**Database Description for Schedule App:**

The database for the "Schedule app" it will be designed to efficiently manage the information required for scheduling and maintaining course details for professors and courses. It consists of four main tables that will need to solve the complexity of the conflict between the parameters, the table below shows the details of tables in our dataset and each column and type:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Professor Table** |  |  |  |
|  | **Column name** | **Description** | **Variable Type** | **Info** |
|  | **Professor\_ID** | A unique identifier for each professor. | *INT* | Primary Key, NOT NULL |
|  | **UDM\_ID** | Professor ID from UDMercy | *NVARCHAR(100)* | NOT NULL |
|  | **Fname** | First name of the professor | *NVARCHAR(50)* |  |
|  | **Lname** | Last name of the professor | *NVARCHAR(50)* |  |
|  | **Professor\_Type** | full-time or adjunct | *BOOLEAN* | Full-time higher priority |
|  | **Department** | The department to which the professor belongs (EE, CE, Robotic) | *LIST* |  |
|  | **Seniority** | The seniority level of the professor | *INT* | Years of experience in the teaching field |
|  | **Courses\_Taught** | A list of courses taught by the professor | *NVARCHAR(5)* |  |
|  | **Email** | Active email (UDMercy mail) for professor | *NVARCHAR(50)* |  |
|  | **Office\_Num** | Office number or location of the professor | *NVARCHAR(50)* |  |
|  | **Phone** | Active email | *INT* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Course Table** |  |  |  |
|  | **Column name** | **Description** | **Variable type** | **Info** |
|  | **Course\_ID** | A unique identifier for each course. | *INT* | Primary Key |
|  | **Department (Major)** | List of three majors | *NVARCHAR(5)* | EE, CE, Robotic |
|  | **Course\_Name** | Course name and number from UDM | *NVARCHAR(50)* | e.g. MTH1410 |
|  | **Course\_Type** | In-person, Online (Synchronous), or (Asynchronous) | *LIST* | Maby two types are available for particular course. |
|  | **Elective** | Yes or no | *BOOLEAN* | If yes, lower priority |
|  | **Course\_Days** | Number of days for a particular course in a week (e.g. Java two days in a week) | *INT* |  |
|  | **Course\_Duration** | Course duration of an hour and a half need six time slots | *LIST* |  |
|  | **Course\_Level (Student level)** | List of course levels which is students levels: (Freshman, Sophomore, Junior, Senior, Graduate) | *LIST* | Some courses are available to multiple levels, and the Admin is able to select more than one level for a specific course.  E.g ML 5350 |
|  | **Course\_Credit** |  | *DECIMAL* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Time Slot Table** | Fixed slots each 15 min 54 slots daily X 5 days = 270 slots for workdays |  |  |
|  | **Column name** | **Description** | **Variable type** | **Info** |
|  | **Time\_Slot\_ID** | A unique identifier for each time slot | *INT* | Primary Key |
|  | **Start\_Time** | DateTime (YYYYMMDD hh:mm format) - Represents the start time of the time slot for the particular day. | *DATETIME* |  |
|  | **End\_Time** | DateTime (YYYYMMDD hh:mm format) - Represents the end time of the time slot for the particular day. | *DATETIME* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Availability Table** | This table is the input for the algorithm |  |  |
|  | **Column name** | **Description** | **Variable type** | **Info** |
|  | **Ava\_ID** | A unique identifier for each availability record | *INT* | Primary Key |
|  | **Prof\_ID** | Foreign key reference to the Professor Table for linking the availability to a specific professor | *INT* | Foreign key |
|  | **Cour\_ID** | Foreign key reference to the Course Table to associate availability with specific courses. | *INT* | Foreign key |
|  | **TSlot\_ID** | Foreign Key Reference to Time Slot Table | *INT* | Foreign key |
|  | **Availability\_Type** | online or in-person(Asynchronous) or (Synchronous) | *LIST* |  |
|  | **Need\_Lab** |  | *BOOLEAN* | In-person only |
|  | **IsAva** | Binary and dynamic change each iteration depends on the available slot or not | *BOOLEAN* | For any slot value 0, the algorithm will ignore it |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Admin Table** |  |  |  |
|  | **Column name** | **Description** | **Variable type** | **Info** |
|  | **Admin\_ID** | A unique identifier for each admin | *INT* | Primary Key |
|  | **Fname** |  | *NVARCHAR(50)* |  |
|  | **Lname** |  | *NVARCHAR(50)* |  |
|  | **Password** |  | *TEXT NOT NULL* |  |
|  | **Type** | For future, only Admin now |  |  |

The table below shows the availability as inputs of the algorithm; these rows should be scheduled without conflict with each other. And give priority to particular factors.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ava\_ID** | **Prof\_ID** | **Cour\_ID** | **TSlot\_ID** | **Is\_Ava** | **Availability\_Type** | **Need\_Lab** |
| 1 | Dr. Smith | Java | M, W, F 8:00-12:00, 15:00- 21:00 | 1 | In-Person | Yes |
| 2 | Dr. Ali | Java | Full time | 1 | In-Person | Yes |
| 3 | Dr. Carla | Data Mining | 0 | 1 | On-Line Asynchronous | N/A |
| 4 | Dr. John | Data Science | Full time | 1 | In-Person | No |
| 5 | Dr. John | Network Security | Full time | 1 | In-Person | Yes |

**Tables relationships:**

* **The time slots table contains 270 slots from Mon – Fri, 54 slots in a day, each slot 15 min.**
* **Professor Table to Availability Table (One-to-Many):** each professor can have multiple availability records.
* **Time Slot Table to Availability Table (Many-to-Many**): Multi slots used one availability record. Multiple availability for one slot is also possible (Scenario: when availability 1 and availability 5 are available between 8:00-9:00 slots).
* **Course Table to Availability Table (One-to-Many):** One Course can have multiple availability, but one availability can have only one course.
* **Professor Table to Course Table (Many-to-Many):** One professor can apply to teach multiple courses. Also, two professors compete to teach a specific course.
* **Professor Table to Time Slot Table (Many-to-Many):** One professor can have multiple Time Slots. One slot can have more than one professor (Scenario: Professor X teaches Course Y for Student Z and Professor A teaches Course B for Student C at the same time).
* **Course Table to Time Slot Table (Many -to-Many):** One course needs multiple Time Slots. Also, one slot can have multi courses.

A diagram of a computer

Description automatically generated