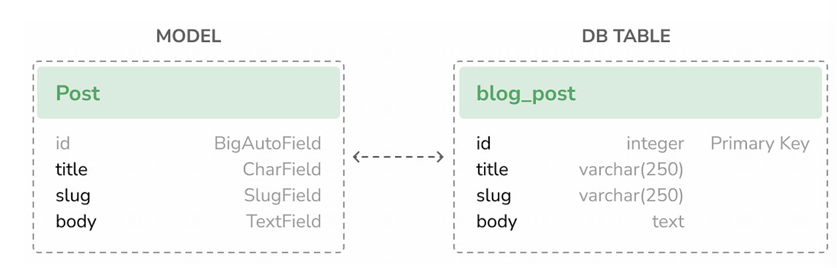
# Building a Blog Application

* This chapter will cover the following topics:
* Installing Python
* Creating a Python virtual environment
* Installing Django
* Creating and configuring a Django project
* Building a Django application
* Designing data models
* Creating and applying model migrations
* Setting up an administration site for your models
* Working with QuerySets and model managers
* Building views, templates, and URLs
* Understanding the Django request/response cycle

**Creating the blog data models**

Classes are the blueprint for bundling data and functionality together. Creating a new class creates a new type of object, allowing you to create instances of that type.

**Creating the Post model**

You can also define one of the model fields to be the primary key by setting primary\_key=True on it.

Another method to define default values for model fields is using database-computed default values.

**Defining a default sort order**

**class Meta:**

**ordering = ['-publish']**

We have added a Meta class inside the model. This class defines metadata for the model. We use the ordering attribute to tell Django that it should sort results by the publish field. We indicate descending order by using a hyphen before the field name, -publish.

**Adding a database index**

Let’s define a database index for the publish field. This will improve performance for query filtering or ordering results by this field.

**class Meta:**

**ordering = ['-publish']**

**indexes = [**

**models.Index(*fields*=['-publish'])**

**]**