

```

1 # Appendix C10 - tables.py
2
3 from sqlalchemy import Table, Column, Integer, String, Text
4 from sqlalchemy import ForeignKey
5 from sqlalchemy.orm import relationship, backref
6 from .base import Base
7
8 class Topic(Base):
9     __tablename__ = 'topics'
10
11     id = Column(Integer, primary_key=True)
12     title = Column(String)
13     base_path = Column(String, unique=True)
14     description = Column(String)
15     web_url = Column(String)
16     api_url = Column(String)
17
18     def __repr__(self):
19         return "<Topic(title='%s', base_path='%s')>" % (self.title, self.base_path)
20
21     def documents(self):
22         documents = set()
23         for subtopic in self.subtopics:
24             for doc in subtopic.documents:
25                 documents.add(doc)
26
27         return list(documents)
28
29     def documents_with_labels(self):
30         doc_with_labels = []
31         for doc in self.documents():
32             doc_with_labels.append([doc, doc.topic_titles()])
33         return doc_with_labels
34
35 # create association table SubtopicDocument
36 subtopics_documents = Table('subtopics_documents', Base.metadata,
37     Column('subtopic_id', ForeignKey('subtopics.id'), primary_key=True),
38     Column('document_id', ForeignKey('documents.id'), primary_key=True)
39 )
40
41 class Subtopic(Base):
42     __tablename__ = 'subtopics'
43
44     id = Column(Integer, primary_key=True)
45     title = Column(String)
46     base_path = Column(String, unique=True)
47     description = Column(String)
48     web_url = Column(String)
49     api_url = Column(String)
50
51     topic_id = Column(Integer, ForeignKey('topics.id'))
52     topic = relationship("Topic", back_populates="subtopics")
53
54     documents = relationship(
55         "Document", secondary=subtopics_documents, back_populates="subtopics"
56     )
57
58     def __repr__(self):
59         return "<Subtopic(title='%s', base_path='%s')>" % (self.title, self.base_path)
60
61 # link topic to subtopics
62 Topic.subtopics = relationship(
63     "Subtopic", order_by=Subtopic.id, back_populates="topic"
64 )
65
66 class Document(Base):
67     __tablename__ = 'documents'

```

```

68
69     id            = Column(Integer, primary_key=True)
70     title         = Column(String)
71     base_path     = Column(String, unique=True)
72     web_url       = Column(String)
73     html          = Column(Text)
74     description   = Column(Text)
75     content       = Column(Text)
76
77     subtopics = relationship(
78         'Subtopic', secondary=subtopics_documents, back_populates='documents'
79     )
80
81     def __init__(self, title, base_path, html=None, description=None, web_url=None, content=None):
82         self.title = title
83         self.base_path = base_path
84         self.html = html
85         self.web_url = web_url
86         self.description = description
87         self.content = content
88
89     def __repr__(self):
90         return "<Document(title=%r, base_path=%r)" % (self.title, self.base_path)
91
92     def topics(self):
93         topics = set()
94
95         for subtopic in self.subtopics:
96             topics.add(subtopic.topic)
97
98         return list(topics)
99
100     def topic_titles(self):
101         return [topic.title for topic in self.topics()]
102

```