

Creating a Datamodel



Reindert-Jan Ekker

@rjekker | <https://nl.linkedin.com/in/rjekker>



Overview



Data model classes

Read/write events.db

Sqlalchemy

Make db url configurable with dotenv



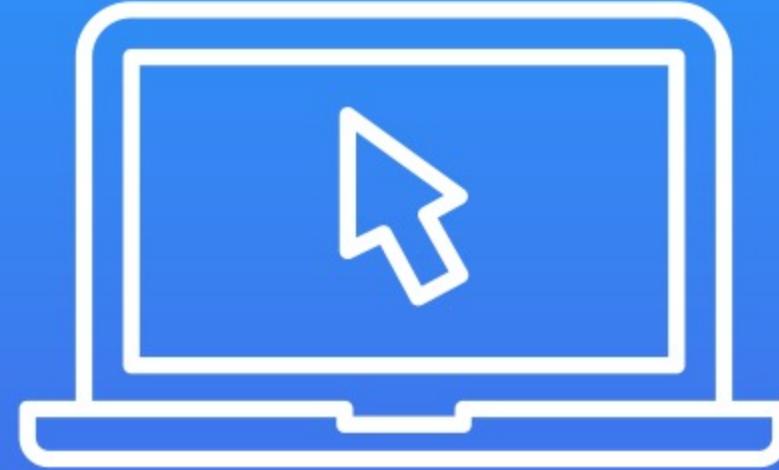
Table Definitions

```
create table category
(
    id      INTEGER not null
            primary key,
    name   VARCHAR not null
            unique
);

create table event
(
    id          INTEGER not null
            primary key,
    product_code VARCHAR not null
            unique,
    date        DATE      not null,
    price       NUMERIC  not null,
    category_id INTEGER  not null
            references category
);
```



Demo



- Create a DB connection
- Create a class for categories
- Create a class for events
- Create the relationship
- Make DB configurable
- Commit



The Event Model

models.db

```
class DBEvent(Base):
    __tablename__ = "event"

    id: Mapped[int] = mapped_column(primary_key=True)
    product_code: Mapped[str] = mapped_column(unique=True)
    date: Mapped[date]
    price: Mapped[decimal.Decimal]
    category_id: Mapped[int] = mapped_column(ForeignKey("category.id"))
    category: Mapped["DBCategory"] = relationship(back_populates="events")
```



The Category Model

models.db

```
class DBCategory(Base):
    __tablename__ = "category"

    id: Mapped[int] = mapped_column(primary_key=True)
    name: Mapped[str] = mapped_column(unique=True)
    events: Mapped[list["DBEvent"]] = relationship(back_populates="category")
```



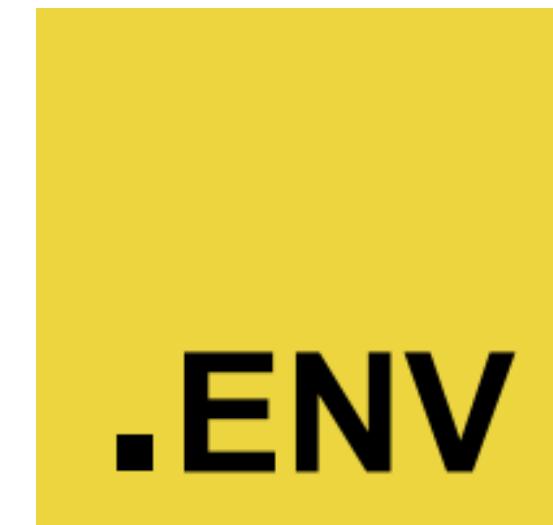
Configurable settings: .env

Install python-dotenv

Configure project settings

.env file or environment variables

.env file is usually not committed to VCS



Using dotenv

.env

```
DATABASE_URL="sqlite:///events.db"
```

my-script.py

```
import os
from dotenv import load_dotenv

load_dotenv()
DATABASE_URL = os.getenv("DATABASE_URL")
```



Summary



Setup DB connection

Create data model classes

Use dotenv for settings



Up Next:

**Make content accessible
through an API**

