Student ID: ____

Full Names: ____

Enterprise Backend Web Developer Test

(June 2023)

Author: Professor Obinna Kalu, MSCS

1. The time allotted for completing this test is $2^1/_2$ hours.
2. You are expected to use your Computer with an IDE or any Code Editor tool of your choice to implement your solution for the question.
3. *For the tasks in the question, you are expected to take screenshot(s) of your result(s), save each into a .png or .jpg image file, placed inside a folder named, screenshots and include these in your submission, making sure to include all your project source code, pushed to a repository on your Github account. For the given question, when you have completed your own solution, you are required to take each of the set of 6 evidential sample screenshots, which have been included at the end of the question.*
4. Upon completion, to submit your work for review and grading, simply push your entire Project Source Code folder (including the screenshots folder) to a repository on your Github account. And send the repository's URL using Microsoft Teams chat to Professor Kalu (okalu@miu.edu).
5. **This Career Center test belongs to MIU CS Department and must not be taken away, or copied or photographed or reproduced or transferred or shared or distributed**. **Any Violation will be penalized.**

Make sure to include the screenshots of your results, as required.

Enterprise Backend Web Developer Test (100 points)

Evaluating your Software Development/Coding ability:

1. (100 points) **Implementing RESTful Web API for an Enterprise Web Application for a Dental Surgeries company**

Assume that a company, named **MIU Dental Surgeries**, has hired you to design and develop a Web API for their Enterprise software solution, named **Appointments Management System (AMS),** which they will be using to administer their Dental Surgery appointments. Specifically, the system will be used in requesting, booking and managing **appointment**s for their **Patients**. They want you to implement a basic RESTful Web API for this purpose.

An important need for the company Managers, is to be able to view the list of **VIP Appointments. A VIP Appointment** is any appointments in the system that is booked for a patient who is aged 65 years and above.

For the purpose of this Career Center Test, here is a simplified description of the data model for the system:

A **Patient** can have multiple **Appointment**s.
And, an **Appointment** can only be booked for one and only one **Patient**.

IMPORTANT: Your solution model should consist of only these two entity classes:

1. Patient
2. Appointment

Here are the attributes for the entities, including some useful descriptions, requirements and/or sample data values:

Patient:

patientId: integer, (Primary Key field)

patientNumber, (required field) (e.g. P101, EP117 etc.)

firstName, (required field) (e.g. Bob, Anna, Carlos etc.)

lastName, (required field) (e.g. Jones, Smith etc.)

dateOfBirth, (required field) (e.g. 1997-12-18, 1958-03-27 etc.)

Appointment:

appointmentId: long (Primary Key field)

appointmentDate, REQUIRED (e.g. 2014-09-21, 2013-10-19, etc.)

appointmentTime, REQUIRED (e.g. 10:00:00, 14:15:00 etc.)

dentistName: (optional field) (e.g. Mary Long, David Pearson etc.)

surgeryLocation, (optional field)

Data:

Here is the Company's existing data, which you are expected to input into your database:

Patients data:

Patient Id	Patient Number	First Name	Last Name	Date of birth
1	P101	Bob	Jones	1997-12-18
2	EP117	Anna	Smith	1958-03-27

Appointments data:

Appointment Id	Appointment Date	Appointment Time	Dentist Name	Surgery	Patient Id
1	2023-11-15	18:00:00	Dr Robin Plevin		2
2	2023-04-30	14:30:00		D4 Med Plaza	1
3	2023-09-01	10:00:00	Dr Tony Elliot	D4 Med Plaza	2

For this question, you are required to do the following:

Using suitable .NET (or Java) backend tools, technologies and frameworks which you know and/or have mastered from your work experience or studies in your MSCS courses, etc., (or some other Enterprise Web application development tool(s) which you prefer), implement the RESTful Web API backend service for the company's AMS system, paying close attention to the required details. You may use any database of your choice.

For the given 2.5 hours in this Test, you are expected to implement only the following:

1. Implement a RESTful Web API endpoint url which presents the list of all the VIP Appointments in JSON format. The Company requires this list to be displayed sorted in ascending order of the Appointment Date and Time. And they also require the data to include the Patient's information (i.e. Patient Id, Patient Number, First Name, Last Name and Date of Birth). ***Note:*** *See the meaning of* ***VIP Appointment*** *given in the system specification/description above.*

2. Implement a RESTful Web API endpoint url which presents the list of all upcoming Appointments, sorted in ascending order of the Appointment Date and Time. ***Note****: Upcoming Appointments are those that have an Appointment date that falls on or after the current date (i.e. from today and into the future).*

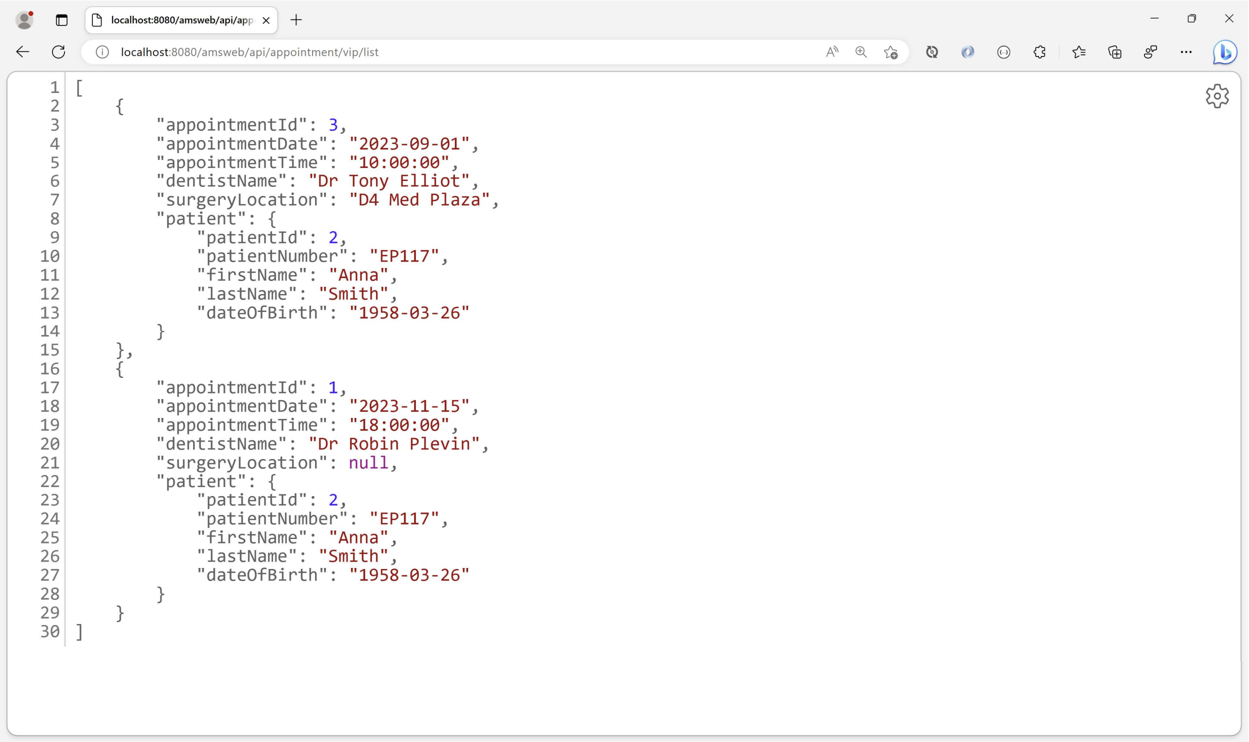
3. Implement a RESTful Web API endpoint url which presents the list of Elderly Patients, in JSON format. **Note:** An Elderly Patient is one who is of age, 65 years or older.

4. Implement a RESTful Web API endpoint url that creates/registers a new Patient in the system, upon receiving the JSON-formatted data submitted to it via an HTTP POST request.

Shown below are sample screenshots and data presentation for the above requirements.

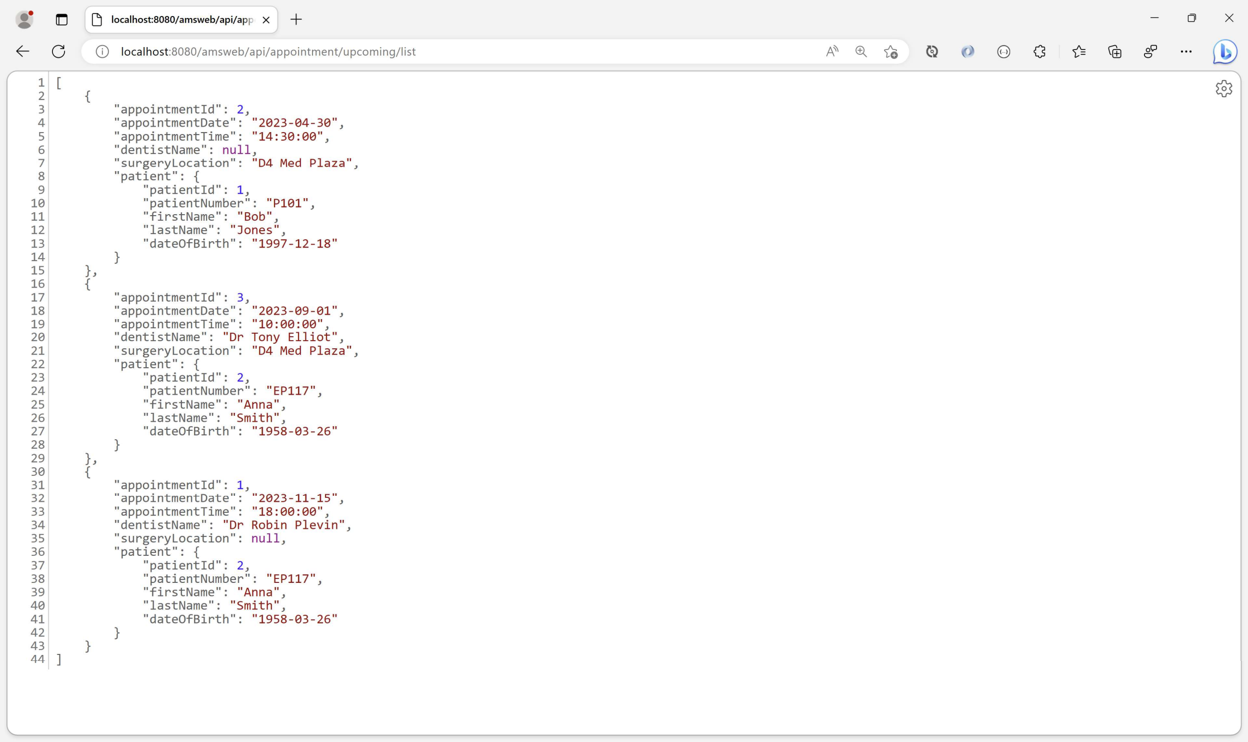
Note: Your own screenshots may be different if you use a different Web API testing tool. However, your screenshots should contain/present similar operations and data and data fields, as required.

JSON-formatted list of all VIP Appointments:
(Note: Sorted in ascending order of their Appointment Date and Time)


localhost:8080/amsweb/api/app
localhost:8080/amsweb/api/appointment/vip/list

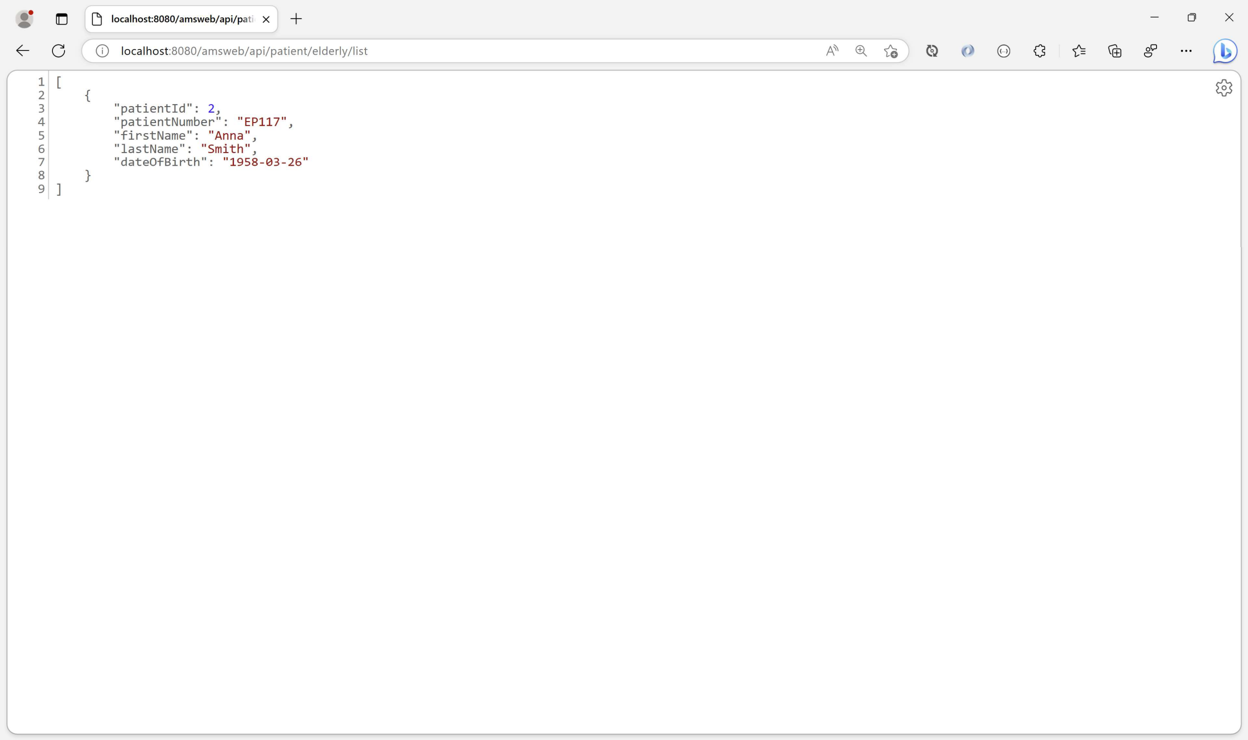
```
[
    {
        "appointmentId": 3,
        "appointmentDate": "2023-09-01",
        "appointmentTime": "10:00:00",
        "dentistName": "Dr Tony Elliot",
        "surgeryLocation": "D4 Med Plaza",
        "patient": {
            "patientId": 2,
            "patientNumber": "EP117",
            "firstName": "Anna",
            "lastName": "Smith",
            "dateOfBirth": "1958-03-26"
        }
    },
    {
        "appointmentId": 1,
        "appointmentDate": "2023-11-15",
        "appointmentTime": "18:00:00",
        "dentistName": "Dr Robin Plevin",
        "surgeryLocation": null,
        "patient": {
            "patientId": 2,
            "patientNumber": "EP117",
            "firstName": "Anna",
            "lastName": "Smith",
            "dateOfBirth": "1958-03-26"
        }
    }
]
```

JSON-formatted list of all Upcoming Appointments:
(Note: Sorted in ascending order of their Appointment Date and Time)


localhost:8080/amsweb/api/app
localhost:8080/amsweb/api/appointment/upcoming/list

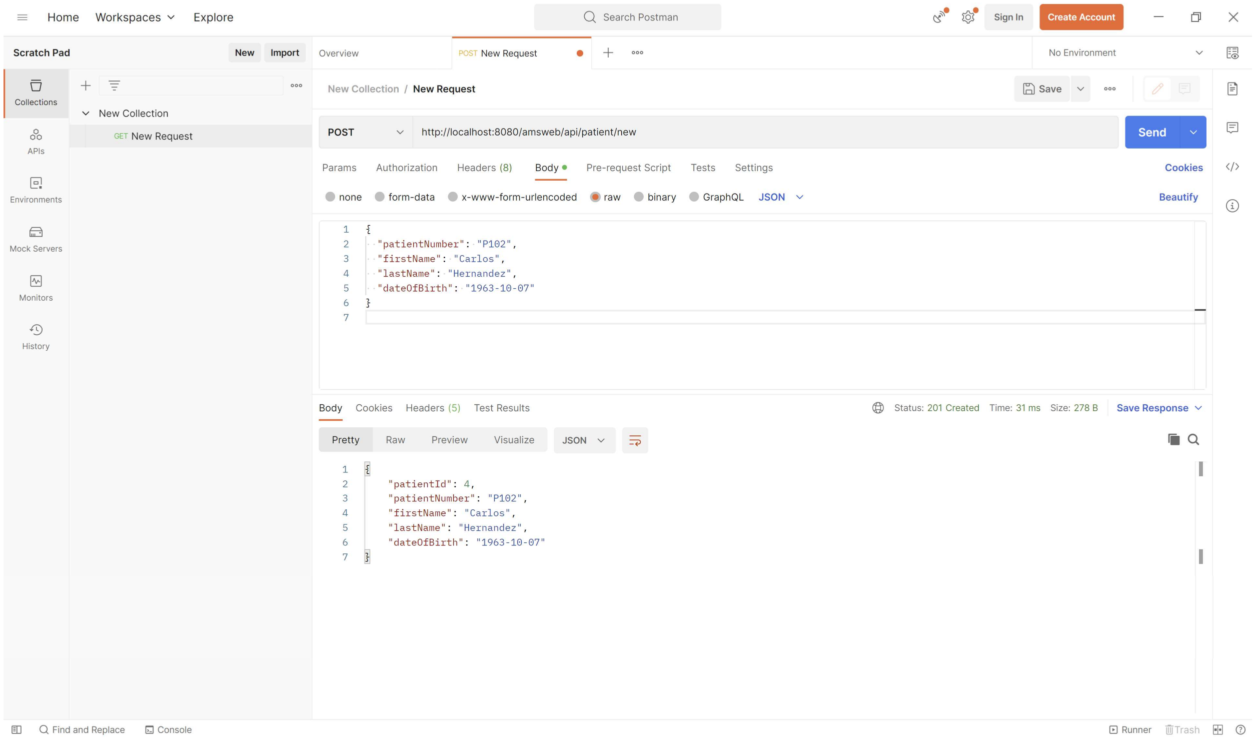
```
[
    {
        "appointmentId": 2,
        "appointmentDate": "2023-04-30",
        "appointmentTime": "14:30:00",
        "dentistName": null,
        "surgeryLocation": "D4 Med Plaza",
        "patient": {
            "patientId": 1,
            "patientNumber": "P101",
            "firstName": "Bob",
            "lastName": "Jones",
            "dateOfBirth": "1997-12-18"
        }
    },
    {
        "appointmentId": 3,
        "appointmentDate": "2023-09-01",
        "appointmentTime": "10:00:00",
        "dentistName": "Dr Tony Elliot",
        "surgeryLocation": "D4 Med Plaza",
        "patient": {
            "patientId": 2,
            "patientNumber": "EP117",
            "firstName": "Anna",
            "lastName": "Smith",
            "dateOfBirth": "1958-03-26"
        }
    },
    {
        "appointmentId": 1,
        "appointmentDate": "2023-11-15",
        "appointmentTime": "18:00:00",
        "dentistName": "Dr Robin Plevin",
        "surgeryLocation": null,
        "patient": {
            "patientId": 2,
            "patientNumber": "EP117",
            "firstName": "Anna",
            "lastName": "Smith",
            "dateOfBirth": "1958-03-26"
        }
    }
]
```

JSON-formatted list of all Elderly Patients (Aged 65 and above):


localhost:8080/amsweb/api/pati
localhost:8080/amsweb/api/patient/elderly/list

```
[
    {
        "patientId": 2,
        "patientNumber": "EP117",
        "firstName": "Anna",
        "lastName": "Smith",
        "dateOfBirth": "1958-03-26"
    }
]
```

RESTful (Web) API endpoint url for Register new Patient:



POST http://localhost:8080/amsweb/api/patient/new

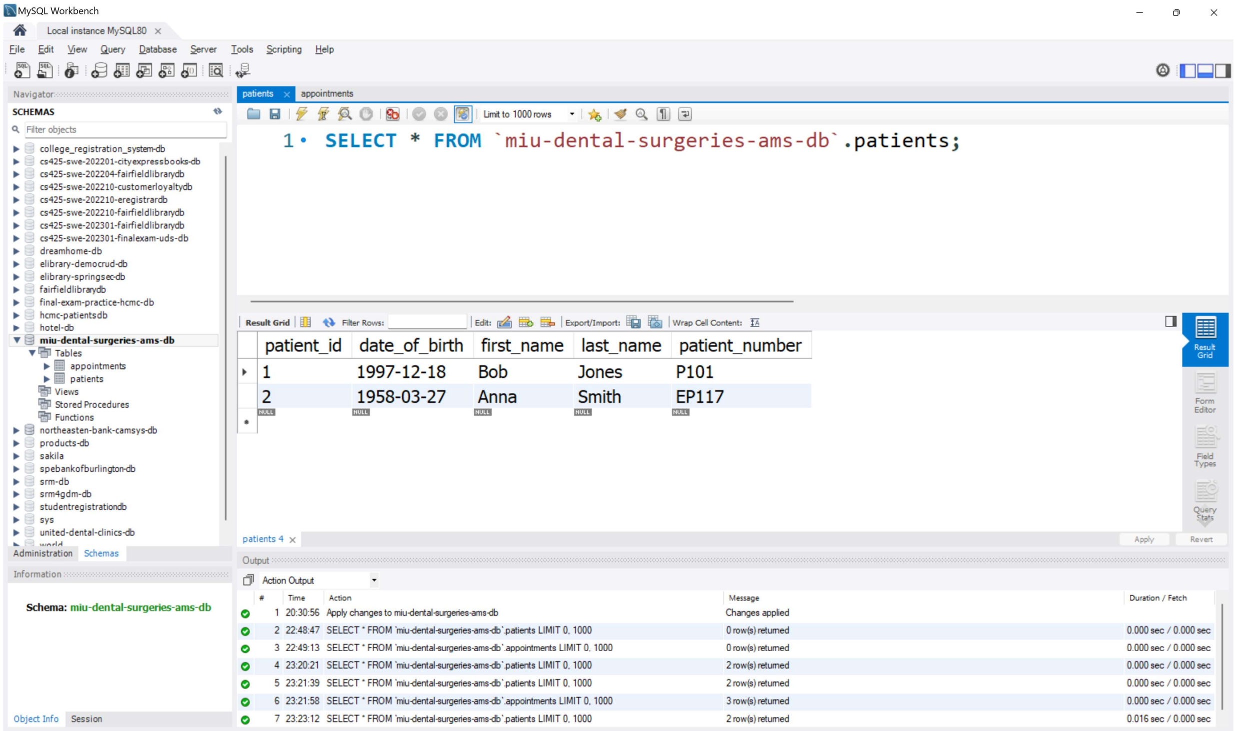
```
{
  "patientNumber": "P102",
  "firstName": "Carlos",
  "lastName": "Hernandez",
  "dateOfBirth": "1963-10-07"
}
```

Status: 201 Created Time: 31 ms Size: 278 B

```
{
    "patientId": 4,
    "patientNumber": "P102",
    "firstName": "Carlos",
    "lastName": "Hernandez",
    "dateOfBirth": "1963-10-07"
}
```

Database Tables screenshot (take a screenshot of your database tables, similar to the two pasted below):

Patients table

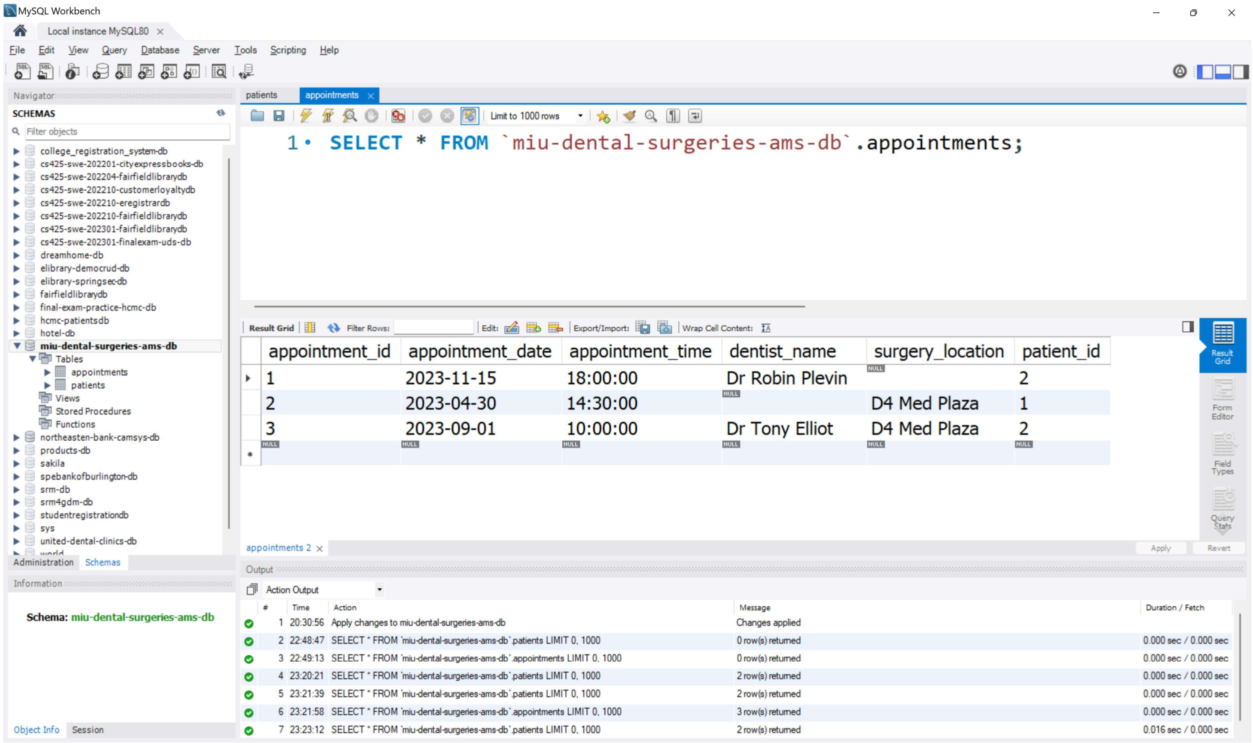


```
1 •  SELECT * FROM `miu-dental-surgeries-ams-db`.patients;
```

patient_id	date_of_birth	first_name	last_name	patient_number
1	1997-12-18	Bob	Jones	P101
2	1958-03-27	Anna	Smith	EP117
NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                                             Message               Duration / Fetch
1  20:30:56  Apply changes to miu-dental-surgeries-ams-db                       Changes applied
2  22:48:47  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000       0 row(s) returned     0.000 sec / 0.000 sec
3  22:49:13  SELECT * FROM 'miu-dental-surgeries-ams-db`.appointments LIMIT 0, 1000   0 row(s) returned     0.000 sec / 0.000 sec
4  23:20:21  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000       2 row(s) returned     0.000 sec / 0.000 sec
5  23:21:39  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000       2 row(s) returned     0.000 sec / 0.000 sec
6  23:21:58  SELECT * FROM 'miu-dental-surgeries-ams-db`.appointments LIMIT 0, 1000   3 row(s) returned     0.000 sec / 0.000 sec
7  23:23:12  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000       2 row(s) returned     0.016 sec / 0.000 sec
```

Appointments table



Schema: miu-dental-surgeries-ams-db

```
1 • SELECT * FROM `miu-dental-surgeries-ams-db`.appointments;
```

appointment_id	appointment_date	appointment_time	dentist_name	surgery_location	patient_id
1	2023-11-15	18:00:00	Dr Robin Plevin	NULL	2
2	2023-04-30	14:30:00	NULL	D4 Med Plaza	1
3	2023-09-01	10:00:00	Dr Tony Elliot	D4 Med Plaza	2
NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                                                  Message            Duration / Fetch
1  20:30:56  Apply changes to miu-dental-surgeries-ams-db                            Changes applied
2  22:48:47  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000      0 row(s) returned  0.000 sec / 0.000 sec
3  22:49:13  SELECT * FROM 'miu-dental-surgeries-ams-db`.appointments LIMIT 0, 1000  0 row(s) returned  0.000 sec / 0.000 sec
4  23:20:21  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000      2 row(s) returned  0.000 sec / 0.000 sec
5  23:21:39  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000      2 row(s) returned  0.000 sec / 0.000 sec
6  23:21:58  SELECT * FROM 'miu-dental-surgeries-ams-db`.appointments LIMIT 0, 1000  3 row(s) returned  0.000 sec / 0.000 sec
7  23:23:12  SELECT * FROM 'miu-dental-surgeries-ams-db`.patients LIMIT 0, 1000      2 row(s) returned  0.016 sec / 0.000 sec
```

//-- The End --//