ASP.NET Core and SQL Tasks

Problem 1

To-Do List update API (Estimated time: 45 mins)

Write the API code to make appropriate changes to the to-do list records in the database based on the description with service and repository classes

Request

HttpMethod	Path	Description
PUT	**/api/to-do**	• Insert **if the record is new,** *(Guid is null not provided)* • Update **if the record exists in the database** • Delete **if the record is missing in the body** but **exists in the database**

Body

A list of objects in the below template **ToDo**

```
[
  {
    "Guid": "b37d8e0a-6d45-4d0e-9cb2-2f586b4f9a78",
    "Name": "Finish Project Report",
    "Description": "Compile all the data and create the final project report",
    "Completed": false
  },
  {
    "Guid": "7c32083b-3ae0-482f-a156-9e12aefab08d",
    "Name": "Prepare Presentation",
    "Description": "Create a compelling presentation for the project",
    "Completed": true
  },
  {
    "Guid": null,
    "Name": "Review Code",
    "Description": "Review and optimize the existing codebase",
    "Completed": false
  },
  {
    "Guid": "e2154ae8-cce0-43af-96ed-9cc9a02de731",
    "Name": "Test Application",
    "Description": "Perform thorough testing on the application",
    "Completed": false
  }
]
```

Property Name	Property Type	Description
Guid	Guid(Nullable)	A unique Identifier generated by database on insertion, *[Required for edit and delete records]*
Name	String	Name of the task, *[Required]*
Description	String	Description of the task
Completed	Boolean	Status of the task, True if completed. Default False. *[Required]*

Response

Status	Body	Description
200	Success	Request completed successfully
400	Bad Request	Any invalid data
500	Internal Server Error	Any exception thrown

Problem 2

Fetch records after combining two tables in a proportion

There are two tables in the database

```
-- CREATE TABLE Posts
CREATE TABLE Posts (
  Id BIGINT IDENTITY PRIMARY KEY NOT NULL,
  Content VARCHAR(MAX) NOT NULL
);

-- INSERT INTO Posts
INSERT INTO Posts (Content)
VALUES ('Content of post 1'),  ('Content of post 2'),  ('Content of post 3'),
  ('Content of post 4'),  ('Content of post 5'),  ('Content of post 6'),
  ('Content of post 7'),  ('Content of post 8'),  ('Content of post 9'),
  ('Content of post 10'),  ('Content of post 11'),  ('Content of post 12'),
  ('Content of post 13');

-- CREATE TABLE Ads
CREATE TABLE Ads (
  Id BIGINT IDENTITY PRIMARY KEY NOT NULL,
  Content VARCHAR(MAX) NOT NULL
);

-- INSERT INTO Ads
INSERT INTO Ads (Content)
VALUES  ('Content of ad 1'),  ('Content of ad 2'),  ('Content of ad 3'),
  ('Content of ad 4'),  ('Content of ad 5');
```

Fetch records after combining the tables in a proportion of 1:5 and apply ORDER BY the column Id and OFFSET from start and limit. i.e., 1 row from Ads table after each 5 rows from Posts table.

(Hint: You can use union with some aggregate function)

Sample output

Id	Type	Content
1	post	Content of post 1
2	post	Content of post 2
3	post	Content of post 3
4	post	Content of post 4
5	post	Content of post 5
1	ad	Content of ad 1
6	post	Content of post 6
7	post	Content of post 7
8	post	Content of post 8
9	post	Content of post 9
10	post	Content of post 10
2	ad	Content of ad 2
...	...	...

Problem 3

Fetch records based on average value (Estimated time: 20 mins)

Write a query to find the average salary of employees in each department along with the department name from the database below.

There are two tables in the database

```
CREATE TABLE departments (
  department_id BIGINT PRIMARY KEY NOT NULL,
  department_name varchar(256)
);

--Insert data into the 'departments' table
INSERT INTO departments (department_id, department_name)
VALUES  (1, 'Engineering'),  (2, 'Sales'),  (3, 'Marketing');

-- Create the 'employees' table
CREATE TABLE employees (
  emp_id BIGINT PRIMARY KEY NOT NULL,
  emp_name varchar(256),
  department_id integer,
  salary numeric
);

-- Insert data into the 'employees' table
INSERT INTO employees (emp_id, emp_name, department_id, salary)
VALUES  (1, 'John Doe', 1, 5500.00),  (2, 'Jane Smith', 1, 4500.00),
(3, 'Michael Johnson', 2, 4000.00), (4, 'Emily Brown', 2, 5000.00),
(5, 'David Lee', 3, 3500.00),  (6, 'Sarah Davis', 3, 4500.00);
```

Sample output

department_name	average_salary
Engineering	5000
Sales	4500
Marketing	4000

Problem 4

Fetch records based on the aggregate value from another table (Estimated time: 20 mins)

Write a SQL query to retrieve the names of all departments along with count of employees having two or more employees.

There are two tables in the database

```
-- Create the 'departments' table
CREATE TABLE departments (
department_id BIGINT PRIMARY KEY,
department_name varchar(256)
);

-- Insert data into the 'departments' table
INSERT INTO departments (department_id, department_name)
VALUES (1, 'Engineering'), (2, 'Sales'), (3, 'Marketing');

-- Create the 'employees' table
CREATE TABLE employees (
emp_id BIGINT PRIMARY KEY,
emp_name varchar(256),
department_id integer
);

-- Insert data into the 'employees' table
INSERT INTO employees (emp_id, emp_name, department_id)
INSERT INTO Employees (emp_id, emp_name, department_id)
VALUES(1, 'John Smith', 1), (2, 'Jane Doe', 2), (3, 'Mark Jones', 1),
 (4, 'Emily Brown', 2), (5, 'Alex Ford', 3);
```

Sample output

department_name	employee count
Engineering	2
Sales	2