SQL Database table:

Table name: Employee

Column	Type	Description
Employee Code	VarChar	The unique employee code that identifies an employee. For example 'ABC1234'
Department	VarChar	The department to which the employee belongs like 'Engineering, HR, Operations' etc
Score	Integer	A numerical score given to an employee.
Date Created	Datetime	Date and time the employee record was created in the database.

Initial Condition:

There are one million employees or records in the Employee Table.

Problem Statement/Requirement:

Create a Django or any other python web framework application which will do the following:

1. Provide a JSON API with the following URL: /employees/?chunk=<chunk number> . where 'chunk' is a query parameter and 'chunk number' is a positive non zero integer starting from 1.
2. When the API is called without any query parameter using HTTP GET, following steps will be followed in the following order:
    a. Application will query the database to fetch the employee records.
    b. Application will then return the results/employees sorted in descending order of score in JSON response. JSON response will contain an array of employee objects, for example:
    { 'employees':
    [{'employee_code': 'E1', 'department':'D1', 'score':100},
    {'employee_code':'E2', 'department': 'D2', 'score':89}.....]
    }
    c. In the employees array, every 5th and 6th indices will contain employees from specific department – 'Waltzz' and every 7th and 8th indices will contain employees created within last 14 days irrespective of their scores. For example, in the response below, the employee objects highlighted in yellow are from department 'Waltzz', the employee objects highlighted in green were created within last 14 days starting today:
    { 'employees':
        [{'employee_code': 'E10', 'department':'D1', 'score':100},
        {'employee_code':'E1', 'department': 'D2', 'score':89},
        {'employee_code':'E8', 'department': 'D2', 'score':88},
        {'employee_code':'E23', 'department': 'D2', 'score':88},