

## Models and Fields

In django we access and manage data through python objects referred as models. Model have a proper structure how we store our data, data field type, max size, default values and so on.

For Example:

```
class details(models.Model):  
  
    first_name=models.CharField(max_length=30)  
  
    last_name=models.CharField(max_length=20)  
  
    age = models.IntegerField()
```

Lets understand little about database and datastructure.

There are two way we can store our data one is SQL in tabular format and another in no SQL. In Django user writes the data in python code and Django automatically converts that key value paired data to SQL and processes the SQL data.

There are many SQL engines that processes data in tabular format. Some of the popular ones are:

- mySQL
- MongoDB
- SQLite
- PostgreSQL
- Inter Base

When we install python we automatically install SQLite engine, so if we are using Django Python we already have SQLite installed. So lets use this in our project.

If you are using SQL engine other than SQLite in your project then you have to make some changes in settings.py DATABASES={'ENGINE':.....}

You can check the details at database documentation section of Django documentation.

Lets start how to create and connect our models and use it in our project using SQLite.

1. `django-admin startproject my_site`
2. `cd my_site`
3. `python manage.py startapp office`
4. `python manage.py migrate`
5. Go to `model.py` in app level to create model

```
from unittest.util import _MAX_LENGTH
```

```
from django.db import models
```

```
from django.forms import CharField
```

```
# Create your models here.
```

```
class details(models.Model):
```

```
    first_name=models.CharField(max_length=30)
```

```
    last_name=models.CharField(max_length=20)
```

```
    age = models.IntegerField()
```

6. Connect and register the above created model to the database

- Go to `app.py` and copy configuration name
- Open `setting.py` and register that name

```
INSTALLED_APPS = ["office.apps.OfficeConfig",
```

7. Run the `makemigration` command for the app you created. This creates SQL table of data using our python code from models inside migrations folder `__inti__.py`. For example `001_initial.py`

```
python manage.py makemigrations office
```

8. Run `migrate` command

```
python manage.py migrate
```

Note- You can look what SQL file django generated behind the scene using our python code just use following command

```
python manage.py sqlmigrate office 0001
```