

---

# Anvil ORM

Owen Campbell

Mar 01, 2023



# CONTENTS

<b>1</b>	<b>Install the ORM</b>	<b>3</b>
<b>2</b>	<b>Create the Data Tables</b>	<b>5</b>
<b>3</b>	<b>Create the Model Module</b>	<b>7</b>
<b>4</b>	<b>Add Some Entries</b>	<b>9</b>
<b>5</b>	<b>Fetch the Entries Back Again</b>	<b>11</b>
<b>6</b>	<b>Make a Change</b>	<b>13</b>
<b>7</b>	<b>Howto...</b>	<b>15</b>
7.1	Install the Library . . . . .	15
7.2	Write Model Classes . . . . .	15
7.3	Create and Update Records . . . . .	15
7.4	Query the Database . . . . .	15
7.5	Secure your App . . . . .	15
7.6	Use Cross References . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>



For this tutorial, we're going to build a model to represent Books and their Authors. You will need:

- A new anvil app
- The data tables service enabled
- This model library installed

For a book, we want to store its title, publication date and author. For an author, we want to store a first name and a last name.



## **INSTALL THE ORM**

Create a package in the server code section of your app named 'orm\_server' and two modules within that package named 'persistence' and 'security'. Replace the contents of both modules with the code from [persistence.py](#) and [security.py](#) respectively.

Create a package named 'orm\_client' in the client code section of your app and a module within that package named 'particles'. Replace its content with the code from [particles.py](#)





## CREATE THE DATA TABLES

Create a data table called 'author' with three columns:

- a text column called 'uid'
- a text column called 'first\_name'
- a text column called 'last\_name'

Create a second table called 'book' with four columns:

- a text column called 'uid'
- a text column called 'title'
- a date column called 'published\_on'
- a column which links to a single column in the 'author' table named 'author'.



## CREATE THE MODEL MODULE

Create a module in the client code section of your app and call it 'model'. Add the following code to define the Book and Author classes:

```
from .orm_client.particles import model_type, Attribute, Relationship

@model_type
class Author:
    first_name = Attribute()
    last_name = Attribute()

@model_type
class Book:
    title = Attribute()
    published_on = Attribute()
    author = Relationship(class_name="Author")
```



## ADD SOME ENTRIES

Create a new module in the client code section of your app, name it 'startup' and set it as the startup module. Delete its content and replace it with:

```
import datetime as dt
from .model import Book, Author

print("Creating Luciano")
luciano = Author(first_name="Luciano", last_name="Ramalho").save()

print("Creating Drew")
drew = Author(first_name="Drew", last_name="Neil").save()

print("Creating Fluent Python")
fluent_python = Book(
    title="Fluent Python",
    published_on=dt.datetime(2015, 8, 21).date(),
    author=luciano
).save()

print("Creating Practical Vim")
practical_vim = Book(
    title="Practical Vim",
    published_on=dt.datetime(2017, 1, 1).date(),
    author=drew
).save()
```

Launch your app and stop it again. You should see the results of the print statements in its output console.

Check the contents of your data tables. You should see both authors and books in their respective tables, with id numbers automatically assigned. You should also see two entries in the sequence table with the 'next' values ready for the next author and book.



## FETCH THE ENTRIES BACK AGAIN

Delete the code in your startup app (or rename it and create a new 'startup' module) and replace the content with:

```
from .model import Book, Author

for book in Book.search():
    print(book.title)
    print(book.author.first_name)
```

Launch the app and you should see the title and author's first name for both books in your output console.





## MAKE A CHANGE

Delete the code in your startup app (or rename it and create a new ‘startup’ module) and replace the content with:

```
from .model import Book, Author

fluent_python = Book.search(title="Fluent Python")[0]
fluent_python.title = "Fluent Python (Clear, Concise, and Effective Programming)"
fluent_python.save()

practical_vim = Book.get(id=2)
practical_vim.title = "Practical Vim (Edit Text at the Speed of Thought)"
practical_vim.save()
```

Start and stop the app and check your data tables. You should see the updated titles for both book rows.



**HOWTO...**

**7.1 Install the Library**

**7.2 Write Model Classes**

**7.3 Create and Update Records**

**7.4 Query the Database**

**7.5 Secure your App**

**7.6 Use Cross References**



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`