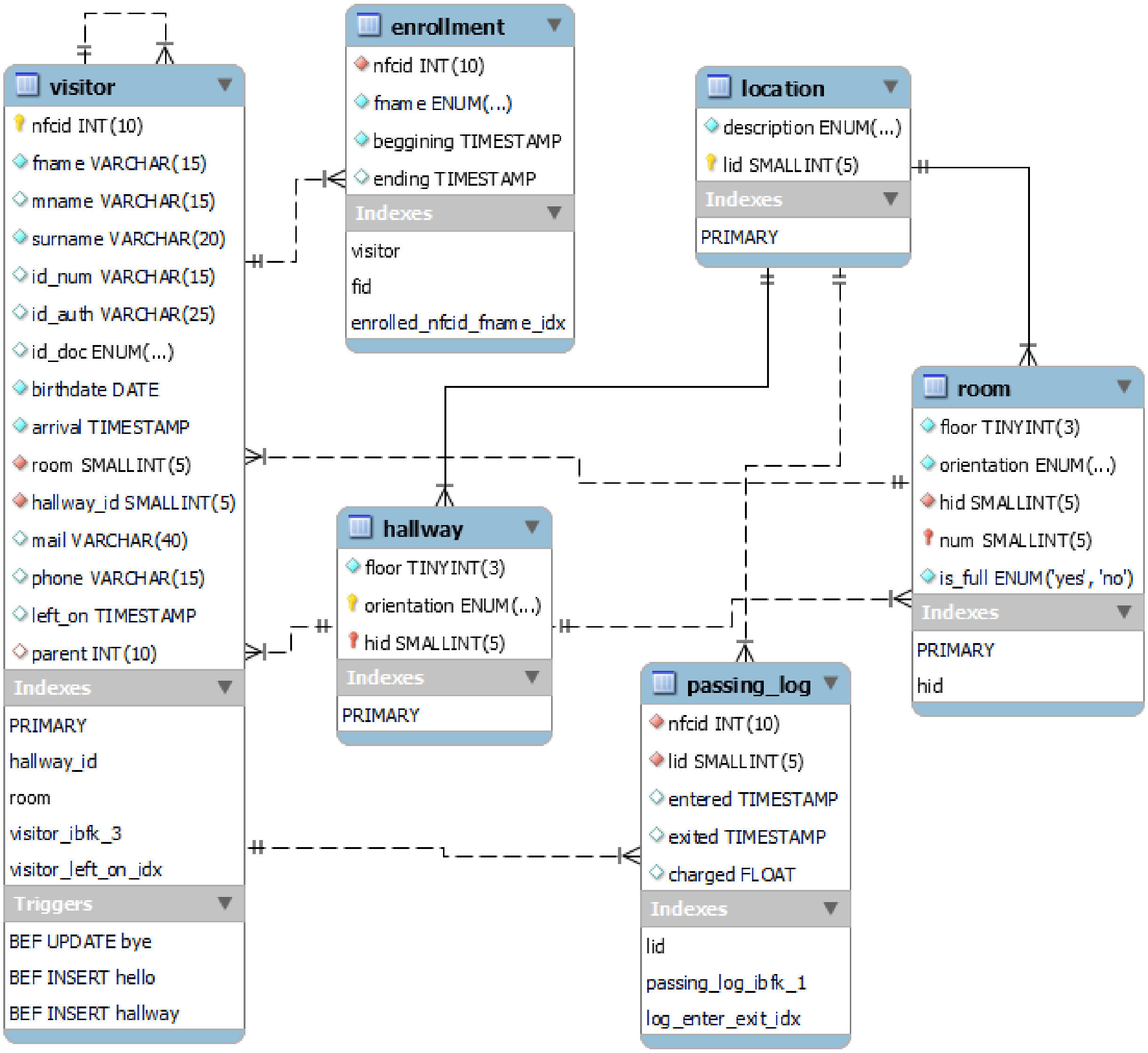
Databases

Team Project

AdditionalgroupT

Tsironis Georgios
03116205

A. Relational Diagram



B. Constraints

Foreign Keys

Name	From	Target	On Delete	On Update
room_ibfk_2	room	hallway (hid→hid)	RESTRICT	RESTRICT
room_ibfk_3	room	location (num→lid)	RESTRICT	RESTRICT
enrollment_ibfk_2	enrollment	visitor (nfcid→nfcid)	CASCADE	CASCADE
hallway_ibfk_1	hallway	location (hid→lid)	RESTRICT	RESTRICT
passing_log_ibfk_1	passing_log	visitor (nfcid→nfcid)	CASCADE	CASCADE
passing_log_ibfk_2	passing_log	location (lid→lid)	RESTRICT	RESTRICT
visitor_ibfk_1	visitor	hallway (hallway_id→hid)	RESTRICT	RESTRICT
visitor_ibfk_2	visitor	room (room→num)	RESTRICT	RESTRICT
visitor_ibfk_3	visitor	visitor (parent→nfcid)	CASCADE	RESTRICT

We connect, using foreign keys, fields of different tables that refer to the same object, in order to make queries faster and more efficient and to limit arbitrary changes to field values that may cause duplication of a field.

Triggers

We also created three triggers in order to automate certain procedures. Two of them automatically set the is_full value of a room to ‘yes’ and ‘no’ when a guest arrives or leaves respectively (we assume that all guests of a room leave together). The third one sets the corridor in which a guest resides based on the room in which they stay, automatically, without requiring the user to enter it.

NOT NULL

We also set certain fields as Not Null in order to prevent the user from leaving required fields empty.

C. Indexes

We created three indexes. The first is on the Visitor table for the left_on field so that it can be immediately determined which visitors are not in the hotel. The second is on the enrollment table for the nfcid and fname fields so that results can be quickly retrieved for the services a visitor has enrolled in and for the visitors who have enrolled in a service. The third is on the passing_log table for the entered and exited fields, because in the case of a **COVID-19 case** we must immediately identify possible cases using the time intervals during which the infected person was present in various rooms.

D. Languages Used

The project was developed on the Windows operating system. For the creation of the database, the MySQL language was used, for the backend of the application we used PHP, and frontend was developed using HTML. Specifically, the WampServer environment was chosen for all of the above.

E. Installation Instructions

The first step before installing the application is the installation of the WampServer environment on the computer that will be used. This will provide MySQL, PHP, as well as the Apache server, that is, everything that will be needed later for the application. The next step is the installation of the database that will be used by the application. This is done by opening a command line and then MySQL with the appropriate credentials and executing the file Database Script.sql, which has been delivered together with this report. Next, the PHP files must be placed in the appropriate folder. In “This PC” directory and then in the OS (C:) subfolder, the folder containing the Wamp environment files will be located, which will have a corresponding name. Inside this folder, the subfolder named “www” will be found, and inside it will be placed all the contents of the PHP folder that was delivered together with this report. One more thing that will be required is that in the “dbconnect.php” file the appropriate parameters must be given to the “mysqli_connect” command so that the application can successfully connect to the database. Afterwards, it is sufficient to search for “localhost/test/” in a browser for the application to open successfully.