Django Aiven Database Connection Using Templates

1. Introduction

Aiven is a cloud database platform that provides managed PostgreSQL, MySQL, and other services. Django can easily connect to an Aiven database using the standard DATABASES setting. This document explains how to configure Django with Aiven, build templates, and run queries.

2. Aiven Database Setup

Steps to get database credentials from Aiven:

1. 1. Login to Aiven dashboard
2. 2. Create PostgreSQL/MySQL service
3. 3. Go to 'Connection Information'
4. 4. Copy host, port, username, password, database name

Diagram: Basic Django ↔ Aiven Flow

```
┌──────────────┐   ┌────────────────┐
│  Django   │ <----> │  Aiven DB  │
└──────────────┘   └────────────────┘
```

Requirements.text

```
asgiref==3.10.0

Django==5.2.8

gunicorn==23.0.0      #  pip install gunicorn

packaging==25.0

psycopg2-binary==2.9.11

sqlparse==0.5.3

tzdata==2025.2

dj-database-url==3.0.1

mysqlclient==2.2.7
```

start.sh

start.sh file

=======

```
#!/bin/bash

python manage.py collectstatic --noinput

python manage.py migrate --noinput

gunicorn trinity_clg.wsgi:application --bind 0.0.0.0:$PORT
```

---

3. Django Database Configuration

```
import os

if os.environ.get("RENDER"):

  # Production (Render + Railway MySQL)

  DATABASES = {

    "default": {

      "ENGINE": "django.db.backends.mysql",

      "NAME": os.environ.get("MYSQLDATABASE"),

      "USER": os.environ.get("MYSQLUSER"),

      "PASSWORD": os.environ.get("MYSQLPASSWORD"),

      "HOST": os.environ.get("MYSQLHOST"),

      "PORT": os.environ.get("MYSQLPORT", "3306"),

    }

  }
```

Staticfiles

```
STATIC_URL = "/static/"

STATIC_ROOT = os.path.join(BASE_DIR, "staticfiles")
```

4. Create App and Model

Example model:

```
from django.db import models

class Student(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()
    city = models.CharField(max_length=100)

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

---

5. View for Fetching Data

```
from django.shortcuts import render
from .models import Student

def show_students(request):
    data = Student.objects.all()
    return render(request, "students.html", {"students": data})
```

---

6. Template (students.html)

```
<!DOCTYPE html>
<html>
<head>
    <title>Students Data</title>
</head>
<body>
    <h2>All Students</h2>
```

```
  <table border="1">
    <tr>
      <th>Name</th>
      <th>Email</th>
      <th>City</th>
    </tr>
    {% for s in students %}
    <tr>
      <td>{{ s.name }}</td>
      <td>{{ s.email }}</td>
      <td>{{ s.city }}</td>
    </tr>
    {% endfor %}
  </table>
</body>
</html>
```

---

7. URL Configuration

```
from django.urls import path
from .views import show_students

urlpatterns = [
  path('students/', show_students),
]
```

---

8. Run Migrations

Run commands:

5. python manage.py makemigrations
6. python manage.py migrate
7. python manage.py runserver