

Esquema do Banco de Dados & DER

Este documento descreve o esquema do banco de dados para o Sistema de Gerenciamento da Biblioteca Escolar usando PostgreSQL.

1. Diagrama de Entidade-Relacionamento (DER)

[Imagen de um Diagrama de Entidade-Relacionamento detalhado para um sistema de biblioteca]

Abaixo está um DER conceitual. PK denota Chave Primária e FK denota Chave Estrangeira.

- **Users** (PK: user_id)
- **Students** (PK: student_id, FK: user_id, class_id)
- **Teachers** (PK: teacher_id, FK: user_id)
- **Classes** (PK: class_id)
- **Teacher_Classes** (FK: teacher_id, class_id) --- *Tabela de Junção*
- **Genres** (PK: genre_id)
- **Books** (PK: book_id, FK: genre_id)
- **Rentals** (PK: rental_id, FK: book_id, student_id)
- **Class_Programs** (PK: program_id, FK: class_id, genre_id, teacher_id)
- **Program_Assignments** (PK: assignment_id, FK: program_id, book_id, student_id)
- **Reading_History** (PK: history_id, FK: student_id, book_id)
- **Quizzes** (PK: quiz_id, FK: book_id)
- **Quiz_Questions** (PK: question_id, FK: quiz_id)
- **Quiz_Attempts** (PK: attempt_id, FK: quiz_id, student_id)
- **Rewards** (PK: reward_id)
- **Student_Rewards** (FK: student_id, reward_id) --- *Tabela de Junção*

2. Definições das Tabelas

users

Armazena informações de login para todos.

- user_id (PK, SERIAL)
- email (VARCHAR, UNIQUE, NOT NULL)
- registration_number (VARCHAR, UNIQUE)
- password_hash (VARCHAR, NOT NULL)
- full_name (VARCHAR)
- role (ENUM: 'admin', 'librarian', 'teacher', 'student')
- created_at (TIMESTAMP)

classes

- class_id (PK, SERIAL)

- class_name (VARCHAR, ex: "5º Ano - Turma A")
- grade_level (INTEGER)

students

- student_id (PK, SERIAL)
- user_id (FK -> users.user_id)
- class_id (FK -> classes.class_id)
- date_of_birth (DATE)

teachers

- teacher_id (PK, SERIAL)
- user_id (FK -> users.user_id)

teacher_classes (Relacionamento Muitos-para-Muitos)

- teacher_id (FK -> teachers.teacher_id)
- class_id (FK -> classes.class_id)

genres

- genre_id (PK, SERIAL)
- genre_name (VARCHAR, UNIQUE, ex: "Fantasia", "Ficção Científica")

books

- book_id (PK, SERIAL)
- title (VARCHAR, NOT NULL)
- author (VARCHAR)
- isbn (VARCHAR, UNIQUE)
- genre_id (FK -> genres.genre_id)
- min_age (INTEGER)
- max_age (INTEGER)
- total_copies (INTEGER, DEFAULT 1)
- status (ENUM: 'disponível', 'emprestado', 'atribuido', 'danificado')

rentals (Para empréstimos padrão da biblioteca)

- rental_id (PK, SERIAL)
- book_id (FK -> books.book_id)
- student_id (FK -> students.student_id)
- rental_date (TIMESTAMP, DEFAULT NOW())
- due_date (TIMESTAMP)
- return_date (TIMESTAMP, NULLABLE)
- fine_amount (DECIMAL, DEFAULT 0)

class_programs (O plano de 3 meses do professor)

- program_id (PK, SERIAL)

- class_id (FK -> classes.class_id)
- genre_id (FK -> genres.genre_id)
- teacher_id (FK -> teachers.teacher_id)
- start_date (DATE)
- end_date (DATE)

program_assignments (Qual aluno recebe qual livro no programa)

- assignment_id (PK, SERIAL)
- program_id (FK -> class_programs.program_id)
- book_id (FK -> books.book_id)
- student_id (FK -> students.student_id)
- assigned_date (TIMESTAMP, DEFAULT NOW())

reading_history

Uma tabela crucial para rastrear cada livro que um aluno termina. Isso simplifica a verificação de 12 meses.

- history_id (PK, SERIAL)
- student_id (FK -> students.student_id)
- book_id (FK -> books.book_id)
- completion_date (DATE, NOT NULL)
- source (ENUM: 'emprestimo', 'programa')

quizzes, quiz_questions, etc.

Essas tabelas armazenariam o conteúdo dos questionários, as tentativas dos alunos e as pontuações.