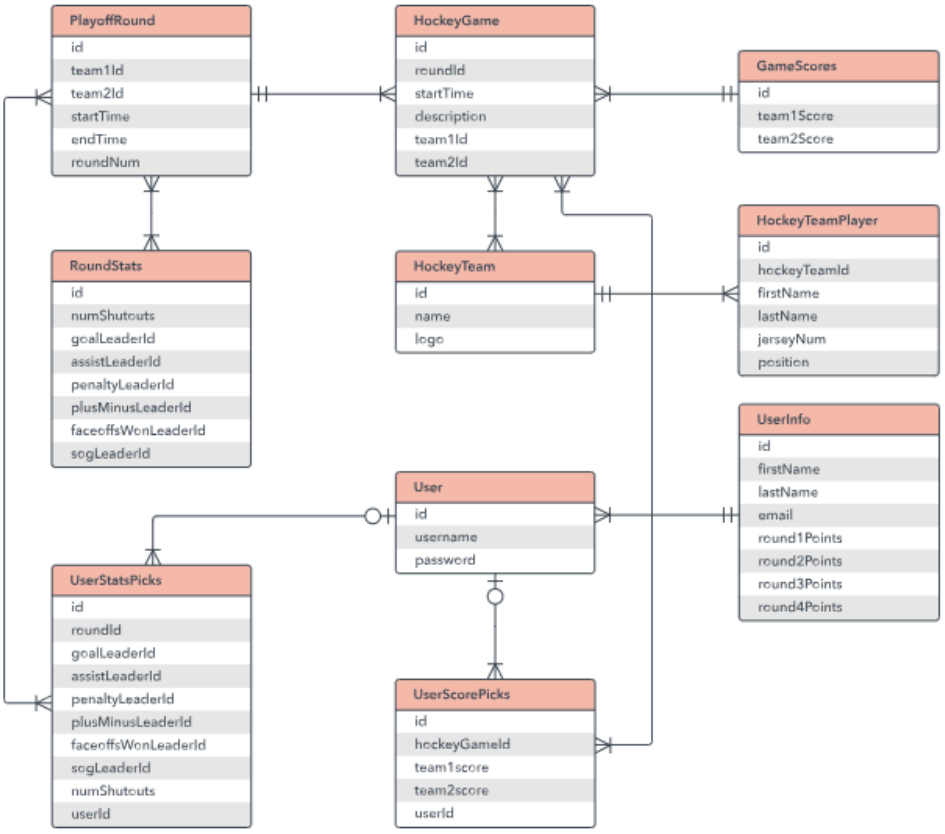


MODELAGEM DE DADOS

TRABALHO : DIAGRAMA ENTIDADE RELACIONAL

CURSO: GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

DISCIPLINA: MODELAGEM DE DADOS

UNIDADE FORMATIVA 3

ALUNO: LUIZ NUNES DE ALMEIDA JÚNIOR

Introdução

O banco de dados `biblioteca_db` foi projetado para gerenciar informações de uma biblioteca acadêmica, com foco no controle de alunos, livros, colaboradores e empréstimos. Ele contém quatro tabelas principais: `tb_aluno`, `tb_livro`, `tb_colaborador` e `tb_emprestimo`. O esquema foi criado utilizando o **MySQL Workbench**, com uso do mecanismo **InnoDB**, garantindo suporte a transações e integridade referencial e implementa um banco de dados relacional com chaves primárias, chaves estrangeiras, índices e restrições

1. Entidades do Modelo Relacional

O modelo relacional é composto pelas seguintes entidades:

1.1. Entidade `tb_aluno`

Tabela `tb_aluno`

Coluna	Tipo	Descrição
`id_aluno`	INT, AUTO_INCREMENT, PK	Chave primária sequencial do aluno.
`ra_aluno`	INT, UNIQUE	Registro acadêmico do aluno, único.
`nome_aluno`	VARCHAR(50)	Nome do aluno.
`email_aluno`	VARCHAR(100)	E-mail de contato do aluno.
`telefone_aluno`	VARCHAR(15)	Telefone de contato do aluno.

Descrição: Armazena os dados dos alunos.

- **Restrições**:

  `PRIMARY KEY (id_aluno)`:

- **Tipo**: Chave Primária
- **Descrição**: Garante que cada aluno tenha um identificador único. A chave primária é automaticamente indexada, o que melhora a performance de buscas e operações de join com esta tabela.

`UNIQUE INDEX (ra_aluno)`:

- **Tipo**: Índice Único
- **Descrição**: Garante que o valor do RA do aluno seja único em toda a tabela. Isso evita duplicação de registros e melhora a eficiência na busca por RA.
    -

1.2. Entidade `tb_livro`

Tabela `tb_livro`

Coluna	Tipo	Descrição
`id_livro`	INT, AUTO_INCREMENT, PK	Chave primária sequencial do livro.
`isbn_livro`	INT, UNIQUE	ISBN do livro, único.
`nome_livro`	VARCHAR(100)	Título do livro.
`autor_livro`	VARCHAR(45)	Nome do autor do livro.
`num_paginas_livro`	INT	Número de páginas do livro.

Descrição: Armazena as informações sobre os livros disponíveis na biblioteca.

- **Restrições**:
- `PRIMARY KEY (id_livro)`:
    - **Tipo**: Chave Primária
    - **Descrição**: Garante que cada livro tenha um identificador único. A chave primária é automaticamente indexada, facilitando buscas e joins.
- `UNIQUE INDEX (isbn_livro)`:
    - **Tipo**: Índice Único

- **Descrição**: Garante que cada ISBN de livro seja único. Isso previne a duplicação de livros e melhora a performance de consultas que buscam livros por ISBN.

1.3. Entidade `tb_colaborador`

Tabela `tb_colaborador`

Coluna	Tipo	Descrição
`id_colaborador`	INT, AUTO_INCREMENT, PK	Chave primária sequencial do colaborador.
`cpf_colaborador`	VARCHAR(14), UNIQUE	CPF do colaborador, único.
`nome_colaborador`	VARCHAR(50)	Nome do colaborador.
`email_colaborador`	VARCHAR(50)	E-mail de contato do colaborador.
`cargo_colaborador`	VARCHAR(45)	Cargo ou função do colaborador.

Descrição: Armazena os dados dos colaboradores da biblioteca.

- **Restrições**:

  `PRIMARY KEY (id_colaborador)`:

- **Tipo**: Chave Primária
- **Descrição**: Garante que cada colaborador tenha um identificador único. Facilita operações de busca e joins por meio de um índice automaticamente criado.

  `UNIQUE INDEX (cpf_colaborador)`:

- **Tipo**: Índice Único
- **Descrição**: Garante que cada CPF de colaborador seja único, evitando duplicações e melhorando a eficiência na busca por CPF.

1.4. Entidade `tb_emprestimo`

Tabela `tb_emprestimo`

Nome do Índice/Chave	Tipo	Colunas Envolvidas	Descrição
`PRIMARY`	Chave Primária	`id_emprestimo`	Define a coluna `id_emprestimo` como chave primária (PK).
`fk_tb_emprestimo_livro_isbn_idx`	Índice Normal	`id_livro_emprestimo`	Índice para a coluna que referencia o livro no empréstimo.
`fk_tb_emprestimo_aluno_idx`	Índice Normal	`id_aluno_emprestimo`	Índice para a coluna que referencia o aluno no empréstimo.
`fk_tb_emprestimo_colaborador_idx`	Índice Normal	`id_colaborador_emprestimo`	Índice para a coluna que referencia o colaborador no empréstimo.
`fk_tb_emprestimo_livro`	Chave Estrangeira	`id_livro_emprestimo`	Chave estrangeira que referencia `id_livro` de `tb_livro`.
`fk_tb_emprestimo_aluno`	Chave Estrangeira	`id_aluno_emprestimo`	Chave estrangeira que referencia `id_aluno` de `tb_aluno`.
`fk_tb_emprestimo_colaborador`	Chave Estrangeira	`id_colaborador_emprestimo`	Chave estrangeira que referencia `id_colaborador` de `tb_colaborador`.

Descrição: Registra os empréstimos de livros feitos pelos alunos.

- **Restrições**:

  `PRIMARY KEY (id_emprestimo)`:

- **Tipo**: Chave Primária
- **Descrição**: Garante que cada empréstimo seja identificado de forma única. A chave primária é automaticamente indexada, o que melhora a performance em buscas e joins relacionados a empréstimos.

  `INDEX fk_tb_emprestimo_livro_isbn_idx (id_livro_emprestimo)`:

- **Tipo**: Índice Normal
- **Descrição**: Cria um índice para a coluna `id_livro_emprestimo`, o que acelera consultas que envolvem a identificação do livro associado ao empréstimo.

  `INDEX fk_tb_emprestimo_aluno_idx (id_aluno_emprestimo)`:

- **Tipo**: Índice Normal
- **Descrição**: Cria um índice para a coluna `id_aluno_emprestimo`, otimizando consultas que envolvem a identificação do aluno que fez o empréstimo.

  `INDEX fk_tb_emprestimo_colaborador_idx (id_colaborador_emprestimo)`:

- **Tipo**: Índice Normal
- **Descrição**: Cria um índice para a coluna `id_colaborador_emprestimo`, melhorando a performance de buscas relacionadas ao colaborador responsável pelo empréstimo.

  `FOREIGN KEY (id_livro_emprestimo)`:

- **Tipo**: Chave Estrangeira
- **Descrição**: Refere-se à coluna `id_livro` na tabela `tb_livro`. Garante que o livro associado ao empréstimo exista na tabela de livros.

`FOREIGN KEY (id_aluno_emprestimo)`:

- **Tipo**: Chave Estrangeira
- **Descrição**: Refere-se à coluna `id_aluno` na tabela `tb_aluno`. Garante que o aluno que fez o empréstimo exista na tabela de alunos.

`FOREIGN KEY (id_colaborador_emprestimo)`:

- **Tipo**: Chave Estrangeira
- **Descrição**: Refere-se à coluna `id_colaborador` na tabela `tb_colaborador`. Garante que o colaborador responsável pelo empréstimo exista na tabela de colaboradores.

2. Relacionamentos entre as Entidades e Restrições de Integridade Referencial

- A tabela `tb_emprestimo` se relaciona com as tabelas `tb_aluno`, `tb_livro` e `tb_colaborador` por meio de chaves estrangeiras (`id_aluno_emprestimo`, `id_livro_emprestimo`, e `id_colaborador_emprestimo`). As ações de exclusão estão configuradas como **NO ACTION**, e as atualizações são **CASCADE**, garantindo que, ao atualizar os IDs nas tabelas relacionadas, os dados em `tb_emprestimo` também sejam atualizados.

3. Chaves Primárias Sequenciais e Otimização de Consultas com Joins

A utilização de chaves primárias sequenciais (**AUTO_INCREMENT**) traz diversas vantagens, especialmente em termos de desempenho:

- **Otimização de Joins**: Quando chaves primárias numéricas sequenciais são utilizadas, o gerenciamento de índices se torna mais eficiente. As consultas envolvendo joins, especialmente em grandes volumes de dados, são mais

rápidas, pois os SGBDs otimizam a busca por valores inteiros e sequenciais, facilitando a combinação de tabelas.

- **Índices e Performance**: A inserção automática de IDs sequenciais facilita a criação de índices em colunas que são frequentemente usadas para filtrar ou agrupar dados. No caso do sistema de biblioteca, colunas como `id_livro`, `id_aluno`, e `id_colaborador` são usadas em várias consultas e joins. A criação de índices sobre estas colunas melhora significativamente o desempenho em operações de leitura, como consultas para listar todos os empréstimos de um aluno ou localizar livros disponíveis.

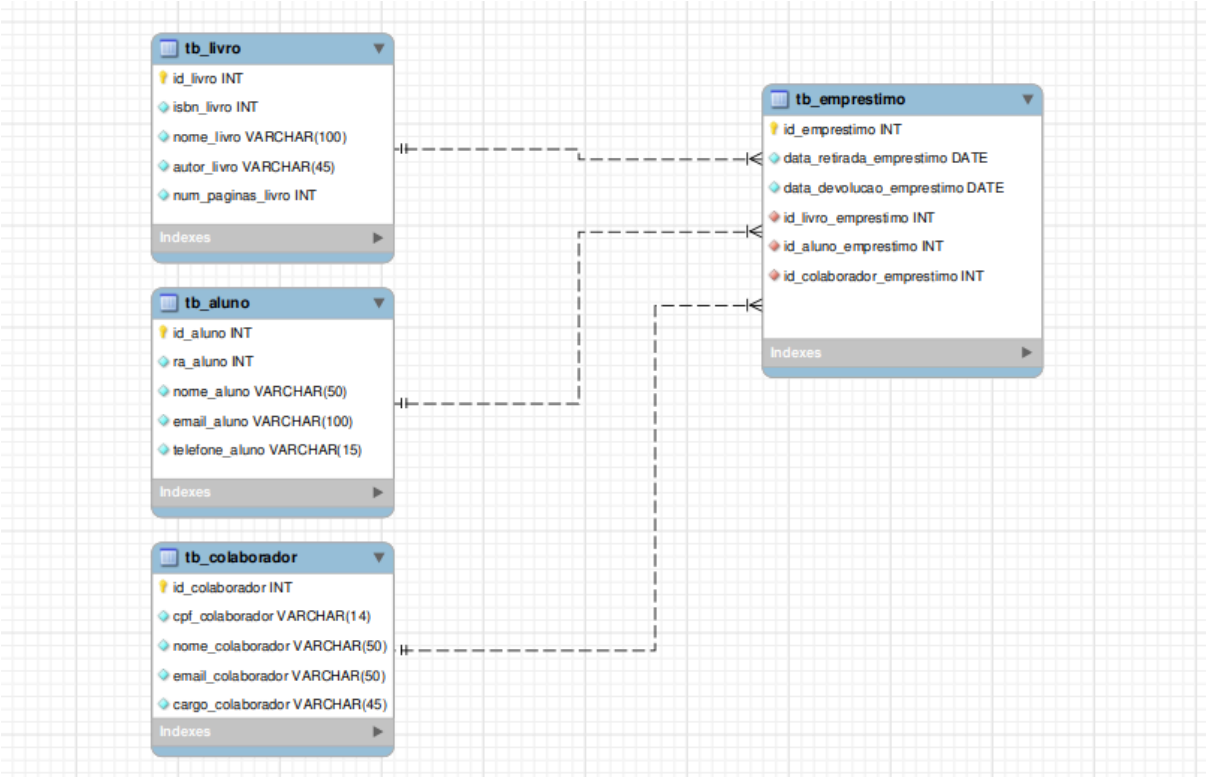
4. Criação de Índices para Consultas Específicas

O script de criação das tabela já define índices em várias colunas:

- **Índice único em `ra_aluno`** (tabela `tb_aluno`): Garante que o registro acadêmico do aluno seja único e melhora o tempo de resposta em consultas específicas que busquem alunos pelo RA.
- **Índice único em `isbn_livro`** (tabela `tb_livro`): O ISBN de cada livro também é único, o que agiliza a busca por este campo.
- **Índices de foreign keys em `tb_emprestimo`**: Foram criados índices para as chaves estrangeiras que referenciam as tabelas `tb_livro`, `tb_aluno` e `tb_colaborador`, otimizando as consultas que envolvem joins entre estas tabelas.

5. Representação visual do DER do Banco de dados Move Rent

Abaixo vemos na imagem a representação visual, construída usando o MySQL Workbench, do diagrama entidade relacional do banco de dados biblioteca_db, conforme proposto no exercício:



6. Scripts SQL para definição de entidades do Banco de Dados biblioteca_db

Abaixo, seguem os scripts SQL de criação das entidades do banco de dados biblioteca_db, criados a partir do DER do mesmo banco de dados, usando a ferramenta de geração de scripts do MYSQL WorkBench:

```
 -- MySQL Script generated by MySQL Workbench
-- Tue Sep 17 21:40:36 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,
NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_E
NGINE_SUBSTITUTION';
-- -----------------------------------------------------
-- Schema biblioteca_db
-- -----------------------------------------------------
```

```
-- -----------------------------------------------------
-- Schema biblioteca_db
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `biblioteca_db` ;
USE `biblioteca_db` ;


-- -----------------------------------------------------
-- Table `biblioteca_db`.`tb_aluno`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `biblioteca_db`.`tb_aluno` (
  `id_aluno` INT NOT NULL AUTO_INCREMENT,
  `ra_aluno` INT NOT NULL,
  `nome_aluno` VARCHAR(50) NOT NULL,
  `email_aluno` VARCHAR(100) NOT NULL,
  `telefone_aluno` VARCHAR(15) NOT NULL,
  UNIQUE INDEX `ra_aluno_UNIQUE` (`ra_aluno` ASC) VISIBLE,
  PRIMARY KEY (`id_aluno`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `biblioteca_db`.`tb_livro`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `biblioteca_db`.`tb_livro` (
  `id_livro` INT NOT NULL AUTO_INCREMENT,
  `isbn_livro` INT NOT NULL,
  `nome_livro` VARCHAR(100) NOT NULL,
  `autor_livro` VARCHAR(45) NOT NULL,
  `num_paginas_livro` INT NOT NULL,
  PRIMARY KEY (`id_livro`),
  UNIQUE INDEX `isbn_livro_UNIQUE` (`isbn_livro` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `biblioteca_db`.`tb_colaborador`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `biblioteca_db`.`tb_colaborador` (
  `id_colaborador` INT NOT NULL AUTO_INCREMENT,
  `cpf_colaborador` VARCHAR(14) NOT NULL,
  `nome_colaborador` VARCHAR(50) NOT NULL,
  `email_colaborador` VARCHAR(50) NOT NULL,
  `cargo_colaborador` VARCHAR(45) NOT NULL,
  UNIQUE INDEX `cpf_colaborador_UNIQUE` (`cpf_colaborador` ASC) VISIBLE,
  PRIMARY KEY (`id_colaborador`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `biblioteca_db`.`tb_emprestimo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `biblioteca_db`.`tb_emprestimo` (
  `id_emprestimo` INT NOT NULL AUTO_INCREMENT,
  `data_retirada_emprestimo` DATE NOT NULL DEFAULT CURRENT_DATE,
  `data_devolucao_emprestimo` DATE NOT NULL,
  `id_livro_emprestimo` INT NOT NULL,
  `id_aluno_emprestimo` INT NOT NULL,
  `id_colaborador_emprestimo` INT NOT NULL,
  PRIMARY KEY (`id_emprestimo`),
  INDEX `fk_tb_emprestimo_livro_isbn_idx` (`id_livro_emprestimo` ASC) VISIBLE,
  INDEX `fk_tb_emprestimo_aluno_idx` (`id_aluno_emprestimo` ASC) VISIBLE,
  INDEX `fk_tb_emprestimo_colaborador_idx` (`id_colaborador_emprestimo` ASC)
VISIBLE,
  CONSTRAINT `fk_tb_emprestimo_livro`
    FOREIGN KEY (`id_livro_emprestimo`)
    REFERENCES `biblioteca_db`.`tb_livro` (`id_livro`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `fk_tb_emprestimo_aluno`
    FOREIGN KEY (`id_aluno_emprestimo`)
    REFERENCES `biblioteca_db`.`tb_aluno` (`id_aluno`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `fk_tb_emprestimo_colaborador`
```

```
    FOREIGN KEY (`id_colaborador_emprestimo`)
    REFERENCES `biblioteca_db`.`tb_colaborador` (`id_colaborador`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

BIBLIOGRAFIA

Os arquivos com os scripts e imagens deste trabalho, bem como o arquivo do modelo gerado pelo MySQL WorkBench encontram-se disponíveis em https://github.com/lnalmeida/der-unopar-modelagem-de-dados.git