

## Описание проекта

Предполагается сделать базу для классических музыкальных инструментальных произведений. У каждого произведения есть композитор (иногда их несколько), предполагаемый инструмент для исполнения (иногда их несколько), тональность, форма (“suite”, “mazurka” и проч.), темп (“largo”, “allegro” и проч.) и стиль (“romanticism”, “impressionism” и проч.). Инструмент в произведении может быть использован в количестве, превышающем 1. Одной паре (TempoMinBPM, TempoMaxBPM) может соответствовать несколько темпов.

## Построение отношений

В результате предварительного проектирования были выделены следующие отношения:

- Pieces — содержит список музыкальных произведений.
- Composers — содержит список композиторов музыкальных произведений.
- ComposersPieces — по сути, содержит список пар для композиторов и произведений. Появилось при переходе от модели “сущность-связь” к физической при преобразовании связи “многие-ко-многим” между Pieces и Composers.
- Instruments — содержит список музыкальных инструментов, используемых при исполнении музыкальных произведений.
- InstrumentsPieces — по сути, содержит список пар для инструментов и произведений. Появилось при переходе от модели “сущность-связь” к физической при преобразовании связи “многие-ко-многим” между Pieces и Instruments.
- Tempos — содержит список терминов, указывающих на темп музыкального произведения.
- Forms — содержит список форм музыкальных произведений.
- Styles — содержит список стилей музыкальных произведений.

## Отношение Pieces

Атрибуты:

- ISWC — естественный ключ. Это уникальный идентификатор музыкальных произведений, аналогичный ISBN для книг.
- PieceName — название произведения. Не всегда присутствует.
- PieceYear — год создания произведения. Не всегда присутствует.
- ComposerId — внешний ключ. Указывает композитора данного произведения. Считаем, что всегда присутствует.

- InstrumentId — внешний ключ. Указывает на инструмент для исполнения произведения. Всегда присутствует. Может быть несколько.
- TonalityTonic — указывает на тонику тональности, записывается в виде названия ноты.
- TonalityModus — указывает на лад тональности.
- FormId — внешний ключ. Указывает на форму музыкального произведения. Считаем, что всегда присутствует.
- Tempold — внешний ключ. Указывает на темп произведения. Всегда присутствует.
- StyleId — внешний ключ. Указывает на стиль произведения. Всегда присутствует.

Функциональные зависимости:

- ISWC→PieceName
- ISWC→PieceYear
- ISWC→ComposerId
- ISWC→InstrumentId
- ISWC→TonalityTonic
- ISWC→TonalityModus
- ISWC→FormId
- ISWC→Tempold
- ISWC→StyleId

Ключи:

- ISWC

В отношении нет повторяющихся групп, все атрибуты атомарны, у отношения есть ключ, следовательно, отношение находится в 1НФ.

По функциональным зависимостям выше делаем вывод, что неключевые атрибуты функционально зависят от ключа в целом (не от части ключа), следовательно, отношение находится в 2НФ.

Данное отношение уже находится в 3НФ, так как для него верно, что неключевые атрибуты непосредственно зависят от ключей.

Все ключи в отношении простые, следовательно, по теореме Дейта-Фейгина 1, отношение находится в 5НФ.

## Отношение Composers

Атрибуты:

- ComposerId — суррогатный ключ. У композиторов могут быть одинаковые имена, потому вводим.
- ComposerName — имя композитора. Всегда присутствует.
- ISWC — внешний ключ. Всегда присутствует, так как композитор без написанных им произведений не композитор.

Функциональные зависимости:

- ComposerId→ComposerName

- ComposerId → ISWC

Ключи:

- ComposerId

## Отношение ComposersPieces

Атрибуты:

- ComposerId — внешний ключ. Всегда присутствует.
- ISWC — внешний ключ. Всегда присутствует.

Функциональные зависимости (нетривиальные) отсутствуют.

Ключи:

- ComposerId ISWC

В отношении нет повторяющихся групп, все атрибуты атомарны, у отношения есть ключ, следовательно, отношение находится в 1НФ.

Все атрибуты ключевые, следовательно, отношение находится в 2НФ и в 3НФ.

У отношения отсутствуют нетривиальные функциональные зависимости, поэтому отношение находится в НФБК.

В отношении 2 атрибута и отношение не является результатом декартова произведения, следовательно, в отношении отсутствуют нетривиальные многозначные функциональные зависимости, поэтому оно находится в 4НФ.

$ComposersPieces(ComposerId, ISWC) = \pi_{ComposerId} \bowtie \pi_{ISWC}$  не выполняется,

поэтому в отношении отсутствуют нетривиальные зависимости соединения, следовательно, отношение находится в 5НФ.

## Отношение Instruments

Атрибуты:

- InstrumentId — суррогатный ключ.
- InstrumentName — название музыкального инструмента. Всегда присутствует.

Функциональные зависимости:

- InstrumentId → InstrumentName
- InstrumentName → InstrumentId

Ключи:

- InstrumentId
- InstrumentName

## Отношение InstrumentsPieces

Атрибуты:

- InstrumentId — внешний ключ. Всегда присутствует.
- ISWC — внешний ключ. Всегда присутствует.

- InstrumentCount — сколько данных инструментов необходимо для данного произведения. Всегда присутствует.

Функциональные зависимости:

- InstrumentId ISWC→InstrumentCount

Ключи:

- InstrumentId ISWC

В отношении нет повторяющихся групп, все атрибуты атомарны, у отношения есть ключ, следовательно, отношение находится в 1НФ.

По функциональным зависимостям выше делаем вывод, что неключевые атрибуты функционально зависят от ключа в целом (не от части ключа), следовательно, отношение находится в 2НФ.

Данное отношение уже находится в 3НФ, так как для него верно, что неключевые атрибуты непосредственно зависят от ключей.

Все ключи в отношении неперекрывающиеся, следовательно, отношение находится в НФБК.

Отношение не является результатом декартова произведения. Отношение имеет 3 возможных варианта нетривиальных многозначных зависимостей:  $InstrumentId \twoheadrightarrow ISWC \mid InstrumentCount$ ,  $ISWC \twoheadrightarrow InstrumentId \mid InstrumentCount$ ,  $InstrumentCount \twoheadrightarrow InstrumentId \mid ISWC$ . Ни одно из них не выполняется, следовательно, в данном отношении нет нетривиальных многозначных функциональных зависимостей. Таким образом, оно находится в 4НФ.

Выясним, есть ли для данного отношения нетривиальная зависимость соединения  $* \{X_1, X_2, \dots, X_n\}$ , где не каждое  $X_i$  — надключ. Возможными вариантами такой зависимости соединения могут быть  $* \{InstrumentId, ISWC\}$ ,  $\{InstrumentId, InstrumentCount\}$ ,  $* \{InstrumentId, ISWC\}, \{ISWC, InstrumentCount\}$ ,  $* \{InstrumentId, ISWC\}, \{InstrumentId, InstrumentCount\}, \{ISWC, InstrumentCount\}$ , но ни одна из них не выполняется, следовательно, отношение находится в 5НФ.

## Отношение Forms

Атрибуты:

- FormId — суррогатный ключ.
- FormName — название музыкальной формы. Всегда присутствует.

Функциональные зависимости:

- FormId→FormName
- FormName→FormId

Ключи:

- FormId
- FormName

## Отношение Tempos

Атрибуты:

- Tempold — суррогатный ключ.
- TempoName — термин, указывающий на темп музыкального произведения. Всегда присутствует.
- TempoMinBPM — минимальное значение bpm, удовлетворяющее данному темпу. Всегда присутствует.
- TempoMaxBPM — максимальное значение bpm, удовлетворяющее данному темпу. Всегда присутствует.

Функциональные зависимости:

- Tempold→TempoName
- Tempold→TempoMinBPM
- Tempold→TempoMaxBPM
- TempoName→Tempold

Ключи:

- Tempold
- TempoName

## Отношение Styles

Атрибуты:

- StyleId — суррогатный ключ.
- StyleName — название музыкального стиля. Всегда присутствует.

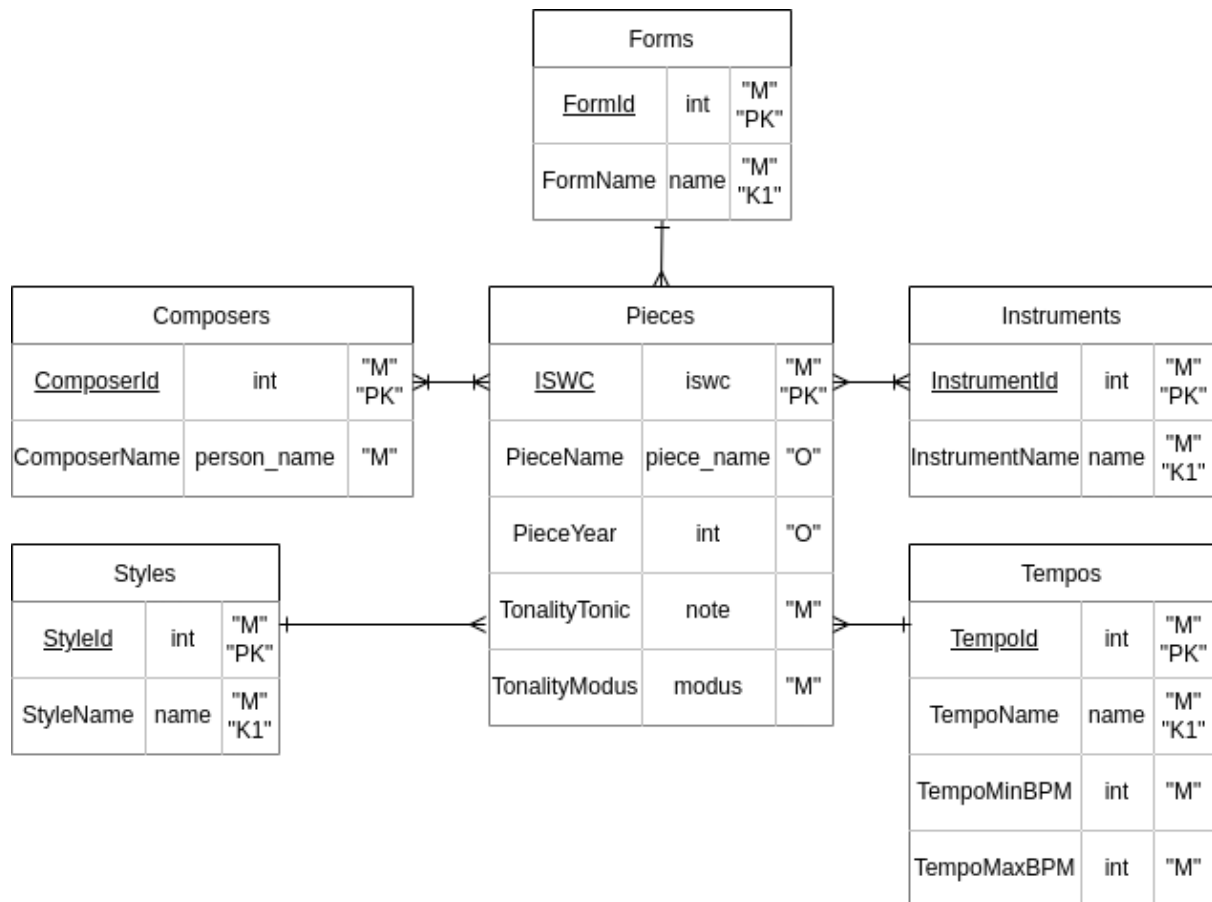
Функциональные зависимости:

- StyleId→StyleName
- StyleName→StyleId

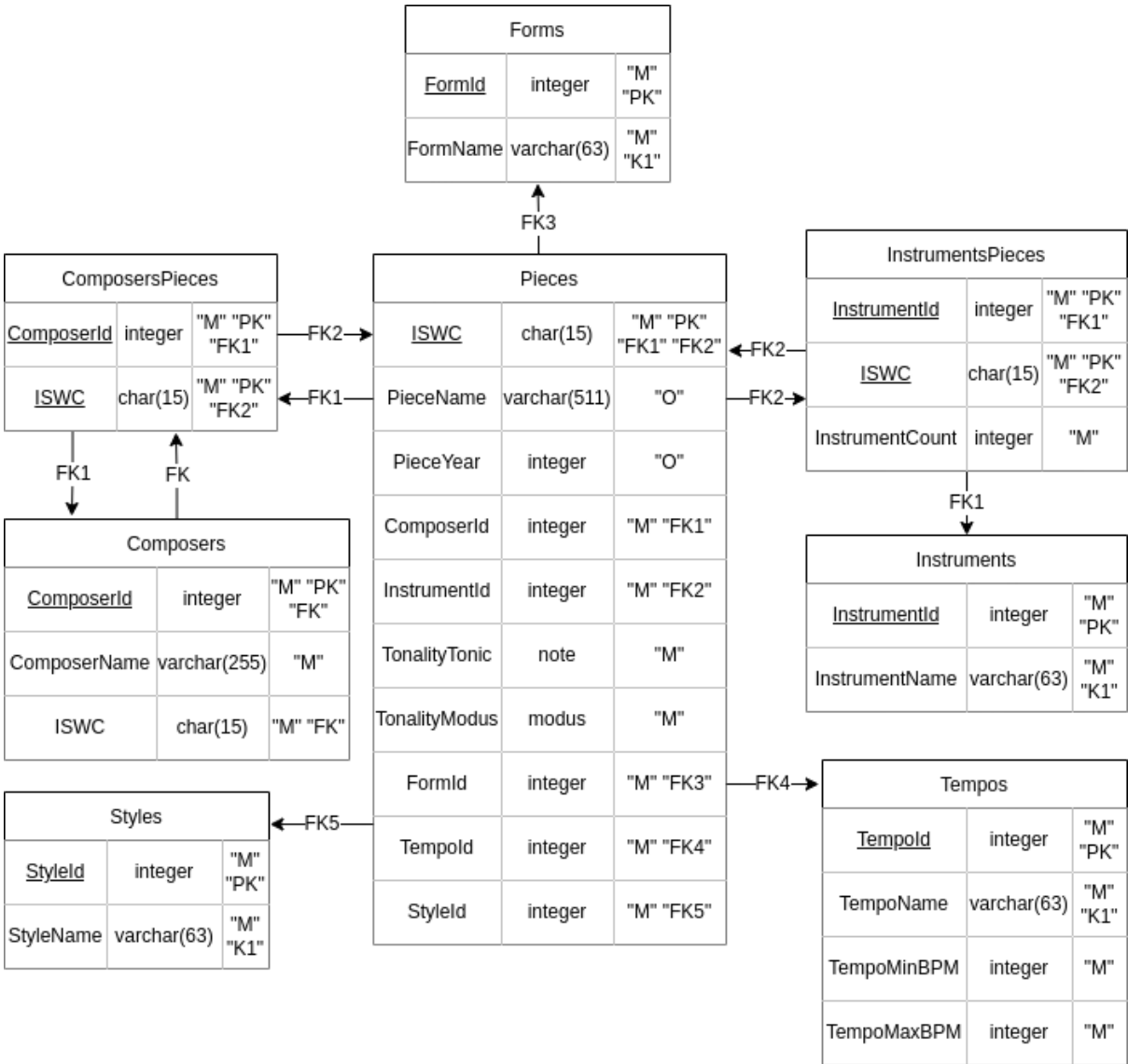
Ключи:

- StyleId
- StyleName

# Модель сущность-связь



# Физическая модель



При построении физической модели использовалось следующее отображение доменов в типы:

Домен	Тип
iswc	char (15)
piece_name	varchar (511)
int	integer
name	varchar (63)
person_name	varchar (255)
note	note

modus	modus
-------	-------

## Определения таблиц

Для реализации проекта использовалась СУБД PostgreSQL 14.5 (Ubuntu 14.5-0ubuntu0.22.04.1). Определения таблиц и их индексов приведено в файле `ddl.sql`.

Ограничение “C\_CP\_ComposerId\_ISWC\_FK” определяется отдельно от таблицы `Composers`, так как невозможно создать внешнюю ссылку на еще не существующую таблицу `ComposersPieces`. Таблица `ComposersPieces` так же содержит внешнюю ссылку на таблицу `Composers`, поэтому перенести создание `ComposersPieces` не является решением.

Ограничение “P\_CP\_ISWC\_ComposerId\_FK1” определяется отдельно от таблицы `Pieces`, так как невозможно создать внешнюю ссылку на еще не существующую таблицу `ComposersPieces`. Таблица `ComposersPieces` так же содержит внешнюю ссылку на таблицу `Pieces`, поэтому перенести создание `ComposersPieces` не является решением.

Ограничение “P\_IP\_ISWC\_InstrumentId\_FK2” определяется отдельно от таблицы `Pieces`, так как невозможно создать внешнюю ссылку на еще не существующую таблицу `InstrumentsPieces`. Таблица `InstrumentsPieces` так же содержит внешнюю ссылку на таблицу `Pieces`, поэтому перенести создание `InstrumentsPieces` не является решением.

## Тестовые данные

Скрипт для добавления тестовых данных приведен в файле `insert.sql`.

## Запросы на получение данных

В рамках проекта были реализованы следующие запросы:

- `GetPiecesInfoAggrByISWC` — получение информации о произведении по ISWC в агрегированном виде.
- `GetPiecesInfoAggrWithComposerId` — получение информации о произведениях по композитору `ComposerId`, принимавшему участие в написании, в агрегированном виде.
- `GetPiecesInfoByInstrumentName` — получение информации о произведениях, написанных только для инструмента `InstrumentName`.
- `GetPiecesInfoWithInstrumentName` — получение информации о произведении, написанному для инструмента `InstrumentName` в том числе.



- `GetPiecesInfoAggrByYears` — получение информации о произведениях, написанных в указанный период, в агрегированном виде.
- `GetPiecesInfoByFormName` — получение информации о произведениях, написанных в указанной форме.
- `GetPiecesInfoByTempoName` — получение информации о произведениях, исполняющихся в указанном темпе.
- `GetPiecesInfoByStyleName` — получение информации о произведениях, написанных в указанном стиле.
- `GetPiecesInfoByFormNameAndComposerId` — получение информации о произведениях, написанных в указанной форме определенным автором.
- `GetComposersIdsByStyleName` — получение `ComposerId` композиторов, работавших в определенном стиле.
- `GetPiecesCountByComposerId` — получение статистики количества написанных произведений по композиторам.
- `GetPiecesCountByComposerIdAndFormName` — получение статистики количества написанных произведений по композиторам и формам.
- `GetPiecesCountByComposerIdAndInstrumentName` — получение статистики количества написанных произведений по композиторам и инструментам.

Для реализации запросов были созданы вспомогательные представления:

- `PiecesWithAllComposers` — представление с информацией о произведениях с учетом множественности композиторов.
- `PiecesWithAllComposersAndAllInstruments` — представление с информацией о произведениях с учетом множественности авторов и инструментов.
- `PiecesInformation` — представление со всей информацией по всем произведениям.
- `PiecesInformationAggr` — представление со всей информацией по всем произведениям в агрегированном виде.

Запросы на получение данных и вспомогательные представления приведены в файле `select.sql`.

## Запросы на изменение данных

В рамках проекта были реализованы следующие запросы:

- `ChangePieceYear` — изменение года создания произведения по `ISWC`.
- `ChangePieceName` — изменение названия произведения по `ISWC`.
- `ChangeComposerName` — изменение имени композитора по `ComposerId`.
- `ChangeInstrumentName` — изменение названия инструмента по старому названию инструмента.

- `ChangeFormName` — изменение названия формы по старому названию формы.
- `ChangeTempoName` — изменение названия темпа по старому названию темпа.
- `ChangePieceTempo` — изменение темпа произведения по ISWC и названию нового темпа.
- `ChangePieceISWC` — изменение ISWC произведения по старому ISWC. При данном запросе вызывается триггер `UpdateISWCBeforeTrigger`.
- `UpdatePieceComposers` — обновление композиторов произведения по ISWC. Реализовано с помощью процедуры, куда передаются ISWC произведения и список `ComposerId` нового состава композиторов. В случае, если какой-то композитор в результате останется без произведения, будет откат, так как композитор обязан иметь произведение, а удалять запись при изменении, кажется, не является ожидаемым поведением.
- `UpdatePieceInstruments` — обновление инструментов, для которых написано произведение с данным ISWC. Реализовано с помощью процедуры, куда передаются ISWC произведения и список пар, где первым значением является `InstrumentId`, а вторым — кол-во инструментов с данным `InstrumentId`.
- `DeleteComposer` — удаление композитора по `ComposerId`. При удалении композитора удаляются все произведения, написанные лично им, в остальных он удаляется из списка композиторов.
- `DeletePiece` — удаление произведения по ISWC. При данном запросе вызывается триггер `DeletePieceBeforeTrigger`.

Запросы на изменение данных, хранимые процедуры и триггеры приведены в файле `update.sql`.