Django ORM

Django ORM (Model)	Description	SQL Equivalent	DB Table Column
models.Model	Base class for all models	CREATE TABLE	Table: employee_employee
name = models.CharField(max_length=100)	Text field with max length 100	VARCHAR(100)	name column
age = models.IntegerField()	Integer field for age	INTEGER	age column
department = models.CharField(max_length=100)	Text field for department	VARCHAR(100)	department column
email = models.EmailField()	Validated email address field	VARCHAR(254)	email column
Employee.objects.all()	Fetch all employee records	SELECT * FROM employee_employee;	—
Employee.objects.create(...)	Insert new record	INSERT INTO ...	—
Employee.objects.get(id=1)	Get single record by ID	SELECT * FROM ... WHERE id=1	—
Employee.objects.filter(age=25)	Get filtered records	SELECT * FROM ... WHERE age=25	—
emp.delete()	Delete a record	DELETE FROM ... WHERE ...	—
emp.save()	Save/update a record	UPDATE ... SET ...	—

❓ FAQs:

Question	Answer
How do I insert employee data?	Via Django Admin Panel /admin/
Where is the homepage?	At / with a simple welcome page
How do I get data as JSON?	Go to /api/employees/
Can I loop through employee data in HTML?	Yes, use Django template tags
Can I change the database to PostgreSQL/MySQL?	Yes, update DATABASES in settings.py
Do I need to create tables manually?	No, Django handles via migrate
Is this a real-world example?	Yes, suitable for portfolio or LinkedIn post
Can I add more fields like joining date or salary?	Yes, extend the model easily

Q.Difference between makemigrations and migrate?

'makemigrations' is responsible to generate SQL code for python model class whereas 'migrate' is responsible to execute the SQL code so that tables will be created in the database.

🎯 Employee Management Mini Project

➤ Admin Panel to insert data

➤ Views to display data in HTML

➤ API to serve JSON response

➤ Home Page

```
# Step 1: Create project & app
django-admin startproject emp_project
cd emp_project
python manage.py startapp employee

# Step 2: Apply migrations
python manage.py makemigrations
python manage.py migrate

# Step 3: Create superuser for admin panel
python manage.py createsuperuser

# Step 4: Run the server
python manage.py runserver
```

models.py

```
# employee/models.py

from django.db import models

class Employee(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    department = models.CharField(max_length=100)
    email = models.EmailField()

    def __str__(self):
        return self.name
```

admin.py

```python
# employee/admin.py

from django.contrib import admin
from .models import Employee

admin.site.register(Employee)
```

views.py

```python
# employee/views.py

from django.shortcuts import render
from django.http import JsonResponse
from .models import Employee

def home(request):
    return render(request, 'home.html')

def employee_list(request):
    employees = Employee.objects.all()
    return render(request, 'employee_list.html', {'employees': employees})

def employee_api(request):
    employees = Employee.objects.all().values('id', 'name', 'age',
'department', 'email')
    return JsonResponse(list(employees), safe=False)
```

```python
# employee/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('employees/', views.employee_list, name='employee_list'),
    path('api/employees/', views.employee_api, name='employee_api'),
]
```

emp_project/urls.py

```python
# emp_project/urls.py

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('employee.urls')),
]
```

```html
<!-- employee/templates/home.html -->

<!DOCTYPE html>
<html>
<head>
    <title>Home - Employee Portal</title>
</head>
<body>
    <h2>Welcome to the Employee Management System</h2>
    <p><a href="/employees/">View Employees</a></p>
```

```
    <p><a href="/api/employees/">View Employees JSON</a></p>
</body>
</html>
```

```
<!-- employee/templates/employee_list.html -->

<!DOCTYPE html>
<html>
<head>
    <title>Employee List</title>
</head>
<body>
    <h2>Employee Data</h2>
    <table border="1" cellpadding="10">
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Age</th>
            <th>Department</th>
            <th>Email</th>
        </tr>
        {% for emp in employees %}
        <tr>
            <td>{{ emp.id }}</td>
            <td>{{ emp.name }}</td>
            <td>{{ emp.age }}</td>
            <td>{{ emp.department }}</td>
            <td>{{ emp.email }}</td>
        </tr>
        {% endfor %}
    </table>
```

```
</body>
</html>
```