**ВВЕДЕНИЕ**

Целью курсового проектирования является развитие навыков работы по проектированию информационных систем и закрепление знаний, полученных за время изучения дисциплины «Организация баз данных и баз знаний».

Курсовой проект предусматривает создание многотабличной базы данных и набора функций и процедур, обеспечивающих интерфейс пользователя.

Выполнение курсового проекта предусматривает:

- составление схемы концептуальной модели данных,

- разработку структуры реляционной базы данных,

- разработку интерфейса пользователя,

- программирование задачи,

- составление контрольного примера,

- оформление пояснительной записки.

Задание на курсовое проектирование выдается в начале 8 семестра. Вариант задания выбирается по номеру студента и согласовывается с руководителем. Завершение курсового проектирование и его защита предусмотрена на 11-12 неделях.

1. **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Проведем полный анализ предметной области «Спортивная программа». Эта область может быть описана следующими основными объектами и действиями, представляющими интерес с точки зрения различных групп пользователей.

Выделим эти объекты:

- Спортсмен

- Команда

- Тренер

- Награда

- Вид спорта

- Соревнование

- Телепередача

- Сюжеты

- Гость программы

Эти объекты имеют следующие информационные характеристики (атрибуты):

* + Объект «СПОРТСМЕН» (Ид\_Спортсмен, ФИО, Страна, Адрес, Пол, СпортРазряд);
  + Объект «КОМАНДА» (Ид\_Команда, Название, Страна);
  + Объект «ТРЕНЕР» (Ид\_Тренер, ФИО, Адрес, Категория);
  + Объект «НАГРАДА» (Ид\_Награда, ВидНаграды, ДатаВручения, Примечание);
  + Объект «ВИД\_СПОРТА» (Ид\_Спорта, Название, Описание);
  + Объект «СОРЕВНОВАНИЕ» (Ид\_Соревнования, Название, Описание, МасштабСоревнования, ДатаПроведения, МестоПроведения, ВремяПроведения, Победитель);
  + Объект «ТЕЛЕПЕРЕДАЧА» (Ид\_передачи, ДатаЭфира, Ведущий);
  + Объект «СЮЖЕТЫ» (Ид\_сюжета, Тематика, Автор, ДатаСоздания, Длительность);
  + Объект «ГОСТЬ\_ПРОГРАММЫ» (Ид\_гостя, ФИО, Адрес, РодЗанятий, КраткаяБиография)

ЭТАПЫ ПРОЕКТИРОВАНИЯ БД

* 1. **Первичные ключи**

Первичный ключ – это атрибут или группа атрибутов, которые однозначно идентифицируют екземпляр объекта.

Обозначим первичные ключи для перечисленнях ранее объектов:

- Спортсмен - Ид\_Спортсмен\*;

- Тренер - Ид\_Тренер\*;

- Команда - Ид\_Команда\*;

- Награда - Ид\_Награда\*;

- Вид спорта - Ид\_Спорта\*;

- Соревнование - Ид\_Соревнования\*;

- Телепередача – Ид\_Передачи\*;

- Сюжеты – Ид\_сюжета\*;

- Гость программы – Ид\_гостя\*.

* 1. **Анализ аномалий**

Избыточность данных ведет не только к потере места в памяти, но и может нарушить целостность данных. Существует три типа аномалий:

1. Аномалия обновления – это противоречивость данных, вызванная их избыточностью и частичным обновлением.

2. Аномалия удаления – это непреднамеренная потеря данных, вызванная удалением временных данных.

3. Аномалия ввода – это невозможность ввести одни данные из-за отсутствия других.

Чтобы избежать этих аномалий, необходимо выполнить нормализацию отношений.

* 1. **Нормализация отношений**

Впроцессе нормализации атрибуты группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, обновлении, и удалении данных, чем все другие наборы таблиц, с помощью которых могут быть представлены те же самые данные.

Чтобы избежать аномалий, разобьем таблицу «СОРЕВНОВАНИЕ» на три: «СОРЕВНОВАНИЕ», «КомандаСоревнование», «ВидСоревнования».

Руководствуясь таким же принципом, выделим таблицу «НаградаСпортсмена», «СюжетПередача» и «ГостьПередача».

Кроме того, добавим еще две таблицы: «User» – для хранения информации о пользователях и их паролях; и «Log» – хранит информацию о действиях, выполненных пользователями в системе.

* 1. **Описание концептуальной модели данных**

Концептуальная модель данных (КМД)– это модель используемой на предприятии информации, которая не зависит от любых физических аспектов представления этой информации (тип СУБД, используемые ЯП, тип вычислительной платформы).

На этапе создания КМД определяются типы сущностей, типы связейЮ атрибутов, домены; создается модель «Сущность-связь» (Entity-Relationship), которая в основе сожержит следующие базове понятия:

Сущность – с ее помощью моделируется класс однотипних объектов. Имеет имя, уникальное в пределах модели. Предполагается, что существует много экземпляров данной сущности.

Атрибут – характеристика, определяющая свойства сущности.

Связь – определяет, как сущности взаимодействуют друг с другом. Может быть установлена между сущностью и другими сущностями, или между сущностью и ей же самой (рекурсивная связь).

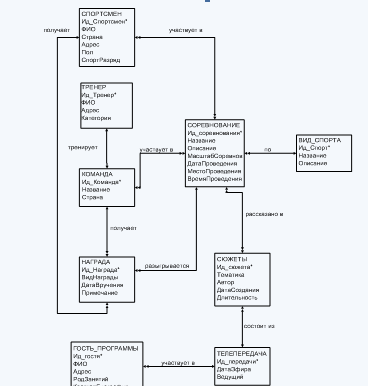
Виды связей:

* + 1. «1:1» (один к одному) – определяет такой вид связи между сущностями А и В, корда каждому экземпляру сущности А соответствует один и только один экземпляр сущности В, и, на оборот.
    2. «1:М» (один ко многим) – экземпляру сущности А соответствует 0, 1 или несколько экземпляров сущности В, и, наоборот.
    3. «М:М» (многие ко многим) – одному экземпляру сущности А соответствует 0, 1 или несколько экземпляров сущности В, и, наоборот.

Между двумя сущностями может быть установлено любое количество связей с разными смысловыми нагрузками.

Любая связь может быть обязательной, если в данной связи должен участвовать каждый экземпляр сущности, или необязательной. Связь может быть обязательной с одной стороны и необязательной с другой.

КМД для предметной области «Спортивная программа» показана на рис.1



*Рис.1 – КМД для предметной области «Спортивная программа»*

Двойная стрелка означает «многие», одинарная стрелка означает «один» во взаимосвязи между объектами.

Ключевые атрибуты обозначены \*.

* 1. **Описание реляционной модели данных**

Реляционная модель данных (РМД) представляет БД в виде множества взаимосвязанных отношений, в том числе и иерархических.

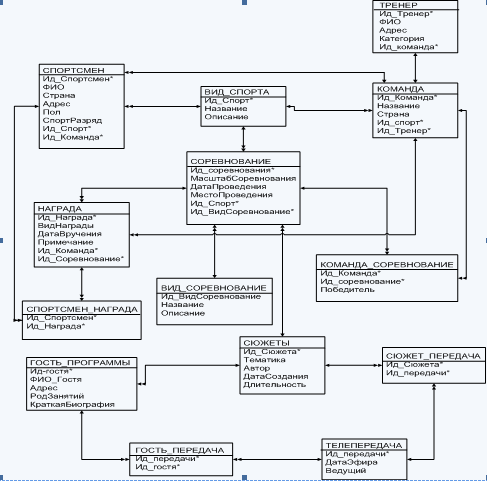
Одно отношение в каждой связи выступает как родительское, а другое - как подчиненное.

Для поддержания связи в отношении должны быть включены специальные атрибуты. В родительском отношении это первичный ключ, а в подчиненном – набор атрибутов, соответствующий внешнему ключу.

Правила перехода от КМД к РМД:

1. Каждый объект с атрибутами преобразуется в таблицу и ключевой атрибут становится ключом таблицы.
2. Для отображения связи «1:1» в одну из таблиц включают первичный ключ второй таблицы.
3. Связь «1:М» отображается так: в таблицу, находящуюся сос тороны многих, добавляют первичный ключ таблицы, находящейся со стороны одного.
4. Для отображения связи «М:М» создается дополнительная таблица, в которую включают первичный ключ и первой и второй таблицы, а также дополнительные атрибуты.

РМД для предметной области «Спортивная программа» строится на основе модели, изображенной на рис.2, где учтены правила пре хода и нормализация отношений.



*Рис. 2 – Модель, на основе которой строится РМД для предметной области «Спортивная программа»*

РМД представляет информацию в виде таблиц.

Строка таблицы называется картежом.

Не всякая таблица является отношением. Для этого должны выполняться условия:

- все записи должны иметь одинаковую структуру;

- названия столбцов должны быть уникальными;

- значения в одном столбце должны принадлежать одному домену.

Физически в памяти компьютера хранится схема БД, изображенная в приложении 1.

Типы данных для полей всех таблиц:

1. **СПОСОБЫ РЕАЛИЗАЦИИ ЗАПРОСОВ И ОТЧЕТОВ**

Посредством языка SQL необходимо реализовать 10 групп запросов:

1. Выбор из нескольких таблиц с сортировкой.

2. Задание условия отбора с использованием предиката LIKE.

3. Задание условия отбора с использованием предиката BETEWEEN.

4. Агрегатная функция без группировки.

5. Агрегатная функция с группировкой.

6. Использование предиката ALL или ANY.

7. Коррелированный подзапрос.

8. Запрос на отрицание. Запрос реализовать в трех вариантах: с использованием LEFT JOIN, предиката IN и предиката EXISTS.

9. Операция объединения UNION с включением комментария в каждую строку.

10. Обновление данных, условие отбора формируется с использованием подзапроса из другой таблицы.

Реализация запросов группы 1)

***Запрос 1.*** Спортсмены являющиеся гражданами заданной страны

*SELECT ID\_Sportsmen, FIO*

*FROM Sportsmen*

*WHERE ((Sportsmen.Strana)=[Введите страну]);*

***Запрос 2.*** Сколько серебряных медалей получили команды

*SELECT count(\*) AS [количество серебра]*

*FROM Komanda*

*WHERE ID\_Komanda in (Select ID\_Komanda from Nagrada where VidNagradi='Медаль серебряная');*

Реализация запросов группы 2)

***Запрос 3.*** Вывести список спортсменов, имя которых начинается с буквы «К»

*SELECT FIO*

*FROM Sportsmen*

*WHERE FIO LIKE 'K\*';*

***Запрос 4.*** Вывести список команд из Украины

*SELECT Komanda.Nazvanie, Komanda.Strana, Komanda.Gorod, Komanda.ID\_Sport*

*FROM Komanda*

*WHERE (((Komanda.Strana) Like 'У\*'));*

Реализация запросов группы 3)

***Запрос 5.*** Вывести список волейбольных команд

*SELECT Komanda.Nazvanie, Komanda.Strana, Komanda.Gorod*

*FROM Komanda*

*WHERE (((Komanda.ID\_Sport) Between 3 And 3));*

***Запрос 6.*** Вывести список сюжетов со 2го по 5й

*SELECT ID\_Sujet, Tema, Avtor*

*FROM Sujet*

*WHERE (((ID\_sujet) Between 2 And 5));*

Реализация запросов группы 4)

***Запрос 7.*** Какие соревнования не проводились в заданный день

*SELECT Nazvanie*

*FROM VidSporta*

*WHERE ID\_Sport in*

*(Select ID\_Sport*

*from Sorevnovania*

*where month([DataProveden])<>month(Date));*

***Запрос 8.*** Спортсмены из команды по заданному виду спорта

*SELECT Komanda.Nazvanie AS Komanda, Sportsmen.FIO AS Sportsmen*

*FROM (VidSporta INNER JOIN Komanda ON VidSporta.ID\_Sport=Komanda.ID\_Sport) INNER JOIN Sportsmen ON Komanda.ID\_Komanda=Sportsmen.ID\_Komanda*

*WHERE (((VidSporta.Nazvanie)=[Введите вид спорта]));*

Реализация запросов группы 5)

***Запрос 9.*** Вывести количество игроков, занимающихся каждым видом спорта

*SELECT VidSporta.ID\_Sport, VidSporta.Nazvanie, Count(Sportsmen.ID\_Sportsmen) AS [Kol\_vo igrokov]*

*FROM VidSporta INNER JOIN Sportsmen ON VidSporta.ID\_Sport=Sportsmen.ID\_Sport*

*GROUP BY VidSporta.ID\_Sport, VidSporta.Nazvanie;*

***Запрос 10.*** Сколько спортсменов тренирует каждый тренер

*SELECT Trener.ID\_Trener, Trener.FIO, Count(Sportsmen.ID\_Sportsmen) AS [Kol\_vo sportsmenov]*

*FROM Trener INNER JOIN Sportsmen ON Trener.ID\_Trener=Sportsmen.ID\_Trener*

*GROUP BY Trener.ID\_Trener, Trener.FIO;*

Реализация запросов группы 6)

***Запрос 11.*** Вывести таблицу «Соревнования», сортируя по полю «Дата проведения » по возрастанию

*SELECT ALL \**

*FROM Sorevnovania*

*ORDER BY DataProveden;*

***Запрос 12.*** Вывести таблицу «Спортсмен-награда» с сортировкой по убыванию по полю «Ид\_спортсмен»

*SELECT ALL \**

*FROM SportsNagrada*

*ORDER BY ID\_Sportsmen DESC;*

Реализация запросов группы 7)

***Запрос 13.*** Количество золотых медалей, выигранных командой

*SELECT count(\*) AS [количество золотых медалей]*

*FROM Komanda*

*WHERE ID\_Komanda in (Select ID\_Komanda from Nagrada where VidNagradi='Медаль золотая');*

***Запрос 14.*** Спортсмены, которые получили введенную награду

*SELECT FIO, Pol, SportRazr*

*FROM Sportsmen*

*WHERE ID\_Komanda in*

*(Select ID\_Komanda*

*from Komanda*

*where ID\_Komanda in*

*(Select ID\_Komanda*

*from Nagrada*

*where (((Nagrada.VidNagradi)=[Введите вид награды])) ))*

*or ID\_Sportsmen in*

*(Select ID\_Sportsmen*

*from SportsNagrada*

*where ID\_Nagrada in*

*(Select ID\_Nagrada*

*from Nagrada*

*where (((Nagrada.VidNagradi)=[Введите вид награды])) ));*

Реализация запросов группы 8)

***Запрос 15.*** Количество кубков, выигранных командой

*SELECT Count(\*) AS [количество кубков]*

*FROM Komanda*

*WHERE (((Komanda.ID\_Komanda) In (Select ID\_Komanda from Nagrada where VidNagradi='Кубок')));*

***Запрос 16.*** Спортсмены, которые выступают не за свою страну

*SELECT Sportsmen.ID\_Sportsmen, Sportsmen.FIO, Sportsmen.Strana*

*FROM Komanda INNER JOIN Sportsmen ON Komanda.ID\_Komanda=Sportsmen.ID\_Komanda*

*WHERE (((Sportsmen.Strana)<>Komanda.Strana));*

Реализация запросов группы 10)

***Запрос 17.*** Дата проведения последнего соревнования заданного масштаба

*SELECT MasshtabSorevn, DataProveden, MestoProveden*

*FROM Sorevnovania*

*WHERE ((MasshtabSorevn)=[Введите масштаб соревнования])*

*ORDER BY 2;*

*SELECT Last(Zapros11\_1.DataProveden) AS [Последнее соревнование было]*

*FROM Zapros11\_1;*

***Запрос 18.*** У какой команды больше всего наград

*SELECT Nagrada.ID\_Komanda, Count(Nagrada.VidNagradi) AS [Count-VidNagradi]*

*FROM Nagrada*

*GROUP BY Nagrada.ID\_Komanda*

*ORDER BY 2;*

*SELECT Last(Zapros12\_1.ID\_Komanda) AS [Больше всего наград у команды №]*

*FROM Zapros12\_1;*

На основе запросов из пунктов 5, 7 и 9 необходимо реализовать отчеты.

***Отчет 1.*** Награды команд

В готовом виде:

***Отчет 2.***Составы команд

В готовом виде:

***Отчет 3.***Каких спортсменов тренирует каждый тренер

**3 ГРУППЫ ПОЛЬЗОВАТЕЛЕЙ**

С разработанной базой данных может работать 3 группы пользователей:

1. Администратор – разрешены все действия, в том числе ввод и редактирование таблиц.

2. Менеджер – может работать со всеми данными, но не может добавлять записи в таблицы.

3. Пользователь – может только просматривать запроси и отчеты

Подробнее о группах пользователей и их правах можно узнать из диаграммы прецедентов в приложении 1.

1. **ИНТЕРФЕЙС БАЗЫ ДАННЫХ**

Интерфейс реализован в среде Microsoft Access. Программа имеет многооконный интерфейс. Для каждого окна реализована форма.

Первоначально загружается форма, которая запрашивает имя и пароль пользователя:

В БД есть таблица, в которой хранятся группы пользователей и пароли для входа в систему:

После входа в систему загружается главная форма. Она связана с конкретными таблицами. Права пользователей разделены за счет видимости некоторых кнопок, выполняющих те или иные действия.

Болем подробно интерфейс программы для каждого пользователя рассмотрен в следующем разделе.

1. **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Чтобы войти в систему в режиме администратора необходимо ввести имя «admin» и пароль «as», в режиме менеджера – «user1», «men», в режиме пользователя – «user2», «us».

Далее при нажатии на кнопку проверяется, правильные ли имя и пароль введены. Если имя и пароль введены верно, то определяется статус пользователя. Затем открывается главная форма.

1. **ОПИСАНИЕ КОНТРОЛЬНОГО ПРИМЕРА**

Для проверки правильности работы программы нужно внести данные в таблицы, а затем поочередно выполнить все запросы и все отчеты. Нужно проверить, что результаты запросов и отчетов соответствуют введенным данным.

Для таблиц созданы формы для удобства добавления нового элемента

1. **СООБЩЕНИЯ ПРОГРАММЫ, ПРИЧИНЫ, ИХ ВЫЗЫВАЮЩИЕ, И РЕАКЦИЯ ПОЛЬЗОВАТЕЛЯ НА СООБЩЕНИЯ**

Могут возникать следующие сообщения системы: «Неверное имя!»

Причина – неверно указано имя пользователя, такого имени нет в таблице пользователей. Нужно внести правильное имя.

«Неверный пароль!»

Причина – имя пользователя указано правильно, но неверно указан пароль пользователя. Нужно внести правильный пароль.

**СПИСОК ЛИТЕРАТУРЫ**

1. Методические указания к курсовому проектированию по дисциплине “Организация баз данных и баз знаний” ля студентов специальности 7.080403/ Сост. С.Л. Зиноватная. – Одесса; ОНПУ, 2007. – ХХ с.

2. Конспект лекций по дисциплине “Организация баз данных и баз знаний”.

3. Основы современных баз данных. – Кузнецов С.Д. - Центр информационных технологий.