ChatGPT

🗃 Database Schema (v1)

This document presents the relational structure used by the School Organizer WebApp. It explains the main tables, their fields and the relationships between them. Cascading delete behavior ensures data consistency when users or courses are removed.

🏷 Table Definitions

Users

Field	Type	Constraints
id	serial	Primary key (auto-increment)
email	varchar(255)	Unique, not null
hashed_password	text	Not null
created_at	timestamp	Defaults to `now()`
is_active	boolean	Defaults to `true`
is_superuser	boolean	Defaults to `false`

Courses

Field	Type	Constraints
id	serial	Primary key
title	varchar(255)	Not null
description	text	Optional
user_id	integer	Foreign key → `users.id` (on delete cascade)
created_at	timestamp	Defaults to `now()`

Documents

Field	Type	Constraints
id	serial	Primary key
title	varchar(255)	Not null
url	text	Not null
type	varchar(100)	Optional (lecture, exercise, correction)
course_id	integer	Foreign key → `courses.id` (on delete cascade)

Field	Type	Constraints
created_at	timestamp	Defaults to `now()`

Events

Field	Type	Constraints
id	serial	Primary key
title	varchar(255)	Not null
date	date	Not null
description	text	Optional
course_id	integer	Foreign key → `courses.id` (on delete cascade)
created_at	timestamp	Defaults to `now()`

Relationships

- **User 1 → n Courses**: Each user can own multiple courses. When a user is deleted, all their courses (and dependent documents/events) are also deleted.
- **Course 1 → n Documents**: A course can have multiple documents. Deleting a course cascades to its documents.
- **Course 1 → n Events**: A course can have multiple events. Deleting a course cascades to its events.

Text Relationship Diagram

```
Users 1──< Courses 1──< Documents
                  │
                  └──< Events
```

Constraints & Indexes

- **Unique Constraint**: `users.email` is unique.
- **Foreign Keys**: `courses.user_id`, `documents.course_id`, `events.course_id` enforce referential integrity with `ON DELETE CASCADE`.
- **Indexes**: Indexes on foreign keys and commonly queried fields (e.g., `courses.user_id`) improve performance.

Future Tables (v2)

In future versions we may introduce:

Table	Purpose
shared_courses	Link between users and courses for sharing

Table	Purpose
tags	Many-to-many tagging of documents
notifications	Track upcoming deadlines and reminders