

Database Design for Web Apps

Web App Dev Guide · Module 7 of 8 · CHERIEDU Dev Series

1. Why Database Design Matters

A well-designed database is fast, reliable, and easy to extend. A poorly designed one causes slow queries, data duplication, and bugs that are hard to fix. Good design at the start saves months of pain later.

2. Core Database Concepts

Concept	Definition	Example
Table	A set of rows and columns	students, fees, attendance
Primary Key	Unique identifier per row	student_id = 1001
Foreign Key	Links rows between tables	fee.student_id references students.id
Index	Speeds up search queries	Index on student name
Join	Combine data from 2+ tables	Students + their attendance records

3. Schema for a School App

```
CREATE TABLE students (  
  id          INT PRIMARY KEY AUTO_INCREMENT,  
  school_id  INT NOT NULL,  
  name       VARCHAR(100) NOT NULL,  
  grade      INT,  
  email      VARCHAR(150) UNIQUE,  
  created_at DATETIME DEFAULT NOW()  
);  
  
CREATE TABLE fee_records (  
  id          INT PRIMARY KEY AUTO_INCREMENT,  
  student_id INT REFERENCES students(id),  
  amount      DECIMAL(10,2),  
  paid_date   DATE,  
  status      ENUM("paid", "pending", "overdue")  
);
```

4. SQL Queries You Use Every Day

```
-- Get all overdue fees with student names  
SELECT s.name, f.amount, f.paid_date  
FROM fee_records f  
JOIN students s ON f.student_id = s.id  
WHERE f.status = "overdue"  
ORDER BY f.amount DESC;
```



```
SELECT grade, COUNT(*) as total
```


5. Normalization Rules (Simplified)

- 1NF: Each column holds one value only — no comma-separated lists.
- 2NF: Every non-key column depends on the whole primary key.
- 3NF: No column should depend on another non-key column.
- Practical rule: If you repeat the same data in many rows, extract it to a separate table.

PROJECT

Design a database schema for CHERI SMS with at least 8 tables: schools, students, teachers, fees, attendance, exams, results, notifications. Define all relationships and foreign keys.

