCHAR(n);
- TEXT.

CHAR é FIXO em **n** caracteres!
VARCHAR aceita **até n** caracteres!
TEXT não tem quantidade fixa, é bastante usado para textos e descrições **extensos**.

Inatel

String

- CHAR(n);
- VARCHAR(n);
- TEXT.

CHAR x VARCHAR

CHAR

VARCHAR

Inatel

Null X Zero

Null

- É a **ausência** de valor, indicando que o valor é desconhecido.

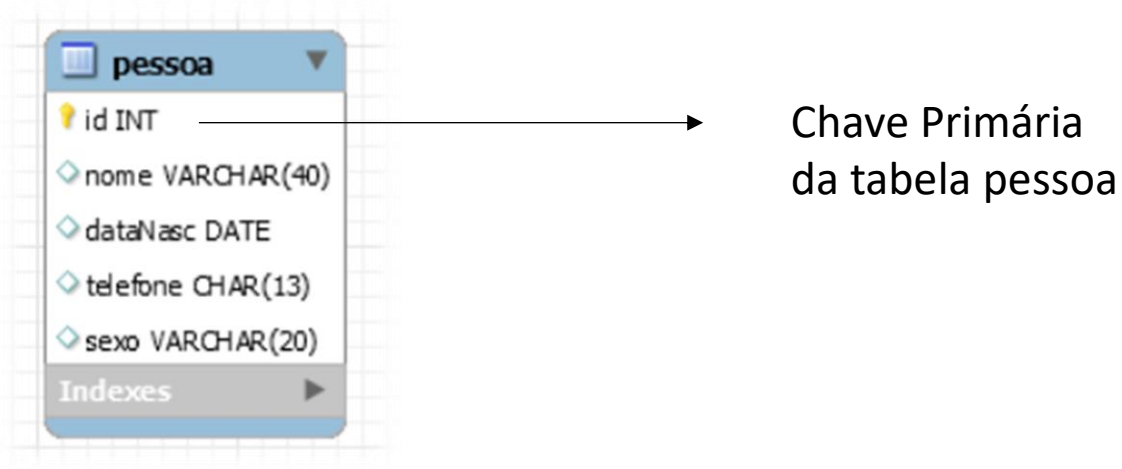
Zero

- É um valor numérico **definido**, indicando que o valor foi preenchido.

Inatel

Chaves Primárias
(Primary Key/PK)

- É um ou mais campos (atributos) que identificam **unicamente** cada registro em uma tabela;
- **NÃO pode conter valores nulos (NULL) nem valores duplicados;**
- **Cada tabela pode ter apenas uma chave primária**, mas essa chave pode envolver **mais de um campo** — nesse caso, é chamada de **chave primária composta;**
- Em relacionamentos entre tabelas, uma chave primária pode ser combinada com a chave estrangeira de outra tabela para compor uma chave — esse conceito será abordado posteriormente.



Inatel

Chaves estrangeiras

```sql
-- Chave Estrangeira (FOREIGN KEY)
CREATE TABLE Filme (
    id INT AUTO_INCREMENT,
    titulo VARCHAR(255) NOT NULL,
    ano_lancamento INT NOT NULL,
    personagem_id INT,
    PRIMARY KEY (id),
    CONSTRAINT fkPersonagem
    FOREIGN KEY (personagem_id)
    REFERENCES Personagem(id)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);
```

- CONSTRAINT - Bloco de comandos para configurar a chave estrangeira. Para cada chave estrangeira da tabela um bloco CONSTRAINT deve ser feito
- ON UPDATE e ON DELETE - Definem o que fazer estrangeira. ao se excluir um registro em outra tabela relacionado pela chave estrangeira
- CASCADE - A alteração na tabela inicial se estende ao registro da tabela com a chave estrangeira
- FOREIGN KEY - Comando que mostra qual coluna é uma chave estrangeira
- REFERENCES - Referencia a qual tabela a chave estrangeira está relacionada

Chave Estrangeira

```sql
CREATE TABLE Personagem (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    ator VARCHAR(255) NOT NULL,
    cidade VARCHAR(255) NOT NULL,
    company_publishing VARCHAR(255) NOT NULL,
);
```

```sql
-- Chave Estrangeira (FOREIGN KEY)
CREATE TABLE Filme (
    id INT AUTO_INCREMENT,
    titulo VARCHAR(255) NOT NULL,
    ano_lancamento INT NOT NULL,
    personagem_id INT,
    PRIMARY KEY (id),
    CONSTRAINT fkPersonagem
    FOREIGN KEY (personagem_id)
    REFERENCES Personagem(id)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);
```

```sql
-- Inserindo um filme relacionado a um personagem existente - Batman (id = 1)
INSERT INTO Filme (titulo, ano_lancamento, personagem_id) VALUES ('The Batman', 2022, 1);
```

Chave Estrangeira (FK)

Chave Primária

Entidade

CONTRATO DE ALUGUEL

Entre _____, RG nº _____, CPF nº _____, residente em _____, LOCADOR, e _____, RG nº _____, CPF nº _____, residente em _____, LOCATÁRIO, fica acordado a locação temporária do imóvel situado em _____ por um período de _____ dias, iniciando em __/__/____ e terminando em __/__/____.

O aluguel acordado será de R$ _____, pago integralmente antes da entrada no imóvel. O LOCATÁRIO compromete-se a cuidar do imóvel e restituí-lo nas mesmas condições em que recebeu.

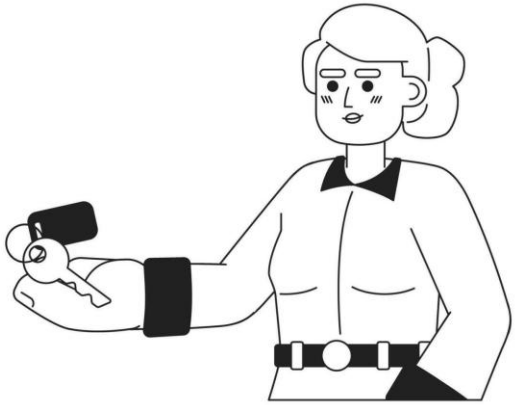
_____, __/__/____

_____
(Assinatura do LOCADOR)

_____
(Assinatura do LOCATÁRIO)

Depende de

Aluga



Depende de

Perguntas:

- Podemos deletar o apartamento enquanto houver um morador residindo lá dentro? R: Não, precisamos primeiro resolver o contrato (remover a FK) para depois excluir o imóvel (PK).
- Se o prédio for demolido, o que acontece com os contratos de aluguel ligados a ele? R: São automaticamente cancelados (Cascata).
- Podemos registrar um aluguel para um apartamento se este não existe? R: Não, o banco impede quando nota a inexistência.

Inatel

Relacionamentos

Conexões lógicas entre entidades que representam como os dados em diferentes tabelas estão associados uns aos outros.
No modelo relacional, **relacionamentos** ajudam a manter a **integridade referencial**, evitando dados duplicados e inconsistentes.

Classificação:

- 1:1
- 1:N
- N:M

Relacionamentos

❖ Relacionamento 1:1

Esse tipo de relacionamento é menos comum, mas útil quando há a necessidade de **separar dados sensíveis** ou dividir lógicas distintas. Por exemplo, em um sistema de RH, cada funcionário pode ter um único registro de pagamento.

❖ Relacionamento 1:N

É o tipo mais frequente de relacionamento em bancos de dados. Imagine um cliente que faz vários pedidos: temos uma relação **1 cliente → N pedidos**. Isso permite organizar o sistema de forma escalável, mantendo os dados centralizados e fáceis de acessar.

❖ Relacionamento N:M

Esse relacionamento ocorre quando várias instâncias de uma entidade se relacionam com várias instâncias de outra. Como não é possível representar isso diretamente em bancos relacionais, utilizamos **tabelas associativas** para intermediar a conexão.

Exercícios

- Se um cliente mudar de endereço, e os dados de endereço estiverem repetidos em cada linha da tabela de "Pedidos", o que acontece se esquecermos de atualizar uma dessas linhas?
- Por que não é recomendável usar o nome completo de uma pessoa como Chave Primária (PK), mesmo que, no momento do cadastro, não existam nomes iguais?
- Em um cadastro de hospital, qual a diferença prática entre um campo Data_de_Óbito com valor "0" e um com valor "NULL"?
- Se tentarmos cadastrar uma nota para um aluno que ainda não foi matriculado no sistema, qual conceito de banco de dados deve impedir essa ação?
- Em um sistema de biblioteca, onde um autor escreve vários livros, em qual das duas tabelas deve ficar a "ponte" (Chave Estrangeira)? Na tabela do Autor ou na tabela do Livro?
- Por que o banco de dados nos impede de excluir um "Curso" enquanto ainda existirem "Alunos" matriculados nele?

Exercícios

- Se você precisa guardar apenas "Sim" ou "Não", por que usar um campo de texto livre (VARCHAR) seria uma má escolha de modelagem?Por que não é recomendável usar o nome completo de uma pessoa como Chave Primária (PK), mesmo que, no momento do cadastro, não existam nomes iguais?
- No meio de uma transferência bancária, a energia do servidor cai. Por que o saldo não pode ter saído de uma conta sem ter entrado na outra?
- Por que as Chaves Estrangeiras são vitais para evitar que o banco de dados se torne um amontoado de "dados órfãos" (informações que existem, mas não se ligam a nada)?

AULA CONCLUÍDA!

Inatel