**Сущности**

1. **MusicMan** 
   * MusicianID (int, pk), NOT NULL
   * NameID (int, foreign\_key -> Fml, one to one), NOT NULL
   * GenreID (int, foreign\_key) //
   * Description (nvarchar(4000))
   * DateOfBirth (date)
   * PlaceOfBirth (nvarchar(100))
   * YearsActive (date)
   * InstrumentID (инструменты) //
   * Labels (звукозаписывающая компания) //
   * MemberOf (состоит в) //
   * RelatedGroups (группы с которыми связан) //
   * Albums (альбомы) //
   * Views (отзывы) //
   * AvgMark (int)
2. **FML** 
   * FmlID (int, pk), NOT NULL
   * LastName (nvarchar(40)), NOT NULL
   * Name (nvarchar(20)), NOT NULL
   * MiddleName (nvarchar(20))
   * MusicianID (int, foreign\_key ->, one to one), NOT NULL
3. **Band**
   * BandID (int), NOT NULL
   * Name (nvarchar(30)), NOT NULL
   * Description (nvarchar(4000))
   * Genres //
   * Albums //
   * Origin ((nvarchar(100)))
   * YearsActive(date)
   * Labels //
   * Members //
   * Website (nvarchar(100))
   * Views(int, foreign\_key)
   * Mark (int)
4. **Genres**
   * GenresID (int, pk), NOT NULL
   * Name (nvarchar(30)), NOT NULL
   * Description (nvarchar(4000))
   * Stylistic origins (стилистическое происхождение)//
   * Origins (происхождение) //
   * Derivative forms (производные формы) //
   * Subgenres (поджанры) //
5. **Instrument**
   * InstrumentID (int, pk), NOT NULL
   * Name (nvarchar(20)), NOT NULL
   * Description (nvarchar(4000))
   * Classification (классификация) //
   * Developed (date)
   * Derivative forms (производные формы) //
6. **Album** 
   * AlbumID (int, pk), NOT NULL
   * Name (nvarchar(30)), NOT NULL
   * Description (nvarchar(4000))
   * Released (date)
   * StudioID (int, foreign\_key)
   * Genres //
   * List of comps (список песен)//
   * Length (time)
   * LabelID (int, foreign\_key)
   * Producer (продюсер) //
   * Mark (int)
   * Reviews //
7. **Song**
   * SongID (int, pk), NOT NULL
   * Name (nvarchar(30)), NOT NULL
   * Description (nvarchar(4000))
   * Length (int)
   * Genres //
   * Author //
   * Mark (int)
   * Reviews //
   * AlbumID (int, foreign\_key)
8. **Mark**
   * Value (int), NOT NULL
   * UserID (int, foreign\_key), NOT NULL
   * PublicationID (int, foreign\_key)
   * BadnID (int, foreign\_key)
   * Musician (int, foreign\_key)
9. **Publication**
   * PublicationID(int, pk), NOT NULL
   * Title (nvarchar(30)), NOT NULL
   * Description (nvarchar(4000))
   * Comments //
   * Mark (int)
10. **Review**
    * ReviewID (int, pk), NOT NULL
    * Title (nvarchar(30)), NOT NULL
    * Text (nvarchar(4000))
    * UserID (int, foreign\_key), NOT NULL
    * PublicationID (int, foreign\_key)
    * BadnID (int, foreign\_key)
    * Musician (int, foreign\_key)
    * Mark\_value (int)
11. **Labels**
    * LabelsID (int, pk), NOT NULL
    * Name (nvarchar(30)), NOT NULL
    * Description (nvarchar(2000))
12. **Studio**
    * StudioID (int, pk), NOT NULL
    * Name (nvarchar(30)), NOT NULL
    * Description (nvarchar(2000))
13. **Log Review**
    * ReviewID (int, foreign\_key), NOT NULL
    * UserID (int, foreign\_key), NOT NULL
    * DateReview (datetime), NOT NULL
14. **Log UpdatePublication**
    * LogUpdateID (int, pk), NOT NULL
    * FmlID (int, foreign\_key), NOT NULL
    * Date (datetime), NOT NULL
    * PublicationID (int, foreign\_key), NOT NULL
15. **UserRole**
    * UserID (int, pk), NOT NULL
    * UserDataID (int, foreign\_key), NOT NULL
    * Date (datetime), NOT NULL
    * RoleID (int, foreign\_key), NOT NULL
16. **Role**
    * RoleID (int, pk), NOT NULL
    * Name (nvarchar(10)), NOT NULL
17. **UserData**
    * UserDataID (int, pk), NOT NULL
    * NickName (nvarchar(30), NOT NULL
    * Date (datetime), NOT NULL
    * Description (nvarchar(50))