Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни “Бази даних”

“Інформаційна система «Клуб любителів бігу»”

Керівник: доц. каф. ПІ Мазурова О.О.

Студент гр. ПІ - 15 – 1 Деревнін О.О.

Комісія:

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мазурова О.О.

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю.

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Широкопетлєва М.С.

Харків 2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Харківський національний університет радіоелектроніки\_\_\_\_\_\_\_\_\_\_

**Кафедра** \_\_\_Програмної інженерії\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дисципліна** \_\_\_\_Бази даних\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Спеціальність** \_\_\_\_\_Програмна інженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Курс** \_\_\_2\_\_\_\_\_ **Група** \_\_\_ПІ-15-1\_\_\_ **Семестр**\_\_\_\_ 3\_\_\_\_

**ЗАВДАННЯ**

на курсову роботу студента

*Деревніна Олександра Олександровича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. **Тема роботи**: «Інформаційна система «*Клуб любителів бігу*»»
2. **Строк здачі закінченої роботи**: \_\_\_\_10.01.2017\_\_\_\_\_\_\_\_
3. **Вихідні дані для роботи**: методичні вказівки до виконання курсової роботи, вимоги до інформаційної системи, предметна область, що пов’язана з обліком творами скульптури. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. **Зміст розрахунково - пояснювальної записки**: вступ, аналіз предметної області; постановка задачі; проектування бази даних; опис програми; висновки; перелік посилань.
5. **Перелік графічного матеріалу**: загальна схема концептуальної моделі, ER-діаграма, структура 1НФ, 2НФ, 3НФ, схема БД в 3НФ, UML-діаграми, копії екранів (“скріншоти”) прикладної програми, приклади звітів прикладної програми.\_\_\_
6. **Дата видачі завдання**: \_\_09.09.16 р.\_\_\_

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Аналіз предметної області | 9.09.16 – 24.09.16 | Виконано |
| 2 | Постановка задачі | 20.09.16 – 30.09.16 | Виконано |
| 3 | Побудова ER-діаграми бази даних | 27.09.16 – 15.10.16 | Виконано |
| 4 | Оформлення розділів 1, 2 та 3.1, 3.2 пояснювальної записки | 15.10.16 - 27.10.16 | Виконано |
| 5 | Перша контрольна точка з курсового проекту | 24.10.16 – 28.10.16 | Виконано |
| 6 | Нормалізація бази даних | 20.10.16 - 10.11.16 | Виконано |
| 7 | Створення демо-версії програми | 20.10.16 – 20.11.16 | Виконано |
| 8 | Тестування програми, наповнення бази даних | 15.11.16 - 25.11.16 | Виконано |
| 9 | Друга контрольна точка з курсового проекту | 21.12.16– 02.12.16 | Виконано |
| 10 | Реалізація остаточної версії програми | 1.12.16-15.12.16 | Виконано |
| 11 | Оформлення інших розділів пояснювальної записки | 1.11.16 – 15.12.16 | Виконано |
| 12 | Захист курсового проекту (третя контрольна точка) | 12.12.16- 27.12.16 | Виконано |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник \_\_\_\_ *доц. Мазурова О.О..*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 р.

ЗМІСТ

1. Аналіз та концептуальне моделювання предметної області…..……………7

1.1 Аналіз предметної області…………………………………..……………….7

1.1 Концептуальне моделювання предметної області…….................................8

2. Постановка задачі………………….…………………………………………16

3. Проектування бази даних…………………………………………...……....18

3.1 UML-моделювання……………………………………………...…………18

3.2 Побудова ER-діаграми…………………………………..............................18

3.3 Побудова схеми реляційної бази даних у третій нормальній формі….....20

4. Опис програми…………………………………………………………….…30

4.1 Загальні відомості ………………………………………………………….30

4.2 Виклик і завантаження………………………………………………….….30

4.3 Призначення і логічна структура………………………………………….31

4.4 Опис фізичної моделі бази даних……………………………………….…32

4.5 Опис програмної реалізації………………………………………………...35

4.6 Опис задачі автоматизації …………………………………………………41

Висновки………………………………………………………………………...47

Перелік посилань……………………………………………………………….48

ВСТУП

Спорт является частью жизни человека, важность которой очень трудно переоценить. Особенность спорта в том, что человечество не сможет от него отказаться, по крайней мере в обозримом будущем, ведь в современном мире, где всё большую долю рынка труда занимают интеллектуальные профессии, человеку необходимо поддерживать свою физическую форму искусственно, занимаясь физическими нагрузками вне рабочего времени. Многие люди для достижения этой цели выбирают бег, как наиболее естественный и простой способ поддержания тонуса.

Клубы любителей бега объединяют людей, которых интересует участие в организованных соревнованиях и пробегах. Многих участников интересуют как собственные результаты, так и результаты других людей, информация о прошедших соревнованиях и прочее. Для этого администрация клубов ведёт бумажный учёт, вручную сортирует результаты участников соревнований (пробегов), что занимает очень много времени, особенно в том случае, если мероприятие имеет довольно крупный масштаб.

Данная проблема имеет глобальный характер, ведь соревнования проводятся всегда и повсеместно. Для упрощения их организации необходимо создать компьютерную систему, которая бы позволяла хранить информацию о соревнованиях, клубах, тренерах и участниках.

Целью курсовой работы является создание информационной системы «Клуб любителей бега», которая бы позволила решить вышеуказанную проблему. Данная система будет реализована как продукт, предназначенный для организаторов соревнований, участников и администраторов клубов. Программа будет работать на компьютерах с установленным MS SQL Server и .NET Framework 4.5. Продукт будет реализован с помощью технологии Windows Forms та СУБД MS SQL Server.

В результате выполнения курсовой работы, мы получим информационную систему, которая хранит в себе информацию о клубах, соревнованиях, тренерах, участниках и дистанциях. Система будет давать возможность получать, добавлять, редактировать, удалять, сортировать, фильтровать информацию и выполнять поиск. Также информационная система даст возможность спортсменам определить для себя график тренировок.

1 АНАЛИЗ И КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Анализ предметной области

Предметной областью данного проекта является информационная система «Клуб любителей бега». Система разрабатывается для использования администраторами определённого клуба любителей бега и его участниками.

Актуальность проблемы КЛБ состоит в том, что необходимо сократить время и усилия, затрачиваемые на ведение учёта и протоколирование проводимых соревнований. Для этого разумно будет прибегнуть к компьютеризации данной области. В это понятие входит создание базы данных и программного обеспечения для работы с БД, пользователями которой будут спортсмены (самостоятельная регистрация на соревнования) и администраторы (протоколирование соревнований).

В базе данных должна храниться информация о шести объектах: соревнования, спортсмены, дистанции, тренера, клубы, забеги.

Первый элемент предметной области – соревнование. Это мероприятие, в котором участвует большое количество спортсменов. Необходимо хранить информацию о названии соревнования и его основных атрибутах для осуществления поиска в базе данных.

Так как в соревнованиях участвует много людей, необходимо хранить информацию о каждом из них. Для этого нужно создать объект, в котором будет храниться следующая информация об участнике: ФИО, пол, дата рождения, тренер. Таким образом, нужно хранить большое количество участников для каждого мероприятия.

На разных соревнованиях доступны различные дистанции. Каждый спортсмен сам выбирает её для себя.

Спортсмен может заниматься самостоятельно, а может тренироваться у тренера. При этом тренер обязательно является членом клуба.

Наконец, определим характеристики объектов:

а) соревнование (название, дата, уровень, вид);

б) спортсмен (ФИО, дата рождения, пол, тренер);

в) дистанция (тип, километраж, точное расстояние, место проведения).

г) тренер (ФИО, клуб, специализация)

д) клуб (название, дата создания, город)

е) забег (дистанция, участники)

На основе информации из базы данных, система позволит выполнять следующие действия:

а) получать, добавлять, редактировать, удалять информацию о спортсменах;

б) получать, добавлять, удалять информацию о соревнованиях;

в) получать, добавлять, редактировать, удалять информацию о тренерах

г) получать, добавлять, редактировать, удалять информацию о клубах

д) создавать новые соревнования;

е) регистрировать спортсменов на соревнования;

ж) печатать протокол соревнований.

з) печатать отчёт клуба;

и) получать статистику о мероприятии;

к) получать статистику по дистанции;

л) получать статистику наименее/наиболее активного и популярного клуба;

м) получать статистику наименее/наиболее активного и популярного тренера.

* 1. Концептуальное моделирование предметной области

1.2.1 Описание функциональной структуры системы. Для описания функциональной структуры информационной системы используется диаграмма прецедентов (см. рис. 1.1), которая отображает отношения между актёрами и прецедентами.

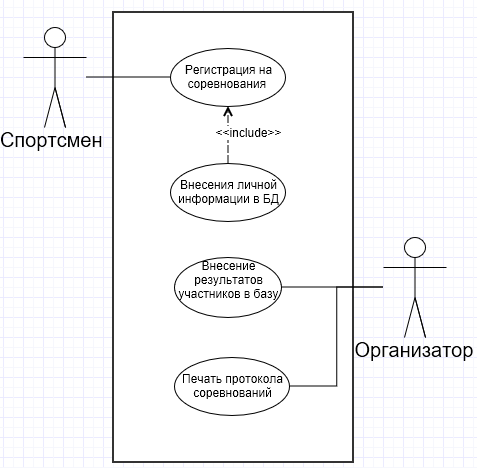


Рисунок 1.1 - Диаграмма прецедентов ИС «Клуб любителей бега»

1.2.2 Описание объектов и связей между ними. В информационной системе «Клуб любителей бега» заложено 6 объектов: соревнования, спортсмен, дистанция, тренер, клуб, забег.

Данные сущности связаны друг с другом следующим образом (см. рис. 1.2):

а) тренер является членом клуба;

б) участник может тренироваться у тренера;

в) забег имеет свою дистанцию и участников;

г) в соревнованиях есть много забегов.

База данных ИС должна содержать следующую информацию:

а) информация про участника: id, ФИО, пол, дата рождения, id тренера (если есть);

б) информация про тренера: id, ФИО, id клуба, специализация;

в) информация про мероприятие: id, тип, уровень, дата проведения;

г) информация про дистанцию: точное расстояние, километраж, тип, место проведения;

д) информация про забег: id забега, id соревнования, дистанция;

е) результат для каждого участника в каждом забеге: id забега, id соревнования, id участника, результат.

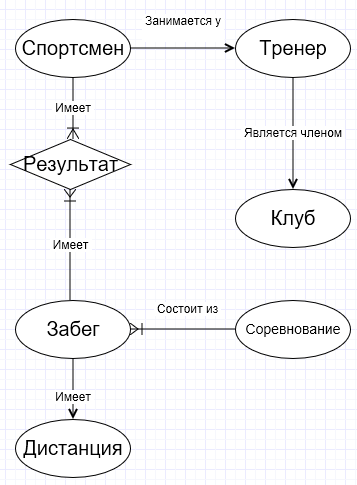


Рисунок 1.2 - Концептуальная модель

1.2.3 Описание информационных потребностей пользователей. ИС «Клуб любителей бега» могут использовать спортсмены и администраторы клуба.

Администраторам необходимо владеть всей информацией об участниках соревнований, об их результатах на текущих соревнованиях, о дистанциях, количестве участников на различных дистанциях, общем количестве участников.

Спортсмены имеют возможность занести себя в базу текущих соревнований (зарегистрироваться), просмотреть информацию о всех участниках текущих соревнований, выбрать дистанцию, просмотреть информацию о своих предыдущих забегах и результатах.

Система должна отображать следующую статистику:

а) статистика количества участников на соревнованиях (общее количество, мужчины, женщины, количество результатов, количество дистанций);

б) статистика количества участников на дистанциях;

в) статистика наиболее/наименее успешного и активного клуба (id, название, количество результатов и участников);

г) статистика наиболее/наименее успешного и активного тренера (id, имя, количество результатов и участников).

1.2.4. Описание документооборота. Документооборот ИС «Клуб любителей бега» содержит в себе протоколы и статистику соревнований.

Система должна печатать следующие отчёты:

а) протокол соревнований (название, дата проведения, результаты всех участников по дистанциям);

б) отчёт клуба (название, город, количество тренеров, участников, результатов, рекорды на дистанциях, принадлежащие клубу).

1.2.5 Ограничения целостности. Ограничения, касающиеся первичных ключей объектов:

а) спортсмены могут иметь одинаковые ФИО, поэтому для однозначной идентификации присваивается уникальный номер;

б) соревнования могут иметь одинаковые названия, поэтому для идентификации присваивается уникальный номер.

в) тренера могут иметь одинаковые ФИО, поэтому для идентификации присваивается уникальный номер;

г) клубы могут иметь одинаковые названия, поэтому для идентификации присваивается уникальный номер;

д) забег имеет составной ключ (id забега, id соревнования), так как их номера повторяются на разных соревнованиях;

Ограничения, касающиеся взаимоотношений между объектами:

а) в соревнованиях участвует много спортсменов;

б) спортсмен может участвовать в нескольких забегах на одном мероприятии;

в) спортсмен может участвовать в неограниченном количестве соревнований;

г) спортсмен может иметь одного тренера или заниматься самостоятельно;

д) тренер обязательно является участником клуба.

1.2.6 Описание алгоритмических зависимостей. ИС «Клуб любителей бега» имеет следующие алгоритмические зависимости:

а) для дистанций в помещении и на стадионе 1 км = 1000м;

б) для дистанций на открытом воздухе 1 км != 1000 м (точную длину дистанции определить очень трудно или невозможно).

1.2.7 Требования к программной системе. Данная программная система использует СУБД MS SQL Server и разработана как программа для рабочего стола Windows. Для работы программы необходимо установить MS SQL Server 2016 и .NET Framework 4.5. Затем восстановить базу данных средствами MS SQL Server Management Studio.

1.2.8 Лингвистические отношения

Пробег – вид соревнований, в котором принимают участие спортсмены всех возрастов. Проводится на открытом воздухе, зачастую на пересечённой местности. Пробеги проводятся в память о каком-то событии или человеке.

1. ПОСТАНОВКА ЗАДАЧИ

Для каждого спортсмена сохраняется такая информация: id, ФИО, пол, дата рождения, id тренера.

Для каждого соревнования сохраняется следующая информация: id, название, уровень, тип.

Для каждой дистанции сохраняется следующая информация: тип, километраж (для пробегов), точное расстояние (для соревнований на стадионах и в зале).

Для каждого клуба сохраняется следующая информация: id, название, дата основания, город.

В программном продукте должны быть реализованы следующие функции:

а) отображение данных:

1) отображение списка соревнований;

2) отображение списка спортсменов;

3) отображение списка дистанций;

4) отображение списка тренеров;

5) отображение списка клубов;

б) поиск и фильтрация:

1) поиск соревнований по id, названию, дате, типу, уровню;

2) поиск спортсменов по id, ФИО, полу, дате рождения, тренеру;

3) поиск тренеров по id, ФИО, клубу, специализации;

4) поиск клубов по id, названию, дате основания, городу;

5) фильтрация информации о спортсменах по полу;

6) фильтрация информации о соревнованиях по типу и уровню;

7) фильтрация информации о тренерах по специализации;

в) добавление, удаление, редактирование данных:

1) добавление, удаление, редактирование информации о спортсмене;

2) добавление, удаление, редактирование информации о тренере;

3) добавление, удаление, редактирование информации о клубе;

4) добавление, удаление, редактирование информации о соревнованиях;

5) добавление забегов в программу соревнований;

6) регистрация существующих в базе спортсменов на соревнования

г) запросы к БД:

1) запросы вывода информации о соревнованиях (id, название, дата, тип, уровень), участниках (id, ФИО, пол, дата рождения, имя тренера), тренерах (id, ФИО, клуб, специализация), клубах (id, название, дата основания, город), дистанциях (длина, километраж, тип, место проведения);

2) запросы фильтрации информации о соревнованиях (по типу, уровню), спортсменах (по полу, дате рождения), тренерах (по специализации);

д) формирование статистики и отчётов:

а) отчёт «Протокол соревнований» (название, дата проведения, результаты всех участников по дистанциям);

б) отчёт клуба (название, город, количество тренеров, участников, результатов, рекорды на дистанциях, принадлежащие клубу).

На основе связей между сущностями в базе данных система должна вычислять следующие значения:

а) общее количество участников на мероприятии;

б) количество женщин и мужчин на мероприятии;

в) количество результатов на мероприятии;

г) количество дистанция на мероприятии;

д) количество результатов клуба;

е) количество участников клуба;

ж) количество результатов учеников тренера;

з) количество учеников тренера;

е) задача автоматизации

Система должна поддерживать выполнение регулярных запросов (статистики):

а) статистика мероприятия (количество участников, количество мужчин и женщин, количество результатов, количество дистанций);

б) статистика дистанций (количество участников на каждой дистанции);

в) статистика клубов (id, название, количество участников и результатов наиболее и наименее популярных и наиболее и наименее активных клубов);

г) статистика тренеров(id, название, количество учеников и результатов наиболее и наименее популярных и наиболее и наименее активных тренеров).

Задача автоматизации: благодаря задаче автоматизации, которую реализует система, спортсмен имеет возможность быстро и удобно определить для себя график тренировок исходя из информации о популярных дистанциях за последнее время и специализации личного тренера.

1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ
   1. UML-моделирование

Важной функцией программы является возможность регистрации участников из базы данных на соревнования. На рисунке 3.1 изображения диаграмма деятельности, отображающая процесс регистрации.

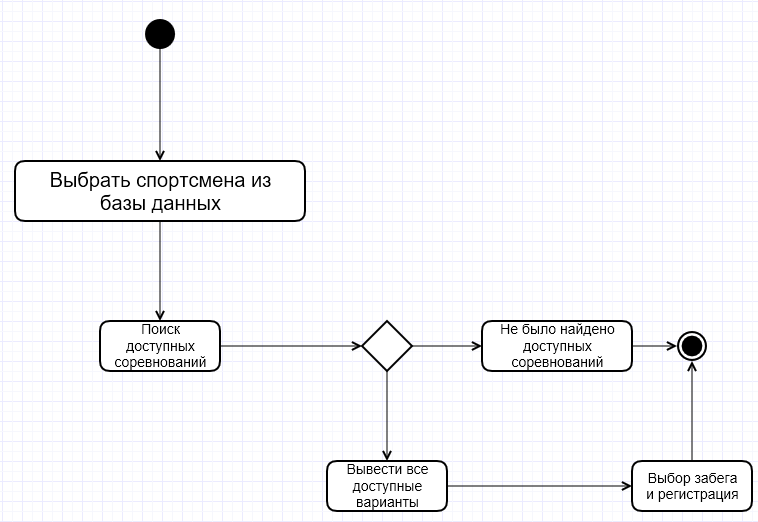


Рисунок 3.1 – Диаграмма деятельности

3.2 Построение ER-диаграммы

Проанализировав концептуальную модель предметной области, можно сделать вывод, что в базу данных ИС «Клуб любителей бега» будут входить следующие объекты:

а) сущность «Участник», которая содержит следующие атрибуты: номер участника, ФИО участника, пол, дата рождения, тренер;

б) сущность «Клуб», которая содержит следующие атрибуты: номер клуба, название клуба, дата основания, город;

в) сущность «Тренер», которая содержит следующие атрибуты: номер тренера, ФИО тренера, клуб, специализация;

г) сущность «Мероприятие», которая содержит следующие атрибуты: номер соревнования, название, дата проведения, тип, уровень;

д) сущность «Забег», которая содержит следующие атрибуты: номер забега, номер мероприятия, дистанция;

е) сущность «Участник забега», которая содержит следующие атрибуты: номер мероприятия, номер забега, номер участника, результат.

ж) сущность «Дистанция», которая содержит следующие атрибуты: точное расстояние, километраж, тип, место проведения;

Между вышеуказанными сущностями существуют связи:

а) между сущностями «Клуб» и «Тренер» связь один-ко-многим: один тренер может быть участником одного клуба, в одном клубе может быть много тренеров;

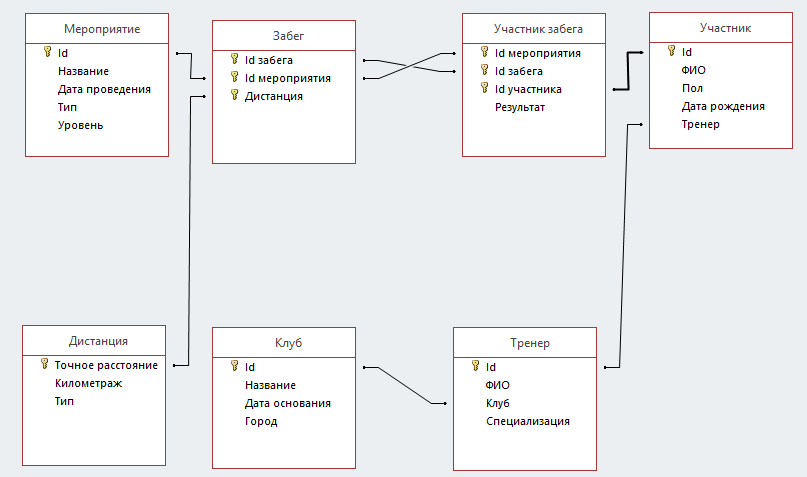
б) между сущностями «Участник» и «Тренер» связь один-ко-многим: один участник может быть учеником одного тренера, один тренер может иметь много учеников;

в) между сущностями «Мероприятие» и «Забег» связь один-ко-многим: забег принадлежит одному мероприятию, в одном мероприятии может быть несколько забегов;

г) между сущностями «Участник» и «Забег» связь много-ко-многим: участник может принимать участие в нескольких забегах, в одном забеге может быть несколько участников. Для этого было введена промежуточная сущность «Участники забега», с которой две вышеупомянутые сущности в связи один-ко-многим.

д) между сущностями «Забег» и «Дистанция» отношение один-ко-многим: забег может иметь одну дистанцию, по одной дистанции может проводиться много забегов.

На основе всего вышесказанного можно построить ER-диаграмму информационной системы «Клуб любителей бега» (см. рис. 3.2)

Рисунок 3.2 - ER-диаграмма ИС «Клуб любителей бега»

На данной диаграмме присутствует 7 сущностей, ключевыми атрибутами для которых являются:

а) соревнование – id;

б) спортсмен – id;

в) дистанция – точное расстояние;

г) клуб – id;

д) тренер – id;

е) забег – id мероприятия + id забега;

ж) участник забега – id мероприятия + id забега + id участника.

3.3 Построение схемы реляционной базы данных в третьей нормальной

форме

Сначала построим схему базы данных в первой нормальной форме, определим первичные ключи и покажем зависимости между атрибутами (см. рис. 3.3)

Т1 (1НФ)

\*Номер клуба

Название клуба

Дата основания клуба

Город клуба

\*Номер тренера

ФИО тренера

Клуб тренера

Специализация тренера

\*Номер участника

ФИО участника

Пол участника

Дата рождения участника

Тренер участника

\*Номер мероприятия

Название мероприятия

Дата проведения мероприятия

Тип мероприятия

Уровень мероприятия

\*Точная дистанция

Километраж

Тип дистанции

\*Номер забега

\*Номер мероприятия

Дистанция забегам

Результат

Рисунок 3.3 – Т1 у 1НФ (продовження)

Для перехода к 3НФ необходимо, чтобы каждый неключевой атрибут был в полной функциональной зависимости от первичного ключа и отсутствовали транзитивные зависимости. Для этого нам необходимо отделить он нашей таблицы в отдельные отношения части первичных ключей и зависимые от этих частей атрибуты. (см. рис. 3.4 – 3.10)

Т1.1 (3НФ)

\*Номер клуба

Название клуба

Дата основания клуба

Город клуба

Рисунок 3.4 – Т1.1 в 3НФ

Т1.2 (3НФ)

\*Номер тренера

ФИО тренера

Клуб тренера

Специализация тренера

Рисунок 3.5 – Т1.2 в 3НФ

Т1.3 (3НФ)

\*Номер участника

ФИО участника

Пол участника

Дата рождения участника

Тренер участника

Рисунок 3.6 – Т1.3 в 3НФ

Т1.4 (3НФ)

\*Номер мероприятия

Название мероприятия

Дата проведения мероприятия

Тип мероприятия

Уровень мероприятия

Рисунок 3.7 – Т1.4 в 3НФ

Т1.5 (3НФ)

\*Точная дистанция

Километраж

Тип дистанции

Рисунок 3.8 – Т1.5 в 3НФ

Т1.6 (3НФ)

\*Номер забега

\*Номер мероприятия

Дистанция забега

Рисунок 3.9 – Т1.6 в 3НФ

Т1.7 (3НФ)

\*Номер забега

\*Номер мероприятия

\*Номер участника

Результат

Рисунок 3.10 – Т1.7 в 3НФ

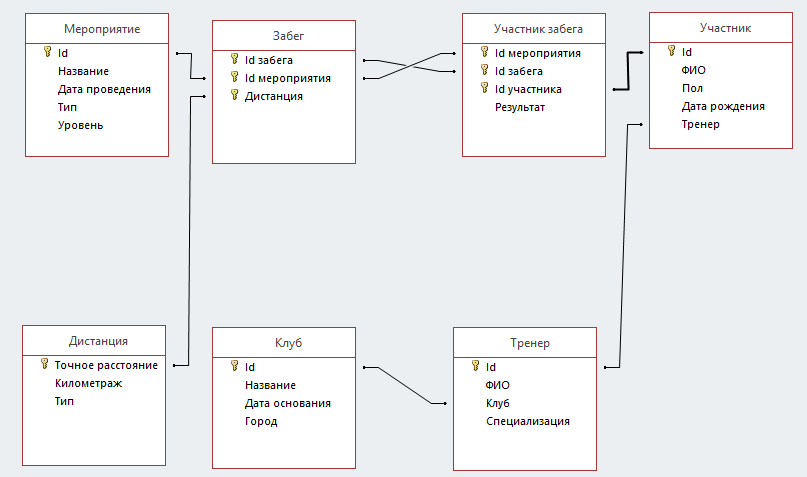
 В результате нормализации была получена следующая схема базы данных (см. рис. 3.11)

Рисунок 3.11 – Схема базы данных после нормализации

1. ОПИСАНИЕ ПРОГРАММЫ
   1. Общие сведения

Информационная система «Клуб любителей бега» - это программа, которая создавалась при помощи среды разработки Visual Studio 2015 на платформе Windows Forms . Разработка велась с помощью языка программирования C#.

В качестве СУБД был выбрал MS SQL Server Express 2016. Доступ к данным осуществляется с помощью технологии ADO.NET.

Программа будет работать на всех компьютерах, на которых установлен .NET Framework 4.5 и MS SQL Server 2016.

* 1. Вызов и загрузка

После загрузки программы на ПК, необходимо восстановить базу данных задачей «Восстановить БД» в MS SQL Server Management Studio. После этого необходимо запустить программу, используемый ПК станет сервером для программы.

* 1. Предназначение и логическая структура

Программная система выполняет следующие функции:

а) добавление, хранение, редактирование, удаление, просмотр информации про каждую сущность в базе данных;

б) поиск, сортировка, фильтрация информации о каждой сущности в базе данных;

в) формирование статистики;

г) печать отчётов;

д) выполнение задачи автоматизации;

Поиск в программе по умолчанию осуществляется по всем атрибутам объекта, однако есть возможность выбрать необходимое поле и осуществлять поиск по нему. Поиск отличается от фильтрации тем, что неподходящие записи не отбрасываются, а подходящие подсвечиваются другим цветом.

Сортировку можно выполнить по определённому атрибуту сущности. Также пользователь может выбрать сортировку по убыванию или возрастанию.

Фильтрация осуществляется с учётом определённых полей сущности. При этом после фильтрации остаются только те записи, которые удовлетворяют параметры фильтрации.

Статистика в программе представлена в виде форм, в которых подана информация, получаемая из разных таблиц базы данных. Некоторые статистики требуют ввода данных пользователем.

При печати отчётов пользователь имеет возможность выбрать необходимый экземпляр сущности, по которому он будет сформирован, а зачем получить файл в формате .pdf.

Пользователь, введя свой Id может получить график тренировок, исходя из информации о популярных дистанциях и специализации тренера.

* 1. Описание физической модели базы данных

В качестве СУБД для программы используется MS SQL Server Express 2016. Этот продукт является одним из лидеров среди самых популярных баз данных. Он используется более c 1989 года для реализации самых разнообразных программ.

К преимуществам MS SQL Server относится его продуктивность. Продукт имеет прозрачное сжатие и шифрование данных. В MS SQL Server есть эффективный контроль доступа и средств управления разрешениями и он обеспечивает высокую эффективность при сборе данных. Также программа имеет высокий уровень защиты данных, поэтому её используют для реализации программ, в которых присутствует аутентификация и авторизация пользователей.

Для доступа к данным используется технология ADO.NET. Особенностью данной технологии является то, что она не взаимодействует напрямую с СУБД, а использует поставщики данных, которые инкапсулируют механизм работы с конкретной СУБД. Данный подход позволяет гибко изменять и создавать адаптеры для любой СУБД.

Ниже приведены структуры таблиц в базе данных: Участник, Тренер, Клуб, Мероприятие, Дистанция, Забег, Участник забега (см. рис. 4.1 – 4.7)

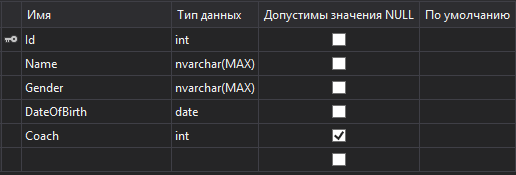


Рисунок 4.1 – Структура таблицы «Участник»

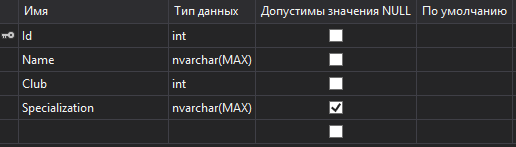


Рисунок 4.2 – Структура таблицы «Тренер»

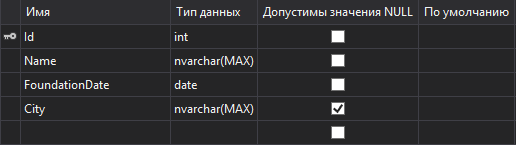


Рисунок 4.3 – Структура таблицы «Клуб»

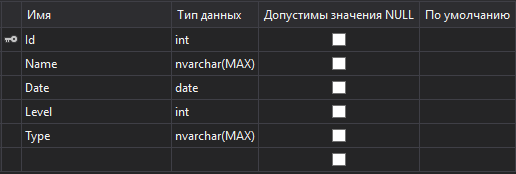


Рисунок 4.4 – Структура таблицы «Мероприятие»

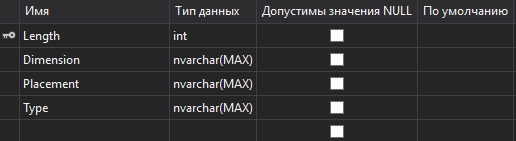


Рисунок 4.5 – Структура таблицы «Дистанция»

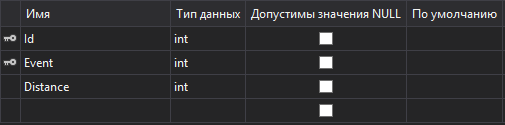


Рисунок 4.6 – Структура таблицы «Забег»

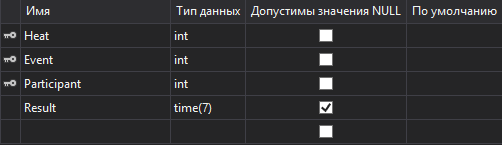


Рисунок 4.7 – Структура таблицы «Участник забега»

* 1. Описание программной реализации

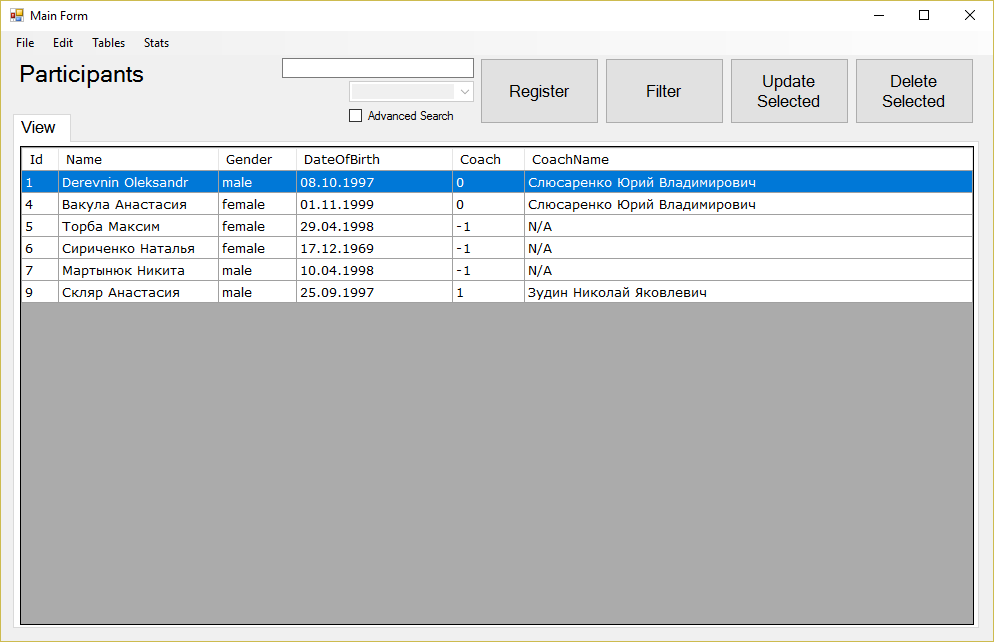
При запуске программы пользователь увидит окно со списком существующих в БД ИС спортсменов (см. рис. 4.8). На данной форме пользователь сможет переключаться между таблицами базы данных, добавить, удалить или отредактировать желаемую сущность с помощью кнопок формы или пунктов меню.

Рисунок 4.8 – Главная форма

При нажатии на пункт меню «Edit – Add» появится форма, которая соответствует текущей таблице (см. рис. 4.9). Пользователь сможет добавить новый экземпляр в базу данных.

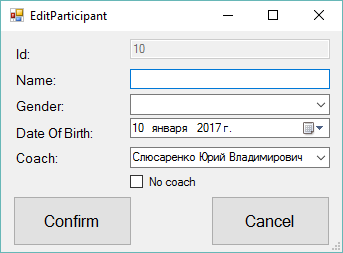


Рисунок 4.9 – Форма добавления участника

При нажатии на кнопку «Update Selected» или пункт меню «Edit – Update» появится форма редактирования, соответсвующая выбранной таблице и идентичная форме добавления, но с заполненными полями (см. рис. 4.10).

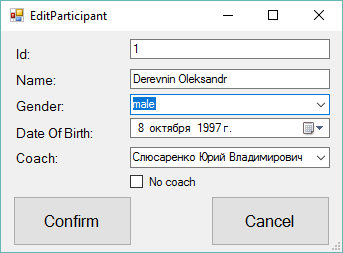


Рисунок 4.10 – Форма редактирования участника

Для фильтрации необходимо нажать кнопку «Filter», после чего появится форма, в которой необходимо выбрать поля для фильтрации (см. рис. 4.11)

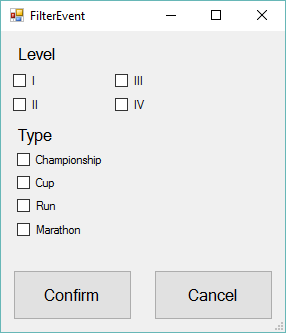


Рисунок 4.11 – Форма фильтрации мероприятий

Для просмотра статистик необходимо выбрать пункт меню «Stats – Get stats», после чего появится форма отображения статистики, в которой необходимо выбрать вкладку с необходимой статистикой или выбрать экземпляр, по которому пользователь хочет получить статистику (см. рис. 4.12)

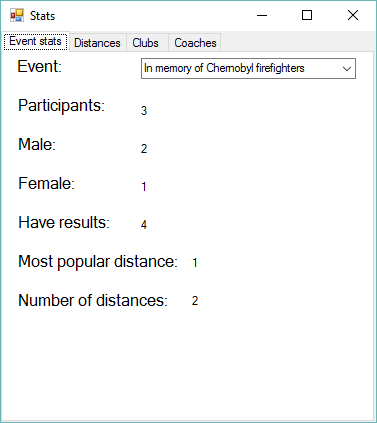


Рисунок 4.12 – Форма отображения статистики мероприятия

Для просмотра отчётов необходимо выбрать пункт меню «Stats – Get Protocol» или «Stats – Get Report», после чего откроется форма отображения протокола соревнований с возможностью печать в файл .pdf (см. рис. 4.13) или форма отчёта клуба с возможностью печати в файл .pdf (см. рис. 4.14) соответственно.

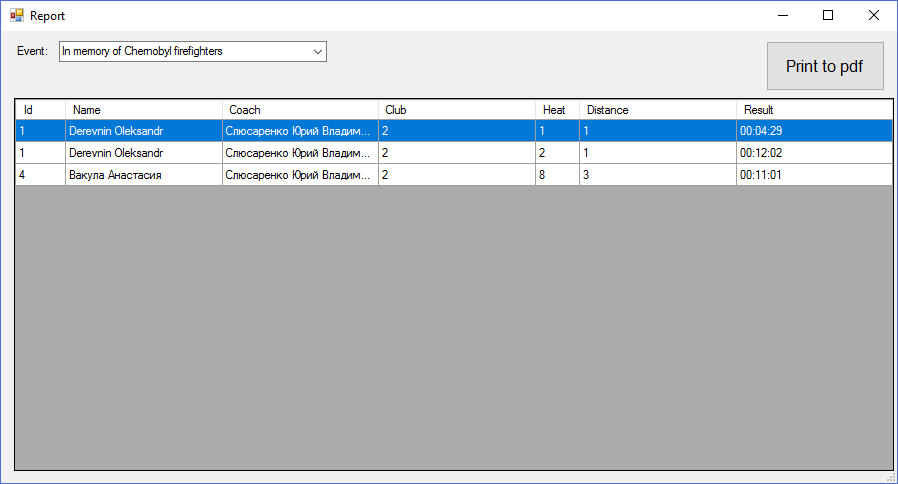


Рисунок 4.13 – Протокол соревнований

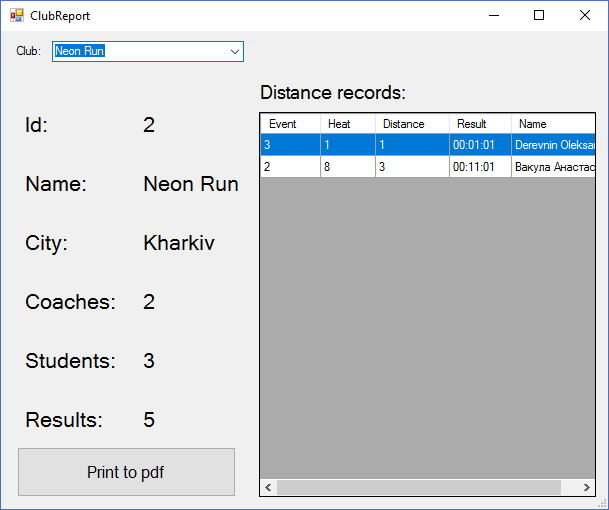


Рисунок 4.14 – Отчёт клуба

Система выполняет такие виды запросов для статистики:

а) статистика про количество участников на соревнованиях:

SELECT COUNT(DISTINCT Participant) FROM HeatParticipants WHERE Event = @Event;

SELECT COUNT(DISTINCT HeatParticipants.Participant) AS Participants

FROM HeatParticipants INNER JOIN

Participant ON HeatParticipants.Participant = Participant.Id

WHERE (Participant.Gender = @Gender) AND Event = @Event;

б) статистика про количество результатов на соревнованиях:

SELECT COUNT(Result) AS Results

FROM HeatParticipants

WHERE Event = @Event;

в) статистика про активность клубов:

SELECT Club.Id, Club.Name, COUNT(HeatParticipants.Result) AS Results, COUNT(Coach.Id) AS Coaches

FROM Club INNER JOIN

Coach ON Club.Id = Coach.Club INNER JOIN

Participant ON Coach.Id = Participant.Coach INNER JOIN

HeatParticipants ON Participant.Id = HeatParticipants.Participant

GROUP BY Club.Id, Club.Name

ORDER BY Results DESC;

г) статистика про популярность клубов:

SELECT Club.Id, Club.Name, COUNT(HeatParticipants.Result) AS Results, COUNT(Participant.Id) AS Participants

FROM Club INNER JOIN

Coach ON Club.Id = Coach.Club INNER JOIN

Participant ON Coach.Id = Participant.Coach INNER JOIN

HeatParticipants ON Participant.Id = HeatParticipants.Participant

GROUP BY Club.Id, Club.Name

ORDER BY Participants DESC;

д) статистика про активность тренеров:

SELECT Coach.Id, Coach.Name, COUNT(HeatParticipants.Result) AS Results, COUNT(DISTINCT Participant.Id) AS Participants

FROM Coach INNER JOIN

Participant ON Coach.Id = Participant.Coach INNER JOIN

HeatParticipants ON HeatParticipants.Participant = Participant.Id

GROUP BY Coach.Id, Coach.Name

ORDER BY Results DESC;

е) статистика про популярность тренеров:

SELECT Coach.Id, Coach.Name, COUNT(HeatParticipants.Result) AS Results, COUNT(DISTINCT Participant.Id) AS Participants

FROM Coach INNER JOIN

Participant ON Coach.Id = Participant.Coach INNER JOIN

HeatParticipants ON HeatParticipants.Participant = Participant.Id

GROUP BY Coach.Id, Coach.Name

ORDER BY Participants DESC;

4.6 Описание задачи автоматизации

Участник соревнований может определить для себя недельный график тренировок, нажав на кнопку «Get training plan» и введя свой уникальный идентификатор в поле. По нажатию кнопки «Confirm» будет определён график тренировок, исходя из информации о популярных дистанциях, специализации тренера и возрасте участника.

Алгоритм задачи автоматизации следующий:

а) система находит самую популярную за последний месяц дистанцию;

б) система сравнивает специализацию тренера с типом дистанции;

в) в случае несовпадения, пользователь получает информацию о том, что его график может сильно измениться и ему предлагается сменить тренера для более продуктивной работы;

г) система выстраивает график тренировок, учитывая допустимые для участника данного возраста нагрузки;

д) система выдаёт результат участнику.