

SQL und SQLite Cheatsheet

PRAXISLEITFADEN | Stand: 27. Oktober 2026 | Kostenlos

1. Tabellen und Constraints

Vorgehen: Nutze geeignete Datentypen, Primärschlüssel, UNIQUE und NOT NULL; aktiviere Foreign Keys pro Verbindung.

Beispiel: PRAGMA foreign_keys = ON;
CREATE TABLE users(id INTEGER PRIMARY KEY, email TEXT NOT NULL UNIQUE);

Prüfkriterium: Ungültige Referenzen und Duplikate werden abgewiesen.

Vermeiden: Foreign-Key-Prüfung als automatisch aktiv voraussetzen.

2. Sichere Abfragen

Vorgehen: Nutze Prepared Statements statt SQL-Zeichenketten mit Nutzereingaben.

Beispiel: SELECT id FROM users WHERE email = ?;

Prüfkriterium: Sonderzeichen verändern die SQL-Struktur nicht.

Vermeiden: Eingaben manuell per Stringverkettung einbauen.

3. Transaktionen

Vorgehen: Fasse zusammengehörige Änderungen atomar zusammen.

Beispiel: BEGIN;
UPDATE accounts SET balance=balance-10 WHERE id=1;
UPDATE accounts SET balance=balance+10 WHERE id=2;
COMMIT;

Prüfkriterium: Entweder alle Änderungen sind übernommen oder keine.

Vermeiden: Bei Fehlern ROLLBACK vergessen.

4. Indizes gezielt setzen

Vorgehen: Prüfe häufige WHERE-, JOIN- und ORDER-BY-Abfragen mit EXPLAIN QUERY PLAN.

Beispiel: CREATE INDEX idx_users_email ON users(email);

Prüfkriterium: Query-Plan und Laufzeit verbessern sich messbar.

Vermeiden: Jede Spalte ohne Bedarf indexieren.

5. Backup und Wartung

Vorgehen: Erzeuge konsistente Sicherungen über SQLite-Backup-API oder .backup und prüfe Integrität.

Beispiel: `.backup backup.sqlite`

`PRAGMA integrity_check;`

Prüfkriterium: Backup lässt sich öffnen und Integritätsprüfung meldet ok.

Vermeiden: Aktive WAL-Datenbank unkoordiniert per Dateikopie sichern.

Fachquellen

SQLite Documentation – <https://www.sqlite.org/docs.html>

Onlineversion: <https://www.cyberknecht.de/wissen/cheatsheets/sqlite-sql-cheatsheet.html>