**SMMC**

By Raymond Hua and George Alexander

# Use Cases

Administrator

* Has access to the application and the database
* Copies sheet music
* Add persons qualification

Student

* Enrol to SMMC
* View their information
  + Contact information, open division
  + Instrument information and their qualifications
  + Lesson, concert and parents information
  + Payment history
* Creates new enrolment also creates a new person in the database
* If they are between Level 1 and 3 they automatically get enrolled to Training Orchestra
* If they are between Level 4 and 6 they get automatically enrolled to Junior Orchestra
* If they are between Level 7 and 8 they get automatically enrolled to Community Orchestra
* Unenrol themselves
* Update their information
  + Open division, address and qualification information
* Performs at a concert
* Enrol and attend lessons
* Pay their bills and expenses
* If they left school, they can enrol for open division
* If under 18, they would need to have a parent/guardian
  + They can select an existing person from the database or a new one with details to be filled out

Tutors

* Get employed if they have a level 8 qualification
* Update themselves (tutor types)
* Are able to teach lessons
* Get paid on hourly basis (junior, senior and head are paid on separate scales)
* Allocate how many copies of sheet music is given
* Gives out sheet music to students
* View their reports
  + Address information
  + What lessons do they teach
  + For each instrument, what type are they
* Can be head of an instrument for multiple instruments but there can be only one head for one instrument
* Only a senior or head tutor are able to teach lessons at levels 7 or 8.
* Can resign and remove their history in the database

Technician can…

* Handle instrumental repairs
* Modifies persons qualification
* Handles payments

Members of SMMC can…

* Manages in-house demonstration
  + Create, update, view
* Manages major concert in September
  + Create, update, view
* Manages venues
  + Create, update, view
* Manages library of sheet music
  + Manages how many is given to tutors
* Provides instruments except recorder
* Gives out sheet music to tutors
  + Manages how many is given to students

# Triggers

[INSERT\_LessonStudents] – If the instrument exists in StudentInstrument for the student and their qualification level matches the lesson level, it would check the lesson size is less and equal to 8 apart from the recorder, which is 16, then it would add into the database.

[INSERT\_LessonTutors] – If the Tutors and what instrument they teach exists, it would check if the lesson level is 7 or 8 if they are, it would check to see if their type is a Senior or a Head tutor, if so it would add the inserted information into the database. If it does exists and not in those levels then it would add the inserted into the database.

[INSERT\_Tutors] – If the qualification of the inserted person and instrument is level 8 and they don’t exists then it would add the Tutor and what instrument they play into the database.

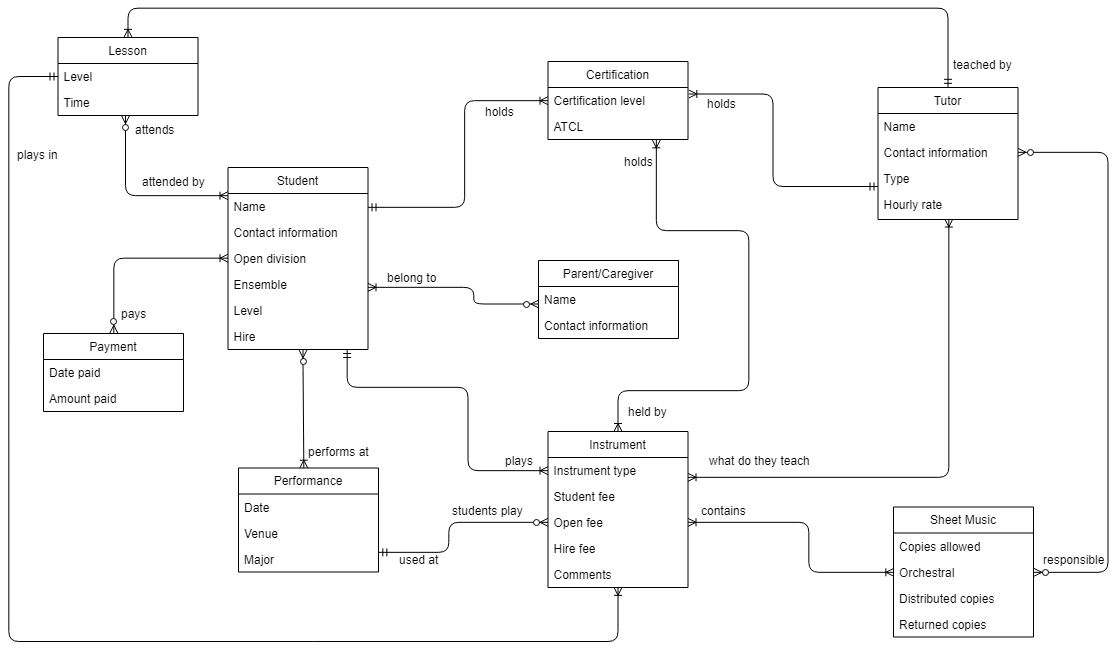
# Stored procedures

DeleteTutors – delete all tutors (and their child tables) by the PersonID.

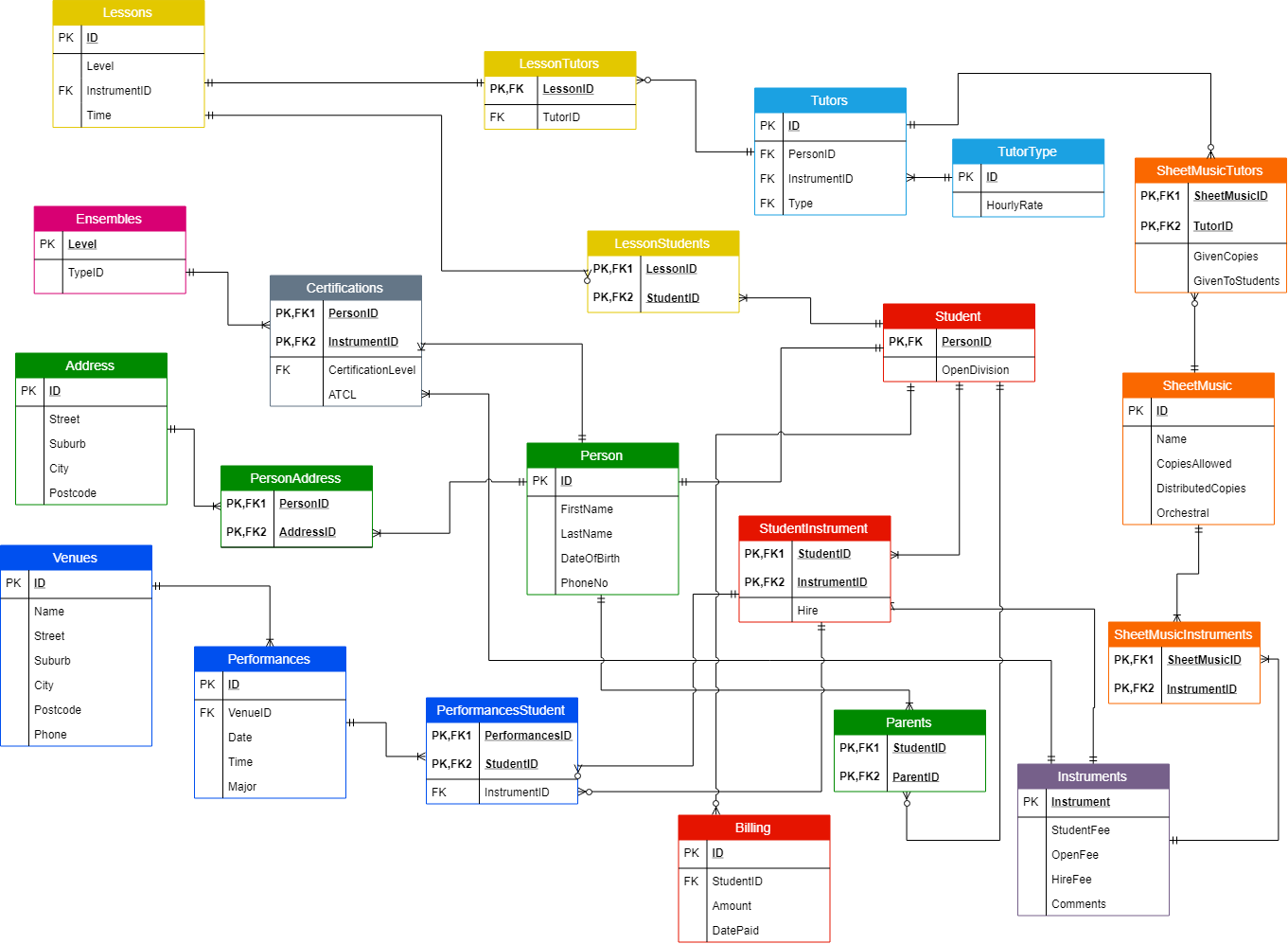
# Business rules

|  |  |
| --- | --- |
| Task | Roles |
| Payroll would be managed by another program. |  |
| The payroll system would be managed by another program. But a report that would shoe all tutors and their weekly pay are. | Administrator |
| The application can only be managed by the administrators, who are the members of the SMMC. | Administrator |
| The application can only be managed by the administrators, who are the members of the SMMC. | Administrator |
| If students want to enrol, they must have an application form and send it along with identification to the administrators for them to enter to the database. | Students, Administrator |
| If tutors or students want to make changes to their programming, they must to talk to the administrators face to face or by phone with their phone number or their date of birth for identification. | Tutors, Students |
| Members can have access to the performances section but must be supervised by the administrator. | Members |
| Repairs are managed by another division and reports them to the paying manager. | Paying manager, Repairs |
| Contact information will always be different for every venue |  |
| All students will start level 0 unless they have a qualification. |  |
| The minimum group size for a lesson is three, otherwise the lesson for the week would be cancelled until they find a third person. |  |
| The maximum group size is 8, except for recorder which is 16. |  |
| Student only play one instrument in a performance |  |
| A lesson can only be taught by one tutor | Tutor |
| Each tutor type will share the same hourly rate | Tutor |
| Adding a new tutor must be an existing person in the database with a certification of level 8 in the instrument |  |
| Adding a new student can either be a new or existing person in the database |  |
| Students who want to play for instruments would need to be in a different section |  |
| A parent can be existing or new to the database |  |

Conceptual Model



# Relational Model



# Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Instruments | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| Instrument (PK) | Int | ID | Unique identifier for each instrument |
| StudentFee | Decimal | Allow | Stores the student fee for a particular instrument |
| OpenFee | Decimal | Allow | Used to store the Open fee for a particular instrument |
| HireFee | Decimal | Allow | Used to store the hire fee for a particular instrument |
| Comments | Varchar | Allow | Notifies the user of important information.  E.g. For recorder the student must bring their own |

|  |  |  |  |
| --- | --- | --- | --- |
| Person | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for each person in the system |
| FirstName | Varchar | Not | Stores the first name of a person |
| LastName | Varchar | Not | Stores the last name of a person |
| DOB | Date | Not | Used to store the date of birth of a person |
| Phone | Varchar | Not | Holds the Phone number of a person in the system |

|  |  |  |  |
| --- | --- | --- | --- |
| Address | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for address |
| Street | Varchar | Not | Holds the name of the street |
| Suburb | Varchar | Not | Holds the name of the suburb |
| City | Varchar | Not | Holds the name of the city |
| Postcode | Varchar | Not | Holds the postcode |

|  |  |  |  |
| --- | --- | --- | --- |
| PersonAddress | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| PersonID (PK) (FK) | Int | ID | Holds the ID of a particular person in the database. Links to the ID in the Person table |
| AddressID(FK) | Int | ID | Holds the Address of a particular person in the database. Links to the ID in the Address table |

|  |  |  |  |
| --- | --- | --- | --- |
| Student | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| PersonID (PK) (FK) | Int | ID | Unique identifier for each student In the database. Links to the ID in the Person table |
| OpenDivision | Bool | Not | Specifies whether or not a student is in the open division. The default value is 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parents | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ParentID (PK)(FK) | Int | ID | Unique identifier for each parent In the database. Links to the ID in the Person table |
| StudentID (PK) (FK) | Int | ID | Links to the ID in the student table |

|  |  |  |  |
| --- | --- | --- | --- |
| Ensembles | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| Level (PK) | Int | ID | Specifies what level the ensemble is for |
| TypeID (FK) | Int | Not | Specifies the type of ensemble.  Links with the ID in the ensembles type table |

|  |  |  |  |
| --- | --- | --- | --- |
| EnsemblesType | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID | Varchar | ID | Specifies the type of ensemble |

|  |  |  |  |
| --- | --- | --- | --- |
| StudentInstrument | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| StudentID (PK) (FK) | Int | ID | Specifies which student is playing what instrument. Links with PersonID the Student table |
| InstrumentID (PK) (FK) | Int | ID | Unique identifier for the instrument. Links with the Instrument in the Instrument table |
| Hire | Bool | Default 0 | Confirms if a person has hired their instrument. If false then the person owns their own instrument and is using that |

|  |  |  |  |
| --- | --- | --- | --- |
| Certifications | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| PersonID (PK) (FK) | Int | ID | Specifies who is associated with what certificate. Links with the ID in the Person table |
| InstrumentID (PK) (FK) | Int | ID | Specifies what instrument is associated with the certificate. Links with the Instrument in the Instrument table |
| CertificationLevel (FK) | Int | Not | Stores the level for which the person is certified e.g. level 7. The default value is 0. Links with the Level in the Ensembles table |
| ATCL | Bool | Default 0 | Confirms if the person has achieved ATCL certification |

|  |  |  |  |
| --- | --- | --- | --- |
| TutorType | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for the tutor type |
| HourlyRate | Varchar | Not | Specifies the hourly rate of the tutor |

|  |  |  |  |
| --- | --- | --- | --- |
| Tutors | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for tutor |
| PersonID (FK) | Int | Not | Foreign key for tutor, links each tutor with the person table where their personal details are stored. Links to the ID in the Person table |
| InstrumentID (FK) | Int | Not | Holds the ID of the instrument the tutor plays. Links with the Instrument in the Instrument table |
| Type | Varchar | Not | Specifies the type of tutor. Links to the ID of the TutorType table |

|  |  |  |  |
| --- | --- | --- | --- |
| Lessons | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for lesson |
| Level | Int | Not | Holds the level at which the lesson is being taught e.g. level 3 |
| Time | Date | Not | Specifies the time in which the lesson will be held |
| InstrumentID (FK) | Int | Not | Specifies the type of instrument the lesson will focus on. Links with the Instrument in the Instrument table |

|  |  |  |  |
| --- | --- | --- | --- |
| LessonTutors | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| LessonID (PK) | Int | ID | Holds the ID of the lesson a particular tutor is taking. Links with the ID in the Lessons table |
| TutorID (FK) | Int | Not | ID of the tutor taking the specified lesson. Links with the ID in the Tutor table |

|  |  |  |  |
| --- | --- | --- | --- |
| LessonStudents | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| LessonID (PK) (FK) | Int | ID | Specifies which lesson the student is taking. Links with the ID in the Lessons table |
| StudentID (PK) (FK) | Int | ID | Specifies what student is taking what lesson. Links with the ID in the Student table |

|  |  |  |  |
| --- | --- | --- | --- |
| SheetMusic | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique Identifier for the sheet of music |
| Name | Varchar | Not | The name of the particular sheet of music |
| CopiesAllowed | Int | Default 0 | The number of copies allowed to be issued |
| DistributedCopies | Int | Default 0 | The number of copies distributed |
| Orchestral | Char | Default 0 | Specifies whether or not the sheet of music is orchestral or not |

|  |  |  |  |
| --- | --- | --- | --- |
| SheetMusicTutors | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| SheetMusicID (PK) (FK) | int | ID | Unique identifier for a particular sheet of music. Links with the ID in the SheetMusic table |
| TutorID (PK) (FK) | Int | ID | Unique identifier for Tutors. Links with the ID in the Tutor table |
| GivenCopies | Int | Not | Holds the number of copies of sheet music allocated to the tutor by the administrator |
| GivenToStudents | Int | Not | Holds the number of copies returned to the tutor by the students |

|  |  |  |  |
| --- | --- | --- | --- |
| SheetMusicInstruments | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| SheetMusicID (PK) (FK) | Int | ID | Unique identifier for a particular sheet of music. Links with the SheetMusic table |
| InstrumentID (PK) (FK) | Int | ID | Associates an instrument with the SheetMusic e.g the music might be for someone playing the trumpet. This is linked from the Instrument table |

|  |  |  |  |
| --- | --- | --- | --- |
| Venues | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for the venue |
| Name | Varchar | Not | Stores the name of the venue |
| Phone | Varchar | Not | Holds the phone number of the venue |
| Postcode | Varchar | Not | Stores the post code associated with the venue |
| City | Varchar | Not | Stores the city in which the venue is located E.g. Dunedin |
| Suburb | Varchar | Not | Stores the suburb the venue is in |
| Street | Varchar | Not | Stores the street name and number of the venue |

|  |  |  |  |
| --- | --- | --- | --- |
| Performances | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique Identifier for each performance |
| Date | Date | Not | Holds the date for when the performance will be performed |
| VenueID (FK) | Int | Not | Links to the Venues table  Used to store the venue at which the performance will be held e.g. Castle Street Lecture Theatre. Links with the ID in the Venue table |
| Major | Bool | Not | Confirms if the concert is a major concert or not |

|  |  |  |  |
| --- | --- | --- | --- |
| PerformancesStudent | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| PerformancesID (PK) (FK) | Int | ID | Unique identifier for Performances. Links with the ID in the Performance table |
| StudentID (PK) (FK) | Int | ID | Unique Identifier for a student. Associates a student with a particular performance. This links with the StudentID in the StudentInstrument table |
| InstrumentID (FK) | Varchar | Not | Links the instrument the student will be playing with the student. E.g. Violin. Links with the Instrument in the StudentInstrument table |

|  |  |  |  |
| --- | --- | --- | --- |
| Payments | | | |
| Field Name | **Data Type** | **Null** | **Description** |
| ID (PK) | Int | ID | Unique identifier for Payments. |
| StudentID (FK) | Int | Not | Unique Identifier for a student. Associates payment with a student. This links with the StudentID in the StudentInstrument table |
| Amount | Decimal | Not | Amount to be paid |
| DatePaid | Date | Default getDate | Date of the payment |