**Starting a new project**

Start a new project by

“django-admin startproject ProjectName”

**Creating an application**

Add an application by

“python manage.py startapp AppName”

Need to add an urls.py file in the application

from django.urls import path

from . import views

urlpatterns = []

Then, include these urls in the url file of the main application

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

    path('admin/', admin.site.urls),

    path('database/', include("database.urls"))

]

Finally, install this application in the settings.py file

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'database'

]

**Django concepts**

A Django project can be separated into multiple applications. Our applications will have a html, a css, and a javascript folder to store the appropriate web programming files for organization.

The “views.py” file defines the functions that can be called when a url is visited. We associate these functions to urls in the “urls.py” file.

e.g

we have in the views.py file:

def index(request, name):

    return render(request, "database/index.html",{"name": name})

and in the urls.py file:

urlpatterns = [

    path("<str:name>", views.index, name="index")

]

This tells Django to call the index function in views.py when the link “…/something” is visited, pass to it the “something” at the end of the url as a parameter which we call “name” in the index function.

The index function renders the index.html file and pass into it the “name” parameter which is subsequently inserted into the html file through Django’s formatting language.

We run the Django server to see our web apps by

“python manage.py runserver”

Quit running server by Ctrl-C

**SQL in Django**

Create tables by defining classes in models.py

e.g:

from django.db import models

# Create your models here.

class Program(models.Model):

    program\_name = models.CharField(max\_length = 50)

    status = models.SmallIntegerField

    phase = models.SmallIntegerField

    enterproj\_id = models.IntegerField

    wbs\_number = models.CharField(max\_length = 30)

    oem = models.CharField(max\_length = 20)

anytime there’s a change in the structure of any table, we need to make a migration, this is done in 2 steps:

“python manage.py makemigrations”

This creates the instructions to apply the migration

“python manage.py migrate”

This makes the actual changes in the database

Columns (fields) also come with options like

    a.primary\_key = True

    b.null = True

or

    q = models.ForeignKey('self', on\_delete =models.CASCADE)

**\*\*Note:** the driver we are using is provided by a third-party and one of the limitations is that it cannot change the name of foreign keys once it has been migrated for the first time.

String representation of Objects, or what gets shown when query sets are returned:

def \_\_str\_\_(self):

return f”{self.attribute 1} ….”

**SQL queries in Django:**

**Inserting into table “Class”**

Variablename = Class(attribute1 = …, attribute 2 = …, attribute 3 = …)

Variablename.save()

**Select All**

Variable1 = Class.objects.all()

**Select Where**

Variable1 = Class.objects.get(…)

**Selecting a row from a query set, selecting columns in a row**

Variable2 = variable1.next()

Variable2.attribute1

**Searching for all foreign keys that references a certain primary key:**

First, give the foreign keys a related name

primarykeyobjectyouarelookingfor.relatedname.all()

will pull up all entries of the foreign key table that refers to that specific primary key.

**Datatypes for columns:**

Below are the datatypes from MSSQL that match those available on Django.

Nvarchar = CharField()

Numeric = DecimalField()

Int = IntegerField()

Bigint = BigIntegerField()

Date = DateField()

Datetimeoffset = DateTimeField()

Float = FloatField()

Smallint = SmallIntegerField()

Time = TimeField()

**Django package for displaying html**

1. Displaying tables in html.

Package installed: django\_tables2, bootstrap

In views.py:

From django\_tables2.views import SingleTableMixin

In views.py, we define classes that instructs how tables should be displayed in html. Each web page will have a class for its table.

SingleTableMixin class allows for a combination of other classes to be used in views.py.

CreateView class allows for adding new rows to our tables or database.

FilterView class allows for filtering rows in our database.

Class attributes:

model = Program (is the table model in models.py, need to import in file)

table\_class (from tables.py, is a class that defines what shows on our tables, whether we have a delete and an update column. What url to visit when we delete, update rows. Class Meta defines what SQL table it should display, what columns it should not,…)

(

Delete, update buttons send you to a url that runs functions in views.py that deletes or updates a row.

)

Filters and add forms takes from forms defined in filters.py, forms.py.

Currently, adding, updating, and deleting actions are defined in separate classes.

**Useful links to help with Django web programming:**

Specifying column datatypes (or fields within models as they are referred to): <https://docs.djangoproject.com/en/5.0/ref/models/fields/>

MSSQL driver and limitations:

<https://pypi.org/project/mssql-django/>

Django template language:

<https://docs.djangoproject.com/en/4.2/ref/templates/language/>

Django custom tags and filters:

<https://docs.djangoproject.com/en/4.2/howto/custom-template-tags/>

Django tutorial and Django with SQL:

CS50-Introduction to Web Programming on EdX (free course)

Database structure:

<https://lucid.app/lucidchart/5a174aaf-0b5b-4878-92e8-f4030b7aa274/edit?viewport_loc=-3514%2C-1634%2C5444%2C2669%2C0_0&invitationId=inv_b7e93a63-5ffc-471b-90af-783e55bf92a7>

Django-mssql documentation (is the package that our driver was forked from, slightly different to our driver):

<https://django-mssql.readthedocs.io/en/latest/>

MSSQL datatypes:

<https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>

Django documentation:

<https://docs.djangoproject.com/en/5.0/>

Making SQL queries in Django:

<https://docs.djangoproject.com/en/5.0/topics/db/queries/>