

## Dokumentation Datenbank:

### Auswahl der Datenbank:

#### Optionen für Datenbanken:

- **SQLite:** eingebaut in Django, aber besser geeignet für die Entwicklung von kleineren Anwendungen.
- **MySQL:** beliebt für Webanwendungen, gut unterstützt, es gibt aber Einschränkungen bei erweiterten Funktionen.
- **PostgreSQL:** empfohlen für mittlere bis große Anwendungen, unterstützt komplexe Abfragen, transaktionale Integrität und hat starke JSON-Unterstützung.

#### Warum PostgreSQL bevorzugt wurde:

- **Erweiterte Funktionen:** unterstützt Geodaten, komplexe Abfragen und hat umfangreiche JSON-Funktionen, also sehr gut für datenintensive Anwendungen, unterstützt Screenreader
- **Skalierbarkeit:** gut geeignet für wachsende Anwendungen, PostgreSQL kann große Mengen von Daten effizient verwalten.
- **Zuverlässigkeit und Integrität:** Bietet robuste Funktionen zur Gewährleistung der Datenintegrität, gut integriert in Django, was die Entwicklung sehr erleichtert.

### Tabellenstruktur und Parameter (Datenbankschema.txt):

#### Tabelle Nodes:

**node\_id:** (Primärschlüssel)  
**name:** (Raumbezeichnung)  
**type:** (Raum, Korridor, WC)

#### Tabelle Edges:

**edge\_id:** (Primärschlüssel)  
**from\_node\_id:** (Fremdschlüssel zu Nodes)  
**to\_node\_id:** (Fremdschlüssel zu Nodes)  
**weight:** (Distanz)  
**description:** (Wegbeschreibung)

### Vorteile Integration von PostgreSQL mit Django:

- einfache Konfiguration
- **ORM (Object-Relational Mapping):** ermöglicht es, Datenbankabfragen in Python zu schreiben, ohne SQL zu verwenden
- **Sicherheit:** Django bietet eingebaute Schutzmaßnahmen gegen SQL-Injection
- **Leistung:** PostgreSQL-Integration bedeutet, dass man von der Leistungsfähigkeit und Effizienz von PostgreSQL profitieren kann

### **Erstellung der Tabellenmodelle in Django:**

(nächster Sprint)

### **Aufbau der Datenbank mit diesen Tabellen:**

Django Modelle definieren → Migrationen erstellen → Daten einfügen

### **Zusammenfassung:**

PostgreSQL in Kombination mit Django bietet eine robuste, skalierbare und sichere Plattform für unser Projekt. Diese Kombination erleichtert die Entwicklung, unterstützt viele Funktionen und erleichtert die Verwaltung von komplexeren Anwendungen.

Datenbankverwaltung wird durch Djangos ORM und die Integration mit PostgreSQL sehr vereinfacht und die Sicherheit wird ebenso verbessert.