**Save Update Delete Form Data to Database Table in Django 5**

**code file 34**

**video no 40**

# 🔥 Django CRUD Operations with Forms

In Django, the common operations for working with the database are called **CRUD**:

* **C → Create (Save new record)**
* **R → Read (Fetch/View data from DB)**
* **U → Update (Modify existing data)**
* **D → Delete (Remove record from DB)**

You’re working with **Forms + Model + Views**, so let’s break down what’s happening.

## ✅ 1. Model (models.py)

**from** django**.**db **import** models

*# Create your models here.*

class **Profilee**(*models***.***Model*)**:**   *# A table 'Profilee' will be created in DB*

    name **=** models**.**CharField(**max\_length=**255)     *# Name column*

    email **=** models**.**EmailField(**max\_length=**255)   *# Email column*

    address **=** models**.**CharField(**max\_length=**255)  *# Address column*

    passward **=** models**.**CharField(**max\_length=**255) *# Password column (typo, should be "password")*

### 🔎 Explanation

* A **Model** in Django = Database Table.
* Each **Field** = Column in database.
* Example: name = models.CharField(max\_length=255) means table column name of type VARCHAR(255).
* Django automatically creates a **primary key id column** (auto-increment integer) for each table.

👉 After making a model:

python manage**.**py makemigrations

python manage**.**py migrate

This applies the model to the database (creates the table).

## ✅ 2. Form (forms.py)

**from** django **import** forms

class **Profile**(*forms***.***Form*)**:**   *# This is a simple form, not linked to DB directly*

    name **=** forms**.**CharField()

    email **=** forms**.**EmailField()

    address **=** forms**.**CharField()

    passward **=** forms**.**CharField(**widget=**forms**.**PasswordInput)  *# hides password input*

### 🔎 Explanation

* forms.Form is a plain Django form (NOT directly connected to Model).
* It just validates input fields.
* Later, we’ll also learn **ModelForm**, which automatically connects to Profilee.

## ✅ 3. View (views.py)

**from** django**.**shortcuts **import** render

**from** student**.**forms **import** Profile

**from** django**.**http **import** HttpResponseRedirect

**from** student**.**models **import** Profilee

*# Create your views here.*

**def** Register(**request**)**:**

**if** request**.**method **==** 'POST'**:**   *# When form is submitted*

        form **=** Profile(request**.**POST)   *# Bind submitted data to form*

**if** form**.**is\_valid()**:**   *# Check validation*

            nm **=** form**.**cleaned\_data['name']       *# Extract form data*

            em **=** form**.**cleaned\_data['email']

            ad **=** form**.**cleaned\_data['address']

            pw **=** form**.**cleaned\_data['passward']

*# ----------- SAVE (CREATE) -----------*

*# Save new data into DB*

*# User = Profilee(name=nm, email=em, address=ad, passward=pw)*

*# User.save()*

*# ----------- UPDATE -----------*

*# Update existing record where id=1*

*# User = Profilee(id=1, name=nm, email=em, address=ad, passward=pw)*

*# User.save()   # If id=1 exists → update, else create new*

*# ----------- DELETE -----------*

*# Delete record where id=1*

*# User = Profilee(id=1)*

*# User.delete()*

**return** HttpResponseRedirect('/student/register/')  *# Redirect to same page*

**else:**

        form **=** Profile()   *# Empty form for GET request*

**return** render(request**,**'student/register.html'**,**{'form'**:**form})

## ✅ 4. Template (register.html)

**<**form method**=**"POST"**>**

    {**%** csrf\_token **%**}

    {{ form**.**as\_p }}   **<**!-- Django renders form fields automatically --**>**

**<**input *type***=**"submit" value**=**"Submit"**>**

**</**form**>**

# 🔥 CRUD Explained with Code

### 1. **CREATE (Save new data into DB)**

User **=** Profilee(**name=**nm**,** **email=**em**,** **address=**ad**,** **passward=**pw)

User**.**save()

* Creates a **new row** in database table Profilee.
* save() is Django’s ORM method to insert into DB.

### 2. **UPDATE (Modify existing record)**

User **=** Profilee(**id=**1**,** **name=**nm**,** **email=**em**,** **address=**ad**,** **passward=**pw)

User**.**save()

* If id=1 already exists → updates that row.
* If id=1 does NOT exist → Django will create new record with id=1.
* This is because save() works as **insert or update (upsert)**.

👉 Another way (better way to update):

User **=** Profilee**.**objects**.**get(**id=**1)   *# Fetch user with id=1*

User**.**name **=** nm

User**.**email **=** em

User**.**address **=** ad

User**.**passward **=** pw

User**.**save()

### 3. **DELETE (Remove record)**

User **=** Profilee(**id=**1)  *# record with id=1*

User**.**delete()

* Deletes the record with id=1.

👉 Another way (more common):

User **=** Profilee**.**objects**.**get(**id=**1)

User**.**delete()

# 📖 Theoretical Points (Don’t Miss These)

1. **Model = Table** → Each class in models.py represents a database table.
2. **Form = User Input** → Forms take input and validate it.
3. **cleaned\_data** → Extracts validated form input.
4. **save() method** → Works as both INSERT and UPDATE depending on primary key.
5. **delete() method** → Removes record from DB permanently.
6. **HttpResponseRedirect** → Redirects after form submission (avoids resubmission when refreshing).
7. **CSRF Token** → Required for POST requests in Django ({% csrf\_token %}).
8. **ORM (Object Relational Mapper)** → Django lets you work with DB using Python objects, no need to write raw SQL.

# ✅ What You Missed (and Should Know Next)

* You’re using forms.Form. In real projects, we mostly use **ModelForm**, which automatically connects to Models and reduces code.
* Example:

**from** django**.**forms **import** ModelForm

**from** student**.**models **import** Profilee

class **ProfileForm**(*ModelForm*)**:**

    class **Meta:**

        model **=** Profilee

        fields **=** ['name'**,** 'email'**,** 'address'**,** 'passward']

Now you don’t need cleaned\_data → just form.save() will insert into DB.

# 🎯 Summary

* You wrote correct code for **Save, Update, Delete** using .save() and .delete().
* You learned difference between forms.Form and forms.ModelForm.
* CRUD is the backbone of every web application in Django.

in detail