Экзаменационный проект   
по дисциплине  
Проектирование баз данных

студента гр. **M34342**

**Ота Никиты Терентьевича**

по теме

**Соревнования по фигурному катанию**

# Описание проекта

База данных состоявшихся соревнований по фигурному катанию.

На ресурсе присутствуют фигуристы, команды и их результаты по разным дисциплинам.

Дисциплины бывают: одиночные, парные, танцы и синхронное катание. У каждой дисциплины свои требования и своя разбалловка.

Система оценивания бралась согласно по [ИСУ](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B4%D0%B5%D0%B9%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%98%D0%A1%D0%A3).

По каждой дисциплине есть предписанные **элементы**, и у каждого участника соревнования имеется выбор, в каком количестве их выполнить на льду. У каждого выполняемого элемента есть стоимость элемента в очках **SOV** (scale of value, или попросту очки за базовое выполнение) и **GOE** (grade of execution) в диапозоне [-n, n] (n меняется из года в год) – усреднённая по алгоритму оценка судей уровня исполнения элемента. *Здесь и далее не будет рассматриваться алгоритм вычисления средней оценки всех судей, так как в базу данных уже вводится эта усреднённая величина.*

По итогу выступления у судей складывается впечатление. За это так же ставятся баллы. Судьи ставят оценки за **пять** составляющих “компонентов” программы от 0 до 10 с шагом в 0.25. За каждый компонент вводится множитель **Factor**, установленный для данного уровня соревнования. Очки за **общее впечатление** (Total Program Component Score или Presentation Score) вычисляются суммой произведения *средней оценки* компонента судей на Factor.

За некоторые обстоятельства также существуют различные **штрафы**, ставятся натуральными числами, но в сумме не превосходит 12.

В общем зачёте победителем становится такой(ие) спортсмен / пара / команда, что больше всех набрал(и) сумму очков за элементы, общее впечатление и штрафы.

Одиночные катания включают в себя **short program** (короткая программа) и **free skating** (произвольная программа), по каждой идёт оценка выступления и награждаются лучшие по суммарной оценке. Делятся на мужские и женские дисциплины.

Парные катания так же, как и одиночные, оцениваются по сумме из **short program** и **free skating**, пары обязательно состоят из спортсменов разных полов.

Танцы на льду включают в себя **short dance** (короткий танец) или rhythm-dance (ритм-танец) и **free dance** (произвольный танец). Оценка суммарная. До сезона 2010-2011 гг. зачёт по танцам был из трёх танцев: *compulsory dance*, *original dance* и *free dance* (обязательный, оригинальный и произвольный соответственно), но в данной базе эта система оценивания не будет рассматриваться.

Синхронные танцы, как одиночные и парные катания, состоят из **short program** и **free skating**. В синхронных танцах выступает команда из десяток фигуристов.

# Построение отношений

В результате предварительного проектирования были выделены следующие отношения:

* People — информация о человеке или фигуристе: имя, дата рождения, город.
* PersonalTrainers — информация о тренерах фигуриста в индивидуальном зачёте.
* Pairs — информация о паре: фигурист и фигуристка.
* Teams — информация о команде на синхронное катание: название, страна.
* TeamsTrainers — тренера команды.
* TeamsParticipants — фигуристы команды. В каждую команду входят десятки фигуристов.
* Competitions — информация о соревновании: дата проведения, город, страна.
* CompetitionStandings — оценка участников соревнования.
* PerformanceDetails — информация о выполненных элементах участником соревнования и оценка за них.
* PerformanceComponents — информация о впечатлениях судей, полученных от выступления участником соревнования, и штрафах. Создано с целью отказа от дублирования атрибутов в отношении CompetitionStandings.
* PerformanceDeductions — информация о штрафах за выступление участника. Создано с целью отказа от дублирования атрибутов в отношении CompetitionStandings.
* ElementInformations — информация об элементе, баллы и полное название.

## Отношение People

Атрибуты:

* PersonId — уникальный идентификатор человека.
* PersonName — имя человека.
* BirthDate — дата рождения человека.
* Country — страна.

Функциональные зависимости:

* PersonId → PersonName
* PersonId → BirthDate
* PersonId → Country

Ключи:

* {PersonId}

### Нормализация

1 НФ:

* Отсутствует повторение групп
* Атрибуты атомарны
* Надключом является PersonId, который является также ключом
* Таким образом, отношение находится в 1 НФ

Отношение в 2 НФ

3 НФ:

* 2 НФ
* Не ключи функционально не транзитивно зависят от ключа
* Таким образом, отношение находится в 3 НФ

Отношение в НФБК, в 4 НФ

5 НФ:

* 4 НФ
* По теореме Дейта-Фейгина все ключи простые
* Таким образом, отношение находится в 5 НФ

## Отношение PersonalTrainers

Атрибуты:

* PersonId — уникальный идентификатор подопечного фигуриста.
* TrainerId — уникальный идентификатор тренера.

Функциональные зависимости:

Ключи:

## Отношение Teams

Атрибуты:

* TeamId — уникальный идентификатор команды.
* TeamName — название команды.
* Country — страна, за которую выступает эта команда.

Функциональные зависимости:

* TeamId → TeamName
* PersonId → Country

Ключи:

* {TeamId}

## Отношение Pairs

Атрибуты:

* PairId — уникальный идентификатор пары.
* PersonMId — идентификатор фигурсита.
* PersonFId — идентификатор фигуристки.
* Country — страна, за которую выступает пара. Может варьироваться по разным обстоятельствам.

Функциональные зависимости:

* PairId → PersonMId
* PairId → PersonFId
* PairId → Country

Ключи:

* {PairId}

## Отношение TeamTrainers

Атрибуты:

* TeamId — идентификатор команды тренера.
* TrainerId — идентификатор тренера.

Функциональные зависимости:

Ключи:

## Отношение TeamParticipants

Атрибуты:

* PersonId — идентификатор фигуриста.
* TeamId — идентификатор команды, в которой фигурист состоит.

Функциональные зависимости:

Ключи:

## Отношение Competitions

Атрибуты:

* CompetitionId — уникальный идентификатор соревнования.
* CompetitionName — название соревнования.
* DateStarts — дата начала соревнования.
* DateEnds — дата окончания соревнования.

Функциональные зависимости:

* CompetitionId → CompetitionName
* CompetitionId → DateStarts
* CompetitionId → DateEnds
* CompetitionName, DateStarts, DateEnds → CompetitionId

Ключи:

* {CompetitionId}
* {CompetitionName, DateStarts, DateEnds}

### Нормализация

1 НФ:

* В отношении нет повторяющихся групп.
* Все атрибуты атомарны.
* Как можно заметить из функциональных зависимостей, замыкамие атрибута {CompetitionId} над множеством ФЗ, дает в результате множество всех атрибутов, так как для любого атрибута из отношения существует ФЗ из CompetitionId в этот атрибут. Исходя из этого, CompetitionId — надключ.
* Так как CompetitionId — множество из одного элемента, то оно уже минимально по включению, следовательно множество {CompetitionId} — ключ. Также если рассмотреть замыкание атрибутов CompetitionName, DateStarts, DateEnds, то по ФЗ CompetitionName, DateStarts, DateEnds → CompetitionId, понимаем, что в нашем замыкание будет CompetitionId (который является ключом), а значит, и все остальные атрибуты. Следовательно {CompetitionName, DateStarts, DateEnds} — тоже ключ из тех же соображений.
* Заметим, что ни один из оставшихся атрибутов не присутствует в левой части какой-либо ФЗ, а значит из них никак не получится вывести остальные атрибуты. Следовательно, они не являются частью какого-либо ключа.
* Таким образом, отношение находится в 1 НФ.

2 НФ:

* Отношение находится в 1 НФ.
* В нашем отношении, атрибуты DateStarts и DateEnds — неключевые атрибуты. Но, так как {CompetitionId} — ключ и есть ФЗ CompetitionId → DateStarts, CompetitionId → DateEnds, и больше нигде в ФЗ DateStarts, DateEnds не входят, то делаем вывод, что нет неключевых атрибутов, которые бы зависели от части ключа.
* Таким образом, отношение находится во 2 НФ.

3 НФ:

* Отношение находится во 2 НФ.
* У неключевых атрибутов DateStarts и DateEnds есть ФЗ, в которых они непосредственно зависят от ключа.
* Таким образом, отношение находится в 3 НФ.

НФБК:

* В отношении ключами являются {CompetitionId} и {CompetitionName, DateStarts, DateEnds}. Во всех приведенных нетривиальных ФЗ, слева стоит ключ, а значит и надключ.
* Таким образом, отношение находится в НФБК.

4 НФ:

* Покажем, что в нашем отношении нет нетривиальных МЗ, где слева стоял бы не надключ. От противного. Пусть такая МЗ есть. Значит слева не надключ, следовательно там не может быть ключевых атрибутов (так как по следствию из определения, любое надмножество ключ — его надключ). Значит слева могут стоять только неключевые атрибуты. Но у нас единственные неключевые атрибуты DateStarts и DateEnds. Тогда для правой части остаются 2 оставшихся атрибута-ключа в каждом случае. Но как бы их не расставили в правой части, чтобы получилось нетривиальная МЗ, всегда будут получаться эквивалентные МЗ (по следствию из теоремы Дейта-Фейгина). Рассмотрим МЗ: DateStarts ↠ CompetitionId | CompetitionName, DateStarts, DateEnds. Это многозначная зависимость не выполняется, так как при фиксированном DateStarts, для разных CompetitionName, DateStarts, DateEnds не могут быть одинаковое множество CompetitionId, так как CompetitionId — ключ и уникален для каждого кортежа. А при разных значениях атрибутов CompetitionName, DateStarts, DateEnds, мы, очевидно, получаем 2 разных кортежа. Такой же исход получится и при DateEnds. Противоречие. Следовательно в данном отношении не существует таких МЗ.
* Таким образом, отношение находится в 4 НФ.

5 НФ:

* Покажем, что в нашем отношении нет ЗС, у которых одно из составляющих было бы не надключом. От противного пусть такое есть. Тогда в нем есть некое X не являющееся надлюче. Как мы уже знаем, такое множество X единственное и является множеством {DateStarts, DateEnds}. Рассмотрим оставшиеся множества из данной ЗС. Их естественное соединения (по ассоциативности и коммутативности) можно записать, как множесто Y. Заметим, что множество Y должно содержать атрибуты CompetitionId и CompetitionName, DateStarts, DateEnds, иначе в итоговом естественном соединении нельзя было бы их ни откуда получить, а значит такое естественно соединение не равнялось бы исходному отношению. Значит для разных кортежей множества Y различны и отличаются как минимум в CompetitionId и CompetitionName. Пусть также наше отношение состоят только из двух кортежей, где, как мы уже знаем множества Y — различны, но также и различны множества X. Тогда, во-первых, если X ∩ Y = ∅, то естественное соединение — декартого произведение в данном случае и мы получим 4 различных кортежа, вместо двух. Во-вторых, если X ∩ Y ≠ ∅, то Y содержит CompetitionId, CompetitionName, DateStarts, DateEnds и DateStarts, DateEnds. Следовательно это тривиальная ЗС, что мы не рассматриваем. Следовательно мы рассмотрели все возможные ЗС, которые нам могли бы мешать. Противоречие. Таких ЗС нет в нашем отношении.
* Таким образом, отношение находится в 5 НФ.

## Отношение CompetitionStandings

Атрибуты:

* CompetitionId — идентификатор соревнования, в котором было выступление.
* CompetitionType — тип соревнования. Одиночный, парный или командный (синхронные танцы). Короткая программа или произвольная.
* CompetitorId — идентификатор выступавшего фигуриста / пары / команды.
* PerformanceId — идентификатор таблицы с описанием выполненных элементов и общих впечатлений
* ShortScore — оценка за короткую программу.
* FreeScore — оценка за произвольную программу.
* TotalScore — суммарная оценка за обе программы участника.

Функциональные зависимости:

* CompetitionId, CompetitionType, CompetitorId → ShortScore
* CompetitionId, CompetitionType, CompetitorId → FreeScore
* CompetitionId, CompetitionType, CompetitorId → TotalScore
* ShortScore, FreeScore → TotalScore
* CompetitionId, CompetitionType, CompetitorId → PerformanceId
* PerformanceId → CompetitionId, CompetitionType, CompetitorId

Ключи:

* {CompetitionId, CompetitionType, CompetitorId}
* {PerformanceId}

## Отношение PerformanceDetails

Атрибуты:

* PerformanceId — идентификатор выступления фигуриста / пары / команды.
* ElementId — идентификатор выполненного элемента выступления по порядку выполнения.
* ElementAbbreviation — аббревиатура выполненного элемента (подробнее в ElementInformations).
* GOE — Grade of Execution, средняя оценка исполнения элемента судей.

Функциональные зависимости:

* PerformanceId, ElementId → ElementAbbreviation
* PerformanceId, ElementId → GOE

Ключи:

* {PerformanceId, ElementId}

## Отношение PerformanceComponents

Атрибуты:

* PerformanceId — идентификатор выступления фигуриста / пары / команды.
* ComponentType — тип компонента, по которому выставляется средняя оценка судей.
* Value — оценка компонента.
* Factor — коэффициент умножения данного компонента.

Функциональные зависимости:

* PerformanceId, ComponentId → Value
* PerformanceId, ComponentId → Factor

Ключи:

* {PerformanceId, ComponentId}

## Отношение PerformanceDeductions

Атрибуты:

* PerformanceId — идентификатор выступления фигуриста / пары / команды.
* Value — штрафные баллы.

Функциональные зависимости:

* PerformanceId → Value

Ключи:

* {PerformanceId}

## Отношение ElementInformations

Атрибуты:

* ElementAbbreviation — аббревиатура элемента.
* ElementDescription — полное название элемента.
* SOV — Scale of Value, базовая оценка за этот элемент.

Функциональные зависимости:

* ElementAbbreviation → ElementDescription
* ElementAbbreviation → SOV
* ElementDescription → ElementAbbreviation

Ключи:

* {ElementAbbreviation}
* {ElementDescription}

### Нормализация

1 НФ:

* Отсутствует повторение групп
* Атрибуты атомарны
* Надключом является ElementAbbrevation, который является так же ключом
* Таким же образом ElementDescription является ключом
* Таким образом, отношение находится в 1 НФ

2 НФ:

* Отношение находится в 1 НФ.
* В нашем отношении, атрибут SOV — неключевой атрибут. Но, так как {ElementAbbrevation} — ключ и есть ФЗ ElementAbbrevation →SOV, и больше нигде в ФЗ SOV не входит, то делаем вывод, что нет неключевых атрибутов, которые бы зависели от части ключа.
* Таким образом, отношение находится во 2 НФ.

3 НФ:

* Отношение находится во 2 НФ.
* У неключевого атрибута SOV есть ФЗ, в котором он непосредственно зависит от ключа.
* Таким образом, отношение находится в 3 НФ.

НФБК:

* В отношении ключами являются {ElementAbbrevation} и {ElementDescription}. Во всех приведенных нетривиальных ФЗ, слева стоит ключ, а значит и надключ.
* Таким образом, отношение находится в НФБК.

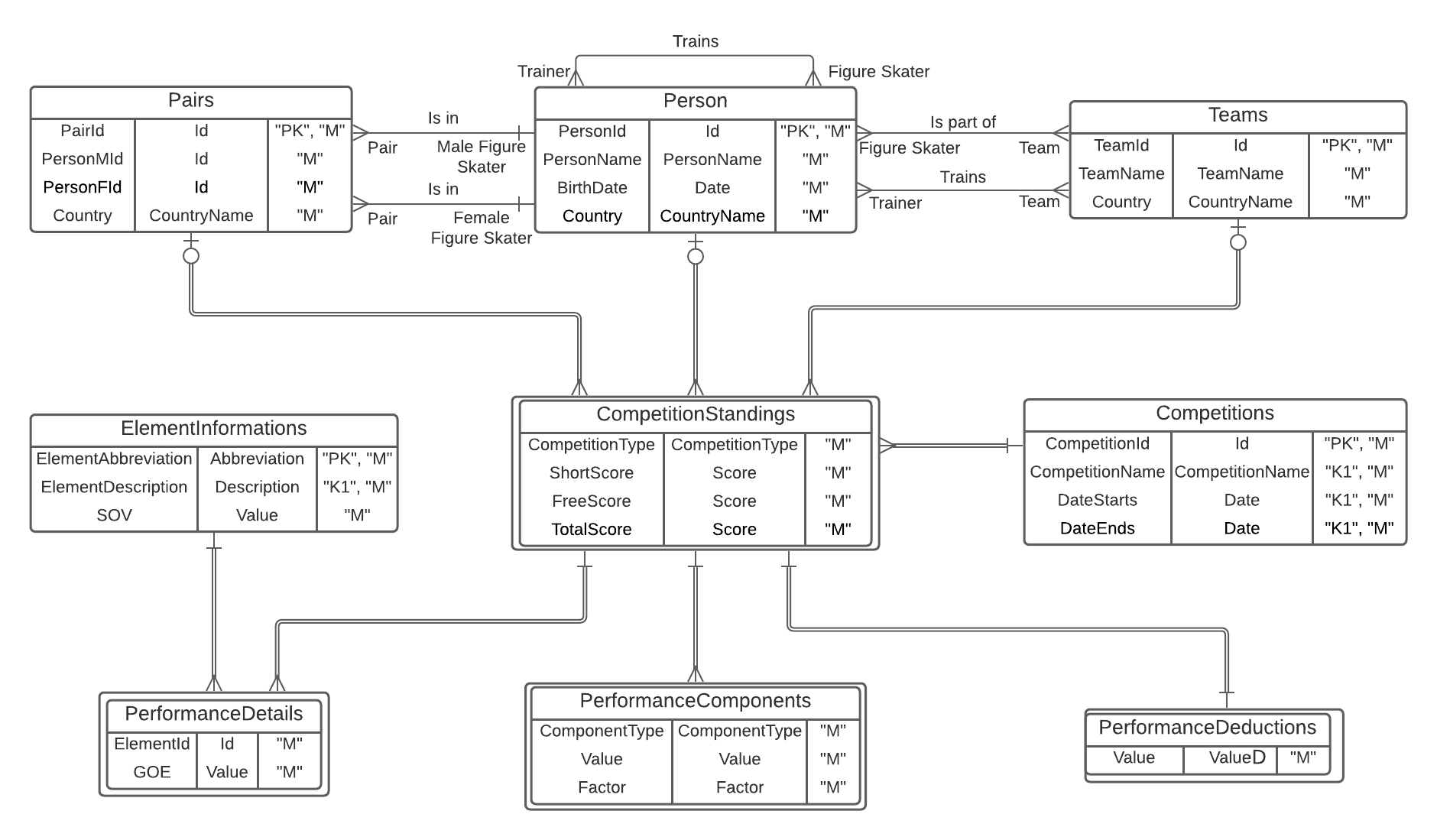
4 НФ:

* Покажем, что в нашем отношении нет нетривиальных МЗ, где слева стоял бы не надключ. От противного. Пусть такая МЗ есть. Значит слева не надключ, следовательно там не может быть ключевых атрибутов (так как по следствию из определения, любое надмножество ключ — его надключ). Значит слева могут стоять только неключевые атрибуты. Но у нас только один атрибут SOV. Тогда для правой части остаются 2 оставшихся атрибута-ключа в каждом случае. Но как бы их не расставили в правой части, чтобы получилось нетривиальная МЗ, всегда будут получаться эквивалентные МЗ (по следствию из теоремы Дейта-Фейгина). Рассмотрим МЗ: SOV ↠ ElementsAbbrevation | ElementDescription. Это многозначная зависимость не выполняется, так как при фиксированном SOV, для разных ElementDescription не могут быть одинаковое множество ElementAbbrevation, так как ElementAbbrevation — ключ и уникален для каждого кортежа. А при разных значениях атрибута ElementDescription, мы, очевидно, получаем 2 разных кортежа. Противоречие. Следовательно в данном отношении не существует таких МЗ.
* Таким образом, отношение находится в 4 НФ.

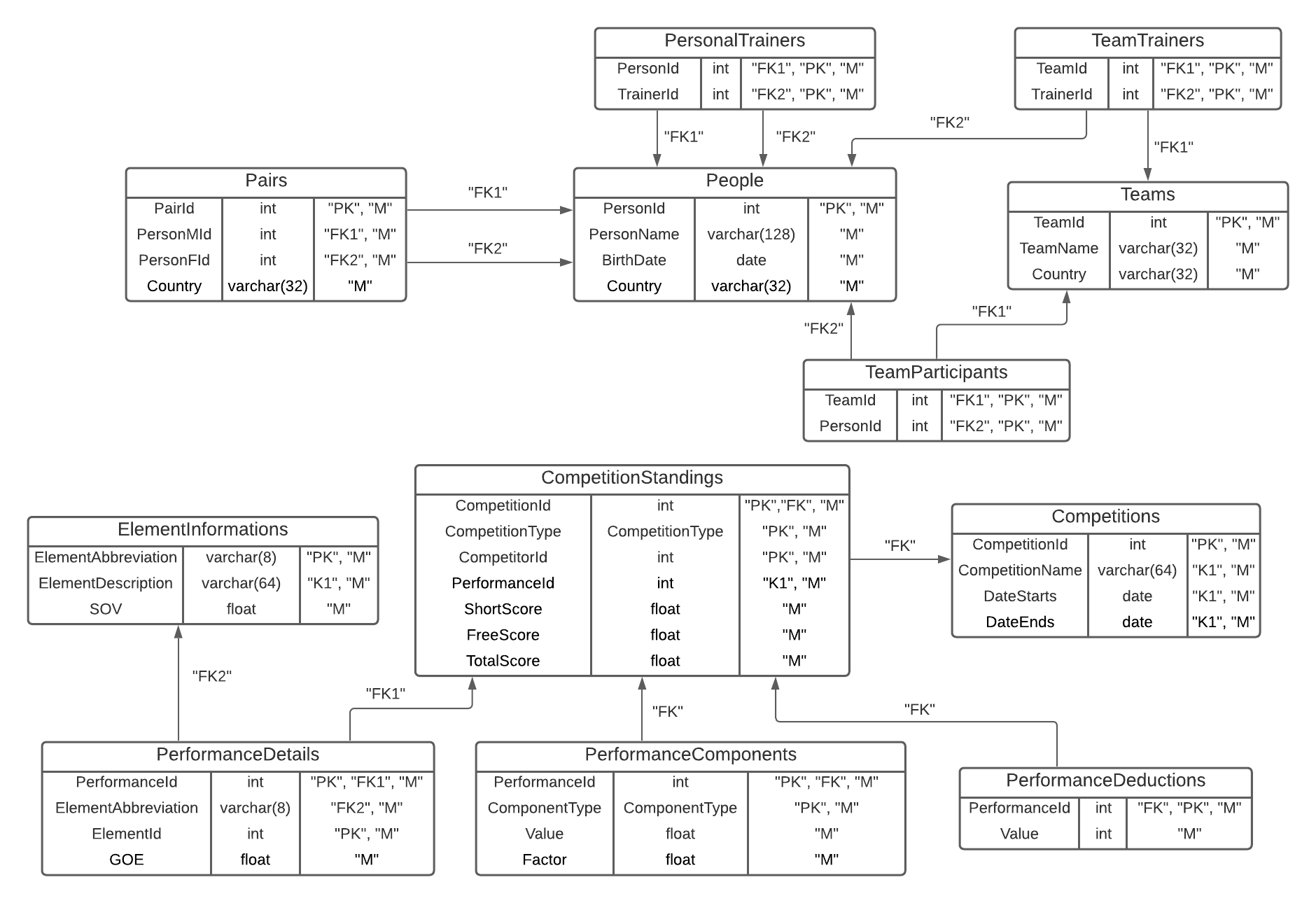
5 НФ:

* Покажем, что в нашем отношении нет ЗС, у которых одно из составляющих было бы не надключом. От противного пусть такое есть. Тогда в нем есть некое X не являющееся надлюче. Как мы уже знаем, такое множество X единственное и является множеством {SOV}. Рассмотрим оставшиеся множества из данной ЗС. Их естественное соединения (по ассоциативности и коммутативности) можно записать, как множесто Y. Заметим, что множество Y должно содержать атрибуты ElementAbbrevation и ElementDescription, иначе в итоговом естественном соединении нельзя было бы их ни откуда получить, а значит такое естественно соединение не равнялось бы исходному отношению. Значит для разных кортежей множества Y различны и отличаются как минимум в ElementAbbrevation и ElementDescription. Пусть также наше отношение состоят только из двух кортежей, где, как мы уже знаем множества Y — различны, но также и различны множества X. Тогда, во-первых, если X ∩ Y = ∅, то естественное соединение — декартого произведение в данном случае и мы получим 4 различных кортежа, вместо двух. Во-вторых, если X ∩ Y ≠ ∅, то Y содержит ElementAbbrevation, ElementDescription и SOV. Следовательно это тривиальная ЗС, что мы не рассматриваем. Следовательно мы рассмотрели все возможные ЗС, которые нам могли бы мешать. Противоречие. Таких ЗС нет в нашем отношении.
* Таким образом, отношение находится в 5 НФ.

# Модель сущность-связь



# Физическая модель



При построении физической модели использовалось следующее отображение доменов в типы:

| Домен | Тип |
| --- | --- |
| Id | int |
| PersonName | varchar(128) |
| CountryName | varchar(32) |
| TeamName | varchar(32) |
| Date | date |
| CompetitionName | varchar(64) |
| Value | float |
| ValueD | int |
| Abbreviation | varchar(8) |
| Description | varchar(64) |
| CompetitionType | enum(‘Single’, ‘Pair’, ‘Dance’, ‘Synchronized’) as CompetitionType |
| ComponentType | enum('Skating skills',  'Transitions',  'Performance',  'Composition',  'Interpretation') as ComponentType |

# Определения таблиц

Для реализации проекта использовалась СУБД PostgreSQL 12.9. Определения таблиц и их индексов приведено в файле ddl.sql.

# Тестовые данные

Скрипт для добавления тестовых данных приведен в файле data.sql.

# Запросы на получение данных

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

# Запросы на изменение данных

В рамках проекта были реализованы следующие запросы:

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