**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСТЕТ им. А.Н. ТУПОЛЕВА-КАИ»**

**Курсовая работа по предмету "Базы данных"**

**Вариант 8**

Выполнил: студент гр. 4281

Иванов Д.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверила:

Валитова Н.Л.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Вариант № 8. Предметная область – Спорт

БД должна хранить сведения о спортсменах и их спортивных достижениях: ФИО спортсмена, его год рождения, название команды, спортивный разряд спортсмена,

названия соревнований, места и даты проведения соревнований, результаты, показанные спортсменами на соревнованиях. Кроме того БД должна хранить сведения о видах спорта: название вида спорта, ед. измерения (метры, секунды, килограммы), мировой рекорд и дата мирового рекорда

**1. Системный анализ предметной области.**

*1.1. Состав и параметры объектов (сущностей) предметной области*

Предметная область имеет два логических объединения “Спортсмен” и “Вид спорта”.

Спортсмен имеет следующие сведения:

* ФИО спортсмена (текст)
* его год рождения (дата)
* название команды (текст)
* спортивный разряд спортсмена (текст)
* названия соревнований (текст)
* места (текст)
* даты проведения соревнований (дата начала – дата конца)
* результаты, показанные спортсменами на соревнованиях (int)
* Также сущности Спортсмен добавляется уникальный внутренний номер(int), т.к. ФИО и другие атрибуты могут совпадать (например участвуют в соревнованиях близнецы)

Вид спорта имеет следующие сведения:

* название вида спорта (текст)
* ед. измерения (метры, секунды, килограммы) (текст)
* мировой рекорд (текст)
* дата мирового рекорда (дата)

*1.2.Ограничения на информацию в системе.*

* Каждый спортсмен может заниматься только одним видом спорта, но каждым видом спорта может заниматься множество спортсменов
* Вид спорта не может иметь пустых сведений (все сведения должны быть заполнены).
* Каждый спортсмен может иметь только один результат на одном соревновании.
* Каждый спортсмен мог участвовать только в одном соревновании в определенное время, т.е. даты проведения различных соревнований, в которых участвовал спортсмен, не должны совпадать или пересекаться.

*1.3. Состав групп пользователей системы.*

С данной информационной системой должны работать следующие группы пользователей:

* администраторы ИС
* все другие пользователи

*1.4. Права доступа групп пользователей к информации в базе данных.*

Администратор ИС должен иметь право на создание/удаление/изменение/чтение БД

Пользователь имеет право только на чтение БД.

*1.5. Перечень задач каждой группы пользователей с кратким описанием алгоритмов их решения.*

При работе с системой администратор должен иметь возможность решать следующие задачи:

* Добавлять новых спортсменов и виды спорта
* Обновлять сведения мировом рекорде и дате мирового рекорда у видов спорта
* Обновлять все записи спортсменов.
* Удалять виды спорта и спортсменов
* Просматривать всех спортсменов и все виды спорта
* По выбранному спортсмену просмотреть вид спорта и наоборот
* По названию команды просматривать всех спортсменов, входящих в данную команду

Пользователь должен иметь возможность решать следующие задачи:

* Просматривать все виды спорта и всех спортсменов
* По выбранному спортсмену просмотреть к какой команде он относится
* По названию команды просматривать всех спортсменов, входящих в данную команду

*1.6. Описание входных документов.*

Для каждого спортсмена необходимо хранить след данные:

* ФИО спортсмена
* его год рождения
* название команды
* спортивный разряд спортсмена
* вид спорта
* названия соревнований
* места
* даты проведения соревнований
* уникальный внутренний номер

Для каждого вида спорта необходимо хранить следующие данные:

* название вида спорта
* ед. измерения (метры, секунды, килограммы)
* мировой рекорд
* дата мирового рекорда

*1.7. Описание выходных документов, которые должны генерироваться в системе.*

Выходными данными являются запросы. Информация выводится на экран в специальных формах, упрощающих работу с записями таблиц БД.

**2. Проектирования инфологической модели предметной области**

*2.1. Формирование перечня атрибутов предметной области*

2.1.1. Первоначальный перечень атрибутов.

У сущности “Спортсмен” следующие атрибуты:

* ФИО спортсмена (текст)
* его год рождения (дата)
* название команды (текст)
* спортивный разряд спортсмена (текст)
* названия соревнований (текст)
* места (текст)
* даты проведения соревнований (дата начала – дата конца)
* результаты, показанные спортсменами на соревнованиях (int)

У сущности “Вид спорта” следующие атрибуты:

* название вида спорта (текст)
* ед. измерения (метры, секунды, килограммы) (текст)
* мировой рекорд (текст)
* дата мирового рекорда (дата)

2.1.2. Уточнение перечня.

У сущностей “Спортсмен” и “Вид спорта” отсутствуют синонимы и омонимы.

2.1.3. Результат составления перечня

Для сущности “Спортсмен”:

|  |  |  |
| --- | --- | --- |
| **Обозначение атрибута** | **Имя атрибута** | **Примечание** |
| FIO | ФИО спортсмена | 3 слова |
| DOB | год рождения | дата |
| CNAME | название команды | 1 и более слов |
| RANG | спортивный разряд спортсмена | целое без знака |
| SNAME | названия соревнований | 1 и более слов |
| PLACE | место соревнований (город) | 1 и более слов |
| DBDE | дата проведения соревнований | 2 даты (начало и конец соревнований) |
| RESULT | результаты, показанные спортсменом на соревнованиях | целое без знака |

Для сущности “Вид спорта”:

|  |  |  |
| --- | --- | --- |
| **Обозначение атрибута** | **Имя атрибута** | **Примечание** |
| STYPE | название вида спорта | 1 и более слов |
| EDIZM | единица измерения | 1 и более слов |
| WRECORD | мировой рекорд | целое без знака |
| WRDATA | дата мирового рекорда | Int |

*2.2. Определение сущностей.*

2.2.1. Агрегация атрибутов в сущности.

|  |  |  |
| --- | --- | --- |
| **Обозначение сущности** | **Имя сущности** | **Список атрибутов** |
| athlete | Спортсмен | 1 .ФИО спортсмена  2. Его год рождения  3. Cпортивный разряд спортсмена  4. Название команды |
| competition | Соревнование | 1. Название соревнования  2. Место проведения  3. Даты проведения соревнований:  3.1. Дата начала  3.2. Дата окончания |
| result | Результат | 1. Результат, показанный спортсменом на соревновании |
| sport | Вид спорта | 1. Название вида спорта  2. Ед. Измерения  3. Мировой рекорд  4. Дата мирового рекорда |

2.2.2. Определение первичных ключей

*Правило целостности сущностей*. Атрибуты, входящие в состав некоторого ключа  
не могут принимать null-значений

У сущности Спортсмен ключом будет уникальный внутренний номер – этот атрибут не может принимать значение NULL

У сущности Вид спорта ключом будет название вида спорта – этот атрибут не может принимать значение NULL

Для сущности “Спортсмен” не существует атрибута, который можно взять за первичный ключ, поэтому добавим дополнительный атрибут “уникальный идентификатор спортсмена”:

Для сущности “Соревнование” первичным ключом будет “Название соревнования”.

Для сущности “Результат” – атрибут “уникальный идентификатор спортсмена”

Для сущности “Вид спорта” – атрибут “Название вида спорта”

|  |  |  |
| --- | --- | --- |
| **Сущность** | **Первичный ключ** | **Примечание** |
| Спортсмен | уникальный идентификатор спортсмена | НОВЫЙ |
| Соревнование | Название соревнования |  |
| Результат | Результат, показанный спортсменом на соревновании |  |
| Вид спорта | Название виде спорта |  |

2.2.3. Нормализация сущностей

Приведем cущность “ Соревнование ” к 1 нормальной форме, т.к. она содержит сложный атрибут “Даты проведения соревнований” – разобьём его на 2 простых:

“Дата начала” и “Дата окончания”.

Остальные сущности не имеют сложных атрибутов.

Полученые отношения уже находятся во 2 нормальной форме, т.к. каждый неключевой

атрибут функционально полно зависит от составного ключа. Приведём отношения к 3 НФ:

|  |  |  |
| --- | --- | --- |
| **Обозначение сущности** | **Имя сущности** | **Список атрибутов** |
| athlete | Спортсмен | 1. Номер спортсмена  2 .ФИО спортсмена  3. Его год рождения  4. Cпортивный разряд спортсмена  5. Название команды |
| competition | Соревнование | 1. Название соревнования  2. Место проведения  3. Дата начала  4. Дата окончания |
| result | Результат | 1. Номер спортсмена  2. Результат, показанный спортсменом на соревновании  3. Название соревнования |
| sport | Вид спорта | 1. Название вида спорта  2. Ед. Измерения  3. Мировой рекорд  4. Дата мирового рекорда |
| command | Команда | 1. Название команды  2. Название вида спорта |

2.2.4. Графическое представление сущностей

|  |
| --- |
| **Спортсмен** |
| Номер спортсмена |
| ФИО спортсмена |
| Его год рождения |
| Cпортивный разряд спортсмена |
| Название команды |

|  |
| --- |
| **Соревнование** |
| Название соревнования |
| Место проведения |
| Дата начала |
| Дата окончания |

|  |
| --- |
| **Результат** |
| Номер спортсмена |
| Результат, показанный спортсменом на соревновании |
| Название соревнования |
|  |

|  |
| --- |
| **Вид спорта** |
| Название вида спорта |
| Ед. Измерения |
| Мировой рекорд |
| Дата мирового рекорда |

|  |
| --- |
| **Команда** |
| Название команды |
| Название вида спорта |

*2.3. Определение связей*

|  |  |  |  |
| --- | --- | --- | --- |
| **Сущность 1** | **Сущность 2** | **Тип связи** | **Обязательность** |
| Результат | Спортсмен | М:1 | 0-1 |
| Результат | Соревнование | M:1 | 1-1 |
| Команда | Вид спорта | М:1 | 0-1 |
| Команда | Спортсмен | 1:M | 0-1 |

*2.4. Графическое изображение инфологической модели предметной области*

|  |
| --- |
| **Спортсмен** |
| Номер спортсмена |
| ФИО спортсмена |
| Его год рождения |
| Cпортивный разряд спортсмена |
| Название команды |

2.2.4. Графическое представление сущностей

|  |
| --- |
| **Соревнование** |
| Название соревнования |
| Место проведения |
| Дата начала |
| Дата окончания |

|  |
| --- |
| **Результат** |
| Номер спортсмена |
| Результат, показанный спортсменом на соревновании |
| Название соревнования |
|  |

|  |
| --- |
| **Вид спорта** |
| Название вида спорта |
| Ед. Измерения |
| Мировой рекорд |
| Дата мирового рекорда |

|  |
| --- |
| **Команда** |
| Название команды |
| Название вида спорта |

**3. Проектирования концептуальной модели**

*3.1. Преобразование сущностей в отношения.*

|  |
| --- |
| **athlete** |
| athlete\_ID |
| FIO |
| DOB |
| rang |
| cname |

|  |
| --- |
| **Спортсмен** |
| Номер спортсмена |
| ФИО спортсмена |
| Его год рождения |
| Cпортивный разряд спортсмена |
| Название команды |

|  |
| --- |
| **Соревнование** |
| Название соревнования |
| Место проведения |
| Дата начала |
| Дата окончания |

|  |
| --- |
| **competition** |
| sname |
| place |
| dbegin |
| dend |

|  |
| --- |
| **Результат** |
| Номер спортсмена |
| Результат, показанный спортсменом на соревновании |
| Название соревнования |
|  |

|  |
| --- |
| **result** |
| athlete\_ID |
| result |
| sname |
|  |

|  |
| --- |
| **Вид спорта** |
| Название вида спорта |
| Ед. Измерения |
| Мировой рекорд |
| Дата мирового рекорда |

|  |
| --- |
| **sport** |
| stype |
| edizm |
| wrecord |
| wrdata |

|  |
| --- |
| **Команда** |
| Название команды |
| Название вида спорта |

|  |
| --- |
| **command** |
| cname |
| stype |

*3.2. Определение свойств атрибутов отношений*

Свойства атрибутов отношения **athlete:**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип (размер)** | **Null** |
| athlete\_ID | Int | Not Null |
| FIO | Varchar(256) | Not Null |
| DOB | Date | Null |
| rang | Int | Null |
| cname | Varchar(256) | Null |

Свойства атрибутов отношения **competition:**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип (размер)** | **Null** |
| sname | Varchar(256) | Not Null |
| place | Varchar(256) | Null |
| dbegin | Date | Null |
| dend | Date | Null |

Свойства атрибутов отношения **result:**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип (размер)** | **Null** |
| athlete\_ID | Int | Not Null |
| result | Int | Null |
| sname | Varchar(256) | Not Null |

Свойства атрибутов отношения **sport:**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип (размер)** | **Null** |
| stype | Varchar(256) | Not Null |
| edizm | Varchar(10) | Null |
| wrecord | Int | Null |
| wrdata | Int | Null |

Свойства атрибутов отношения **command:**

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип (размер)** | **Null** |
| cname | Varchar(256) | Not Null |
| stype | Varchar(256) | Null |

*3.3. Определение внешних ключей*

*Правило целостности внешних ключей*. Внешние ключи не должны быть  
несогласованными, т.е. для каждого значения внешнего ключа должно  
существовать соответствующее значение ключевого атрибута в родительском  
отношении

В сущностях Спортсмен и Вид спорта внешними ключами будут название вида спорта, сущности Спортсмен нужно будет добавить атрибут название вида спорта.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Дочернее отношение** | **Внешний ключ** | **Тип (размер)** | **NULL** | **Ссылка** | **Примечание** |
| command | stype | Varchar(256) | NULL | sport:stype |  |
| athlete | cname | Varchar(256) | NULL | command:cname |  |
| result | athlete\_ID | Int | Not Null | athlete:athlete\_ID |  |
| sname | Varchar(256) | Not Null | competition:sname |  |

*3.4. Реализация связей типа «многие-ко-многим»*

Отсутствуют.

*3.5. Графическое изображение концептуальной модели предметной области*

|  |  |  |
| --- | --- | --- |
| **competition** | | |
| sname | Varchar(256) | Not Null |
| place | Varchar(256) | Null |
| dbegin | Datetime | Null |
| dend | Datetime | Null |

|  |  |  |
| --- | --- | --- |
| **athlete** | | |
| athlete\_ID | Int | Not Null |
| FIO | Varchar(256) | Not Null |
| DOB | Datetime | Null |
| rang | Int | Null |
| cname | Varchar(256) | Null |

|  |  |  |
| --- | --- | --- |
| **result** | | |
| athlete\_ID | Int | Not Null |
| result | Int | Null |
| sname | Varchar(256) | Not Null |

|  |  |  |
| --- | --- | --- |
| **command** | | |
| cname | Varchar(256) | Not Null |
| stype | Varchar(256) | Null |

|  |  |  |
| --- | --- | --- |
| **sport** | | |
| stype | Varchar(256) | Not Null |
| edizm | Varchar(10) | Null |
| wrecord | Int | Null |
| wrdata | Int | Null  **command.** stype= **sport.** stype |

**athlete.** cname= **command.** cname

**result .** athlete\_ID **=athlete.** athlete\_ID

**result.** sname **=competition.** sname

**4. Листинг лабораторных работ.**

*4.1. Лабораторная работа 1.*

*USE master --использовать БД master*

*GO*

*--создание БД с параметрами БД и лог-файла*

*IF DB\_ID('MyDB') IS NULL CREATE DATABASE MyDB*

*ON*

*(*

*NAME='MyDB\_dat',*

*FILENAME='c:\Users\USER\Desktop\Четвертый семестр\Базы данных\LR1\MyDB\_dat.mdf',*

*SIZE=4MB,*

*MAXSIZE=10MB,*

*FILEGROWTH=1MB*

*)*

*LOG ON*

*(*

*NAME='MyDB\_log',*

*FILENAME ='c:\Users\USER\Desktop\Четвертый семестр\Базы данных\LR1\MyDB\_log.ldf',*

*SIZE=2MB,*

*MAXSIZE=5MB,*

*FILEGROWTH=1MB*

*)*

*GO -- выполнить*

*--Сообщает информацию об указанной базе данных*

*EXEC sp\_helpdb MyDB*

*--проверятся наличие файла MyDB\_dat в БД MyDB, если такой файл есть - то меняем размер файла БД на ---15MB*

*IF (SELECT name FROM sys.master\_files WHERE database\_id = DB\_ID('MyDB') AND name='MyDB\_dat') IS NOT NULL ALTER DATABASE MyDB MODIFY FILE(*

*NAME='MyDB\_dat',*

*SIZE=15MB*

*)*

*GO*

*--удаляются снапшот и БД если есть*

*IF DB\_ID('MyDB\_snapshot') IS NOT NULL DROP DATABASE MyDB\_snapshot*

*IF DB\_ID('MyDB') IS NOT NULL DROP DATABASE MyDB*

*GO*

*--создается БД MyDB если её нет*

*IF DB\_ID('MyDB') IS NULL CREATE DATABASE MyDB*

*GO*

*--подклюяаемся к БД MyDB*

*USE MyDB*

*GO*

*-- если в БД MyDB нет таблицы table1, то создается таблица table1*

*IF OBJECT\_ID('dbo.table1') IS NULL CREATE TABLE table1 (*

*id int NOT NULL IDENTITY(1,1),*

*fio varchar(20) NULL,*

*datar datetime,*

*PRIMARY KEY(id)*

*)*

*GO*

*--ставим файлу БД MyDB атрибут только для чтения*

*ALTER DATABASE MyDB SET READ\_ONLY*

*--пытаемся добавить запись в таблицу - ошибка*

*--INSERT INTO dbo.table1 VALUES('family name',GETDATE())*

*--ставим файлу БД MyDB атрибут для чтения и записи*

*ALTER DATABASE MyDB SET READ\_WRITE*

*--добавляем запись в таблицу*

*INSERT INTO dbo.table1 VALUES('family name',GETDATE())*

*GO*

*--выборка всего из таблицы table1*

*SELECT \* FROM dbo.table1*

*GO*

*--Файлы базы данных объединены в группы (filegroups).*

*--добавляем новую группу файлов к БД MyDB под названием new\_filegroup с параметрами по умолчанию*

*ALTER DATABASE MyDB ADD FILEGROUP new\_filegroup*

*GO*

*--меняем параметры new\_filegroup*

*ALTER DATABASE MyDB ADD FILE (*

*NAME='MyDB\_new\_dat',*

*FILENAME='c:\Users\USER\Desktop\Четвертый семестр\Базы данных\LR1\MyDB\_new\_dat.mdf',*

*SIZE=4MB,*

*MAXSIZE=10MB,*

*FILEGROWTH=1MB*

*) TO FILEGROUP new\_filegroup*

*GO*

*--делаем новую группу дефолтной*

*ALTER DATABASE MyDB MODIFY FILEGROUP new\_filegroup DEFAULT*

*GO*

*--создание снапшота (т.к. в БД 2 файловые группы - 1 дефолтная (название файла = названию БД, т.к. параметры*

*--при создании не прописывались), а вторая new\_filegroup, которую добавили выше) - то указываем все 2*

*--имени файлов (NAME=...) БД и расположения (FILENAME=...) куда будут скопированы снапшоты*

*CREATE DATABASE MyDB\_snapshot ON*

*(*

*NAME='MyDB\_new\_dat',*

*FILENAME='c:\Users\USER\Desktop\Четвертый семестр\Базы данных\LR1\SNAPSHOT\Snapshot\_MyDB\_new\_dat.mdf'*

*),(*

*NAME='MyDB',*

*FILENAME='c:\Users\USER\Desktop\Четвертый семестр\Базы данных\LR1\SNAPSHOT\Snapshot\_MyDB.mdf'*

*) AS SNAPSHOT OF MyDB*

*GO*

*--выборка первых записей из table1 БД MyDB и MyDB\_snapshot*

*SELECT TOP 1 \* FROM MyDB.dbo.table1 TEST*

*SELECT TOP 1 \* FROM MyDB\_snapshot.dbo.table1 TEST*

*--сравнение 2-х таблиц*

*IF(SELECT COUNT(\*) FROM MyDB.dbo.table1 AS A ,MyDB\_snapshot.dbo.table1 AS B WHERE A.id=B.id AND A.fio=B.fio AND A.datar=B.datar)>0*

*BEGIN*

*print N'snapshot=original'*

*END*

*ELSE*

*BEGIN*

*print N'snapshot!=original'*

*END*

*--апдейтим 1-ую запись таблицы table1 БД MyDB*

*UPDATE MyDB.dbo.table1 SET fio='new FIO' WHERE MyDB.dbo.table1.id=(SELECT top 1 id FROM MyDB.dbo.table1)*

*--сравнение 2-х таблиц*

*IF(SELECT COUNT(\*) FROM MyDB.dbo.table1 AS A ,MyDB\_snapshot.dbo.table1 AS B WHERE A.id=B.id AND A.fio=B.fio AND A.datar=B.datar)>0*

*BEGIN*

*print N'snapshot=original'*

*END*

*ELSE*

*BEGIN*

*print N'snapshot!=original'*

*END*

*--делетим снапшот и БД*

*USE master*

*GO*

*IF DB\_ID('MyDB\_snapshot') IS NOT NULL DROP DATABASE MyDB\_snapshot*

*IF DB\_ID('MyDB') IS NOT NULL DROP DATABASE MyDB*

*4.2. Лабораторная работа 2.*

*USE master*

*GO*

*--создаем БД BookshopDB*

*IF DB\_ID('BookshopDB') IS NULL CREATE DATABASE BookshopDB*

*GO*

*EXEC sp\_helpdb BookshopDB*

*USE BookshopDB*

*GO*

*--создаем таблицы в БД с первичными/внешними ключами*

*IF OBJECT\_ID('dbo.Authors')IS NULL CREATE TABLE Authors(*

*AuthorID int not null IDENTITY,*

*FirstName varchar(30) not null DEFAULT 'unknown',*

*LastName varchar(30) null,*

*YearBorn char(4) null,*

*YearDied char(4) not null DEFAULT 'no',*

*PRIMARY KEY(AuthorID)*

*)*

*GO*

*EXEC sp\_help Authors*

*IF COL\_LENGTH('dbo.Authors', 'Descr') IS NULL*

*BEGIN*

*ALTER TABLE Authors ADD Descr varchar(200) NOT NULL DEFAULT 'empty'*

*END*

*IF OBJECT\_ID('dbo.Books')IS NULL CREATE TABLE Books(*

*BookID int not null IDENTITY,*

*Title varchar(100) not null,*

*Janr varchar(50) null,*

*Primary key(BookID)*

*)*

*GO*

*IF OBJECT\_ID('dbo.BookAuthor')IS NULL*

*--создаем таблицу BookAuthor,редактируем таблицу, добавляем первичный и внешний ключи, а также ограничения*

*BEGIN*

*CREATE TABLE BookAuthor(*

*BookID int not null,*

*AuthorID int not null*

*)*

*ALTER TABLE BookAuthor ADD PRIMARY KEY (BookID,AuthorID)*

*ALTER TABLE BookAuthor ADD CONSTRAINT BookAuthor\_BookID\_const FOREIGN KEY (BookID) REFERENCES Books(BookID)*

*ALTER TABLE BookAuthor ADD CONSTRAINT BookAuthor\_Authors\_const FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)*

*ALTER TABLE Authors ADD CONSTRAINT CK\_Authors\_YearBorn CHECK (YearBorn LIKE '[1-2][0,6-9][0-9][0-9]')*

*ALTER TABLE Authors ADD CONSTRAINT CK\_Authors\_YearDied CHECK ((YearDied LIKE '[1-2][0,6-9][0-9][0-9]' OR*

*YearDied='no') AND YearDied>YearBorn)*

*END*

*GO*

*--заполняем таблицы*

*INSERT INTO dbo.Authors VALUES('Mihail','Bulgakov','1891','1940','some description'),*

*('Lev','Tolstoy','1828','1910','some description'),*

*('Zahar','Prilepin','1975','no','some description')*

*INSERT INTO dbo.Books VALUES('Master and Margarita','Roman'),*

*('The war and the peace','Roman'),*

*('Черная обезьяна','роман')*

*GO*

*INSERT INTO dbo.BookAuthor VALUES(1,1),(2,2),(3,3)*

*GO*

*4.3. Лабораторная работа 3.*

*USE master*

*GO*

*--создаем БД Sport*

*IF DB\_ID('Sport') IS NULL CREATE DATABASE Sport*

*GO*

*USE Sport*

*GO*

*--создаем таблицы в БД с ограничениями и первичными/внешними ключами*

*IF OBJECT\_ID('dbo.sport') IS NULL CREATE TABLE sport(*

*stype Varchar(256) Not Null,*

*edizm Varchar(10) Null,*

*wrecord Int Null,*

*wrdata Int Null,*

*PRIMARY KEY(stype),*

*CHECK (wrdata<=YEAR(getdate()) AND wrdata>1000 AND wrecord>0)*

*)*

*IF OBJECT\_ID('dbo.command') IS NULL CREATE TABLE command(*

*cname Varchar(256) Not Null,*

*stype Varchar(256) Null,*

*PRIMARY KEY(cname),*

*FOREIGN KEY (stype) REFERENCES sport(stype)*

*)*

*IF OBJECT\_ID('dbo.athlete') IS NULL CREATE TABLE athlete(*

*athlete\_ID Int Not Null IDENTITY,*

*FIO Varchar(256) Not Null,*

*DOB Datetime Null,*

*rang Int Null,*

*cname Varchar(256) Null,*

*PRIMARY KEY(athlete\_ID),*

*FOREIGN KEY (cname) REFERENCES command(cname),*

*CHECK (YEAR(DOB)<=YEAR(getdate()) AND FIO LIKE '% % %')*

*)*

*IF OBJECT\_ID('dbo.competition') IS NULL CREATE TABLE competition(*

*sname Varchar(256) Not Null,*

*place Varchar(256) Null,*

*dbegin Datetime Null,*

*dend Datetime Null,*

*PRIMARY KEY(sname),*

*CHECK (YEAR(dbegin)<=YEAR(getdate()) AND YEAR(dend)<=YEAR(getdate()) AND dbegin<dend)*

*)*

*IF OBJECT\_ID('dbo.result') IS NULL CREATE TABLE result(*

*athlete\_ID Int Not Null,*

*result Int Null,*

*sname Varchar(256) Not Null,*

*PRIMARY KEY(athlete\_ID),*

*FOREIGN KEY (athlete\_ID) REFERENCES athlete(athlete\_ID),*

*FOREIGN KEY (sname) REFERENCES competition(sname)*

*)*

*GO*

*--Сообщает информацию об указанных БД*

*EXEC sp\_help sport*

*EXEC sp\_help command*

*EXEC sp\_help athlete*

*EXEC sp\_help competition*

*EXEC sp\_help result*

*--заполняем таблицы*

*INSERT INTO sport VALUES('Бег на 100 м','сек',9.58,2009),*

*('Метание молота','м',79.42,2011),*

*('Прыжки с трамплина','м',246.5,2011)*

*INSERT INTO command VALUES('Ямайка','Бег на 100 м'),*

*('Германия','Метание молота'),*

*('Россия','Прыжки с трамплина')*

*INSERT INTO athlete VALUES('Джереми Уотерспун Отчество',convert(datetime, '10/23/1980', 101),1,'Ямайка'),*

*('Уве Хон Отчество',convert(datetime, '10/23/1984', 101),2,'Германия'),*

*('Иванов Иван Отчество',convert(datetime, '10/23/1990', 101),3,'Россия')*

*INSERT INTO competition VALUES('Прыжки с трамплина','Рио Де Жанейро',convert(datetime, '10/23/2014', 101),convert(datetime, '11/23/2014', 101)),*

*('Метание молота','Мюнхен',convert(datetime, '3/11/2000', 101),convert(datetime, '4/11/2000', 101))*

*INSERT INTO result VALUES ((SELECT athlete\_ID FROM athlete WHERE FIO='Иванов Иван Отчество'),1,'Прыжки с трамплина'),*

*((SELECT athlete\_ID FROM athlete WHERE FIO='Уве Хон Отчество'),10,'Метание молота'),*

*((SELECT athlete\_ID FROM athlete WHERE FIO='Джереми Уотерспун Отчество'),1,'Прыжки с трамплина')*

*4.4. Лабораторная работа 4.*

*USE master*

*GO*

*--создаем БД Sport*

*IF DB\_ID('Sport') IS NULL CREATE DATABASE Sport*

*GO*

*USE Sport*

*GO*

*--создаем таблицы в БД с ограничениями и первичными/внешними ключами*

*IF OBJECT\_ID('dbo.sport') IS NULL CREATE TABLE sport(*

*stype Varchar(256) Not Null,*

*edizm Varchar(10) Null,*

*wrecord Int Null,*

*wrdata Int Null,*

*PRIMARY KEY(stype),*

*CHECK (wrdata<=YEAR(getdate()) AND wrdata>1000 AND wrecord>0)*

*)*

*IF OBJECT\_ID('dbo.command') IS NULL CREATE TABLE command(*

*cname Varchar(256) Not Null,*

*stype Varchar(256) Null,*

*PRIMARY KEY(cname),*

*FOREIGN KEY (stype) REFERENCES sport(stype)*

*)*

*IF OBJECT\_ID('dbo.athlete') IS NULL CREATE TABLE athlete(*

*athlete\_ID Int Not Null IDENTITY,*

*FIO Varchar(256) Not Null,*

*DOB Datetime Null,*

*rang Int Null,*

*cname Varchar(256) Null,*

*PRIMARY KEY(athlete\_ID),*

*FOREIGN KEY (cname) REFERENCES command(cname),*

*CHECK (YEAR(DOB)<=YEAR(getdate()) AND FIO LIKE '% % %')*

*)*

*IF OBJECT\_ID('dbo.competition') IS NULL CREATE TABLE competition(*

*sname Varchar(256) Not Null,*

*place Varchar(256) Null,*

*dbegin Datetime Null,*

*dend Datetime Null,*

*PRIMARY KEY(sname),*

*CHECK (YEAR(dbegin)<=YEAR(getdate()) AND YEAR(dend)<=YEAR(getdate()) AND dbegin<dend)*

*)*

*IF OBJECT\_ID('dbo.result') IS NULL CREATE TABLE result(*

*athlete\_ID Int Not Null,*

*result Int Null,*

*sname Varchar(256) Not Null,*

*PRIMARY KEY(athlete\_ID),*

*FOREIGN KEY (athlete\_ID) REFERENCES athlete(athlete\_ID),*

*FOREIGN KEY (sname) REFERENCES competition(sname)*

*)*

*GO*

*--заполняем таблицы*

*INSERT INTO sport VALUES('Бег на 100 м','сек',9.58,2009),*

*('Метание молота','м',79.42,2011),*

*('Прыжки с трамплина','м',246.5,2011)*

*INSERT INTO command VALUES('Ямайка','Бег на 100 м'),*

*('Германия','Метание молота'),*

*('Россия','Прыжки с трамплина')*

*INSERT INTO athlete VALUES('Джереми Уотерспун Отчество',convert(datetime, '10/23/1980', 101),1,'Ямайка'),*

*('Уве Хон Отчество',convert(datetime, '10/23/1984', 101),2,'Германия'),*

*('Иванов Иван Отчество',convert(datetime, '10/23/1990', 101),3,'Россия')*

*INSERT INTO competition VALUES('Прыжки с трамплина','Рио Де Жанейро',convert(datetime, '10/23/2014', 101),convert(datetime, '11/23/2014', 101)),*

*('Метание молота','Мюнхен',convert(datetime, '3/11/2000', 101),convert(datetime, '4/11/2000', 101))*

*INSERT INTO result VALUES ((SELECT athlete\_ID FROM athlete WHERE FIO='Иванов Иван Отчество'),1,'Прыжки с трамплина'),*

*((SELECT athlete\_ID FROM athlete WHERE FIO='Уве Хон Отчество'),10,'Метание молота'),*

*((SELECT athlete\_ID FROM athlete WHERE FIO='Джереми Уотерспун Отчество'),1,'Прыжки с трамплина')*

*--запросы из методички*

*--1*

*INSERT INTO sport (stype,edizm) VALUES*

*('Становая тяга','кг'),*

*('Плавание брасом','сек')*

*--2*

*DELETE FROM sport WHERE stype='Становая тяга'*

*--3*

*UPDATE sport SET wrecord=200,wrdata=1980 WHERE stype='Плавание брасом'*

*--4) запрос на выборку записей:*

*SELECT \* FROM sport WHERE wrdata>1985*

*--5) запрос на объединение таблиц:*

*SELECT \* FROM competition UNION SELECT \* FROM sport*

*--6) запрос на пересечение таблиц:*

*SELECT \* FROM result WHERE result.athlete\_ID IN (SELECT athlete\_ID FROM athlete)*

*--7) запрос на разность таблиц:*

*SELECT \* FROM sport WHERE sport.stype NOT IN (SELECT stype FROM command)*

*--8) запрос-декартово произведение таблиц:*

*SELECT \* FROM sport,command*

*--9) запрос-проекция таблицы*

*SELECT DISTINCT FIO ,DOB ,rang FROM athlete*

*--10) запрос-соединение таблиц*

*SELECT \* FROM athlete,sport,command WHERE athlete.cname=command.cname AND sport.stype=command.stype*

*--Мои запросы*

*SELECT \* FROM athlete WHERE athlete.athlete\_ID IN (SELECT athlete\_ID FROM result) AND athlete.cname IN (SELECT cname FROM command)*

*SELECT FIO FROM athlete AS A WHERE A.rang>1 UNION SELECT sname FROM competition AS C WHERE C.sname IN (SELECT cname FROM command)*

*SELECT DISTINCT FIO ,DOB ,rang FROM athlete AS A WHERE A.rang>1 AND A.cname IN (SELECT cname FROM command AS C WHERE C.stype IN (SELECT stype FROM sport))*