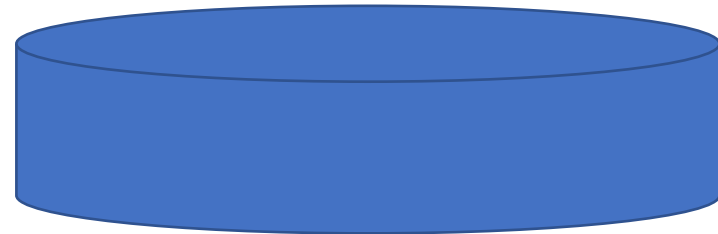


Datenbanken

Django unterstützt zahllose Datenbanken. Durch den integrierten **ORM** (Objekt Relation Mapper) muss sich der Code nicht ändern, wenn auf eine andere Datenbank gewechselt wird.

Der Mapper erzeugt je nach genutzter Datenbank anderes, der Datenbank entsprechendes SQL.



SQLITE

Im Entwicklungsstadium oder zu Testzwecken bietet es sich an, auf die Default Datenbank **SQLITE** zurückzugreifen. Sie ist Bestandteil von Python und muss nicht extra installiert werden.

In diesem Kurs nutzen wir SQLITE.

SQLITE allerdings NIEMALS im produktiven Umfeld, d.h. in Liveversionen eingesetzt werden, vor allem nicht mit vielen Zugriffen.



unterstützte Datenbanken

Django unterstützt
offiziell 5
Datenbank-
Systeme:

PostgreSQL

MySQL

SQLite

Oracle

MariaDB (Django 3
only)

Settings

in den Projekteinstellungen settings.py können wir die Datenbank festlegen, die genutzt werden soll:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

SQLITE Browser

für Windows gibt es ein gutes Tool, um sich Datenbankinhalte genauer zu betrachten:


<https://sqlitebrowser.org>

A large orange circle occupies the left side of the slide, partially cut off by the edge.

Postgres

Das in der Pythonwelt am häufigsten genutzte Datenbanksystem ist wohl Postgres. Es ist geeignet für Large-Scale Projekte mit mehreren hundert Millionen Usern und Datensätzen.




A large orange circle is positioned on the left side of the slide, partially cut off by the edge.

Multiple Databases in einem Projekt

Es ist durchaus möglich, mehr als ein Datenbanksystem im gleichen Projekt zu nutzen (zb. eine DB für user, eine für content).

Das ist allerdings für kleine bis mittelgroße Projekte eher unüblich und erfordert auch hohen Mehraufwand bei der Programmierung.

A series of four yellow dashed line segments are arranged in a curved, upward-pointing arc in the bottom right corner of the slide.