Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

**БАЗА ДАННЫХ ОЛИМПИЙСКИХ ИГР**

|  |  |  |
| --- | --- | --- |
| Студент |  | М.Г. Гладкий |
| Руководитель |  | Е.Е. Фадеева |

Минск 2021

**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 4](#_Toc90240077)

[ВВЕДЕНИЕ 5](#_Toc90240078)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc90240079)

[1.1 Информация об Олимпийских играх 7](#_Toc90240080)

[1.2 Обзор аналогов 8](#_Toc90240081)

[1.3 Постановка задачи 12](#_Toc90240082)

[2 ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ 13](#_Toc90240083)

[3 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 14](#_Toc90240085)

[3.1 Выбор СУБД 14](#_Toc90240089)

[3.2 Спецификация функциональных требований 14](#_Toc90240090)

[3.3 Разработка даталогической модели 15](#_Toc90240091)

[3.4 Физическая модель базы данных 21](#_Toc90240092)

[4 ТЕСТИРОВАНИЕ 26](#_Toc90240093)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 28](#_Toc90240095)

[ЗАКЛЮЧЕНИЕ 29](#_Toc90240096)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc90240097)

[ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ 31](#_Toc90240098)

**ВВЕДЕНИЕ**

В связи с постоянным увеличением объема окружающей информации, появлением новых сфер деятельности и развитием технологий особенно актуальной становится задача создания баз данных, которые смогли бы охватить все существующие предметные области. Хранить большие объемы данных на печатных носителях нецелесообразно не только с точки зрения экономии природных ресурсов, но и прежде всего в связи с отсутствием либо сложностью организации всей информации в структурированном виде.

Понятие базы данных можно применять к любой связанной по определенному признаку информации, хранимой и ограниченной особым образом - что выполняется в СУБД в виде таблиц. По сути БД – это некоторое подобие картотеки, электронного хранилища данных, которые хранятся в компьютере в виде одного или нескольких файлов.

Хранение данных с помощью таблиц, с последующим объединением в более сложные иерархические структуры и установлением логических связей между сущностями базы данных позволяет легко редактировать существующие либо добавлять новые данные, получать сведения для анализа в удобном виде, мгновенно находить необходимую информацию для принятия важных решений.

Методология проектирования баз данных может рассматриваться как совокупность методов и средств, последовательное применение которых обеспечивает разработку проекта баз данных, удовлетворяющего заданным целям. Рассматриваемая методология позволяет пользователю лучше понять, как следует специфицировать требования к данным.

Следуя этой методологии, проектировщик может выполнять более глубокий и содержательный анализ требований к данным, осуществлять контроль и управление кодом проектирования.

Целью курсовой работы является проектирование базы данных Олимпийских игр.

Олимпийские игры вызывают неподдельный интерес как самих спортсменов, так и зрителей, а, следовательно, выбранная тема будет пользоваться популярностью среди пользователей.

Олимпийские игры проводятся каждые 2 года на протяжении нескольких столетий, поэтому проектируемая база данных должна обладать высокой гибкостью архитектуры для быстрой адаптации и последующего применения в рамках нескольких Олимпийских сезонов, в целях снижения затрат при ее реализации.

В связи с длительными интервалами в проведении игр, данная тема еще не полностью раскрыта на рынке мобильных и веб-приложений, а в совокупности с быстрым развитием компьютерных технологий созданная база данных станет мощным инструментом в руках разработчиков программного обеспечения.

Разработка базы данных «Олимпийские игры» включает:

– проектирование модулей «Результаты соревнований», «Информация о спортсменах», «Расписание соревнований», «Трансляция Олимпийских игр», «Администрирование»;

– создание таблиц данных в каждом разрабатываемом модуле;

– организацию связей в таблицах;

– тестирование базы данных.

Итоговая задача курсовой работы заключается в создании функционирующей базы данных «Олимпийские игры» для автоматизации обработки данных, полученных в ходе проведения Олимпийских игр.

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

* 1. Информация об Олимпийских играх

Ежедневно во всем мире проходит огромное количество спортивных соревнований различного уровня. Они отличаются и местом проведения, и возрастной категорией участников, и количеством спортивных дисциплин. Однако кульминацией в карьере любого спортсмена по праву считается выступление на Олимпийских играх.

В отличие от Чемпионатов мира, Олимпийские игры включают огромное число спортивных дисциплин, а география участников и зрителей охватывает весь мир.

Олимпийские игры состоялись уже более полусотни раз, а борьба за право проведения начинается задолго до олимпийского года.

Участниками последней Олимпиады в Токио в 339 событиях и 33 видах спорта стали 11417 спортсменов из 206 команд со всего мира[1]. В связи с постоянно увеличивающимся масштабом проведения Олимпийских игр, а также изменениями в программе олимпийских дисциплин особенно важной становится задача автоматизации и систематизации результатов спортсменов на всех этапах соревнований по всем олимпийским направлениям.

Олимпийские игры XXI века – это не только соревнования сильнейших спортсменов, но и соответствующая инфраструктура (стадионы, гостиницы, транспортное сообщение), мультимедийное сопровождение, грандиозные церемонии открытия и закрытия.

Это обуславливает необходимость создания базы данных «Олимпийские игры», в которой будут отражены не только результаты спортивных соревнований, но и вся сопутствующая проведению Олимпиады информация.

На Игры в Сочи в 2014 было продано 1,1 миллионов билетов для гостей из 126 стран, трансляция соревнований проводилась на 464 каналах, а зрителями стали более 2 миллиардов человек, что наглядно свидетельствует об интересе огромного числа пользователей к выбранной теме.

График проведения Олимпийских игр известен на ближайшие 8 лет, что говорит об актуальности поставленной задачи не только в настоящее время.

* 1. Обзор аналогов

Для получения наиболее полной информации о состоянии рынка мобильных приложений, охватывающих тему Олимпийских игр, будет проведен сравнительный анализ доступного ПО.

«Olympics» – официальное приложение Олимпийского коммитета, предоставляющее возможности по отслеживанию последних обновлений олимпиад, новостей из мира спорта, количества завоеванных медалей, расписания игр. Внешний вид приложения представлен на рисунке 1.1

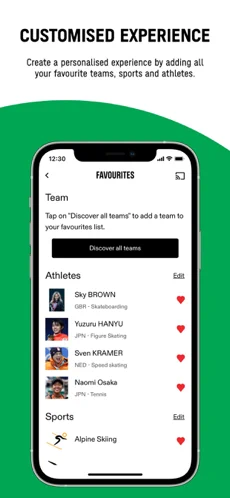
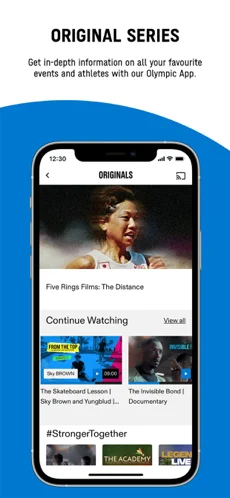


Рисунок 1.1 – Внешний вид приложения «Olympics»

Достоинства:

– возможность просмотра результатов спортивных соревнований текущей Олимпиады;

– возможность просмотра расписания соревнований;

– возможность просмотра информации о спортсменах;

– просмотр медального зачета;

– возможность добавить любимые виды спорта и спортсменов в раздел «Избранное» для сокращения времени поиска;

– приложение доступно для всех типов мобильных операционных систем.

Недостатки:

– отсутствие локализации;

– отсутствие персонализации для пользователя.

«Sochi 2014: Результаты» — официальное приложение, предназначенное для отслеживания болельщиками результатов соревнований Олимпийских игр в Сочи. Главная страница приложения на рисунке 1.2:

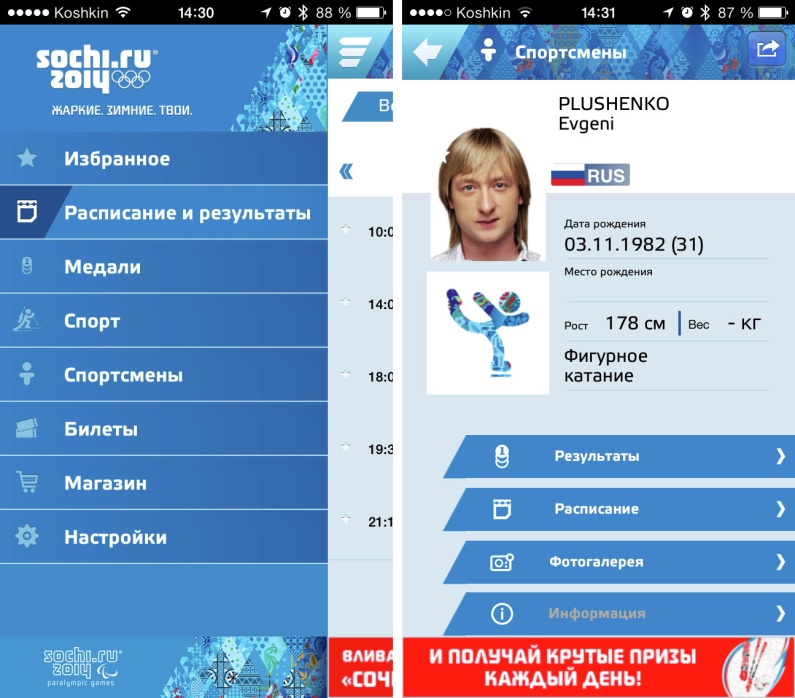


Рисунок 1.2 – Главная страница приложения «Сочи 2014: Результаты»

Достоинства «Сочи 2014: Результаты»:

– полная база данных результатов спортивных соревнований текущей Олимпиады;

– возможность просмотра расписания соревнований;

– возможность просмотра информации о спортсменах;

– просмотр медального зачета;

– покупка и бронирование билетов на спортивные мероприятия;

– возможность добавить любимые виды спорта в раздел «Избранное» для сокращения времени поиска;

– возможность просмотра подробной информации об интересующем виде спорта с загрузкой соответствующего мультимедийного контента;

– результаты уже прошедших соревнований не требуют доступа к интернету;

– приложение доступно для всех типов мобильных операционных систем.

Недостатки:

– приложение актуально только для конкретных Олимпийских игр;

– отсутствие расписание трансляций спортивных соревнований;

– отсутствие возможности просмотра личных и национальных достижений спортсменов либо команд.

«Сочи 2014 WOW (Wireless Olympic Works)» – официальное приложение от глобального партнера зимних Олимпийских игр в Сочи компании Samsung, предназначенное для просмотра результатов соревнований. Основной экран приложения прилагается на рисунке 1.3:

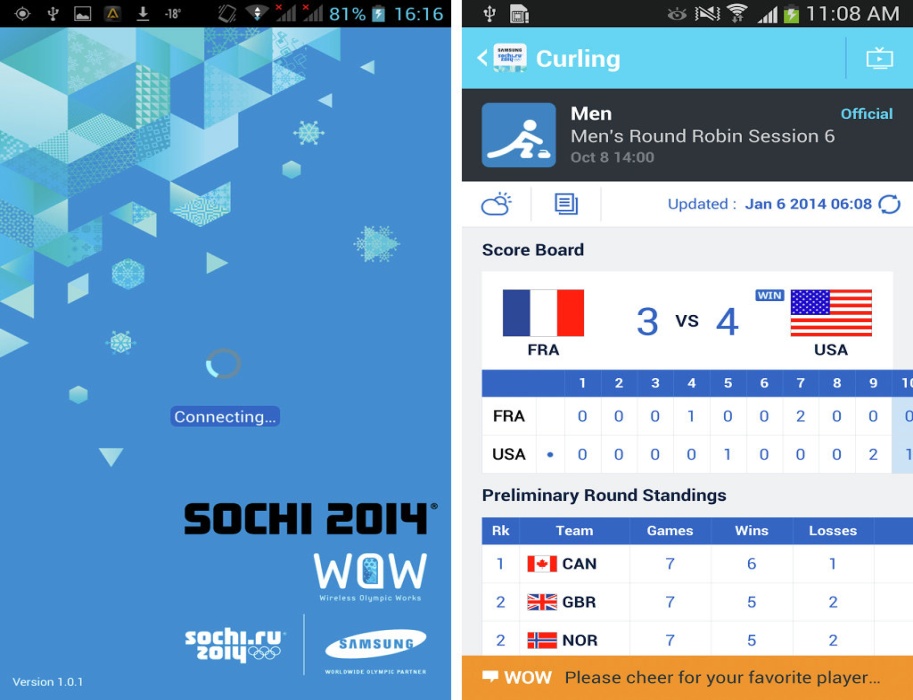


Рисунок 1.3 – Основной экран приложения «Сочи 2014 WOW»

Достоинства приложения «Сочи 2014 WOW»:

– наличие полного расписания спортивных соревнований;

– возможность просмотра результатов соревнований;

– интуитивно понятный интерфейс;

– добавление видов спорта в избранное;

– возможность редактирования интерфейса для более наглядного отображения видов спорта, занесенных в «Избранное»;

– возможность добавления напоминай к выбранным спортивным мероприятиям;

– возможность просмотра подробной информации о спортсменах;

– статистика завоеванных медалей каждой страной-участницей.

Недостатки приложения «Сочи 2014 WOW»:

– приложение предназначено только для операционной системы Android;

– отсутствие возможности просмотра истории выступления спортсменов;

– актуальность только для текущей Олимпиады.

Приложение «Зимние игры: Новости в режиме Live» является совместным продуктом РИА Новости и IT-компании Atos. Предназначено для предоставления официальных результатов и статистики Олимпийских игр с использованием мультимедийного контента РИА Новости. Меню приложения прилагается далее:

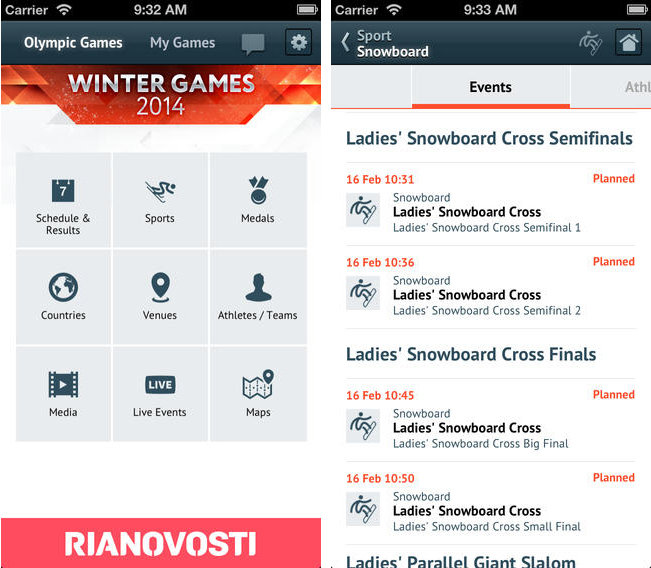
****

Рисунок 1.4 – Меню приложения «Зимние игры: Новости в режиме Live»

Достоинства приложения:

– возможность просмотра фото, видео и аудио репортажей с мест проведения олимпийских соревнований;

– календарь соревнований представлен в наиболее удобном пользователю виде;

– наличие карты спортивных мероприятий;

– возможность описания маршрута между олимпийскими объектами;

– доступна полная информация о спортсменах и видах спорта.

Недостатки приложения:

– приложение доступно только в двух языках;

– малая степень детализации спортивных результатов;

– отсутствие статистики выступлений;

– отсутствие персонализации пользователя.

* 1. Постановка задачи

В рамках данного курсового проекта планируется разработка базы данных Олимпийских игр.

Необходимо разработать инфологическую, даталогическую, физическую модель базы данных.

База данных должна обеспечивать возможность реализации следующих функциональных требований:

– просмотр олимпийских дисциплин;

– просмотр расписания Олимпийских игр;

– просмотр результатов спортивных соревнований;

– просмотр истории выступления спортсменов на различных Олимпийских играх;

– просмотр истории включения дисциплин в Олимпийскую программу;

– просмотр информации об олимпиаде: место и дата проведения, страны-участницы, спортивные дисциплины, количество зрителей;

– просмотр расписания телевизионных трансляций (включая прямые эфиры и повторы);

– просмотр истории выступления спортсмена за выбранную сборную;

– просмотр погодных условий во время спортивных соревнований;

– просмотр личных данных о спортсменах;

– просмотр заявленных на соревнования участников;

– просмотр информации о травмах спортсменов;

– привязка пользователя к приложению;

– организация различных степеней доступа к приложению;

– возможность добавления спортивных соревнований в раздел

«Избранное»;

– просмотр списка Олимпийских объектов с подробным адресом.

**2 ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ**



Создание инфологической модели базы данных заключается в описании предметной области без привязки к конкретной СУБД в наиболее понятной для восприятия пользователя форме.

Вся схема базы данных состоит из нескольких модулей:

Модуль Олимпийских игр

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Год Олимпиады | Дата | Сезон | Страна | Город | Количество участников | Количество дисциплин |
| Число | Дата | Текст | Текст | Текст | Число | Число |

Модуль спортсменов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фамилия | Имя | Отчество | Страна | Дисциплина | Дата рождения | Количество медалей |
| Текст | Текст | Текст | Текст | Текст | Дата | Число |

Модуль расписания соревнований

|  |  |  |  |
| --- | --- | --- | --- |
| Дисциплина | Этап | Дата | Место |
| Текст | Текст | Дата | Текст |

Модуль результатов соревнований

|  |  |  |  |
| --- | --- | --- | --- |
| Соревнование | Участник | Результат | Место |
| Текст | Текст | Число | Число |

Модуль пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Пользователь | Роль | Действия | Язык |
| Текст | Текст | Текст | Текст |

Модуль заявок на участие

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дисциплина | Этап | Участник | Дата | Время | Место |
| Текст | Текст | Текст | Дата | Дата | Текст |

**3 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

3. 1. Выбор СУБД

Одним из наиболее удобных инструментов для реализации данного программного средства является система управления базами данными MySQL. Для работы приложений на стороне клиента провайдеры предоставляют небольшое количество ресурсов. Поэтому для успешного функционирования приложения, взаимодействующего с разрабатываемой базой данных, необходима высокоэффективная СУБД, обладающая при этом высокой надежностью[2].

Основные преимущества MySQL:

– многопоточность, поддержка одновременных запросов;

– оптимизация связей с присоединением многих данных;

– записи фиксированной и переменной длины;

– ODBC драйвер;

– гибкая система привилегий и паролей;

– гибкая поддержка форматов чисел, строк переменной длины и меток времени;

– интерфейс с языками C и Perl, PHP;

– быстрая работа, масштабируемость;

– совместимость с ANSI SQL;

– СУБД в большинстве случаев бесплатна;

– поддержка со стороны провайдеров услуг хостинга;

– быстрая поддержка транзакций через механизм InnoDB.

Все это позволило MySQL стать эталоном в области СУБД для web-разработок. В настоящий момент развитие MySQL направлено на возможность ее использования в любых критичных бизнес-приложениях, что позволяет конкурировать на равных с такими производителями СУБД, как Oracle, Microsoft [3].

* 1. Спецификация функциональных требований

Одним из предъявляемых к разрабатываемому программному средству функциональных требований является возможность просмотра расписаний соревнований. Так как информация о проведении соревнований хранится на сервере в виде перечня не связанных между собой положений о проведении соревнований, для получения расписания необходимо применить соответствующий алгоритм, построение которого возможно на основании разработанной базы данных.

Спецификация данного функционального требования может иметь следующий вид:

– расписание соревнований представляется в виде таблиц;

– реализована возможность выбора дисциплины для составления соответсвующего расписания;

– выбор дисциплины осуществляется из выпадающего списка;

– есть возможность выбора нескольких дисциплин, соревнования по вновь добавленным дисциплинам выделяются другим цветом;

– должна быть предусмотрена возможность экспортирования сформированного расписания в документ формата .doc либо .exсel;

– графическое отображение в календаре выбранных соревнований;

– реализована возможность напоминания о предстоящих соревнованиях.

* 1. Разработка даталогической модели

Даталогическая модель – модель предметной области в привязке к СУБД определенного вида или конкретной СУБД. Описывает способ организации данных безотносительно их физического размещения.

Соглашения, распространяющиеся на всю работу по созданию даталогической модели:

– в таблице с описанием таблиц и полей БД тип данных указывается явно, если из названия полей и здравого смысла не очевидно, что нужно использовать «обычные» текстовые или числовые типы данных;

– все типы данных при создании полей таблиц БД формируются из расчёта совместимости с MySQL версии 5.5 и выше;

– имя первичного ключа формируется из названия таблицы в единственном числе и суффикcа «ID»;

– в именах таблиц существительные используются во множественном числе;

– имена полей и таблиц начинаются с маленьких букв;

– все поля, содержащие дату и/или время, имеют тип Date;

– все поля, содержащие названия категорий имеют тип Varchar (50), имеют тип Date.

– все поля, содержащие целые числа, имеют тип Integer [4].

Таблица 1 – Основные таблицы и поля, используемые в ходе проектирования базы данных.

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Основные поля** | **Описание** |
| persons | personID (int, PK)  firstName (varchar(100), not null)  lastName (varchar(100), not null) middleName (varchar(100)) birthdate (date, not null) genderID (int, FK, not null) | Определяет сущность Person, которой однозначно соответствует Фамилия, Имя, Отчество (необязательно к заполнению), дата рождения, гендер. |
| personinfohistory | personInfoID (int, PK) personID (int, FK)  weight (int, not null)  height (int, not null)  modifiedDate (timestamp, not null)  marriedID (int, FK)  disabilityID (int, FK) | Информация о человеке, которая может изменяться со временем. Предусмотрена возможность сохранения даты внесения изменений. |
| married | marriedID (int, PK)  status (varchar(20), not null) | Статус нахождения в браке: холост, женат, замужем, в разводе и т.д. |
| disabilitity | disabilitityID (int, PK)  status (varchar(50), not null) | Статус инвалидности и ее описание. |
| gender | genderID (int, PK)  type (varchar(10), not null) | Тип гендера: мужчина, женщина, не указан и др. |
| personinjuries | injuryID (int, PK)  injuryTypeID (int, FK)  dateStart (date, not null)  duration (date, not null)  injuryComment(text) | Полное описание полученной травмы спортсмена, включая ее тип. |
| injurytype | injuryTypeID (int, PK)  name (varchar(50), not null) | Тип полученной травмы. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| persondisqhistory | personID (int, PK)  startTime (date, PK) endTime(date)  disqReasonID(int, FK) | Таблица содержит периоды дисквалификации с привязкой к текущему времени. |
| disqreason | disqReasonID(int, PK)  reason (text, not null) | Причина дисквалификации. |
| countries | countryID (int, PK) name (varchar (50), not null) | Справочник стран. |
| cities | cityID (int, PK) name (varchar (50), not null)  countryID (int, FK) | Справочник городов. |
| sportobjectplaces | sportObjectPlaceID (int, PK) sportObjectID (int, FK) placeType (varchar (50), not null) name (varchar (50), not null) number (int) | Место проведения соревнований относится к одному спортивному объекту. Например, арена №2 на стадионе «Фишт». Типы – арена, бассейн, велотрек. |
| sportobjects | sportObjectsID (int, PK)  name (varchar (50), not null)  type (varchar (50), not null) capacity (int, not null)  olympicGamesID (int, FK) | Спортивный объект, который однозначно характеризуется адресом. Олимпийский объект относится только к одной Олимпиаде. Например, стадион «Фишт». Тип объекта – стадион, ледовый дворец, горнолыжный центр. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| sportobjectaddresses | sportObjectID (int, PK) cityID (int, FK)  streetName (varchar (50), not null)  buildingNumber (int, not null) latitude (float, not null) longitude (float, not null) | В таблице хранятся конкретные адреса, которые однозначно определяют месторасположение спортивного объекта. |
| sportapplication | applicationID (int, PK)  parent\_appID (int, FK) eventID (int, FK) personID (int, FK) countryID (int, FK) | Таблица, в которой фикcируются заявки на спортивные соревнования. parent\_appID – рекурсивный ключ, может быть Null только у одной заявки (командный вид спорта). Позволяет просмотреть участников команды. |
| olympicgames | olympicGameID (int, PK) cityID (int, FK) year (int, not null, unique) sequence (int, not null, unique)  seasonID (int, FK) | Таблица Олимпийских игр, описывающая год проведения, последовательность, тип сезона. |
| season | seasonID (int, PK)  name (varchar(50), not null) | Тип сезона Олимпийских игр (зимний, летний). |
| tvchannels | tvChannelsID (int, PK) name (varchar (50), not null) | Таблица телевизионных каналов. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| tvchannelsbroadcasts | tvChannelBroadcastID (int, PK)  eventID (int, FK) tvChannelID (int, FK) date (date, not null) timeStart (time, not null) timeEnd (time)  isLive (char(2), not null) | Таблица расписания телевизионных трансляций. Трансляция одного мероприятия может повторяться на одном канале в разное время. |
| users | userID (int, PK) login (varchar (50), mot null) password (varchar (50), not null) email (varchar (50), not null)  dateRegistration (date, not null)  dateBirth (date, not null) | Таблица пользователей системы и базы данных. |
| roles | roleID (int, PK) name (varchar (20), not null) | Таблица ролей (администратор. пользователь, модератор). |
| actions | actionID (int, PK) name (varchar (50), not null) | Таблица действий, которые будут относится к ролям. Добавить действие к пользователю напрямую нельзя. |
| languages | languageID (int, FK)  name (varchar (20), not null) | Таблица языков для возможности локализации. |
| sportkinds | sportKindID (int, PK) name (varchar (50), not null) | Справочник видов спорта. |
| sportdisciplines | sportDisciplineID (int, PK) name (varchar (50), not null) sportKindID (int, FK, null) | Справочник спортивных дисциплин. |
| sportcompetitions | sportCompetitionID (int, PK)  sportDisciplineID (int, FK) name (varchar (50), not null) | Справочник соревнований (вольные упражнения, прыжок). |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| sportevents | sportEventID (int, PK) competitionID (int, FK)  competitionPlaceID (int, FK)  date (date, not null) timeStart (time, not null) timeEnd (time)  stageID (int, FK) | Расписание спортивных соревнований. |
| stage | stageID (int, PK)  name (varchar(50), not null) | Этап соревнований (квалификация, финал). |
| sportresults | applicationID (int, PK) place (int, not null) isDisq (char (2), null)  isFail (char (2), null) failComment (text)  resultTypeID (int, FK)  medalTypeID (int, FK) | Таблица результатов спортивных событий. |
| resulttype | resultTypeID (int, PK)  name (varchar(10), not null) | Тип результата (победа, проигрыш, ничья). |
| medaltype | medalTypeID (int, PK)  name (varchar(10), not null) | Тип медали (золото, серебро, бронза). |
| documents | documentID (int, PK) title (varchar (50), not null) date (date, not null)  priority (int, not null) dbLoadingDate (timestamp) | Описание документов, в которых хранится информация о соревнованиях. |
| documentcontent | documentID (int, PK) content (Blob, not null) | Содержимое документа. |
| sponsors | sponsorID (int, PK)  name (varchar(50), not null)  tagline (text) | Информация о спонсоре. |
| olympictalismans | olympicTalismanID (int, PK)  name (varchar(50), not null)  description (text)  olympicGameID (int, FK) | Информация о талисманах Олимпийских игр |

* 1. Физическая модель базы данных

**Отношения**

Таблица 2 – отношения между сущностями

|  |  |  |
| --- | --- | --- |
| Отношение | Сущности | Комментарий |
| 1–1 | Документ – Содержание документа | Одному документы соответствует один контент. |
| Заявка – Результат | По одной заявке есть только 1 результат. |
| Спортивный объект – Адрес | По одному адресу расположен один спортивный объект. Одному адресу соответствует один Олимпийский объект. |
| Олимпийский талисман –Изображение | Олимпийский талисман имеет одно официальное изображение. |
| Погодные условия – Спортивное событие | Во время спортивного события информация о погоде получается 1 раз. |
| Спортивный объект – Олимпийские игры | Один спортивный объект относится к одним Олимпийским играм. |
| 1–many | Олимпийский объект – Место соревнований | На одном стадионе много спортивных площадок (несколько беговых дорожек, площадки для многоборцев). |
| Человек – Информация о человеке | Изменение семейного положения, физического состояния, состояния здоровья для конкретного человека. |
| Город – Страна | Один город относится к одной стране, в стране много крупных городов. |
| Трансляция соревнования – Телевизионный канал | На одном канале может идти много трансляций, но одна трансляция относится к одному каналу. |
| Дисциплина – Виды спорта | Одна дисциплина относится к одному виду, у одного вида может быть несколько дисциплин. |
| Спортивное соревнование – Дисциплина | Спортивная гимнастика – вольные упражнения, опорный прыжок. |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 1–many | Событие – Соревнование | Одному событию соответствует одно соревнование, но может быть несколько стадий соревнований, которые являются разными событиями. |
| Заявка – Событие | В одной заявке одно событие, но на одно событие несколько заявок. |
| many–to–many | Событие – Документ | Один документ может описывать несколько событий, одно событие описано в нескольких документах. |
| Роль – Действие | С одной ролью связано несколько действий, одно действие доступно нескольким ролям. |
| Роль – Пользователь | У пользователя может быть несколько ролей. Одна роль может относится к нескольким пользователям. |
| Пользователь – Язык | Пользователь может выбрать много языков, один язык могут выбирать многие пользователи. |
| Олимпийские игры –Спонсоры | Одним и те же компании могут являться генеральными спонсорами разных Олимпийских Игр. |
| Спортивные дисциплины – Олимпийские игры | На одних Олимпийских играх представлено множество дисциплин. Одна дисциплина представлена на многих Олимпийских играх |

**Индексы**

Таблица 3 – Индексы, используемые в базе данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Индекс** | **Таблица** | **Поле** | **Комментарий** |
| unique | user | login | В полях хранятся уникальные названия. |
| sportdiscipline | name |
| sportkind | name |
| country | name |
| city | name |
| awards | name |
| sportobject | name |
| ulanguage | name |
| uaction | name |
| urole | name |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| unique | tvchannels | name |  |
| sponsors | name |
| olympicGames | sequence | Порядковый номер Олимпиады уникальный. |
| Составной первичный / уникальный ключ | persondisciplinehistory | personID (PK) disciplineID (PK)  startDate (PK) | В одно время спортсмен может выступать только в одной дисциплине (биатлон). |
| olympicgames  participants | olympicGameID (PK) personID (PK) countryID (FK) disciplineID (FK) | На одних Олимпийских играх спортсмен может выступать в одной дисциплине и за одну страну. |
| sportapplications | applicationID (PK)  parent\_appID (FK) personID (unique) eventID (unique)  countryID (unique) | К одной заявке относится только один участник и событие. Если соревнование командное – заявку подает команда (страна). |
| personawardshistory | awardID (PK) date (PK) personID (FK) | В одно время награда может принадлежать одному спортсмену. |
| personinjurieshistory | personID (PK) injuryID (PK) | Один спортсмен не может получить одну и ту же травму в одно время, но может получить несколько травм одновременно. |

**Триггеры**

Таблица 4 – Триггеры, используемые в базе данных

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Комментарий** | **Триггер** |
| documents | Дата документа не должна превышать дату загрузки документа в систему. | DROP TRIGGER IF EXISTS  `reg\_date\_loading`;  delimiter //  CREATE TRIGGER  `reg\_date\_loading` BEFORE  INSERT ON `documents`  FOR EACH ROW  BEGIN  IF NEW.`dbLoadingDate` <  NEW.`date` THEN  SIGNAL SQLSTATE '45000' SET  MESSAGE\_TEXT = 'Loading date  can not be less or equal to dateDoc',  MYSQL\_ERRNO = 1001;  END IF;  END |
| uactions | Поле name может совпадать для действий, созданных разными пользователями, но не может  быть двух одноимённых  прав, созданных одним  и тем же пользователем. | DROP TRIGGER IF EXISTS  `unique\_uaction\_bi`;  delimiter //  CREATE TRIGGER  ` unique\_uaction\_bi ` BEFORE  INSERT ON `uAction`  FOR EACH ROW  BEGIN  DECLARE duplicates integer;  SELECT count(\*) INTO duplicates  from `uAction` where  `name`=NEW.`name` AND  (ISNULL(`userID`) OR  `userID`=NEW.`userID`); |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
|  | Также имя действия  (name), созданного  пользователем, не может совпадать с именем  «предустановленного  права». | IF duplicates > 0 THEN  SIGNAL SQLSTATE '45000' SET  MESSAGE\_TEXT = 'Action  name must be unique for one user and  must not be similar to predefined  name', MYSQL\_ERRNO = 1001;  END IF;  END. |

**4 ТЕСТИРОВАНИЕ**



Тестирование проводится с целью обеспечить качество разрабатываемого программного продукта [4]. Результаты тестирования программного средства представлены в таблице 5.

Таблица 5 – Результаты тестирования программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| **Описание теста** | **Ожидаемый результат** | **Полученный результат** | **Решение проблемы** |
| Добавление пользователя с существующим логином. | Ошибка ввода | Ввод пользователей с одинаковым логином. | Создание уникального индекса:  ALTER TABLE `users` ADD UNIQUE KEY `login` (`login`) |
| Добавление документа с более поздней датой, чем текущая. | Запрет на добавление документа в базу. | Добавление в базу данных документа с датой более поздней, чем текущая. | Добавление триггера trg\_doc\_date |
| Добавление пользователя с датой рождения большей, чем текущая. | Запрет на добавление пользователя в базу данных. | Пользователь добавлен. | Создание триггера trg\_user\_date |
| Добавление в таблицу olympicgames participants спортсмена, выступающего на одних ОИ за разные страны. | Запрет на добавление. | Добавление в базу данных. | Создание составного первичного ключа:  ADD PRIMARY KEY (`olympicGameID`, `personID`) |
| Добавление в таблицу persondisciplineshistory спортсмена, выступающего в одно время в разных спортивных дисциплинах. | Запрет на добавление. | Добавление в базу данных. | Создание составного первичного ключа: ADD PRIMARY KEY (`startDate`,`personID`) |
| Добавление информации об Олимпийских играх с существующим порядковым номером. | Запрет на введение информации. | Новые данные о существующих Олимпийский играх добавлены. | Создание уникального индекса на поле sequence: ALTER TABLE `olympicGames`  ADD UNIQUE KEY `sequence` (`sequence `) |

В результате тестирования был выявлен ряд ошибок, для исправления которых понадобилось внести изменения в схему данных, изменить тип связей, установить необходимые индексы, добавить триггеры.

**5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Комментарии для заполнения базы данными приведены в таблице 6.

Таблица 6 – Комментарии для базы данных «Олимпийские игры»

|  |  |  |
| --- | --- | --- |
| **Таблица.Поле** | **Возможные значения** | **Комментарий** |
| sportresults.isFail | 1,0 | 1 – если в результате всех попыток спортсмен не преодолел нужный рубеж |
| sportresults.isDisq | 1,0 | 1 – cпортсмен дисквалифицирован |
| sportresults.isRecord | 1,0 | 1 – результат засчитан как новый Олимпийский рекорд |
| sportcompetition.isCommand | 1,0 | 1 – командное соревнование |
| olympicGames.sequence | 1..n | Порядковый номер Олимпийских игр. Является уникальным. |
| persondisqhistory.dateEnd | NULL | NULL – если спортсмен до сих пор дисквалифицирован |

Данные сведения помогут избежать ошибок при использовании базы данных для разработчиков ПО.

**ЗАКЛЮЧЕНИЕ**

Разработанная в рамках данного курсового проекта база данных «Олимпийские игры» является универсальным средством для автоматизации данных, получаемых в результате проведения Олимпийских игр. Разработанная база данных отличается гибкостью архитектуры, кроссплатформенностью, высокой надежностью и производительностью.

Таким образом, в ходе выполнения данного курсового проекта была разработана полностью функционирующая база данных, с помощью которой могут быть реализованы функции упорядоченного хранения данных о результатах спортивных соревнований, олимпийских чемпионах, истории включения спортивных дисциплин в Олимпийскую программу. Были получены более глубокие практические знания в области программирования на языке Sql, а также администрирования СУБД MySQL.

Были реализованые следующие функциональные требования:

– просмотр олимпийских дисциплин;

– просмотр расписания Олимпийских игр;

– просмотр результатов спортивных соревнований;

– просмотр истории выступления спортсменов на различных Олимпийских играх;

– просмотр истории включения дисциплин в Олимпийскую программу;

– просмотр информации об олимпиаде: место и дата проведения, страны-участницы, спортивные дисциплины, количество зрителей;

– просмотр расписания телевизионных трансляций (включая прямые эфиры и повторы);

– просмотр истории выступления спортсмена за выбранную сборную;

– просмотр погодных условий во время спортивных соревнований;

– просмотр личных данных о спортсменах;

– просмотр заявленных на соревнования участников;

– просмотр информации о травмах спортсменов;

– привязка пользователя к приложению;

– организация различных степеней доступа к приложению;

– возможность добавления спортивных соревнований в раздел

«Избранное»;

– просмотр списка Олимпийских объектов с подробным адресом.

**СПИСОК ЛИТЕРАТУРЫ**

1. Olympic Games, Medals, Results, Latests News [Электронный ресурс] – Режим доступа: https://olympics.com
2. Аткинсон, Л. MySQL. Библиотека профессионала / Л. Аткинсон // Пер. с англ. – М.: Издательский дом «Вильяме», 2002 – 624 с.
3. MSDN [Электронный ресурс] – Режим доступа: http://msdn.microsoft.com/en-us/MySQL.aspx.
4. Wikipedia [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Mysql/phpMyadmin>.
5. Майерс Г., Искусство тестирования программ – М.: Финансы и статистика, 1984. – 174 с.

**ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ**

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema olympic\_games

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema olympic\_games

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `olympic\_games` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci ;

USE `olympic\_games` ;

-- -----------------------------------------------------

-- Table `olympic\_games`.`countries`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`countries` (

`countryID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`countryID`),

INDEX `name` (`name` ASC) VISIBLE)

ENGINE = InnoDB

AUTO\_INCREMENT = 2

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`cities`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`cities` (

`cityID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`countryID` INT NOT NULL,

PRIMARY KEY (`cityID`),

INDEX `countryID` (`countryID` ASC) VISIBLE,

INDEX `name` (`name` ASC) VISIBLE,

CONSTRAINT `cities\_ibfk\_1`

FOREIGN KEY (`countryID`)

REFERENCES `olympic\_games`.`countries` (`countryID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

AUTO\_INCREMENT = 2

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`season`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`season` (

`seasonID` INT UNSIGNED NOT NULL,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`seasonID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympicgames`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympicgames` (

`olympicGameID` INT NOT NULL AUTO\_INCREMENT,

`cityID` INT NOT NULL,

`year` INT NOT NULL,

`sequnce` INT NOT NULL,

`seasonID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`olympicGameID`, `seasonID`),

UNIQUE INDEX `sequnce` (`sequnce` ASC) VISIBLE,

INDEX `cityID` (`cityID` ASC) VISIBLE,

INDEX `fk\_olympicgames\_season1\_idx` (`seasonID` ASC) VISIBLE,

UNIQUE INDEX `year\_UNIQUE` (`year` ASC) VISIBLE,

CONSTRAINT `olympicgames\_ibfk\_1`

FOREIGN KEY (`cityID`)

REFERENCES `olympic\_games`.`cities` (`cityID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_olympicgames\_season1`

FOREIGN KEY (`seasonID`)

REFERENCES `olympic\_games`.`season` (`seasonID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 2

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportkinds`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportkinds` (

`sportKindID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`sportKindID`),

INDEX `name` (`name` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportdisciplines`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportdisciplines` (

`sportDisciplineID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`sportKindID` INT NULL DEFAULT NULL,

PRIMARY KEY (`sportDisciplineID`),

INDEX `sportKindID` (`sportKindID` ASC) VISIBLE,

INDEX `name` (`name` ASC) VISIBLE,

CONSTRAINT `sportdiciplines\_ibfk\_1`

FOREIGN KEY (`sportKindID`)

REFERENCES `olympic\_games`.`sportkinds` (`sportKindID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympicgamesdiciplines`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympicgamesdiciplines` (

`olympicGameID` INT NOT NULL,

`diciplineID` INT NOT NULL,

PRIMARY KEY (`olympicGameID`, `diciplineID`),

INDEX `diciplineID` (`diciplineID` ASC) VISIBLE,

CONSTRAINT `olympicgamesdiciplines\_ibfk\_1`

FOREIGN KEY (`olympicGameID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`)

ON UPDATE CASCADE,

CONSTRAINT `olympicgamesdiciplines\_ibfk\_2`

FOREIGN KEY (`diciplineID`)

REFERENCES `olympic\_games`.`sportdisciplines` (`sportDisciplineID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympicgameslogos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympicgameslogos` (

`olympicGameID` INT NOT NULL AUTO\_INCREMENT,

`nameLogo` VARCHAR(50) NOT NULL,

`imageLogo` BLOB NOT NULL,

PRIMARY KEY (`olympicGameID`),

CONSTRAINT `olympicgameslogos\_ibfk\_1`

FOREIGN KEY (`olympicGameID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`gender`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`gender` (

`genderID` INT UNSIGNED NOT NULL,

`type` VARCHAR(10) NOT NULL,

PRIMARY KEY (`genderID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`persons`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`persons` (

`personID` INT NOT NULL AUTO\_INCREMENT,

`firstName` VARCHAR(100) NOT NULL,

`lastName` VARCHAR(100) NOT NULL,

`middleName` VARCHAR(100) NULL DEFAULT NULL,

`dateBirth` DATE NOT NULL,

`genderID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`personID`, `genderID`),

INDEX `firstName` (`firstName` ASC) VISIBLE,

INDEX `fk\_persons\_gender1\_idx` (`genderID` ASC) VISIBLE,

CONSTRAINT `fk\_persons\_gender1`

FOREIGN KEY (`genderID`)

REFERENCES `olympic\_games`.`gender` (`genderID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympicgamesparticipants`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympicgamesparticipants` (

`olympicGameID` INT NOT NULL,

`personID` INT NOT NULL,

`countryID` INT NOT NULL,

`disciplineID` INT NOT NULL,

PRIMARY KEY (`olympicGameID`, `personID`),

INDEX `countryID` (`countryID` ASC) VISIBLE,

INDEX `diciplineID` (`disciplineID` ASC) VISIBLE,

INDEX `personID` (`personID` ASC) VISIBLE,

CONSTRAINT `olympicgamesparticipants\_ibfk\_1`

FOREIGN KEY (`olympicGameID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`)

ON UPDATE CASCADE,

CONSTRAINT `olympicgamesparticipants\_ibfk\_2`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE,

CONSTRAINT `olympicgamesparticipants\_ibfk\_3`

FOREIGN KEY (`countryID`)

REFERENCES `olympic\_games`.`countries` (`countryID`)

ON UPDATE CASCADE,

CONSTRAINT `olympicgamesparticipants\_ibfk\_4`

FOREIGN KEY (`disciplineID`)

REFERENCES `olympic\_games`.`sportdisciplines` (`sportDisciplineID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sponsors`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sponsors` (

`sponsorID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`tagline` TEXT NOT NULL,

PRIMARY KEY (`sponsorID`))

ENGINE = InnoDB

AUTO\_INCREMENT = 3

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympicgamessponsors`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympicgamessponsors` (

`olympicGameID` INT NOT NULL,

`sponsorID` INT NOT NULL,

PRIMARY KEY (`olympicGameID`, `sponsorID`),

INDEX `sponsorID` (`sponsorID` ASC) VISIBLE,

CONSTRAINT `olympicgamessponsors\_ibfk\_1`

FOREIGN KEY (`olympicGameID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`)

ON UPDATE CASCADE,

CONSTRAINT `olympicgamessponsors\_ibfk\_2`

FOREIGN KEY (`sponsorID`)

REFERENCES `olympic\_games`.`sponsors` (`sponsorID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympictalismanlogos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympictalismanlogos` (

`olympicTalismanID` INT NOT NULL,

`image` BLOB NOT NULL,

PRIMARY KEY (`olympicTalismanID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`olympictalismans`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`olympictalismans` (

`olympicTalismanID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`description` TEXT NOT NULL,

`olympicGameID` INT NOT NULL,

PRIMARY KEY (`olympicTalismanID`),

INDEX `olympicGameID` (`olympicGameID` ASC) VISIBLE,

CONSTRAINT `olympictalismans\_ibfk\_1`

FOREIGN KEY (`olympicGameID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`),

CONSTRAINT `olympictalismans\_ibfk\_2`

FOREIGN KEY (`olympicTalismanID`)

REFERENCES `olympic\_games`.`olympictalismanlogos` (`olympicTalismanID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`documents`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`documents` (

`documentID` INT NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(50) NOT NULL,

`date` DATE NOT NULL,

`priority` INT NOT NULL,

`dbLoadingDate` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`documentID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`documentcontent`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`documentcontent` (

`documentID` INT NOT NULL,

`content` BLOB NOT NULL,

PRIMARY KEY (`documentID`),

CONSTRAINT `otherdocumentcontent\_ibfk\_1`

FOREIGN KEY (`documentID`)

REFERENCES `olympic\_games`.`documents` (`documentID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportcompetitions`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportcompetitions` (

`sportCompetitionID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`sportDisciplineID` INT NOT NULL,

`isCommand` CHAR(2) NOT NULL,

PRIMARY KEY (`sportCompetitionID`),

INDEX `sportDiciplineID` (`sportDisciplineID` ASC) VISIBLE,

INDEX `name` (`name` ASC) VISIBLE,

CONSTRAINT `sportcompetitions\_ibfk\_1`

FOREIGN KEY (`sportDisciplineID`)

REFERENCES `olympic\_games`.`sportdisciplines` (`sportDisciplineID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportobjects`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportobjects` (

`sportObjectID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`type` VARCHAR(50) NOT NULL,

`capacity` INT NOT NULL,

`olympicGamesID` INT NOT NULL,

PRIMARY KEY (`sportObjectID`),

INDEX `name` (`name` ASC) VISIBLE,

INDEX `olympicGamesID` (`olympicGamesID` ASC) VISIBLE,

CONSTRAINT `sportobjects\_ibfk\_1`

FOREIGN KEY (`olympicGamesID`)

REFERENCES `olympic\_games`.`olympicgames` (`olympicGameID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportobjectplaces`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportobjectplaces` (

`sportObjectPlaceID` INT NOT NULL AUTO\_INCREMENT,

`sportObjectID` INT NOT NULL,

`placeType` VARCHAR(50) NOT NULL,

`number` INT NULL DEFAULT NULL,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`sportObjectPlaceID`),

INDEX `sportObjectID` (`sportObjectID` ASC) VISIBLE,

CONSTRAINT `sportobjectplaces\_ibfk\_1`

FOREIGN KEY (`sportObjectID`)

REFERENCES `olympic\_games`.`sportobjects` (`sportObjectID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`stage`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`stage` (

`stageID` INT UNSIGNED NOT NULL,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`stageID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportevents`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportevents` (

`sportEventID` INT NOT NULL AUTO\_INCREMENT,

`competitionID` INT NOT NULL,

`competitionPlaceID` INT NOT NULL,

`date` DATE NOT NULL,

`timeStart` TIME NOT NULL,

`timeEnd` TIME NULL DEFAULT NULL,

`stageID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`sportEventID`, `stageID`),

INDEX `competitionID` (`competitionID` ASC) VISIBLE,

INDEX `competitionPlaceID` (`competitionPlaceID` ASC) VISIBLE,

INDEX `date` (`date` ASC) VISIBLE,

INDEX `fk\_sportevents\_stage1\_idx` (`stageID` ASC) VISIBLE,

CONSTRAINT `sportevents\_ibfk\_1`

FOREIGN KEY (`competitionID`)

REFERENCES `olympic\_games`.`sportcompetitions` (`sportCompetitionID`)

ON UPDATE CASCADE,

CONSTRAINT `sportevents\_ibfk\_2`

FOREIGN KEY (`competitionPlaceID`)

REFERENCES `olympic\_games`.`sportobjectplaces` (`sportObjectPlaceID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_sportevents\_stage1`

FOREIGN KEY (`stageID`)

REFERENCES `olympic\_games`.`stage` (`stageID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`documentevent`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`documentevent` (

`documentID` INT NOT NULL,

`eventID` INT NOT NULL,

PRIMARY KEY (`documentID`, `eventID`),

INDEX `eventID` (`eventID` ASC) VISIBLE,

CONSTRAINT `otherdocumentevent\_ibfk\_1`

FOREIGN KEY (`documentID`)

REFERENCES `olympic\_games`.`documents` (`documentID`)

ON UPDATE CASCADE,

CONSTRAINT `otherdocumentevent\_ibfk\_2`

FOREIGN KEY (`eventID`)

REFERENCES `olympic\_games`.`sportevents` (`sportEventID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`personawards`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`personawards` (

`awardID` INT NOT NULL AUTO\_INCREMENT,

`name` INT NOT NULL,

`disciplineID` INT NOT NULL,

PRIMARY KEY (`awardID`),

UNIQUE INDEX `name` (`name` ASC) VISIBLE,

INDEX `diciplineID` (`disciplineID` ASC) VISIBLE,

CONSTRAINT `personawards\_ibfk\_1`

FOREIGN KEY (`disciplineID`)

REFERENCES `olympic\_games`.`sportdisciplines` (`sportDisciplineID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`personawardshistory`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`personawardshistory` (

`personID` INT NOT NULL,

`awardID` INT NOT NULL,

`date` DATE NOT NULL,

`currency` VARCHAR(50) NULL DEFAULT NULL,

PRIMARY KEY USING BTREE (`awardID`, `date`),

INDEX `personID` (`personID` ASC) VISIBLE,

CONSTRAINT `personawardshistory\_ibfk\_2`

FOREIGN KEY (`awardID`)

REFERENCES `olympic\_games`.`personawards` (`awardID`)

ON UPDATE CASCADE,

CONSTRAINT `personawardshistory\_ibfk\_3`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`persondiciplineshistory`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`persondiciplineshistory` (

`personID` INT NOT NULL,

`diciplineID` INT NOT NULL,

`startDate` DATE NOT NULL,

PRIMARY KEY (`personID`, `diciplineID`, `startDate`),

INDEX `diciplineID` (`diciplineID` ASC) VISIBLE,

CONSTRAINT `persondiciplineshistory\_ibfk\_1`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE,

CONSTRAINT `persondiciplineshistory\_ibfk\_2`

FOREIGN KEY (`diciplineID`)

REFERENCES `olympic\_games`.`sportdisciplines` (`sportDisciplineID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`disqreason`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`disqreason` (

`disqreasonID` INT UNSIGNED NOT NULL,

`reason` TEXT NOT NULL,

PRIMARY KEY (`disqreasonID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`persondisqhistory`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`persondisqhistory` (

`personID` INT NOT NULL,

`startDate` DATE NOT NULL,

`endDate` DATE NULL DEFAULT NULL,

`disqReasonID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`personID`, `startDate`, `disqReasonID`),

INDEX `fk\_persondisqhstr\_disqreason1\_idx` (`disqReasonID` ASC) VISIBLE,

CONSTRAINT `persondisqhstr\_ibfk\_1`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_persondisqhstr\_disqreason1`

FOREIGN KEY (`disqReasonID`)

REFERENCES `olympic\_games`.`disqreason` (`disqreasonID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`married`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`married` (

`marriedID` INT UNSIGNED NOT NULL,

`status` VARCHAR(20) NOT NULL,

PRIMARY KEY (`marriedID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`disability`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`disability` (

`disabilityID` INT UNSIGNED NOT NULL,

`status` VARCHAR(50) NOT NULL,

PRIMARY KEY (`disabilityID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`personinfohistory`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`personinfohistory` (

`personInfoID` INT NOT NULL AUTO\_INCREMENT,

`personID` INT NOT NULL,

`height` INT NOT NULL,

`weight` INT NOT NULL,

`modifiedDate` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`marriedID` INT UNSIGNED NOT NULL,

`disabilityID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`personInfoID`, `marriedID`, `disabilityID`),

INDEX `personID` (`personID` ASC) VISIBLE,

INDEX `fk\_personinfohistory\_married1\_idx` (`marriedID` ASC) VISIBLE,

INDEX `fk\_personinfohistory\_disability1\_idx` (`disabilityID` ASC) VISIBLE,

CONSTRAINT `personinfohistory\_ibfk\_1`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_personinfohistory\_married1`

FOREIGN KEY (`marriedID`)

REFERENCES `olympic\_games`.`married` (`marriedID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_personinfohistory\_disability1`

FOREIGN KEY (`disabilityID`)

REFERENCES `olympic\_games`.`disability` (`disabilityID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`injurytype`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`injurytype` (

`idinjuryTypeID` INT UNSIGNED NOT NULL,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`idinjuryTypeID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`personinjuries`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`personinjuries` (

`injuryID` INT NOT NULL AUTO\_INCREMENT,

`idinjuryTypeID` INT UNSIGNED NOT NULL,

`dateStart` DATE NOT NULL,

`duration` DATE NOT NULL,

`injuryComment` TEXT NOT NULL,

PRIMARY KEY (`injuryID`, `idinjuryTypeID`),

INDEX `fk\_personinjuries\_injurytype1\_idx` (`idinjuryTypeID` ASC) VISIBLE,

CONSTRAINT `fk\_personinjuries\_injurytype1`

FOREIGN KEY (`idinjuryTypeID`)

REFERENCES `olympic\_games`.`injurytype` (`idinjuryTypeID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`personinjurieshistory`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`personinjurieshistory` (

`injuryID` INT NOT NULL,

`personID` INT NOT NULL,

PRIMARY KEY (`injuryID`, `personID`),

INDEX `personID` (`personID` ASC) VISIBLE,

CONSTRAINT `personinjurieshistory\_ibfk\_1`

FOREIGN KEY (`injuryID`)

REFERENCES `olympic\_games`.`personinjuries` (`injuryID`)

ON UPDATE CASCADE,

CONSTRAINT `personinjurieshistory\_ibfk\_2`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportapplications`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportapplications` (

`applicationID` INT NOT NULL AUTO\_INCREMENT,

`parent\_appID` INT NULL DEFAULT NULL,

`eventID` INT NOT NULL,

`personID` INT NULL DEFAULT NULL,

`countryID` INT NULL DEFAULT NULL,

PRIMARY KEY (`applicationID`),

INDEX `eventID` (`eventID` ASC) VISIBLE,

INDEX `parent\_appID` (`parent\_appID` ASC) VISIBLE,

INDEX `personID` (`personID` ASC) VISIBLE,

INDEX `countryID` (`countryID` ASC) VISIBLE,

CONSTRAINT `sportapplications\_ibfk\_1`

FOREIGN KEY (`parent\_appID`)

REFERENCES `olympic\_games`.`sportapplications` (`applicationID`)

ON UPDATE CASCADE,

CONSTRAINT `sportapplications\_ibfk\_2`

FOREIGN KEY (`eventID`)

REFERENCES `olympic\_games`.`sportevents` (`sportEventID`)

ON UPDATE CASCADE,

CONSTRAINT `sportapplications\_ibfk\_3`

FOREIGN KEY (`personID`)

REFERENCES `olympic\_games`.`persons` (`personID`)

ON UPDATE CASCADE,

CONSTRAINT `sportapplications\_ibfk\_4`

FOREIGN KEY (`countryID`)

REFERENCES `olympic\_games`.`countries` (`countryID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportobjectaddresses`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportobjectaddresses` (

`sportObjectID` INT NOT NULL,

`cityID` INT NOT NULL,

`streetName` VARCHAR(50) NOT NULL,

`buildingNumber` INT NOT NULL,

`longitute` FLOAT NOT NULL,

`latitude` FLOAT NOT NULL,

PRIMARY KEY (`sportObjectID`),

INDEX `cityID` (`cityID` ASC) VISIBLE,

CONSTRAINT `sportobjectaddresses\_ibfk\_1`

FOREIGN KEY (`sportObjectID`)

REFERENCES `olympic\_games`.`sportobjects` (`sportObjectID`)

ON UPDATE CASCADE,

CONSTRAINT `sportobjectaddresses\_ibfk\_2`

FOREIGN KEY (`cityID`)

REFERENCES `olympic\_games`.`cities` (`cityID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`medaltype`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`medaltype` (

`medalTypeID` INT UNSIGNED NOT NULL,

`name` VARCHAR(10) NOT NULL,

PRIMARY KEY (`medalTypeID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`resulttype`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`resulttype` (

`resultTypeID` INT UNSIGNED NOT NULL,

`name` VARCHAR(10) NOT NULL,

PRIMARY KEY (`resultTypeID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`sportresults`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`sportresults` (

`applicationID` INT NOT NULL,

`place` INT NOT NULL,

`isDisq` CHAR(2) NOT NULL,

`isFail` CHAR(2) NOT NULL,

`failComment` TEXT NULL DEFAULT NULL,

`isRecord` CHAR(2) NOT NULL,

`medalTypeID` INT UNSIGNED NOT NULL,

`resultTypeID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`applicationID`, `medalTypeID`, `resultTypeID`),

INDEX `fk\_sportresults\_medaltype1\_idx` (`medalTypeID` ASC) VISIBLE,

INDEX `fk\_sportresults\_resulttype1\_idx` (`resultTypeID` ASC) VISIBLE,

CONSTRAINT `sportresults\_ibfk\_1`

FOREIGN KEY (`applicationID`)

REFERENCES `olympic\_games`.`sportapplications` (`applicationID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_sportresults\_medaltype1`

FOREIGN KEY (`medalTypeID`)

REFERENCES `olympic\_games`.`medaltype` (`medalTypeID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_sportresults\_resulttype1`

FOREIGN KEY (`resultTypeID`)

REFERENCES `olympic\_games`.`resulttype` (`resultTypeID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`tvchannels`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`tvchannels` (

`tvChannelID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

PRIMARY KEY (`tvChannelID`),

INDEX `name` (`name` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`tvchannelsbroadcasts`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`tvchannelsbroadcasts` (

`tvChannelsBroadcastID` INT NOT NULL AUTO\_INCREMENT,

`tvChannelID` INT NOT NULL,

`eventID` INT NOT NULL,

`date` DATE NOT NULL,

`timeStart` TIME NOT NULL,

`timeEnd` TIME NULL DEFAULT NULL,

`isLive` CHAR(2) NOT NULL,

PRIMARY KEY (`tvChannelsBroadcastID`),

INDEX `tvChannelID` (`tvChannelID` ASC) VISIBLE,

INDEX `eventID` (`eventID` ASC) VISIBLE,

CONSTRAINT `tvchannelsbroadcasts\_ibfk\_1`

FOREIGN KEY (`tvChannelID`)

REFERENCES `olympic\_games`.`tvchannels` (`tvChannelID`)

ON UPDATE CASCADE,

CONSTRAINT `tvchannelsbroadcasts\_ibfk\_2`

FOREIGN KEY (`eventID`)

REFERENCES `olympic\_games`.`sportevents` (`sportEventID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`users`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`users` (

`userID` INT NOT NULL AUTO\_INCREMENT,

`login` VARCHAR(50) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`email` VARCHAR(50) NOT NULL,

`dateBirth` DATE NOT NULL,

`dateRegistration` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`userID`),

UNIQUE INDEX `login` (`login` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`uactions`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`uactions` (

`actionID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(50) NOT NULL,

`userID` INT NOT NULL,

PRIMARY KEY (`actionID`),

UNIQUE INDEX `name` (`name` ASC) VISIBLE,

INDEX `fk\_uactions\_users1\_idx` (`userID` ASC) VISIBLE,

CONSTRAINT `fk\_uactions\_users1`

FOREIGN KEY (`userID`)

REFERENCES `olympic\_games`.`users` (`userID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`ulanguages`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`ulanguages` (

`languageID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(20) NOT NULL,

PRIMARY KEY (`languageID`),

UNIQUE INDEX `name` (`name` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`uroles`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`uroles` (

`roleID` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(20) NOT NULL,

PRIMARY KEY (`roleID`),

UNIQUE INDEX `name` (`name` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`uroleactions`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`uroleactions` (

`roleID` INT NOT NULL,

`actionID` INT NOT NULL,

PRIMARY KEY (`roleID`, `actionID`),

INDEX `actionID` (`actionID` ASC) VISIBLE,

CONSTRAINT `uroleactions\_ibfk\_1`

FOREIGN KEY (`roleID`)

REFERENCES `olympic\_games`.`uroles` (`roleID`)

ON UPDATE CASCADE,

CONSTRAINT `uroleactions\_ibfk\_2`

FOREIGN KEY (`actionID`)

REFERENCES `olympic\_games`.`uactions` (`actionID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`userlanguage`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`userlanguage` (

`userID` INT NOT NULL,

`languageID` INT NOT NULL,

PRIMARY KEY (`userID`, `languageID`),

INDEX `languageID` (`languageID` ASC) VISIBLE,

CONSTRAINT `userlanguage\_ibfk\_1`

FOREIGN KEY (`userID`)

REFERENCES `olympic\_games`.`users` (`userID`)

ON UPDATE CASCADE,

CONSTRAINT `userlanguage\_ibfk\_2`

FOREIGN KEY (`languageID`)

REFERENCES `olympic\_games`.`ulanguages` (`languageID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`userroles`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`userroles` (

`userID` INT NOT NULL,

`roleID` INT NOT NULL,

PRIMARY KEY (`userID`, `roleID`),

INDEX `roleID` (`roleID` ASC) VISIBLE,

CONSTRAINT `userroles\_ibfk\_1`

FOREIGN KEY (`userID`)

REFERENCES `olympic\_games`.`users` (`userID`)

ON UPDATE CASCADE,

CONSTRAINT `userroles\_ibfk\_2`

FOREIGN KEY (`roleID`)

REFERENCES `olympic\_games`.`uroles` (`roleID`)

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

-- -----------------------------------------------------

-- Table `olympic\_games`.`winddirection`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`winddirection` (

`winddirectionID` INT UNSIGNED NOT NULL,

`name` VARCHAR(10) NOT NULL,

PRIMARY KEY (`winddirectionID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `olympic\_games`.`weatherconditions`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `olympic\_games`.`weatherconditions` (

`eventID` INT NOT NULL,

`temperature` INT NOT NULL,

`snowLevel` INT NULL DEFAULT NULL,

`windSpeed` INT NOT NULL,

`waterLevel` INT NULL DEFAULT NULL,

`preassure` INT NOT NULL,

`date` DATETIME NOT NULL,

`winddirectionID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`eventID`, `winddirectionID`),

INDEX `fk\_weatherconditions\_winddirection1\_idx` (`winddirectionID` ASC) VISIBLE,

CONSTRAINT `weatherconditions\_ibfk\_1`

FOREIGN KEY (`eventID`)

REFERENCES `olympic\_games`.`sportevents` (`sportEventID`)

ON UPDATE CASCADE,

CONSTRAINT `fk\_weatherconditions\_winddirection1`

FOREIGN KEY (`winddirectionID`)

REFERENCES `olympic\_games`.`winddirection` (`winddirectionID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb3;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;