Abaixo estão todas as tabelas necessárias para o sistema funcionar corretamente:

Tabela	Descrição
users	Armazena os dados de todos os usuários do sistema
user_courses	Relaciona usuários com cursos e semestres
classes	Registra as aulas criadas pelos professores
qr_codes	Registra os QR Codes gerados para cada aula
attendances	Guarda os registros de presença dos alunos
attendance_history	Histórico de alterações nos registros de presença
system_logs	Armazena logs de atividades administrativas

Agora vamos documentar as tabelas restantes e garantir que todos os campos e relacionamentos foram cobertos.



### 1. Tabela user\_courses (Relacionamento Usuário-Curso)



**Descrição:** Relaciona alunos e professores com cursos e semestres.

sql

```
CopiarEditar
CREATE TABLE user_courses (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
 course VARCHAR(100) NOT NULL,
 semester INT NOT NULL,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

### **\*** Explicação dos Campos:

Campo	Tipo	Descrição
id	INT (PK)	Identificador do vínculo
user_id	INT (FK)	ID do usuário vinculado ao curso
course	VARCHAR(100)	Nome do curso

Campo	Тіро	Descrição
semester	INT	Semestre do aluno ou professor

 ★ Consulta para Buscar Todos os Cursos de um Aluno:

sql

CopiarEditar

SELECT course, semester FROM user\_courses WHERE user\_id = 5;



#### 2. Tabela classes (Aulas)



**Descrição:** Registra as aulas criadas pelos professores.

sql

CopiarEditar

CREATE TABLE classes (

```
id INT AUTO_INCREMENT PRIMARY KEY,
```

professor\_id INT NOT NULL,

name VARCHAR(100) NOT NULL,

course VARCHAR(100) NOT NULL,

semester INT NOT NULL,

workload INT DEFAULT 60,

class\_type ENUM('presential', 'online') NOT NULL DEFAULT 'presential',

location VARCHAR(100),

description TEXT,

max\_students INT NULL,

status ENUM('scheduled', 'in\_progress', 'completed') NOT NULL DEFAULT 'scheduled',

date\_time DATETIME NOT NULL,

FOREIGN KEY (professor\_id) REFERENCES users(id) ON DELETE CASCADE

);

# Explicação dos Campos:

Campo	Tipo	Descrição
id	INT (PK)	Identificador único da aula

Campo	Tipo	Descrição
professor_id	INT (FK)	ID do professor responsável
name	VARCHAR(100)	Nome da disciplina
course	VARCHAR(100)	Curso associado
semester	INT	Semestre da turma
workload	INT	Carga horária total da disciplina
class_type	ENUM	Define se a aula é presential ou online
location	VARCHAR(100)	Sala de aula ou link de videoconferência
max_students	INT	Número máximo de alunos na turma
status	ENUM	Status da aula (scheduled, in_progress, completed)
date_time	DATETIME	Data e hora da aula

 ★ Consulta para Buscar Todas as Aulas de um Professor:

sql

CopiarEditar

SELECT \* FROM classes WHERE professor\_id = 3;



### 3. Tabela attendance\_history (Histórico de Presença)



**Descrição:** Mantém o histórico de alterações nos registros de presença.

sql

CopiarEditar

```
CREATE TABLE attendance_history (
  id INT AUTO_INCREMENT PRIMARY KEY,
  attendance_id INT NOT NULL,
  status ENUM('present', 'late', 'justified', 'absent') NOT NULL,
  changed_by INT NOT NULL,
  change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (attendance_id) REFERENCES attendances(id) ON DELETE CASCADE,
  FOREIGN KEY (changed_by) REFERENCES users(id) ON DELETE CASCADE
);
```

## **\*** Explicação dos Campos:

Campo	Tipo	Descrição
id	INT (PK)	Identificador do histórico
attendance_id	INT (FK)	Presença que foi alterada
status	ENUM	Novo status da presença (present, late, justified, absent)
changed_by	INT (FK)	ID do usuário que fez a alteração
change_date	TIMESTAMP	Data e hora da modificação

Consulta para Verificar Histórico de Presença de um Aluno:

sql

CopiarEditar

SELECT \* FROM attendance\_history WHERE attendance\_id = 10;

#### 4. Tabela system\_logs (Auditoria do Sistema)

**Descrição:** Registra todas as ações administrativas realizadas no sistema.

sql

CopiarEditar

```
CREATE TABLE system_logs (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  action TEXT NOT NULL,
  action_type ENUM('insert', 'update', 'delete', 'access') NOT NULL,
  affected_table VARCHAR(50),
  affected_row_id INT,
  old_value TEXT,
  new_value TEXT,
  ip_address VARCHAR(50),
  user_agent VARCHAR(255),
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

# Explicação dos Campos:

Campo	Tipo	Descrição
id	INT (PK)	Identificador único do log
user_id	INT (FK)	Usuário que executou a ação
action	TEXT	Descrição da ação realizada
action_type	ENUM	Tipo da ação (insert, update, delete, access)
affected_table	VARCHAR(50)	Nome da tabela afetada
affected_row_id	INT	ID do registro alterado
old_value	TEXT	Valor antes da alteração
new_value	TEXT	Valor após a alteração
ip_address	VARCHAR(50)	IP de onde foi feita a ação
user_agent	VARCHAR(255)	Informações do navegador/dispositivo usado
timestamp	TIMESTAMP	Data/hora da ação

sql

CopiarEditar

SELECT \* FROM system\_logs ORDER BY timestamp DESC LIMIT 10;